

**Emanzipation der Nutzer, Web 2.0 und Topic Maps –
Analyse einer Evolution und Entwicklung eines Tagging-
Modells mit Topic Maps unter Verwendung einer
eigenentwickelten PHPTMAPI-Implementierung**

Hausarbeit
zur Diplomprüfung

an der
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Fakultät Design, Medien und Information
Studiendepartment Information

vorgelegt von
Johannes Schmidt
Hamburg, Dezember 2006

Referent: Prof. Dr. Franziskus Geeb
Korreferent: Prof. Dr. Ulrich Hofmann

Abstract

In dieser Diplomarbeit wird ein Vorschlag erarbeitet, um die Effizienz und Effektivität von Informationsaustauschprozessen im Kontext des *Taggings* zu erhöhen.

Ausgehend von der Untersuchung der Bedeutung der Information in der Internetökonomie wird zunächst in der Analyse der Rolle und Position der Nachfrager festgestellt, dass sich diese bzgl. der Organisation von Informationsaustauschprozessen im Internet gegenüber den Unternehmen emanzipieren. Dieses Phänomen wird in den Kontext weiterer wahrnehmbarer Entwicklungen eingeordnet, die aktuell im Zusammenhang mit „Web 2.0“ diskutiert werden. Der intensive Austausch von Daten zwischen diesen Anwendungen über RSS oder offen zugängliche Schnittstellen ist dabei ein zentrales Prinzip. Das kollaborative Verschlagworten von Ressourcen mit sog. *Tags*, das *Tagging*, als ein Merkmal spezifischer Anwendungen wird eingehend untersucht. Es wird festgestellt, dass jenen Anwendungen ein gleichartiges konzeptionelles Modell zu Grunde liegt, das die Konzepte „Nutzer“, „Ressource“ und „Tag“ strukturiert und hier in dieser Arbeit als *conceptual tagging model* bezeichnet wird. Die Analyse der über RSS bereitgestellten Formate und der Schnittstellen zeigt jedoch, dass diese uneinheitlich definiert sind und lediglich eine vage Semantik etablieren können.

Der ISO/IEC-Standard 13250 Topic Maps bezeichnet eine Technologie zur Informationsstrukturierung und Wissensrepräsentation, die auch die Verknüpfung relevanter Ressourcen ermöglicht. Ausgehend vom *conceptual tagging model* wird das Topic Maps-Tagging-Modell erarbeitet. Mit einer eigenentwickelten Implementierung von PHPTMAPI, der standardisierten Schnittstelle für die Bearbeitung von Topic Maps in PHP, werden die theoretischen Überlegungen über die Implementierung und Anwendung des Modells praktisch überprüft. Die Verwendung des Topic Maps-Tagging-Modells zeigt das semantische Potential von Topic Maps und die Möglichkeit zur Verbesserung des Datenaustausches über Anwendungsgrenzen hinweg.

Schlagworte

Content-Syndication, Datenbanken, Indexierung, Information, Internet, Internetökonomie, Metadaten, MySQL, Ontologie, PHP, PHPTMAPI, PHPTMAPI-Implementierung, Tagging, TMAPI, Topic Maps, Web 2.0, XTM

Inhaltsverzeichnis

Abstract	iii
Anhangverzeichnis	vii
Abbildungsverzeichnis	viii
Verzeichnis der benutzten Abkürzungen	x
1 Ausgangssituation und Problemstellung	1
1.1 Zielformulierung	2
1.2 Vorgehensweise	3
2 Informationsgesellschaft	4
2.1 Internetökonomie	7
2.1.1 Transparenz der Märkte	11
2.1.2 Konvergenz	14
2.2 Die Bedeutung der Information aus der ökonomischen Perspektive	18
3 Rolle und Position der Nachfrager	23
3.1 Nachfrager als Informationsquelle	25
3.2 Virtuelle Gemeinschaften: Nachfrager und „Prosumer“	28
3.3 Emanzipation der Nutzer	31
4 Paradigmenwechsel: Web 2.0?	35
4.1 Strategische Positionierung: Das Internet als Plattform	38
4.2 Engagement der Nutzer: kollektive Intelligenz	41
4.3 Erste Bewertung der Diskussion	44
4.4 Das Tagging: kollektive Metadaten	45
4.4.1 Analyse des Prinzips	46
4.4.2 Abstraktion des Kontextes auf Modellebene	50
4.4.3 Analyse der Praxis	53
4.4.4 Konklusion	60

5	Topic Maps	65
5.1	Das „TAO“: Topic, Association, Occurrence	67
5.2	Identität von Aussagegegenständen	70
5.3	Das Topic Maps Data Model	73
5.4	Syntax XTM	79
5.5	TMAPI	85
5.6	PHPTMAPI	86
5.6.1	Exkurs: PHP 5	88
5.6.2	PHPTMAPI Kernpaket	91
5.6.3	Unterschiede zu TMAPI	98
5.6.4	Basiserweiterung: PHPTMAPI Indexpaket	100
5.6.5	Bewertung und Ausblick	102
5.7	Implementierung von PHPTMAPI	104
5.7.1	Anforderungen an eine Implementierung und resultierende Ergebnisse	106
5.7.2	Datenmodell und Datenbankschema der Implementierung	112
5.7.3	Bewertung des Status quo	122
6	Tagging-Modell	125
6.1	Definition des Modells: Entwicklung einer Ontologie	125
6.1.1	Etablieren der Semantik: Published Subject Identifiers und Published Subject Indicators	132
6.1.2	Spezifikation des Tagging-Modells	135
6.2	Implementierung des Modells	138
6.3	Anwendung des Modells	142
7	Bewertung und Fazit	147
	Literatur- und Quellenverzeichnis	152
	Anhang A	A-1
	Anhang B	B-1
	Anhang C	C-1
	Anhang D	D-1
	Anhang E	E-1
	Anhang F	F-1
	Anhang G	G-1
	Anhang H	H-1
	Anhang I	I-1

Anhang J	J-1
Anhang K	K-1
Anhang L	L-1
Anhang M	M-1
Eidesstattliche Versicherung	xii

Anhangverzeichnis

Anhang A – E-Mail von Lars Heuer über die TMAPI-E-Mailliste zur Klärung der Frage der Notwendigkeit des Erstellens eines Topic C beim Merging von zwei Topics A und B	A-1
Anhang B – UML Klassendiagramm für das PHPTMAPI Kernpaket (core)	B-1
Anhang C – UML Klassendiagramm der Exceptions in PHPTMAPI	C-1
Anhang D – UML Klassendiagramm für das PHPTMAPI Indexpaket (index, index.core)	D-1
Anhang E – Poster zur Präsentation von PHPTMAPI auf der „Conferences on Topic Maps Research and Applications“, TMRA, am 11.10.2006 in Leipzig	E-1
Anhang F – UML Klassendiagramm (informativ) als Resultat der durch die Abbildung der DTD von XTM 2.0 auf ein Objektschema resultierenden Klassen nach der Methode des <i>object-relational mappings</i>	F-1
Anhang G – Revers erstelltes Entity-Relationship-Diagramm in Crows-Foot-Notation aus der Datenbank der PHPTMAPI-Implementierung	G-1
Anhang H – CREATE TABLE-Anweisungen zur Definition der Tabellen für die MySQL-Datenbank der PHPTMAPI-Implementierung mit InnoDB als <i>storage engine</i>	H-1
Anhang I – Published Subject Indicators in XTM 1.0 für die zentralen Konzepte des Topic Maps-Tagging-Modells	I-1
Anhang J – Spezifikation des Topic Maps-Tagging-Modells in XTM 1.0	J-1
Anhang K – Installationshilfe für die PHPTMAPI-Implementierung (QuaaxTM)	K-1
Anhang L – E-Mail von Graham Moore über die englischsprachige Topic Maps-E-Mailliste zum weiteren Vorgehen bzgl. TMWSI	L-1
Anhang M – CD-ROM, enthält PHPTMAPI 1.0, PHPTMAPI-Implementierung (QuaaxTM), XTM-Beispiele	M-1

Abbildungsverzeichnis

Abbildung 1: Beziehungen der Ebenen der Begriffshierarchie Zeichen, Daten, Information, Wissen	5
Abbildung 2: Charakteristika der Internetökonomie	8
Abbildung 3: Preisvergleich für das VAIO VGN-BX196VP von Sony und die IXUS 55 von Canon mit kelkoo	11
Abbildung 4: Konvergenz von Telekommunikations-, IT- und Medienbranche	17
Abbildung 5: Innerbetriebliche Steuerungsfunktion produktionsfaktoradhärenter Informationen	19
Abbildung 6: Bedürfnispyramide nach Maslow	23
Abbildung 7: Durchschnittliche Arbeitszeit in Deutschland für alle Arbeitnehmer	24
Abbildung 8: Lebenserwartung für Frauen und Männer in Deutschland	25
Abbildung 9: Multimedia-Wertschöpfungsnetzwerk	26
Abbildung 10: „Meme map“ Web 2.0	37
Abbildung 11: Darstellung aller Tags des Autors als <i>tag cloud</i> bei del.icio.us	47
Abbildung 12: Kognitiver Prozess beim Tagging	49
Abbildung 13: Kognitiver Prozess beim Kategorisieren	49
Abbildung 14: Rudimentäres ERD des Tagging-Kontextes	53
Abbildung 15: IST-SOLL-Vergleich für Systeme, die den Zugriff auf die durch die zentralen Konzepte <i>tag</i> , <i>tagging user</i> und <i>taggable resource</i> beschriebene Semantik des Taggings erlauben	63
Abbildung 16: Topic und Topic Type	68
Abbildung 17: Topic und Topic Type, Occurrence und Occurrence Type	68
Abbildung 18: Visualisierung einer Association unter Verwendung eines Aussagenmodells im Draft Topic Maps Reference Model, Revision 1.0	70
Abbildung 19: Ebenen und Bestandteile von ISO/IEC 13250 Topic Maps	74
Abbildung 20: Metamodell des Topic Maps Data Model	75

Abbildung 21: Topic Item	76
Abbildung 22: Zeitlicher Entwicklungszusammenhang zwischen HyTM und XTM 1.0	80
Abbildung 23: PHPTMAPI Kernpaket (core)	91
Abbildung 24: PHPTMAPI Exceptions	97
Abbildung 25: PHPTMAPI Indexpaket (index, index.core)	100
Abbildung 26: Beziehungsdreieck zwischen TMDM, Topic Maps-Tagging-Modell und Anwendung des implementierten Topic Maps-Tagging-Modells	130
Abbildung 27: Visualisierung der Association zwischen <i>tag</i> und <i>taggable resource</i> unter Verwendung eines Aussagenmodells im Draft Topic Maps Reference Model, Revision 1.0	137
Abbildung 28: Darstellung des Topics <i>tag</i> innerhalb des Topic Maps-Tagging-Modells mit dem Topic Maps-Browser „Omnigator“	138
Abbildung 29: Darstellung einer Ressource in einer Anwendung des Topic Maps-Tagging-Modells mit dem Topic Maps-Browser „Omnigator“	144
Abbildung 30: Grafische Darstellung einer Ressource in einer Anwendung des Topic Maps-Tagging-Modells mit „Vizigator“	144
Abbildung 31: Darstellung der Associations in „Omnigator“ für die Ressource unter http://www.topicmaps.org nach Mehrfachzuweisung eines identischen Tags	145
Abbildung 32: Darstellung der Associations in „Vizigator“ für die Ressource unter http://www.topicmaps.org nach Mehrfachzuweisung eines identischen Tags	145
Abbildung 33: Skizze eines Remote Access-fähigen Topic Maps-Systems für das Topic Maps-Tagging-Modell und Anwendung mit PHPTMAPI-Implementierungen	149

Verzeichnis der benutzten Abkürzungen

ABI	Application Binary Interface
API	Application Programming Interface
bspw.	beispielsweise
bzgl.	bezüglich
Darst.	Darstellung
DC	Dublin Core
DNS	Domain Name System
DOM	Document Object Model
DTD	Document Type Definition
ebd.	ebenda
ERD	Entity Relationship Diagramm
ERM	Entity Relationship Modell
et al.	et alii
etc.	et cetera
FTP	File Transfer Protocol
GNU	GNU's Not UNIX
GPL	GNU General Public License
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HyTM	HyTime Topic Maps
i. a.	im allgemeinen
i. d. R.	in der Regel
IRI	Internationalized Resource Identifiers
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
o. g.	oben genannt, -e, -es, -en
PHP	PHP: Hypertext Preprocessor
PHPTMAPI	PHP TMAPI
PSI	Published Subject Indicator, Published Subject Identifier
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
REST	Representational State Transfer
RM	(Topic Maps) Reference Model
RSS	Really Simple Syndication, RDF Site Summary
SAM	(Topic Maps) Standard Application Model
SAX	Simple API for XML
SGML	Structured General Markup Language

SLUO	Subject Location Uniqueness Objective
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
s. u.	siehe unten
TMAPI	Topic Maps API
TMDM	Topic Maps Data Model
TMIP	Topic Maps Interaction Protocol
TMQL	Topic Maps Query Language
TMRAP	Topic Maps Remote Access Protocol
TMWSI	Topic Maps Web Service Interface
u. a.	und andere
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
v. a.	vor allem
vgl.	vergleiche
W3C	World Wide Web Consortium
XLink	XML Linking Language
XML	Extensible Markup Language
XML-RPC	XML Remote Procedure Calls
XTM	XML Topic Maps
z. B.	zum Beispiel
z. Zt.	zur Zeit

Statt sich auf eine riesige alexandrinische Bibliothek hinzubewegen, ist die Welt ein Computer geworden, ein elektronisches Gehirn, wie wir das in einem kindischen Zukunftsroman lesen können. (McLuhan 1995, S. 40)

1 Ausgangssituation und Problemstellung

Die Informations- und Kommunikationstechnologie sowie elektronische Netze auf Basis von TCP/IP als hervorzuhebende Komponente haben nachhaltige Veränderungen in unserer Gesellschaft ausgelöst. Signifikante Umwälzungen können dabei in der Ökonomie registriert werden. Die Wirkung ist hier vor allem in den Feldern deutlich, in denen Information und Kommunikation eine wesentliche Rolle spielen. Dabei sind Information und Kommunikation allgegenwärtige Konzepte, die neben der Ökonomie gleichermaßen auch Politik und Gesellschaft betreffen und dort ebenfalls ihr Veränderungspotential entfalten.

Elektronische Netze ermöglichen Unternehmen neuartige Kooperationsformen und Geschäftsmodelle sowie die Entstehung neuer transparenter Märkte als Resultat einer Konvergenz der Computer-, Medien- und Telekommunikationsbranche.

Auf Seiten der Nachfrager setzt sich der Trend zur Individualisierung weiter fort, was zu einer zunehmenden Fragmentierung der Märkte führt. Eine allseits wahrnehmbare jüngere Entwicklung ist die, dass Privatpersonen, die Nutzer bzw. User, selbstinitiativ digitale Inhalte produzieren oder kombinieren und über elektronische Netze allgemein zugänglich und kostenfrei veröffentlichen – und dabei wiederum Privatpersonen als Zielgruppe fokussieren und deren Beteiligung kalkulieren und/oder erwarten (Interaktionsaspekt). „Social Software“-Konzepte wie Weblogs, Wikis, Online Foto- und Linkverwaltungen u. a. ermöglichen und unterstützen derartige Aktivitäten, ohne tiefgehende Kenntnisse der technischen Infrastruktur vorauszusetzen. Aktuell werden die durch das Nutzerengagement veränderten Informations- und Kommunikationsprozesse im Internet im Zusammenhang der Auseinandersetzung mit dem Thema „Web 2.0“ intensiv diskutiert.

Ein erster Ausgangspunkt zur Artikulation der Problemstellung dieser Arbeit ist die Registrierung, dass eine Vielzahl der genannten Anwendungen aus dem Bereich „Social Software“ mit Tagging eine simple und äußerst populäre Methode zur Erschließung von digitalen Inhalten und somit zur Generierung von Metadaten implementiert hat. Auf dieser Feststellung aufbauend kann weiter geäußert werden, dass diesen Systemen ein gleichartiges

konzeptionelles Modell zu Grunde liegt, das die Konzepte „Nutzer“, „Tag“ und „Ressource“ involviert und strukturiert.

Ein zweiter Ausgangspunkt für die Problemformulierung reflektiert zum einen die Beobachtung, dass aus Anwendungen im Bereich „Social Software“ heraus ein intensiver Datentransfer in andere Anwendungen bzw. ihre Speichersysteme stattfindet (Stichwort *mashups*), zum anderen diese Systeme ihre i. d. R. auf XML basierenden Formate zur Datenintegration und -aggregation sowie ihre Schnittstellen z. B. über Webservices, die auch eine Datenmanipulation ermöglichen, jedoch individuell strukturiert bzw. individuell definiert haben und mit der eingesetzten bzw. an der Schnittstelle zur Verfügung gestellten Technologie nur eine vage Semantik etablieren können.

So kann die Problemstellung formuliert werden, dass der Aufwand für die Integration von Daten aus verschiedenen Anwendungen erhöht ist und solche Daten in ihrer Semantik wenig verlässlich sind, obwohl der Datenaustausch ein Prinzip aktuell populärer Anwendungen ist.

In der Auseinandersetzung mit dieser Problemstellung können unterschiedliche Aufgaben und Lösungswege beschrieben werden.

1.1 Zielformulierung

Zielstellung dieser Diplomarbeit ist, unter Verwendung von ISO/IEC 13250 Topic Maps ein Modell zu definieren, zu implementieren und dessen Anwendung prototypisch zu demonstrieren, das es ermöglicht, die für die Integration und Aggregation in externen Systemen im Kontext des Taggings relevanten Daten standardisiert zu strukturieren und zu speichern. Aufbauend auf einer homogenisierten Struktur können weiterführend standardisierte Austauschformate und -mechanismen definiert werden. Neben diesem strukturellen Aspekt, der das Effizienzpotential betrifft, stellt die mit Topic Maps realisierbare Etablierung von Semantik einen zweiten Aspekt in der Modellentwicklung dar, der das Effektivitätspotential betrifft.

In der Betrachtung der Zielstellung als einen exemplarischen Praxisfall kann die These formuliert werden, dass Topic Maps zur Standardisierung von Kontext repräsentierenden Datenstrukturen eingesetzt werden kann und der Einsatz dieser Technologie in der Konsequenz einen Datenaustausch auf Basis von Semantik ermöglicht.

1.2 Vorgehensweise

In der Bearbeitung dieser Aufgabe werden in Kapitel 2 zunächst ausgewählte Entwicklungen in der Internetökonomie unter Heraushebung des Informationsaspektes aufgezeigt und analysiert, die primär auf der Existenz und Verbreitung elektronischer Netze beruhen, um den Bearbeitungsrahmen zu kennzeichnen. In Kapitel 3 werden anschließend die Rolle und Position der Nachfrager in diesem Kontext betrachtet.

In der Fortführung der erarbeiteten Ergebnisse wird in Kapitel 4 im Zusammenhang mit der Analyse weiterer aktueller Entwicklungen, die unter dem Schlagwort „Web 2.0“ subsumiert werden, das Tagging und sein Kontext eingehend untersucht, um daraus Rückschlüsse und Anforderungen für die Erarbeitung eines Modells zu gewinnen.

In Kapitel 5 wird der ISO/IEC-Standard 13250 Topic Maps unter Berücksichtigung gegenwärtiger, den Standard und die Technologie betreffender Entwicklungen vorgestellt. Vor diesem Hintergrund werden konstitutive Voraussetzungen für die Implementierung und Anwendung des zu entwickelnden Modells eingeführt: In Kapitel 5.6 die Programmierschnittstelle, das API, für Topic Maps-Systeme in PHP, PHPTMAPI, sowie eine erste Implementierung von PHPTMAPI in Kapitel 5.7.

Anschließend wird in Kapitel 6 das Tagging-Modell mit Topic Maps entwickelt und dessen Implementierung und Anwendung gezeigt.

Den Abschluss dieser Arbeit bilden die Beurteilung der Ergebnisse sowie davon ausgehend die Bewertung der These und resultierende Schlussfolgerungen.

2 Informationsgesellschaft

Unter dem vielfach verwendeten Terminus „Informationsgesellschaft“ können die generellen gesellschaftlichen, politischen und wirtschaftlichen Veränderungen zusammengefasst bzw. erfasst werden, welche durch die Entwicklungen in der Informations- und Kommunikationstechnologie, im folgenden auch mit IuK-Technologie abgekürzt, induziert worden sind und deren Auswirkungen ihre volle Entfaltung wohl längst nicht erreicht haben.

Informationsgesellschaft beschreibt dabei einen Zustand, in der der sekundäre Sektor der Wirtschaft nicht mehr durch den Faktor „Produktion“, sondern durch den Faktor „Information“ bestimmt wird (vgl. Wirtz 2001, S. 2). Sie, die Informationsgesellschaft, stellt eine Evolutionsstufe zeitlich nach der Industriegesellschaft dar. Bereits in den 1970er Jahren prägte Daniel Bell den Begriff der *postindustrial society* und stellte folgende Treiber in den jeweiligen Gesellschaftsformen gegenüber:

	vorindustriell	industriell	nachindustriell
Umwandlungsprozess	natürliche Kraft (Wind, Wasser, Muskelkraft)	erzeugte Energie	Information
umwandelnde Kraft	Handwerkliches Können	Kapital	theoretisches Wissen

Tab. 1: Treiber gesellschaftlicher Wandlungen (vgl. Bell 1985, S. 12)

Für seine Konzeption stellt er fest:

Das Konzept der nachindustriellen Gesellschaft ist ein Analyserahmen der Veränderungen einer Gesellschaft entlang der Achse der aufkommenden „intellektuellen Technologie“. (Bell 1985, S. 10)

Tabelle 1 zeigt die Treiber „Information“ und „theoretisches Wissen“. Dies zeigt die Verknüpfung, die Beziehung dieser beiden Begriffe (Prozess – Kraft/Treiber), womit auch die Frage anschließen kann, ob die nachindustrielle Gesellschaft eine Informations- oder eine Wissensgesellschaft ist.

Für die Begriffe „Information“ und „Wissen“ kann festgestellt werden, dass diese nicht das gleiche gedankliche bzw. geistige Konzept bezeichnen, dass zwischen ihnen ein Bedeutungsunterschied besteht. Eine sinnvolle Annäherung an eine Definitionsarbeit, die hier nicht erfolgen soll, ist die Unterscheidung der Begriffe „Zeichen“, „Daten“, „Information“ und „Wissen“ in einer Begriffshierarchie, wie sie in Abbildung 1 dargestellt ist.

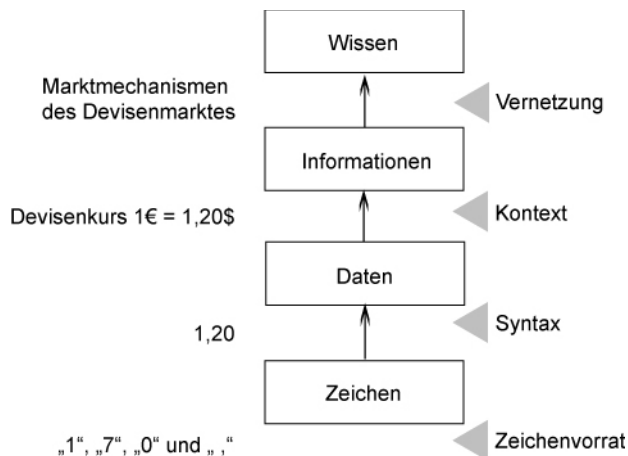


Abb. 1: Beziehungen der Ebenen der Begriffshierarchie Zeichen, Daten, Information, Wissen (vgl. Probst/Raub/Romhardt 1998, S. 34; in Anlehnung an Rehäuser/Krcmar [vgl. Krcmar 2005, S. 14]; eigene Darst.)

Für die Transformation von Information in Wissen kennzeichnen Probst, Raub und Romhardt den Aspekt der Vernetzung (siehe Abb. 1). Entsprechend kann nur derjenige Organismus oder, in der Theorie, diejenige technische Konstruktion etwas wissen, der bzw. die autonom Informationen vernetzen kann (präziser: zur Lösung von Problemen vernetzen/verknüpfen kann). Eine solche Fähigkeit ist dem Menschen eindeutig zuzuordnen. Für das „Besitzen“ von Wissen gilt, dass dies für Menschen möglich ist, während Informationen nicht an Menschen gebunden sind und mittels Zeichen und Daten(ströme) erstellt, verteilt und gespeichert werden können. Die Definition von Wissen von Probst, Raub und Romhardt fasst die Beobachtung zusammen:

Wissen bezeichnet die Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen. Dies umfasst sowohl theoretische Erkenntnisse als auch praktische Alltagsregeln und Handlungsanweisungen. Wissen stützt sich auf Daten und Informationen, ist im Gegensatz zu diesen immer an Personen gebunden. Es wird von Individuen konstruiert und repräsentiert deren Erwartungen über Ursache-Wirkungs-Zusammenhänge. (Probst/Raub/Romhardt 1998, S. 44)

Mertins, Heisig und Vorbeck erstellen eine Sammlung von quasi Prüffragen, um Information von Wissen unterscheiden bzw. abgrenzen zu können:

For data and information the questions are: who – what – where – when? etc. while typical questions for knowledge are: how? and why? (Mertins/Heisig/Vorbeck 2003, S. 2).

Information soll dabei zunächst pragmatisch als Daten im Kontext verstanden werden, d. h., dass Daten in einem bestimmten Kontext an Bedeutung gewinnen und so zu Informationen werden (vgl. Krcmar 2005, S. 15).

Entsprechend ergibt sich aus dem Beispiel in Abbildung 1, dass mit „1,20“ der Wert des Dollars in Euro bezeichnet ist. Weiterhin ist festzustellen, dass Individuen

1. stetig mit Informationen konfrontiert werden und
2. diese rezipieren, verarbeiten, selektieren und je nach Bedarf speichern.

Die in der Betriebswirtschaft häufig zitierte Definition von Wittmann, der Information als zweckorientiertes Wissen versteht (z. B. in Keuper 2002, S. 121; Picot/Reichwald/Wigand 2003, S. 25), ist unglücklich, da der Begriff „Information“ durch den in der oben gezeigten Hierarchie übergeordneten (also noch zu definierenden), ein anderes Konzept bezeichnenden Begriff „Wissen“ in einem direkten Zusammenhang beschrieben wird.

Für eine Unterscheidung der mit „Informationsgesellschaft“ und „Wissensgesellschaft“ bezeichneten Begriffe bzw. für die Beantwortung der Frage, ob die nachindustrielle Gesellschaft eine Informations- oder Wissensgesellschaft ist, lässt sich konstatieren, dass dies eine Frage der Sichtweise, der Perspektive zu sein scheint. Wird bei der Betrachtung der Mensch in den Fokus gerückt, z. B. im Personalmanagement einer betrieblichen Organisation, so ist eher von Wissensmanagement zu sprechen, da das Individuum mit seinem Wissen und Fähigkeiten für die betriebliche Organisation eine bestimmte Funktion übernimmt. In der Betrachtung z. B. der technischen Aspekte auf Ebene der IuK-Technologien ist eher von Informationsgesellschaft zu sprechen, da bis dato kein allgemeiner Konsens darüber besteht, dass technische Konstruktionen wissen können (wie oben skizziert), wohl aber darüber, dass die Speicherung von Informationen in Form von Text, Audio und Video sowie deren Verarbeitung zu ihren herausragenden Merkmalen zu rechnen und Ursprung zu kennzeichnender Entwicklungen ist.

Da in dieser Arbeit vor allem technische Aspekte erörtert werden, soll im folgenden der Begriff der Informationsgesellschaft verwendet werden.

Grundsätzlich kann für die Informationsgesellschaft festgestellt werden, dass Menge und Verfügbarkeit an Informationen neue Dimensionen angenommen haben (vgl. Wirtz 2001, S. 5) und ein Ende dieser Entwicklung nicht abzusehen ist.

Als weitere Merkmale auf gesellschaftlicher Ebene lassen sich u. a. identifizieren:

- Die wirtschaftliche Bedeutung des Informationssektors wächst,
- naturwissenschaftliche Erkenntnisse nehmen exponentiell zu [...],
- die Notwendigkeit einer informationstechnischen Basisqualifikation für alle wird politisch erkannt,
- Informationstechnologien bestimmen den Berufsalltag auch traditionell technikferner Arbeitsplätze. (Lehmann 2005, S. 37)

Weiterhin ist festzustellen, dass die IuK-Technologie private und Freizeitaktivitäten nachhaltig beeinflusst.

Die Voraussetzung (*enabler*) dieser Entwicklungen sind elektronische Netze auf Basis des „universalen Netzwerkprotokoll[s] TCP/IP“ (Keuper 2002, S. 611), die einen zeit- und ortsunabhängigen Zugriff auf digitale bzw. digitalisierte Informationen in einem globalen Rahmen ermöglichen. Dabei soll in den folgenden Ausführungen die zugängliche Gesamtheit der TCP/IP-basierten elektronischen Netze als Internet bezeichnet werden. Einhergehend mit den Entwicklungen bei elektronischen Netzen in puncto Kapazität und Verfügbarkeit ist exemplarisch als weitere Schubkraft die Entwicklung der Rechenleistung von Computerprozessoren zu nennen, die sich nach Moores „Gesetz“ alle 18 Monate bei gleich bleibenden Produktionskosten verdoppelt (vgl. Keuper 2002, S. 609; Wirtz 2001, S. 23f.). Das umfassende Technologiebündel kann als IuK-Technologie bezeichnet werden.

2.1 Internetökonomie

Die wirtschaftlichen Aspekte der Informationsgesellschaft manifestieren sich in der Internetökonomie. (Wirtz 2001, S. 18)

Analog zur obigen Beschreibung des Bedeutungs- bzw. Bezeichnungszusammenhangs zwischen elektronischen Netzen und dem Internet können in den folgenden Ausführungen Internet- und Netzwerkökonomie synonym verwendet werden („Netzwerk-Ökonomie“ z. B. bei Hofmann 2001, S. 1). Ausgehend von der Aussage von Wirtz lassen sich einige Schlussfolgerungen bzgl. der Merkmale der Internetökonomie auf Basis der Erarbeitungen aus dem vorigen Kapitel aufzeigen:

- Die Internetökonomie nutzt das Internet als Infrastruktur;
- Die Internetökonomie findet folglich in einem globalen Rahmen statt;
- Die Internetökonomie fußt folgend aus 1. auf Digitalität und dem Prinzip der Vernetzung;
- In der Internetökonomie nehmen Informationen eine herausragende Rolle ein.

Diese Informationen liegen entsprechend der Architektur von TCP/IP in digitaler Form vor, sind also in digitale Einheiten zerlegt, sogenannte Bits, die durch 0 und 1 ausgedrückt werden (vgl. Zerdick et al. 1999, S. 140). Eine synthetische Definition von Wirtz soll „Internetökonomie“ wie folgt skizzieren:

Die Internetökonomie ist eine im wesentlichen digital basierte Technologie, welche die computerbasierte Vernetzung nutzt, um Kommunikation, Interaktionen und Transaktionen in einem globalen Rahmen zu ermöglichen. (Wirtz 2001, S. 23)

Dabei wird es zunehmend schwieriger, Internetökonomie von einer Nicht-Internetökonomie abzugrenzen, da durch die Entwicklung „hybrider Konzepte“ (Stähler 2001, S. 28) die Grenzen immer weiter aufweichen. Eine Verschmelzung ganzer Märkte ist bereits als ein Ergebnis von Entwicklungen zu erkennen, die als Konvergenz bezeichnet wird und einen neuen Multimedia-Markt entstehen lassen (vgl. Keuper 2002, S. 620). Dieser Aspekt wird in Kapitel 2.1.2 noch eingehender erläutert.

Dies zeigt, dass die Internetökonomie keinen Zustand darstellt, sondern durch unterschiedliche Entwicklungsprozesse und Strömungen weiter ausgeprägt und entwickelt wird.

Die wesentlichen Charakteristika der Internet- bzw. Netzwerkökonomie und ihre Zusammenhänge werden in Abbildung 2 zusammenfassend visualisiert.

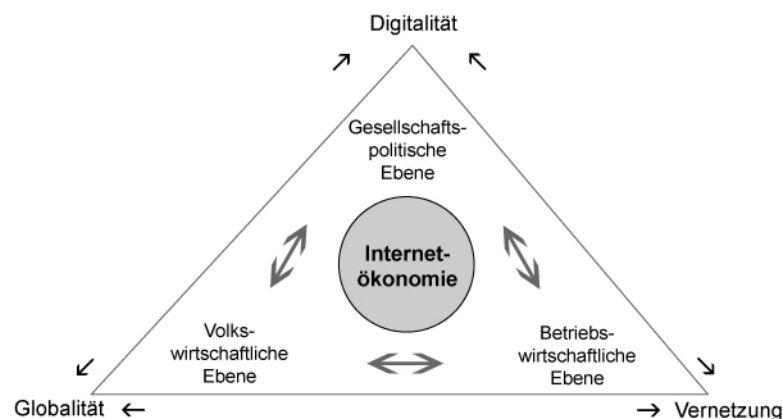


Abb. 2: Charakteristika der Internetökonomie (vgl. Wirtz 2001, S. 31; eigene Darst.)

Grundsätzlich ist eine Beschleunigung wirtschaftlicher Entwicklungen, z. B. bei Produktinnovationen, festzustellen, da sich die Basistechnologie schnell fortentwickelt (vgl. Stähler 2001, S. 30) und sich, als Infrastruktur, bzgl. der Kapazität und Verfügbarkeit global weiter verbreitet. Hofmann beschreibt in diesem Rahmen folgende Entwicklungen:

- neue Märkte für alte Produkte,
- neue informationsintensive Produkte,
- neue Lieferkanäle,
- neue Interaktionsbeziehungen mit den Kunden,
- neue Strukturen bei der Beschaffung,
- neue Kooperationen und Allianzen (vgl. Hofmann 2001, S. 1f.).

Zur Verdeutlichung des Aspektes der neuartigen informationsintensiven Produkte, wobei hier auch Dienstleistungen mit eingeschlossen werden können, soll an dieser Stelle kurz auf die Möglichkeiten für die Entwicklung neuer Geschäftsmodelle auf Basis der IuK-Technologien eingegangen werden. Zum Verständnis dessen, was „Geschäftsmodell“ meint, soll hier die Definition von Rentmeister und Klein dienen, die in Anlehnung an die Definitionen von Timmers und Stähler quasi als Synthese erarbeitet worden ist:

Ein Geschäftsmodell ist ein Modell, das bezogen auf eine Geschäftstätigkeit

- die beteiligten Akteure, ihre Rollen und ihren Beitrag zur Wertschöpfung (Architektur der Wertschöpfung),
- den Nutzen, den Kunden oder andere Akteure aus der Geschäftstätigkeit ziehen können (Value Propositions), und
- die Einnahmenquellen, die die Geschäftstätigkeit eröffnet (Ertragsmodell), abbildet. (Rentmeister/Klein 2003, S. 19).

Deutlich wird das dreigliedrige Modell von Stähler, bestehend aus Architektur der Wertschöpfung, Value Propositions und Ertragsmodell.

Timmers beschreibt ein Schema zur Konstruktion von Geschäftsmodellen¹, die nur mittels der Nutzung bzw. Verwendung von IuK-Technologien realisierbar sind. Als Ausgangspunkt betrachtet Timmers die Wertschöpfungskette. Sein Schema unterscheidet die drei Prozessschritte

- value chain de-construction,
- interaction patterns
- value chain re-construction (Timmers 1998, S. 4).

¹ Auf Grund ihres Abstraktionsgrades können diese auch als Geschäftsmodelltypen bezeichnet werden.

Als Ausgangspunkt betrachtet Timmers die Wertschöpfungskette als Modell von Porter, das einen linearen Ablauf wertsteigernder Tätigkeiten abbildet. *Value chain de-construction* bezeichnet die Unterscheidung der Elemente der Wertschöpfungskette, Timmers bezieht sich hier konkret auf Porter (vgl. Timmers 1998, S. 4) und seine neun Elemente Eingangslogistik, Operationen, Ausgangslogistik, Marketing und Vertrieb, Kundendienst (vgl. Porter 1999, S. 78).

Interaction patterns beschreiben die Konstellationen, in denen involvierte Akteure interagieren können: 1 – zu – 1, 1 – zu – viele, viele – zu – viele etc. So beschreibt die Konstellation 1 – zu – 1 die Interaktion zweier Akteure, aber nicht den Prozess des One-to-one Marketing (vgl. Timmers 1998, S. 4). „Value chain re-construction, that is integration of information processing across a number of steps of the value chain.“ (Timmers 1998, S. 4), wobei verschiedene Elemente der Wertschöpfungskette kombiniert werden können (vgl. Timmers 1998, S. 4). *Integration of information processing* kann als zielgerichtete Integration der an den verschiedenen Schnittstellen generierten bzw. erhobenen Informationen mittels elektronischer Netze verstanden werden.

Als Beispiel eines neuen Geschäftsmodells, das eine neuartige Leistung beinhaltet, führt Timmers *value chain integration* an, das die enge Integration verschiedener Geschäftsprozesse entlang der Wertschöpfungskette kennzeichnet. Motivation bzw. Treiber eines solchen Geschäftsmodells ist die Anforderung der Nachfrager, dass ihre Bedürfnisse zu jeder Zeit erfüllt werden (vgl. Timmers 2003, S. 127 f.). Eine derartige effiziente Integration verschiedener Prozesse zwischen Beschaffung und Absatz kann nur mit Hilfe der IuK-Technologien realisiert werden.

An dieser Stelle wird im Ansatz deutlich, welche Auswirkungen die neuen Technologien auf die Leistungs(angebots)struktur haben. Wie bereits erwähnt, beschleunigen dabei die Netzinfrastrukturen insgesamt die Prozesse, was gleichzeitig eine effizientere Koordination und einen effektiveren Einsatz von Informationen nach sich ziehen muss. Ein Stichwort in diesem Kontext lautet Informationslogistik.

Zunächst sollen fortführend einige signifikante Entwicklungen im Unternehmensumfeld, dem Wettbewerb, erörtert werden, um die umwälzenden Entwicklungen in der Internetökonomie weiter zu präzisieren und den Fokus auch in Richtung des Nachfragers zu lenken.

2.1.1 Transparenz der Märkte

Im folgenden Abschnitt soll mit „Transparenz“ ein entscheidender Treiber skizziert werden, der für die Intensivierung des Wettbewerbs verantwortlich ist. Allgemein soll unter Transparenz das Maß der Informationsverfügbarkeit und der Informationszugänglichkeit marktrelevanter Informationen verstanden werden, wobei für den Aspekt der Zugänglichkeit der Kosten- und Zeitaufwand berücksichtigt werden muss.

Eine marktrelevante Information ist bspw. der Preis eines Gutes oder einer Leistung, die diesen grundsätzlich adhärent ist. Inwieweit das Internet einen Beitrag zur Erhöhung der Markttransparenz geleistet hat, zeigt folgendes Beispiel, die Nutzung eines Preisvergleichsdienstes, auch als Preis- oder Shopping-Roboter bezeichnet. In diesem Falle wird diese Leistung bei kelkoo (<http://www.kelkoo.de>), einem Unternehmen von Yahoo!, angefragt. Es soll zunächst nach dem Preis für ein höherpreisiges Produkt, einen tragbaren Personalcomputer, auch als Notebook oder Laptop bezeichnet, recherchiert werden. Bei kelkoo ist es möglich, auf Grund der Angabe verschiedener Produkteigenschaften einen passenden Artikel zu suchen, z. B. unter der Definition der Prozessorleistung, der Festplattengröße oder der Größe des Arbeitsspeichers. Der Einfachheit genügend soll ein bestimmtes Modell, das VAIO VGN-BX196VP von Sony mit dem 1,73 GHz Prozessor, recherchiert und im Preis verglichen werden. Zur Gegenüberstellung dazu wird ein im Preis niedrigeres Produkt, eine kompakte Digitalkamera, die IXUS 55 von Canon, im Preis verglichen. Abbildung 3 zeigt das Ergebnis beider Anfragen.

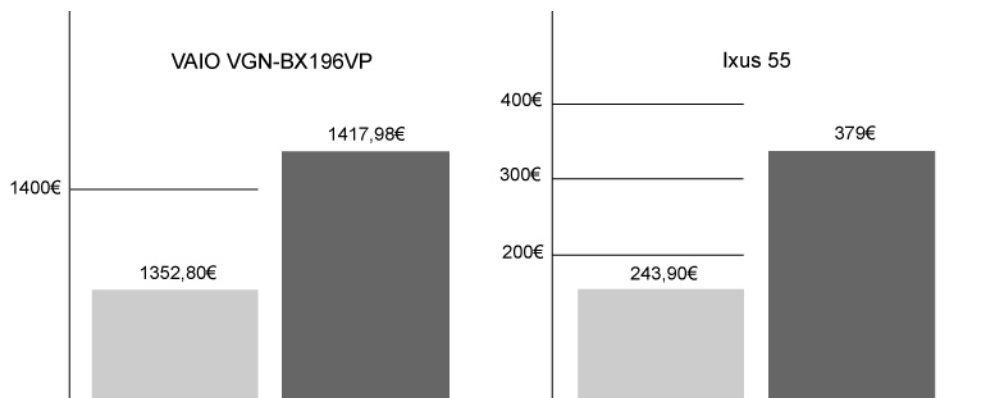


Abb. 3: Preisvergleich für das VAIO VGN-BX196VP von Sony und die IXUS 55 von Canon mit kelkoo

Für das Notebook kann ein Preisunterschied von 65,18 € ermittelt werden. Das entspricht einer möglichen Ersparnis von knapp 4,6 %. Bei der Digitalkamera fällt der Unterschied im Preis drastischer aus, hier ermittelt kelkoo eine Differenz von 135,10 €, was einer Einsparung von knapp 35,65 % entspricht.

Anhand der Skizze eines Informationsversorgungsprozesses in der Zeit vor der allgemeinen Verfügbarkeit des Internets und der Existenz entsprechender inhaltlicher Angebote werden die Veränderungen aus ökonomischer Sicht klar. In der traditionellen Ökonomie gestaltete sich ein solcher Prozess derart, dass ein interessierter Nachfrager die Angebote verschiedener, meist stationärer Anbieter mit Ladengeschäft verglich und sich vom Fachpersonal beraten ließ. Das hieß für den Interessenten bzw. potentiellen Nachfrager, dass ihm einerseits Opportunitätskosten entstanden, weil er für den Recherche- und Informationsprozess Zeit aufwenden musste. Opportunitätskosten bezeichnen die Kosten des entgangenen Nutzens (vgl. Wirtz 2005, S. 25): Der Interessent hätte in dieser Zeit auch andere Aktivitäten durchführen können, z. B. Lesen oder Arbeiten. Andererseits entstehen dem Interessenten Transaktionskosten. Eine Transaktion bezeichnet in der Ökonomie die Übertragung von Besitzrechten (vgl. Picot/Reichwald/Wigand 2003, S. 49). Die Kosten, die in diesem Prozess anfallen, sind die Transaktionskosten. Diese entfallen bspw. in der Anbahnungsphase für Reise und/oder Transport, bspw. für eine Fahrt mit öffentlichen Verkehrsmitteln zu verschiedenen Ladengeschäften. Durch die Verfügbarkeit elektronischer Netze, die einen freien Fluss von Informationen und einen Zugriff auf diese zu jeder Zeit von fast jedem Ort aus ermöglichen, können Nachfrager diese Kosten minimieren. Im geschilderten Falle des Preisvergleichs sind Fahrten zu verschiedenen Anbietern, um Preisangebote einzuholen, obsolet – in Sekundenschnelle kann dieser Vergleich durchgeführt werden, möglicherweise auch während einer parallel stattfindenden Tätigkeit.

Traditionell verfügten die Anbieter auf Grund ihrer Position im Markt über einen Informationsvorsprung, z. B. bzgl. der Preise, aber auch hinsichtlich qualitativer Kriterien wie der Ausstattung eines Gutes. Entsprechend bestand eine Informationsasymmetrie zwischen Anbieter und Nachfrager (vgl. Wirtz 2001, S. 152). In der Internetökonomie können auch die Nachfrager die Märkte auf Ebene der Informationen über Eigenschaften von Produkten und Leistungen durchdringen. Zu früherer Zeit mussten die Nachfrager abwägen zwischen dem Nutzen einer Information und den dargestellten Kosten für einen

Informationsbeschaffungsprozess in Form von weiteren Recherchen und Beratungen.

Diese Kosten der Informationsversorgung waren in der traditionellen Ökonomie ein bedeutender Bestandteil der Friktionen, also der Faktoren, die einen Kunden an einen Anbieter binden (vgl. Wirtz 2001, S. 153). Auch stellt es sich so dar, dass sich die Nachfrager fundiertere Kenntnisse über Produkte und Leistungen aneignen können, z. B. über den Austausch in Foren, den grundlegenden Kommunikationsinstrumenten virtueller Gemeinschaften, oder über digitale Angebote von Expertengruppen, bspw. der Stiftung Warentest.

So lässt sich eine erhöhte Transparenz der Märkte konstatieren. Heute können Nachfrager auf Informationsebene dem Anbieter gleichwertig gegenüber treten, weshalb auch von reversen Märkten gesprochen werden kann (vgl. Wirtz 2001, S. 153). Dabei muss angemerkt werden, dass die große Menge sowie die hohe Verfügbarkeit von Informationen, zentrale Merkmale der Informationsgesellschaft, zu Effekten führen können, die allgemein als *information overload* bezeichnet werden und das Problem kennzeichnen, die Fülle an Informationen nicht mehr erfassen und verarbeiten zu können. In diesem Zusammenhang ist auch die Relevanz von Medienkompetenz als Teilgebiet der Medienpädagogik zu erwähnen, die u. a. auch das Verstehen von Medien in ihren Ursache-Wirkungszusammenhängen beinhaltet.

Die signifikant erhöhte Transparenz der Märkte hat zu einer Verschärfung des Wettbewerbs geführt, da auch kleinere Anbieter ohne Ladengeschäft, die sich auf Grund der durch das Internet induzierten verringerten Markteintrittsbarrieren leichter in Märkten positionieren können, mit ihren digitalen Angeboten, z. B. Online-Shops, unmittelbar über Preisvergleichsdienste auch von den Nachfragern wahrgenommen werden können. Bei den Nachfragern hat u. a. diese Veränderung zu einer deutlichen Preissensibilität geführt. Die Kampagne „Geiz ist geil!“ des Elektrogerätefachhändlers Saturn steht sinnbildlich für diese Entwicklung, die jedoch zunehmend von einem erstarkenden Qualitätsbewusstsein der Nachfrager eingeholt zu werden scheint.

2.1.2 Konvergenz

Auf einer höheren Ebene, der volkswirtschaftlichen Ebene, kann ein Zusammenwachsen ehemals disparater Branchen beobachtet werden. Der Begriff „Konvergenz“, wie er seit längerem in der Ökonomie diskutiert und beschrieben wird, manifestiert sich in der allgemeinen Wahrnehmung häufig in der Vorstellung eines multifunktionalen mobilen Gerätes, das neben der Kommunikation mittels Text, Audio und Video auch die Rezeption von Inhalten in den drei Informationsdimensionen erlaubt und dabei ebenfalls einen Zugang zum Internet ermöglicht. Mit dieser Skizze ist das Feld der involvierten Branchen umrissen, konkret sind dies die Medien-, die Telekommunikations- und die Computer- bzw. IT-Branche. Ergebnis bzw. Synthese dieses Verschmelzungsprozesses ist die Multimediabranche bzw. der Multimediabereich (vgl. Keuper 2002, S. 605; Wirtz 2001, S. 136; Picot/Reichwald/Wigand 2003, S. 162). Im folgenden soll unter Konvergenz die (Neu)Kombination einzelner Wertschöpfungsbereiche der unterschiedlichen Branchen und letztendlich ein Zusammenwachsen der Märkte auf Basis der IuK-Technologie verstanden werden (in Anlehnung an Keuper 2002, S. 605; Wirtz 2005, S. 41).

Auslöser dieser Entwicklung sind bestimmte Treiber in den Feldern Ökonomie, Technologie und Gesellschaft (vgl. Keuper 2002, S. 607 ff.). Deshalb werden diese auch als „Konvergenztreiber“ (ebd.) bezeichnet.

Ende der 1990er Jahre ermöglichten diverse Deregulierungsmaßnahmen das Aufbrechen verschiedener Anbietermonopole, exemplarisch hierfür steht die Entwicklung im nationalen Telekommunikationssektor. Während ehemals die Deutsche Telekom das Fest- und Kabelnetz sowie entsprechende Leistungen wie Telefonanschluss und Gesprächsvermittlung exklusiv anbieten konnte, können Nachfrager heute zwischen zahlreichen Anbietern wählen, da diesen die Möglichkeit der Nutzung der bestehenden Netze zugesprochen worden ist. Quasi analog zu dem im vorigen Kapitel dargestellten Beispiel können Nachfrager u. a. unter <http://www.billiger-telefonieren.de> in Sekundenschnelle eine Übersicht über das tagesaktuelle Angebot für die Leistung „Gesprächsvermittlung per Telefon“ erhalten.

Die Deregulierung hat zu erhöhtem Wettbewerb und im Effekt zu niedrigeren Preisen geführt. Zudem sind die Angebote vielfältiger und maßgeschneiderter. Dieser Aspekt wird bei der Erörterung eines Treibers auf gesellschaftlicher Ebene aufgenommen.

Auf technologischer Ebene ist hier als grundlegender Treiber die Digitalisierung zu nennen, die, wie bereits erwähnt, die Zerlegung von Informationen in digitale, durch 0 und 1 repräsentierte Einheiten bezeichnet. So können Informationen verlustfrei vervielfacht und über elektronische Netze distribuiert und weiterverarbeitet werden. Einhergehend mit der Digitalisierung ist ebenfalls die angesprochene Entwicklung der Rechenleistung bei Computerprozessoren zu nennen, für die das o. g. „Gesetz“ von Moore angewendet werden kann. Mit steigender Rechenleistung können

1. digitale bzw. digitalisierte Informationen schneller und
2. in größerem Umfang transferiert werden.

Entsprechend sorgt die Entwicklung bei der Rechenleistung von Prozessoren für eine erhöhte Dynamik in der Verbreitung der IuK-Technologie. Im Zusammenhang mit der Diffusion sind die positiven Rückwirkungen (*feedbacks*) im Zusammenhang mit Netz- (z. B. in Picot/Reichwald/Wigand 2003) bzw. Netzwerkeffekten (z. B. in Wirtz 2001) zu erwähnen. Grundsätzlich ist als Netzefekt ein Wirkungsprozess zu kennzeichnen, der die Auswirkung der Teilnahme eines Individuums an einem Netzwerk beschreibt. Bei den direkten positiven Effekten steigt der Wert des Netzwerks mit steigender Zahl der Teilnehmer, ein häufig zitiertes Beispiel ist das Telefon bzw. Telefonnetz. Ein Nutzen ergibt sich für den einzelnen erst bei einer bestimmten Anzahl an Teilnehmern („Kritische-Masse-System“ [Keuper 2002, S. 610]). Mit dem Überschreiten der kritischen Masse ergibt sich für nicht teilnehmende Individuen eine erhöhte Anziehungskraft, am Netzwerk teilzunehmen – die tatsächliche Teilnahme erhöht dann rekursiv den Nutzen, den das Netz- bzw. Netzwerk bereitstellen kann.

So kann resümiert werden, dass bereits Netzeffekte an sich einen wesentlichen Beitrag für die Verbreitung der IuK-Technologie beisteuern und neben den determinierenden Merkmalen „Digitalisierung“ und „Rechenleistung“ sowie „Speicherkapazität“ für die rapide Verbreitung der digitalen Technologie sorgen.

Auf gesellschaftlicher Ebene ist zunehmend ein Trend zur Individualisierung der Nachfrager zu beobachten, d. h., dass zunehmend Produkte nachgefragt werden, die ein gewisses Maß an Einzigartigkeit besitzen (vgl. Meffert 2005, S. 107), dennoch im Preis Massenprodukten entsprechen sollen (vgl. Keuper 2002, S. 608). Mögliche Motivationsaspekte seitens der Nachfrager werden in Kapitel 3 erörtert.

Diese Tendenz äußert sich auf Unternehmensseite im Konzept der *mass customization*, das die Individualisierung von Leistungen beinhaltet und demnach ein verbreitertes Angebot nach sich zieht. Dabei entsteht eine „kundenindividuelle Leistungsbündelung mit überwiegendem Dienstleistungscharakter“ (Keuper 2002, S. 609). Solche personalisierten Leistungsbündel können nur entstehen, wenn die Anbieter ein präzises Bild über den Nachfrager und seine Bedürfnisse auf Grund der Sammlung und Analyse von Informationen gewinnen können. Mit Hilfe der IuK-Technologien ist eine derart effiziente und effektive Logistik von Informationen zu leisten. Die Analyse von Informationen beinhaltet deren zielgerichtete Verknüpfung bzw. Vernetzung, weshalb die erwähnten Leistungsbündel in erheblichem Maße die Wissenskomponente berücksichtigen.

Die Gleichrangigkeit der einzelnen Treiber bleibt zu diskutieren. Es kann dahingehend argumentiert werden, dass die Digitalisierung und ihre Veräußerung in der IuK-Technologie im Grunde konstitutiv für die Entwicklungen ist, die hier als Konvergenz erörtert werden. Bspw. ermöglicht die IuK-Technologie im Zusammenhang mit der erwähnten Informationslogistik im Rahmen von Individualisierung und Personalisierung einerseits die Erarbeitung von Kundenprofilen und die auf Schlussfolgerungen basierende Erstellung entsprechender, kundengerechter Leistungen, woraufhin seitens der Nachfrager in der Folge weitere Begehrlichkeiten entstehen und letztlich noch individuellere Bedürfnisse geweckt werden können. Andererseits können digitale Inhalte zu vernachlässigenden Kosten zu individuellen Leistungsportfolien kombiniert werden. Als Beispiel kann die personalisierbare Startseite von Google angeführt werden, auf der aus einem breiten Inhaltspool individuell durch den Nachfrager ausgewählte Inhalte integriert werden können.

Konvergenz als das Verschmelzen von Wertschöpfungsbereichen der Medien-, Telekommunikations- und IT-Branche zu einem Multimedia-Markt als Schnittmenge der drei Branchen tritt nach Keuper dann auf, wenn die drei Dimensionen der technologischen, nachfrageseitigen und angebotsseitigen Konvergenz zeitgleich auftreten (vgl. Keuper 2002, S. 620). Dabei bedeutet

- a) technologische Konvergenz die Verbreitung der grundlegenden Technologie, die IuK-Technologie,
- b) nachfrageseitige Konvergenz das Bedürfnis, zeitgleich und medienbruchfrei ursprünglich separate Leistungen, z. B. TV und *video-on-demand*, zu konsumieren,

- c) angebotsseitige Konvergenz die (möglichst antizipierte) Erstellung branchenübergreifender Leistungsbündel (in Anlehnung an Keuper 2002, S. 611ff.).

Nach Zerdick et al. stellt die letztlich aus drei Dimensionen resultierende Konvergenz einen zweistufigen Prozess dar. Im ersten, bereits seit langem vollzogenen Konvergenzschritt fusionieren Wertschöpfungsprozesse des Telekommunikationssektors mit denen der IT-Branche (Bsp. Vernetzung von Computern über Telefonleitungen). In der zweiten Stufe, die sich vor wenigen Jahren präzisierete, verschmelzen zusätzlich die Wertschöpfungsketten des Mediensektors mit denen der Telekommunikations- und IT-Branche (vgl. Zerdick et al. 1999, S. 132f.). Abbildung 4 illustriert zusammenfassend die Zusammenhänge.

Konvergenz bewirkt Konkurrenzsituationen zwischen Unternehmen, die ursprünglich auf unterschiedlichen Märkten agierten. Um sich gegenüber anderen Anbietern aus der originären Branche zu differenzieren, versuchen bspw. Internet Service Provider (ISP) mittels attraktiver digitalisierter Medieninhalte zusätzliche Kunden zu gewinnen sowie neue Einnahmequellen zu eröffnen und treten somit zu TV-Sendern in Konkurrenz. Dies führt folglich zu einer Verschärfung des Wettbewerbs.

Aktuell zeigt das Beispiel der Verwertung von Übertragungsrechten für die Spiele der Fußball Bundesliga für die Saison 2006/07, wie ein ISP, T-Com, neuerdings stark nachgefragte Inhalte, die ehemals exklusiv über das Fernsehen zu empfangen waren, zusätzlich über das Internet überträgt.

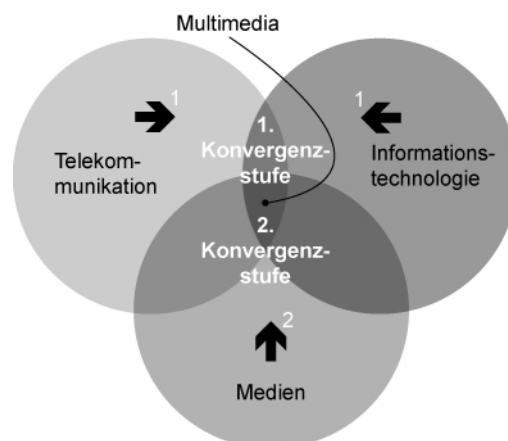


Abb. 4: Konvergenz von Telekommunikations-, IT- und Medienbranche (in Anlehnung an Keuper 2002, S. 621 und Zerdick et al. 1999, S. 133)

Die Ausführungen in den letzten beiden Kapiteln haben gezeigt, welche Veränderungen und Dynamik die IuK-Technologien entfacht haben. Unter dem Aspekt der Transparenz sind marktinterne Umwälzungen skizziert worden. Hier deutet sich auch an, dass die Nachfrager in ihrer Position an Relevanz gewinnen, d. h., dass Unternehmen sich zunehmend stärker auf sie ausrichten und einzustellen haben. Deshalb wird dieser Gesichtspunkt in Kapitel 2 noch einmal aufgenommen und der Versuch unternommen, die Ursachen eingehender zu erläutern.

Im Rahmen der Konvergenz zeigt sich, dass die Digitalisierung und ihre Manifestation in der IuK-Technologie ein wesentlicher Treiber für das Zusammenwachsen ganzer Märkte ist, wenn nicht sogar die Voraussetzung. Ein weiterer Treiber, die nachfrageseitige Individualisierung, verweist ebenfalls wieder auf die Position der Nachfrager und ihre veränderten Anforderungen an betriebliche Leistungen.

Die Situation, die sich aus der Interdependenz zwischen erhöhtem Wettbewerb und verändertem Nachfrageverhalten ergibt, rückt zwangsläufig Information und Wissen in den Fokus wirtschaftlichen Handelns und erfordert entsprechend ein effizientes und effektives Management von Informationen und Wissen.

2.2 Die Bedeutung der Information aus der ökonomischen Perspektive

Ein zentrales Merkmal der Informationsgesellschaft ist die kaum mehr erfassbare, weiter exponentiell wachsende Menge und Verfügbarkeit an Informationen und deren steigende Bedeutung in der Internetökonomie. Dabei muss festgestellt werden, dass die Existenz von Informationen sicher kein Phänomen der Informationsgesellschaft und folglich der Internetökonomie ist. In der Ökonomie waren Informationen schon immer vorhanden und spielten eine wichtige Rolle, z. B. in ihrer Ausprägung als Preise, Markennamen oder Beschaffenheit von Produkten. Informationen waren den elementaren Produktionsfaktoren „objektbezogene menschliche Arbeit“, „Betriebsmittel“, „Werkstoffe“ und den dispositiven Produktionsfaktoren „Unternehmensführung“, „Planung“, „Organisation“ (vgl. Dichtl/Issing 1993, S. 1709) schon immer anhaftend bzw. adhärent. Information stellt im betrieblichen Leistungserstellungsprozess keine separate Ressource dar, sondern ist ein übergreifendes, omnipräsentes Konzept. Entsprechend stellen Informationen keinen eigenständigen

gen (neuen) Produktionsfaktor dar (vgl. Keuper 2002, S. 124f.), während andere Autoren Information den Status eines Produktionsfaktors gleichwohl zusprechen (vgl. Wirtz 2001, S. 10; Krcmar 2005, S. 17).

Die bereits schon immer existente Bedeutung der Information für das Marktgeschehen zeigt sich in der Betrachtung von Transaktionen, wie sie bereits in den Ausführungen zur Markttransparenz geäußert worden sind. Die alleinige Betrachtung des Preises führt zu Unsicherheiten, da dieser keine Informationen zu qualitativen Aspekten enthält und zu der Frage führt, ob die Leistung diesen Preis wert sei. Deshalb versuchen die Nachfrager, weitere Informationen z. B. durch den Fachverkäufer zu erhalten und diese in Wissen über das Produkt zu transformieren. Auf Grund der Informationsasymmetrie konnten die Marktteilnehmer mit Informationsvorsprung gegenüber ihren Transaktionspartnern Vorteile z. B. in Form höherer Preise realisieren, während die anderen Marktteilnehmer versuchten, den vorhandenen Informationsrückstand zu egalisieren. Dabei spielt bzgl. des Informationsvorsprungs der Faktor „Zeit“ eine wesentliche Rolle. Folglich kann Markt auch als Ort eines Zyklus' von Wissensveränderungen gekennzeichnet werden (vgl. Zerdick et al. 1999, S. 138). Die den elementaren und dispositiven Faktoren anhaftenden Informationen waren schon immer Voraussetzung für die Koordination und Steuerung der innerbetrieblichen Prozesse und letztlich der Arbeitsteilung (vgl. Keuper 2002, S. 126f.). Abbildung 5 verdeutlicht diesen Zusammenhang.

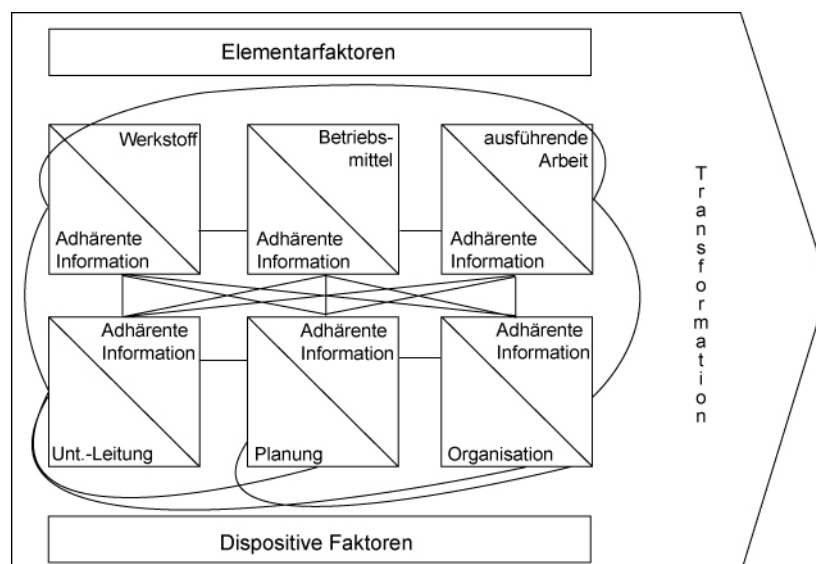


Abb. 5: Innerbetriebliche Steuerungsfunktion produktionsfaktoradhärenter Informationen (vgl. Keuper 2002, S. 127; eigene Darst.)

Aus Sicht der Internetökonomie können neben der skizzierten Bedeutung der Information in Märkten und für die betriebliche Organisation zwei weitere Bedeutungsdimensionen ausgemacht werden, die sich erst mit den Entwicklungen im Bereich der IuK-Technologie entscheidend ausprägen und durchsetzen konnten. Zum einen werden Informationen, die an den externen Schnittstellen, insbesondere der Kundenschnittstelle, „generiert“ werden, für die Wertschöpfung bzw. Wertschöpfungsprozesse als Inputfaktoren auf Grund ihres Potentials in stärkerem Maße verwertet. Die Spiegelung von Informationen über Kunden und deren Verhalten und Erwartungen in die Leistungserstellung kann zu entscheidenden Wettbewerbsvorteilen führen, weil diese Informationen es den Unternehmen erlauben, Leistungen und Leistungsbündel zu erstellen, die den aktuellen Anforderungen der Nachfrager entsprechen.

Neben dem oft zitierten Erfolgsbeispiel des Computerherstellers Dell entwickelte u. a. der Haushaltsgerätehersteller Whirlpool eine erfolgreiche Strategie. In Kooperation mit der Boston Consulting Group initialisierte das Unternehmen das Portal brandwise.com, das Nachfragern umfassende Informationen über Haushaltsgeräte anbot sowie die Möglichkeit, Produkte direkt zu erwerben oder die Adresse eines nahe gelegenen Händlers zu erfahren. Dieses Portal ermöglichte jedoch auch den Zugang zu Konkurrenzprodukten. Im Falle des Erwerbs eines Konkurrenzangebots erhielt Whirlpool dennoch wertvolle Informationen, da die Entscheidungsprozesse des Kunden vor dem Kauf ausgewertet worden sind. Aus der Analyse dieser Informationen konnte Whirlpool Wissen über Kundenpräferenzen entwickeln und anhand dessen Produkt- und Verkaufsstrategie auf die Kundenbedürfnisse anpassen (vgl. Raub/Probst 2000, S. 416f.).

Hier zeigt sich wiederum, dass Informationsasymmetrien zu Wettbewerbsvorteilen führen können. Essentiell ist hierbei die explizite Kundenorientierung. Das Prinzip könnte „mehr wissen über Kunden“ anstelle „mehr wissen als der Kunde“ lauten.

Zu beobachten ist auch das Phänomen, dass originär für betriebsinterne Prozesse vorgesehene Informationen zur Differenzierung gegenüber Konkurrenten dem Kunden als Mehrwert stiftende Leistungen angeboten werden. So können Kunden des Versandhändlers OTTO den Standort ihrer Warensendung online einsehen (hierzu auch Hofmann 2001, S. 29).

So lässt sich feststellen, dass Informationen ein Erfolgspotential darstellen (vgl. Keuper 2002, S. 133).

Eine andere Bedeutungsdimension stellt sich in der Entwicklung des Informationssektors und folglich in der Bedeutung von Information als Wirtschaftsgut dar. Medien, Software, Beratung, Datenbanken und Finanzdienstleistungen gehören zu den Wachstumsmärkten (vgl. Zerdick et al. 1999, S. 139). Exemplarisch wird i. d. R. der Medienmarkt angeführt. Dabei sorgen die Grundeigenschaften von digitalen bzw. digitalisierten Informationen für eine besondere Dynamik. Folgende Liste zeigt zunächst eine Sammlung spezifischer Eigenschaften von Informationen:

- Niedrige Vervielfältigungskosten,
- Grenzkosten der (Re-)Produktion nahe Null,
- Kein Wertverlust durch Gebrauch,
- Vielfacher Besitz möglich,
- Kein Wertverlust durch Teilung, fast beliebige Teilbarkeit,
- Logistik einfach [...]. (Krcmar 2005, S. 18).

Dabei deuten speziell die ersten zwei Punkte darauf hin, dass diese Eigenschaftensammlung die Existenz leistungsfähiger IuK-Technologie berücksichtigt. Beispielhaft soll dies im Zusammenhang mit Skaleneffekten im Medienmarkt skizziert werden.

In der Produktion von Inhalten (als Kombination von Informationen) entstehen Medienunternehmen für die Erst- bzw. Urfassung hohe Kosten, die sog. *first copy costs*. Die Kosten entstehen unabhängig davon, ob bzw. wie oft eine Kopie später abgesetzt werden kann. *First copy costs* sind *sunk costs*, da bei unzureichendem Absatz die bereits angefallenen Kosten zur Herstellung der Urfassung unwiederbringlich sind (vgl. Gerpott 2006, S. 319; Wirtz 2005, S. 33). Die Relevanz in der Betrachtung von *first copy costs* ist dann umso deutlicher, wenn das Verhältnis dieser Kosten zu den Kosten für eine einzelne Kopie in der Herstellung und in der Distribution sehr groß ist. Skaleneffekte beschreiben den Effekt, dass der Durchschnittspreis für eine Kopie mit steigender Anzahl an Kopien sinkt. Mit der Verbreitung elektronischer Netze gewinnt die Möglichkeit der erheblichen Kostendegression durch Skaleneffekte für Medienunternehmen entscheidend an Bedeutung. Digitale Inhalte können von Nutzern selbstinitiativ bei den Anbietern herunter geladen werden, wodurch dem anbietenden Unternehmen in der Distribution zu vernachlässigende Kosten für eine (Inhalts)Kopie entstehen. Entsprechend ergibt sich aus diesem Zusammenhang aus einer Eigenschaft von Information und der Existenz von IuK-Technologie eine erhebliche Marktdynamik.

Die Ursache für eine erhöhte Komplexität im Management von Informationen liegt in der Eigenschaft von Informationen als Erfahrungsgut begründet (vgl. Picot/Reichwald/Wigand 2003, S. 68). Erfahrungsgüter können in ihrer Qualität ex ante nicht beurteilt werden (vgl. Schumann/Hess 2000, S. 24). Im Gegensatz zu vielen materiellen Gütern können Prüfkriterien nur schwer definiert werden. In diesem Zusammenhang ist auch auf das „Informationsparadoxon“ hinzuweisen. Dies beschreibt das Phänomen, dass eine Information von einem potentiellen Rezipienten nicht bewertet werden kann, da er diese nicht kennt. Wenn er die Information rezipiert hat, braucht er sie nicht mehr zu erwerben, da sie dann bereits in Wissen transformiert worden ist (vgl. Picot/Reichwald/Wigand 2003, S. 299).

Der Aspekt der „einfachen Logistik“ in der obigen Eigenschaftensammlung weist auf die Bedeutung von Standards hin. So stellen Standards eine Voraussetzung für die Logistik von Informationen dar, bspw. XML im Dokumentenmanagementbereich.

Mit der Existenz von Standards und leistungsfähiger IuK-Technologie ist es möglich, große Mengen an Informationen effizient und geforderter Qualität zu verarbeiten. So können bspw. Online-Shops eine große Zahl an detaillierten Kundenprofilen anlegen, verwalten und austauschen und daraus eine Optimierung ihres Angebots ableiten.

Der Einsatz von Information hat also eine neue Bedeutung erfahren. Hier lässt sich der Ursprung für die Argumentation finden, dass Informationen ein neuer Produktionsfaktor seien. Informationen rekrutieren sich dabei jedoch nicht von neuen Informationsobjekten, sondern von den oben genannten elementaren Faktoren und den Marktteilnehmern, denen sie schon immer adhärent waren.

So bleibt zusammenfassend festzustellen, dass Informationen aus der ökonomischen Perspektive eine Steuerungsfunktion betrieblicher Prozesse übernehmen sowie ein Erfolgspotential und ein an ein Trägermedium gebundenes Wirtschaftsgut darstellen. Im Zusammenhang mit der erwähnten Relation zu Wissen muss sich konsequenterweise die Feststellung ableiten, dass analog zur Bedeutung der Information im Kontext des Personalmanagements Wissen entsprechend ein Erfolgspotential darstellt.

3 Rolle und Position der Nachfrager

Die Ausführungen der letzten Kapitel haben gezeigt, dass sich Rolle und Position der Nachfrager mit den Entwicklungen in der Internetökonomie verändert haben. Zum einen gelingt es ihnen, die Informationsasymmetrie mit Hilfe des Internet und darüber verfügbare Inhalts- und Leistungsangebote zunehmend zu beseitigen. Auf der anderen Seite hat der Stellenwert des Nachfragers in dynamischen Märkten mit verkürzten Produktlebenszyklen (vgl. Wirtz 2001, S. 194) dahingehend zugenommen, als dass er für Unternehmen als Informationsquelle zur Ermittlung von Kundenpräferenzen dient, um in fragmentierten Märkten im Zusammenwirken mit der IuK-Technologie auch unter Zeitdruck Leistungen in der von den Nachfragern erwarteten Qualität erstellen zu können und damit den Absatz zu sichern. Ausgehend von der Definition des Qualitätsbegriffs nach DIN 55350, nach der Qualität die Beschaffenheit einer Einheit bzgl. ihrer Eignung ist, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen, soll hier der kundenbezogene Qualitätsbegriff zu Grunde gelegt werden, der primär die Kundenwahrnehmung in der Bewertung von Qualität berücksichtigt (vgl. Bruhn 2000, S. 25).

Eine Erarbeitung möglicher Ursachen der Individualisierung der Nachfrager kann initial über die Betrachtung der Bedürfnispyramide von Maslow erfolgen. Dieses Modell zeigt die menschlichen Bedürfnisse in einer Hierarchie, wie sie in Abbildung 6 dargestellt ist. Dabei kann eine nächste Bedürfnisstufe erst erreicht werden, wenn die darunter evolvierten Bedürfnisse befriedigt sind. Die physiologischen Bedürfnisse stellen dabei die Basis dar (vgl. Maslow 2002, S. 62ff.).

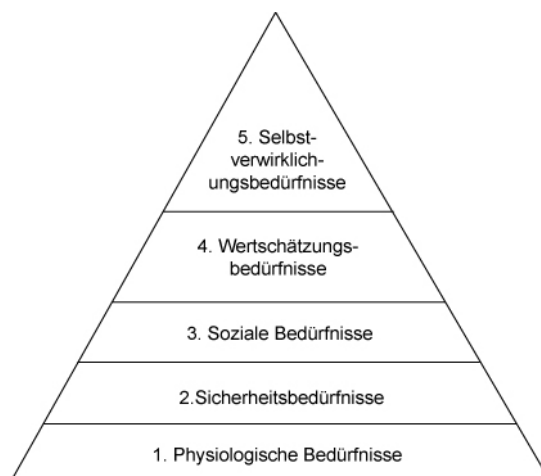


Abb. 6: Bedürfnispyramide nach Maslow

Bereits für die Industrieländer lässt bzw. ließ sich konstatieren, dass die Bedürfnisse der ersten drei Stufen i. d. R. befriedigt sind (vgl. Keuper 2002, S. 608). Dem Modell von Maslow folgend streben die Individuen dann über die Befriedigung der Bedürfnisse nach Achtung, die heute v. a. im Beruf zur Entfaltung kommen, zunehmend in Richtung Selbstverwirklichung. Maslow bemerkt zu diesem Bedürfniskomplex:

Die Tatsache besteht, daß selbstverwirklichende Menschen gleichzeitig die am meisten individualistischen und am meisten altruistischen, sozialen und liebenden aller Menschen sind. (Maslow 2002, S. 232)

So kann argumentiert werden, dass die Tendenz zur Individualisierung aus dem Streben nach Selbstverwirklichung resultiert.

Unterstützend wirken dabei gesellschaftliche und demographische Veränderungen. So ist ein Trend zur Verkürzung der Arbeitszeit zu registrieren, während gleichzeitig die Lebenserwartung steigt. Die Säulendiagramme in den Abbildungen 7 und 8 veranschaulichen die Entwicklungen.

Folglich vergrößern sich die Zeitfenster, um die entstandenen erweiterten Bedürfnisse wie das nach Selbstverwirklichung auch befriedigen zu können. Andererseits verstärkt vermehrte Freizeit auch deren Entstehung und Entwicklung.

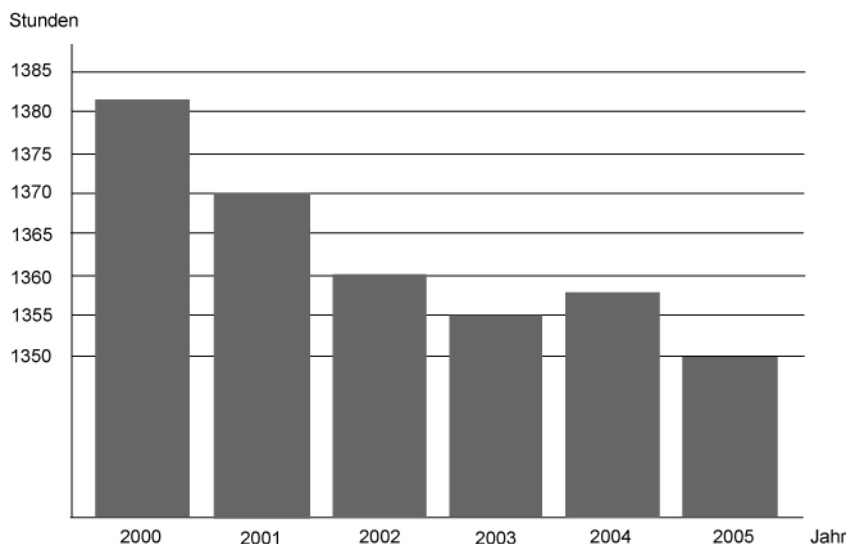


Abb. 7: Durchschnittliche Arbeitszeit in Deutschland für alle Arbeitnehmer (Quelle: IAB 2006)

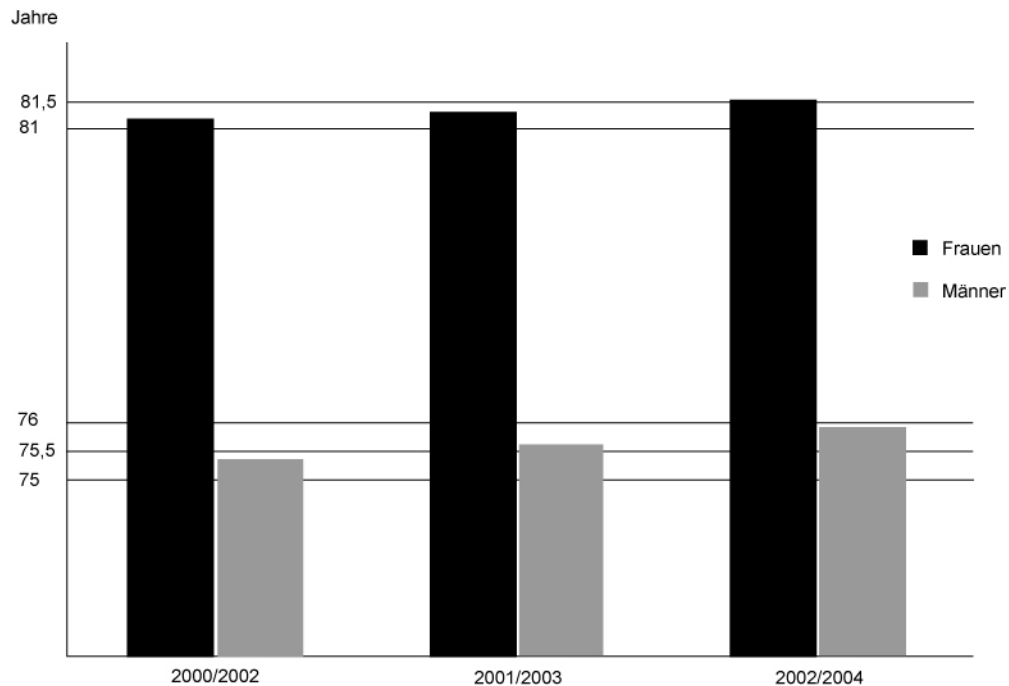


Abb. 8: Lebenserwartung für Frauen und Männer in Deutschland (Quelle: Statistisches Bundesamt 2006)

Im folgenden soll der angesprochene Aspekt der Informationsgenerierung bzgl. der Nachfragerseite detaillierter erarbeitet werden.

3.1 Nachfrager als Informationsquelle

Die Veränderung der Position der Nachfrager unter dem Aspekt, dass diese als zunehmend relevante Informationsquelle einzuschätzen sind, lässt sich anschaulich anhand des im Zuge der skizzierten Konvergenz entstandenen Multimedia-Marktes verdeutlichen. Traditionell dominiert das Modell der Wertschöpfungskette als linearer Ablauf wertsteigernder Tätigkeiten. Für die multimediale Wertschöpfung scheint die Einstufung der Nachfrager als passives letztes Element in einem linearen Ablauf nicht mehr geeignet (vgl. Keuper 2002, S. 622). Stattdessen löst ein „Multimedia-Wertschöpfungsnetzwerk“ (Keuper 2002, S. 622; Zerdick et al. 1999, S. 174), das den Nachfrager in das Zentrum aller wertschöpfenden Aktivitäten stellt, die lineare Betrachtungsweise ab (vgl. Keuper 2002, S. 622; Zerdick et al. 1999, S. 174), wie Abbildung 9 veranschaulicht.

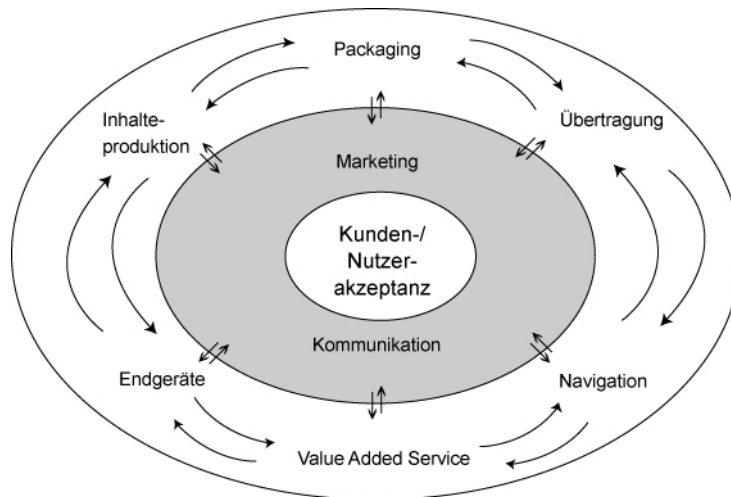


Abb. 9: Multimedia-Wertschöpfungsnetzwerk (vgl. Zerdick et al. 1999, S. 174; eigene Darst.)

Das Modell eines konzentrischen Netzwerkes trägt der gestiegenen Flexibilität der Wertschöpfung Rechnung. So existiert nicht mehr eine Schnittstelle zum Kunden am Ende eines Wertschöpfungsprozesses, sondern an jeder Stufe bzw. Sektor, wie Abbildung 9 verdeutlicht. Prinzipiell wird in jedem Leistungssektor eine für sich konsumierbare Leistung erstellt. Zusätzlich sind verschiedene Kombinationen möglich. Damit ist gemeint, dass z. B. Filme (Inhalte) über das Internet (Übertragung) geladen und auf einem Computer (Endgerät) rezipiert oder klassisch auf VHS bzw. DVD oder über die Ausstrahlung eines TV-Senders auf einem Fernsehgerät konsumiert werden können. Entsprechend resultiert daraus eine große Leistungsvielfalt. Es wird auch deutlich, dass die bereits erwähnte Leistungsbündelung mit überwiegendem Dienstleistungsanteil in Gestalt eines Systemproduktes, das eine Kernleistung sowie *value-added services* umfasst (vgl. Keuper 2002, S. 623), das zentrale Leistungskonzept darstellt. So hat die bloße Einführung einer komplexen Groupware-Lösung in einem Unternehmen ohne vorherige Beratung der Unternehmensleitung und Schulung der Mitarbeiter keinen Wert, da Einsatzzweck und Nutzungskompetenz unbekannt bzw. nicht ausgeprägt sind.

Wie sind nun die Auswirkungen auf den Informationsaspekt und die Nachfragerperspektive zu bewerten? Der grundsätzliche Zusammenhang ist in Kapitel 2.2 in Verbindung mit dem Beispiel „Whirlpool“ bereits umrissen worden. Mit der Existenz leistungsfähiger IuK-Technologien ist es möglich, umfassende Daten über Nachfrager zu Informationen zu verdichten bzw. zu kombinieren und auswerten zu lassen.

Hier ist exemplarisch das *data warehouse* zu nennen, das als

Datenbanksystem, in dem alle zur Gestaltung der Geschäftsprozesse des Unternehmens und zur Unterstützung sonstiger Managemententscheidungen erforderlicher Daten unter einer einheitlichen, zentralen Schnittstelle aus den operativen Informationssystemen des Unternehmens und aus zusammengeführten Quellen zusammengeführt werden (Wilde 2001, S. 6),

verstanden werden kann.

Das oben genannte Beispiel hat gezeigt, dass für Unternehmen selbst der Kauf eines Konkurrenzproduktes auf Informationsebene einen Gewinn darstellen kann, da auch, wenn nicht gar besonders aus solchen Informationen Schlüsse für die eigene Leistungs politik gezogen werden können. Auf die Notwendigkeit solchen Handelns ist hingewiesen worden. Zusammenfassend geht es grundsätzlich darum, unter Verwendung von Informationen über Nachfrager eine effiziente und effektive, möglichst antizipative Leistungsgestaltung zu steuern.

Das Beispiel „Whirlpool“ spiegelt dabei einen klassischen linearen Prozess wider. Im Zusammenhang mit den skizzierten Multimedia-Wertschöpfungsnetzwerken ergibt sich nun eine entscheidende Veränderung. Informationen über die Nachfragerseite werden nicht mehr an einer Schnittstelle am Ende eines Wertschöpfungsprozesses generiert, sondern an mehreren Schnittstellen zu Wertschöpfungssektoren, die wiederum zusammen eine Art Wertschöpfungskonglomerat darstellen. Für den erwähnten Prozess des Konsumierens eines Films als Kombination verschiedener Wertschöpfungssektoren ergibt sich dabei die Tatsache, dass am Ende eines Prozesses Informationen zu verschiedenen Leistungen, die auch separat angeboten werden können, verarbeitet worden sind. Für die Rezeption eines Films über das Internet ergeben sich Informationen zum Inhalt (Welcher Film?), zur Übertragung (DSL, ISDN etc.), zur Navigation (Welcher Browser?) und zum Endgerät, das zwangsläufig internetfähig sein muss. Somit sind i. d. R. auch unterschiedliche Unternehmen betroffen, sofern es sich nicht um ein horizontal und vertikal integriertes, der Konvergenz angepasstes Unternehmen wie AOL Time Warner (vgl. Keuper 2002, S. 621) handelt, das alle Wertschöpfungssektoren abbildet. So multipliziert sich quasi der Informationseinfluss eines Nachfragers mit der Anzahl der im Leistungsbündel involvierten Wertschöpfungssektoren.

Insofern kann konstatiert werden, dass die Bedeutung der Nachfrager als Informationsquelle mit dem Entstehen neuer Märkte und Leistungsbündel im Zuge der skizzierten Konvergenz sich noch dynamischer entwickelt.

3.2 Virtuelle Gemeinschaften: Nachfrager und „Prosumer“

Neben den erörterten nachfragerspezifischen Veränderungen im Kontext der Internetökonomie soll im folgenden das Phänomen der virtuellen Gemeinschaften bzw. *virtual communities*² diskutiert werden, das die zunehmende (Pro)Aktivität des Nachfragers kennzeichnet und den Aspekt der Informationsversorgung, der im Zusammenhang der Informationsasymmetrie bereits angeschnitten worden ist, tangiert.

Grundsätzlich lassen sich ein erhöhtes Kooperations- und Koordinationsniveau auf der Nachfragerseite feststellen (vgl. Wirtz 2001, S. 175). Diese Beobachtungen sind auf die Ubiquität der IuK-Technologie zurückzuführen, die eine zeit- und ortsunabhängige Koordination mehrerer Interaktionspartner in einem globalen Rahmen kostengünstig ermöglicht.

Eine ausführliche Definition zum Begriff der virtuellen Gemeinschaft liefern Leimeister und Krcmar:

Eine Virtual Community ist eine besondere Form von Gemeinschaft und damit eine Unterart sozialer Gruppen. Sie ist ein Zusammenschluss von Menschen mit einem Bedürfnis nach Information und Interaktion oder dem Bedürfnis, eine spezifische Rolle in einer Gemeinschaft auszufüllen. Grundlage und verbindendes Element ist eine Idee oder ein Ziel, das auf Basis von impliziten oder expliziten Verhaltensregeln verfolgt wird. Die Interaktion wird durch eine technische Plattform vermittelt und unterstützt, die den Aufbau von Vertrauen und einem Gemeinschaftsgefühl auch ohne die unmittelbare physische Präsenz der Gemeinschaftsmitglieder ermöglicht. Zusammen mit dem technischen Subsystem, bestehend aus der Plattform des Community-Systems und der Infrastruktur des Internet, bilden virtuelle Gemeinschaften damit interdependente soziotechnische Systeme. (Leimeister/Krcmar 2002, S. 5)

Die Ursprünge reichen dabei bis Mitte der 1980er Jahre zurück. Die virtuelle Gemeinschaft „The Well“ (<http://www.well.com>) besteht bspw. seit 1985.

Armstrong und Hagel unterscheiden dabei vier Ausprägungsformen elektronischer Gemeinschaften:

- Transaktionsgemeinschaften, die primär dem Kauf und Verkauf von Leistungen dienen,
- Interessengemeinschaften, die unterschiedlichste Inhalte thematisieren,
- Phantasiegemeinschaften,
- Beziehungsgesellschaften, die v. a. emotionale Lebensaspekte zum Inhalt haben (vgl. Armstrong/Hagel 2000, S. 197ff.).

² Verbreitete alternative Bezeichnungen sind „elektronische Gemeinschaften“ oder „Online-Gemeinschaften“

Im folgenden soll der Fokus auf die eher langfristig konzipierten Interessengemeinschaften (vgl. Wirtz 2001, S. 176) gelegt werden.

Neben der quasi klassischen Methode, Beiträge per E-Mail über E-Maillisten zu verteilen, werden in diesem Kontext heute Kommunikationsinhalte i. d. R. über Online-Foren ausgetauscht, in denen Beiträge thematisch und chronologisch verwaltet und registrierten Teilnehmern zugeordnet werden. Im Gegensatz zu Transaktionsgemeinschaften findet hier ein intensiverer Kommunikationsaustausch statt (vgl. Armstrong/Hagel 2000, S. 198). Die Teilnahme an virtuellen Interessengemeinschaften ermöglicht die Befriedigung der menschlichen Bedürfnisse nach Information und Kommunikation. Dabei wächst ein themenkonzentrierter Informationspool, aus dem einerseits neue Mitglieder oder jegliche Besucher in kurzer Zeit Wissen über das diskutierte Thema aufbauen und den andererseits länger partizipierende Mitglieder als Basis weiterführender Beiträge nutzen können. An dieser Stelle lassen sich drei der fünf zentralen Merkmale von virtuellen Gemeinschaften nach Armstrong und Hagel erkennen:

- Ein spezifischer Interessenschwerpunkt,
- Das Vermögen, Inhalt und Kommunikation zu integrieren,
- Die Verwendung von Informationen, die Mitglieder bereitstellen³ (Armstrong/Hagel 1997, S. 23).

Armstrong und Hagel haben diese mit der Intention definiert, virtuelle Gemeinschaften als Kernobjekt eines eigenständigen Geschäftsmodells zu bestimmen, was sich jedoch in dieser Form nicht durchgesetzt hat.

Es wird deutlich, dass diese Merkmale auf jede Form virtueller Gemeinschaften zutreffen. Sie können also als Basismerkmale angesehen werden.

Grundsätzlich kann für die Integration in virtuelle Gemeinschaften ein bestimmtes Maß an intrinsischer Motivation seitens der Mitglieder festgestellt werden (vgl. Seufert/Moisseeva/Steinbeck 2002, S. 7). Diese ist bei einer Teilnahme aus rein privater Motivation mit vollständiger Freiwilligkeit sicherlich stärker ausgeprägt als bei einer Teilnahme aus bspw. beruflichen Gründen.

In der Betrachtung der Themen ergeben sich aus Sicht von Unternehmen zwei Dimensionen: unternehmensautarke Inhaltsgegenstände wie Sportarten, Orte, Länder, Pflanzen etc. einerseits sowie auf der anderen Seite unterneh-

³ Die weiteren Merkmale in dieser Aufstellung sind „der Zugang zu konkurrierenden Anbietern“ und „eine kommerzielle Orientierung“ (Armstrong/Hagel 1997, S. 24)

mensspezifische Themen wie Produkte, Produktgattungen oder evtl. ein Unternehmen selbst. Für die ein Unternehmen betreffenden Communities sind wiederum diejenigen zu unterscheiden, die von den Unternehmen selbst initiiert worden sind. Solche Communities sind von den Unternehmen als Kundenbindungsinstrument im E-Business schnell erkannt worden (vgl. Bullinger et al. 2002, S. 24), wobei unter E-Business pragmatisch die Durchführung vollständiger Leistungsaustauschprozesse über elektronische Netze verstanden werden soll. In diesem Zusammenhang wird deshalb auch von *business communities* gesprochen. Zusätzlich ergibt sich für die Unternehmen ein erheblicher Informationswert, da in der Analyse der Beiträge Nutzerpräferenzen erkannt und daraus Rückschlüsse für die Leistungsangebotspolitik gezogen werden können. Zudem kann sich ein Unternehmen ein präzises Bild über seine Zielgruppe verschaffen.

Auf der anderen Seite stehen die Teilnehmer der Community. Sie sind bereits Kunden des initiiierenden Unternehmens oder potentielle Nachfrager. Sie profitieren von der Menge und Qualität der Informationen im Informationspool, die im gemeinsamen Austausch generiert worden sind. So besteht u. a. ein konkreter Mehrwert in der Information bzgl. Transaktionsentscheidungen (vgl. Bullinger et al. 2002, S. 25). Da virtuelle Gemeinschaften primär Interessengemeinschaften der Mitglieder sind, die eine größtmögliche Auswahl hochwertiger Ressourcen zur Verfügung stellen möchte, impliziert dieser Aspekt auch den Zugang zu konkurrierenden Unternehmen (vgl. Armstrong/Hagel 1997, S. 24), bspw. wenn diese in den Augen der Gemeinschaft höherwertige Leistungen erstellen. Entsprechend erfüllt sich hier das vierte Charakteristikum nach Armstrong und Hagel (siehe S. 29, Fußnote 3).

So lässt sich konstatieren, dass *virtual communities* zur Beseitigung von Informationsasymmetrien und folglich zur Erhöhung der Markttransparenz beitragen. Zentraler innovativer Aspekt hierbei ist, dass nicht die Unternehmen, sondern die Mitglieder der Gemeinschaft die Informationen erarbeiten, aus denen letztlich Mehrwerte entstehen. Somit transformiert sich die Position des Nachfragers über die eines Nutzers einer Communitysoftware zu der eines Informationsproduzenten.

Bullinger et al. fassen diesen Zusammenhang unter der Bezeichnung „Prosumertum“ (Bullinger et al. 2002, S. 25) als ein Alleinstellungsmerkmal für *business communities* aus der Perspektive des Netzwerks bzw. der Gemeinschaft wie folgt zusammen:

Der Kunde [...] arbeitet durch seine Interaktion mit an der Wertschöpfung des Netzwerks. Er wird vom einfachen Konsumenten zum Co-Produzenten. Der Business Community wachsen so Werte zu, die ein einzelner nicht zu erstellen imstande ist. (Bullinger et al. 2002, S. 25)

3.3 Emanzipation der Nutzer

Die Ausführungen in den vorangegangenen Kapitel haben gezeigt, dass sich Rolle und Position der Nachfrager mit der Verbreitung und Etablierung der IuK-Technologie grundsätzlich verändert haben. So kann schlussendlich eine deutliche Zunahme der Marktmacht der Nachfrager konstatiert werden, die auf die Koordination und Kooperation von Nachfragepotential zurückzuführen ist, aus der letztendlich Vorteile im Markt resultieren (vgl. Wirtz 2001, S. 175). Mit den virtuellen Gemeinschaften ist dabei ein Prinzip eingehender beleuchtet worden. Im folgenden wird zunehmend der Begriff eines Nutzers bzw. Users oder Anwenders verwendet werden, der sich grundsätzlich daraus ergibt, dass jegliche Teilnahme an Informationsaustauschprozessen im Internet die Bedienung von Software voraussetzt. Zudem führt der Gebrauch des Nutzerbegriffes auf den technischen Hauptteil dieser Arbeit hin.

Unter dem Schlagwort „Social Software“ können in jüngerer Vergangenheit weitergehende Veränderungen sozialer Interaktion über elektronische Netze auf technischer Ebene registriert werden. Dabei kann zum Verständnis des Begriffes „Social Software“ die pragmatische Definition von Shirky gelten. Demzufolge ist Social Software „software that supports group interaction“ (Shirky 2003). Przepiorka erwähnt zusätzlich Kommunikation und Zusammenarbeit (vgl. Przepiorka 2006, S. 13), wobei Interaktion Kommunikation als explizit wechselseitig spezifiziert (vgl. Sjurts 2004, S. 318). Entsprechend ist die einer virtuellen Gemeinschaft zu Grunde liegende Anwendung auch als Social Software einzustufen. Was jedoch im Zuge der Verbreitung der Weblogs bzw. Blogs, Wikis oder Social Bookmarking-Systeme u. a. zu analysieren ist, bezeichnet eine weiterführende Entwicklung.

In der früheren Entwicklungsphase des World Wide Web bedeuteten Entwicklung, Bereitstellung und/oder Betrieb der technischen Infrastruktur als Voraussetzung für die Existenz virtueller Gemeinschaften auf Grund der noch gerin-

gen Verbreitung elektronischer Netze und darauf aufsetzender Technologie einen erhöhten Aufwand an Kosten. Dieser Kostenaspekt galt auf der anderen Seite auch für den Zugang.

Aus dieser Perspektive wird deutlich, warum einzelne Privatpersonen i. d. R. nicht als Initiatoren von virtuellen Gemeinschaften auftreten und folglich auch den Themenschwerpunkt vorgeben konnten und so die Initiative für die Gründung virtueller Gemeinschaften verstärkt von Unternehmen ausging, da diese, wie erwähnt, das Potential virtueller Gemeinschaften im Kundenbeziehungsmanagement erkannten (oder das Potential für ein eigenständiges Geschäftsmodell zusprachen, s. o.). Daraus resultiert eine Interaktion Initiator/Unternehmen – Kunde bzw. 1 Unternehmen – n Kunden oder *one-to-many* (sowie auch Kunden untereinander: *many-to-many*), in der Unternehmen das Nachfragepotential in eine konkrete Nachfrage umzuwandeln versuchen und die Nachfrager auf der anderen Seite ihre Bedürfnisse nach Information und Kommunikation befriedigen und aus der Informationsverfügbarkeit Vorteile für Transaktionen (das subjektiv beste Produkt in Preis/Leistung) ableiten wollen.

In diesem Zusammenhang ergibt sich die Betrachtung hinsichtlich des Nachfragers als Co-Produzent bzw. „Prosumer“. Die Initiatoren der Communities, die Unternehmen, bestimmen dabei die Inhalte, i. d. R. ihre Leistungen, und stellen die technische Infrastruktur zur Verfügung.

In der exemplarischen Betrachtung der Existenz und der Verbreitung von Blogs als einfach zu bedienende Publikationssysteme mit Interaktions- und Verknüpfungsmechanismen, häufig als Online-Tagebücher verwendet, oder Wikis als Publikationssysteme, die jedem Benutzer das Ändern der Seiteninhalte erlauben, zeigt sich, dass Unternehmen zusehends in Informationsaustauschprozessen umgangen werden und entsprechend ihren Einfluss auf die Informationsausprägung (Welche Inhalte werden diskutiert?) verlieren. So werden unabhängig, d. h. ohne Einfluss von Unternehmen Inhalte von Nutzern für Nutzer erstellt. Der Weblog-Suchdienst Technorati (<http://www.technorati.com>) z. B. indexiert dabei bereits nach eigenen Angaben über 50,1 Millionen Weblogs (Stand August 2006).

Die ausführlich dargestellten Entwicklungen in der IuK-Technologie haben zu einer Kostenstruktur geführt, die den Einzelnen nicht mehr von der Nutzung der skizzierten technischen Möglichkeiten ausschließt. Weiter verschafft die Verfügbarkeit freier oder kostenloser Software bspw. für Weblogs (z. B. Drupal, Textpattern, WordPress u. v. a.) und deren einfache Installation Privat-

personen, die Nutzer, weitere Unabhängigkeit gegenüber Unternehmen, weil diese lediglich noch für die Bereitstellung von Speicherplatz und Datenbanken auf Webservern sowie den Zugang zum Internet „gebraucht“ werden. Unterstützend wirkt sich auch die Tatsache aus, dass das Internet inzwischen Bestandteil der alltäglichen Mediennutzung ist und Lerneffekte zum Tragen kommen, die durch die Vermittlung entsprechender technischer Kompetenzen bereits in der Schule immer früher zur Geltung kommen.

So kann bereits an dieser Stelle für die Erstellung und das Publizieren von Informationen und Inhalten im Internet ein „Jedermann“-Prinzip analysiert werden.

Dieser Prozess soll analog zur Bezeichnung dieses Kapitels als Emanzipation der Nutzer gekennzeichnet werden. In diesem Kontext kann weiter festgestellt werden, dass der Begriff des „Prosumers“ nicht mehr die (neue) Realität erfasst, wenn sich der (Inhalte)Produktionsaspekt unabhängig von Nachfrage oder Konsum ausprägt. Auf eine Wortschöpfung soll hier jedoch verzichtet werden.

Der Paradigmenwechsel wird im folgenden im Zusammenhang mit der Diskussion zu Phänomenen, die unter der Bezeichnung „Web 2.0“ subsumiert werden, eingehender betrachtet. Für diese Ausführungen sollen jedoch zunächst ausgewählte, zentrale Begriffe bezeichnende Termini, die den Leserkapitelübergreifend begleiten werden, kurz definiert werden.

Unter „Standard“ wird die Definition einheitlicher Regeln zur Verarbeitung, Speicherung und Übertragung von Informationen mit dem Ziel der Realisierung oder Vereinfachung der Interaktion zwischen Systemkomponenten verstanden (vgl. Buxmann 2001, S. 435). Die Betonung soll auf den Aspekt gelegt werden, dass Standards zwischen unterschiedlichen Komponenten eines Systems oder systemübergreifend Interaktion erst ermöglichen.

Für die Verwendung von Uniform Resource Identifier (URI) und Uniform Resource Locator (URL) gilt im folgenden, dass URLs eine Untermenge von URIs bezeichnen, die, neben der Funktion, eine Ressource zu identifizieren, eine Methode zum Lokalisieren über den primären Zugriffsmechanismus bezeichnen (vgl. IETF RFC 3986, S. 7), z. B. in einem Netzwerk wie dem Internet. Eine Ressource wird dabei als ein Informationsobjekt wie bspw. eine Webseite oder ein Bild verstanden, das einen eindeutigen URI aufweist (vgl. Geroimenko 2004, S. 137).

Die Bezeichnungen „Open Source“ und „freie Software“ sollen synonym für Software verwendet werden, die vier „Freiheiten“ (Zendel 2005, S. 152) garantiert:

Die Freiheit, das Programm zu jedem Zwecke auszuführen. Die Freiheit, Kopien des Programms anzufertigen und zu verbreiten, sowohl kostenpflichtig als auch kostenlos. Die Freiheit, die Funktionsweise des Programms zu studieren – Voraussetzung dafür ist der Zugang zum Quellcode. [...] die Freiheit, das Programm an eigene Bedürfnisse anzupassen oder zu verbessern – inklusive der Freiheit, das veränderte Programm zu verbreiten. (Zendel 2005, S. 152)

In dieser Erklärung kann eine Konformität zu den Kernaspekten der GNU General Public License, GPL (<http://www.gnu.org/licenses/gpl.html>), festgestellt werden.

4 Paradigmenwechsel: Web 2.0?

In der vorangegangenen Analyse aktueller Entwicklungen ist die Intensivierung internetspezifischer Aktivitäten seitens der Nutzer und die zunehmende Fokussierung der User auf die eigene Perspektive, die Nutzerseite, angedeutet worden. Die wesentlichen unterstützenden Entwicklungen bzw. Treiber sind bereits angemerkt worden, einen Überblick gibt noch einmal folgende Auflistung:

- individualistische Nachfrager/Nutzer, die intrinsisch motiviert Informationsaustauschprozesse untereinander mittels unterschiedlicher multimedialer Anwendungen organisieren und unterstützen,
- Ubiquität der IuK-Technologie und einhergehender Preisverfall sowie Integration der Technologie in alle Lebensbereiche,
- wachsende Verfügbarkeit freier oder kostenloser Software, die Informationsaustauschprozesse unterstützt.

Im Grundsatz ist hier die Wechselwirkung aus *market pull* (Punkt 1) und *technology push* (Punkte 2 und 3) zu erkennen.

Die Individualisierung der Nachfrager ist unter der Verwendung der Erkenntnisse Maslows im Ansatz zu erklären versucht worden. Neben diesem Aspekt prägen sich zusätzlich parallel die o. g. intrinsische Motivation, Inhalte ohne Entgelt zur Verfügung zu stellen sowie zusätzlich die intrinsische Motivation von Softwareentwicklern, Anwendungen Open Source anzubieten (vgl. Kuhlen 2005, S. 35), aus.

Kuhlen sieht in der Heterogenität der Open-Source-Bewegung zudem einen Spiegel der Gesellschaft (vgl. Kuhlen 2005, S. 39).

Die Betrachtung der Bedürfnispyramide von Maslow kann letztendlich ebenfalls als Erklärungsgrundlage für die zunehmende skizzierte Motivation der Nutzer und Entwickler hinzugezogen, das Streben nach Selbstverwirklichung also als Motor dieser Motivation angesehen werden.

Unter dem Schlagwort „Web 2.0“ werden zur Zeit neue Entwicklungsströmungen diskutiert. Handelt es sich hierbei um tiefgreifendere Entwicklungen, die die Kennzeichnung als Evolution rechtfertigt, was durch den Ausdruck „2.0“ suggeriert wird?

Die Wortschöpfung „Web 2.0“ lässt sich auf Tim O'Reilly, Gründer und CEO des O'Reilly Verlages, und Dale Dougherty, Vizepräsident des O'Reilly Verla-

ges, zurückführen (vgl. O'Reilly 2005, S. 1), die in diesem Zusammenhang im Oktober 2004 die „Web 2.0 Conference“ (<http://www.web2con.com>) ins Leben riefen, um die Diskussion über die stärker wahrnehmbaren Phänomene auf eine breitere Ebene zu tragen. Auf der Web 2.0 Conference im November 2006 waren Redebeiträge von Vertretern von Google, Yahoo!, AOL, Amazon, NBC und anderen namhaften Unternehmen zu hören, was den Bedeutungsrahmen von „Web 2.0“ als Debattengegenstand aufzeigt.

„Web 2.0“ wird im Internet intensiv diskutiert, Google liefert für „Web 2.0“ (als Phrase) 109 Millionen, für „Web2.0“ 31,3 Millionen Treffer (Stand 14.8.2006). Im Vergleich dazu liefert der GBV Regionalkatalog Hamburg der Hamburger Hochschulen für „Web 2.0“ sieben Treffer, für „Web2.0“ null Treffer in der Universaloption „Alle Wörter [ALL]“. Von diesen sind drei nicht themenrelevant, die übrigen vier Treffer behandeln Programmierung im Zusammenhang mit AJAX (Stand 15.8.2006).

Dies zeigt zum einen die Neuartigkeit dieses Themas, zum anderen unterstützt diese Tatsache die Feststellung, dass „Web 2.0“ kein allgemein anerkanntes, empirisch entwickeltes Konzept und demnach einen Begriff benennt. Was ist nun kennzeichnend für die Tendenzen, die unter dem Stichwort „Web 2.0“ diskutiert werden?

O'Reilly und Dougherty formulierten in einem initialen Brainstorming folgende Dichotomie von Merkmalen, um sich dem Themenkomplex zu nähern:

Web 1.0	Web 2.0
DoubleClick	Google AdSense
Ofoto	Flickr
Akamai	BitTorrent
mp3.com	Napster
Britannica Online	Wikipedia
personal websites	blogging
evite	upcoming.org and EVDB
domain name speculation	search engine optimization
page views	cost per click
screen scraping	web services
publishing	participation
content management systems	wikis
directories (taxonomy)	tagging („folksonomy“)
stickiness	syndication

Tab. 2: Merkmale des „Web 2.0“ (O'Reilly 2005, S. 1)

Wie bereits erwähnt, bezeichnet „Web 2.0“ (noch) keinen anerkannt definierten Begriff, der einen eindeutigen gedanklichen Rahmen umschließt. Vielmehr verhält es sich derart, dass unter „Web 2.0“ zunächst charakteristische Prinzipien, Praktiken und Techniken zusammengefasst werden, von denen sich einige als Kernmerkmale herauskristallisieren, die als Ansatzpunkte für eine Begriffsdefinitionsarbeit gelten können. O’Reillys „meme map“ (O’Reilly 2005, S. 1) in Abbildung 10 zeigt einen Status quo dieser Erarbeitungen.



Abb. 10: „Meme map“ Web 2.0 (O’Reilly 2005, S. 1; eigene Darst.)

Im folgenden sollen ausgewählte Aspekte untersucht werden, die einige Punkte aus Tabelle 2 und Abbildung 10 wieder aufgreifen.

4.1 Strategische Positionierung: Das Internet als Plattform

In der Analyse der durch elektronische Netze induzierten Veränderungen in der Internetökonomie wird in der Literatur häufig exemplarisch die Entwicklungsgeschichte von Netscape Communications und dessen wichtigsten Produktes, der Browser Netscape Communicator, angeführt (z. B. in Picot/Reichwald/Wigand 2003, S. 22f.). Netscape soll auch hier in den folgenden Ausführungen als beispielhafter Repräsentant für die Handlungsprinzipien und –strategien in der ersten Entwicklungsphase des World Wide Web dienen.

Netscapes Strategie war, über die kostenlose Abgabe des Browsers und die daraus folgende Dominanz in diesem Markt eine Nachfrage von Unternehmen nach hochpreisiger Serversoftware zu generieren (vgl. Picot/Reichwald/Wigand 2003, S. 22f.; O'Reilly 2005, S. 1) und über die Kontrolle des Standards, wie über das Internet transferierte Inhalte in einem Browser dargestellt werden, eine langfristige Marktmacht aufzubauen. Netscape proklamierte in diesem Zusammenhang den „Webtop“ als Ersatz für den Desktop (vgl. O'Reilly 2005, S. 1). Dass Netscapes Strategie fehlschlug, weil Microsoft mit der ebenfalls kostenlosen Abgabe des Internet Explorers und der konsequenten Ausnutzung von Netzeffekten über das dominierende Betriebssystem „Windows“ Netscape verdrängen konnte, ist bekannt und braucht nicht weiter dargelegt werden. Heute ist Browsersoftware Massenware, die zudem i. d. R. kostenlos von den Herstellern oder Open Source von unabhängigen Entwicklern zur Verfügung gestellt wird (z. B. Firefox, Internet Explorer, Netscape u. a.). Den eigentlichen Wert stellen die über das Internet absetzbaren Leistungen dar (vgl. O'Reilly 2005, S. 1).

Netscapes Strategie basierte auf altbekannten Prinzipien der Softwareindustrie, die sich für die PC-Plattform mit einem dominierenden Betriebssystem bewährt hatten:

- Entwicklung, Durchsetzung und Kontrolle eines proprietären Standards⁴,
- Entwicklung und Kontrolle der Schnittstellen (ABI und API),
- Entwicklung von versionierter Software auf Basis der Standards,
- Erlöse aus dem Verkauf von Softwarelizenzen.

⁴ Dieser Prozess kann als „marktliche Standardisierung“ (Buxmann 2001, S. 434) bezeichnet werden

Demnach war Netscapes Praxis, das Internet als (Geschäfts)Plattform zu nutzen, lediglich der Transfer traditioneller Strategien auf eine neuartige Architektur. Aus dem Scheitern Netscapes kann letztlich geschlossen werden, dass diese neue Plattform neue Strategien erfordert.

Im Gegensatz dazu soll anhand der Suchmaschine Google aufgezeigt werden, welche Prinzipien, aus der ökonomischen Perspektive betrachtet, erfolgversprechender zu sein scheinen.

Google ist seit jeher eine reine Internetanwendung, die eine Dienstleistung zur Verfügung stellt und nicht als einzelne lizenzierte und versionierte Software verkauft wird. Die Nutzung der zentralen Leistung, die Internetsuche, ist kostenfrei, Erlöse werden z. B. über kontextspezifische Werbung und spezielle Services erzielt. Google konzentriert sich in der Leistungserstellung ausschließlich auf Ressourcen, die quasi in dem Bereich „zwischen“ den Browsern und den bedienenden Endnutzern, also im Netz selbst zu finden sind. Entsprechend profitiert (genauer: expandiert) Google dann, wenn die Zahl der Ressourcen und Hyperlinks, auf denen der Gewichtungsalgorithmus „PageRank“ basiert, steigt. Hier wird deutlich, dass das Potential in der Gesamtheit und nicht nur der stark frequentierten Webseiten begründet ist und der (geschäftliche) Aktionsrahmen diese Gesamtheit berücksichtigt. Diese Sichtweise bezeichnet Anderson bildhaft als „the long tail“ (Anderson 2004, S. 2).

Im Rahmen seiner strategischen Positionierung im Internet, die Plattform, negiert Google die erwähnten traditionellen Prinzipien. Google erlaubt bspw. für Google Maps den Zugriff und die Einbindung von Daten in Fremdseiten über offengelegte dokumentierte Schnittstellen und gehört damit zu zahlreichen anderen Internetunternehmen, die offene Schnittstellen anbieten. Eine Übersicht hat bspw. Musser unter <http://www.programmableweb.com/apis> zusammengetragen.

Ein anderer, für den Wert von Schnittstellen konstitutiver Aspekt ist der Datenaspekt. Die Kernkompetenz von Google ist Daten- bzw. Datenbankmanagement, eine Kompetenz, die für Netscape nie eine Rolle gespielt hat (vgl. O'Reilly 2005, S. 1). Google ist im Prinzip eine hochspezialisierte Datenbank, deren Wert allein von den verwalteten Daten abhängt, die wiederum ohne die Software nicht handhabbar wären. O'Reilly konstatiert:

In fact, the value of the software is proportional to the scale and dynamism of the data it helps to manage. (O'Reilly 2005, S. 1)

Google konnte mit diesem Geschäftsmodell für das zweite Quartal 2006 Erlöse in Höhe von knapp 2,5 Milliarden US\$ erzielen und damit eine Steigerungsrate zum Vorjahreszeitraum von 77% vorweisen (vgl. Google Inc. 2006).

An dieser Stelle muss angemerkt werden, dass Google und seine Dominanz als Suchdienst für Internetressourcen mit etwa 80% Anteil am Nutzungsmarkt (vgl. Dreyer et. al 2005, S. 11) grundsätzlich einer kritischen Betrachtung in Bezug auf Meinungsvielfalt zu unterziehen ist.

Es kann analysiert werden, dass das Internet als eigenständige Plattform zu betrachten ist. D. h., dass das Internet selbst Erfolgspotentiale bereitstellt, die auf dem Gut Information in Kombination mit dem Prinzip Masse (viele Nutzer, viele Ressourcen) sowie Interaktion beruhen. Dies berechtigt zu einer absoluten, d. h. keine weiteren Tätigkeitsfelder außerhalb des Internets verfolgend und strategischen Positionierung. Im Gegensatz dazu steht die Ansicht, das Internet als reinen Vertriebskanal anzusehen und das Prinzip der Massenmedien zu imitieren (*one-to-many*).

Diese Plattform basiert auf offenen Standards wie bspw. TCP/IP, HTTP, XML und HTML (hierzu auch Kuhlen 2005, S. 41), die wiederum u. a. die massive Verbreitung der Basisinfrastruktur und somit deren Erfolg, Stichwort Netzeffekte, implizieren. Der Versuch, proprietäre Standards zu etablieren und das Prinzip der traditionellen Softwareindustrie zu transferieren, käme quasi dem Bemühen gleich, eine parallele konkurrierende Architektur installieren zu wollen. Bereits in der Betrachtung der Humanressourcen und der beteiligten Institutionen, die an der Weiterentwicklung der Standards im Internet beteiligt sind, zeichnet sich der notwendige Aufwand und folglich das Risiko derartiger Bemühungen ab.

Der zweite Aspekt umfasst die Bedeutung der Daten und der Datenmenge, die eine Software verwaltet (und fortführend die Möglichkeit zur Verwaltung von Informationen analog zu der in Abb. 1 skizzierten Hierarchie). Offene dokumentierte Schnittstellen unterstützen die Verbreitung und gleichsam die Sammlung von Daten, Beispiel Google Maps. Geodaten können via HTTP über eine Schnittstelle in andere Webseiten eingebunden werden (Verbreitung), der Zugriff, d. h. welche Daten abgefragt worden sind sowie der URL der Zielseite wiederum, von der die Anfrage gesendet wird, können von Google als Daten gespeichert werden (Sammlung).

Der Gedanke von Kelly, der ein einziges „Web OS“ (Kelly 2005, S. 5) und die Irrelevanz von Desktop Betriebssystemen aufkommen sieht (vgl. Kelly 2005, S. 5), scheint nicht abwegig. Letztlich entspräche dies auch einer konsequenten Entwicklung im Rahmen der erwähnten Konvergenz auf technologischer Ebene.

4.2 Engagement der Nutzer: Kollektive Intelligenz

In der Betrachtung der Internetunternehmen, die bereits in den 1990er Jahren gegründet worden sind und heute erfolgreich operieren, kann analysiert werden, dass ihr Erfolg darauf beruht, ihre Kunden von Anfang an als Co-Produzenten von Informationen und Inhalten in die Wertschöpfung mit einzubeziehen. Das Sortiment des Buchhändlers Amazon unterscheidet sich unwesentlich von dem vieler Konkurrenten wie bspw. Libri, alle Händler können von den Verlagen neben den Basisdaten die gleichen Zusatzinformationen wie Buchcover, Foto des Autors etc. erhalten. Amazon jedoch motiviert Kunden über Anreizprinzipien wie Rabatte (extrinsische Motivation) z. B. zum Verfassen von Rezensionen und reichert somit den originären Datensatz „Buch“, der in gleicher Weise allen Konkurrenten zur Verfügung steht, mit zusätzlichen Inhalten an und kann sich so von seinen Konkurrenten differenzieren. Außerdem kann dieser individuelle Datenbestand nur schwer von etablierten, vor allem jedoch potentiellen Konkurrenten imitiert werden, so dass diese Strategie auch als Aufbau struktureller Markteintrittsbarrieren angesehen werden kann.

Es ist bereits bemerkt worden, dass das Konzept des Co-Produzenten und Kunden bzw. „Prosumers“ überholt zu sein scheint, da das Publizieren von Inhalten mehr und mehr unabhängig von Unternehmen und ihren Anreizprinzipien erfolgt und zusätzlich Möglichkeiten und Mechanismen zur Interaktion und Beteiligung vor allem auf technischer Ebene geschaffen werden. Exemplarisch für diese Entwicklung steht die „Blogosphäre“, die Gesamtheit aller Weblogs, die die höchste Abstraktionsebene dieses interpersonalen Netzwerkes beschreibt (vgl. Bucher/Büffel 2005, S. 141). Zum einen können die Intermediäre der Medienindustrie wie Journalisten oder Presseagenturen umgangen werden (vgl. Picot/Fischer 2006, S. 8). Andererseits können auch die Medien selbst als Informationsvermittler ausgeschaltet werden, so dass die Blogosphäre eine zweistufige Disintermediation für die Informationsbeschaffung ermöglicht.

Wie in Tabelle 2 dargestellt, beschreibt eine Beobachtung den Übergang vom Publizieren (*publishing*) zur Beteiligung (*participation*). Blogs unterstützen durch ihre Hypermedialität Kommunikationsprozesse (vgl. Picot/Fischer 2006, S. 5). Genauer: Sie erlauben Interaktion, da der Kommunikation explizit eine Wechselseitigkeit zu Grunde liegt, z. B. dadurch, dass Einträge kommentiert werden können. Andere Mechanismen bzw. Techniken ermöglichen Kommunikationsprozesse in effizienterer Weise:

- RSS beschreibt ein Verfahren, den Inhalt eines Blogs oder beliebiger anderer Webseiten in einem XML-Format bereitzustellen und z. B. Software so das Einbinden der Inhalte vieler Blogs in einer Oberfläche zu erlauben, wobei eine solche Software das regelmäßige Abrufen der Daten übernehmen kann;
- Permalinks bzw. Permanentlinks stellen die langfristige Referenzierung eines Blogeintrags sicher;
- Trackback erlaubt das Registrieren der Blogeinträge, die einen Link auf einen eigenen Beitrag enthalten.

Mit der Blogosphäre ist ein Informationsraum entstanden, dessen hypertextuelle Inhaltsbasis auf Grund der dargelegten Ursachen auf der Nutzerseite selbst, der infrastrukturellen Entwicklungen und mittels der unterstützenden Kommunikations- und Interaktionsmechanismen stetig anwächst. Ähnlich dem Prinzip des *peer-reviews* entscheiden die Nutzer auf gleicher Ebene, ob Beiträge von Autoren für lesenswert erachtet werden, indem sie sie bspw. verlinken. Im Zusammenhang mit der Suchmaschine Google und sein „Page-Rank“-Verfahren zur Gewichtung von Webseiten, das auf der Linkstruktur aufsetzt und z. B. die Verlinkung einer Webseite A auf einer Webseite B als ein Votum von B für A bewertet (vgl. Google Inc. 2004), bedeutet das intensive Verlinken in der Theorie, dass Blogs hier hoch gewichtet werden und somit die Voraussetzungen für gute Platzierungen in Googles Ranking aufweisen.

„Kollektive Intelligenz“ bezeichnet dabei anschaulich das Grundprinzip: Einzelne intellektuelle Leistungen wie Inhaltsbeiträge als separate Ressourcen bilden durch die Verknüpfung einen kollektiven Inhaltspool. Mit steigender Anzahl der Verknüpfungen erhöht sich die Erschließbarkeit dieses Pools. Die hochgradige Selbstreferenzierung in der Blogosphäre (vgl. O’Reilly 2005, S. 3; Hövel 2006) unterstützt zum einen die These eines eigenen Informationsraumes, zum anderen zeigt sie die fortgeschrittene Entwicklung des skizzierten

Grundprinzips, was durch die oben genannten Mechanismen weiter unterstützt wird.

Dabei wird die Wirkung von direkten Netzeffekten deutlich. Je mehr Blogs entstehen und Beiträge beigesteuert werden, desto größer wird der Wert für die einzelnen Nutzer, der sich z. B. in der Möglichkeit zur Befriedigung der Informations- und Kommunikationsbedürfnisse äußert. Für Außenstehende erhöht sich die Attraktivität, Mitglied der Blogosphäre zu werden. Dabei basiert der Wert des Netzwerks allein auf dem Engagement der Nutzer, d. h. auf ihren Beiträgen.

Amazon hat dieses Prinzip im Grundsatz implementiert. Dabei ist jedoch festzustellen, dass der Aufbau dieses Informationspools letztlich durch die Gewährung von Rabatten unterstützt wird, also bezahlt werden muss. Junge Internetunternehmen hingegen setzen darauf, die durch den Einzelnen für originär eigene Zwecke produzierten oder gesammelten Daten und Informationen quasi als Nebeneffekt in einen allgemein zugänglichen Pool einzuspeisen bzw. dort parallel zur Verfügung zu stellen.

O'Reilly fasst dies wie folgt zusammen:

Therefore, Web 2.0 companies set inclusive defaults for aggregating user data and building value as a side-effect of ordinary use of the application. (O'Reilly 2005, S. 3)

Ein Beispiel: Die Online-Linkverwaltung del.icio.us (<http://del.icio.us>), die mittlerweile zu Yahoo! gehört, bietet registrierten Nutzern die Möglichkeit, zur eigenen Verwendung und/oder für andere registrierte Nutzer URLs zu speichern sowie diese mit einer Bezeichnung, einer Bemerkung und sog. „Tags“ zur Inhaltsbeschreibung zu versehen. Andere, auch nicht registrierte Personen können z. B. über den Zugriff auf alle Tags die mit diesen Tags verknüpften Links (Einträge bestehend aus URL, Bezeichnung und Bemerkung) einsehen und letztlich über den konkreten URL eine Ressource aufrufen. Andersherum können über Links die mit diesen verknüpften Tags und Nutzer identifiziert werden. Somit stellt jeder persönliche Eintrag gleichzeitig eine Anreicherung des gemeinsamen Informationspools dar. Solche öffentlichen Linkverwaltungen werden auch als Social Bookmarking-Systeme bezeichnet (vgl. Hammond et al. 2005) und stellen somit eine Ausprägung von Social Software-Konzepten dar.

Dabei scheint die Frage nach einem Erlösmodell bei del.icio.us noch ungeklärt. Die Nutzung ist uneingeschränkt kostenfrei, Werbung wird nicht platziert.

Es bleibt allgemein abzuwarten, wie Unternehmen dieses Kollektivpotential zur Erlösgenerierung einsetzen werden.

4.3 Erste Bewertung der Diskussion

In den vorangegangenen Ausführungen sind einige neue Entwicklungen speziell aus der Nutzerperspektive aufgezeigt worden. Andererseits sind Aspekte bzw. Beispiele wie Amazon besprochen worden, die nicht neu sind, aber im Kontrast zum zahlreichen Scheitern vieler Internet-Startups in den Jahren 2000/2001 und der einsetzenden Ernüchterung bzgl. der Beurteilung von Geschäftsmöglichkeiten im Internet zeigen, welche Prinzipien aus ökonomischer Sicht erfolgversprechend sind und welche nicht. Eine erstes Fazit kann lauten, dass die Möglichkeiten der Internetnutzung (ökonomisch, wissenschaftlich, privat) je nach Betrachtung entweder neu definiert oder nun erst tatsächlich ausgeschöpft werden, weil das Internet neben der enormen Verbreitung in seiner technischen und inhaltlichen Gesamtheit mittlerweile mehr und mehr als eigenständige Plattform, als singuläre Architektur betrachtet werden kann, für die entsprechend eigene, ansatzweise genannte Prinzipien gelten. Da die untersuchten Aspekte zudem eine große Zahl an Individuen einerseits involvieren und andererseits auch ungewollt betreffen, z. B. über die Neugestaltung von Arbeitsplätzen und -umgebungen, die Wirkungen also weitreichend sind, kann der Status einer Evolution durchaus zugesprochen werden. Dabei stellt „Web 2.0“ eine (erste) prägante Bezeichnung dar, um die unterschiedlichen Begriffsaspekte sprachlich zusammenfassen und nennen zu können, sicherlich auch mit der Intention einer dadurch verbesserten Vermarktungsfähigkeit von Anwendungen oder Diensten. Der enthaltene Ausdruck „2.0“, der eine Weiterentwicklung gegenüber „1.0“ markiert, ist nach obigem Votum dann berechtigt.

Es muss auch festgestellt werden, dass eine Reduktion von „Web 2.0“ auf konkrete Anwendungen bzw. Dienste wie Wikipedia oder zu Grunde liegende Anwendungstypen wie in diesem Falle Wikis entsprechend nicht zutreffend sein kann. Solche Systeme sind die ermöglichende oder unterstützende Technologie. Einzelne „Artefakte“ wie Wikipedia können möglicherweise als Symbole (vgl. Hornig 2006, S. 62) für die aktuellen Entwicklungen angesehen werden.

Signifikant und beispielhaft für eine nach diesem Fazit und im besonderen unter der Betrachtung des intensivierten Nutzerengagements als antiquiert zu bezeichnende Sichtweise ist die Tatsache, dass die Internet Service Provider für ihre Anschlüsse für Privatanwender i. a. wesentlich höhere Download- als Uploadraten anbieten und so noch immer eine Rollenverteilung wie in den Zeiten der Dominanz der klassischen Massenmedien reflektieren.

4.4 Das Tagging: kollektive Metadaten

Im vorangegangenen Kapitel sind Tags bereits unter dem Aspekt der kollektiven Informationsgenerierung im Zusammenhang mit der Linkverwaltung del.icio.us erwähnt worden. Tags sind beliebige Zeichenketten („software“, „Hochschule“, „0815“, „XML“, „**g**“ etc.), die funktional mit Schlagwörtern vergleichbar sind und zur inhaltlichen Beschreibung einer Ressource von Nutzern individuell und unabhängig, d. h. frei definiert werden. Die Inhaltserschließung im Bibliothekswesen und der Dokumentation unterscheidet beim freien Indexieren Schlag- und Stichwörter derart, dass ein Stichwort der Dokumentationseinheit entnommen wird, während ein Schlagwort intuitiv vom Indexierer vergeben wird (vgl. Gaus 2003, S. 51f.), also nicht im Text vorkommen muss. Bei Ressourcen, die keinen Text enthalten, wären Tags dann immer Schlagwörter. Somit handelt es sich bei Tags um Metadaten, die in ihrer Gesamtheit in einem System, z. B. einer Datenbank (auf höherer Abstraktionsebene: ein Kontext), einen nicht-hierarchischen, also einstufigen Index repräsentieren, wobei diese Schlagwörter hier meist Deskriptoren sind (vgl. Gaus 2003, S. 105).

Der Prozess, eine Ressource mit einem oder mehreren Tags inhaltlich zu erschließen, soll als Tagging bezeichnet werden. Tagging stellt somit eine freie intellektuelle Indexierung dar.

Anhand der Nutzung von del.icio.us soll die Prozedur des Taggings zur Veranschaulichung kurz dargestellt werden, und auch die weiteren Praxisbezüge werden sich hauptsächlich auf diesen Dienst beziehen, was dem Leser ein praktisches Nachvollziehen erlaubt.

Zur Vereinfachung des Vorgangs kann in den Browser, z. B. Firefox oder Internet Explorer, ein Button in die Menüleiste integriert werden, über den der URL und der Titel aus dem (X)HTML `title`-Element einer besuchten Seite per

HTTP-GET-Request direkt an das Eingabeformular gesendet werden kann. Der Nutzer kann dann

- optional einen beschreibenden Freitext verfassen
- sowie optional kein, ein oder mehrere Tags neu definieren oder
- aus der eigenen Tag-Liste, einer Liste von empfohlenen oder einer Liste „populärer“ Tags diese zuweisen.

Solche Tag-Empfehlungen können dann angezeigt werden, wenn der URL bereits von anderen Nutzern gespeichert worden ist.

Nach dem Speichern wird der Nutzer wieder zur ursprünglich besuchten Seite zurück geleitet.

In kürzester Zeit ist so ein neuer Eintrag in den eigenen und den kollektiven Informationspool hinzugefügt worden. Bereits während des Editierens des neuen Eintrags kann der Nutzer selbst von diesem Kollektivprinzip profitieren, wenn ihm die von anderen Nutzern für diesen URL vergebenen Tags vorgeschlagen werden.

4.4.1 Analyse des Prinzips

Wie bereits erwähnt, kann eine Menge von Tags als Index interpretiert werden. Für das Retrieval bedeutet dies ganz allgemein formuliert, dass eine Anfrage nicht mehr auf den Volltext durchgeführt werden muss, sondern auf dessen Repräsentation beschränkt werden kann, die das Ergebnis eines intellektuellen Prozesses ist. Für den Anwender impliziert dies einen verbesserten Zugriff auf die Inhalte, da ähnlich wie bei Klassifikationen oder Thesauri ein Wissensgebiet, eine Domäne, auf eine den Inhalt repräsentierende, an der Nutzung einer bestimmten Zielgruppe orientierten Deskriptorensammlung reduziert wird. Der Aspekt der Zielgruppenoptimierung ist beim kollaborativen Tagging, das zunächst rein subjektiv und egoistisch erfolgt, jedoch neu zu bewerten, da eine übergeordnete Instanz, die die Zielgruppenorientierung interpretiert, nicht existiert.

Für das Tagging ist die Reduktion anhand der in Abbildung 11 dargestellten Tag-Übersicht deutlich sichtbar, welche die Tags des Autors für die bei del.icio.us verwalteten Links darstellt. Dabei zeigt die Schriftgröße des Tags die Häufigkeit deren Zuweisung.



Abb. 11: Darstellung aller Tags des Autors als *tag cloud* bei del.icio.us (Stand 3.12.2006)

Dem Anwender erschließt sich hier auf den ersten Blick, zu welchen Themen und Schwerpunkten Inhalte in dem jeweiligen Account gespeichert sind. Der Zugriff kann zielgerichtet erfolgen und ist nicht mehr davon abhängig, ob der Anwender bspw. für die Freitextsuche im Inhalt vorkommende Stichwörter kennt. Ein weiterer Vorteil ist, dass Aggregatoren Inhalte anhand ausgewählter Tags filtern sowie z. B. mittels RSS bereitstellen können und Nutzer solcher RSS-Feeds Inhalte entsprechend themenspezifisch abonnieren können (vgl. Przepiorka 2005, S. 24).

In diesem Zusammenhang zeigt sich ein Nachteil des Taggings. Wird ein RSS-Feed anhand der Filterung nach einem bestimmten Tag, z. B. „Mac“, generiert, so kann es sein, dass dem Abonnenten Informationen verloren gehen, weil Nutzer neue relevante Inhalte nicht mit dem Tag „Mac“, sondern bspw. mit einem Tag „Apple“ oder „Macintosh“ versehen, da sie in Social Software-Systemen wie del.icio.us diese Freiheit haben.

Dieser Aspekt tangiert das Synonymproblem. Ein Begriff (im Beispiel ein Personalcomputer der US-amerikanischen Firma Apple) kann beim simplen Tagging wie bei del.icio.us nur über autarke Tags und folglich nur über autarke Benennungen im System repräsentiert und mit Inhalten verknüpft werden. D. h., dass verschiedene Tags nicht über spezifische Relationen zum Zwecke der Repräsentation eines Begriffes direkt verknüpft werden können.

Das Prinzip des Taggings steht im Gegensatz zu klassischen Ordnungssystemen wie Klassifikationen oder Thesauri, bei denen es darum geht, eine Domäne mit einer überschaubaren Anzahl an Begriffen und den sie bezeichnenden Deskriptoren (Vorzugsbenennungen eines Begriffes) unter Berücksichtigung der Zielgruppenbedürfnisse zu strukturieren (genauer: vorzustrukturieren) und wie in Thesauri das Vokabular zusätzlich über Relationen, z. B. Quasisynonym, zu kontrollieren. Dabei ist die Intention bspw. bei Klassifikationen i. d.

R. die, einen Inhalt genau einer Klasse zuordnen zu können (vgl. Gaus 2003, S. 67). Für den Aufbau solcher Ordnungssysteme ist zu klären

- welches Sachgebiet erschlossen werden soll und
- wie die Struktur des Nutzerkreises beschaffen ist,

um die Domäne abzugrenzen (vgl. Gaus 2003, S. 169). Der zweite Punkt spiegelt hier die erwähnte Anforderung der Zielgruppenorientierung wider.

Ferner geht es auch darum zu wissen, welcher Art die Informationseinheiten sind, welche Zahl in einem definierten Zeitraum verarbeitet wird und wie viele Nutzer in einem definierten Zeitraum den Informationsbestand nachfragen werden (vgl. Gaus 2003, S. 169).

Hier wird deutlich, dass für Social Software-Systeme im Internet diese Aspekte realistischerweise nur in der Theorie berücksichtigt werden können (der Aufwand einer ordnungssystematischen Erschließung des Internet kann kaum quantifiziert werden), klassische Ordnungssysteme v. a. bzgl. ihres organisatorischen Aufbaus hier also nicht das Mittel der Wahl darstellen.

Insofern kann konstatiert werden, dass Tagging eine konsequente Entsprechung auf die Plattform Internet mit der immensen Menge an Informationen und einer potentiell milliardenstarken Nutzerzahl darstellt. Das Zugeständnis, einen Inhalt mit beliebigen, frei definierbaren Tags zu belegen, korrespondiert mit der Tatsache, dass die Struktur der potentiellen Nutzerseite (Interessen, Herkunft, Beruf, gesellschaftlicher Status etc.) kaum antizipierbar und erfassbar ist. Es kann auch interpretiert werden, dass dies eine Anpassung an die Individualisierung der Nutzer darstellt.

In dem Freiheitsprinzip ist ein Erfolgspotential zu sehen, weil es neben den genannten Aspekten auch in der rein praktischen Anwendung Vorteile mit sich bringt und den Nutzer bspw. von aufwendigeren kognitiven Prozessen entlastet. Die Psychologin Rashmi Sinha hat Tagging aus kognitionspsychologischer Perspektive analysiert. Abbildung 12 visualisiert den Prozess des Taggings, Abbildung 13 den Prozess der Kategorisierung, also die Einordnung einer Information in ein bestehendes festes Ordnungssystem.

Cognitive Process behind tagging

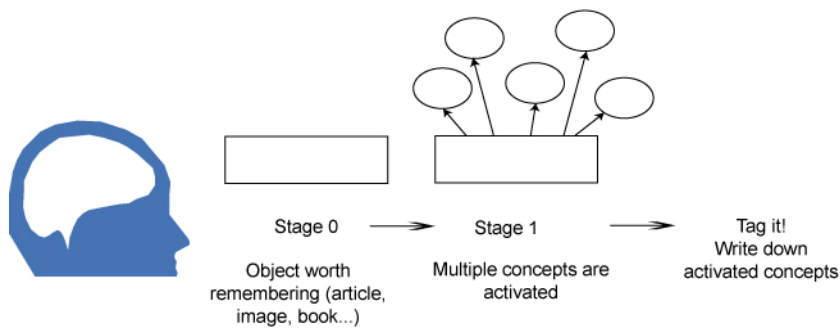


Abb. 12: Kognitiver Prozess beim Tagging (Sinha 2005; eigene Darst.)

Cognitive Process behind categorization

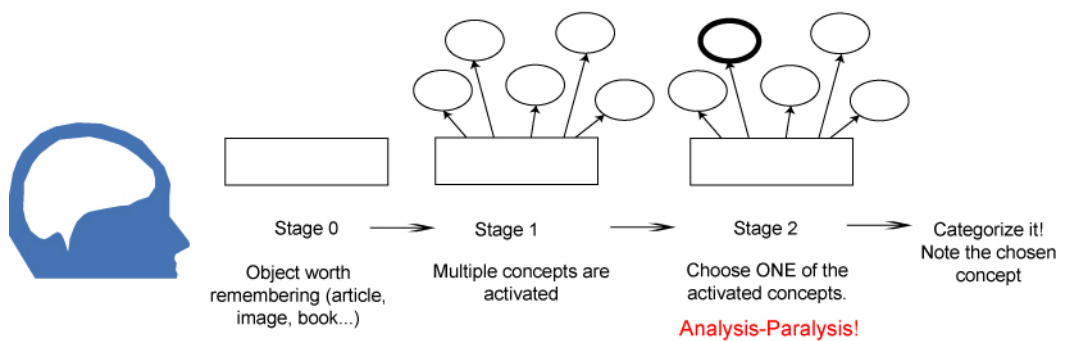


Abb. 13: Kognitiver Prozess beim Kategorisieren (Sinha 2005; eigene Darst.)

Für den Prozess des Kategorisierens beschreibt Sinha den Effekt der „post-activation analysis paralysis“ (Sinha 2005). Zunächst stellt sie fest, dass über die Themen der „digital domain“ (ebd.) wegen ihrer noch kurzen Historie kein allgemeiner kultureller Konsens ausgeprägt ist (vgl. ebd.). Demnach besteht häufig auch kein klarer begrifflicher Konsens. Deutlich wird dies für die Aspekte, die im Zusammenhang mit „Web 2.0“ diskutiert werden. Das heißt, dass die Kategorisierung solcher Informationen einen erhöhten kognitiven Aufwand erfordert als bspw. die Kategorisierung von Alltagsgegenständen. Zudem geht es für den Nutzer grundsätzlich darum, die Information wiederzufinden. Eine nachträgliche Neuordnung ist immer mit Mehraufwand verbunden. Daraus ergibt sich der Effekt der „post-activation analysis paralysis“ (Sinha 2005), den Sinha als eine Befürchtung beschreibt, eine vermeintlich falsche Entscheidung hinsichtlich einer Einordnung bzw. Ablage der Informationseinheit zu treffen und diese so nicht mehr wiederzufinden. Tagging eliminiert diese Entscheidung (vgl. Sinha 2005), da die Freiheit besteht, alle, wie in Abbildung 13 dargestellt, aktivierten Konzepte über Tags zu beschreiben und zu definieren.

Dieser Aspekt könnte auch als Ad hoc-Prinzip des Taggings bezeichnet werden.

Diese Analyse ist grundsätzlich kausal und kann über eine resultierende erhöhte Motivation basierend auf der kognitiven „Entlastung“ letztlich zu einem deutlicheren Anstieg des Informationsumfangs führen als in Systemen, in denen die Einordnung in feste Kategorien erfolgt. Es ist jedoch auch denkbar, dass Nutzer, die bereits über einen gewissen Umfang an Tags verfügen, in einen Prozess des Kategorisierens „geraten“, um bspw. ihre Anzahl an Tags übersichtlich zu halten.

Dieses durch Tagging entstehende simple Organisationsprinzip von Informationen wird auch als Folksonomy (zusammengesetzt aus engl. *folk* und *taxonomy*) bezeichnet (vgl. Mathes 2004; Hammond et al. 2005). Garshol betrachtet eine Folksonomy als eine Taxonomie ohne Hierarchien (vgl. Garshol 2006).

Durch den permanenten, von Nutzern gesteuerten Entwicklungsprozess der Folksonomy können Interessenschwerpunkte (welche Tags), Tendenzen (welche Tags wie häufig) und Interessenschwerpunktveränderungen (Häufigkeit der Tags in zeitlichem Vergleich) abgebildet werden, was als ein Informationspotential für die Nutzer angesehen werden kann.

4.4.2 Abstraktion des Kontextes auf Modellebene

Tagging ist als einfache Indexierungsmethode beschrieben und analysiert worden. In der Betrachtung weiterer populärer Anwendungen, die Tagging implementieren, wie die ebenfalls zu Yahoo! gehörende Online-Fotoverwaltung flickr (<http://www.flickr.com>) oder Weblog-Systeme wie Wordpress, kann festgestellt werden, dass es in derartigen Anwendungen grundsätzlich darum geht, digitale Ressourcen, die entsprechend potentiell über ein Netzwerk zugreifbar sind, mit einem oder mehreren Tags zu verknüpfen. Wird kein Tag definiert, vergibt bspw. das System bei del.icio.us das Universaltag „system:unfiled“, ein Inhalt kann entsprechend unter diesem wiedergefunden werden. Dabei wird die Indexierung und Speicherung der mit Tags versehenen Ressource von einem registrierten Nutzer vorgenommen.

Es lassen sich demnach auf höchster Abstraktionsebene für den Kontext „Tagging“ die drei Konzepte (im Sinne einer gedanklichen Vorstellung; so

auch *concept* als mögliche Übersetzung für „Begriff“ in das Englische und derart im folgenden verwendet), identifizieren:

- Tag bzw. *tag*,
- tagbare Ressource bzw. *taggable resource* sowie
- taggender Nutzer bzw. *tagging user*

In der Modellierung von Entity-Relationship-Modellen sind diese Konzepte als „Entity-Menge“ (Kastens/Kleine Büning 2005, S. 179) oder „Entitätstyp“ (Wiborny 1991, S. 58) bzw. Entity Type zu bezeichnen, da sie eine Zusammenfassung von Objekten repräsentieren (Kastens/Kleine Büning 2005, S. 179).⁵

Im folgenden werden diese als „zentrale Konzepte“ bezeichnet. Tabelle 3 zeigt die Konzepte bzw. Entity Types und mögliche Objekte bzw. Entities, also konkrete Einzelausprägungen.

Zentrales Konzept, Entity Type	Objekt, Entity
Tag	„XML“
tagbare Ressource	Topic Maps XML Syntax
taggender Nutzer	Johannes Schmidt

Tab. 3: Zentrale Konzepte bzw. Entity Types und Entities des Tagging-Kontextes

In einem System wie eines der genannten werden dabei die Objekte bzw. Entities durch einen URL identifiziert. Tabelle 4 zeigt dies exemplarisch für del.icio.us.

Zentrales Konzept, Entity Type	Objekt, Entity	URL
Tag	„XML“	http://del.icio.us/tag/XML
tagbare Ressource	Spezifikation von XTM 2.0	http://del.icio.us/joha06?url=http://isotopicmaps.org/sam/sam-xtm/&jump=no (Zugriff erfordert Login)
taggender Nutzer	joha06	http://del.icio.us/joha06

Tab. 4: Zentrale Konzepte bzw. Entity Types und Entities des Tagging-Kontextes am Beispiel del.icio.us

Für tagbare Ressourcen, die bereits über einen URL per HTTP oder FTP aufrufbar sind (im Bsp. die Spezifikation der Topic Maps XML Syntax unter <http://isotopicmaps.org/sam/sam-xtm/>), muss auf einer systeminternen adres-

⁵ Folglich werden in einem ERM eigentlich Entity Types modelliert. Wiborny bemerkt in diesem Zusammenhang: „Das Entity-Relationship-Modell müsste also genauer 'Entitytype-Relationship-type-Modell' heißen.“ (Wiborny 1991, S. 58).

sierbaren Ressource (<http://del.icio.us/joha06?url=http://isotopicmaps.org/sam/sam-xtm/&jump=no>) lediglich eine Referenz per Hyperlink erstellt werden, so dass durch weitere Informationen wie Beschreibungen quasi eine weiter angereicherte Ressource resultiert. Ressourcen wie Bilddateien, die zunächst ausschließlich in lokalen Dateisystemen vorliegen, müssen erst in Systeme wie flickr über einen Dateiupload transferiert und können dann per URL identifiziert werden: <http://www.flickr.com/photos/49181191@N00/214956473/>. Tagbare Ressourcen wie Beiträge in Blogs werden nach dem Editieren direkt im System gespeichert und per URL referenziert (<http://blog.t8d.de/2006/10/16/tmra-2006-%e2%80%93-phptmapi-und-quaaxtm/>).

Im Ergebnis sind alle Objekte

- a) im Domain-Name-System (DNS), also im Internet, eindeutig identifiziert und
- b) über HTTP unter der Berücksichtigung von Zugriffsrechten im Internet aufrufbar.

Die Bedeutung von URIs wird im Zusammenhang mit Topic Maps in Kapitel 4.2 noch eingehender erläutert.

Für die Beziehungen (*relationships*) der Objekte zueinander gilt dabei:

- Einer tagbaren Ressource werden ein oder mehrere Tags zugeordnet; ein Tag wird einer oder mehreren tagbaren Ressourcen zugewiesen.
- Ein Nutzer kann kein, ein oder mehrere Tags definieren; ein Tag gehört zu einem oder mehreren Nutzern.
- Ein Nutzer kann keine, eine oder mehrere Ressourcen speichern; eine Ressource gehört zu einem oder mehreren Nutzern.

Die Entitätstypen, Entitäts-Attribute und ihre Beziehungen zueinander lassen sich in einem Entity-Relationship-Modell spezifizieren, dessen grafische Notation in Entity-Relationship-Diagrammen (ERD) eine Visualisierung ermöglicht, was als ein großer Vorteil dieser Modellform anzusehen ist (vgl. Schreier 2001, S. 184).

Abbildung 14 zeigt ein rudimentäres ERD für *tag*, *taggable resource* und *tagging user*.

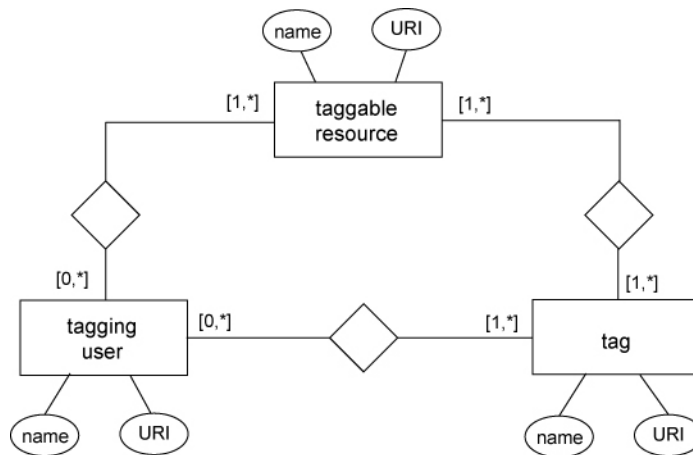


Abb. 14: Rudimentäres ERD des Tagging-Kontextes

Damit wird deutlich, welche zentralen Konzepte in welcher Beziehung im Zusammenhang mit Tagging, wie es hier definiert worden ist, zu berücksichtigen sind.

4.4.3 Analyse der Praxis

Die Bedeutung der Fremddatenintegration und -aggregation im Zusammenhang mit den unter „Web 2.0“ zusammengefassten Entwicklungen ist bereits erläutert und mit RSS in diesem Zusammenhang eine Technologie angesprochen worden. RSS bezeichnet jedoch kein standardisiertes Verfahren, das auch ein einheitliches Format für den Datenexport definiert. RSS bezeichnet zwei unterschiedliche parallele Formate in verschiedenen Versionen (vgl. Pilgrim 2002, S. 1). Unter „Really Simple Syndication“ wird RSS als Spezifikation eines offenen⁶ Formats vom RSS Advisory Board (<http://www.rssboard.org>) verwaltet. Als „RDF Site Summary“ wird ein offenes Format von der RSS-DEV Working Group (<http://web.resource.org/rss/1.0/>) definiert. Das RSS Advisory Board spezifiziert z. Zt. fünf Versionen (vgl. RSS Advisory Board 2006), RDF Site Summary ist einmalig in Version 1.0 spezifiziert (vgl. RSS-DEV Working Group 2001a). Mit Atom, entwickelt von der AtomPub Working Group (<http://www.ietf.org/html.charters/atompub-charter.html>) innerhalb der Internet Engineering Task Force (IETF), existiert zusätzlich ein drittes offenes *syndication format*. Alle drei Formate basieren

⁶ „Offen“ soll hier wie im Zusammenhang mit offenen Standards so verstanden werden, dass die jeweilige Spezifikation allgemein zugänglich und von jedem verwendbar ist (vgl. Picot/Reichwald/Wigand 2003, S. 67)

auf XML 1.0, RDF Site Summary ist zusätzlich zu dem vom W3C spezifizierten Resource Description Framework (RDF) konform (vgl. RSS-DEV Working Group 2001a). RDF Site Summary und Really Simple Syndication weisen in ihrer Elementestruktur (d. h. folglich auch in ihrer Document Type Definition und/oder ihrem Schema) gewisse Ähnlichkeiten auf.

Bei del.icio.us kann der Inhalt eines bestimmten Accounts ausschließlich mit RDF Site Summary exportiert werden. Flickr bietet verschiedene Versionen von Really Simple Syndication sowie RDF Site Summary und Atom an. Die folgenden XML-beispiele zeigen, wie Einzelausprägungen bzw. Objekte der drei erarbeiteten Konzepte **tag**, **taggable resource** und **tagging user** bei der Fotoverwaltung flickr in den Formaten RDF Site Summary und Really Simple Syndication jeweils als nutzerspezifischer Feed strukturiert werden.

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:admin="http://webns.net/mvcb/"
  xmlns="http://purl.org/rss/1.0/">
  <channel rdf:about="http://www.flickr.com/photos/49181191@N00/">
    <title>mmhelstar's Photos</title>
    <link>http://www.flickr.com/photos/49181191@N00/</link>
    <description>A feed of mmhelstar's Photos</description>
    <dc:date>2006-08-14T04:35:18-08:00</dc:date>
    <admin:generatorAgent rdf:resource="http://www.flickr.com/" />
    <image
rdf:resource="http://www.flickr.com/images/buddyicon.jpg?49181191@N00" />
    <items>
      <rdf:Seq>
        <rdf:li
rdf:resource="http://www.flickr.com/photos/49181191@N00/214956473/" />
        </rdf:Seq>
      </items>
    </channel>
    <item rdf:about="http://www.flickr.com/photos/49181191@N00/214956473/">
      <title>list_image</title>
      <link>http://www.flickr.com/photos/49181191@N00/214956473/</link>
      <description>[...]</description>
      <dc:date>2006-08-14T04:35:18-08:00</dc:date>
      <dc:date.Taken>2006-08-14T04:35:18-08:00</dc:date.Taken>
      <dc:creator>mmhelstar
(http://www.flickr.com/people/49181191@N00/)</dc:creator>
    </item>
  </rdf:RDF>
```

RSS-Listing 1: RSS Feed bei flickr mit RDF Site Summary

```

<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0"
  xmlns:media="http://search.yahoo.com/mrss"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <channel>
    <title>mmhelstar's Photos</title>
    <link>http://www.flickr.com/photos/49181191@N00/</link>
    <description>A feed of mmhelstar's Photos</description>
    <pubDate>Mon, 14 Aug 2006 04:35:18 -0800</pubDate>
    <lastBuildDate>Mon, 14 Aug 2006 04:35:18 -0800</lastBuildDate>
    <generator>http://www.flickr.com/</generator>
    <image>

    <url>http://www.flickr.com/images/buddyicon.jpg?49181191@N00</url>
    <title>mmhelstar's Photos</title>
    <link>http://www.flickr.com/photos/49181191@N00/</link>
  </image>
  <item>
    <title>list_image</title>

    <link>http://www.flickr.com/photos/49181191@N00/214956473/</link>
    <description>[...]</description>
    <pubDate>Mon, 14 Aug 2006 04:35:18 -0800</pubDate>
    <dc:date.Taken>2006-08-14T04:35:18-08:00</dc:date.Taken>
    <author>nobody@flickr.com (mmhelstar)</author>
    <guid
isPermaLink="false">tag:flickr.com,2004:/photo/214956473</guid>
    <media:content
url="http://static.flickr.com/94/214956473_2aa92fe2f7_o.jpg"
      type="image/jpeg"
      height="10"
      width="10"/>
    <media:title>list_image</media:title>
    <media:text type="html">
    </media:text>
    <media:thumbnail
url="http://static.flickr.com/94/214956473_2aa92fe2f7_s.jpg" height="75"
width="75" />
    <media:credit
role="photographer">mmhelstar</media:credit>
    <media:category scheme="urn:flickr:tags">test
<!--></media:category>
  </item>
</channel>
</rss>

```

RSS-Listing 2: RSS Feed bei flickr mit Really Simple Syndication 2.0

Die Beispiele zeigen eine unterschiedliche Strukturierung der jeweiligen Objekte und einen generellen Unterschied im Datenumfang. Das Fehlen der Tags im RDF Site Summary Feed stellt jedoch einen eklatanten Informationsverlust in Verbindung mit der externen Weiterverarbeitung der Daten dar. Bei Really Simple Syndication 2.0 werden die Benennungen der Tags über

```
<media:category scheme="urn:flickr:tags">test fun</media:category>
```

mit ausgegeben, jedoch fehlt deren URL zur eindeutigen Identifizierung für das System, in dem sie definiert sind, und zum Zugriff im Internet. „media:“ kennzeichnet dabei die namensraumbasierte Erweiterung für das Media RSS Modul für Really Simple Syndication 2.0 von Yahoo! (<http://search.yahoo.com/mrss>), das also dienstspezifisch ist.

Die inhaltliche Inkonsistenz zwischen den Formaten bleibt unverständlich.

Ein Vergleich mit einem RSS Feed der Linkverwaltung del.icio.us, der, wie bereits erwähnt, ausschließlich über RDF Site Summary erfolgt, zeigt bzgl. der zentralen Konzepte bzw. der jeweiligen Einzelausprägung einen inhaltlichen Unterschied und letztlich eine Differenz in der Betrachtung hinsichtlich der Bedeutung des Konzeptes „Tag“ in den jeweiligen Systemen.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://purl.org/rss/1.0/"
  xmlns:cc="http://web.resource.org/cc/"
  xmlns:taxo="http://purl.org/rss/1.0/modules/taxonomy/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:syn="http://purl.org/rss/1.0/modules/syndication/"
  xmlns:admin="http://webns.net/mvcb/"
  <channel rdf:about="http://del.icio.us/joha06">
    <title>del.icio.us/joha06</title>
    <link>http://del.icio.us/joha06</link>
    <description></description>
    <items>
      <rdf:Seq>
        <rdf:li rdf:resource="http://www.topicmaps.org/xtm/1.0/xtml.dtd"
        />
        [...]
      </rdf:Seq>
    </items>
  </channel>
  <item rdf:about="http://www.topicmaps.org/xtm/1.0/xtml.dtd">
    <title>DTD XTM 1.0</title>
    <link>http://www.topicmaps.org/xtm/1.0/xtml.dtd</link>
    <description>This is the DTD of XTM 1.0.</description>
```

```

<dc:creator>joha06</dc:creator>
<dc:date>2006-08-22T08:20:51Z</dc:date>
<dc:subject>XTM</dc:subject>
<taxo:topics>
  <rdf:Bag>
    <rdf:li resource="http://del.icio.us/tag/XTM" />
  </rdf:Bag>
</taxo:topics>
</item>
[...]
</rdf:RDF>

```

RSS-Listing 3: RSS Feed bei del.icio.us mit RDF Site Summary

Strukturelle Übereinstimmung besteht bei den Einzelausprägungen von *taggable resource* und *tagging user*. Während bei flickr Tags nicht ausgegeben werden, zeigt der Export bei del.icio.us bezüglich der Tags einen zweidimensionalen Aufbau. RDF Site Summary ist u. a. für das Dublin Core (DC) Metadaten-set über eine namensraumbasierte Modularisierung (vgl. RSS-DEV Working Group 2001a) erweitert. Die Namensraumerweiterung für Dublin Core ist über das Attribut `xmlns` definiert (`xmlns:dc=http://purl.org/dc/elements/1.1/`). So können Dublin Core-Elemente im Dokument über `dc:` verwendet werden. Das Tag „XTM“ bzw. seine Benennung wird hier nun zum einen über das Dublin Core `subject`-Element eingebunden:

```
<dc:subject>XTM</dc:subject>
```

Das Taxonomy-Modul, ebenfalls eine RDF-Anwendung (vgl. RSS-DEV Working Group 2001b), erlaubt ebenso die Identifizierung der Themen in einem Feed. Es wird ebenfalls über Namensraumerweiterung eingebunden. Hier wird zum anderen der identifizierende URL des Tags „XTM“ für das System del.icio.us über

```

<taxo:topics>
  <rdf:Bag>
    <rdf:li resource="http://del.icio.us/tag/XTM" />
  </rdf:Bag>
</taxo:topics>

```

als Wert des Attributes „resource“ des `rdf:li`-Elements eingebunden.

So kann konstatiert werden, dass bei del.icio.us mit dem Bezug zu RDF und Dublin Core *tag* explizit als Metadateneinheit für die Verwendung in einem globalen Kontext, dem Internet, angesehen wird. Zudem ist hier die Zuweisung Objekt: URL gegeben.

In der grundsätzlichen Betrachtung der unterschiedlichen Formate zeigt sich, dass die Mehrfachverwendung von Inhalten durch Fremddatenintegration und -aggregation mit RSS und RSS vergleichbaren Verfahren nicht auf einem einzigen Format basiert. So müssen Anbieter mehrere Formate bereitstellen (flickr), wenn sie nicht darauf verzichten (del.icio.us), während Feeds verarbeitende Software unterschiedliche Formate beherrschen muss, was letztlich auf beiden Seiten zu erhöhtem Aufwand führt. Da die Formate wenig komplex sind, ist dieser Aufwand überschaubar, so dass FeedReader i. a. alle Formate verarbeiten können und unter dem Aspekt von Netzeffekten die Nutzergruppe von RSS (inklusive Atom) als ein Netz bzw. Netzwerk definiert werden kann.

Ein zweiter Aspekt bezieht sich auf den erwähnten Unterschied in Struktur und Inhalt bzgl. der erarbeiteten Konzepte, die auf eine ebenfalls erwähnte unterschiedliche Sichtweise hinsichtlich ihrer jeweiligen Bedeutung innerhalb des Gesamtzusammenhanges schließen lässt. Daraus ist letztlich zu schlussfolgern, dass auf (Daten)Empfängerseite nicht auf den originären inhaltlichen bzw. Bedeutungszusammenhang, den das konzeptionelle Modell (siehe Abb. 14) repräsentiert, zurückgegriffen werden kann. Dabei geht es um Informationen aus Anfragen wie „Welche Ressourcen zu welchem Thema werden von wem verwaltet?“. Dieser Zusammenhang, der weitere Informationen, die sich auf Basis der (quasi syntaktischen) Kombination der drei zentralen Konzepte ableiten lassen, umschließt, soll im folgenden in der Funktion eines Leitaspektes als „Semantik des Taggings“ verstanden werden.

Ein abschließender Blick auf die spezifizierten Schnittstellen, die neben dem externen Zugriff auf Daten auch als entscheidende Erweiterung deren Manipulation zulassen, zeigen, dass die Systeme für ein zweites Prinzip der Fremddatenintegration neben RSS und RSS-vergleichbare hier jeweils unterschiedliche Datenstrukturen definieren.

Flickr bietet mit Webservices über SOAP und XML-RPC als dessen Vorgänger (vgl. Geroimenko 2004, S. 205) eine Zugriffsmöglichkeit, während del.icio.us ausschließlich einen Zugriff über Representational State Transfer (REST) bereitstellt, das als Architekturkonzept einen Zugriff auf Ressourcen z. B. über einfache HTTP GET- oder POST-Requests erlaubt. Eine Antwort, d. h. die Repräsentation des Zugriffs erfolgt i. d. R., wie auch in den folgenden Beispielen, in XML. Die Anfrage per HTTP GET bei flickr (sensible Daten unkenntlich gemacht) lautet

```
http://api.flickr.com/services/rest/?method=flickr.photos.getInfo&api_key=af36XXXX&photo_id=214956473&secret=07e8XXXX&auth_token=1084XXXX&api_sig=7031XXXX
```

und liefert über das Quasi-Objekt „photos“ (repräsentiert *taggable resource*) und dessen Methode „getInfo“ den Inhalt einer einem Nutzer zugeordneten Ressource:

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
<photo id="214956473" secret="2aa9XXXX" server="94" dateuploaded="115555318"
isfavorite="0" license="0" rotation="0" originalformat="png">
  <owner nsid="49181191@N00" username="mmhelstar" realname="" location=""
/>
  <title>list_image</title>
  <description />
  <visibility ispublic="1" isfriend="0" isfamily="0" />
  <dates posted="115555318" taken="2006-08-14 04:35:18"
takengranularity="0" lastupdate="1156235658" />
  <permissions permcomment="3" permaddmeta="2" />
  <editability cancomment="1" canaddmeta="1" />
  <comments>0</comments>
  <notes />
  <tags>
    <tag id="3896075-214956473-320" author="49181191@N00"
raw="Test">test</tag>
    <tag id="3896075-214956473-1743" author="49181191@N00"
raw="Fun">Fun</tag>
  </tags>
  <urls>
    <url
type="photopage">http://www.flickr.com/photos/49181191@N00/214956473/</url>
  </urls>
</photo>
</rsp>
```

REST-Response-Listing 1: Repräsentation eines Zugriffs über REST über das API von flickr

Bei del.icio.us erfolgt der Zugriff über HTTPS und erfordert entsprechend eine Authentifizierung, hier mit Benutzernamen und Kennwort eines Benutzerkontos, die damit den Nutzer zugleich im System identifizieren. Die GET-Anfrage bei del.icio.us (sensible Daten ebenfalls unkenntlich gemacht) lautet

```
https://joha06:XXXX@api.del.icio.us/v1/posts/recent?count=1
```

und liefert die zuletzt angelegte Ressource bzw. deren URL des entsprechenden Nutzers:

```

<?xml version='1.0' standalone='yes'?>
<posts tag="" user="joha06">
  <post href="http://www.topicmaps.org/xTm/1.0/xTm1.dtd" description="DTD
XTM 1.0" extended="This is the DTD of XTM 1.0."
hash="22a0200e3204cf649b30d8ab14a8db4e" tag="XTM" time="2006-08-22T08:20:51Z"
/>
</posts>

```

REST-Response-Listing 2: Repräsentation eines Zugriffs über REST über das API von del.icio.us

Der Vergleich der beiden Beispiele zeigt einen erheblichen Unterschied bzgl. Struktur und Inhalt. Hinsichtlich der Einbindung der konkreten Ausprägungen der zentralen Konzepte zeigen beide Systeme Schwächen, da sie nicht die vollständige Einbindung aller Objekte und ihrer identifizierenden URLs realisieren.

4.4.4 Konklusion

Die Erarbeitungen der letzten Kapitel haben gezeigt, dass auf Ebene des Datenaustausches zur Datenintegration und Datenaggregation über RSS und vergleichbare Verfahren eine Anzahl paralleler konkurrierender Formate existiert, die auf Sender- und Empfängerseite auf technischer bzw. Softwareebene jeweils zu Überlegungen über die Verarbeitung der verschiedenen Formate zwingt und letztendlich zu erhöhten Aufwänden in der Verarbeitung mit Software führt. Für den Aspekt der Schnittstellen über Webservices oder REST, die neben dem bloßen Zugriff häufig auch die Manipulation von Daten erlauben, ist für die populären Systeme der Link- und Fotoverwaltungen, z. B. del.icio.us und flickr, sowie für Webloganbieter, z. B. Blogger (<http://www.blogger.com>), festzustellen, dass ihre Schnittstellen jeweils individuell definiert sind. Daraus resultiert, dass die Integration von Inhalten unterschiedlicher Anwendungen bzw. Systeme jeweils eine spezifische/individuelle Ansprache der Schnittstellen erfordert und somit auch hier zu höheren Aufwänden in der Entwicklung und Implementierung Inhalte integrierender und aggregierender Anwendungen führt. Die entscheidende Schwäche ist darin zu sehen, dass die Daten und Metadaten an der Quelle nicht standardisiert strukturiert und ebenso keine standardisierten Mechanismen für die Etablierung verlässlicher Semantik implementiert sind: „Was ist ein Tag?“, „Wie ist die Relation zwischen „joha06“ und „XML“ zu interpretieren?“ (Möglicherweise auch eine Relation zwischen „**g**“ und „\$&%“).

Unterschiedliche Formate können aus dieser Perspektive quasi als zwangsläufige Folgefehler interpretiert werden. Diese Schwäche potenziert sich weiter, wenn Inhalte verschiedener Dienste zu integrieren sind und führt letztlich zu der Frage, wie Inhalte zu einem bestimmten Thema/Begriff aus unterschiedlichen Anwendungen semantisch verlässlich zusammengeführt werden können.

Dabei hat die Abstraktion des Tagging-Kontextes in Kapitel 4.4.2 gezeigt, dass den beschriebenen Anwendungen ein gleichartiges konzeptionelles Modell zu Grunde liegt, das die Entitätstypen *tag*, *taggable resource* und *tagging user* und ihre Beziehungen zueinander strukturiert. Dieses Modell soll als konzeptionelles Tagging-Modell (*conceptual tagging-model*) bezeichnet werden, welches die skizzierte Tagging-Semantik reflektiert. Dieses Modell ist simpel, genügt jedoch dennoch als exemplarischer Bearbeitungsgegenstand dieser Arbeit. Die in diesem Modell verwalteten Daten pro Objekt sind in ihrer Größe gering, prinzipiell geht es um den Austausch von URIs bzw. URLs und Bezeichnen von Ressourcen, um auf diese auf der integrierenden Zielressource verweisen zu können. Entsprechend muss jeder Instanz eines Informationsobjektes ein URL zugewiesen sein, im folgenden in REST-konformer Definition:

Testuser: <http://example.com/users/testuser>

Testtag: <http://example.com/users/testuser/testtag>

Testphoto:

<http://example.com/users/testuser/photos/76t5ge87k38hft4@testphoto>

und für eine Anzeige auf der Zielressource der URL einer spezifischen Resource exemplarisch:

Testphoto:

<http://photos.example.com/photos/76t5ge87k38hft4@testphoto.jpg>.

Wie bereits erwähnt, stellen Tags Metadaten einer Ressource dar. Insofern ist das Konzept *tag* streng genommen mit *taggable resource* und *tagging user* funktional nicht gleich zu ordnen, da es eine Metaebene beschreibt. Für die Erarbeitung von bspw. Relationalen Datenmodellen zum Aufbau einer Datenbank, in der Daten einer konkreten Anwendung wie del.icio.us gespeichert werden, ist diese Unterscheidung nicht relevant, da nur der Gesamtzusammenhang, der in einem Modell gleichberechtigter Entity Types und ihrer Beziehungen zueinander dargestellt wird, die Funktionsweise der Anwendung abbildet. So könnte formuliert werden, dass in der Datenhaltung eines „Produkti-

onssystemen“ *tag* also *taggable resource* und *tagging user* gleich geordnet wird. Für *content syndication* ist neben der Semantik des Taggings die eindeutige Identifizierung von einzelnen Ausprägungen der erarbeiteten Entity Types über URIs bzw. URLs, die zudem einen Zugriff über das Internet erlauben, von Bedeutung, also ein bestimmtes Attribut dieses Objektes von entscheidender Relevanz.

Tag ist das tragende semantische Element in dem Zusammenhang, der als „Semantik des Taggings“ bezeichnet worden ist. Die Intention im Retrieval ist primär die, Ressourcen zu einem Thema/Begriff zu finden. Das Retrieval wird, wie erwähnt, durch intellektuelle Indexierung optimiert. Für den externen Zugriff auf Datenstrukturen, die die Semantik des Taggings reflektieren, ist die Existenz eines parallelen, für diesen Zugriff optimierten standardisierten Modells berechtigt, in dem nun quasi *taggable resource* und *tagging user* dem Konzept *tag* funktional gleich geordnet sind und so ebenfalls eine Metaebene beschreiben.

Hinsichtlich einer verlässlichen Semantik von Tags besteht Handlungsbedarf, da diese lediglich Benennungen/Zeichenketten in einem geschlossenen System sind. Sigel, Bünten und Frank bemerken hierzu im Zusammenhang mit Weblogs treffend:

Tags [are] not semantically grounded (only names/strings). Tags [are] are not interrelated by typed semantic relations (Sigel/Bünten/Frank 2006, S. 5)

Das konzeptionelle Tagging-Modell, das in dieser Arbeit diskutiert wird, soll als ein Modell definiert werden, das parallel zu einem (Daten)Modell zu definieren und implementieren ist, das für die Planung und Umsetzung von Datenbanken, die Massendatenhaltung und hohe Zugriffszahlen und -frequenzen gewährleisten müssen, entworfen ist und auf Grund der daraus resultierenden Möglichkeit ökonomischer Aktivität i. a. auch zuerst implementiert wird. Im Resultat liegt somit aus der Perspektive dieser Speicherstruktur eine diese repräsentierende Parallelstruktur, eine Meta(daten)ebene, und in der Gesamtbeachtung eine duale Datenspeicherstruktur vor.

Ihnen liegt jeweils ein gleichartiges Modell zu Grunde (konzeptionelles Tagging-Modell), sie können jedoch prinzipiell unterschiedlich implementiert werden, bspw. einerseits auf Basis eines Relationalen Modells und andererseits auf Basis des Topic Maps Data Model, das noch vorgestellt wird. Abbil-

dung 15 verdeutlicht zusammenfassend die Erkenntnisse als IST-SOLL-Vergleich.

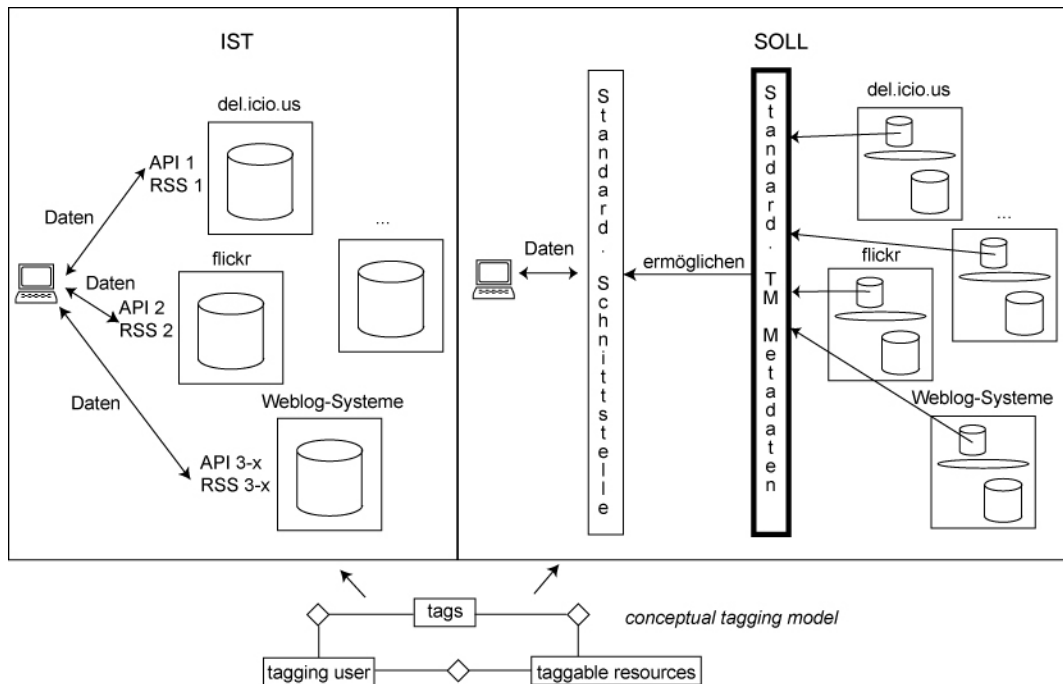


Abb. 15: IST-SOLL-Vergleich für Systeme, die den Zugriff auf die durch die zentralen Konzepte *tag*, *tagging user* und *taggable resource* beschriebene Semantik des Taggings erlauben

Der Datenzugriff über eine Schnittstelle z. B. via Webservice oder REST kann sich auf ein solches Parallelmodell beschränken, da in diesem bereits alle relevanten Daten für *content syndication* vorgehalten werden, das vor allem jedoch eine einheitliche Struktur und die Möglichkeit zur Etablierung von Semantik gewährleisten soll.

Das simple konzeptionelle Tagging-Modell ist die Grundlage für die Definition eines konkreten Tagging-Modells, das implementiert werden kann. Auf dieser Basis können standardisierte Formate und Zugriffsmechanismen auf die Objekte und ihre Beziehungen zueinander (Semantik des Taggings) angewendet werden.

In dieser Arbeit soll im praktischen Teil gemäß der Zielstellung ein konkretes Tagging-Modell, das die Meta(daten)struktur abbildet, mit dem ISO/IEC-Standard 13250 Topic Maps entwickelt werden. Für den Modellbegriff, der in verschiedenen Zusammenhängen mit recht unterschiedlichen Bedeutungen verwendet wird (vgl. Kastens/Kleine Büning 2005, S. 16), gilt für das Topic Maps-Tagging-Modell im wesentlichen dessen Eigenschaften zur vereinfachten Dar-

stellung (des Ablaufs) eines Sachverhalts und zur Verwendung als Muster bzw. Vorbild (vgl. DUDEN 1990, S. 507).

Nachfolgend werden die Technologie sowie konstitutive Voraussetzungen für die Implementierung und Anwendung des Modells ausführlicher eingeführt und erklärend dargestellt.

5 Topic Maps

Vor den folgenden Ausführungen sei bemerkt, dass die im Zusammenhang mit Topic Maps verwendete Terminologie der in den unter ISO/IEC 13250 zusammengefassten Standards verwendeten entspricht und die englischen Begriffsbenennungen in der im Deutschen für Substantive üblichen Großschreibung übernommen werden. So kann jederzeit eine weiterführende oder prüfende Lektüre in den Standardspezifikationen erfolgen. Anfang 2006 erarbeitete Maicher in öffentlicher Diskussion über die deutschsprachige Topic Maps-E-Mailliste (<http://www.topicgarden.com/maillinglist/>) unter Beteiligung des Autors zwar eine deutschsprachige Topic Maps-Terminologie (<http://www.informatik.uni-leipzig.de/~maicher/tmt/TMT.html>), diese stellt jedoch keinen offiziellen Standard dar (vgl. Maicher 2006a). Neben dem angesprochenen Aspekt der Lektüre ist die Sachlage die, dass neben den englischsprachigen Standardspezifikationen die weiteste und maßgebliche Diskussion über Topic Maps in englischer Sprache stattfindet und deshalb die deutschsprachige Terminologie hier nur ergänzend und zur Unterstützung des Verständnisses verwendet wird.

Topic Maps bezeichnet eine Technologie zur Informationsstrukturierung, zur Repräsentation von Wissen (vgl. Obrst/Liu 2003, S. 126) und zum Austausch von Wissensstrukturen. Topic Maps liegt ein abstraktes Metamodell zu Grunde. In einer Topic Map⁷ werden Aussagegegenstände und Informationen über Aussagegegenstände über diese repräsentierende Topics strukturiert. Relevante Informationsquellen können in solch einer Struktur, einer Topic Map, referenziert werden, wodurch eine Verknüpfung zwischen Metastruktur und Resource entsteht. Diese Trennung und Verknüpfungsmöglichkeit kennzeichnen die Grundidee und das Potential von Topic Maps.

Topic Maps entspricht in der grundlegenden Intention der des Resource Description Framework (RDF), der Initiative des World Wide Web Consortiums (W3C) im Bereich des Semantic Web.

Very similar claims are made for both RDF/RDFS [...] and topic maps. Both are promoted as methods for associating arbitrary metadata and arbitrary content. (Freese 2003, S. 284)

⁷ Also bezeichnet Topic Maps einerseits die umfassende Technologie, die unter ISO/IEC 13250 spezifiziert ist, sowie andererseits den Plural von Topic Map als eine Menge von Topics und Associations.

Eine eigenständige Arbeitsgruppe unter dem Dach des W3C, die RDF/Topic Maps Interoperability Task Force (RDFTM), erarbeitet Richtlinien für die kombinierte Anwendung der beiden Technologien (vgl. RDFTM 2006).

Mit Topic Maps lassen sich Ontologien formulieren. Eine Ontologie im Kontext der Wissensrepräsentation bezeichnet nach der Kurzdefinition von Gruber eine explizite Spezifizierung einer Konzeptualisierung (vgl. Gruber 1993, S. 1) bzw. eine Formalisierung einer Konzeptualisierung (vgl. Obrst/Liu 2003, S. 124), wobei eine Konzeptualisierung eine Menge an Konzepten und ihre Beziehungen zueinander darstellt (vgl. Obrst/Liu 2003, S. 124) als eine abstrahierte, vereinfachte Sicht auf irgendeine Domäne. Eine Ontologie bildet Wissensstrukturen ab und vermeidet durch die Formalisierung Mehrdeutigkeiten (vgl. Studer/Oppermann/Schnurr 2001, S. 4). Das Wissen bezieht sich auf einen spezifischen Wissensbereich, der Domäne bzw. „domain of discourse“ (Gruber 1993, S. 1).

Eine Ontologie umfasst

- *Entities* (things)
- The *relationship* between those entities
- The *properties* (and property values) of those entities,
- The *functions* and *processes* involving those entities
- *Constraints* on and *rules* about those entities (Obrst/Liu 2003, S. 125)

Obrst und Liu bemerken zu der Beziehung zwischen Ontologien und Topic Maps, dass die Spezifikation von Topic Maps eine syntaktische Interoperabilität gewährleistet, während Ontologien eine semantische Interoperabilität ermöglichen. Wenn eine Topic Map aus einer konsistenten Ontologie („a sound ontology“ [Obrst/Liu 2003, S. 125f.]) erstellt wird, können diese eine semantische Interoperabilität nicht nur untereinander, sondern auch über auf sie aufsetzende Anwendungen hinweg realisieren (vgl. Obrst/Liu 2003, S. 125f.).

Die Anfänge von Topic Maps gehen auf das Jahr 1991 und die Gründung der Davenport Gruppe zurück, die sich u. a. aus Vertretern verschiedener UNIX-Anbieter und des O'Reilly-Verlages zusammensetzte, um das Problem der inkonsistenten Terminologie zu gleichen Aspekten in den Softwaredokumentationen zu diskutieren. Ziel war die Integration der von O'Reilly entwickelten Dokumentation in die Dokumentationen der Hersteller und letztlich die Entwicklung integrierender Masterindizes (vgl. Newcomb 2003, S. 37). Für die Realisierung sollte ISO/IEC 10744 HyTime, eine Erweiterung der Structured General Markup Language (SGML), verwendet werden. Der erste Ansatz na-

mens SOFABED (Standard Open Formal Architecture for Browsable Electronic Documents) ist unter der Bezeichnung „Topic Maps“ weiter ausgearbeitet und 1995 von der ISO als Arbeitsgegenstand akzeptiert worden (vgl. Newcomb 2003, S. 38). Im Januar 2000 ist Topic Maps als Standard ISO/IEC 13250 mit der Syntax HyTM veröffentlicht worden. (Ausführliche Informationen zur Entstehung und Entwicklung von Topic Maps finden sich bei Newcomb 2003 und Hunting 2003.)

5.1 Das „TAO“: Topic, Association, Occurrence

Die Basis des Topic Maps-Paradigmas bilden die Konstrukte „Topic“, „Association“ und „Occurrence“, auch als das TAO von Topic Maps bezeichnet (vgl. Pepper 2000).

Ein Topic repräsentiert ein *subject* innerhalb einer Topic Map. Diese Repräsentation ermöglicht den Diskurs über einen Aussagegegenstand in einer Topic Map. Dabei kann ein Aussagegegenstand einen Begriff, eine Idee, ein Individuum etc. darstellen:

A subject can be anything whatsoever, regardless of whether it exists or has any other specific characteristics, about which anything whatsoever may be asserted by any means whatsoever. In particular, it is anything about which the creator of a topic map chooses to discourse. (ISO/IEC 13250-2)

Der Topic Maps-Standard gibt also nicht vor, welche Aussagegegenstände für die Verarbeitung in Topic Maps zulässig sind. Also kann für ein Beispiel der Autor dieser Arbeit von einem Topic repräsentiert werden. Topics können zu Topic Types generalisiert werden. Das den Autoren repräsentierende Topic kann mit einem Topic verknüpft werden, das die Studierenden der HAW Hamburg repräsentiert. Dabei sind Topic Types wiederum nur Topics, die Funktion/Rolle als Type wird durch die spezifische Relation definiert. Diese Beziehung kann als typische Klasse-Instanz-Relation angesehen werden (vgl. Pepper 2000). Eine Instanz ist im Hinblick auf die bis dato verwendete Terminologie der Modellbildung auf dieser Abstraktionsebene als einem Objekt/Entity/Einzelausprägung entsprechend anzusehen. Topics können beliebig viele Topic Names zugewiesen werden. Die Benennung eines Topics ist aber nicht vorgeschrieben, also fakultativ. Alternativbezeichnungen einer Benennung können über das Variant-Konstrukt definiert werden.

So können zwei Topics mit Benennungen [Johannes Schmidt] und [Studierende] definiert werden, die in einer Klasse-Instanz-Relation zueinander stehen, wie Abbildung 16 illustriert.

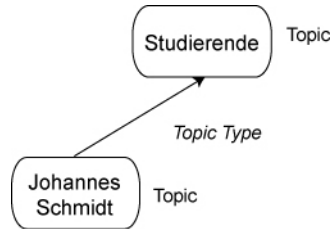


Abb. 16: Topic und Topic Type

Occurrences verknüpfen relevante Ressourcen mit Topics. Eine Occurrence referenziert entweder eine externe Ressource wie z. B. eine Webseite über Adressierungsmechanismen wie URI oder eine Occurrence stellt selbst eine (interne) Ressource dar. I. d. R. werden externe Ressourcen referenziert, wodurch sich die bereits genannte Trennung zwischen Metaebene und die durch Occurrences verknüpften Informationsressourcen ergibt. Occurrences können ebenfalls durch Topics typisiert werden, um diese zu spezifizieren. Exemplarisch kann ein Topic [Webseite] definiert werden und einer Occurrence als Type zugewiesen werden. Abbildung 17 erweitert Abbildung 16 um das Prinzip der Occurrences.

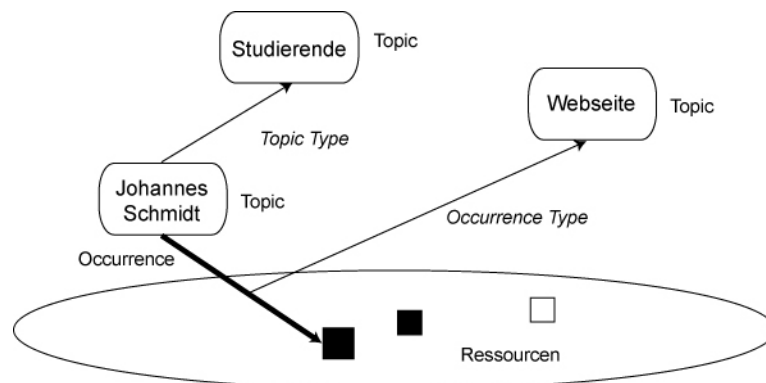


Abb. 17: Topic und Topic Type, Occurrence und Occurrence Type

Die bis dato erläuterten Konstrukte erlauben eine grundlegende hierarchische Organisation von Informationen, bspw. in der Form einfacher Indexe (vgl. Pepper 2000). Associations erlauben die freie Definition von Beziehungen zwischen zwei oder mehreren Topics. Associations stellen das wesentliche Kon-

strukt zur Modellierung von Wissensstrukturen dar, indem das menschliche Denkprinzip simuliert wird (vgl. Rath 2003, S. 14).

So könnten folgende Associations definiert werden:

[Diplomarbeit] - [Johannes Schmidt]

[Studierende] - [Hochschule]

[Hamburg] - [Deutschland]

Associations können ebenso mit Topics verknüpft werden, die als Association Types eine Beziehung klassifizieren:

[Diplomarbeit] - [verfasst von] - [Johannes Schmidt]

[Studierende] - [studieren an] - [Hochschule]

[Altona] - [Teil von] - [Hamburg]

Es sei bemerkt, dass eine Klasse-Instanz-Relation zwischen Topics nicht allein über das Prinzip der Types ausgedrückt werden kann, sondern ebenso über eine Association, der dann z. B. ein Topic [Instanz von] als Association Type zugewiesen wird. Entsprechend sind Types spezielle, in Topic Maps bereits (vor)definierte Beziehungskonstrukte.

Assoziationen, die das Topic Maps-Paradigma definiert, implizieren keine Richtung der Beziehung zwischen Topics (vgl. Rath 2003, S. 14). Rath bemerkt:

Associations are assertions, statements, which are valid independently the direction you traverse them [...]. You can look at the association from all involved topics – it always expresses the same assertion. (Rath 2003, S. 14)

Dennoch ist nicht klar, welche Rollen die Topics in den obigen Beispielen einnehmen, also ob Altona ein Teil von Hamburg oder Hamburg ein Teil von Altona ist.

Association Roles stellen die Teilnahme aller Topics in einer Association dar. Dabei charakterisiert Association Role Type, ein Topic, die Beziehungsrolle. Association Role Player sind die Topics, die in einer Beziehungsrolle einen bestimmten Beziehungstyp, Association Role Type, repräsentieren.

So können im Beispiel die Association Role Types [Stadtteil] und [Stadt] definiert werden, um die Beziehung zwischen den Topics [Altona] und [Hamburg] zu spezifizieren. In Abbildung 18 wird das Beispiel schematisch in Anlehnung an das auf Graphen basierende Aussagenmodell aus einem Entwurf zum Topic Maps Reference Model, Revision 1.0, dargestellt (vgl. ISO/IEC N0344).

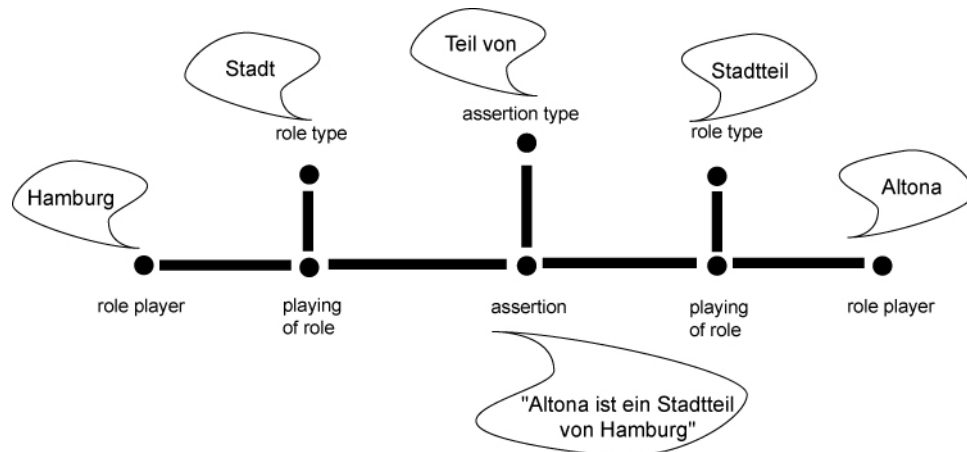


Abb. 18: Visualisierung einer Association unter Verwendung eines Aussagenmodells im Draft Topic Maps Reference Model, Revision 1.0

Grundsätzlich können auch Beziehungen modelliert werden, die reflexiv, symmetrisch oder transitiv sind (vgl. Rath 2000, S. 9). Transitive Relationen z. B. erlauben Ableitungen, die nicht explizit festgelegt worden sind. Aus dem Beispiel

[Pennsylvania] - [liegt in] - [USA]

[Philadelphia] - [liegt in] - [Pennsylvania]

kann eine Topic Map verarbeitende Software schließen:

[Philadelphia] - [liegt in] - [USA] (vgl. Rath 2000, S. 10).

Die Möglichkeit solcher Inferenzen im Zusammenwirken von Topic Maps und Software stellt ein erhebliches Potential dar.

5.2 Identität von Aussagegegenständen

Topics repräsentieren Aussagegegenstände. Die Frage, was ein Aussagegegenstand ist, stellt eine zentrale Frage des Wissensaustausches dar (vgl. Vatant 2003, S. 67). Entsprechend ist dies auch ein zentraler Aspekt in der Konzeption und Erstellung von Topic Maps. Neben einer primär philosophischen Fragestellung nach dem Grundsätzlichen ist aus der praktischen Perspektive zu klären, wer oder was gemeint ist, also darum, die Identität eines Aussagegegenstandes möglichst eindeutig zu bestimmen, um auf die Semantik bzw. die Bedeutung und letztlich auf einen Begriff oder ein menschliches Individuum schließen zu können. Die Identifizierung über

Benennungen kann die Anforderung nach Eindeutigkeit nicht gewährleisten: Ein Topic mit der Benennung „Johannes Schmidt“ kann viele Subjekte der realen Welt repräsentieren. Ebenso können gleiche Benennungen z. B. in verschiedenen sprachlichen Kontexten auf unterschiedliche Begriffe verweisen, Beispiel „chat“ im Englischen (Unterhaltung) und im Französischen (Katze, Kater). In diesem Zusammenhang ist das Homonymproblem als Gegensatz zum Synonymproblem zu erwähnen. Der Autor einer Topic Map möchte oder muss jedoch Aussagen zu einem ganz bestimmten Aussagegegenstand treffen.

Topics sollen innerhalb einer Topic Map genau einen Aussagegegenstand repräsentieren. Diese grundlegende Idee wird auch als SLUO, Subject Location Uniqueness Objective, bezeichnet (vgl. Sigel 2004, S. 7).

Es können adressierbare und nicht adressierbare Aussagegegenstände unterschieden werden. Ist eine konkrete Ressource selbst der Aussagegegenstand eines Topics, bspw. eine Webseite, so hat dieser Aussagegegenstand eine Adresse, in diesem Bsp. einen URL. Diese Adresse wird als Subject Locator bezeichnet.

Personen oder Begriffe können nicht direkt adressiert werden. In diesem Fall kann auf Ressourcen als Indikatoren für die Existenz dieser Subjekte verwiesen werden (Proxyfunktion). Die Adressen, die auf diese Ressourcen, die sog. Subject Indicators, verweisen, werden als Subject Identifier bezeichnet. So verweist bspw. der URL <http://www.t8d.de/johannesschmidt.php> auf eine Ressource, auf der Hinweise auf die Identität eines Johannes Schmidt enthalten sind, der an der HAW Hamburg Bibliotheks- und Informationsmanagement studiert.

Über die Eindeutigkeit von URIs kann eine eindeutige Identität von Aussagegegenständen für den Aspekt der technischen Interoperabilität und der Interaktion sowie dem Verschmelzen, das Merging, von Topic Maps in einem globalen Kontext gewährleistet werden. Im Hinblick auf die Semantik bleibt der für menschliche Individuen rezipierbare Inhalt einer Ressource entscheidend. Für den diskutierten Wissensbegriff, der nicht allgemeingültig definiert ist, können Verweise, je nach Ansicht des Topic Map-Autoren, auf unterschiedliche Indikatorressourcen zeigen. Gleiche Verweise besonders bei abstrakten Konzepten/Begriffen wie „Wissen“ oder „Information“ in verschiedenen, von unterschiedlichen Autoren erarbeiteten Topic Maps lassen auf ein übereinstimmendes Verständnis zu den jeweiligen Sachverhalten schließen.

Dieser Aspekt ist für die Bildung von Communities von besonderer Bedeutung.

Der Prozess, einen Aussagegegenstand durch ein Topic zu repräsentieren, wird in der Literatur auch als Reification bezeichnet (vgl. Vatant 2003, S. 67; Rath 2003, S. 11). Die Spezifikation zum Topic Maps Data Model definiert Reification allerdings wie folgt:

The act of reification is the act of making a topic represent the subject of another topic map construct in the same topic map. (ISO/IEC 13250-2)

Konkret: Sollen einem Topic Maps-Konstrukt, bspw. einer Occurrence, weitere Informationen wie bspw. eine Benennung zugeordnet werden, so muss ein Topic definiert werden, das dieses Konstrukt bzw. dessen *subject* repräsentiert. Diesem Topic kann dann eine Benennung über Topic Name zugewiesen werden.

Eine besondere Bedeutung kommt den Published Subject Indicators zu, die auf sie verweisenden Adressen werden Published Subject Identifiers genannt. Das Akronym PSI kann dabei für beide Termini stehen.

Die Schwierigkeit bei der Definition von Subject Identifiers, also der Verweis auf eine Indikatorressource, ist eben die, einen qualifizierten Verweis zu erstellen, d. h. auf eine inhaltlich (die Semantik betreffend) und bzgl. des Zugriffsorts verlässliche Quelle zu zeigen, die speziell bei abstrakten Konzepten/Begriffen dem individuellen Verständnis des Autors oder einer Gemeinschaft entsprechen muss.

Published Subject Indicators sollen dieser Anforderung entsprechen, zusätzlich allgemein über einen URI zugänglich sein und letztlich einen eindeutigen Hinweis über die Identität des Aussagegegenstandes erlauben (vgl. Vatant 2003, S. 73). PSIs reflektieren in erster Linie die Problematik der gewünschten Kongruenz zwischen Ressourceninhalt (die identifizierende Beschreibung, über die auf die Semantik geschlossen werden kann) und Verständnis des Autors zu jeweiligen Konzepten bzw. Begriffen.

Die organisatorische Frage hinsichtlich einer prüfenden Autorität, die die Verlässlichkeit, d. h. die Qualität der Quelle bestätigt, bleibt bis dato ungelöst. Im Zusammenhang mit der Erstellung von Topic Maps regt Vatant an, PSI-Repositories zum einen selbst als Topic Map anzulegen und dort andererseits die dieses Repository verwendenden Autoren sukzessive einzubinden und so *bottom-up* einen Autoritäten- und Vertrauensbildungsprozess zu unterstützen

(vgl. Vatant 2003, S. 76). Ein theoretisches Gegenbeispiel und *top-down* Ansatz wäre eine global akzeptierte und verstandene Universalenzyklopädie, deren Einträge per URI referenziert werden können.

Die PSIs für die Elemente der Topic Maps-Syntax XTM 1.0 bspw. sind als Topic Map in XTM 1.0 definiert (<http://topicmaps.org/xtm/1.0/core.xtm>).

5.3 Das Topic Maps Data Model

Es ist bereits erwähnt worden, dass der Ursprung von Topic Maps darin lag, Lösungen für die Integration verschiedener Indizes zu erarbeiten, deren erstes Ergebnis eine Markupsprache war, die den Austausch der Indizes ermöglichte. Der Syntax ISO 13250 HyTM sowie auch die zunächst von einer Autorengruppe des unabhängigen Konsortiums TopicMaps.org entwickelte XML-basierte Syntax XTM 1.0 liegt jedoch ein abstraktes Metamodell zu Grunde, das vor der Standardisierung der Syntaxen nicht standardisierter Bestandteil (genauer: Grundlage) von ISO/IEC 13250 Topic Maps war. Aus solch einem Modell muss sich eine Syntax ableiten, die letztlich einen Austausch von Instanzen von Repräsentationen dieses Modells, Topic Maps, ermöglicht. Entsprechend befindet sich ISO/IEC 13250 hinsichtlich der Syntax in einem Konsolidierungsprozess, da mit dem Standard Application Model (SAM) bzw. Topic Maps Data Model (TMDM) als spätere offizielle Bezeichnung ein offizieller Entwurf einer Spezifikation für ein anwendungsbezogenes Modell in 2006 vorliegt (siehe ISO/IEC 13250-2).

Auf die Veränderungen in der Syntax wird im Kapitel 5.4 kurz eingegangen. Im folgenden wird der Entwurf der Spezifikation als Spezifikation bezeichnet.

ISO/IEC 13250 lässt sich dreischichtig gliedern:

- Modellebene,
- Syntaxebene,
- Constraint- und Abfrageebene.

Abbildung 19 zeigt die Ebenen und Bestandteile.

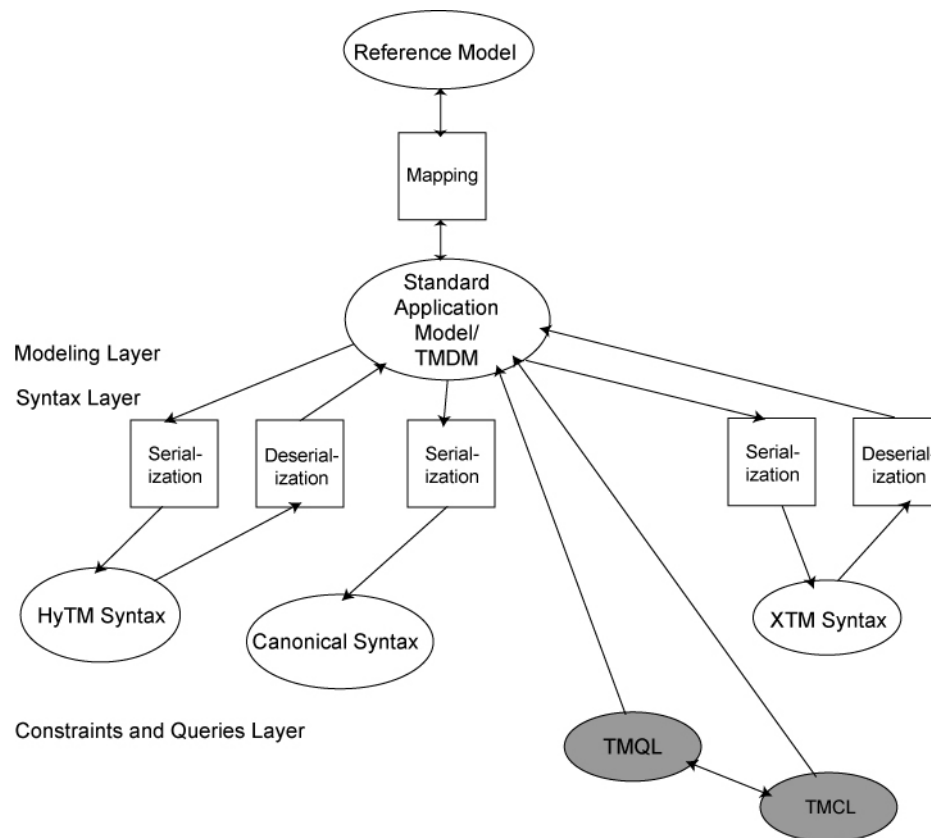


Abb. 19: Ebenen und Bestandteile von ISO/IEC 13250 Topic Maps (vgl. Hunting 2003, S. 56; eigene Darst.)

Bis auf HyTM und XTM 1.0, das nicht innerhalb der ISO veröffentlicht worden ist, liegen alle Bestandteile noch als Entwurf, jedoch teilweise final, vor (Stand 9.2006). So lässt sich konstatieren, dass ISO/IEC 13250 noch kein vollständiges homogenes Standardpaket darstellt. Daraus resultieren einige Nachteile, z. B. in der Implementierung von Software, da diese nur einen Status quo des Standardisierungsprozesses abbilden kann. Andererseits besteht ein höheres Diskussions- und Beteiligungspotential für die Community.

Das Reference Model (RM) definiert die Grundmenge an ontologischen Festlegungen (vgl. Hunting 2003, S. 56). Das TMDM erweitert das RM als formales Datenmodell, das alle Topic Maps-Konstrukte mit ihren Benennungen und Eigenschaften definiert, entsprechend der W3C XML InfoSet-Spezifikation (vgl. Biezunski/Newcomb/Bryan 2002; Hunting 2003, S. 62). Zusätzlich werden die zulässigen Beziehungen der Topic Maps-Konstrukte zueinander spezifiziert. Die grafische Darstellung von Zusammenhängen erfolgt in der Spezifikation mit Klassendiagrammen in UML-Notation, die Diagramme haben jedoch einen rein informativen Charakter und dienen nicht als verbindliche Vorgabe (vgl. ISO/IEC 13250-2).

Abbildung 20 zeigt das Metamodell des Datenmodells in hierarchischer Anordnung der die Topic Maps-Konstrukte repräsentierenden Klassen.

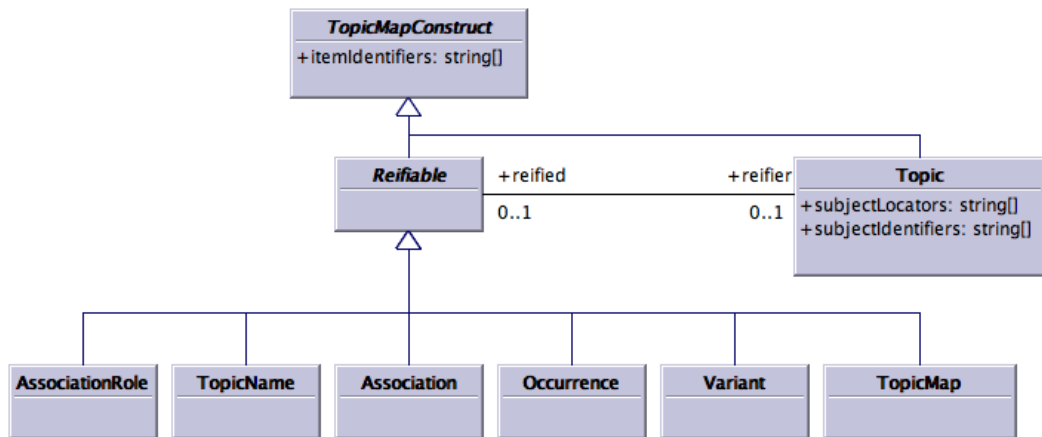


Abb. 20: Metamodell des Topic Maps Data Model (ISO/IEC 13250-2)

In der Spezifikation sind die Klassenentsprechungen Information Item Types, Instanzen sind Information Items wie konkret Topic Items oder Association Items.

TopicMapConstruct ist eine abstrakte Superklasse, um mit dem Prinzip der Vererbung die Darstellung in UML Klassendiagrammen zu vereinfachen (vgl. ISO/IEC 13250-2). Dennoch ist diese Superklasse im Hinblick auf anwendungsinterne Repräsentationen des TMDM praktisch, da sie ein Gesamtverzeichnis, einen Index, aller Instanzen, die Topic Maps-Konstrukte repräsentieren, darstellt. Dieser Aspekt wird in Kapitel 5.7.2 noch einmal aufgegriffen.

Reifiable ist ebenfalls eine abstrakte Klasse, die über das Vererbungsprinzip illustriert, welche Items eines Item Types in der Eigenschaft [reified] eines Topic Items auftreten können, also von einem Topic repräsentiert werden können (siehe auch Kapitel 5.2).

Die Spezifikation definiert, welche Werte eine Eigenschaft eines Information Items neben Information Item selbst enthalten darf:

- „String“ gemäß UNICODE und ISO 10646;
- „Set“, entspricht einer math. Menge, die ausschließlich Information Items oder Strings enthält;
- „NULL“ als genau ein Wert, der anzeigt, dass eine Eigenschaft keinen Wert hat. NULL darf nicht in Set enthalten sein;

- „Locator“ als ein String, der einer bestimmten Adressnotation entspricht (vgl. ISO/IEC 13250-2).

Das TMDM selbst spezifiziert nur die grundlegenden Typen String und NULL. Mit Hilfe des Prinzips der Datentypen ist es möglich, jeglichen Datentypen darzustellen. Jeder konkrete Wert eines Datentyps wird in einer Topic Map als String repräsentiert. Eine separat gespeicherte Adresse in IRI-Notation identifiziert den Datentypen. Alle Datentypen sind in der Spezifikation zu den XML Schema Datentypen (<http://www.w3.org/TR/xmlschema-2>) definiert. Folgend ein Beispiel für Integer:

Wert: „2“ (String), Identifier des Datentyps: <http://www.w3.org/TR/xmlschema-2/#integer>. Also ist der Wert als die Ganze Zahl 2 zu verarbeiten.

Im folgenden soll Topic Item exemplarisch dargestellt werden, das im Modell das Topic Konstrukt repräsentiert.

Aussagegegenstände, Subject Identity und Reification sind im Zusammenhang mit Topics bereits vorgestellt worden. Ein weiteres mächtiges Prinzip im Topic Maps-Paradigma ist Scope. Scope definiert den Kontext, in dem eine Aussage (*statement*) gültig ist (vgl. ISO/IEC 13250-2). Ein Scope wird durch eine Menge von Topics bestimmt bzw. definiert. Eine typische Anwendung für Scope ist die Einschränkung auf einen sprachlichen Kontext.

Abbildung 21 zeigt die Beziehungsstruktur zwischen Topic Item und anderen Items des TMDM.

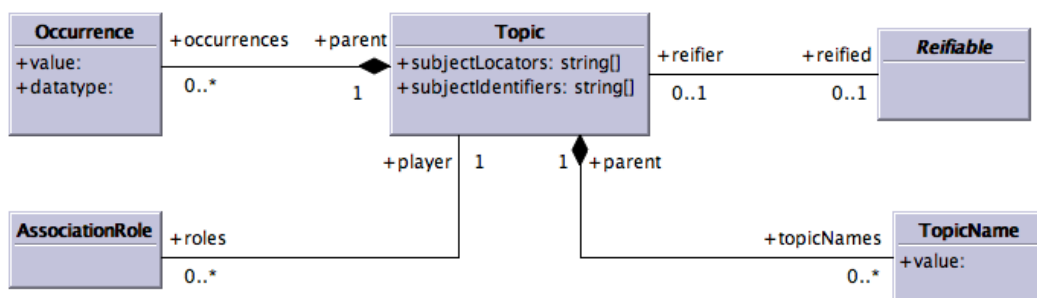


Abb. 21: Topic Item (ISO/IEC 13250-2)

Die Komposition zwischen Occurrence und Topic Name zeigt an, dass diese existenzabhängig zu Topic sind und Topic das Parent-Objekt ist. Entsprechend haben Occurrence Item und Topic Name Item eine [parent]-Eigenschaft, die das beinhaltende Topic Item enthält.

Einem Topic Item sind optional ein oder mehrere Occurrence oder Topic Name Items zugeordnet. Ein Occurrence und ein Topic Name Item gehören immer zu genau einem Topic Item. Die Assoziation zwischen Topic Item und Association Role Item zeigt die Konstruktion, wie ein Topic bzw. der repräsentierte Aussagegegenstand in einer oder mehreren Associations über die Beziehungsrolle beteiligt sein kann. Eine Beziehungsrolle ist immer genau einem Topic zugeordnet. Die Assoziation zur abstrakten Klasse Reifiable zeigt, dass ein Topic optional ein anderes Item reifizieren kann (siehe Abb. 20). Die Beziehungen zu den genannten Items werden in der Spezifikation unter den Eigenschaftenbezeichnungen [topic names], [occurrences], [roles played] und [reified] präzisiert. Die folgende Liste zeigt ergänzend die weiteren möglichen Eigenschaften eines Topic Items im Überblick:

- [subject identifiers]: A set of locators. The locators referring to the subject indicators of this topic.
- [subject locators]: A set of locators. The locators referring to the information resource that is the subject of this topic.
- [item identifiers]: A set of locators. The item identifiers of the topic.
- [parent]: An information item. The topic map containing the topic.
Computed value: the topic map item whose [topics] property contains this topic item. (ISO/IEC 13250-2)

Das TMDM definiert weiter, wann zwei Topic Item als identisch zu betrachten sind. Dies gilt für folgende Bedingungen:

- Sie haben mind. einen gleichen String in ihren [subject identifiers] Eigenschaften;
- Sie haben mind. einen gleichen String in ihren [item identifiers] Eigenschaften;
- Sie haben mind. einen gleichen String in ihren [subject locators] Eigenschaften;
- Sie haben einen gleichen String in der [subject identifiers] Eigenschaft des einen Topic Items und der [item identifiers] Eigenschaft des anderen Topic Items;
- Sie haben das gleiche Information Item in ihren [reified] Eigenschaften, repräsentieren also das gleiche Information Item bzw. dessen *subject*. (vgl. ISO/IEC 13250-2)

Im Falle einer erfüllten Bedingung ist das Merging, das Verschmelzen von Topics zur Vermeidung von Redundanzen, durchzuführen (siehe auch SLUO

S. 71). Die Prozedur für Topic Items ist in der Spezifikation unter <http://isotopicmaps.org/sam/sam-model/#sect-merge-topics> definiert. Bei einer Gleichheit von Topic Items A und B entsprechend der genannten Bedingungen ist zunächst ein neues Topic Item C zu erstellen. Dieses ersetzt dann alle Vorkommen der Topic Items A und B in den Eigenschaften anderer Information Items, z. B. [type] oder [reifier] und nimmt selbst alle Eigenschaften der Topic Items A und B wie z. B. [topic names] an.

Warum die Spezifikation vorgibt, zunächst ein neues Topic Item C zu erstellen, ist nicht ganz klar. Ein gleiches Resultat des Mergings, nämlich ein Topic Item, das

- alle Eigenschaften von A und B annimmt und
- alle Vorkommen von A und B in den Eigenschaften anderer Information Items besetzt,

wird auch erreicht, wenn A alle Eigenschaften von B zusätzlich aufnimmt und B in allen Eigenschaftenvorkommen anderer Information Items ersetzt.

Diese Frage stellte der Autor an die TMAPI E-Mail-Liste (<https://lists.sourceforge.net/lists/listinfo/tmap-discuss>). Lars Heuer, einer der Autoren von TMAPI, bestätigte die oben erarbeitete Feststellung, dass ein drittes Topic Item C für den Prozess des Mergings überflüssig ist. Er verwies noch einmal darauf, dass nach einem wie oben skizzierten Prozess alle ehemaligen Eigenschaften und Referenzen von B erhalten bleiben müssen, in dem diese auf A übertragen werden (vgl. Anhang A).

Die Spezifikation definiert, dass Topic Items einen Wert in mind. einer der [subject identifiers], [subject locators] und [item identifiers] Eigenschaften aufweisen sollen, der nicht einer leeren Menge (Set) entspricht. Die UND-Verknüpfung von [subject identifiers], [subject locators] und [item identifiers] im Zusammenhang mit „[...] einen Wert in mind. einer[...]“ („[...] a value for at least one of the [...]“ [ISO/IEC 13250-2]) ist so nicht ganz verständlich. Es müsste eine ODER-Verknüpfung sein und demnach lauten, dass Topic Items einen Wert in mind. einer der [subject identifiers], [subject locators] oder [item identifiers] Eigenschaften aufweisen sollen.

Die Spezifikation definiert abschließend informativ Subject Identifier mit dem Zweck, für jeden im Topic Maps Data Model verwendeten Terminus eine eindeutige Referenz auf die Aussagegegenstände zu ermöglichen, die sie identifizieren, um so bspw. das Zusammentragen von Informationen über die Aus-

sagegegenstände zu ermöglichen. So würden zwei Topics A und B, die in ihrer [subject identifiers] Eigenschaft jeweils eine Adresse mit dem Wert „<http://psi.topicmaps.org/iso13250/glossary/topic>“ aufweisen, in ihrer [occurrence] Eigenschaft jedoch jeweils ein unterschiedliches Occurrence Item besitzen, gemäß der obigen Regeln zu einem Topic vereint, das nach der Prozedur zwei Occurrence Items beinhaltet.

5.4 Syntax XTM

HyTM, das auf HyTime basiert, war die erste Syntax für Topic Maps und ist Ende 1999 als Bestandteil von ISO 13250 veröffentlicht worden. HyTM ist in SGML spezifiziert und definiert keine alleinstehende Syntax über eine DTD, sondern erfordert die Ableitung eigener Elemente von einer Meta-DTD, was letztlich eine erhöhte Komplexität bedeutet (vgl. Hunting 2003, S. 53). Kurz nach der Veröffentlichung von HyTM entwickelten Autoren des erwähnten unabhängigen Konsortiums TopicMaps.org (<http://www.topicmaps.org>) die Syntax XTM 1.0, die auf XML basiert und damit die technologischen Entwicklungen im Internet Ende der 1990er Jahre berücksichtigt. Das Akronym „XTM“ steht dabei für „XML Topic Maps“; im folgenden können XTM 1.0 und XTM synonym verwendet werden. Neben der Spezifikation einer XML Grammatik definiert die XTM Autorengruppe ein abstraktes Modell (vgl. TopicMaps.org 2001), da in ISO/IEC 13250, wie erwähnt, die Modellebene (siehe Abb. 19) zur Zeit der Verabschiedung von XTM 1.0 Anfang 2001 noch nicht spezifiziert war, einer Syntax jedoch ein solches Modell zu Grunde liegt.

Die wichtigen Unterschiede gegenüber HyTM sind:

- Definition einer einzigen DTD;
- Nicht-Berücksichtigung des `facet`-Elements;
- Generalisierung von `<sortName>` und `<dispName>` zu `<variant>` (Erhalt der Semantik der HyTM-Elementtypen über die Zuweisung von Published Subject Indicators für Kindelemente des `parameter-Elements`);
- Unterscheidung zwischen Ressourcen, die Indikatoren für einen Aussagegegenstand darstellen und Ressourcen, die selbst der Aussagegegenstand sind;
- XLink zur Definition von Hyperlinks;

- Weitestgehende Ersetzung von Attributen durch Elemente. (vgl. Hunting 2003, S. 54)

Abbildung 22 zeigt den zeitlichen Entwicklungszusammenhang zwischen HyTM und XTM 1.0.

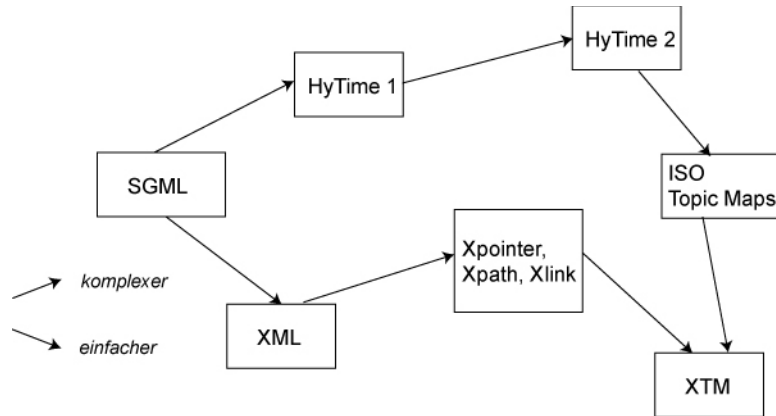


Abb. 22: Zeitlicher Entwicklungszusammenhang zwischen HyTM und XTM 1.0 (vgl. Biezunski 2003, S. 24; eigene Darst.)

Im folgenden soll die Syntax anhand einiger bereits verwendeter Beispiele kurz vorgestellt werden. Die Spezifikation ist unter <http://www.topicmaps.org/xtm/> formuliert.

Das `topicMap`-Element ist das Root-Element eines XTM Dokumentes, das nur eine Topic Map enthält und welches Elternelement für `<topic>`, `<association>` und `<mergeMap>` ist. Das Inhaltsmodell in der DTD sieht für `<topicMap>` entsprechend wie folgt aus:

```

<!ELEMENT topicMap
      ( topic | association | mergeMap ) *
>

```

(Altheim/Biezunski/Hunting/Newcomb 2001)

`<topic>`, `<association>` und `<mergeMap>` können demnach optional mehrfach auftreten. Folgendes Listing zeigt das `topicMap`-Element mit allen zulässigen Attributen, dies ist zugleich Inhalt eines gültigen XTM Dokumentes:

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE topicMap PUBLIC "-//TopicMaps.Org//DTD XML Topic Map (XTM)
1.0//EN" "http://topicmaps.org/xtm/1.0/xtml.dtd">
<topicMap
  xmlns="http://www.topicmaps.org/xtm/1.0/"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xml:base="http://localhost/phptmapi/impexp/xtm/"
  id="topicmap-1">
</topicMap>

```

XTM-Listing 1: Element <topicMap>

Das Inhaltsmodell für <topic> lautet:

```

<!ELEMENT topic
  ( instanceOf*, subjectIdentity?, ( baseName | occurrence ) * )
>

```

(Altheim/Biezunski/Hunting/Newcomb 2001)

Folgend werden zwei Topics mit Benennungen definiert:

```

<topic id="topic-1">
  <baseName>
    <baseNameString>Johannes Schmidt</baseNameString>
  </baseName>
</topic>
<topic id="topic-2">
  <baseName>
    <baseNameString>Studierende</baseNameString>
  </baseName>
</topic>

```

XTM-Listing 2: Element <topic>

Zur Identifikation der Aussagegegenstände, welche diese Topics repräsentieren, wird über einen Subject Identifier jeweils auf eine Indikatorressource gezeigt:

```

<topic id="topic-1">
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://www.t8d.de/johannesschmidt.php"/>
    </subjectIndicatorRef>
  </subjectIdentity>
  <baseName>
    <baseNameString>Johannes Schmidt</baseNameString>
  </baseName>
</topic>

```

```

        </baseName>
</topic>
<topic id="topic-2">
    <subjectIdentity>
        <subjectIndicatorRef xlink:href="http://www.haw-
hamburg.de/Studierendenzentrum.7250.0.html"/>
    </subjectIdentity>
    <baseName>
        <baseNameString>Studierende</baseNameString>
    </baseName>
</topic>

```

XTM-Listing 3: Element `<topic>` und Verweis auf einen Subject Indicator

Über das `instanceOf`-Element kann die Klassenzugehörigkeit eines Topics ausgedrückt werden. Das Topic mit der Benennung „Studierende“ soll als Topic Type für das Topic mit der Benennung „Johannes Schmidt“ definiert werden. Folgendes Listing zeigt diese Beziehung:

```

<topic id="topic-1">
    <instanceOf>
        <topicRef xlink:href="#topic-2"/>
    </instanceOf>
    <subjectIdentity>
        <subjectIndicatorRef
xlink:href="http://www.t8d.de/page_15.php"/>
    </subjectIdentity>
    <baseName>
        <baseNameString>Johannes Schmidt</baseNameString>
    </baseName>
</topic>
<topic id="topic-2">
    <subjectIdentity>
        <subjectIndicatorRef xlink:href="http://www.haw-
hamburg.de/Studierendenzentrum.7250.0.html"/>
    </subjectIdentity>
    <baseName>
        <baseNameString>Studierende</baseNameString>
    </baseName>
</topic>

```

XTM-Listing 4: Element `<topic>` und Klasse-Instanz-Beziehung über `<instanceOf>`

Die Abbildung einer Beziehung zwischen den Topics [Diplomarbeit] und [Johannes Schmidt] erfolgt über das `association`-Element. Das Inhaltsmodell für `<association>` sieht wie folgt aus:

```
<!ELEMENT association
    ( instanceOf?, scope?, member+ )
>
```

(Altheim/Biezunski/Hunting/Newcomb 2001)

Um die Beziehung abzubilden, dass Johannes Schmidt eine Diplomarbeit verfasst hat, müssen neben einem Topic mit der Benennung „Diplomarbeit“ zusätzlich Topics definiert werden, die als Association Type zur Typisierung der Association und Association Role Type zur Charakterisierung der Beziehungsrolle zugewiesen werden. Listing 5 zeigt eine entsprechende Association:

```
<association id="association-1">
    <instanceOf>
        <topicRef xlink:href="#topic-6"/>
    </instanceOf>
    <member>
        <roleSpec>
            <topicRef xlink:href="#topic-4"/>
        </roleSpec>
        <topicRef xlink:href="#topic-1"/>
    </member>
    <member>
        <roleSpec>
            <topicRef xlink:href="#topic-5"/>
        </roleSpec>
        <topicRef xlink:href="#topic-3"/>
    </member>
</association>
```

XTM-Listing 5: Element `<association>`

Es wurde bereits darauf hingewiesen, dass die Spezifikation für XTM 1.0 vor einem offiziellen Entwurf für das Topic Maps Data Model (TMDM) und zudem außerhalb der ISO veröffentlicht worden ist. Inzwischen liegt seit dem 19.6.2006 ein Entwurf für eine XML Syntax für Topic Maps als ISO/IEC 13250-3 unter <http://www.isotopicmaps.org/sam/sam-xtm/> vor. Synonym ist diese als XTM 2.0 zu bezeichnen. In der Spezifikation wird explizit auf die Interpretation in Abhängigkeit zum TMDM hingewiesen (vgl. ISO/IEC 13250-3).

Es kann festgestellt werden, dass sich die modifizierte Topic Maps XML Syntax nun am Datenmodell orientiert.

Der Prozess, eine Topic Map aus einer implementierungsinternen Repräsentation des TMDM als eine Instanz einer Topic Maps-Syntax zu exportieren, wird als Serialisierung bezeichnet. Deserialisierung als umgekehrter Prozess beschreibt die Erstellung einer Instanz der implementierungsinternen Repräsentation des TMDM aus einer Instanz einer Topic Maps-Syntax (vgl. ISO/IEC 13250-3; Abb. 19). Diese Prozesse müssen konsistent erfolgen. D. h., dass einem „element item“ (ISO/IEC 13250-3) x der Syntax immer exakt ein bestimmtes „information item“ (ISO/IEC 13250-3) y der Repräsentation des Topic Maps Data Model zugewiesen werden muss und umgekehrt.

XTM 1.0 konnte auf Grund der zeitlichen Differenz in der Veröffentlichung zwangsläufig einige Spezifikationen des TMDM nicht berücksichtigen, was letztlich zu Schwierigkeiten in Serialisierungs- und Deserialisierungsprozessen führt. Entsprechend steht die Einführung von XTM 2.0 für eine Konsolidierung auf XML Syntaxebene, die konsistente Serialisierungs- und Deserialisierungsprozesse erlauben soll. Ein Beispiel: Eine Klasse-Instanz-Beziehung wird in XTM 1.0 mit dem `<instanceOf>`-Element formuliert. In XTM 2.0 wird `<instanceOf>` außer bei `<topic>` durch `<type>` ersetzt (vgl. ISO/IEC 13250-3). Das TMDM erlaubt für Topic Name Items die Definition eines Topic Name Type (vgl. ISO/IEC 13250-2), das eine Präzisierung der Bedeutung eines Topic Name Items erlaubt. Das Inhaltsmodell für `<name>`, das `<baseName>` ersetzt, stellt sich in XTM 2.0 entsprechend wie folgt dar:

```
<!ELEMENT name
    ( itemIdentity*, type?, scope?, value, variant* )
>
```

(ISO/IEC 13250-3)

Zum Vergleich das Inhaltsmodell für `<baseName>` in XTM 1.0, das keine Definition einer solchen Relation vorsieht:

```
<!ELEMENT baseName
    ( scope?, baseNameString, variant* )
>
```

(Altheim/Biezunski/Hunting/Newcomb 2001)

In einem Serialisierungsprozess in XTM 1.0 Syntax aus einer anwendungsinternen Repräsentation des TMDM geht dann Information verloren, wenn gemäß TMDM ein Topic Name typisiert worden ist, eine solche Beziehung aber über die Syntax nicht abgebildet werden kann. Letztendlich bedeutet dies auch einen Verlust an Semantik.

Neben terminologischen Anpassungen von XTM 2.0 gegenüber dem TMDM wird in der modifizierten XML Syntax auf XLink und XML Base verzichtet. Zur Reduzierung von Komplexität kann das `variant`-Element, das zur Definition von Alternativbezeichnungen verwendet werden kann, nicht mehr in sich verschachtelt werden (*nesting*). Eine detaillierte Übersicht der Änderungen kann unter ISO/IEC 13250-3 in dortigem Anhang D eingesehen werden.

5.5 TMAPI

Ein Application Programming Interface (API), das kurz mit „Programmierschnittstelle“ übersetzt werden kann, erlaubt Softwareentwicklern die Anbindung bzw. Verwendung anderer Software für eigene Programme. Ein API definiert z. B. Klassen mit Methoden, über die bzw. über von diesen instanziierte Objekte dann eine Software bzw. ihre Funktionalität genutzt werden kann.

TMAPI ist das Topic Maps Application Programming Interface und das Ergebnis der Arbeit einer internationalen Autorengruppe. TMAPI steht ohne Lizenz einschränkungen (*public domain*) frei zur Verfügung. Es beschreibt eine einheitliche Schnittstelle über eine Mindestmenge an Interfaces für die Einbindung von Anwendungen, die das Erstellen und Bearbeiten von Topic Maps erlaubt. Es soll dazu dienen,

- Entwicklern durch die Einheitlichkeit den Lernaufwand für kompatible Anwendungen zu reduzieren,
- die Portierbarkeit von Anwendungen zu erleichtern und
- die Entwicklung von Programmbibliotheken zu vereinfachen (vgl. Ahmed 2001).

Die Autoren fassen die Intention von TMAPI anhand eines Vergleiches wie folgt zusammen: „TMAPI hopes to do for topic maps what SAX and DOM did for XML [...]“ (TMAPI 2005a). TMAPI berücksichtigt in der Abbildung das Paradigma der Objektorientierten Anwendungsentwicklung (Klassen, Eigenschaften, Methoden, Vererbung).

Ähnlich wie in der Entwicklung der Syntaxen liegt bei TMAPI und der Referenzimplementierung in Java ebenso das Problem vor, dass im Grunde zwei Modelle, das TMDM und das im Zusammenhang mit XTM 1.0 spezifizierte Modell berücksichtigt werden müssen. So definiert TMAPI zwei Konfigurationsparameter für TMAPI-implementierende Anwendungen, damit eine korrekte Verarbeitung der Daten entsprechend des zu Grunde liegenden Modells erfolgen kann:

- <http://www.tmapi.org/features/model/xtm1.0> (Modell für XTM 1.0)
- <http://www.tmapi.org/features/model/xtm1.1> (TMDM)

XTM 1.1 bezieht sich auf einen ersten Entwurf für eine modifizierte, das TMDM berücksichtigende XML Syntax, die nun als XTM 2.0 endgültig spezifiziert werden soll.

TMAPI 1.0 umfasst neben dem Bytecode der Referenzimplementierung in Java als Jar-Archiv deren Javadoc-Dokumentation sowie eine knappe Einführung in TMAPI und UML Klassendiagramme der einzelnen Pakete. Anhand der Portierung von TMAPI nach PHP bzw. PHP 5 wird das Topic Maps-API eingehender vorgestellt.

5.6 PHPTMAPI

PHPTMAPI ist die Portierung von TMAPI nach PHP 5. PHPTMAPI 1.0 ist unter Mitwirkung von Andreas Stephan sowie beratender Unterstützung von Lars Heuer maßgeblich vom Autoren dieser Diplomarbeit erarbeitet worden. PHPTMAPI ist bei sourceforge.net als Projekt „TMAPI/PHP“ mit der Projektseite <http://phptmapi.sourceforge.net> gelistet und steht ebenso wie TMAPI ohne jegliche Lizenz einschränkungen frei zur Verfügung. Initiator des Projekts bei sourceforge.net ist Espen Holje.

Bestandteile von PHPTMAPI 1.0 sind

- der Quellcode der Pakete core, index und index.core (Archiv src),
- die mit phpDocumentor (<http://www.phpdoc.org>) erstellte, an Javadoc angelehnte API-Dokumentation des Quellcodes (Archiv phpdocs) sowie
- UML Klassendiagramme der Pakete (Archiv uml).

Diese können bei sourceforge.net heruntergeladen werden, ein Verweis auf den Downloadbereich befindet sich auf der o. g. Projektseite. PHPTMAPI 1.0

ist als Bestandteil dieser Diplomarbeit auch auf der beigelegten CD-ROM (Anhang M) im Verzeichnis `phptmapi1.0` enthalten.

In der internen und externen Kommunikation wird das Projekt bisweilen auch als Implementierung von TMAPI bezeichnet, weil die mit UML-Klassendiagrammen und Javadoc spezifizierte Schnittstelle in andere Programmiersprachen unter Berücksichtigung sprachentypischer Eigenheiten letztlich mit konkretem Quellcode umgesetzt wird.

Die grundsätzliche persönliche Motivation für die Entwicklung und Veröffentlichung von PHPTMAPI lag darin begründet, eine Basis für die Verwendung von Topic Maps als Technologie in der Web-Entwicklung mit der populären Skriptsprache PHP und letztendlich auch für eigene Projekte zu schaffen. In der Verwendung der eingeführten Motivationstypologie kann primär eine intrinsische Motivation und entsprechend des Open Source-Gedankens eine altruistisch-gemeinschaftliche Intention zu Grunde gelegt werden.

Der Bereich der Topic Maps-Software, auch nicht TMAPI kompatibel, ist Java-dominiert. Einen Überblick hat bspw. Woodman unter <http://www.topicmap.com/topicmap/tools.html> zusammengestellt. Im Bereich der Web-Entwicklung wird Java im Vergleich zu PHP jedoch deutlich seltener angewendet. E-Soft Inc. ermittelt in seiner monatlichen Auswertung von über 20 Millionen Webservern für den September 2006, dass über 73% einen Apache Webserver einsetzen (vgl. E-Soft Inc. 2006a). Auf diesen sind in knapp 41% PHP als Modul und etwa 2,5% als PHP-CGI installiert, Tomcat für Java-Anwendungen dagegen auf 0,03% (vgl. E-Soft Inc. 2006b). Netcraft Ltd. misst in seinem monatlichen Report in der Auswertung von knapp 97 Millionen Webseiten für September 2006 einen Anteil von etwa 62% (vgl. Netcraft Ltd. 2006). Der tatsächliche Wert für den Apache Webserver ist zwischen diesen Messungen zu vermuten. Für PHP-Installationen liefert Netcraft Ltd. keine Statistiken.

Die explizite Portierung nach PHP 5 in der Verwendung der Konstrukte des erweiterten und verbesserten Objektmodells in PHP zieht zwangsläufig eine Nicht-Abwärtskompatibilität zu PHP-Versionen < 5.0 nach sich und sorgt auf Objektmodellebene gleichsam für eine größere Ähnlichkeit zu bspw. Java, das als grundsätzlich objektorientiert konzipiert ist. Mit PHP 5 ist eine sauberere, d. h. enger angelegte Portierung möglich, als es mit PHP 4.x der Fall wäre. Deshalb sollen ausgewählte, in PHPTMAPI verwendete Konstrukte der

Objektorientierten Programmierung (OOP) in PHP 5 in einem Exkurs kurz vorgestellt werden.

5.6.1 Exkurs: PHP 5

PHP 5 ermöglicht die Verwendung von Interfaces als spezielle Klassen. Interfaces erlauben die bloße Vordefinition von Schnittstellen einer Klasse (vgl. Fuecks 2005, S. 83), definieren also nur deren Struktur ohne Attributdefinitionen und mit Funktionen, die keinen Inhalt (*body*) enthalten sowie als einzig möglichen Zugriffstypen *public* (s. u.) festlegen dürfen, wenn dies explizit definiert werden soll. Mit Interfaces kann so einerseits die Grundstruktur einer objektorientierten Anwendung konzipiert und zugleich für die die Interfaces implementierenden Klassen eine verbindliche Implementierungsvorschrift vorgegeben werden. Da TMAPI eine Struktur aus Interfaces definiert, stellt dieses Konstrukt einen wesentlichen Aspekt für eine konsistentere Portierung dar.

PHP 5 erlaubt wie Java keine Mehrfachvererbung (*multiple inheritance*) bei konkreten und abstrakten Klassen. Bei Interfaces jedoch ist dies möglich. So lassen sich auf Konzeptionsebene zur besseren Übersicht gleichartige Funktionen unterschiedlicher Klassen in ein separates Interface ausgliedern und über Vererbung den entsprechenden Interfaces wieder zuordnen. PHP-Listing 1 zeigt *multiple inheritance* bei Interfaces über zwei Ebenen von `i_a` über `i_b` zu `i_c` mit zusätzlich vererbtem Interface `i_commoncontainer`.

```
<?php
interface i_a {
    function a_function();
}
interface i_b extends i_a {
    function b_function();
}
interface i_commoncontainer {
    function commonfunction();
}
interface i_c extends i_b, i_commoncontainer {
    function c_function();
}
?>
```

PHP-Listing 1: Mehrfachvererbung bei Interfaces in PHP 5

Wird die in Listing eins gezeigte Vererbungsstruktur mit konkreten Klassen umgesetzt, muss die das Interface `i_c` implementierende Klasse die Funktion `commonfunction()` aus dem Interface `i_commoncontainer` dann konkret definieren, damit der bei den Interfaces vorgegebenen multiplen Hierarchie entsprochen werden kann. PHP-Listing 2 zeigt die Implementierung der Struktur in PHP-Listing 1.

```
<?php
class c_a implements i_a {
    function a_function(){}
}
class c_b extends c_a implements i_b {
    function b_function(){}
}
class c_d extends c_b implements i_c {
    function c_function(){}
    function commonfunction(){}
}
?>
```

PHP-Listing 2: Umsetzung einer Mehrfachvererbung bei Interfaces mit konkreten Klassen in PHP 5

PHP 5 bietet die für OO-Sprachen typischen Zugriffsmodifikatoren für Eigenschaften und Methoden *public*, *private* und *protected*, über die sich die Sichtbarkeit eines Objektes steuern lässt (vgl. Fuecks 2005, S. 52). Wenn die Methode `c_function()` der Klasse `c_d` in PHP-Listing 2 als *private* deklariert würde, könnte diese Methode nur innerhalb der Klasse, nicht aber von außen verwendet werden. Zusammen mit dem Schlüsselwort *static* lassen sich so bekannte Entwurfsmuster der Softwareentwicklung wie bspw. Singleton als ein Erzeugungsmuster (vgl. Bergmann 2005, S. 78) realisieren.

Mit Exceptions ist in PHP 5 eine Technik der Fehlerbehandlung eingeführt worden, die auch in Java implementiert ist. In einem vorgesehenen Bereich, dem try-Block, kann die Ausführung von Programmcode auf Fehler überprüft werden. Im Falle des Auftretens eines Fehlers kann ein Fehlerobjekt als Instanz der nativ in PHP definierten Exception- oder einer durch Vererbung abgeleiteten Klasse erzeugt und im catch-Block behandelt werden. PHP-Listing 3 zeigt eine solche Routine.

```
<?php
try{
    throw new Exception('In PHP-Listing 3 ist ein Fehler aufgetre-
ten!');
}
    catch(Exception $e){
        echo $e->__toString();
    }
?>
```

PHP-Listing 3: Exceptions in PHP 5

Dabei wird das Fehlerobjekt `$e` nur verarbeitet, wenn es eine Instanz/ein Objekt der Klasse `Exception` ist, was durch `class_type_hint` erreicht wird. Es ist auch möglich, mehrere `catch`-Blöcke untereinander zu definieren, um unterschiedliche Fehlerobjekte behandeln zu können.

Eine detaillierte Einführung in das Objektmodell und weitere Spezifika von PHP 5 bieten Gutmans, Bakken und Rethans in ihrem Werk „PHP 5 power programming“.

5.6.2 PHPTMAPI Kernpaket

Abbildung 23 zeigt die Interfaces und die abstrakte Klasse `TopicMapSystemFactory` des Kernpaketes (core) von PHPTMAPI und deren Hierarchie als UML Klassendiagramm zur Übersicht.

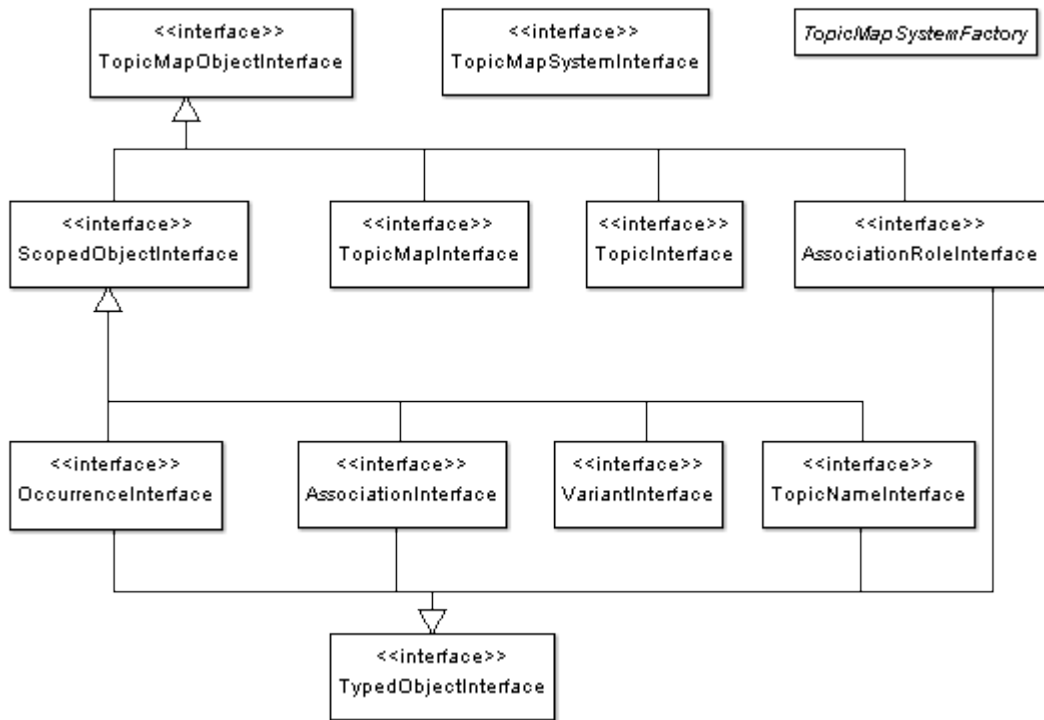


Abb. 23: PHPTMAPI Kernpaket (core)

Das vollständige Diagramm des Kernpakets mit allen verfügbaren Methoden kann in Anhang B nachgeschlagen werden.

Nachfolgend werden die Topic Maps-spezifischen Interfaces des Kernpaketes eingeführt.

TopicMapObjectInterface – Basisklasse für alle Objekte in einem PHPTMAPI-System, die Konstrukte einer Topic Map repräsentieren. Jedes Topic Map Objekt kann aus unterschiedlichen Quellen, z. B. XTM, generiert werden, die jeweils auf eine eigenständige Adresse verweisen können (vgl. TMAPI 2005b).

Funktionalität:

- Zugriff auf Identifizierungseigenschaften von Objekten,
- Definition von Identifizierungseigenschaften, um bspw. einen Verweis erstellen zu können,
- Entfernen von Identifizierungseigenschaften,
- verallgemeinerter Zugriff auf Elternklassen,
- Entfernen von Objekten aus dem System.

Über die Vererbungshierarchie (siehe Abb. 23 oder Anhang B) stehen diese Grundfunktionen allen folgenden Objekten zur Verfügung.

TopicMapInterface – Repräsentiert das Topic Map Konstrukt.

Grundlegende Funktionalität:

- Erstellen von Topics und Associations und Zugriff auf diese,
- Merging von Topic Maps.

XTM 1.0 DTD Element: <topicMap>

Entwurf XTM 2.0 DTD Element: <topicMap>

TopicInterface – Repräsentiert das Topic Konstrukt.

Grundlegende Funktionalität:

- Erstellen und Zugriff auf Occurrences und Topic Names,
- Erstellen und Entfernen von Subject Identifier und Subject Locator,
- Merging von Topics.

XTM 1.0 DTD Element: <topic>

Entwurf XTM 2.0 DTD Element: <topic>

AssociationRoleInterface – Repräsentiert das Rollenkonstrukt in Associations.

Grundlegende Funktionalität:

- Definition von Role Player und Role Type.

XTM 1.0 DTD Element: <member>

Entwurf XTM 2.0 DTD Element: <role>

ScopedObjectInterface – Basisklasse für alle Objekte, die Scope oder Parameter definieren.
Funktionalität: <ul style="list-style-type: none"> • Definition von Topics, die einen Scope oder Parameter bestimmen, • Zugriff auf Scope oder Parameter, • Löschen von Scope oder Parameter. Über die Vererbungshierarchie (siehe Abb. 23 oder Anhang B) stehen diese Grundfunktionen allen folgenden Objekten zur Verfügung.
XTM 1.0 DTD Element: <scope>, <parameters>
Entwurf XTM 2.0 DTD Element: <scope>

OccurrenceInterface – Repräsentiert das Occurrence Konstrukt. Eine Occurrence enthält einen Wert vom PHP Datentyp String ⁸ . Der XML Schema Datentyp bestimmt, ob diese Zeichenkette als Referenz auf eine externe Ressource zu interpretieren ist, also einen Locator einer bestimmten Notation repräsentiert.
Grundlegende Funktionalität: <ul style="list-style-type: none"> • Definition von Werten und Bestimmung durch XML Schema Datentypen.
XTM 1.0 DTD Element: <occurrence>
Entwurf XTM 2.0 Element: <occurrence>

TopicNameInterface – Repräsentiert das Topic Name Konstrukt.
Grundlegende Funktionalität: <ul style="list-style-type: none"> • Erstellen und Zugriff auf Variants, • Definition von Benennungen.
XTM 1.0 DTD Element: <baseName>
Entwurf XTM 2.0 DTD Element: <name>

VariantInterface – Repräsentiert das Variant Name Konstrukt. Ein Variant Name als Alternativbenennung eines Topic Names wird durch Parameter spezifiziert.
Grundlegende Funktionalität: <ul style="list-style-type: none"> • Definition von Werten und Bestimmung durch XML Schema Datentypen.
XTM 1.0 DTD Element: <variant>
Entwurf XTM 2.0 DTD Element: <variant>

⁸ In PHP müssen Variablen nicht deklariert werden (*loosely typed*), dies kann aber mit Cast Operatoren oder der Funktion `settype()` explizit erfolgen (vgl. Pushman 2000).

AssociationInterface – Repräsentiert das Association Konstrukt.
Grundlegende Funktionalität: <ul style="list-style-type: none"> • Erstellen und Zugriff auf Association Roles.
XTM 1.0 DTD Element: <association>
Entwurf XTM 2.0 DTD Element: <association>

Systemspezifisch und entsprechend nicht Bestandteil der dargestellten Hierarchie ist zum einen das `TopicMapSystemInterface`. Es stellt die Verbindung zu einem Topic Maps-System dar. Ein solches System muss eine oder auch mehrere Topic Maps verwalten können, die z. B. in einer Relationalen Datenbank vorgehalten werden (vgl. TMAPI 2005c). Das `TopicMapSystemInterface` bietet somit den Zugriff auf die vom System verwalteten Topic Maps. Daneben bietet es die elementare Funktionalität, eine Topic Map zu erstellen, wenn die Konfiguration dies zulässt. Das `TopicMapSystemInterface` erlaubt auch, Informationen über die Konfiguration des Systems zu erhalten.

`TopicMapSystemFactory` ist eine abstrakte Klasse und nimmt damit eine Sonderstellung in dem durch Interfaces beschriebenen Gerüst ein. Diese Klasse repräsentiert das Topic Maps-System in seiner grundlegenden Programmlogik, bspw. welches Datenmodell unterstützt wird (siehe Kapitel 5.5). In `TopicMapSystemFactory` wird entsprechend das Topic Maps-System konfiguriert. Instanzen bzw. Objekte von `TopicMapSystemInterface` implementierenden Klassen sollen aus diesem Grunde nicht mit dem `new`-Operator direkt erzeugt werden, sondern mit Hilfe von `TopicMapSystemFactory` (vgl. TMAPI 2005c).

Die Übernahme der Trennung des Systemzugriffs in `TopicMapSystem` und `TopicMapSystemFactory` von TMAPI macht auch für die Portierung nach PHP Sinn. `TopicMapSystemFactory` kapselt die Laufzeit-Logik (vgl. TMAPI 2005c) einer Implementierung von PHPTMAPI, z. B. welches Datenmodell unterstützt wird. Entsprechend erfolgt die Konfiguration eines Topic Maps-Systems und die Erzeugung einer Instanz von `TopicMapSystem` für den Zugriff auf dieses in `TopicMapSystemFactory`. Dies das System repräsentierende Objekt kann dann nicht mehr in seiner grundlegenden Einstellung bzw. Logik, Stichwort Datenmodell, geändert werden, was die konsistente Bearbeitung einer Topic Map in einem Programmablauf sicherstellt.

`TopicMapSystemFactory` enthält in der statischen Funktion `newInstance()` einen Algorithmus für die Rückgabe einer Instanz einer konkreten Umsetzung von `TopicMapSystemFactory` in einer Implementierung von PHPTMAPI, die entspre-

chend der Deklaration als *abstract* von dieser erbt (*extends*) und nicht implementiert (*implements*). Dabei benutzt `newInstance()` die statische Funktion `getClassFromFile()`, die als *private* deklariert ist und welche die eigentliche Prozedur zur Objekterzeugung übernimmt. Intention dieses einzigen realen Prozesses ist, dass Entwickler, die TMAPI- oder PHPTMAPI-Implementierungen benutzen, nur die API-Spezifikation ansprechen müssen und Abhängigkeiten und Zusammenhänge in Implementierungen nicht berücksichtigen brauchen.

Wie wird diese sog. „lookup procedure“ (TMAPI 2005c) realisiert? TMAPI definiert für Java verschiedene Stufen, die jeweils eine Alternative darstellen und in Reihe abgearbeitet werden. Dabei werden z. B. Einstellungen in *system property* oder Property-Dateien verwendet. Eine dritte, sehr einfache Alternative ist die, im Verzeichnis `META-INF/services` des aktuellen `CLASSPATH` einer Implementierung von TMAPI eine Datei `org.tmapi.core.TopicMapSystemFactory` zu erstellen, die einzig in der ersten Zeile den vollständigen Klassennamen der konkret umgesetzten, von `TopicMapSystemFactory` erbdenden Klasse enthält (vgl. TMAPI 2005c). Die TMAPI-Implementierung `tinyTIM` (<http://tinytim.sourceforge.net>) verweist zur Veranschaulichung auf `org.tinyTIM.TopicMapSystemFactoryImpl`.

Da PHP nicht kompiliert, sondern interpretiert wird, könnte der Prozess vereinfacht werden, da Pfad und Name der konkreten `TopicMapSystemFactory` vom jeweiligen Entwickler in den Quelltext der abstrakten `TopicMapSystemFactory` im PHPTMAPI-Verzeichnis editiert werden könnte. Der Eingriff in den Quellcode soll jedoch möglichst vermieden werden, da dies die Aufweichung des normativen Charakters von PHPTMAPI impliziert. Der Autor dieser Diplomarbeit hat für PHPTMAPI simulierend zu TMAPI folgende Prozedur festgelegt:

- Auf der gleichen Ebene eines Verzeichnisses, z. B. `phptmap1.0`, das die Verzeichnisse `org.phptmap1.core`, `org.phptmap1.index` und `org.phptmap1.index.core` mit dem Quellcode von PHPTMAPI 1.0 enthält, wird ein Verzeichnis `META` mit dem Unterverzeichnis `factory` angelegt;
- In diesem Verzeichnis `factory` wird eine ASCII-Datei ohne Dateiendung angelegt, deren Name der konkreten `TopicMapSystemFactory`-Klasse entspricht, z. B. `TopicMapSystemFactoryImpl`;

- Diese Datei enthält einzig in der ersten Zeile ohne Leer- und Steuerzeichen den Pfad zur TopicMapSystemFactory-Klasse, z. B. `/phptmapi/src/phptmapi/core/TopicMapSystemFactoryImpl.class.php`;
- Auf der Ebene der Verzeichnisse `META` und, wie im Beispiel, `phptmapi1.0` wird zuletzt das Verzeichnis angelegt, in dem die Implementierung benutzt werden soll, z. B. `tmodel`.

PHP-Listing 4 zeigt nun, wie im Verzeichnis `tmodel` eine Implementierung von PHPTMAPI über die Einbindung der abstrakten Klasse `TopicMapSystemFactory` verwendet wird:

```
<?php
require_once($_SERVER['DOCUMENT_ROOT'].'/quaaxtm/phptmapi1.0/org.phptmapi.core/TopicMapSystemFactory.abstract.php');
$tmSystemFactory = TopicMapSystemFactory::newInstance();
$tmSystem = $tmSystemFactory->newTopicMapSystem();
?>
```

PHP-Listing 4: Einbindung einer PHPTMAPI-Implementierung

`TopicMapSystemFactory::newInstance()` gibt die konkrete `TopicMapSystemFactory` zurück. Entwickler arbeiten ab diesem Aufruf mit der Implementierung und können, wie im Beispiel gezeigt, ein `TopicMapSystem`-Objekt erzeugen.

Dies zeigt, wie bereits erwähnt, dass Entwickler, die die Implementierung benutzen, lediglich die Abhängigkeit zu PHPTMAPI, der Schnittstellendefinition, beachten brauchen. Hierin ist das Bestreben zur Vereinheitlichung der Benutzung von PHPTMAPI-Implementierungen zu erkennen.

Die PHPTMAPI-Implementierung kann unter der Berücksichtigung von Zugriffsrechten beliebig auf einem Server positioniert werden und definiert eigenständig Abhängigkeiten zu anderen Klassen der Implementierung und weiteren Dateien, die bspw. PHPTMAPI enthalten. Da `getClassFromFile()` die native Funktion `opendir()` benutzt, kann die Implementierung prinzipiell unter einem beliebigen URL abgelegt sein. Hierbei sind ebenfalls Zugriffsrechte und Servereinstellungen wie z. B. *directory listing* sowie die Beschränkungen auf die Zugriffsmechanismen `ftp://` und `file://` zu beachten (vgl. PHP Group 2006a). Dieser an eine Datei gebundene Prozess über die Verzeichnisstruktur `META/ein_Unterverzeichnis`, der für `TopicMapSystemFactory` dargelegt wor-

den ist, erlaubt weiterhin ein gleichartiges Verfahren für die Einbindung von Erweiterungen von PHPTMAPI. Dies wird in Kapitel 5.6.4 exemplarisch für den Index beschrieben.

Die unterschiedlichen Exceptions in PHPTMAPI zur spezifischen Behandlung von Fehlern sind konkrete Klassen. Also können diese direkt in PHPTMAPI-Implementierungen verwendet werden. Auf der ersten Hierarchiestufe sind `TMAPIRuntimeException` und `TMAPIException` von der nativ in PHP verfügbaren Klasse `Exception` abgeleitet. Im jeweiligen Konstruktor wird der Konstruktor von `Exception`, der Elternklasse, über `parent::__construct()` gerufen. Somit übernehmen `TMAPIRuntimeException` und `TMAPIException` für die spezifischen Exceptions eine Vermittlungsfunktion, um mit `Exception` arbeiten zu können. Abbildung 24 zeigt die beiden Hierarchien für `TMAPIRuntimeException` und `TMAPIException`.

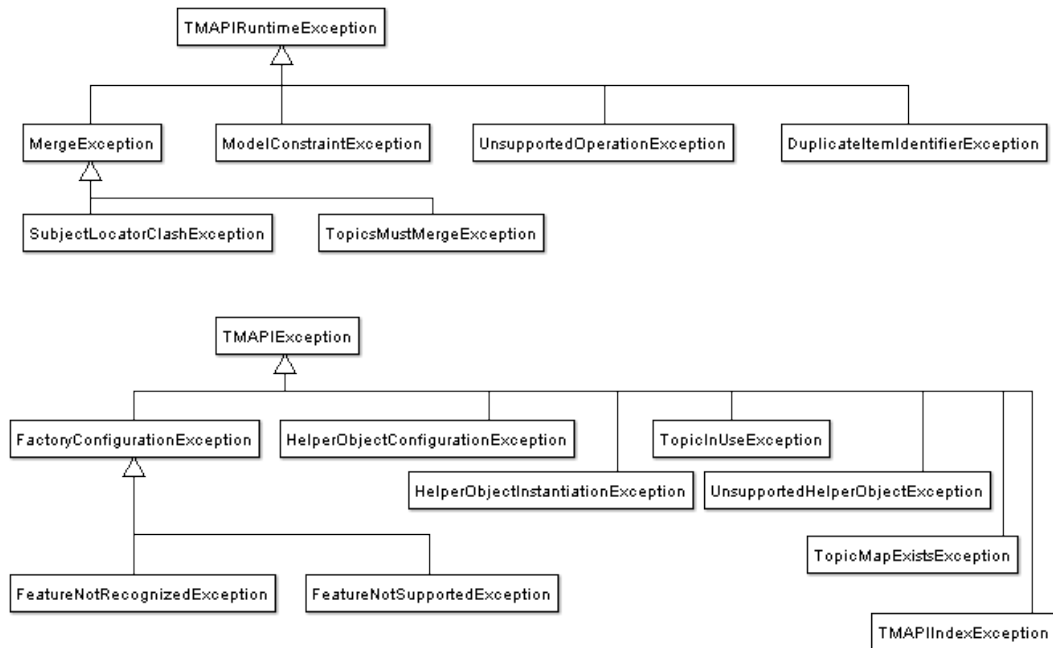


Abb. 24: PHPTMAPI Exceptions

Das vollständige Diagramm mit allen verfügbaren Methoden ist in Anhang C aufgeführt werden.

Im Sinne größtmöglicher funktionaler Vergleichbarkeit zu TMAPI ist die Struktur der Exceptions inklusive ihrer Benennungen nahezu identisch übernommen worden. Ein Unterschied zu TMAPI besteht darin, dass PHPTMAPI eine Klasse `UnsupportedOperationException` zur Verfügung stellt (siehe Abb. 24), mit der Ausnahmen behandelt werden können, die dann abzufangen sind,

wenn eine Operation durchgeführt werden soll, die mit dem TMDM konform ist, das PHPTMAPI-System jedoch für die Verarbeitung des für XTM 1.0 definierten Meta- und Datenmodells konfiguriert ist. Damit haben die Autoren auf die aktuellen Entwicklungen im Standardisierungsprozess für das Topic Maps Data Model reagiert.

5.6.3 Unterschiede zu TMAPI

PHPTMAPI orientiert sich deutlicher am Topic Maps Data Model. Gegenüber TMAPI weist PHPTMAPI dahingehend folgende Änderungen auf:

- Umbenennung von `SourceLocator` in `ItemIdentifier`, die den eindeutigen Verweis auf ein Topic Maps-Konstrukt innerhalb eines Systems ermöglichen;
- Einführung einer Methode `setReifier()` in den Interfaces `AssociationInterface`, `AssociationRoleInterface`, `OccurrenceInterface`, `TopicNameInterface`, `TopicMapInterface` und `VariantInterface` für einen Reification-Prozess;
- Einführung von XML Schema Datentypen, die die Identifizierung und korrekte Interpretation von String-Werten in den Value-Eigenschaften in `OccurrenceInterface` und `VariantInterface` ermöglichen.

Weiter ist ein `TypedObjectInterface` als Hilfskonstruktion eingeführt (siehe Abb. 23 oder Anhang B), von dem alle typisierbaren Interfaces die Methoden `getType()` und `setType()` erben. Dies verbessert zum einen die Übersicht im Anwendungsdesign (siehe auch Kapitel 5.6.1), zum anderen kann so in einer konkreten Implementierung bzw. Anwendung durch ein Array von Objekten spezifisch nach Klassenzugehörigkeit iteriert werden, wodurch innerhalb einer Schleife folgende Routine möglich wäre:

```
<?php
if($obj instanceof TypedObjectInterface){
    $type = $obj->getType();
}
?>
```

Es ist diskutiert worden, mit der gleichen Absicht auch ein `ReifiableObjectInterface` mit den Methoden `getReifier()` und `setReifier()` umzusetzen (und diese entsprechend aus den Interfaces herauszuziehen) und reifizierbare Objekte von diesem Interface erben zu lassen. Mit der Ab-

sicht einer schnelleren Veröffentlichung ist dies in Version 1.0 von PHPTMAPI jedoch noch nicht berücksichtigt worden.

PHPTMAPI enthält im Gegensatz zu TMAPI kein Locator-Interface mehr, das dort ein Adressierungsschema repräsentiert. In PHPTMAPI wird ein Locator immer durch einen Wert vom Datentyp String beschrieben. In der Web-Entwicklung werden hauptsächlich URLs verarbeitet, so dass die Notation, URI, bekannt und eine Repräsentation eines Locators durch eine Zeichenkette ausreichend ist. Durch diese Maßnahme konnte Komplexität vermindert werden, da neben der Reduzierung um ein Interface das gesamte Locator-Handling in PHPTMAPI auf das Verarbeiten von Zeichenketten beschränkt werden kann, wofür PHP zahlreiche Funktionen bereitstellt.

Jedes Topic Maps-spezifische Interface des Kernpakets ist um eine Methode `getParent()` erweitert, um die jeweilige Elternklasse ermitteln zu können. So muss ein Entwickler nicht mehr wissen, von welcher Klasse ein Objekt instanziiert worden ist, um die spezifische Methode, z. B. `getAssociation()` in `AssociationRoleInterface`, für das Aufrufen des Elternobjekts verwenden zu können, was ebenfalls eine Vereinfachung darstellt.

Eine grundlegende Veränderungsmaßnahme beinhaltet die Transferierung der Methoden `getTopicBySubjectIdentifier()` und `getTopicBySubjectLocator()` aus dem `TopicsIndexInterface` des Indexpaketes (siehe Kapitel 5.6.4) in das `TopicMapInterface`. Beide Methoden bieten eine primäre Zugriffsmöglichkeit auf Topics. Für einen Zugriff auf diese müssen Anwender nun nicht zunächst ein `TopicsIndex`-Objekt erstellen, sondern können mit dem Basisobjekt, das eine Topic Map repräsentiert, auf die genannten Funktionalitäten zugreifen. Dies befreit Entwickler zudem davor, neben dem Kernpaket auch das Indexpaket bzw. Teile davon implementieren zu müssen, was den Umfang und Aufwand einer PHPTMAPI-Implementierung reduziert.

Ebenso ist die Methode `getObjectByItemIdentifier()` aus dem Interface `TopicMapsObjectIndex`, das nicht mehr in PHPTMAPI implementiert ist (siehe Kapitel 5.6.4), nun im `TopicMapInterface` verfügbar. Die Intention entspricht der im vorigen Abschnitt genannten. Mit `getObjectByItemIdentifier()` steht dabei ein allgemeiner Zugriff auf Topic Maps-Konstrukte repräsentierende Objekte in einem PHPTMAPI-System zu Verfügung. Die durch diese Umstellungen beabsichtigten Vereinfachungen haben sich in der in Kapitel 5.7 beschriebenen Implementierung bestätigt.

Für die Nennung der Unterschiede der Spezifikationen bleibt zu erwähnen, dass Java und PHP Differenzen im Bereich der Datentypen und -strukturen aufweisen. Java definiert für Datenstrukturen im Collection-Framework unterschiedliche Möglichkeiten. So können bspw. Collection, Set (s. o.) oder List für eine Reihe/Sammlung von Datenelementen verwendet werden. PHP stellt lediglich Array zur Verfügung. Entsprechend sind in PHPTMAPI alle Datenstrukturen Arrays.

5.6.4 Basiserweiterung: PHPTMAPI Indexpaket

Abbildung 25 zeigt die Interfaces des Indexes eines PHPTMAPI-Systems und ihre Hierarchie zur Übersicht. Das Indexpaket stellt ebenso wie in TMAPI eine Basiserweiterung von PHPTMAPI dar.

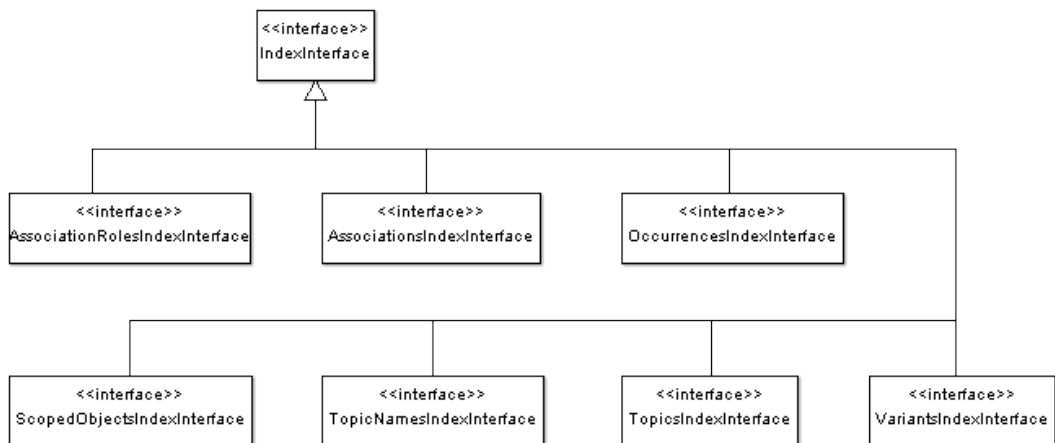


Abb. 25: PHPTMAPI Indexpaket (index, index.core)

Das vollständige Diagramm des Indexpakets mit allen verfügbaren Methoden kann in Anhang D nachgeschlagen werden.

Das zweiteilige Indexpaket setzt sich aus dem `IndexInterface`, das systemspezifische Methoden wie `open()` oder `close()` bereitstellt (index) sowie den Topic Maps spezifischen Index Interfaces, die den Index einer Topic Map repräsentieren (index.core), zu einer Hierarchie zusammen (siehe Abb. 25).

Auf eine Einführung der Interfaces soll hier verzichtet werden, da die Grundfunktionalität des Indexpaketes, nämlich einen Index zur Verfügung zu stellen, deutlich ist und die Methoden der einzelnen Interfaces, wie sie in Anhang D nachgesehen werden können, die individuelle Funktionalität eindeutig kenn-

zeichnen. Im folgenden werden die Änderungen gegenüber TMAPI genannt und erläutert.

Das Indexpaket enthält für den systemspezifischen Bestandteil nur noch ein `IndexInterface` und kein Interface `IndexFlags` mehr, dessen Methode `isAutoUpdated()` ist nun im `IndexInterface` definiert. Das Interface `TopicMapsObjectIndex` ist ebenfalls aus dem Indexpaket entfernt worden, dessen einzige Methode `getTopicMapObjectBySourceLocator()`, die TMAPI vorgibt, ist zunächst in `getObjectByItemIdentifier()` umbenannt worden und wird aktuell im `TopicMapInterface` deklariert.

Für die Umsetzung dieser Maßnahmen ist zu bemerken, dass die Autoren von PHPTMAPI der Meinung sind, dass Interfaces, die lediglich eine Methode deklarieren, unnötigen Aufwand in der Implementierung verursachen und deshalb eine Auflösung in andere Interfaces sinnvoll ist, sofern die Struktur der Spezifikation berücksichtigt bleibt (z. B. Vererbungshierarchien).

Insgesamt ist das Indexpaket gegenüber TMAPI also für eine einfachere und intuitivere Benutzung im Umfang reduziert. Dies zeigt sich darin, dass alle Interfaces von `index.core` in der angeglichenen Anzahl an Methoden, wie erwähnt, primär gleichartige Funktionen definieren, bspw. die Rückgabe aller Objekte eines bestimmten Typs wie exemplarisch `getTopicsByType()` in `TopicsIndexInterface`.

Der Index wird als sog. konfigurierbares Hilfsobjekt verwendet. PHPTMAPI definiert wie TMAPI das Interface `TopicMapSystemConfigurableHelperObject`. Dieses systemspezifische Interface ist in der Java-Implementierung innerhalb des `TopicMapSystem`-Interface deklariert (*inner class nesting*). In PHP ist es jedoch nicht möglich, Klassen in Klassen zu definieren, weswegen PHPTMAPI zwangsläufig ein eigenständiges Interface bereitstellt. Das `IndexInterface` erbt von `TopicMapSystemConfigurableHelperObjectIInterface`, das die Methode `configure()` enthält. In dieser Funktion werden Hilfsobjekte für die Topic Map konfiguriert, für die sie, wie der Index, zusätzliche Funktionalitäten zur Verfügung stellen. Diese Methode ist vor dem Aufruf anderer öffentlicher Methoden mit Ausnahme des Konstruktors zu rufen (vgl. TMAPI 2005c).

Ein Hilfsobjekt wird über die Methode `getHelperObject()` erzeugt, die im `TopicMapInterface` definiert ist. Über die Parameter `$implementationDirectoryName` und `$implementationClassName` wird die in Kapitel 5.6.2 dargelegte Auffindungsprozedur (*lookup procedure*) ermöglicht. So

bezeichnet `$implementationDirectoryName` das Unterverzeichnis im Verzeichnis `META`, in dem `$implementationClassName` die ASCII-Datei mit dem Namen der konkreten Klasse kennzeichnet, von der das Hilfsobjekt erzeugt wird. So könnten bspw. die Parameter `$implementationDirectoryName = 'index'` und `$implementationClassName = 'IndexImpl'` aufnehmen, die folglich im Verzeichnis `META` auf das Unterverzeichnis `index` mit der Datei `IndexImpl` verweisen, in der nach den oben genannten Bedingungen der Pfad zur Implementierung des `Index` Interfaces festgelegt ist. Nach diesem Schema können für PHPTMAPI-Implementierungen beliebig viele Hilfsobjekte eingebunden werden.

5.6.5 Bewertung und Ausblick

Die Motivation für die Portierung der TMAPI Spezifikation von Java nach PHP 5 ist angerissen worden. Welche Wirkungen kann dieses Projekt auf die Topic Maps- und die PHP-Community haben? Der Autor hat PHPTMAPI 1.0 und in diesem Zusammenhang „QuaaxTM“ (<http://quaaxtm.sourceforge.net>) als eine erste Implementierung des Kernpaketes, die im folgenden Kapitel vorgestellt wird, im Rahmen der Postersession auf der „Conferences on Topic Maps Research and Applications“, kurz TMRA (<http://www.tmra.de>), am 11.10.2006 in Leipzig einem Fachpublikum vorgestellt. Das Poster ist als Anhang E dieser Arbeit beigelegt. Ein Kernpunkt der kurzen Präsentation betraf die Äußerung der Ansicht, dass ein standardisiertes API eine Basis dafür ist, Mitglieder der aktiven und praxisorientierten PHP-Community, sourceforge.net bspw. registriert knapp 16000 PHP-Projekte (Stand Oktober 2006), an Topic Maps als noch recht junge, aber sehr mächtige Technologie im Bereich der Informationsstrukturierung, Wissensrepräsentation und Ontologien heranzuführen und letztlich auch zu gewinnen. Eine Technologie kann sich langfristig nur durchsetzen, wenn sie praktisch, d. h. mit Software angewendet werden kann.

In der Resonanz ist deutlich geworden, dass die Verbreitung von Topic Maps im allgemeinen und für den Bereich der populären Skriptsprache PHP im speziellen Bedeutung zugemessen wird. Jack Park, Herausgeber des bereits zitierten Werkes „XML Topic Maps – Creating and Using Topic Maps for the Web“ (Park/Hunting 2003) und Gründungsmitglied von TopicMaps.org, verwies in diesem Zusammenhang auf die Bedeutung der im Internet weit verbreiteten Linux-Apache-MySQL-PHP-Architektur, kurz LAMP. Ganz allgemein

spielen Skriptsprachen wie Perl, Python, PHP u. a. im Kontext „Web 2.0“ eine große Rolle (vgl. O'Reilly 2005, S. 4). Die Grundlage für diese Aussage mag darin liegen, dass der zeitliche Aufwand für permanente Änderungen und Anpassungen mit Skriptsprachen wesentlich geringer ist als bspw. in einem JSP-Framework. Eine konkrete Verknüpfung von Topic Maps und Tagging als „Web 2.0“ kennzeichnender Trend wird als exemplarischer Praxisfall noch betrachtet und ist auch im Rahmen der Präsentation auf der TMRA 2006 angedeutet worden.

Auf höherer Abstraktionsebene geht es letztendlich um die Möglichkeit zur Standardisierung von Metadaten bzw. Metadatenstrukturen und die Möglichkeit zum Etablieren von Semantik für eine Vielzahl populärer Anwendungen, die eben in PHP geschrieben sind. Beispielhaft seien WordPress, PHP-Nuke, Movable Type, ExpressionEngine, Typo 3, phpBB oder moodle genannt. Somit kann durchaus behauptet werden, dass das Projekt „PHPTMAPI“ als Türöffner bzw. *enabler* und in dieser Funktion als eine Basisschnittstelle für das Interaktionspotential zwischen den beiden Technologien und den dahinter stehenden Communities angesehen werden kann.

Im Hinblick auf die konkrete Portierung, PHPTMAPI 1.0, bleibt zu erwähnen, dass die Entwicklung zum einen durch eine starke Orientierung an der TMAPI-Spezifikation geprägt war. Grund dafür war ein nur grundlegendes Verständnis für das Topic Maps Data Model. Dieser Umstand hat sich nun mit einer eingehenden Beschäftigung mit dem Datenmodell zu Gunsten einer fundierteren Kenntnis geändert. Andererseits befindet sich die Technologie, Topic Maps, speziell auf Syntax- und Modellebene noch in einem Standardisierungsprozess, wenn auch in der Endphase. Einzelne Aspekte sind angesprochen worden. Deshalb konnten und mussten neue Ergebnisse abweichend von der Vorbilddefinition berücksichtigt werden.

So ist festzuhalten, dass PHPTMAPI in Bezug auf PHP und das PHP-typische Einsatzumfeld unter Berücksichtigung der Konformität zum TMDM noch einmal analysiert und ggf. auch verändert werden sollte. Konkrete Rückschlüsse kann eine Implementierung von PHPTMAPI 1.0 liefern. Eine solche wird im anschließenden Kapitel vorgestellt. Es ist davon auszugehen, dass PHPTMAPI weiter vereinfacht werden kann. Eine derartige Entwicklung hätte zusätzliche positive Auswirkungen für die Verbreitung von Topic Maps in der PHP-Community.

Für das grundlegende Design von PHPTMAPI ist zu sagen, dass die erwähnte Einführung eines `ReifiableObjectInterface` äquivalent zum `TypedObjectInterface` sinnvoll hinsichtlich einer verbesserten Übersicht ist.

Es sei zuletzt auf eine fehlerhafte Definition in PHPTMAPI 1.0 hingewiesen. Die Methode `getReified()` in `TopicInterface` definiert als Rückgabe ein Array, das Instanzen von `TopicMapObjectInterface` enthält (siehe Verzeichnis `phptmap1.0/src/org.phptmap1.core` auf der beigelegten CD-ROM). Dies entspricht der Definition in TMAPI (vgl. TMAPI 2005b). Das TMDM jedoch sieht vor, dass ein Topic Maps-Konstrukt bis auf Topic nur von einem Topic „reifiziert“ werden kann (vgl. ISO/IEC 13250-2). Insofern muss die Rückgabe von Array auf `TopicMapObjectInterface` korrigiert werden.

In der Betrachtung dieser Feststellungen wird deutlich, dass die Berechtigung für eine überarbeitete Version von PHPTMAPI gegeben ist.

5.7 Implementierung von PHPTMAPI

Für die Implementierung eines Topic Maps-Tagging-Modells wird eine Software, die in PHP geschrieben ist, benötigt, die die Erstellung und Manipulation von Topic Maps erlaubt. Eine prototypische Anwendung des Modells mit Software hilft dabei neben der alleinigen Demonstration zweitens zugleich, die Überprüfung der Anwendbarkeit und Praxistauglichkeit des theoretischen Modells vorzunehmen sowie drittens eine realitätsnahe Skizze eines möglichen Einsatzszenarios und -kontextes entwerfen zu können. Insofern trägt die Implementierung zur Lösung von drei Aufgabenstellungen bei.

In der Betrachtung der vorangegangenen Arbeitsergebnisse läuft eine Schlussfolgerung darauf hinaus, dass eine solche Software eine Implementierung von PHPTMAPI, genauer: des PHPTMAPI 1.0 Kernpaketes, sein sollte. Insofern kann im vorigen Kapitel zur Bewertung hinzugefügt werden, dass PHPTMAPI die Voraussetzung für eine Implementierung und letztlich auch zur Zielerreichung ist. Eine derartige Implementierung ist zum aktuellen Zeitpunkt, Dezember 2006, weder unter einer kommerziellen noch unter einer Open Source-Lizenz zu erhalten, eine Tatsache, die angesichts der noch sehr jungen Historie von PHPTMAPI jedoch zwangsläufig zu sein scheint. Deshalb wird eine Implementierung zum Einsatz kommen, die der Autor dieser Arbeit entwickelt hat.

Hierbei handelt es sich um die Implementierung der Interfaces in der Struktur (Hierarchie), wie sie PHPTMAPI 1.0 für das Kernpaket vorgibt. Der Entwicklungsstatus dieser Software ist, etwas pessimistisch ausgedrückt, als Alpha-Status oder als der eines Prototypen einzustufen. Diese Einschätzung rührt vor allem daher, dass diese Implementierung für eine erste Veröffentlichung gerade fertiggestellt und bis dato lediglich durch den Autoren und von Lars Heuer getestet worden ist. Die Bewährung in intensivem bzw. Alltagsgebrauch konnte zwangsläufig noch nicht erfolgen.

Über die Maßnahme, die Implementierung als Projekt „QuaaxTM“ auf der Open Source-Plattform sourceforge.net mit der Projektseite <http://quaaxtm.sourceforge.net>, die Informationen für Interessierte bereithält sowie auf den Downloadbereich verweist, zu registrieren, könnte die Beseitigung dieses Zustandes zeitnaher erfolgen. Diese initiale 0.1-Veröffentlichung enthält

- die PHP-Quellen der Implementierung und die SQL-Quellen zum Aufbau der Datenbanktabellen,
- die PHP-Quellen von PHPTMAPI 1.0,
- die mit phpDocumentor erstellte API-Dokumentation des Quellcodes der Implementierung,
- eine Installationsanleitung in englischer Sprache.

QuaaxTM steht unter einer Open Source-Lizenz, der GNU Lesser General Public License, kurz LGPL (<http://www.gnu.org/licenses/lgpl.html>), zur Verfügung. Sie basiert auf der GNU General Public License, GPL, und definiert wie diese, kurz zusammengefasst, folgende zentrale Charakteristika:

- Die Ausführung der Software unterliegt keiner Beschränkung.
- Die Software darf kopiert, verbreitet und modifiziert werden; die Quellen von Kopien und Bearbeitungen müssen jedoch wieder unter den gleichen Lizenzbedingungen verfügbar sein.
- Der Urheber ist von Haftungs- und Gewährleistungsansprüchen befreit (vgl. Free Software Foundation, Inc. 2006).

Es bleibt festzuhalten, dass die GPL in Berücksichtigung US-amerikanischen Urheberrechts formuliert worden ist, die Kernpunkte sich aber auch auf das deutsche Urheberrecht abbilden lassen (vgl. Grassmuck 2004, S. 286).

Die Motivation, die Implementierung Open Source gemäß der LGPL bereitzustellen, entspricht der, die in Kapitel 5.6 im Zusammenhang mit PHPTMAPI bereits genannt worden ist. Die potentiell weite Verbreitung kann sich positiv auf den folgenden Entwicklungsprozess auswirken, da das Erkennen und Registrieren von Fehlern auf mehrere Anwender verteilt wird. Und dadurch, dass eine Beteiligung Dritter am Projekt ausdrücklich erwünscht ist, kann die Weiterentwicklung und Optimierung der Anwendung langfristig einem Kollektiv übertragen werden. Denn es muss resümiert werden, dass diese Implementierung erst am Anfang einer notwendigen Weiterentwicklung steht und keine stabile Software, im Sinne von ausgereift, darstellt. Deshalb wird die Vorstellung der Implementierung auf einige ausgewählte Merkmale, die die persistente Speicherung von Daten betreffen (sich also kurzfristig nicht ändern werden), begrenzt.

5.7.1 Anforderungen an eine Implementierung und resultierende Ergebnisse

Die Realisierung der Implementierung ist parallel zur Fertigstellung von PHPTMAPI begonnen worden. Ein Ziel war unter anderem, PHPTMAPI auf seine Praxistauglichkeit zu überprüfen. Insofern konnte der o. g. Aspekt einer quasi Machbarkeitsprüfung in einem ersten Ansatz bereits berücksichtigt werden. Ein erstes Ergebnis wird in der Bewertung in Kapitel 5.7.3 genannt. Eine diesbezügliche Vermutung ist im Zusammenhang mit der Bewertung von PHPTMAPI auch bereits angemerkt worden. Daraus lässt sich als grundsätzliche Anforderung ableiten, dass die Anwendung die Spezifikation möglichst in ihrer Gesamtheit abbilden sollte, mindestens aber das Kernpaket implementieren muss. Für die detaillierteren Anforderungen lassen sich zwei Dimensionen unterscheiden. Zum einen sind die Anforderungen an Eigenschaften (*features*) der Anwendung an sich zu klären, zum anderen ist festzulegen, welche Technik bzw. Technologie zur persistenten Speicherung verwendet werden soll. Konkret: Wie wird eine Topic Map gespeichert?

Für die Eigenschaften einer PHPTMAPI-Implementierung können wiederum zwei Dimensionen unterschieden werden. Wesentlich sind diejenigen Basiseigenschaften, die TMAPI als Spezifikation vorgibt und welche auch von PHPTMAPI 1.0 übernommen worden sind. Diese Eigenschaften bzw. *features* sind indirekt bereits mehrfach im Zusammenhang mit der Konfiguration eines

Topic Maps-Systems und einmal konkret mit den Konfigurationsparametern für TMAPI-implementierende Anwendungen bzgl. des Datenmodells angesprochen worden. Solche Eigenschaften werden über sog. „feature strings“ (TMAPI 2005d), die URLs darstellen, determiniert und in der „feature string registry“ (TMAPI 2005d) definiert, wobei der URL auf diese Definition verweist. Tabelle fünf zeigt die aktuellen *feature strings* von TMAPI und PHPTMAPI, die die einzelnen Eigenschaften repräsentieren.

<i>feature string</i> (Eigenschaft)	Bedeutung
http://tmapi.org/features/automerger	Unterstützung für automatisches Merging von Topics gemäß der Vorgaben/Einstellungen unter http://tmapi.org/features/merge (z. Zt. lediglich http://tmapi.org/features/merge/byTopicName , s. u.)
http://tmapi.org/features/model/xtm1.0	Unterstützung des in der XTM 1.0 Spezifikation definierten Datenmodells
http://tmapi.org/features/model/xtm1.1	Unterstützung des in ISO 12350 definierten Topic Maps Data Model (TMDM)
http://tmapi.org/features/merge/byTopicName	Unterstützung für Registrierungsprozess, ob zwei Topics identische Topic Names aufweisen und ggf. automatisches Merging durchzuführen ist (s. o.)
http://tmapi.org/features/notation/URI	Unterstützung der URI Adressnotation
http://tmapi.org/features/readOnly	Unterstützung für Nur-Lese oder Lese-und-Schreib-Zugriff auf eine Topic Map

Tab. 5: TMAPI *feature strings* und ihre Bedeutung (vgl. TMAPI 2005d)

Für die Unterstützung des automatischen Mergings ist anzumerken, dass der oben geschilderte Modus, Automerging abhängig von Einstellungen unter <http://tmapi.org/features/merge> (aktuell lediglich <http://tmapi.org/features/merge/byTopicName>) zu machen, vom Autoren dieser Diplomarbeit und der hier vorgestellten PHPTMAPI-Implementierung nicht für sinnvoll erachtet wird. Automerging sollte davon abhängig betrachtet werden, welches Datenmodell unterstützt wird. Die jeweilige Spezifikation des Datenmodells definiert, wann Topics identisch sind (in der Betrachtung ihrer *properties*). Die Bedingungen, die das TMDM für Topic Items definiert, sind auf Seite 77 aufgeführt. Das namensbasierte Merging ist im TMDM nicht vorgesehen, sondern allein im Datenmodell für XTM 1.0 (siehe <http://www.topicmaps.org/xtm/index.html#desc-merging>). Eine Implementierung, die das TMDM unterstützt, kann somit gleichzeitig kein namensbasiertes Merging unterstützen. Also ist eine Plausibilitätsprüfung in der Eigenschaftenkonfiguration zu empfehlen.

Der jeweilige Konfigurationsmechanismus in einer Implementierung der TopicMapSystemFactory legt über *true* (ja) und *false* (nein) fest, ob eine Eigenschaft für eine Instanz einer TopicMapSystem-Implementierung zum einen verfügbar ist bzw. unterstützt wird und zum anderen konfigurierbar, d. h. einstellbar ist. Tabelle sechs zeigt nun die Einstellungen und die Konfigurationsfähigkeit für die hier vorgestellte Implementierung und spiegelt dabei über die Unterstützung oder Nicht-Unterstützung einer Eigenschaft zugleich die Anforderungen hinsichtlich der Basiseigenschaften wider.

Eigenschaft	Unterstützung	konfigurierbar
http://tmapi.org/features/automerge	ja	nein
http://tmapi.org/features/model/xtm1.0	nein	nein
http://tmapi.org/features/model/xtm1.1	ja	nein
http://tmapi.org/features/merge/byTopicName	nein	nein
http://tmapi.org/features/notation/URI	ja	nein
http://tmapi.org/features/readOnly	nein	nein

Tab. 6: Einstellungen für die Basiseigenschaften der PHPTMAPI-Implementierung

Es ist festzustellen, dass die Implementierung keine Konfiguration zulässt und quasi maßgeschneidert für die Anforderungen des Autors hinsichtlich der Lösung der Aufgaben in der Diplomarbeit ist. Dabei wird allein das Topic Maps Data Model unterstützt, was die Zukunftsfähigkeit der Implementierung sichert, allerdings hinsichtlich der Serialisierung in XTM-Syntax im Moment, Dezember 2006, noch zu Schwierigkeiten führt, da XTM 2.0 lediglich als Entwurf vorliegt⁹.

Für den angesprochenen Aspekt des Mergings ist weiter zu erkennen, dass automatisches Merging gemäß der im TMDM definierten und auf S. 77 genannten Bedingungen realisiert wird. Im Hinblick auf die Verbreitung der Anwendung ist es langfristig jedoch wünschenswert, dass die Eigenschaftensunterstützung individuell zu konfigurieren sein kann.

Durch das Merging soll das Auftreten redundanter Topic Maps-Konstrukte in einer Topic Map vermieden werden (vgl. ISO/IEC 13250-2). Neben dem geschilderten automatischen Verschmelzen von Topics, wodurch die Repräsen-

⁹ Wie in persönlichen Gesprächen auf der TMRA 2006 zu vernehmen war, ist es sehr wahrscheinlich, dass der letzte o. g. Entwurf vom 19.6.2006 die endgültige Fassung von XTM 2.0 sein wird.

tation eines Aussagegegenstandes durch ein Topic erreicht werden soll, ist zusätzlich zu gewährleisten, dass im Erstellungsprozess über create()- und add()-Methoden von Topic Names, Occurrences, Subject Locators und Subject Identifiers als Eigenschaften (*properties*) von Topics Redundanzen über Prüfalgorithmen ebenfalls vermieden werden, so dass möglichst keine zwei oder mehrere identische o. g. Topic Maps-Konstrukte existieren. Ein solches Verfahren sichert die inhaltliche Konsistenz einer Topic Map und vermeidet die Beanspruchung zusätzlicher unnötiger Speicherressourcen.

Steven Newcomb, Mitherausgeber der ISO 10744:1992 und 10744:1997 HyTime und des 13250 Topic Maps-Standards sowie Mitgründer von TopicMaps.org und Mitherausgeber der XTM-Spezifikation (vgl. Park/Hunting 2003, S. XXVII) hat in seiner Grundsatzrede auf der TMRA 2006 noch einmal darauf hingewiesen, dass die Repräsentation eines Aussagegegenstandes durch ein Topic einen zentralen Standpunkt der Topic Maps-Community darstellt (vgl. Newcomb 2006, S. 2).

Für den Aspekt der persistenten Speicherung im allgemeinen ist eine Anforderung in der Betrachtung aus zwei Blickwinkeln zu formulieren. Zum einen kann der Entwickler, da er die erste Veröffentlichung allein erarbeitet, ausschließlich seine Kompetenzen berücksichtigen, zum anderen ist er zwangsläufig darauf angewiesen, welche Möglichkeiten PHP bietet, auf persistent gespeicherte, als Topic Map strukturierte Daten zuzugreifen und sie zu manipulieren (letzte Forderung ergibt sich aus der in Tabelle 6 für die über <http://tmapi.org/features/readOnly> repräsentierte Eigenschaft definierte Anforderung, einen Lese- und Schreibzugriff auf eine Topic Map zu realisieren).

Die andere Determinante betrachtet die allgemeine Verfügbarkeit einer Architektur bzw. Technologie, wenn diese Implementierung Open Source als QuaaxTM einer Öffentlichkeit zur Verfügung gestellt werden soll, woraus sich für die Weiterentwicklung wiederum o. g. positive Rückwirkungen ergeben können. Hohe Verfügbarkeit und Popularität implizieren i. a. auch eine weite Verbreitung von Kompetenz im Umgang mit einer Technologie. Eine exotische und/oder komplexe Softwarekonstellation ist der Verbreitung der Implementierung als QuaaxTM nicht dienlich. Die Erwägung, auf Speicherungsebene ein *framework* wie bspw. Propel (<http://propel.phpdb.org/trac/>) einzusetzen, mit dem Object-Relational Mapping (ORM) realisiert werden kann und das es letztendlich ermöglicht, ein Relationales Datenmodell bzw. Relationale Daten-

strukturen über Objekte anzusprechen, ist aus diesem Grunde abgewiesen worden.

Für den Gesichtspunkt der Performanz bzw. Leistung ist der Standpunkt von Fuecks zu teilen, dass Leistung im wesentlichen eine Frage der Leistungsfähigkeit der Hardware ist (vgl. Fuecks 2005, S. 1). Wie stellt sich nun ein Ergebnis auf Basis der Anforderungen dar?

Unter Berücksichtigung der genannten Faktoren war letztlich zwischen XML auf Dokumentenbasis und einer Relationalen Datenbank bzw. einem Relationalen Datenbank-Management-System (RDBMS) zu entscheiden. Unter einem Dokument wird dabei eine strukturierte Einheit gespeicherter Informationen, die als eigenständige Einheit in einem Informationssystem verarbeitet wird, verstanden (vgl. Geroimenko 2004, S. 40). XML ist deshalb in Erwägung zu ziehen, da zum einen die DOM-Erweiterung in PHP 5 im Gegensatz zu PHP 4 deutlich verbessert worden ist (vgl. Fuecks 2005, S. 247), zum anderen mit XTM eine XML-Syntax für Topic Maps existiert. Ein Gegenargument mag lauten, dass XTM 2.0, das sich, wie erwähnt, am TMDM orientiert, noch nicht endgültig verabschiedet worden ist. Es ist jedoch, wie bemerkt, davon auszugehen, dass der aktuelle Entwurf vom 19.6.2006 (vgl. ISO/IEC 13250-3) nicht mehr geändert wird.

Die Entscheidung für den Einsatz eines RDBMS ist letztendlich deshalb gefallen, da es sich bei Datenbanken i. a. um eine ausgewachsene, etablierte Technologie handelt, deren Anfänge bis in die späten 1960er Jahre zurückreichen (vgl. Sauer 2002, S. 13). Konkret wird MySQL eingesetzt, ein RDBMS der schwedischen Firma MySQL AB (<http://www.mysql.com>). MySQL überzeugt zum einen in der Betrachtung objektiver Qualitätsmerkmale wie Geschwindigkeit¹⁰ und Leistungsfähigkeit, Einfachheit der (Be-)Nutzung und Administration, Sicherheit und Portabilität (vgl. DuBois 2005, S. 3). Zum anderen ist MySQL in der Webentwicklung weit verbreitet. Als Indikator können die Serverkonfigurationen der wichtigsten deutschen Hostingunternehmen herangezogen werden. Webhostlist.de ermittelt wöchentlich über die Anzahl der aufgerufenen Seiten, *page views* (vgl. Bernhard Medien GmbH 2006), die am stärksten frequentierten Server. Eine Inspektion der ersten fünf Unternehmen und deren mittelpreisige Tarife, die Datenbanken bzw. Datenbanknutzung

¹⁰ Details zum Benchmarking von MySQL können unter <http://www.mysql.com/why-mysql/benchmarks/> nachgeschlagen werden.

beinhalten, zeigt, dass alle fünf MySQL als RDBMS anbieten (siehe <http://www.all-inkl.com>, <http://www.greatnet.de>, <http://www.internet4ever.de>, <http://www.strato.de>, <http://www.netclusive.de>; Stand 12.10.2006). Es ist bereits erwähnt worden, dass über die Verbreitung einer Technologie auf deren Popularität geschlossen werden kann. Rekursiv unterstützt eine hohe Popularität die Verteilung und Verfügbarkeit von Kompetenz. Das allseits bekannte und ebenfalls eingeführte Akronym „LAMP“ kennzeichnet insgesamt den Stellenwert von PHP und MySQL.

In Bezug auf die Kosten für die Beschaffung und den Betrieb von MySQL ist festzuhalten, dass das System u. a. unter der GPL verwendet werden kann, wodurch dem Nutzer de facto keinerlei Kosten für einen Einsatz an sich entstehen.

Konsistenz innerhalb einer im System gespeicherten Topic Map spielt eine entscheidende Rolle, und deshalb ist dieser Aspekt auch im Zusammenhang des Mergings bereits angesprochen worden. Die tatsächliche Speicherung der Daten erfolgt also in einer Relationalen Datenbank, die diese in Tabellen, den Relationen, vorhält. Für das zu Grunde liegende Relationale Modell ist die Garantie der Entity- und referenziellen Integrität von besonderer Bedeutung. Integrität bezeichnet im Zusammenhang mit Datenbanken grundsätzlich die Vermeidung widersprüchlicher Daten (vgl. Sauer 2002, S. 36). Dabei heißt Entity Integrität konkret, dass das als Primärschlüssel oder Bestandteil eines Primärschlüssels definierte Attribut zu keiner Zeit einen NULL-Wert enthalten darf. Referentielle Integrität bedeutet, dass der in einer Relation B enthaltene Fremdschlüssel auf einen Primärschlüssel in der referenzierten Relation A verweist (vgl. Sauer 2002, S. 36f.).

Diese Regeln sollen zugleich als Anforderung an das Datenbanksystem, MySQL, gestellt werden, auf das die PHPTMAPI-Implementierung aufsetzt, damit auch bzw. vor allem auf Systemseite Datenkonsistenz gewährleistet bleibt.

Als ein weiterer Punkt in diesem Kontext sind Transaktionen zu nennen. Eine Transaktion ist eine Abfolge von Befehlen in Structured Query Language (SQL), die als eine Einheit ausgeführt werden (vgl. DuBois 2005, S. 172). Transaktionsfähigkeit wird i. d. R. einem Datenbanksystem dann zugestanden, wenn deren Transaktionsimplementierung den ACID-Eigenschaften entspricht; ACID steht dabei als Akronym für *atomic*, *consistent*, *isolated* und *durable* (vgl. DuBois 2005, S. 173). Zusammenfassend bedeutet dies, dass

eine Transaktion ganz oder gar nicht ausgeführt werden muss (vgl. Sauer 2002, S. 178), die Konsistenz der Datenbank nicht zerstören und andere Transaktionen nicht beeinflussen darf sowie der durch die erfolgreiche Ausführung der Operationen hergestellte Datenbestand bzw. Datenzustand permanent gespeichert werden muss (vgl. DuBois 2005, S. 173).

Transaktionsfähigkeit ist also als eine weitere Anforderung in Bezug auf Sicherstellung der Datenkonsistenz an das RDBMS zu definieren.

MySQL kann diese letztgenannten Anforderungen dann erfüllen, wenn als sog. „storage engine“ (DuBois 2005, S. 123) und die Verwendung eines entsprechend spezifischen Tabellentyps InnoDB verwendet wird (vgl. MySQL AB 2006a; DuBois 2005, S. 123). Insofern ist die Verwendung von MySQL mit der Verfügbarkeit von InnoDB weiter zu spezifizieren, womit sich für das einzusetzende RDBMS und für zu definierende Teile des Datenbankschemas ein klares Anforderungsprofil ergibt.

5.7.2 Datenmodell und Datenbankschema der Implementierung

Eine oder auch mehrere Topic Maps (Genauer: die Daten, die als Topic Map strukturiert sind), die mittels der PHPMAPI-Implementierung erstellt, bearbeitet und auch wieder aus dem System entfernt werden können, werden, wie aus den vorherigen Ausführungen deutlich geworden ist, also in einer Relationalen Datenbank gespeichert. Eine Aufgabe, die dafür zu lösen ist, ist die Datenmodellierung, also das logische Datenbankdesign (vgl. Sauer 2002, S. 223), aus dem letztlich das Datenbankschema entwickelt wird. Ein solcher Prozess sieht z. B. wie folgt aus:

1. Festlegung der Entity Types¹¹ und ihrer Beziehungen zueinander;
2. Definition der Relationen und Attribute für die Entity Types und Beziehungen gemäß der Normalisierungsregeln (in Anlehnung an Sauer 2002, S. 223);
3. Entwicklung bzw. Ableitung des Datenbankschemas und Definition in Data Definition Language (DDL).

¹¹ Sauer verwendet den Terminus „Entities“ für die zu modellierenden Elemente in einem ERM, für die Attribute und Beziehungen zueinander definiert werden und die in einem ERD als Rechtecke dargestellt werden (vgl. Sauer 2002, S. 224). Entities sollen hier jedoch, wie erwähnt, als konkrete Einzelausprägungen von Entity Types verstanden werden.

Der Unterschied in der zu lösenden Aufgabe besteht jedoch darin, dass keine Entity Types festgelegt werden müssen, sondern das Topic Maps Data Model auf ein Relationales Modell abgebildet werden muss. Demnach sind die relevanten Entity Types und ihre Attribute als Information Item Types und ihre Eigenschaften (*properties*) im Topic Maps Data Model bereits definiert. So kann ein Datenmodellierer, der das TMDM kennt und die Technik bzw. Methodik der Erstellung von Entity-Relationship-Modellen und die Ableitung und Definition von Datenbankschemata beherrscht, diese Aufgabe lösen.

Ein anderer Ansatz mit nachvollziehbarer Methodik führt über XML und das sog. „object-relational mapping“ (Bourret 2001), das im folgenden kurz skizziert werden soll, da der Autor einen Mittelweg in der Definition der Tabellen der Datenbank beschritten hat.

Ausgangspunkt ist die jeweilige Document Type Definition (DTD). Bourret beschreibt einen Prozess, der zunächst DTDs auf Objektschemata und diese Objektschemata dann auf Datenbankschemata¹² abbildet (vgl. Bourret 2001).

Für den ersten Schritt ist die Unterscheidung einfacher und komplexer Elementtypen in einer DTD erforderlich. Einfache Elementtypen sind solche, die lediglich PCDATA-Inhalt enthalten. Komplexe Elementtypen sind Elementtypen, die wiederum ein oder mehrere Elementtypen oder gemischten Inhalt enthalten oder Attribute besitzen bzw. besitzen können. Einfache Elementtypen beinhalten also nur einen Datenwert und entsprechen skalaren Datentypen in Objektorientierten Programmiersprachen (bspw. String oder Integer), während komplexe Elementtypen eine Struktur definieren und in Objektorientierten Programmiersprachen Klassen entsprechen (vgl. Bourret 2001).

Nach dieser Distinktion werden einfache Elementtypen auf skalare Datentypen und komplexe Datentypen auf Klassen abgebildet, wobei jeder Elementtyp des Inhaltsmodells eines komplexen Typs als Eigenschaft der resultierenden Klasse abgebildet wird. Der Datentyp der Eigenschaft entspricht dabei dem Datentypen, auf den der referenzierte Elementtyp abgebildet worden ist. Referenzen auf komplexe Elementtypen entsprechen Referenzen auf die Klassen, die komplexe Elementtypen repräsentieren. In einem letzten Schritt

¹² Bei Datenbanken besteht das Schema aus einer Menge von Tabellendefinitionen, Views und Berechtigungen (vgl. Sauer 2002, S. 54). Im folgenden werden lediglich die Tabellendefinitionen betrachtet.

werden Attribute als Eigenschaften von Klassen abgebildet, wobei der Datentyp des Attributs den Datentyp der Eigenschaft bestimmt (vgl. Bourret 2001).

In Tabelle sieben wird das *mapping* einer DTD auf ein Objektschema beispielhaft gezeigt.

Beispiel DTD	Objektschema, Klassendefinition (informativ)
<pre><?xml version="1.0" encoding="UTF-8"?> <!ELEMENT A_komplex (B_einfach, C_komplex) > <!ELEMENT B_einfach (#PCDATA) > <!ATTLIST A_komplex F_einfach CDATA #REQUIRED > <!ELEMENT C_komplex (D_einfach) > <!ELEMENT D_einfach (#PCDATA) ></pre>	<pre>class A_komplex { b_einfach: String; c_komplex: C_komplex; f_einfach: String; } class C_Komplex { d_einfach: String; }</pre>

Tab. 7: Abbildung einer DTD auf ein Objektschema

Aus dem Objektschema wird nun das Datenbankschema bzw. der die Tabellendefinitionen betreffende Teil entwickelt. Skalare Datentypen werden als Spalten definiert, während Referenzen, im Beispiel `c_komplex` in der Klasse `A_komplex`, über Primärschlüssel-Fremdschlüssel-Beziehungen abgebildet werden (vgl. Bourret 2001), wobei Primär- und Fremdschlüsseldefinitionen in entsprechenden Spalten definiert werden müssen.

Eine derartige Abbildung wird in Tabelle 8 gezeigt.

Objektschema, Klassendefinition (informativ)	Datenbankschema, Tabellendefinition (informativ)
<pre>class A_komplex { b_einfach: String; c_komplex: C_komplex; f_einfach: String; } class C_Komplex { d_einfach: String; }</pre>	<pre>Tabelle A: Spalte a_pk Spalte b Spalte c_pk_fk Spalte f Tabelle C: Spalte c_pk Spalte d</pre>

Tab. 8: Abbildung eines Objektschemas auf ein Datenbankschema

Da das Inhaltsmodell eine 1:1-Beziehung der Elemente `A_komplex` und `C_komplex` definiert (kein Frequenzindikator), kann der Fremdschlüssel auch in Tabelle C definiert werden, also entsprechend Spalte `a_pk_fk`. Bei mehrfachen

Auftretens eines Elementes, was über die Frequenzindikatoren * oder + angezeigt wird:

```
<!ELEMENT A_komplex  
    ( B_einfach, C_komplex* )  
>
```

oder

```
<!ELEMENT A_komplex  
    ( B_einfach, C_komplex+ )  
>
```

und demnach einer 1:n-Beziehung entspricht, muss der Fremdschlüssel in Tabelle C definiert werden.

Mit dieser Methode lassen sich zügig und präzise die Datenbanktabellen ableiten, die für die Speicherung spezifischer Datenstrukturen, welche zunächst mit einer DTD definiert worden sind, benötigt werden. Es wird deutlich, dass eine Visualisierung entfällt und nach der Bestimmung der Objekte direkt auf den Teil des Datenbankschemas, das die Tabellendefinition betrifft, abgeleitet werden kann. Die Kardinalitäten sind im jeweiligen Inhaltsmodell durch die Frequenzindikatoren bestimmt.

Das Verhältnis von XTM 2.0 und dem Topic Maps Data Model sowie der Status von XTM 2.0 sind eingehend dargelegt worden. Also muss sich aus der DTD von XTM 2.0 ein Datenbankschema ableiten lassen, das die Modellvorgaben des TMDM reflektiert. Das Datenbankschema ist jedoch gegen das Topic Maps Data Model zu prüfen, bevor es implementiert wird.

Es kann an dieser Stelle konstatiert werden, dass durch Object-Relational Mapping der XTM 2.0 DTD ein adäquates Modell als Ausgangspunkt für die Definition der Tabellen geschaffen werden kann, da sich präzise die Topic Maps-Konstrukte in ihren Beziehungen zueinander bis auf Relationen-, d. h. Tabellenebene ableiten lassen.

Anhang F zeigt die Klassen, die aus der DTD des Entwurfs zu XTM 2.0 abgeleitet worden sind, als informatives (!) UML Klassendiagramm, das das Objektschema der Ableitung repräsentiert. Die Kompositionen als existenzabhängige Beziehungen zwischen den Klassen (Teil – Ganzes) (vgl. Oestereich 2005, S. 92) sollen die ursprüngliche Eltern-Kind-Beziehung der zu Grunde liegenden Elemente der DTD verdeutlichen.

Die Ableitung der konkreten Tabellen soll jedoch nicht streng nach dem dargestellten Schema erfolgen, da zum einen, wie erwähnt, immer ein Abgleich mit dem Topic Maps Data Model erfolgen muss und zum anderen ein möglichst effizientes Schema, was sich z. B. in der Zahl der Tabellen ausdrückt,

implementiert werden soll. Zuerst werden die Spezifika zur Vereinfachung erläutert.

Die Elemente `itemIdentity`, `subjectLocator`, `subjectIdentifier`, `resourceRef` und `topicRef` sind leere Elemente, die jeweils einzig das Attribut `href` enthalten. Definiert das Inhaltsmodell der DTD keinen Mehrfach-Frequenzindikator, so ergäbe eine konsequente Ableitung jeweils eine Relation (Tabelle), die o. g. Element repräsentiert, mit nur einem Attribut (Spalte) `href` neben dem Primärschlüssel, die in 1:1-Beziehung zu der Relation steht, die das Elternelement repräsentiert. Diese Konstellationen werden aufgelöst, indem `href` als Attribut der Relation zugewiesen wird, die das ursprüngliche Eltern-element repräsentiert. 1:cn-Beziehungen werden gemäß der dargestellten Methodik über zwei Relationen abgebildet.

Ebenso wird `type` aufgelöst, da die Beziehung von einer Relation zu `type` 1:1 oder 1:c und von `type` zu `topicRef` 1:1 ist. Also kann das `href`-Attribut von `topicRef` als optionales `type`-Attribut der ersten Relation zugewiesen werden. Einen Sonderfall stellt `instanceOf` dar. `InstanceOf` repräsentiert eine Selbstreferenz der Topic-Relation in n:m-Beziehung und kann über die zwei Relationen `topic` und `instanceof` abgebildet werden.

Eine weitere Vereinfachungsmöglichkeit bietet sich dort, wo ein Objekt in mehreren Objekten referenziert wird, Beispiel `scope`. Konsequenterweise müsste jede Referenz über zwei Relationen gelöst werden, also exemplarisch `association` und `association_scope` oder `occurrence` und `occurrence_scope`. Es ist jedoch auch möglich, eine Relation `scope` zu definieren und jeweils Primärschlüssel-Fremdschlüssel-Beziehungen über entsprechende Attribute/Spalten, die NULL-Werte enthalten dürfen, festzulegen. Die Beziehung zwischen `scope` und `topicRef` wird dann nach Methode definiert.

Gleichfalls wird mit `itemIdentity` verfahren, das in jedem Objekt referenziert wird, das ein Topic Maps-Konstrukt repräsentiert. Hier ist ebenfalls nicht jeweils eine `itemIdentity`-Relation abgeleitet, sondern eine Relation `topicmapconstruct` definiert worden, die die Fremdschlüssel der referenzierten Relationen, welche die Topic Maps-Konstrukte repräsentieren, als Attribute/Spalten, die NULL-Werte enthalten dürfen, aufnimmt. Somit stellt `topicmapconstruct` quasi einen Index der Topic Maps-Konstrukte dar, was bereits im Zusammenhang der Ausführungen zum Metamodell des TMDM angedeutet worden ist. In `topicmapconstruct` wird zudem die reifier-Eigenschaft der

Objekte, die Topic Maps-Konstrukte außer Topic repräsentieren (siehe Abb. 20), als Attribut definiert. Die 1:cn-Beziehung der einzelnen Objekte zu `itemIdentity` wird mit Hilfe der Relation `topicmapconstructref` abgebildet.

Die veränderte Abbildung von `resourceData` und `resourceRef` in `occurrence` und `variant` aus dem Objektschema auf das Datenbankschema ergibt sich aus der Betrachtung der jeweiligen Definition im TMDM. `ResourceRef` nimmt in XTM 1.0 und 2.0 eine URI-Referenz auf eine Ressource auf, während `resourceData` einen Datenwert enthält, der in XTM 2.0 zusätzlich durch einen Datentypen spezifiziert werden muss. Das TMDM definiert jedoch, dass diese Unterscheidung lediglich durch den Datentypen vorgenommen wird und gibt entsprechend nur die Eigenschaften `value` und `datatype` vor (vgl. ISO/IEC 13250-2). In der Serialisierung werden Datenwerte vom Datentyp bzw. der identifizierenden Adresse `http://www.w3.org/TR/xmlschema-2/#anyURI` innerhalb des `resourceRef`-Elementes definiert, während alle anderen Datenwerte in das `resourceData`-Element geschrieben werden.

Im Ergebnis sind die erarbeiteten Relationen die folgenden (in alphabetischer Reihenfolge):

Tabelle `qtm_association`

Spalte	Typ	Beschreibung
<code>id</code>	<code>int</code>	ID der Association; Primärschlüssel
<code>type_id</code>	<code>int</code>	ID des Topics, das den Typ definiert; Fremdschlüssel
<code>topicmap_id</code>	<code>int</code>	ID der Topic Map, zu der die Association gehört (<i>parent</i>); Fremdschlüssel

Tabelle `qtm_assocrole`

Spalte	Typ	Beschreibung
<code>id</code>	<code>int</code>	ID der Association Role; Primärschlüssel
<code>association_id</code>	<code>int</code>	ID der Association, zu der die Association Role gehört (<i>parent</i>); Fremdschlüssel
<code>type_id</code>	<code>int</code>	ID des Topics, das den Typ definiert; Fremdschlüssel
<code>reference</code>	<code>int</code>	ID des Topics, das der Role Player ist; Fremdschlüssel

Tabelle qtm_instanceof

Spalte	Typ	Beschreibung
id	int	ID eines Tuples; Primärschlüssel
topic_id	int	ID des Topics, das als Instanz definiert ist; Fremdschlüssel
type_id	int	ID des Topics, das den Typ definiert; Fremdschlüssel

Tabelle qtm_occurrence

Spalte	Typ	Beschreibung
id	int	ID der Occurrence; Primärschlüssel
topic_id	int	ID des Topics, zu dem die Occurrence gehört (<i>parent</i>); Fremdschlüssel
type_id	int	ID des Topics, das den Typ definiert; Fremdschlüssel
value	text	Datenwert der Occurrence
datatype	varchar(255)	Datentyp, der den Datenwert spezifiziert

Tabelle qtm_scope

Spalte	Typ	Beschreibung
id	int	ID des Scope; Primärschlüssel
association_id	int	ID der Association, für die ein Scope definiert ist; Fremdschlüssel
topicname_id	int	ID des Topic Name, für den ein Scope definiert ist; Fremdschlüssel
occurrence_id	int	ID der Occurrence, für die ein Scope definiert ist; Fremdschlüssel
variant_id	int	ID des Variant (alternativer Topic Name), für den ein Scope definiert ist; Fremdschlüssel

Tabelle qtm_scoperef

Spalte	Typ	Beschreibung
id	int	ID eines Tuples; Primärschlüssel
scope_id	int	ID des Scope, für den eine Referenz auf ein Topic definiert ist; Fremdschlüssel
reference	int	ID des Topics, das den Scope definiert; Fremdschlüssel

Tabelle qtm_subjectidentifier

Spalte	Typ	Beschreibung
id	int	ID eines Tuples; Primärschlüssel
topic_id	int	ID des Topics, für das ein Subject Identifier definiert ist; Fremdschlüssel
locator	varchar(255)	URI einer Ressource, die Subject Indicator ist

Tabelle qtm_subjectlocator

Spalte	Typ	Beschreibung
id	int	ID eines Tuples; Primärschlüssel
topic_id	int	ID des Topics, für das ein Subject Locator definiert ist; Fremdschlüssel
locator	varchar(255)	URI einer Ressource, die der Aussagegegenstand eines Topics <i>ist</i>

Tabelle qtm_topic

Spalte	Typ	Beschreibung
id	int	ID des Topics; Primärschlüssel
topicmap_id	int	ID der Topic Map, zu der das Topic gehört (<i>parent</i>); Fremdschlüssel

Tabelle qtm_topicmap

Spalte	Typ	Beschreibung
id	int	ID der Topic Map; Primärschlüssel
baselocator	varchar(255)	Base Locator der Topic Map

Tabelle qtm_topicmapconstruct

Spalte	Typ	Beschreibung
id	int	ID eines Tuples; Primärschlüssel
association_id	int	ID einer Association; Fremdschlüssel
assocrole_id	int	ID einer Association Role; Fremdschlüssel
occurrence_id	int	ID einer Occurrence; Fremdschlüssel
topic_id	int	ID eines Topics; Fremdschlüssel
topicmap_id	int	ID einer Topic Map; Fremdschlüssel
topicname_id	int	ID eines Topic Name; Fremdschlüssel
variant_id	int	ID eines Variant; Fremdschlüssel
reifier	int	ID eines Topics, das ein Topic Maps-Konstrukt „reifiziert“; Fremdschlüssel

Tabelle qtm_topicmapconstructref

Spalte	Typ	Beschreibung
id	int	ID eines Tuples; Primärschlüssel
topicmapconstruct_id	int	ID des Topic Maps-Konstruktes, für das ein Item Identifier definiert ist; Fremdschlüssel
locator	varchar(255)	URI einer Ressource, die ein Topic Maps-Konstrukt eindeutig identifiziert

Tabelle qtm_topicname

Spalte	Typ	Beschreibung
id	int	ID des Topic Name; Primärschlüssel
topic_id	int	ID des Topics, zu dem der Topic Name gehört (<i>parent</i>); Fremdschlüssel
type_id	int	ID des Topics, das den Typ definiert; Fremdschlüssel
value	varchar(255)	Datenwert eines Topic Name (Datentyp ist immer String)

Tabelle qtm_variant

Spalte	Typ	Beschreibung
id	int	ID des Variant; Primärschlüssel
topicname_id	int	ID des Topic Name, zu dem der Variant gehört (<i>parent</i>); Fremdschlüssel
value	varchar(255)	Datenwert des Variant
datatype	varchar(255)	Datentyp, der den Datenwert spezifiziert

Ein revers aus der Datenbank erstelltes Entity-Relationship-Diagramm in Crowsfoot-Notation zur Übersicht ohne Attributangaben ist in Anhang G beigefügt. Auf eine Diskussion des Relationalen Modells aus Perspektive der Normalisierungsregeln bzw. der Normalformen soll in dieser Arbeit verzichtet werden.

Eine Anforderung an die Implementierung der Datenbank betraf die Referentielle Integrität. Auf Ebene der Datendefinitionsanweisungen mit den CREATE TABLE-Anweisungen muss diese explizit definiert werden. Für MySQL mit InnoDB als *storage engine* sind dazu folgende Bedingungen zu beachten:

- Die betroffenen Tabellen müssen vom Typ InnoDB sein;
- In der referenzierenden Tabelle muss ein Index definiert sein, in dem zuerst die Fremdschlüsselspalten in der richtigen Reihenfolge aufgeführt sind;

- In der referenzierten Tabelle muss ein Index definiert sein, in dem die referenzierten Spalten zuerst aufgelistet sind (vgl. MySQL AB 2006b). Ein Primärschlüssel ist dabei an sich ein eindeutiger Index und muss demnach nicht als Index definiert werden (vgl. MySQL AB 2006c).

Folgendes Beispiel zeigt die vollständigen SQL-Anweisungen zur Erstellung der Tabellen *qtm_topicmap* (*parent*) und *qtm_topic* (*child*):

```
CREATE TABLE qtm_topicmap (
    id INT NOT NULL AUTO_INCREMENT,
    baselocator VARCHAR(255) NOT NULL,
    PRIMARY KEY(id)
) ENGINE=InnoDB;

CREATE TABLE qtm_topic (
    id INT NOT NULL AUTO_INCREMENT,
    topicmap_id INT NOT NULL,
    PRIMARY KEY(id),
    INDEX t_ref_ind (topicmap_id),
    FOREIGN KEY (topicmap_id) REFERENCES qtm_topicmap(id) ON DELETE
CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;
```

Über die Anweisungen `ON DELETE CASCADE` sowie `ON UPDATE CASCADE` kann auf RDBMS-Ebene gesteuert werden, was mit betroffenen Datensätzen in *child*-Tabellen geschieht, wenn Datensätze in referenzierten *parent*-Tabellen entfernt oder geändert werden. Für das gezeigte Beispiel bedeutet dies, dass wenn eine Topic Map aus der Tabelle *qtm_topicmap* entfernt wird, alle Datensätze in *qtm_topic*, in denen als Fremdschlüssel die ID der zu entfernenden Topic Map als Fremdschlüssel *topicmap_id* definiert ist, ebenfalls gelöscht werden. Wird die ID der Topic Map geändert, wird der Fremdschlüssel in der referenzierenden Tabelle ebenfalls geändert. Somit muss bspw. kaskadierendes Löschen nicht „manuell“ z. B. in PHP vorgenommen werden, was eine potentielle Fehlerquelle darstellt (Datensätze können z. B. vergessen werden) und im Resultat zu Inkonsistenzen führen kann. `ON DELETE/UPDATE CASCADE`-Anweisungen erlauben also die Erstellung effizienterer Anwendungen und erhöhen die Sicherheit bzgl. der Datenkonsistenz.

Die `CREATE TABLE`-Anweisungen zur Definition aller genannten Tabellen als Teil des Schemas können in Anhang H nachgeschlagen werden.

5.7.3 Bewertung des Status quo

Eine Bewertung der vorgestellten Implementierung von PHPTMAPI in der Art, wie Software i. a. beurteilt wird, z. B. in Benchmarks zu Performanzaspekten, kann zu diesem Zeitpunkt noch nicht formuliert werden und könnte auch zur Zielführung keinen wesentlichen Beitrag leisten. Für die Untersuchung des exemplarischen Praxisfalls genügt eine erste Implementierung. Diese Aussage liegt zwangsläufig in der Tatsache begründet, dass für einen Benchmark eine vergleichbare Implementierung von PHPTMAPI als Vergleichsobjekt nicht existiert.

Eine erste Bewertung bzgl. der zu Grunde liegenden Spezifikation, PHPTMAPI, ist jedoch möglich und sinnvoll deshalb, da es im Rahmen der Untersuchung im Grundsatz um die Erarbeitung eines Modells zur standardisierten Strukturierung und Speicherung von Metadaten in Internetanwendungen und die Etablierung von Semantik mit Topic Maps geht und einem standardisierten API für den Zugriff und zur Verarbeitung derart strukturierter Daten auf Applikationsebene in diesem Kontext eine wesentliche Bedeutung zukommt. Die Überprüfung der Praxistauglichkeit der Spezifikation ist, wie auch schon erwähnt, zudem als eine Kernaufgabe der Implementierung anzusehen. Welche Schlüsse können also bis dato gezogen werden?

Grundsätzlich ist festzustellen, dass PHPTMAPI in der erarbeiteten Struktur implementierbar ist und bis auf den angesprochenen Fehler in der Definition des Rückgabewertes in der Methode `getReified()` in `TopicInterface` weitere Nichtkonformitäten zum Topic Maps Data Model nicht zu erkennen sind und auch Rückmeldungen von Anwendern diesbezüglich gegenüber den Autoren noch nicht geäußert worden sind. Da der erarbeitete Teil des Schemas der Datenbank, auf welche die hier vorgestellte Implementierung aufsetzt, das Topic Maps Data Model reflektiert, enthält das Array in der Rückgabe von `getReified()` in der Implementierung immer nur eine Instanz von `TopicMapObjectInterface`, wodurch dieser Fehler auf Anwendungsebene inhaltlich egalisiert werden kann.

Es ist die Vermutung artikuliert worden, dass PHPTMAPI weiter vereinfacht werden kann. Für die grundsätzliche Struktur ist mit der Einführung eines `ReifiableObjectInterface` ein solcher Aspekt angesprochen worden. In der Realisierung der Implementierung haben sich weitere Möglichkeiten für speziellere Punkte ergeben. Die dargelegte Auffindungsprozedur einer konkreten

TopicMapSystemFactory-Klasse ist in dieser Form umsetzbar und zeigt eine Möglichkeit zur Einbindung weiterer Klassen, bspw. solcher, die einen Index repräsentieren. Eine Bewertung nach der praktischen Anwendung sollte aus zwei Perspektiven erfolgen, welche einerseits die Frage der Praktikabilität und zum anderen die Frage der Sicherheit berücksichtigt.

Bzgl. des erstgenannten Punktes ist zu äußern, dass die derzeit festgelegte Methode in der vergleichenden Betrachtung anderer Open Source-PHP-Anwendungen ungewöhnlich ist. Konfigurationen i. a. werden häufig in einer `configuration.php`, `config.php` o. ä. vorgenommen, z. B. in WordPress, phpMyAdmin, phpBB, PHP-Nuke oder moodle. Für die Datenbankeinstellungen in der vorgestellten Implementierung alias QuaaxTM wird ebenfalls derart verfahren. Somit kann für diese Methode ein quasi Referenz- bzw. Vorbildstatus konstatiert werden. Der Inhalt solcher Konfigurationsdateien kann über `include()` oder `require()` bei Bedarf wie PHP-Code bequem integriert werden, sofern in diesen Dateien die gültigen PHP-Start- und -Ende-Tags definiert sind, da der Parser am Beginn der Zieldatei in den HTML-Modus wechselt (vgl. PHP Group 2006b). Für einen Prozess wie die *lookup procedure* der vorgestellten PHTMAPI-Implementierung muss mit Dateisystem- oder Verzeichnisfunktionen operiert werden, was im Vergleich etwas aufwändiger in der Programmierung ist.

In der Betrachtung aus der Sicherheitsperspektive ist festzustellen, dass der vollständige Pfad zur konkreten TopicMapSystemFactory-Klasse im Klartext unmaskiert in der spezifischen ASCII-Datei vorliegt. Eine Maskierung kann nur über expliziten Zugriffsschutz mit serverspezifischen Methoden wie bspw. über `.htaccess`-Dateien erfolgen. Bei PHP-Dateien, die inkludiert werden, kann eine Maskierung einfach mit PHP erfolgen, indem Konfigurationswerte Variablen zugewiesen oder mit `define()` als globale Konstanten deklariert werden.

Somit ist zu überlegen, den Auffindungsprozess für die konkrete TopicMapSystemFactory-Klasse zu Gunsten einer Definition in einer PHP-Datei zu ändern.

Ein weiterer, hier zuletzt genannter Ansatzpunkt zur Vereinfachung ist in der Vielzahl der Ausnahmen, den Exceptions, zu sehen. Diese sind in ihrer Benennung, Anzahl und Struktur, wie erwähnt, nahezu identisch von TMAPI übernommen worden. Die Unterscheidung in zwei Exception-Superklassen, z. Zt. `TMAPIException` und `TMAPIRuntimeException`, ist prinzipiell sinnvoll, um

Fehler, die das System an sich betreffen und Fehler, die Topic Maps-spezifisch sind, zu unterscheiden. Es bleibt zu diskutieren, ob die Vielzahl an jeweiligen Unterklassen, die Exceptions, die konkret „geworfen“ werden, notwendig ist. Im besonderen dann, wenn diese keine Methoden außer dem Konstruktor implementieren, der als einzigen Parameter eine spezifische Fehlermeldung aufnimmt. Es wäre alternativ möglich, die Spezifität der Exception über einen Parameter zu definieren:

```
class SpecificTMAPIException extends TMAPIException{
    public function __construct($specifier, $msg){}
}
```

Folgend ein Beispiel für den Aufruf einer Exception, die TopicInUseException entspricht:

```
throw new SpecificTMAPIException('TopicInUseException', 'Topic is in
the scope of an occurrence!');
```

Dies wäre eine mögliche Ausgangsbasis, die Struktur der Exceptions zu überarbeiten und ihre Anzahl im Sinne einer verbesserten Übersicht und in der Folge einer vereinfachten Anwendung zu reduzieren.

In der Bewertung dieser ersten Ergebnisse kann die aufgestellte Argumentation für die Überarbeitung von PHPTMAPI weiter unterstützt werden. Weitere wertvolle Hinweise dazu aus der Community sind nach der Veröffentlichung von QuaaxTM zu erwarten.

Die Schwierigkeit in der Beurteilung der Implementierung liegt darin zu sehen, dass diese, wie erwähnt, am Anfang einer Entwicklung steht, somit also kein Zustand, sondern die Momentaufnahme eines Prozesses dargestellt werden muss. Die Bedeutung liegt in der gelieferten Beurteilung der Spezifikation. Eine fundierte Spezifikation, die standardtauglich ist, ist essentiell für zu skizzierende Anwendungsszenarios, die auf Standards basieren.

Der aktuelle Entwicklungsstatus der Implementierung ist auf der beigefügten CD-ROM im Verzeichnis `quaaxtm` enthalten und entspricht in der Struktur im wesentlichen dem Paket, das bei [sourceforge](http://sourceforge.net) heruntergeladen werden kann. Die Projektseite ist unter <http://quaaxtm.sourceforge.net> erreichbar. Gegenüber den dort verfügbaren Versionen stellt die beigelegte Version den letzten Entwicklungsstand dar, wobei der Autor die Absicht verfolgt, verbesserte Versionen der das PHPTMAPI-Kernpaket implementierenden Klassen regelmäßig in das Concurrent Versions System (CVS) zu laden, von wo aus Interessierte diese herunterladen und die im Paket enthaltenen Klassen ersetzen können.

6 Tagging-Modell

Im folgenden soll ein konkretes Tagging-Modell mit Topic Maps definiert und unter Verwendung der vorgestellten Software auch implementiert und angewendet werden. Für das zu definierende Tagging-Modell sind aus der Perspektive der Standardisierung zwei Dimensionen zu unterscheiden. Zum einen beschreibt das Modell an sich einen Vorschlag zur Standardisierung des skizzierten Parallelmodells, die Meta(daten)ebene im Tagging-Kontext, zum anderen repräsentiert die zu Grunde liegende Technologie, ISO/IEC 13250 Topic Maps, an sich einen Standard.

6.1 Definition des Modells: Entwicklung einer Ontologie

Ausgangspunkt ist das in Kapitel 4.4.2 erarbeitete rudimentäre Entity-Relationship-Diagramm, das das konzeptionelle Tagging-Modell in seiner Grundsätzlichkeit darstellt. Dieses Modell ist nun aus der Topic Maps-Perspektive zu betrachten.

Wie erwähnt, lassen sich mit Topic Maps Ontologien formulieren. In der Betrachtung dessen, was eine Ontologie umfasst bzw. sie charakterisiert (siehe S. 66), wird deutlich, dass dieses konzeptionelle Tagging-Modell jene Basisvoraussetzungen schafft, indem es Entities (besser: Entity Types, s. o.) und ihre Beziehungen zueinander beschreibt. Folglich bemerken Obrst und Liu richtig, dass die Entwicklung von Entity-Relationship-Diagrammen für Datenbanken (besser: für deren Entwurf und Umsetzung, Anm. d. Autors) als ein gleichartiger Prozess wie der der Entwicklung einer Ontologie anzusehen ist (vgl. Obrst/Liu 2003, S. 126) und beschreiben in diesem Zusammenhang folgende grobe Arbeitsschritte:

1. Define a universe of discourse [...] Start this universe-of-discourse-defining process by listing the concepts you want to include in that universe [...]
2. List how the various things relate to each other [...]
3. Now build the ontology based on your analysis in steps 1-2 [...] (Obrst/Liu 2003, S. 127)

Die Domäne, äquivalent zu Obrsts und Lius *universe of discourse*, ist mit dem Tagging-Kontext eingehend skizziert worden. Das ERD, welches das konzeptionelle Tagging-Modell als ein Entity-Relationship-Modell visualisiert, zeigt die zentralen Konzepte (*concepts*) und ihre Beziehungen (*relations*) zueinander. Es kann konstatiert werden, dass die Entwicklung des konkreten Tagging-Mo-

dells auch als Entwicklung einer (Basis)Ontologie angesehen werden kann, die die zentralen Konzepte als „schematic portion“ (Obrst/Liu 2003, S. 127) einer Ontologie beinhaltet.

Im folgenden wird die Definition des Modells mit Hilfe der vorgestellten Topic Maps-Konstrukte in entsprechender, ebenfalls eingeführter Topic Maps-Terminologie erfolgen. Aus Modellbildungsperspektive kann argumentiert werden, dass ISO 13250-2 Topic Maps, das TMDM, neben seiner Funktion als Datenmodell (siehe Kapitel 5.3) das Metamodell für Topic Maps definiert, das u. a. die verfügbaren Arten von Modellbausteinen und Beziehungen zwischen Modellbausteinen spezifiziert (vgl. Sinz 2001, S. 312). Diese Position nehmen z. B. Gulbrandsen (vgl. Gulbrandsen 2005, S. 6) oder Colomb ein (vgl. Colomb 2006, S. 1)¹³.

Es ist nun zu diskutieren, wie die zentralen Konzepte und ihre Beziehungen zueinander mit Topic Maps-Konstrukten abgebildet werden. Auf den ersten Blick ist die Repräsentation von *tag*, *taggable resource* und *tagging user* durch jeweils ein Topic offensichtlich. Das TMDM sieht für die Beziehung zwischen Aussagegegenstände (*subjects*), die durch Topics repräsentiert werden, und Informationsressourcen die Verwendung von Occurrences als eine spezielle Beziehung zwischen den genannten Protagonisten vor (vgl. ISO/IEC 13250-2). Damit wäre es möglich, Ressourcen als Occurrences den Instanzen des Topics zuzuordnen, welches im Modell das *tag*-Konzept repräsentiert und lediglich nur *tag* und *tagging user* mit dem Topic-Konstrukt abzubilden, was dem üblichen Procedere entspricht. Derartiges Vorgehen beschreibt bspw. Garshol (vgl. Garshol 2006). Diese Vorgehensweise ist nicht zwingend. Welche Argumente sprechen dafür, eine Ressource als ein Topic zu definieren, diese also als Aussagegegenstand zu betrachten?

Da letztendlich aus einem implementierten Tagging-Modell über eine Schnittstelle der externe Zugriff auf relevante Daten möglich sein soll, müssen einer Ressource weitere Informationen zugeordnet werden können, bspw. das Datum des Anlegens bzw. Speicherns. Dies könnte durch Reification (siehe S. 72) gelöst werden. Reification ist jedoch recht abstrakt und trägt dementsprechend nicht zum rascheren Verständnis und zur einfachen Implementierung des Tagging-Modells bei. Des weiteren führt die Abbildung über das

¹³ Es ist weiter möglich, das Reference Model (RM) für Topic Maps (siehe Abb. 19) als das Metamodell des Topic Maps Data Model zu betrachten.

Occurrence-Konstrukt zu einer existenzabhängigen Bindung zwischen dem einen Aussagegegenstand repräsentierenden Topic und einer Ressource: Wird ein Tag-Topic gelöscht, so wird die über Occurrence definierte Verknüpfung zu einer Ressource ebenfalls entfernt.

Ein weiteres Argument betrifft die ontologische Erweiterungsfähigkeit eines implementierten Modells. So könnten die Topics, die Instanzen von *tag* bzw. die von diesen repräsentierten *subjects* repräsentieren, bspw. weiter als Thesaurus strukturiert werden, indem spezifische Relationen zwischen Deskriptoren wie „übergeordneter Begriff“ (OB) oder „untergeordneter Begriff“ (UB) über das Association-Konstrukt abgebildet werden. Deskriptoren werden dann wiederum von ausgewählten Tag-Topics repräsentiert.

Ggf. soll allein diese Untermenge der Gesamtstruktur weiterverarbeitet, z. B. exportiert werden, unabhängig existierender bzw. zugewiesener Ressourcen.

Ein letztes Argument betrifft das möglichst Redundanzen zu vermeidende Design des Modells. Das Prinzip des Mergings ist eingeführt worden. Wird eine Ressource von einem Topic repräsentiert, so ist diese Ressource der Aussagegegenstand dieses Topics. Die Adresse der Ressource, i. d. R. ein URL, der Subject Locator. Das TMDM definiert, dass zwei Topics identisch sind, wenn sie jeweils einen identischen Subject Locator aufweisen (vgl. ISO/IEC 13250-2; siehe S. 77). In der konkreten Anwendung eines implementierten Modells bedeutet dies, dass wenn n Nutzer eine identische Ressource speichern, diese Ressource lediglich von einem Topic mit einem Subject Locator repräsentiert werden muss, jedoch nicht über n Occurrences zu einem jeweiligen Tag, das mit dem Topic-Konstrukt abgebildet wird. Zusätzlich ist dann die Beziehung zu den „besitzenden“ Nutzern und den zugewiesenen Tags darzustellen.

Deshalb sollen die Konzepte *tag*, *taggable resource* und *tagging user* jeweils über das Topic-Konstrukt abgebildet werden (im folgenden wird der Term „Konstrukt“ zur verbesserten Lesbarkeit vernachlässigt). In ihrer Funktion sind diese Topics Topic Types, da in einer Implementierung Instanzen (z. B. ein konkretes Tag wie „XML“, die „data portion“ (Obrst/Liu 2003, S. 127) der Ontologie) als Topics in der spezifischen Klasse-Instanz-Relation diesen Topics, die dann die Topic Types darstellen, zugewiesen werden.

Die Relationen zwischen diesen Topic Types werden als Associations modelliert. Um die Associations zu spezifizieren und zu charakterisieren, müssen

Topics definiert werden, die diesen Associations als Association Types zugeordnet werden:

- Spezifizierung der Association zwischen *tag* und *tagging user*: Topic *assigned-by*;
- Spezifizierung der Association zwischen *tagging user* und *taggable resource*: Topic *saved-by*;
- Spezifizierung der Association zwischen *tag* und *taggable resource*: Topic *specified-by*.

In einer Association stellen Association Roles die Beteiligungen von Topics dar. Association Roles werden durch die Typ-Zuweisung von Topics als deren Association Role Types spezifiziert und charakterisiert, die für die erarbeiteten Associations im folgenden zu definieren sind:

- Spezifizierung der Association Roles in der Association zwischen *tag* und *tagging user*: Topics *user-to-tag* und *tag-to-user*;
- Spezifizierung der Association Roles in der Association zwischen *taggable resource* und *tagging user*: Topics *user-to-resource* und *resource-to-user*;
- Spezifizierung der Association Roles in der Association zwischen *taggable resource* und *tag*: Topics *tag-to-resource* und *resource-to-tag*.

Folgende Tabelle gliedert alle bisher definierten Topics anhand ihrer Funktion/Rolle im Modell, dabei entspricht die zeilenweise Zuordnung der Association Types und der Association Role Types der Abbildung der o. g. Beziehungen, woraus die Doppeldefinition in der Spalte „Association Role Types“ resultiert:

Topic Types	Association Types	Association Role Types
tag	assigned-by	user-to-tag, tag-to-user
taggable resource	saved-by	user-to-resource, resource-to-user
tagging user	specified-by	tag-to-resource, resource-to-tag

Tab. 9: Zentrale Konzepte repräsentierende Topics und ihre Funktion im Topic Maps-Tagging-Modell

Diese Topics bilden das Tagging-Modell in seiner Grundstruktur ab, aus der Ontologieperspektive definieren sie den schematischen Teil.

Weiter ist es sinnvoll, die Ressourcen zu spezifizieren. Konkret: Um welche Art von Ressource handelt es sich? Dafür sollen folgende Arten unterschieden werden: Audio, Video, Grafik/Bild, Webseite sowie undefiniert bzw. undefinierte Ressource. Wie erwähnt, sollen Ressourcen weitere Informationen wie das Datum der Erstellung/Speicherung zugewiesen werden. Diese können als (interne) Occurrences einem Topic, einer Instanz von *taggable resource*, zugeordnet werden. Die Occurrence wird ebenfalls durch eine Verknüpfung in der spezifischen Type-Beziehung mit einem Topic, das somit die Funktion eines Occurrence Types einnimmt, spezifiziert und charakterisiert.

Entsprechend muss Tabelle neun wie nachfolgend erweitert werden, womit die Sammlung der Bausteine des Topic Maps-Tagging-Modells vervollständigt werden kann:

Topic Types	Association Types	Association Role Types	Occurrence Type
tag	assigned-by	user-to-tag, tag-to-user	date
taggable resource	saved-by	user-to-resource, resource-to-user	
tagging user	specified-by	tag-to-resource, resource-to-tag	
audio			
video			
image			
website			
any content			

Tab. 10: Topics des Topic Maps-Tagging-Modells und ihre Funktion

Tabelle 10 zeigt also die konstituierenden Topics des Topic Maps-Tagging-Modells. Daraus impliziert, dass das Topic Maps-Tagging-Modell als Topic Map strukturiert und auch als Topic Map spezifiziert werden kann (normative Funktion). In der konkreten Anwendung werden die (Meta)Daten entsprechend als Topic Map strukturiert. Somit ist festzustellen, dass das Modell sowie dessen Implementierung und Anwendung in ihrer formalen Beschaffenheit/Ausprägung/Struktur identisch sind, im Beispiel illustriert anhand der Syntax XTM:

- Modellelement [tag] `<topic id="tag-as-core-concept" />`
- Modellanwendung [XML] `<topic id="xml-as-a-concrete-tag" />`

Somit kann eine gleichartige Beziehungskonstellation bzgl. des Meta- und Datenmodells, das TMDM, festgestellt werden. Abbildung 26 visualisiert diese Beobachtung.

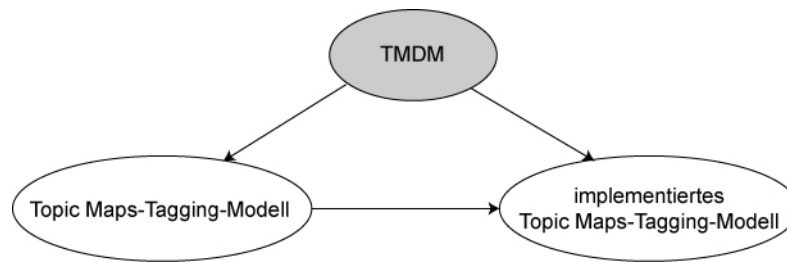
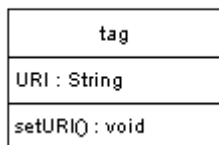


Abb. 26: Beziehungsdreieck zwischen TMDM, Topic Maps-Tagging-Modell und Anwendung des implementierten Topic Maps-Tagging-Modells

Zur weiteren Verdeutlichung dieses Gedankens kann das Verhältnis von UML-Klassendiagrammen¹⁴ und ihrer Umsetzung in Quellcode als ein gegenteiliges Beispiel dienen. Eine Klasse *tag* kann in einem UML-Klassendiagramm wie folgt dargestellt werden:



Die Umsetzung dieses Vorbildes in grafischer Notation z. B. mit PHP 5 sähe wie nachfolgend aus:

```

class tag{
    public $URI = null;
    public function setURI(){}
}
  
```

Modell und Implementierung sind also formal unterschiedlich, die Umsetzung weist keinen Bezug zum Metamodell des Klassendiagramms mehr auf. Es kann nun dahingehend argumentiert werden, dass die Beziehung des Topic Maps-Tagging-Modells zu seiner Implementierung eher der Beziehung der *schematic portion* und der *data portion* in Ontologien anzunähern ist. Das Topic Maps-Tagging-Modell definiert Entities, die Mengen von Entities repräsentieren und so dem erläuterten Begriff des Entity Types entspricht. Ebenso werden die Relationen zwischen Entities und weiter auch die Rollen in diesen

¹⁴ Im Zusammenhang der Modellierung ist zu erwähnen, dass die UML auf dem Entity-Relationship-Modell basiert (vgl. Kastens/Kleine Büning 2005, S. 165).

Relationen typisiert. An dieser Stelle ergibt sich ein Ansatzpunkt für die kritische Diskussion bzgl. des Terminus' „Modell“ in der Bezeichnung „Topic Maps-Tagging-Modell“.

So ist zu konstatieren, dass „Implementierung des Modells“ bedeutet, die einzelnen Elemente, z. B. *tag* oder *tagging user*, initial als Topics in einem System zu erstellen, um in der Anwendung des Modells konkrete Objekte, z. B. analog „XML“ oder „joha06“, als Instanzen zuweisen zu können (wodurch o. g. Topics zu Topic Types werden) und so die grundlegende Semantik für den Kontext des Taggings etablieren zu können (Was ist ein *tag*? Was ist ein *tagging user*? etc.).

Es sei der Vollständigkeit halber bemerkt, dass ein (schriftlicher und mündlicher) Diskurs nur mit Benennungen geführt werden kann. Entsprechend schließt die Deklaration der Topics jeweils ihre Benennung, die Topic Names, mit ein.

Um die Funktion der einzelnen Topics explizit auszudrücken, werden die spezifischen Beziehungstypen Topic Type, Association Type, Association Role Type und Occurrence Type, die Bestandteile bzw. Bausteine des Metamodells repräsentieren, als Topics definiert und die o. g. Topics über die Klasse-Instanz-Relation als Instanzen zugewiesen. Damit ist zusätzlich der Bezug zum Metamodell für die jeweiligen Konstrukte explizit ausgedrückt.

Die Definition der identifizierenden URLs für Instanzen von *taggable resource* als Subject Locators ist in der Argumentation der Modellierung dargelegt worden. Für Instanzen von *tag* und *tagging user* werden diese URLs als Subject Identifier definiert, da diese URLs auf eine Ressource verweisen, deren Inhalt auf die Identität des repräsentierten Aussagegegenstandes (ein bestimmter Nutzer einer Anwendung im Tagging-Kontext; ein Tag in einer Anwendung im Tagging-Kontext, das einen bestimmten Begriff repräsentiert) schließen lässt.

Somit sind alle notwendigen Bestandteile des Topic Maps-Tagging-Modells über die Abbildung als Topic Maps-Konstrukte definiert.

6.1.1 Etablieren der Semantik: Published Subject Identifiers und Published Subject Indicators

Auf die Bedeutung und Funktion veröffentlichter identifizierender Hinweise über Aussagegegenstände mit Hilfe von Ressourcen (Published Subject Indicators) und deren veröffentlichten Adressen (Published Subject Identifiers) ist in Kapitel 5.2 hingewiesen worden. Für die Implementierung und Verwendung des Topic Maps-Tagging-Modells ist die Definition von Published Subject Identifiers für die in Tabelle 10 aufgeführten Topics, die die zu identifizierenden Aussagegegenstände bzw. Konzepte repräsentieren, essentiell. Ohne deren Definition ist das Topic Maps-Tagging-Modell nicht eindeutig. D. h., dass die Semantik der konstituierenden Konzepte des Modells, mit denen die Objekte bzw. Instanzen (konkrete Nutzer, konkrete Tags) über die Klasse-Instanz-Relation in Beziehung gesetzt werden, zu definieren ist, womit fortführend auch der Bedeutungsrahmen dieser Instanzen bestimmt ist: Was ist ein Tag? Was meint die Relation „XML“ zu <http://www.example.org>?

Der Definitionsprozess, die Formulierung und Zuweisung, schafft die Voraussetzung dafür, die skizzierte Semantik des Taggings etablieren zu können.

Folglich ist aus der Perspektive der Standardisierung zu bemerken, dass nur die Zuweisung von Published Subject Identifiers die Verwendung als standardisiertes Modell bzw. als ein Vorschlag für ein derartiges Modell, das den Tagging-Kontext abbildet, in einem globalen Netzwerk, dem Internet, ermöglicht.

In Tabelle 11 sind die Published Subject Identifiers der verwendeten Konzepte, die durch Topics repräsentiert werden, aufgelistet. Dabei werden für Topic Type, Association Type, Association Role Type und Occurrence Type bereits im Topic Maps Data Model definierte Adressen (wieder)verwendet (vgl. ISO/IEC 13250-2), für das allgemeine Konzept „Datum“ wird auf die Definition für die Verwendung als Datentyp in XML Schema verwiesen.

Topic	Published Subject Identifier
tag	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tag-description
taggable resource	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-taggable-resource-description
tagging user	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tagging-user-description
audio	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-audio-description
video	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-video-description
image	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-image-description
any content	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-any-content-description
website	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-website-description
assigned-by	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-assigned-by-description
saved-by	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-saved-by-description
specified-by	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-specified-by-description
user-to-tag	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-user-to-tag-description
tag-to-user	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tag-to-user-description
user-to-resource	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-user-to-resource-description
resource-to-user	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-resource-to-user-description
tag-to-resource	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tag-to-resource-description
resource-to-tag	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-resource-to-tag-description
date	http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/datatypes.html#date
Topic Type	http://psi.topicmaps.org/iso13250/glossary/topic-type
Association Type	http://psi.topicmaps.org/iso13250/glossary/association-type
Association Role Type	http://psi.topicmaps.org/iso13250/glossary/association-role-type
Occurrence Type	http://psi.topicmaps.org/iso13250/glossary/occurrence-type
english	http://www.topicmaps.org/xtm/1.0/language.xtm#en
Topic Maps-Tagging-Model	http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tmodel-description

Tab. 11: Published Subject Identifiers der Topics des Topic Maps-Tagging-Modells

Die Adressen für die Topics *english* und *Topic Maps-Tagging-Model* sind zur Vollständigkeit mit aufgeführt, da diese für die Spezifikation des Modells (siehe Kapitel 6.1.2) relevant sind.

Neben der Offenlegung eindeutiger Adressen sollen die in dieser Arbeit entwickelten Konzepte des Tagging-Modells auch in ihrer Semantik bzw. Bedeu-

tung definiert werden, also Indikatorressourcen mit der Intention, diese als Published Subject Indicators einzusetzen, erarbeitet werden. Entsprechend der Empfehlung von Vatant (siehe Kapitel 5.2) werden die Begriffsdefinitionen in einer Topic Map strukturiert und in Syntax XTM 1.0 formuliert. Entsprechend sind die Verweise auf die Datei tmodelpsi.xtm in Tabelle 11 zu interpretieren. XTM-Listing sechs zeigt exemplarisch die Definition von *tag*, deklariert als gültige Topic Map.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE topicMap PUBLIC "-//TopicMaps.Org//DTD XML Topic Map (XTM)
1.0//EN" "http://topicmaps.org/xtm/1.0/xtml.dtd">
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/"
xmlns:xlink="http://www.w3.org/1999/xlink" id="tmodelpsi.xtm">
  <topic id="tag">
    <subjectIdentity>
      <subjectIndicatorRef xlink:href="#psi-tag-
description"/>
    </subjectIdentity>
    <baseName>
      <scope>
        <topicRef xlink:href="#english"/>
      </scope>
      <baseNameString>tag</baseNameString>
    </baseName>
    <occurrence>
      <resourceData id="psi-tag-description">
        A tag is a keyword that is assigned to a taggable
resource by a tagging user. Assigning a tag is the result of an
intellectual indexing process realised by a tagging user. A tag names
any subject/concept/idea (representing a real world or an abstract
thing) the tagging user keeps in mind when examining a resource.
      </resourceData>
    </occurrence>
  </topic>
</topicMap>
```

XTM-Listing 6: Definition von *tag* innerhalb der Published Subject Indicators in XTM 1.0 für die zentralen Konzepte des Topic Maps-Tagging-Modells

Die vollständige Topic Map, die die Published Subject Indicators der Konzepte des Topic Maps-Tagging-Modells repräsentiert, ist dieser Arbeit als Anhang I beigefügt.

Es ist bewusst XTM 1.0 als Syntax gewählt worden, da auf Grund des dargestellten Status' im Standardisierungsprozess bzgl. der Syntax verfügbare Anwendungen bzw. Parser ausschließlich XTM 1.0 verarbeiten, z. B. der XTM-Parser des TMAPI-Utills-Paketes (<http://tmapi-utils.sourceforge.net>). Diese Begründung gilt ebenfalls für die im folgenden darzustellende Spezifikation des Topic Maps-Tagging-Modells. Entsprechend konnten in der Modellierung Konstrukte wie Topic Name Types nicht verwendet werden. Diese sind in diesem Zusammenhang auch nicht als unterstützend bzw. sinnvoll zu betrachten.

6.1.2 Spezifikation des Tagging-Modells

In Kapitel 6.1 ist das Topic Maps-Tagging-Modell textuell bzw. in Prosa beschrieben und definiert worden. Damit dieses Modell in Topic Maps-Anwendungen bzw. -Systemen korrekt (gemäß der Intention) und als normatives Vorbild angewendet werden kann, muss es spezifiziert werden. Das bedeutet, dass formal definiert werden muss, wie das Modell auf konkreter Datenebene umzusetzen ist. Dieser Prozess beinhaltet die Festlegung,

- welches Topic Maps-Konstrukt ein jeweiliges Konzept repräsentiert,
- wie ein jeweiliges Konzept bzw. seine Repräsentation ggf. benannt ist,
- wie ein Konzept ggf. identifiziert ist,
- wie die Topic Maps-Konstrukte zueinander in Beziehung stehen und
- wie die Beziehungen zwischen den Topic Maps-Konstrukten beschaffen sind.

Wie erwähnt, kann das Modell als Topic Map spezifiziert werden. Eine Syntax für den Austausch von Topic Maps ist XTM. Die Syntax ermöglicht also das „Explizitmachen“ einer Topic Map als Ergebnis einer Serialisierung und wird deshalb in dieser Arbeit zur Spezifikation des Topic Maps-Tagging-Modells verwendet.

Nachfolgend wird die Spezifikation von *tag* in XTM 1.0 exemplarisch gezeigt:

```
<topic id="topic-93">
  <instanceOf>
    <topicRef xlink:href="#topic-89"/>
  </instanceOf>
  <subjectIdentity>
```

```

        <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tag-
description"/>
    </subjectIdentity>
    <baseName id="topicname-38">
        <scope>
            <topicRef xlink:href="#topic-84"/>
        </scope>
        <baseNameString>tag</baseNameString>
    </baseName>
</topic>

```

XTM-Listing 7: Definition von *tag* innerhalb der Spezifikation des Topic Maps-Tagging-Modells in XTM 1.0

Der in Kapitel 6.1.1 für *tag* definierte Subject Identifier ist zugewiesen und verweist auf den Subject Indicator als einen eindeutigen Hinweis auf die Identität dieses Aussagegegenstandes bzw. Konzeptes, wodurch folglich die Bedeutung dieses Konzeptes öffentlich geklärt und so eindeutig „gemacht“ werden kann: „Was ist ein *tag*?“ bzw. „Was ist mit *tag* gemeint?“

Da *tag* als Topic Type für konkrete Tags definiert ist bzw. derart verwendet werden soll, ist dies explizit über die Klasse-Instanz-Relation zu einem Topic *Topic Type* ausgedrückt (siehe `instanceOf-` (*parent*) und `topicRef`-Element (*child*)), das über einen entsprechend definierten Subject Identifier, der auf einen Subject Indicator verweist, als das dieses Topic Maps-Konstrukt repräsentierende Topic identifiziert wird:

```

<topic id="topic-89">
    <subjectIdentity>
        <subjectIndicatorRef
xlink:href="http://psi.topicmaps.org/iso13250/glossary/topic-type"/>
    </subjectIdentity>
    <baseName id="topicname-34">
        <scope>
            <topicRef xlink:href="#topic-84"/>
        </scope>
        <baseNameString>Topic Type</baseNameString>
    </baseName>
</topic>

```

XTM-Listing 8: Definition von *Topic Type* innerhalb der Spezifikation des Topic Maps-Tagging-Modells in XTM 1.0

Für die Benennungen aller Topics ist ein Scope bestimmt, der durch ein Topic mit dem Topic Name „english“ definiert ist. Also ist die jeweilige Benennung gültig in der englischen Sprache, d. h. die einzelnen Wörter Bestandteil dieses Vokabulars (siehe auch PSI für *english* in Kapitel 6.1.1).

Des weiteren sind die Relationen von *tag* zu *tagging user* und *taggable resource*, die über das Association-Konstrukt abgebildet werden, zu spezifizieren. Dafür sind die in Kapitel 6.1 erarbeiteten Association Types und Association Role Types einzusetzen. Anhand des bereits vorgestellten Aussagenmodells aus einem Entwurf zum Topic Maps Reference Model, Revision 1.0, wird die Association zwischen *tag* und *taggable resource* in Abbildung 27 veranschaulicht.

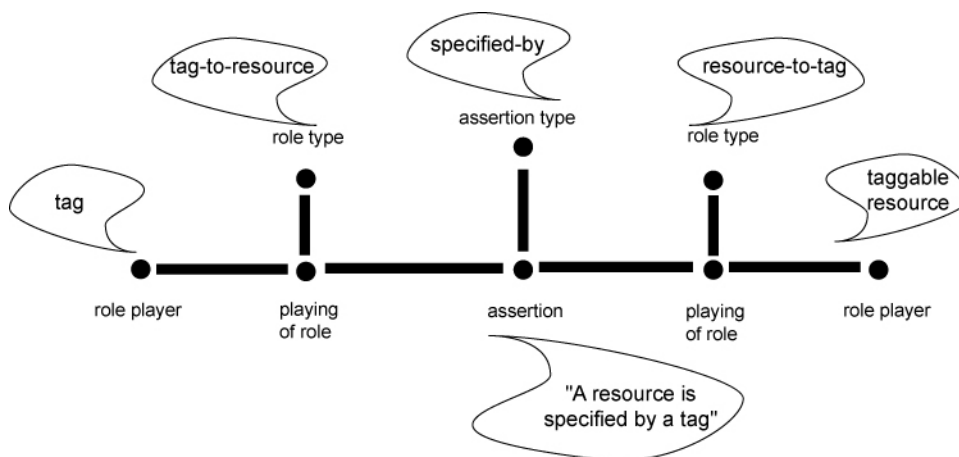


Abb. 27: Visualisierung der Association zwischen *tag* und *taggable resource* unter Verwendung eines Aussagenmodells im Draft Topic Maps Reference Model, Revision 1.0

Entsprechend ließen sich die Relationen zwischen *tag* und *tagging user* sowie *tagging user* und *taggable resource* visuell darstellen und die Intentionen der Beziehungen, die jeweils eine Aussage mit einschließt bzw. diese ermöglicht, verdeutlichen.

Abbildung 28 zeigt das Topic *tag* und seine Beziehungen innerhalb der in XTM 1.0 formulierten Topic Map, die das Topic Maps-Tagging-Modell spezifiziert, visualisiert mit dem Topic Maps-Browser „Omnigator“ des norwegischen Unternehmens Ontopia AS (<http://www.ontopia.net>), der zusammen mit der funktional limitierten „Ontopia Knowledge Suite“ zum Zweck der Evaluation kostenlos heruntergeladen werden kann. Weitere Informationen zu dieser Software, ihrer Installation und Verwendung können unter <http://www.ontopia.net/download/freedownload.html> nachgeschlagen werden.

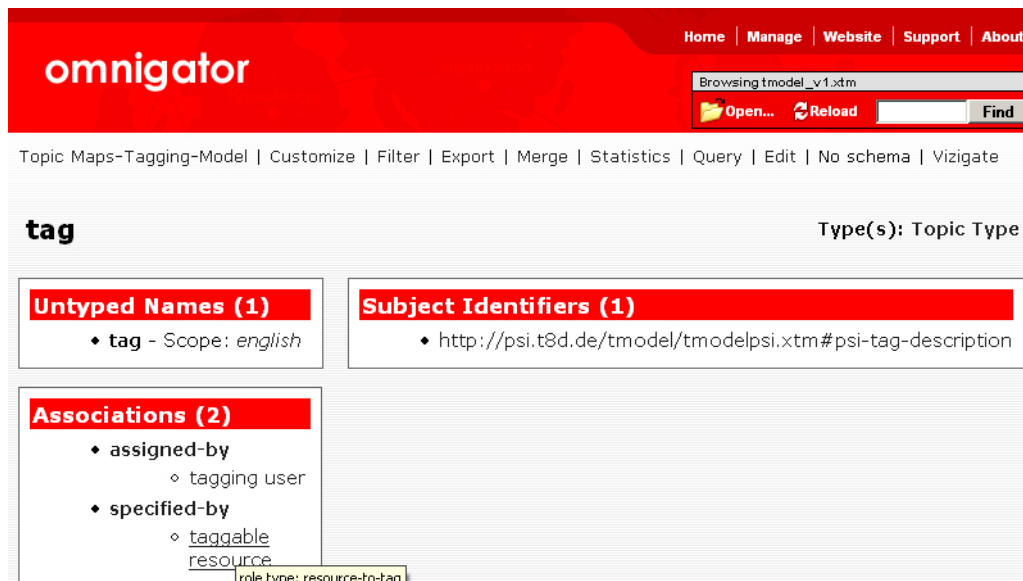


Abb. 28: Darstellung des Topics *tag* innerhalb des Topic Maps-Tagging-Modells mit dem Topic Maps-Browser „Omnigator“

Diese Ansicht in Abbildung 28 zeigt also die (visualisierte) Spezifikation des Konzeptes *tag* mit Topic Maps.

Die vollständige Spezifikation des Topic Maps-Tagging-Modells in XTM 1.0 ist als Anhang J und als Datei `tmodelspec.xtm` im Verzeichnis `xtm` auf der zugehörigen CD-ROM dieser Arbeit beigefügt, so dass dieses Modell mit Topic Maps-Software evaluiert werden kann.

6.2 Implementierung des Modells

Wie eingangs des Kapitels 5.7 bereits geäußert, ermöglichen Implementierung und Anwendung bzw. Verwendung des Topic Maps-Tagging-Modells unter Einsatz der vorgestellten Implementierung von PHPTMAPI die Überprüfung der Anwendbarkeit und Praxistauglichkeit der theoretischen Überlegungen.

Die Spezifikation des Topic Maps-Tagging-Modells definiert die Umsetzung. Da die Spezifikation selbst eine Topic Map ist, bedeutet dies für die Implementierung des Modells, diese Topic Map im PHPTMAPI-System abzubilden. Über das API ist dieses Modell dann innerhalb einer Applikation verwendbar, wie im anschließenden Kapitel gezeigt wird. Die Implementierung des Modells beinhaltet also die Abbildung der in XTM-Syntax formulierten Topic Maps-Konstrukte und ihrer Eigenschaften über PHPTMAPI oder TMAPI für Java: `<topic>` wird abgebildet mit dem Interface `TopicInterface` (PHPTMAPI) bzw.

dem Interface `Topic` (TMAPI) und entsprechend für alle anderen verwendeten Konstrukte.

Die Implementierung von PHPTMAPI alias QuaaXTM im Verzeichnis `quaaxtm/quaaxtm` auf der CD-ROM, im folgenden Software genannt, wird anhand der in Anhang K aufgeführten Installationshilfe installiert. Voraussetzung für die Lauffähigkeit sind ein Webserver, z. B. Apache, mit PHP 5.x als Modul oder CGI eingebunden sowie ein MySQL-Server mit InnoDB als verfügbare *storage engine*. Die API-Dokumentation der Software befindet sich im Verzeichnis `quaaxtm/quaaxtm_apidocs`¹⁵ auf der CD-ROM.

Die Erstellung der Topics und Associations, die das Topic Maps-Tagging-Modell repräsentieren, erfolgt in zwei Schritten, die jeweils in den PHP-Skripten `build_tmodel_basic.php` und `build_tmodel_assocs.php` implementiert sind. Diese Dateien befinden sich nach der Installation anhand der Anleitung im Verzeichnis `/quaaxtm/tmodel` auf dem Webserver oder auf der CD-ROM im Verzeichnis `quaaxtm/quaaxtm/tmodel` (siehe auch Anhang K).

Zunächst wird das Systemobjekt erstellt, wie es bereits in PHP-Listing 4 auf Seite 96 gezeigt worden ist. Im erstgenannten Skript, `build_tmodel_basic.php`, ist initial eine Topic Map bzw. ein Topic Map-Objekt zu erstellen, welches das Topic Maps-Tagging-Modell repräsentiert und in der die (Meta)Daten später gespeichert werden:

```
$tm = $tmSystem->createTopicMap('http://psi.t8d.de/tmodel/');
```

Der Parameter „baselocatorref“, der den im TMDM nicht definierten Base Locator einer Topic Map darstellt, kann als Speicherort der Topic Map aufgefasst werden, an dem die Topic Map z. B. in XTM-Syntax vorgehalten wird.

Mit dem Topic Map-Objekt `$tm` können nun Operationen für diese Topic Map ausgeführt werden. Neben der Definition eines Topics *english*:

```
$t = $tm->createTopic();  
$t->  
>addSubjectIdentifier('http://www.topicmaps.org/xtm/1.0/language.xtm#e  
n') ;  
$t->createUntypedTopicName('english', null);
```

¹⁵ Die API-Dokumentation kann alternativ auch unter <http://quaaxtm.sourceforge.net/apidocs/> nachgeschlagen werden.

das den sprachlichen Kontext, Scope, der weiter anzulegenden Topics bestimmt:

```
$namespace = array($t);
```

wird nachfolgend ein Topic angelegt, das der Topic Map als „Reifiable Object“ bzw. *reifier* zugewiesen wird:

```
$t = $tm->createTopic();  
$t->createUntypedTopicName('Topic Maps-Tagging-Model', $namespace);  
$t->addItemIdentifier('http://psi.t8d.de/tmodel/tmodeltopic');  
$tm->setReifier($t);
```

Somit kann der Topic Map eine Bezeichnung/ein Name, in diesem Fall „Topic Maps-Tagging-Model“, zugeordnet werden. Des Weiteren werden Metadaten, die im Bedarfsfall nach Dublin Core abgebildet werden können, für die das Topic Maps-Tagging-Modell repräsentierende Topic Map erstellt, im Beispiel für den Ersteller der Topic Map:

```
$tcr = $tm->createTopic();  
$tcr->addSubjectIdentifier('http://purl.org/dc/elements/1.1/creator');  
$tcr->createUntypedTopicName('Creator', null);  
$occ = $t->createOccurrence('Johannes Schmidt, HAW Hamburg',  
'http://www.w3.org/TR/xmlschema-2/#string', $tcr, null);
```

Anschließend werden die Elemente des Metamodells repräsentiert, um die Funktion der die zentralen Konzepte repräsentierenden Topics explizit auszudrücken, Beispiel Topic Type:

```
$t = $tm->createTopic();  
$t->  
>addSubjectIdentifier('http://psi.topicmaps.org/iso13250/glossary/topic-type');  
$t->createUntypedTopicName('Topic Type', $namespace);
```

Dann können die die zentralen Konzepte repräsentierenden Topics angelegt werden, im Beispiel *tag*, nachdem jenes das Topic Type-Konstrukt repräsentierende Topic-Objekt erstellt worden ist, um dieses als Type zuweisen zu können:

```

$ttype = $tm->
getTopicBySubjectIdentifier('http://psi.topicmaps.org/iso13250/glossar
y/topic-type');
$t = $tm->createTopic();
$t->addSubjectIdentifier('http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-
tag-description');
$t->createUntypedTopicName('tag', $namespace);
$t->addType($ttype);

```

In einem zweiten Schritt, implementiert in `build_tmodel_assocs.php`, werden die Associations gemäß der Spezifikation erzeugt. Dazu werden zunächst die das Modell konstituierenden Topics bzw. Topic-Objekte angelegt. Dieser Prozess bedeutet bereits eine konkrete Verwendung des Modells.

Folgender Auszug zeigt, wie die Association zwischen *tag* und *tagging user* etabliert wird:

```

$ttag = $tm-
>getTopicBySubjectIdentifier('http://psi.t8d.de/tmodel/tmodelpsi.xtm#p
si-tag-description');
$tuser = $tm-
>getTopicBySubjectIdentifier('http://psi.t8d.de/tmodel/tmodelpsi.xtm#p
si-tagging-user-description');
$tassignedby = $tm-
>getTopicBySubjectIdentifier('http://psi.t8d.de/tmodel/tmodelpsi.xtm#p
si-assigned-by-description');
$tttu = $tm-
>getTopicBySubjectIdentifier('http://psi.t8d.de/tmodel/tmodelpsi.xtm#p
si-tag-to-user-description');
$tutt = $tm-
>getTopicBySubjectIdentifier('http://psi.t8d.de/tmodel/tmodelpsi.xtm#p
si-user-to-tag-description');

$assoc = $tm->createAssociation();
$assoc->setType($tassignedby);
$role = $assoc->createAssociationRole($ttag, $tttu);
$role = $assoc->createAssociationRole($tuser, $tutt);

```

Nach dem Ausführen beider Skripte ist das Topic Maps-Tagging-Modell in einem PHPTMAPI-System implementiert und kann über das standardisierte API angewendet bzw. verwendet werden.

Eine Alternative für die Implementierung des Modells stellt die in Kapitel 5.4 skizzierte Deserialisierung dar. In der praktischen Durchführung hieße dies, die Spezifikation in XTM-Syntax mit einem Parser zu lesen und diese in eine implementierungsinterne Repräsentation des TMDM zu übersetzen/abzubilden. Ein solcher Prozess wäre für TMAPI-Implementierungen möglich, da das erwähnte TMAPI-Utills-Paket einen XTM-Parser, der wiederum selbst die TMAPI-Implementierung benutzt, bereitstellt. So ist die XTM-Spezifikation des Topic Maps-Tagging-Modells zusätzlich für die konkrete, d. h. technische Implementierung einsetzbar.

6.3 Anwendung des Modells

Nach der Implementierung, die im vorigen Kapitel erläutert worden ist, kann folgender Status resümiert werden:

- Die Technologie für die zu speichernden (Meta)Daten, die konkreten Ausprägungen, ist mit Topic Maps standardisiert und kann innerhalb eines Systems über ein standardisiertes API, PHPTMAPI, angesprochen werden;
- Das skizzierte Parallelmodell zur Speicherung der (Meta)Daten ist spezifiziert (struktureller Aspekt);
- Die Bestandteile des Modells, die Konzepte, sind mit Hilfe von Published Subject Indicators identifiziert (semantischer Aspekt).

Wie gestaltet sich nun die konkrete Anwendung? Im folgenden Beispiel wird ein Anwendungsfall für eine Online-Linkverwaltung unter <http://www.example.org> mit den Nutzern „joha06“ und „andyman3000“ simuliert. Beide User versehen eine Ressource, die Webseite <http://www.topicmaps.org>, mit unterschiedlichen Benennungen, „XML Syntax for Topic Maps“ sowie „XTM Homepage“, und unterschiedlichen Tags, „XML“ und „TopicMaps“. Dieses Beispiel ist im PHP-Skript `use_tmodel_1.php` umgesetzt, das sich nach der Installation im Verzeichnis `quaaxtm/tmodel` des Webservers befindet, ausgehend vom Document-Root-Verzeichnis.

Analog zum Procedere in `build_tmodel_assocs.php` werden dort zunächst die das Topic Maps-Tagging-Modell repräsentierenden Topics bzw. Topic-Objekte erzeugt, um diese als Types zuweisen zu können.

Nutzer repräsentierende Topics haben einen Subject Identifier, z. B. <http://www.example.org/joha06> für „joha06“ (korrespondierend zu del.icio.us mit <http://del.icio.us/joha06>), ebenso Tags repräsentierende Topics, z. B. <http://www.example.org/tag/XML> für „XML“ (korrespondierend zu <http://del.icio.us/tag/XML>). Ressourcen repräsentierende Tags nehmen den URI bzw. URL der Ressource als Subject Locator auf: die Ressource ist der Aussagegegenstand des Topics.

Mit diesen Informationen werden Topics und Associations gemäß der Spezifikation angelegt und typisiert (siehe `use_tmodel_1.php`).

Für die Evaluation dieses Beispiels, z. B. mit Omnigator, muss die gesamte Topic Map in Syntax XTM 1.0 exportiert werden (Serialisierung). Dafür hat der Autor eine Klasse `XTMSerializer` entwickelt, welche die PHPTMAPI-Implementierung sowie die DOM-Erweiterung für PHP benutzt. Somit handelt es sich hierbei um eine Anwendung der PHPTMAPI-Implementierung. Diese Klasse ist auf der beigelegten CD-ROM im Verzeichnis `quaaxtm/quaaxtm/src/phptmapi/phptmapiutils/impexp` enthalten und kann nach der Installation auf einem Webserver (Bsp. <http://localhost/>) über das Skript http://localhost/quaaxtm/tmodel/serialize_tmodel.php für die Topic Map mit dem definierten Base Locator „<http://psi.t8d.de/tmodel/>“ direkt angewendet werden.

`XTMSerializer` ist noch nicht Open Source verfügbar, soll jedoch langfristig analog in einem dem TMAPI-Utills-Paket entsprechenden Paket frei zur Verfügung gestellt werden.

Das gültige XTM ist nach dem Aufrufen über die Ansicht des Quelltextes im Browser zu sehen und kann an dieser Stelle als Datei gespeichert werden.

Auf die nicht-kongruenten Datenmodelle, das TMDM und das für XTM 1.0 spezifizierte, und die resultierenden Schwierigkeiten ist hingewiesen worden. So beschränkt diese Klasse bspw. die Ausgabe von Referenzen auf adressierbare Aussagegegenstände auf eine Referenz (z. B. einen URL), da das Modell von XTM 1.0 nur eine Referenz (vgl. TopicMaps.org 2001), das TMDM jedoch mehrere zulässt (vgl. ISO/IEC 13250-2). Da im Topic Maps-Tagging-Modell keine zur Spezifikation von XTM 1.0 inkompatiblen Konstrukte wie Topic Name Types definiert worden sind, kann die Serialisierung so weit konsistent erfolgen. Abbildung 29 zeigt die Visualisierung der von mehreren Nutzern „getaggten“ Ressource bzw. deren Referenz <http://www.topicmaps.org>.

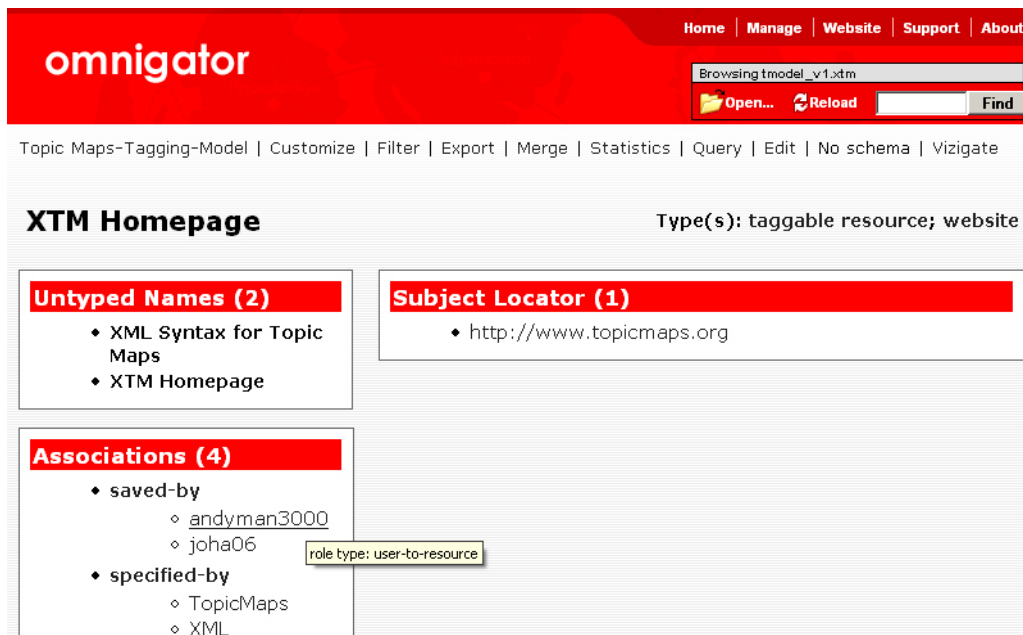


Abb. 29: Darstellung einer Ressource in einer Anwendung des Topic Maps-Tagging-Modells mit dem Topic Maps-Browser „Omnigator“

Das Prinzip des Mergings wird an dieser Stelle sichtbar: Nach der ursprünglichen Definition von zwei Topics in `use_tmodel_1.php` (Zeile 32-36 und 58-62) resultiert im Ergebnis über die Angabe eines identischen URLs als Subject Locator ein Topic mit mehreren Benennungen, zu sehen im linken oberen Bereich, das in mehreren Associations mit Instanzen, also konkreten Ausprägungen, von *tagging user* und *tag* verknüpft ist, zu erkennen im linken unteren Bereich. Dabei ist die Ressource eine Instanz von *taggable resource* und *website*, zu sehen im rechten oberen Bereich.

Mit dem in der Ontopia Knowledge Suite integrierten „Vizigator“, ein Java-Applet, lässt sich der Zusammenhang grafisch visualisieren, wie Abbildung 30 zeigt.

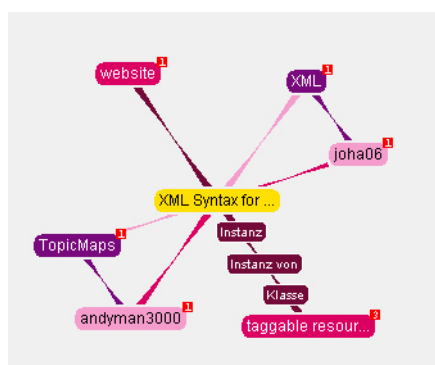


Abb. 30: Grafische Darstellung einer Ressource in einer Anwendung des Topic Maps-Tagging-Modells mit „Vizigator“

Das gezeigte Beispiel ist in der Datei `use_tmodel_2.php` um weitere Tags und User sowie andere Tag-User-Ressourcen-Konstellationen erweitert. In dieser Erweiterung wird die Ressource unter `http://www.topicmaps.org` zweimal mit dem Tag „XML“ von zwei unterschiedlichen Nutzern belegt, woraus zwei identische Associations zwischen der Ressource unter `http://www.topicmaps.org` und dem Tag „XML“ resultieren. Die textbasierte Visualisierung in Omnigator zeigt dieses Resultat als jeweilige Doppelbelegung, die unglücklich scheint, wie Abbildung 31 für die Ressourcenperspektive zeigt.

XTM Homepage Type(s): taggable resource; website

Untyped Names (4)

- ◆ A syntax for Topic Maps
- ◆ The XTM page
- ◆ XML Syntax for Topic Maps
- ◆ XTM Homepage

Subject Locator (1)

- ◆ <http://www.topicmaps.org>

Associations (8)

- ◆ saved-by
 - ◊ andyman3000
 - ◊ joha06
 - ◊ terry15
 - ◊ tonitest
- ◆ specified-by
 - ◊ SemanticWeb
 - ◊ TopicMaps
 - ◊ XML
 - ◊ XML

role type: tag-to-resource

Abb. 31: Darstellung der Associations in „Omnigator“ für die Ressource unter `http://www.topicmaps.org` nach Mehrfachzuweisung eines identischen Tags

Der „Vizigator“ löst diesen Zusammenhang übersichtlicher auf, wie in Abbildung 32 auf der linken unteren Seite zu erkennen ist.

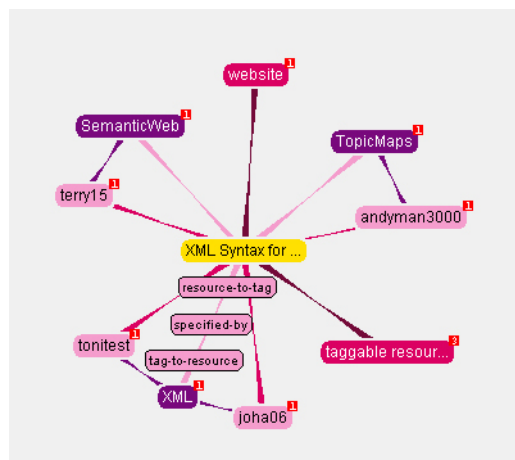


Abb. 32: Darstellung der Associations in „Vizigator“ für die Ressource unter `http://www.topicmaps.org` nach Mehrfachzuweisung eines identischen Tags

Die Topic Maps in XTM 1.0-Syntax der jeweils nach dem Ausführen der in den Dateien `use_tmodel_1.php` und `use_tmodel_2.php` implementierten Beispiele sind auf der beigelegten CD-ROM im Verzeichnis `xm` als Dateien `use_tmodel_1.xtm` und `use_tmodel_2.xtm` gespeichert.

Im Zusammenhang mit Überlegungen zu Einsatzszenarios ist zu überlegen, ein API für Tagging-Kontext-spezifische Anwendungen zu entwickeln, die eine PHPTMAPI-Implementierung benutzen und so den Topic Maps-Hintergrund des Modells maskieren, was zur Reduzierung von Komplexität und zur Vereinfachung der Applikationsentwicklung beitragen kann. So könnte der Prozess des Taggings, die Zuweisung eines Tags zu einer Ressource durch einen Nutzer, z. B. wie folgt vereinfacht werden:

```
$taggingmodel = getTaggingModel('models_base_locator');  
$taggingmodel = doTagging('a_users_locator', 'a_tags_locator',  
'a_resources_locator');
```

7 Bewertung und Fazit

In den folgenden, diese Arbeit abschließenden Ausführungen sollen die Ergebnisse und die These bewertet werden. Dabei sind konsequente Schlussfolgerungen zu ziehen und resultierende Arbeitsansätze anzudeuten.

Die Implementierung und Anwendung des Topic Maps-Tagging-Modells haben gezeigt, dass die Strukturierung und Speicherung von Daten, die für den Kontext des Taggings relevant sind, mit Topic Maps zu realisieren sind. Gemäß der Erarbeitungen stellen diese Daten Metadaten dar, die die dargestellte Semantik des Taggings reflektieren. Mit der Spezifikation des Modells einerseits und der Verwendung des Standards ISO/IEC 13250 Topic Maps andererseits kann das Standardisierungspotential für den strukturellen Aspekt zugesprochen werden. Das Topic Maps-Tagging-Modell ist in diesem Zusammenhang ein Vorschlag.

Für die Skizze eines Einsatzszenarios kann darauf aufbauend festgestellt werden, dass die Implementierung und Anwendung innerhalb existierender Systeme, konkret: Webserver, praktisch möglich ist: PHPTMAPI definiert die Programmierschnittstelle zur internen Anbindung, die vorgestellte Implementierung alias QuaaxTM führt die Operationen konkret aus. Das Topic Maps-Tagging-Modell definiert die inhaltliche Struktur. Mit PHPTMAPI ist die problemlose Austauschbarkeit von Implementierungen wie QuaaxTM durch andere, ggf. leistungsfähigere möglich, da nur das API, PHPTMAPI, in der Programmierung zu berücksichtigen ist. Hier wird die Bedeutung von PHPTMAPI noch einmal deutlich. Hinsichtlich des Standardisierungspotentials bzw. der Beantwortung der Frage, ob PHPTMAPI das Standard-Topic Maps-API für PHP ist, kann zum jetzigen Zeitpunkt zunächst nur registriert werden, dass einerseits ein alternatives Projekt dem Autoren und den involvierten Mitgliedern der Community nicht bekannt ist, andererseits die Anzahl der Downloads des Quellenpaketes seit dem 27.6.2006 104 beträgt (Stand 8.12.2006).

Diese Skizze berücksichtigt ein singuläres System, einen einzelnen Webserver. Für den Aspekt der systemübergreifenden Datenintegration und -aggregation sind Schnittstellen und Mechanismen für den Austausch notwendig, die idealerweise innerhalb von ISO/IEC 13250 Topic Maps definiert sein sollten. Aktuell existiert eine solche Komponente in ISO/IEC 13250 nicht. Garshol hat mit TMRAP (Topic Maps Remote Access Protocol) einen Vorschlag für ein abstraktes Web Service Interface in 2005 vorgelegt (vgl.

Garshol 2005, S. 1). TMRAP soll in der kommerziellen „Ontopia Knowledge Suite“ implementiert werden (vgl. Garshol 2005, S. 14). TMRAP berücksichtigt das Ausdruckspotential von Topic Maps, indem es Anfragen in der von Ontopia entwickelten Abfragesprache tolog ermöglicht. So liefert die tolog-Anfrage

```
select $USER, $TAG from
i"http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-assigned-by-description"
($USER: i"http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-user-to-tag-
description",
$TAG: i"http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tag-to-user-
description")
order by $TAG?
```

alle Tags und User, die über Rollen in einer Association mit dem Association Type „assigned-by“ verknüpft sind. Somit wären ganz individuelle Anfragen an Systeme möglich, die z. B. das Topic Maps-Tagging-Modell implementieren, wie das Beispiel zeigt. Garshol legt als mögliches Rückgabeformat neben XTM LTM, AsTMA, TM/XML und tolog fest (vgl. Garshol 2005, S. 12).

Eine detailliert ausgearbeitete Alternative stellt das Topic Maps Interaction Protocol (TMIP) von Barta dar, das einen REST-Ansatz verfolgt und Datenmanipulationsoperationen erlaubt (vgl. Barta 2005, S. 11). Ein entscheidender Punkt ist, dass TMIP die Topic Maps Query Language (TMQL) berücksichtigt, die Teil des Topic Maps-Standards ist, jedoch noch als Entwurf vorliegt (siehe <http://www.isotopicmaps.org/tmql/spec.html>).

Der Aspekt der systemübergreifenden Interaktion ist ein drängendes Thema und wird in der Topic Maps-Community entsprechend intensiv diskutiert. Auf der TMRA 2006 sind fünf Implementierungen für Topic Maps-Webservices vorgestellt worden, was zugleich den Auslöser für die Standardisierung eines Topic Maps Web Service Protokolls bedeutete (Maicher 2006b). Moore informierte nach einem ISO-Treffen der Topic Maps-Arbeitsgruppe über die Topic Maps-E-Mailliste (<http://www.infoloom.com/mailman/listinfo/topicmapmail>) im Oktober 2006, dass ein solches Projekt nicht innerhalb der ISO, sondern innerhalb der Community ähnlich wie TMAPI entwickelt werden sollte (vgl. Anhang L) und initialisierte das Projekt „Topic Maps Web Service Interface“ (TMWSI) bei sourceforge.net (<http://sourceforge.net/projects/tmws>). Damit scheint die Möglichkeit, Topic Maps-basierte Datenintegration- und

aggregation auf Basis eines standardisierten Web Services zu implementieren, lediglich noch eine Frage der Zeit zu sein.

Für den Aspekt der Rückgabeformate hat Garshol die Vielfalt genannt (s. o.), wobei aber gerade die Einheitlichkeit des Formats wesentlich für die Effizienz in Datenaustauschprozessen ist und XTM als Bestandteil des Standards deshalb eine zentrale Bedeutung zukommt.

Des Weiteren ist eine Herausforderung darin zu sehen, den Umfang von Topic Maps- bzw. XTM-Fragmenten als Resultat einer Anfrage zu bestimmen bzw. bestimmen zu können, da ein Topic durch andere Topics bzw. spezifische Beziehungen definiert werden kann, z. B. Topic Types, Occurrence Types, Association Types, Association Role Types u. a. Für TMRAP hat Garshol bspw. einen View-Parameter definiert, über den der Umfang des zurückgegebenen Fragments bestimmt werden kann (vgl. Garshol 2005, S. 4). Ein solcher Mechanismus ist als Anforderung an ein TMWSI zu definieren, um die Datenmenge für den jeweiligen Bedarf optimieren zu können.

Abbildung 33 zeigt als eine Zusammenfassung die Skizze eines Remote Access-fähigen Topic Maps-Systems für das Topic Maps-Tagging-Modell, das die vorgestellten Komponenten integriert.

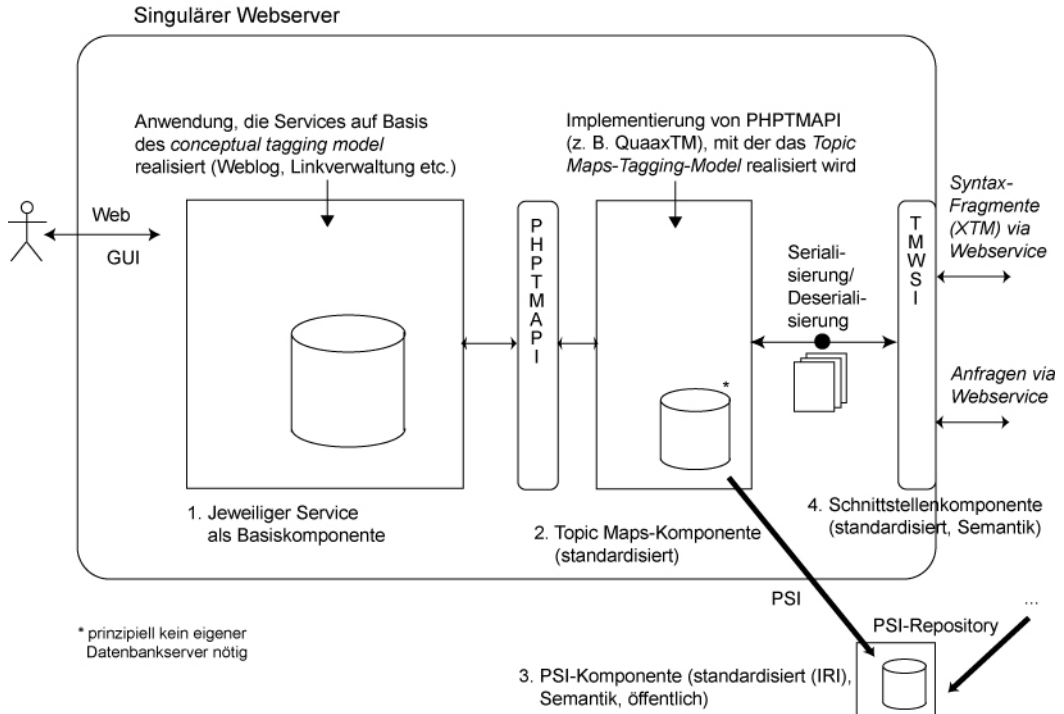


Abb. 33: Skizze eines Remote Access-fähigen Topic Maps-Systems für das Topic Maps-Tagging-Modell und Anwendung mit PHPTMAPI-Implementierungen

Ein für die nahe Zukunft resultierender Arbeitsansatz bzgl. PHP und Topic Maps bildet die Implementierung des TMWSI sowie die Entwicklung eines XTM-Parsers, mit dem die XTM-Fragmente als Ergebnis einer Anfrage über eine TMWSI-Schnittstelle gelesen und in eine Topic Map auf einem Zielsystem geschrieben werden können (Deserialisierung).

Nach der Bewertung und Betrachtung des strukturellen Aspektes ist der semantische Aspekt zu beurteilen. Das tolog-Beispiel bildet hier einen anschaulichen Zugang. Die Abfrage nach Tags und User erfolgte über die Subject Identifier bzw. Published Subject Identifier der Topics des Topic Maps-Tagging-Modells. D. h., dass die Abfrage über die Proxies von Aussagegegenständen bzw. deren Adressen definiert worden ist und so immer eine Menge an Ausprägungen der in <http://psi.t8d.de/tmodel/tmodelpsi.xtm> determinierten Konzepte bzw. Bestandteile des Tagging-Modells liefert, unabhängig ihrer Benennungen. Die obige tolog-Query bzw. ein TMQL-Äquivalent könnte nach einmaliger Definition an eine beliebige Implementierung des in dieser Arbeit entwickelten Topic Maps-Tagging-Modells gesendet werden und lieferte immer ein semantisch identisches und somit verlässliches Ergebnis: eine Menge an Ausprägungen von Objektmengen, die von Topics repräsentiert und in <http://psi.t8d.de/tmodel/tmodelpsi.xtm> in ihrer Bedeutung definiert sind. Für <http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-user-to-tag-description>: „User-to-tag specifies the nature of the participation of the role player "tagging user" in an association specified by 'assigned-by'" (siehe auch Anhang I-5).

Unter <http://psi.t8d.de/tmodel/> bzw. innerhalb der in XTM in [tmodelpsi.xtm](http://psi.t8d.de/tmodel/tmodelpsi.xtm) formulierten Topic Map ist das PSI-Repository für das Topic Maps-Tagging-Modell angelegt, auf das alle Subject Identifier der Topics des Topic Maps-Tagging-Modells verweisen. PSI-Repositories erhalten eine besondere Relevanz, die deutlich wird, wenn das Szenario, Topics innerhalb Topic Maps, die das Topic Maps-Tagging-Modell implementieren, über ihre Subject Identifier auf öffentliche Repositories verweisen zu lassen, weitergedacht wird. Wenn bspw. die Subject Identifier von Tags repräsentierenden Topics nicht mehr auf systeminterne Ressourcen zeigen (<http://www.example.org/tag/XML>), sondern auf öffentliche PSI-Repositories, können über eine Abfrage mit der Verwendung von Published Subject Identifier mehrere Topic Maps-Systeme angefragt werden, die im Resultat jeweils semantisch identische Rückgaben liefern. Die Verwaltung und Pflege solcher Repositories, an die spezifische Qua-

litätsanforderungen zu stellen sind (siehe Kapitel 5.2), könnten als eine zukünftige Aufgaben von Bibliotheken angesehen werden.

PSI stellt das Konzept im Topic Maps-Paradigma dar, um Semantik etablieren zu können und auf deren Basis zu operieren (Queries, Merging). Dessen konsequente Anwendung im Topic Maps-Tagging-Modell sichert Implementierungen ein semantisches Fundament, in dem dargestellten Fall für die skizzierte Semantik des Tagging.

Das Zusammenwirken der strukturellen und semantischen Aspekte kann für die Möglichkeit der Integration und Aggregation von Daten über Systemgrenzen hinweg eine nachhaltige Wirkung in der Durchsetzung als Prinzip und damit die langfristige Sicherung der Existenz bedeuten, da Standards (Topic Maps mit XTM, TMAPI/PHPTMAPI, TMWSI in naher Zukunft) Kosten- und Zeiteinsparungen bewirken sowie als Netzeffektgüter zu behandeln sind (Buxmann 2001, S. 434) (Effizienzpotential) und PSIs in entsprechenden Repositories die Qualität der Information erhöhen (Effektivitätspotential). Im dargelegten Fall kann dies also für Weblogs, Foto- und Linkverwaltungen u. a. Anwendungen und Anwendungsformen gelten, die Tagging implementieren.

Für die Bewertung der eingangs dieser Arbeit aufgestellten These ist nach der Betrachtung des exemplarischen Praxisfalls festzustellen, dass sich mit Topic Maps Datenstrukturen, die einen Kontext/eine Domäne repräsentieren, abbilden lassen und dabei über die gezeigten Prinzipien die Semantik der den Kontext abbildenden Konzepte, die *subjects*, etabliert und somit erhalten werden kann. Für den Datenaustausch auf Basis von Standards kann diese Semantik „benutzt“ werden.

Eine nächste Evolutionsstufe des World Wide Web, das Semantic Web, ist in seinen Grundzügen von Berners-Lee u. a. gekennzeichnet. Eine wichtige Technologie in diesem Zusammenhang ist RDF (Berners-Lee/Hendler/Lassila 2001). Das semantische Potential von Topic Maps und die Möglichkeit einer Interoperabilität mit RDF unterstützen die Existenzberechtigung dieser Technologie, die dadurch wiederum einen Beitrag für die Verwirklichung dieser Idee leisten kann.

Literatur- und Quellenverzeichnis

Ahmed 2001

Ahmed, Kal: *Justification for developing a topic map programming model*. URL http://www.techquila.com/pmodel_1.html. 2001. – Zuletzt aufgerufen: 11.9.2006

Altheim/Biezunski/Hunting/Newcomb 2001

Altheim, Murray; Biezunski, Michel; Hunting, Sam; Newcomb, Steven R.: *XML Topic Map (XTM) DTD*. URL <http://topicmaps.org/xtm/1.0/xtm1.dtd>. 8.2.2001
Zuletzt aufgerufen: 7.12.2006

Anderson 2004

Anderson, Chris: *The long tail*. URL <http://www.wired.com/wired/archive/12.10/tail.html>. Oktober 2004
Zuletzt aufgerufen: 17.8.2006

Armstrong/Hagel 1997

Armstrong, Arthur; Hagel, John: *Net gain – Profit im Netz: Märkte erobern mit virtuellen Communities*. Wiesbaden : Gabler, 1997. – ISBN 3-409-18959-9

Armstrong/Hagel 2000

Armstrong, Arthur; Hagel, John: Der wahre Wert von Online-Gemeinschaften. In: Tapscott, Don (Hrsg.): *Erfolg im E-Business*. München [u.a.] : Hanser, 2000
ISBN 3-446-21350-3

Barta 2005

Barta, Robert: *TMIP, a RESTful Topic Maps interaction protocol*. URL <http://www.idealliance.org/papers/extreme/Proceedings/xslfo-pdf/2005/Barta01/EML2005Barta01.pdf>. August 2005
Zuletzt aufgerufen: 29.11.2006

Bell 1985

Bell, Daniel: *Die nachindustrielle Gesellschaft*. Frankfurt/Main : Campus, 1985
ISBN 3-593-33555-7

Bergmann 2005

Bergmann, Sebastian: *Professionelle Softwareentwicklung mit PHP 5: Objektorientierung. Entwurfsmuster. Modellierung. Fortgeschrittene Datenbankprogrammierung*. Heidelberg : dpunkt, 2005. – ISBN 3-89864-229-1

Berners-Lee/Hendler/Lassila 2001

Berners-Lee, Tim; Hendler, James; Lassila, Ora: *The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities*. URL http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21. 17.5.2001. – Zuletzt aufgerufen: 30.11.2006

Bernhard Medien GmbH 2006

Bernhard Medien GmbH, Olching (Hrsg.): *Beliebtester Anbieter der Woche*. URL <http://www2.webhostlist.de/host/t10/views.php>. 2006. – Zuletzt aufgerufen: 12.10.2006

Biezunski/Newcomb/Bryan 2002

Biezunski; Michel; Newcomb, Steven R.; Bryan, Martin: *Guide to the topic map standards*. URL <http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0323.htm>. 23.6.2002. – Zuletzt aufgerufen: 1.9.2006

Bourret 2001

Bourret, Ronald: *Mapping DTDs to databases*. URL <http://www.xml.com/pub/a/2001/05/09/dtdtodbs.html>. 9.5.2001. – Zuletzt aufgerufen: 18.10.2006

Bruhn 2000

Bruhn, Manfred: Qualitätssicherung im Dienstleistungsmarketing – eine Einführung in die theoretischen und praktischen Probleme. In Bruhn, Manfred (Hrsg.); Strauss, Bernd (Hrsg.): *Dienstleistungsqualität: Konzepte – Methoden – Erfahrungen*. 3., vollst. überarb. und erw. Aufl. Wiesbaden : Gabler, 2000. – ISBN 3-409-33655-9, S. 21-48

Picot/Fischer 2005

Bucher, Hans-Jürgen; Büffel, Steffen: Weblogs – Journalismus in der Weltgesellschaft. Grundstrukturen einer netzwerkorientierten Form der Medienkommunikation. In: Picot, Arnold (Hrsg.); Fischer, Tim (Hrsg.): *Weblogs professionell*. Heidelberg : dpunkt, 2005. – ISBN 3-89864-375-1, S. 131-156

Bullinger et al. 2002

Bullinger, Hans-Jörg; Baumann, Timo; Fröschle, Norbert; Mack, Oliver; Trunzer, Thomas; Waltert, Jochen: *Business Communities: Professionelles Beziehungsmanagement von Kunden, Mitarbeitern und B2B-Partnern im Internet*. Bonn : Galileo, 2002. – ISBN 3-89842-121-X

Buxmann 2001

Buxmann, Peter: Standards und Standardisierung. In: Mertens, Peter (Haupthrsg) ... : *Lexikon der Wirtschaftsinformatik*. 4., vollst. neu bearb. und erw. Aufl. Berlin [u.a.] : Springer, 2001. – ISBN 3-540-42339-7, S. 434-435

Colomb 2006

Colomb, Robert: *The topic map metamodel*. URL http://www.itee.uq.edu.au/~infs4206/Metamodeling/_Resources/15_Topic_Map_MetamodelR1.pdf. 2006
Zuletzt aufgerufen: 17.11.2006

Dichtl/Issing 1993

Dichtl, Erwin (Hrsg.); Issing, Otmar (Hrsg.): *Vahlens großes Wirtschaftslexikon*. 2. überarb. und erw. Aufl. München : Beck, 1993. – ISBN 3-8006-1698-X

Dreyer et al. 2005

Dreyer, Stephan; Held, Thorsten; Laudien, Arne; Schulz, Wolfgang: Medien- und Telekommunikationsrecht. In: Hans-Bredow-Institut (Hrsg.): *Forschungsbericht 2004/2005*. URL <http://www.hans-bredow-institut.de/publikationen/fb/FB05.pdf>. S. 10-11. Mai 2005. – Zuletzt aufgerufen: 17.8.2006

DuBois 2005

DuBois, Paul: *MySQL – the definitive guide to using, programming, and administering MySQL 4.1 and 5.0*. 3. ed. Indianapolis, Ind. : Sams Publishing, 2005
ISBN 0-672-32673-6

DUDEN 1990

Wiss. Rat d. Dudenred. (Bearb.): *Duden Fremdwörterbuch*. Mannheim [u.a.] : Dudenverl., 1990. – ISBN 3-411-20915-1

E-Soft Inc. 2006a

E-Soft Inc., Burlington Ont, Canada (Hrsg.): *Web server survey*. URL http://www.securityspace.com/s_survey/data/200608/index.html. 1.9.2006
Zuletzt aufgerufen: 13.9.2006

E-Soft Inc. 2006b

E-Soft Inc., Burlington Ont, Canada (Hrsg.): *Apache module report*. URL http://www.securityspace.com/s_survey/data/man.200608/apachemods.html. 1.9.2006
Zuletzt aufgerufen: 13.9.2006

Free Software Foundation, Inc. 2006

Free Software Foundation, Inc., Boston, Ma., USA (Hrsg.): *GNU general public license*. URL <http://www.gnu.org/licenses/gpl.html>. Aktualisierungsdatum: 8.10.2006
Zuletzt aufgerufen: 2.10.2006

Freese 2003

Freese, Eric: Topic Maps and RDF. In: Park, Jack (Ed.); Hunting, Sam (Technical Ed.): *XML Topic Maps : creating and using Topic Maps for the web*. Boston, Ma. [u.a.] : Addison-Wesley, 2003. – ISBN 0-201-74960-2, S. 31-50

Fuecks 2005

Fuecks, Harry: *PHP 5 für Fortgeschrittene: Wie man PHP-Code besser, schneller und sicherer macht*. Heidelberg : dpunkt, 2005. – ISBN 3-89864-300-X

Garshol 2005

Garshol, Lars M.: *TMRAP – Topic Maps remote access protocol*. URL <http://www.garshol.priv.no/download/text/tmrap.pdf>. 2005
Zuletzt aufgerufen: 29.11.2006

Garshol 2006

Garshol, Lars M.: *Tags/folksonomies and Topic Maps*. URL <http://www.garshol.priv.no/blog/33.html>. 6.4.2006. – Zuletzt aufgerufen: 18.8.2006

Gaus 2003

Gaus, Wilhelm: *Dokumentations- und Ordnungslehre – Theorie und Praxis des Information Retrieval*. 4., überarb. und erw. Aufl. Berlin [u.a.] : Springer, 2003
ISBN 3-540-43505-0

Geroimenko 2004

Geroimenko, Vladimir: *Dictionary of XML technologies and the semantic web*. London [u.a.] : Springer, 2004. – ISBN 1-85233-768-0

Gerpott 2006

Gerpott, Thorsten J.: Wettbewerbsstrategien – Überblick, Systematik und Perspektiven. In: Scholz, Christian (Hrsg.): *Handbuch Medienmanagement*. Berlin [u.a.] : Springer, 2006. – ISBN 3-540-23540-X, S. 307-356

Google Inc. 2004

Google Inc., Mountain View, Ca., USA (Hrsg.): *Our search: Google technology*. URL <http://www.google.com/technology/>. 2004. – Zuletzt aufgerufen: 8.11.2006

Google Inc. 2006

Google Inc., Mountain View, Ca., USA (Hrsg.): *Investor relations – financial data*. URL http://investor.google.com/fin_data.html. 2006. – Zuletzt aufgerufen: 15.8.2006

Grassmuck 2004

Grassmuck, Volker: *Freie Software: zwischen Privat- und Gemeineigentum*. 2., korr. Aufl. Bonn : Bundeszentrale für politische Bildung, 2004. – ISBN 3-89331-569-1

Gruber 1993

Gruber, Tom: *A translation approach to portable ontology specifications*. URL ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-92-71.ps.gz. April 1993
Zuletzt aufgerufen: 7.12.2006

Gulbrandsen 2005

Gulbrandsen, Are D.: *Conceptual modeling of Topic Maps with ORM versus UML*.

URL <http://www.informatik.uni-leipzig.de/~tmra05/PRES/AG.pdf>. 2005

Zuletzt aufgerufen: 9.11.2006

Hammond et al. 2005

Hammond, Tony; Hannay, Timo; Lund, Ben; Scott, Joanna: *Social bookmarking tools (I) : a general review*. URL <http://www.dlib.org/dlib/april05/hammond/04hammond.html>.

April 2005. – Zuletzt aufgerufen: 28.11.2006

Hövel 2006

Hövel, Jörg Auf dem: *Lass das doch die Community machen*. URL

<http://www.heise.de/tp/r4/artikel/22/22832/1.html>. 7.6.2006

Zuletzt aufgerufen: 16.8.2006

Hofmann 2001

Hofmann, Ulrich: *Netzwerk-Ökonomie*. Heidelberg : Physica, 2001

ISBN 3-7908-1394-X

Hornig 2006

Hornig, Frank: Du bist das Netz. In: *Der Spiegel* (2006-07-17), Nr. 29, S. 60-74

Hunting 2003

Hunting, Sam: The rise and rise of Topic Maps. In: Park, Jack (Ed.); Hunting, Sam (Technical Ed.): *XML Topic Maps : creating and using Topic Maps for the web*.

Boston, Ma. [u.a.] : Addison-Wesley, 2003. – ISBN 0-201-74960-2, S. 51-66

IAB 2006

Institut für Arbeitsmarkt- und Berufsforschung der Bundesagentur für Arbeit (Hrsg.):

Daten zur kurzfristigen Entwicklung von Wirtschaft und Arbeitsmarkt - Juli 2006. URL

http://doku.iab.de/werkber/2006/wb_arbeitszeit.pdf. Juli 2006

Zuletzt aufgerufen: 23.5.2006

ISO/IEC 13250-2

International Organization for Standardization/International Electrotechnical

Commission JTC1/SC34 (Hrsg.): *Topic Maps – Data Model*. URL

<http://www.isotopicmaps.org/sam/sam-model/>. 7.12.2006

Zuletzt aufgerufen: 28.8.2006

ISO/IEC 13250-3

International Organization for Standardization/International Electrotechnical

Commission JTC1/SC34 (Hrsg.): *Topic Maps – XML Syntax*. URL

<http://www.isotopicmaps.org/sam/sam-xtm/>. 7.12.2006. – Zuletzt aufgerufen: 6.9.2006

ISO/IEC N0344

International Organization for Standardization/International Electrotechnical Commission JTC1/SC34 (Hrsg.): *Reference Model for ISO 13250 Topic Maps (RM4TM)*. URL <http://www.isotopicmaps.org/rm4tm/RM4TM-1.0.html>. 12.11.2002
Zuletzt aufgerufen: 29.8.2006

IETF RFC 3986

Internet Engineering Task Force, Network Working Group (Hrsg.): *Uniform Resource Identifier (URI) : generic syntax*. URL <http://www.ietf.org/rfc/rfc3986.txt>. Januar 2005
Zuletzt aufgerufen: 28.8.2006

Kastens/Kleine Büning 2005

Kastens, Uwe; Kleine Büning, Hans: *Modellierung : Grundlagen und formale Methoden*. München [u.a.] : Hanser, 2005. – ISBN 3-446-40460-0

Kelly 2005

Kelly, Kevin: *We are the web*. URL <http://www.wired.com/wired/archive/13.08/tech.html>. August 2005
Zuletzt aufgerufen: 15.8.2006

Keuper 2002

Keuper, Frank: *Ökonomische Bedeutung der Information in der Informationsgesellschaft* In : Keuper, Frank (Hrsg.): *Electronic Business und Mobile Business*. Wiesbaden : Gabler, 2002. – ISBN 3-409-11793-8, S. 119-141
Keuper, Frank: *Convergence-based View – ein strategie-strukturierungstheoretischer Ansatz zum Management der Konvergenz digitaler Erlebniswelten*. In : Keuper, Frank (Hrsg.): *Electronic Business und Mobile Business*. Wiesbaden : Gabler, 2002. – ISBN 3-409-11793-8, S. 603-654

Krcmar 2005

Krcmar, Helmut: *Informationsmanagement*. Berlin [u.a.] : Springer, 2005. – ISBN 3-540-23015-7

Kuhlen 2005

Kuhlen, Francis: *E-World : Technologien für die Welt von morgen*. Berlin [u.a.] : Springer, 2005. – ISBN 3-540-21471-2

Lehmann 2005

Lehmann, Kai: *Der lange Weg zur Wissensgesellschaft*. In Lehman, Kai (Hrsg.); Schetsche, Michael (Hrsg.) : *Die Google-Gesellschaft – vom digitalen Wandel des Wissens*. Bielefeld : Transcript, 2005. – ISBN 3-89942-305-4, S. 33-40, S. 33-40

Leimeister/Krcmar 2002

Leimeister, Jan M.; Krcmar, Helmut: *Das Geschäftsmodell „Virtual Community – Revisited“*. URL

[http://www.winfobase.de/lehrstuhl/publikat.nsf/intern01/11F3B5A701049360C1256E430045AE9E/\\$FILE/04-02.pdf](http://www.winfobase.de/lehrstuhl/publikat.nsf/intern01/11F3B5A701049360C1256E430045AE9E/$FILE/04-02.pdf). 2002. – Zuletzt aufgerufen: 2.8.2006

Maicher 2006a

Maicher, Lutz: *Deutsche Topic Maps-Terminologie*. URL <http://www.informatik.uni-leipzig.de/~maicher/tmt/TMT.html>. 27.7.2006. – Zuletzt aufgerufen: 28.8.2006

Maicher 2006b

Maicher, Lutz: *Deutsche Topic Maps-Terminologie*. URL <http://www.semantic-web.at/10.36.134.article.rueckblick-auf-die-tmra-2006-topic-maps-research-and-applications-von-lutz-maicher.htm>. 2006. – Zuletzt aufgerufen: 28.8.2006

Maslow 2002

Maslow, Abraham H.: *Motivation und Persönlichkeit*. 9. Aufl. Reinbek : Rowohlt, 2002
ISBN 3-499-17395-6

Mathes 2004

Mathes, Adam: *Folksonomies - cooperative classification and communication through shared metadata*. URL <http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html>. Dezember 2004. – Zuletzt aufgerufen: 18.8.2006

McLuhan 1995

McLuhan, Marshall: *Die Gutenberg-Galaxis : das Ende des Buchzeitalters*. Bonn [u.a.] : Addison-Wesley, 1995. – ISBN 3-89319-999-3

Meffert 2005

Meffert, Heribert: *Marketing – Grundlagen marktorientierter Unternehmensführung. Konzepte – Instrumente – Praxisbeispiele*. 9., überarb. und erw. Aufl. Wiesbaden : Gabler, 2005. – ISBN 3-409-69017-4

Mertins/Heisig/Vorbeck 2003

Mertins, Kay; Heisig, Peter; Vorbeck, Jens: Introduction. In: Mertins, Kay (Ed.); Heisig, Peter (Ed.); Vorbeck, Jens (Ed.): *Knowledge management – concepts and best practices*. 2. ed. Berlin [u.a.] : Springer, 2003. – ISBN 3-540-00490-4, S. 1-11

MySQL AB 2006a

MySQL AB, Uppsala, Schweden (Hrsg.): *Überblick über InnoDB-Tabellen*. URL <http://dev.mysql.com/doc/refman/4.1/en/innodb-overview.html>. 2006
Zuletzt aufgerufen: 13.10.2006

MySQL AB 2006b

MySQL AB, Uppsala, Schweden (Hrsg.): *Foreign key constraints*. URL

<http://dev.mysql.com/doc/refman/5.0/en/innodb-foreign-key-constraints.html>. 2006
Zuletzt aufgerufen: 20.10.2006

MySQL AB 2006c

MySQL AB, Uppsala, Schweden (Hrsg.): *Create table syntax*. URL
<http://dev.mysql.com/doc/refman/5.0/en/create-table.html>. 2006
Zuletzt aufgerufen: 20.10.2006

Netcraft Ltd. 2006

Netcraft Ltd., Bath, England (Hrsg.): *Web server survey*. URL
http://news.netcraft.com/archives/web_server_survey.html. 2006
Zuletzt aufgerufen: 13.9.2006

Newcomb 2003

Newcomb, Steven R.: A perspektive on the quest for global knowledge interchange.
In: Park, Jack (Ed.); Hunting, Sam (Technical Ed.): *XML Topic Maps : Creating and Using Topic Maps for the Web*. Boston, Ma. [u.a.] : Addison-Wesley, 2003. – ISBN 0-201-74960-2, S. 31-50

Newcomb 2006

Newcomb, Steven R.: *Flat topic mapping for a flat world*. URL
<http://www.informatik.uni-leipzig.de/~tmra/2006/contributions/tmra2006-keynote.zip>.
2006. – Zuletzt aufgerufen: 27.10.2006

Obrst/Liu 2003

Obrst, Leo; Liu, Howard: Knowledge representation, ontological engineering, and Topic Maps. In: Park, Jack (Ed.); Hunting, Sam (Technical Ed.): *XML Topic Maps : creating and using Topic Maps for the web*. Boston [u.a.] : Addison-Wesley, 2003 ISBN 0-201-74960-2, S. 103-147

Oestereich 2005

Oestereich, Bernd: *Die UML 2.0 Kurzreferenz für die Praxis : kurz, bündig, ballastfrei*. 4. Aufl. München [u.a.] : Oldenbourg, 2005. – ISBN 3-486-57788-3

O'Reilly 2005

O'Reilly, Tim: *What Is Web 2.0 – design patterns and business models for the next generation of software*. URL
<http://www.oreilynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>.
30.9.2005. – Zuletzt aufgerufen: 14.8.2006

Park 2003

Park, Jack (Ed.); Hunting, Sam (Technical Ed.): *XML Topic Maps : creating and using Topic Maps for the web*. Boston, Ma. [u.a.] : Addison-Wesley, 2003. – ISBN 0-201-74960-2

Pepper 2000

Pepper, Steve: *The TAO of Topic Maps – finding the way in the age of infoglut*. URL <http://www.ontopia.net/topicmaps/materials/tao.html>. 2000
Zuletzt aufgerufen: 6.12.2006

PHP Group 2006a

PHP Group (Hrsg.): *PHP manual. Opendir*. URL <http://de2.php.net/manual/en/function.opendir.php>. Aktualisierungsdatum: 19.11.2006
Zuletzt aufgerufen: 23.9.2006

PHP Group 2006b

PHP Group (Hrsg.): *PHP manual. Include()*. URL <http://de2.php.net/manual/de/function.include.php>. Aktualisierungsdatum: 3.11.2006
Zuletzt aufgerufen: 27.10.2006

Picot/Reichwald/Wigand 2003

Picot, Arnold; Reichwald, Ralf; Wigand, Rolf T.: *Die grenzenlose Unternehmung : Information, Organisation und Management*. 5., aktualisierte Aufl. Wiesbaden : Gabler, 2003. – ISBN 3-409-52214-X

Picot/Fischer 2006

Picot, Arnold; Fischer, Tim: Einführung – Veränderte mediale Realitäten und der Einsatz von Weblogs im unternehmerischen Umfeld. In: Picot, Arnold (Hrsg.); Fischer, Tim (Hrsg.): *Weblogs professionell*. Heidelberg : dpunkt, 2006. – ISBN 3-89864-375-1, S. 3-12

Pilgrim 2002

Pilgrim, Mark: *What is RSS*. URL <http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>. 18.12.2002. – Zuletzt aufgerufen: 22.8.2006

Porter 1999

Porter, Michael. E.: *Wettbewerbsvorteile : Spitzenleistungen erreichen und behaupten*. 5., durchges. und erw. Aufl. Frankfurt/Main [u.a.]: Campus, 1999
ISBN 3-593-36178-7

Probst/Raub/Romhardt 1998

Probst, Gilbert J.B.; Raub, Steffen P.; Romhardt, Kai: *Wissen managen – Wie Unternehmen ihre wertvollste Ressource optimal nutzen*. Frankfurt a. M.: Frankfurter Allg., Zeitung für Deutschland; Wiesbaden: Gabler, 1998 – ISBN 3-409-29317-5

Przepiorka 2006

Przepiorka, Sven: Weblogs, Wikis und die dritte Dimension. In: Picot, Arnold (Hrsg.); Fischer, Tim (Hrsg.): *Weblogs professionell*. Heidelberg : dpunkt, 2006
ISBN 3-89864-375-1, S. 13-30

Pushmann 2000

Pushman, Jalal: *Switching to PHP: What's the language difference?* URL
<http://www.zend.com/zend/art/langdiff.php>. 2.7.2000. – Zuletzt aufgerufen: 24.9.2006

Rath 2000

Rath, H. Holger: *Making Topic Maps more colourful*. URL
<http://www.gca.org/papers/xml europe2000/pdf/s29-01.pdf>. 2000
Zuletzt aufgerufen: 25.9.2006

Rath 2003

Rath, H. Holger: *The Topic Maps handbook*. URL
http://empolis.de/downloads/empolis_TopicMaps_Whitepaper20030206.pdf. 6.2.2003
Zuletzt aufgerufen: 29.8.2006

Raub/Probst/Gilbert 2000

Raub, Steffen P.; Probst, Gilbert J. B.: Knowledge Management und Electronic Business. In: Weiber, Rolf (Hrsg.): *Handbuch Electronic Business: Informationstechnologien - Electronic Commerce – Geschäftsprozesse*. Wiesbaden : Gabler, 2000. – ISBN 3-409-21636-7, S. 403-427

RDFTM 2006

RDF/Topic Maps Interoperability Task Force (Hrsg.): *Objectives*. URL
<http://www.w3.org/2001/sw/BestPractices/RDFTM>. Aktualisierungsdatum 12.7.2006
Zuletzt aufgerufen: 15.11.2006

Rentmeister/Klein 2003

Rentmeister, Jahn; Klein, Stefan: Geschäftsmodelle – ein Modebegriff auf der Waagschale. In *ZfB-Ergänzungsheft 1* (2003), S. 17-30

RSS Advisory Board 2006

RSS Advisory Board (Hrsg.): *Specification history*. URL <http://www.rssboard.org/rss-history>. – Zuletzt aufgerufen: 22.8.2006

RSS-DEV Working Group 2001a

RSS-DEV Working Group (Hrsg.): *RDF Site Summary (RSS) version 1.0*. URL
<http://web.resource.org/rss/1.0/spec>. 2001. – Zuletzt aufgerufen: 22.8.2006

RSS-DEV Working Group 2001b

RSS-DEV Working Group (Hrsg.): *RSS 1.0 modules: taxonomy*. URL
<http://web.resource.org/rss/1.0/modules/taxonomy/>. 2001
Zuletzt aufgerufen: 22.8.2006

Sauer 2002

Sauer, Hermann: *Relationale Datenbanken – Theorie und Praxis*. 5., aktualisierte und erw. Aufl. München [u.a.] : Addison-Wesley, 2002. – ISBN 3-8273-2060-7

Schreier 2001

Schreier, Ulf: Entity-Relationship-Darstellung. In: Mertens, Peter (Hauptthrg) ... : *Lexikon der Wirtschaftsinformatik*. 4., vollst. neu bearb. und erw. Aufl. Berlin [u.a.] : Springer, 2001. – ISBN 3-540-42339-7, S. 184-185

Schumann/Hess 2002

Schumann, Matthias; Hess, Thomas: *Grundfragen der Medienwirtschaft: eine betriebswirtschaftliche Einführung*. 2., verbesserte und erw. Aufl. Berlin [u.a.] : Springer, 2002. – ISBN 3-540-43387-2

Seufert/Moisseeva/Steinbeck 2002

Seufert, Sabine; Moisseeva, Marina; Steinbeck, Reinhold: *Virtuelle Communities gestalten*. URL <http://www.scil.ch/seufert/docs/virtuelle-communities.pdf>. 2002
Zuletzt aufgerufen: 2.8.2006

Shirky 2003

Shirky, Clay: *A group is its own worst enemy*. URL
http://www.shirky.com/writings/group_enemy.html. April 2003
Zuletzt aufgerufen: 3.8.2006

Sigel 2004

Sigel, Alexander: *Organisation verteilten Wissens mit semantischen Wissensnetzen und der Aggregation semantischer Wissensdienste am Beispiel Digitale Bibliotheken/Kulturelles Erbe*. URL http://www.bonn.iz-soz.de/wiss-org/wissorg04/isko2004papers/Sigel%20-%20ISKO2004_SIGEL_Organisation_verteilten_Wissens_v15.pdf. 2004
Zuletzt aufgerufen: 9.11.2006

Sigel/Büntten/Frank 2006

Sigel, Alexander; Büntten, Lars; Frank, Marcus: *Topic Maps-based semblogging with semblog-tm*. URL <http://www.informatik.uni-leipzig.de/~tmra/2006/contributions/tmra2006-semblogging.pdf>. 2006
Zuletzt aufgerufen: 9.11.2006

Sinha 2005

Sinha, Rashmi: *A cognitive analysis of tagging : or how the lower cognitive cost of tagging makes it popular*. 27.5.2005. URL
http://www.rashmisinha.com/archives/05_09/tagging-cognitive.html
Zuletzt aufgerufen: 1.9.2006

Sinz 2001

Sinz, Elmar J.: Modell. In: Mertens, Peter (Hauptthrg) ... : *Lexikon der Wirtschaftsinformatik*. 4., vollst. neu bearb. und erw. Aufl. Berlin [u.a.] : Springer, 2001
ISBN 3-540-42339-7, S. 311-312

Sjurts 2004

Sjurts, Insa (Hrsg.): *Gabler Lexikon Medienwirtschaft*. Wiesbaden : Gabler, 2004
ISBN 3-409-12451-9

Stähler 2001

Stähler, Patrick: *Geschäftsmodelle in der digitalen Ökonomie : Merkmale, Strategien und Auswirkungen*. Lohmar [u.a.] : Eul, 2001. – ISBN 3-89012-888-2

Statistisches Bundesamt 2006

Statistisches Bundesamt (Hrsg.): *Durchschnittliche weitere Lebenserwartung*. URL <http://www.destatis.de/basis/d/bevoe/bevoetab3.php>. Aktualisierungsdatum: 19.10.2006. – Zuletzt aufgerufen: 31.7.2006

Studer/Oppermann/Schnurr 2001

Studer, Rudi; Oppermann, Henrik; Schnurr, Hans-Peter: *Die Bedeutung von Ontologien für das Wissensmanagement*. URL http://www.ontoprise.de/documents/Bedeutung_von_Ontologien_fuer_WM.pdf. September 2001. – Zuletzt aufgerufen: 28.8.2006

Timmers 1998

Timmers, Paul: Business Models for Electronic Markets. In *EM – Electronic Markets* Vol. 8 (1998), No. 2, S. 3-8

Timmers 2003

Timmers, Paul: Lessons from B2B E-Business Models. In *ZfB-Ergänzungsheft 1* (2003), S. 121-140

TMAPI 2005a

TMAPI.org (Hrsg.): *TMAPI: common Topic Map application programming interface*. URL <http://www.tmap.org>. Aktualisierungsdatum: 26.2.2005
Zuletzt aufgerufen: 23.9.2006

TMAPI 2005b

TMAPI.org (Hrsg.): *Javadoc*. URL <http://www.tmap.org/apiDocs/index.html>.
Zuletzt aufgerufen: 23.9.2006

TMAPI 2005c

TMAPI.org (Hrsg.): *A guide to TMAPI*. URL <http://www.tmap.org/guide/index.html>.
Zuletzt aufgerufen: 23.9.2006

TMAPI 2005d

TMAPI.org (Hrsg.): *Features*. URL <http://www.tmap.org/features/>.
Zuletzt aufgerufen: 3.10.2006

TopicMaps.org 2001

TopicMaps.org (Hrsg.): *XML Topic Maps (XTM) 1.0 - TopicMaps.Org specification*.

URL <http://www.topicmaps.org/xtm/index.html>. 6.8.2001

Zuletzt aufgerufen: 11.9.2006

Vatant 2003

Vatant, Bernard: Topic Maps from representation to identity : conversation, names, and published subject indicators. In: Park, Jack (Ed.); Hunting, Sam (Technical Ed.): *XML Topic Maps : creating and using Topic Maps for the web*. Boston [u.a.] : Addison-Wesley, 2003. – ISBN 0-201-74960-2, S. 67-79

Wiborny 1991

Wiborny, Werner: *Datenmodellierung, CASE Management*. Bonn [u.a.] : Addison-Wesley, 1991. – ISBN 3-89319-278-6

Wilde 2001

Wilde, Klaus D.: Data Warehouse, OLAP und Data Mining im Marketing – Moderne Informationstechnologien im Zusammenspiel. In: Hippner, Hajo (Hrsg); Küsters, Ulrich L. (Hrsg.); Meyer, Matthias (Hrsg.): *Handbuch Data Mining im Marketing: Knowledge Discovery in Marketing Databases*. Braunschweig [u.a.] : Vieweg, 2001
ISBN 3-528-05713-0, S. 1-21

Wirtz 2001

Wirtz, Bernd W.: *Electronic Business. 2.*, vollständig überarb. und erw. Aufl.
Wiesbaden : Gabler, 2001. – ISBN 3-409-21660-X

Wirtz 2005

Wirtz, Bernd W.: *Medien- und Internetmanagement. 4.*, überarb. Aufl. Wiesbaden : Gabler, 2005 - ISBN 3-409-41661-7

Zendel 2005

Lüttcher, Beatrice: Eine kurze Geschichte freier Software. Interview mit Oliver Zendel. In Lehman, Kai (Hrsg.); Schetsche, Michael (Hrsg.) : *Die Google-Gesellschaft – vom digitalen Wandel des Wissens*. Bielefeld : Transcript, 2005. – ISBN 3-89942-305-4, S. 151-156

Zerdick et al. 1999

Zerdick, Axel; Picot, Arnold; Schrape, Klaus; Artopé, Alexander; Goldhammer, Klaus; Lange, Ulrich T.; Vierkant, Eckart; López-Escobar, Esteban; Silverstone, Roger: *Die Internet-Ökonomie – Strategien für die digitale Wirtschaft. 2.* korrigierte Aufl. Berlin [u.a.] : Springer, 1999. – ISBN 3-540-64915-8

Anhang A – E-Mail von Lars Heuer über die TMAPI-E-Mailliste zur Klärung der Frage der Notwendigkeit des Erstellens eines Topic C beim Merging von zwei Topics A und B

Betreff: Re: [Tmapi-discuss] Merging Topic Items
Von: Lars Heuer <heuer@semagia.com>
Datum: Fri, 8 Sep 2006 18:30:23 +0200
An: Johannes Schmidt <tmap-discuss@lists.sourceforge.net>

Hi Johannes,

> > 1. Create a new topic item C.
> > ...

> > I cannot see why to create an item C first. If A and B are equal

It's perfectly okay if you take in practice topic A and attach all characteristics from B to it and replace all occurrences of topic B with A and delete topic B.

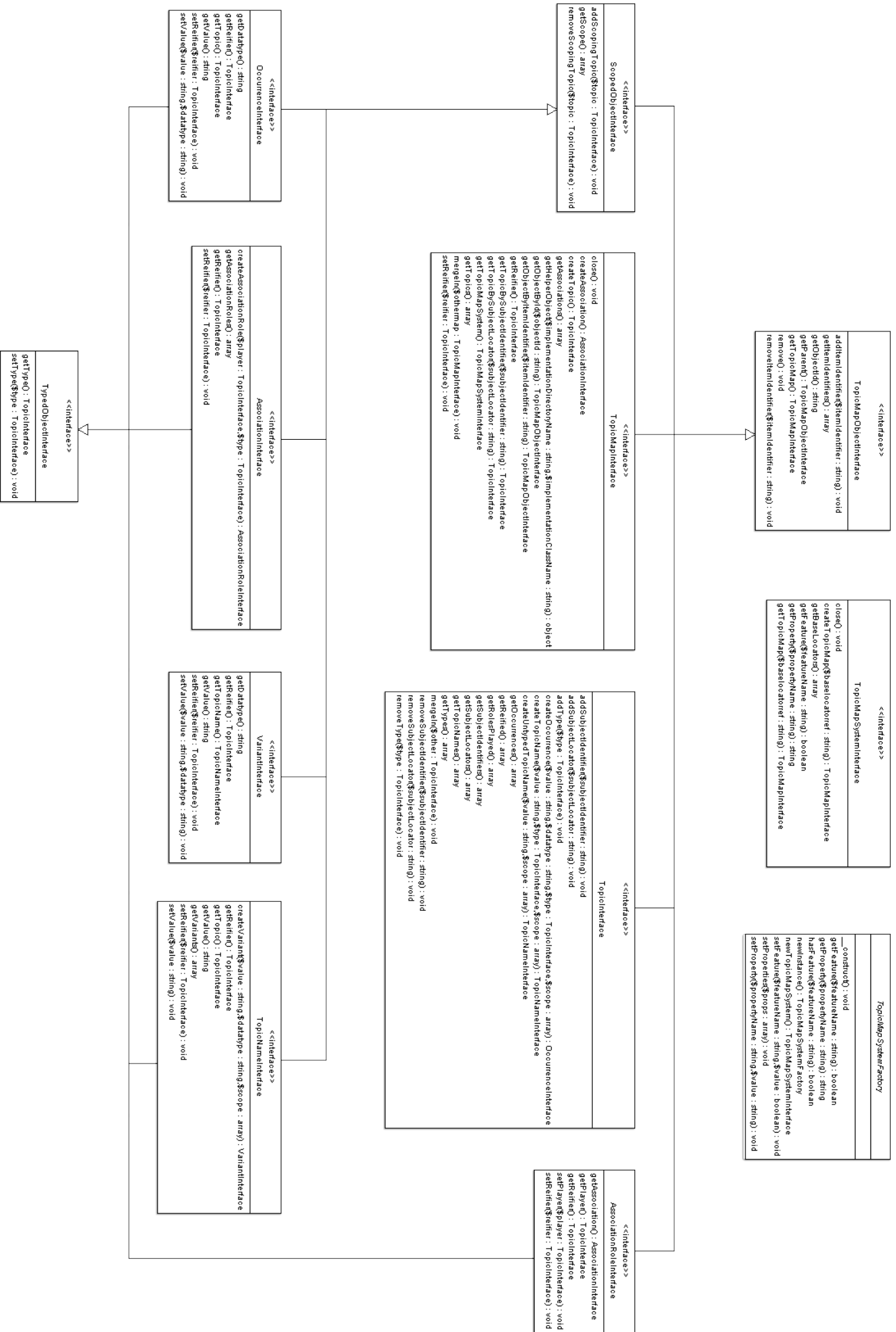
Note, that TMAPI requires, that the reference to topic B stays valid:

```
""""
[...] This means that the client must be able to continue
using a
reference to either of the two original Topics but should be
viewing/manipulating the same merged Topic.
[...]
""""
http://tmap.org/features/automerge/
```

TMDM uses topic C for some reason I don't know. It was discussed to reuse topic A but the approach was rejected.

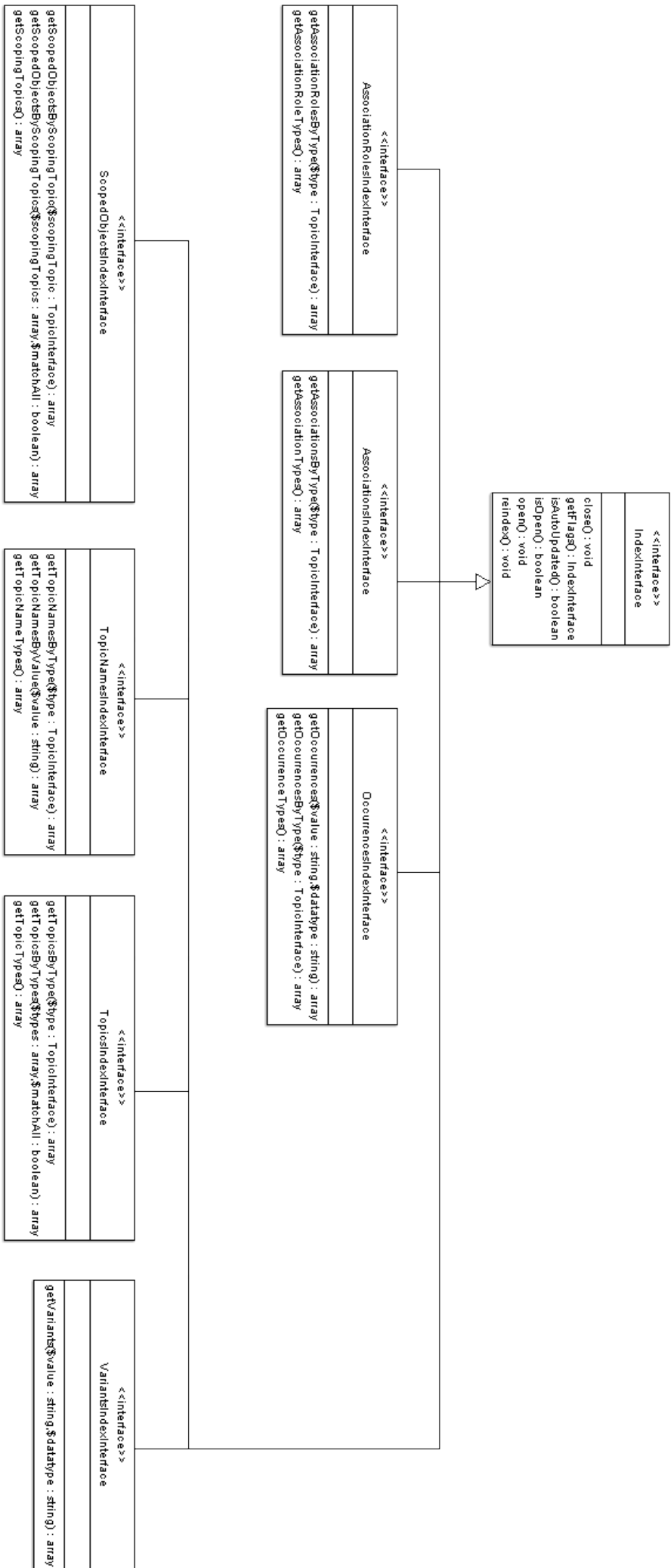
Best regards,
Lars
--
<http://www.semagia.com>

Anhang B – UML Klassendiagramm für das PHPTMAPI Kernpaket (core)



Anhang C – UML Klassendiagramm der Exceptions in PHPTMAPI

**Anhang D – UML Klassendiagramm für das PHPTMAPI
Indexpaket (index, index.core)**



Anhang E – Poster zur Präsentation von PHPTMAPI auf der „Conferences on Topic Maps Research and Applications“, TMRA, am 11.10.2006 in Leipzig

PHPTMAPI

<http://phptmapi.sourceforge.net>

Authors: Johannes Schmidt, Andreas Stephan

for more info have a look at <http://www.t8d.de>
or contact us via kontakt@t8d.de

Facts

Implementation of the TMAPI specification (<http://www.tmapi.org>) for PHP 5

Is closer to the TMDM, e.g.

- defines *setReifier()* method for reifiable Topic Maps constructs
- use of XML schema datatype
- source locators are called item identifiers

Open Source (public domain)

Simplifying aspects:

- use of strings instead of a locator interface to represent locators
- *getTopicBySubjectIdentifier()*, *getTopicBySubjectLocator()*, *getObjectByItemIdentifier()* are directly available in *TopicMapInterface*

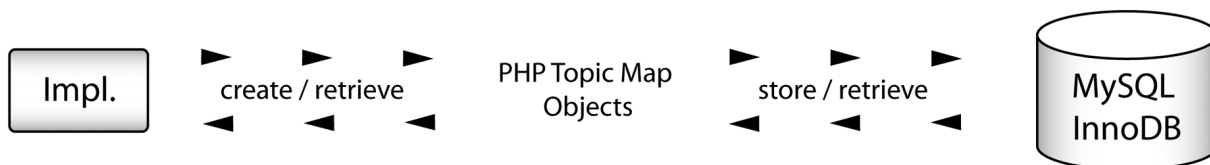
„Topic Maps for the masses“

Allows developers to write Topic Maps applications in PHP using a standardized API

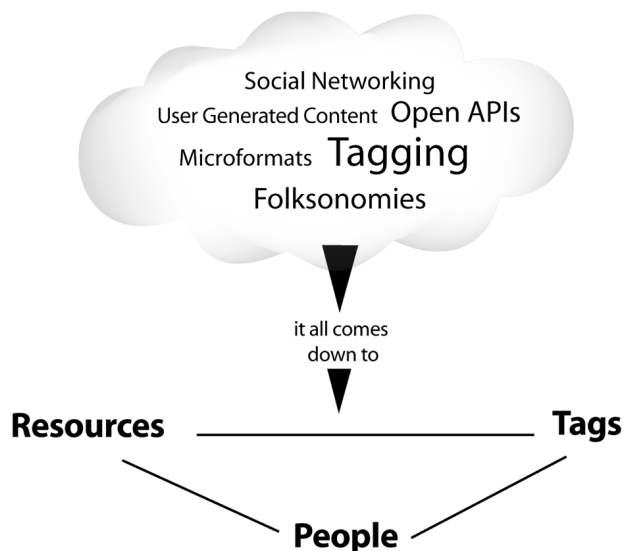
Easy integration into popular

- Blogging Systems
- Forum Systems
- CMS

An implementation of the core interfaces: QuaaXTM (<http://quaaXTM.sourceforge.net>)



Topic Maps and „Web 2.0“



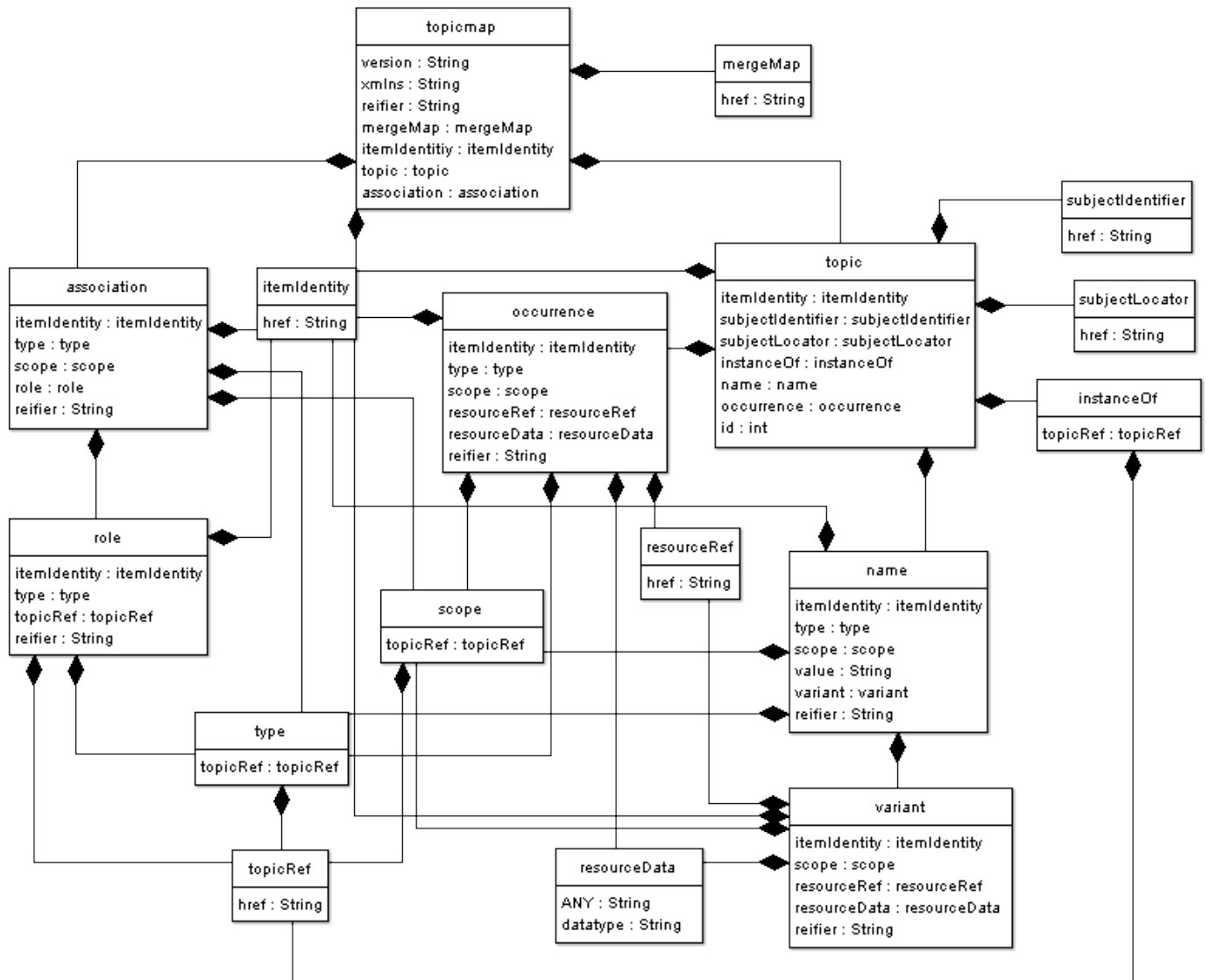
The Problem

Most of the new social networking services (respectively their APIs) offer proprietary I/O interfaces and use output formats that structure the essential information inconsistently. This increases the effort for integrating external information and makes it hard to establish reliable semantics between content of different service providers.

The Vision

Use a PHPTMAPI application to implement a TM-based conceptual tagging-model and store information about resources, users and tags in a Topic Map. A standardized meta layer like this would allow new services to establish efficient and semantically reliable information interchange processes.

Anhang F – UML Klassendiagramm (informativ) als Resultat der durch die Abbildung der DTD von XTM 2.0 auf ein Objektschema resultierenden Klassen nach der Methode des *object-relational mappings*



Anhang H – CREATE TABLE-Anweisungen zur Definition der Tabellen für die MySQL-Datenbank der PHPTMAPI-Implementierung mit InnoDB als *storage engine*

```
CREATE TABLE qtm_topicmap (  
  id INT NOT NULL AUTO_INCREMENT,  
  baselocator VARCHAR(255) NOT NULL,  
  PRIMARY KEY(id)  
) ENGINE=InnoDB;  
  
CREATE TABLE qtm_topic (  
  id INT NOT NULL AUTO_INCREMENT,  
  topicmap_id INT NOT NULL,  
  PRIMARY KEY(id),  
  INDEX t_ref_ind (topicmap_id),  
  FOREIGN KEY (topicmap_id) REFERENCES qtm_topicmap(id) ON DELETE  
CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;  
  
CREATE TABLE qtm_subjectidentifier (  
  id INT NOT NULL AUTO_INCREMENT,  
  topic_id INT NOT NULL,  
  locator VARCHAR(255) NULL,  
  PRIMARY KEY(id),  
  INDEX si_ref_ind (topic_id),  
  FOREIGN KEY (topic_id) REFERENCES qtm_topic(id) ON DELETE CASCADE ON  
UPDATE CASCADE  
) ENGINE=InnoDB;  
  
CREATE TABLE qtm_subjectlocator (  
  id INT NOT NULL AUTO_INCREMENT,  
  topic_id INT NOT NULL,  
  locator VARCHAR(255) NULL,  
  PRIMARY KEY(id),  
  INDEX sl_ref_ind (topic_id),  
  FOREIGN KEY (topic_id) REFERENCES qtm_topic(id) ON DELETE CASCADE ON  
UPDATE CASCADE  
) ENGINE=InnoDB;  
  
CREATE TABLE qtm_instanceof (  
  id INT NOT NULL AUTO_INCREMENT,  
  topic_id INT NOT NULL,  
  type_id INT NOT NULL,  
  PRIMARY KEY(id),  
  INDEX io_ref_ind (topic_id, type_id),  
  FOREIGN KEY (topic_id) REFERENCES qtm_topic(id) ON DELETE CASCADE ON  
UPDATE CASCADE,  
  FOREIGN KEY (type_id) REFERENCES qtm_topic(id) ON DELETE RESTRICT ON  
UPDATE CASCADE  
) ENGINE=InnoDB;  
  
CREATE TABLE qtm_topicname (  
  id INT NOT NULL AUTO_INCREMENT,  
  topic_id INT NOT NULL,  
  type_id INT NULL,  
  value VARCHAR(255) NULL,  
  PRIMARY KEY(id),  
  INDEX tn_ref_ind (topic_id, type_id),  
  FOREIGN KEY (topic_id) REFERENCES qtm_topic(id) ON DELETE CASCADE ON  
UPDATE CASCADE,  
  FOREIGN KEY (type_id) REFERENCES qtm_topic(id) ON DELETE RESTRICT ON  
UPDATE CASCADE  
) ENGINE=InnoDB;
```

```

CREATE TABLE qtm_variant (
  id INT NOT NULL AUTO_INCREMENT,
  topicname_id INT NOT NULL,
  value VARCHAR(255) NULL,
  datatype VARCHAR(255) NULL,
  PRIMARY KEY(id),
  INDEX var_ref_ind (topicname_id),
  FOREIGN KEY (topicname_id) REFERENCES qtm_topicname(id) ON DELETE
CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;

CREATE TABLE qtm_occurrence (
  id INT NOT NULL AUTO_INCREMENT,
  topic_id INT NOT NULL,
  type_id INT NULL,
  value TEXT NULL,
  datatype VARCHAR(255) NULL,
  PRIMARY KEY(id),
  INDEX occ_ref_ind (topic_id, type_id),
  FOREIGN KEY (topic_id) REFERENCES qtm_topic(id) ON DELETE CASCADE ON
UPDATE CASCADE,
  FOREIGN KEY (type_id) REFERENCES qtm_topic(id) ON DELETE RESTRICT ON
UPDATE CASCADE
) ENGINE=InnoDB;

CREATE TABLE qtm_association (
  id INT NOT NULL AUTO_INCREMENT,
  type_id INT NULL,
  topicmap_id INT NOT NULL,
  PRIMARY KEY(id),
  INDEX assoc_ref_ind (topicmap_id, type_id),
  FOREIGN KEY (topicmap_id) REFERENCES qtm_topicmap(id) ON DELETE
CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (type_id) REFERENCES qtm_topic(id) ON DELETE RESTRICT ON
UPDATE CASCADE
) ENGINE=InnoDB;

CREATE TABLE qtm_assocrole (
  id INT NOT NULL AUTO_INCREMENT,
  association_id INT NOT NULL,
  type_id INT NULL,
  reference INT NULL,
  PRIMARY KEY(id),
  INDEX role_ref_ind (association_id, reference, type_id),
  FOREIGN KEY (association_id) REFERENCES qtm_association(id) ON
DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (reference) REFERENCES qtm_topic(id) ON DELETE RESTRICT
ON UPDATE CASCADE,
  FOREIGN KEY (type_id) REFERENCES qtm_topic(id) ON DELETE RESTRICT ON
UPDATE CASCADE
) ENGINE=InnoDB;

CREATE TABLE qtm_scope (
  id INT NOT NULL AUTO_INCREMENT,
  association_id INT NULL,
  topicname_id INT NULL,
  occurrence_id INT NULL,
  variant_id INT NULL,
  PRIMARY KEY(id),
  INDEX scope_ref_ind (association_id, topicname_id, occurrence_id,
variant_id),
  FOREIGN KEY (association_id) REFERENCES qtm_association(id) ON
DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (topicname_id) REFERENCES qtm_topicname(id) ON DELETE
CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (occurrence_id) REFERENCES qtm_occurrence(id) ON DELETE
CASCADE ON UPDATE CASCADE,

```

```

FOREIGN KEY (variant_id) REFERENCES qtm_variant(id) ON DELETE
CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;

CREATE TABLE qtm_scoperef (
  id INT NOT NULL AUTO_INCREMENT,
  scope_id INT NOT NULL,
  reference INT NOT NULL,
  PRIMARY KEY(id),
  INDEX scref_ref_ind (reference, scope_id),
  FOREIGN KEY (scope_id) REFERENCES qtm_scope(id) ON DELETE CASCADE ON
UPDATE CASCADE,
  FOREIGN KEY (reference) REFERENCES qtm_topic(id) ON DELETE RESTRICT
ON UPDATE CASCADE
) ENGINE=InnoDB;

CREATE TABLE qtm_topicmapconstruct (
  id INT NOT NULL AUTO_INCREMENT,
  association_id INT NULL,
  assocrole_id INT NULL,
  occurrence_id INT NULL,
  topic_id INT NULL,
  topicmap_id INT NULL,
  topicname_id INT NULL,
  variant_id INT NULL,
  reifier INT NULL,
  PRIMARY KEY(id),
  INDEX tmc_ref_ind (association_id, assocrole_id, occurrence_id,
topic_id, topicmap_id, topicname_id, variant_id, reifier),
  FOREIGN KEY (association_id) REFERENCES qtm_association(id) ON
DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (assocrole_id) REFERENCES qtm_assocrole(id) ON DELETE
CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (occurrence_id) REFERENCES qtm_occurrence(id) ON DELETE
CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (topic_id) REFERENCES qtm_topic(id) ON DELETE CASCADE ON
UPDATE CASCADE,
  FOREIGN KEY (topicmap_id) REFERENCES qtm_topicmap(id) ON DELETE
CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (topicname_id) REFERENCES qtm_topicname(id) ON DELETE
CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (variant_id) REFERENCES qtm_variant(id) ON DELETE
CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (reifier) REFERENCES qtm_topic(id) ON DELETE SET NULL ON
UPDATE CASCADE
) ENGINE=InnoDB;

CREATE TABLE qtm_topicmapconstructref (
  id INT NOT NULL AUTO_INCREMENT,
  topicmapconstruct_id INT NOT NULL,
  locator VARCHAR(255) NOT NULL,
  PRIMARY KEY(id),
  INDEX tmcref_ref_ind (topicmapconstruct_id),
  FOREIGN KEY (topicmapconstruct_id) REFERENCES
qtm_topicmapconstruct(id) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;

```

Anhang I – Published Subject Indicators in XTM 1.0 für die zentralen Konzepte des Topic Maps-Tagging-Modells

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE topicMap PUBLIC "-//TopicMaps.Org//DTD XML Topic Map (XTM)
1.0//EN" "http://topicmaps.org/xtm/1.0/xtml.dtd">
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/"
xmlns:xlink="http://www.w3.org/1999/xlink" id="tmodelpsi.xtm">

    <!-- the language topic: language is english -->

    <topic id="english">
        <subjectIdentity>
            <subjectIndicatorRef
xlink:href="http://www.topicmaps.org/xtm/1.0/language.xtm#en"/>
        </subjectIdentity>
        <baseName>
            <baseNameString>english</baseNameString>
        </baseName>
    </topic>

    <!-- the topic hat identifies the topic maps-tagging-model topic
map -->

    <topic id="topic-maps-tagging-model">
        <subjectIdentity>
            <subjectIndicatorRef xlink:href="#psi-tmodel-
description"/>
        </subjectIdentity>
        <baseName>
            <scope>
                <topicRef xlink:href="#english"/>
            </scope>
            <baseNameString>Topic Maps-Tagging-Model</baseNameString>
        </baseName>
        <occurrence>
            <resourceData id="psi-tmodel-description">
                The Topic Maps-Tagging-Model is an implementation
of the conceptual tagging model. The conceptual tagging model
represents the tagging context: the domain of discourse which can be
described by !users that assign tags to resources!
            </resourceData>
        </occurrence>
    </topic>

    <!-- the descriptions of the core concepts of the topic maps-
tagging-model -->

    <!-- topic types -->

    <topic id="tag">
        <subjectIdentity>
            <subjectIndicatorRef xlink:href="#psi-tag-
description"/>
        </subjectIdentity>
        <baseName>
            <scope>
                <topicRef xlink:href="#english"/>
            </scope>
            <baseNameString>tag</baseNameString>
        </baseName>
        <occurrence>
            <resourceData id="psi-tag-description">
                A tag is a keyword that is assigned to a taggable
resource by a tagging user. Assigning a tag is the result of an
intellectual indexing process realised by a tagging user. A tag names
```

```

any subject/concept/idea (representing a real world or an abstract
thing) the tagging user keeps in mind when examining a resource.
    </resourceData>
  </occurrence>
</topic>

  <topic id="taggable-resource">
    <subjectIdentity>
      <subjectIndicatorRef xlink:href="#psi-taggable-
resource-description"/>
    </subjectIdentity>
    <baseName>
      <scope>
        <topicRef xlink:href="#english"/>
      </scope>
    <baseNameString>taggable resource</baseNameString>
    </baseName>
    <occurrence>
      <resourceData id="psi-taggable-resource-
description">
        A taggable resource is a particular object of
information on the Internet which has a unique URI. A resource is the
object of the intellectual indexing process using tags that is
realised by a tagging user.
      </resourceData>
    </occurrence>
  </topic>

  <topic id="tagging-user">
    <subjectIdentity>
      <subjectIndicatorRef xlink:href="#psi-tagging-user-
description"/>
    </subjectIdentity>
    <baseName>
      <scope>
        <topicRef xlink:href="#english"/>
      </scope>
    <baseNameString>tagging user</baseNameString>
    </baseName>
    <occurrence>
      <resourceData id="psi-tagging-user-description">
        A tagging user is an individual that realises
intellectual indexing processes of resources with a unique URI using
tags.
      </resourceData>
    </occurrence>
  </topic>

  <topic id="audio">
    <subjectIdentity>
      <subjectIndicatorRef xlink:href="#psi-audio-
description"/>
    </subjectIdentity>
    <baseName>
      <scope>
        <topicRef xlink:href="#english"/>
      </scope>
    <baseNameString>audio</baseNameString>
    </baseName>
    <occurrence>
      <resourceData id="psi-audio-description">
        Audio specifies the nature of a resource as
information represented by an audio file or audio streaming.
      </resourceData>
    </occurrence>
  </topic>

```

```

    <topic id="video">
      <subjectIdentity>
        <subjectIndicatorRef xlink:href="#psi-video-
description"/>
      </subjectIdentity>
      <baseName>
        <scope>
          <topicRef xlink:href="#english"/>
        </scope>
      <baseNameString>video</baseNameString>
      </baseName>
      <occurrence>
        <resourceData id="psi-video-description">
          Video specifies the nature of a resource as
information represented by a video file or video streaming.
        </resourceData>
      </occurrence>
    </topic>

    <topic id="image">
      <subjectIdentity>
        <subjectIndicatorRef xlink:href="#psi-image-
description"/>
      </subjectIdentity>
      <baseName>
        <scope>
          <topicRef xlink:href="#english"/>
        </scope>
      <baseNameString>image</baseNameString>
      </baseName>
      <occurrence>
        <resourceData id="psi-image-description">
          Image specifies the nature of a resource as
information represented graphically.
        </resourceData>
      </occurrence>
    </topic>

    <topic id="website">
      <subjectIdentity>
        <subjectIndicatorRef xlink:href="#psi-website-
description"/>
      </subjectIdentity>
      <baseName>
        <scope>
          <topicRef xlink:href="#english"/>
        </scope>
      <baseNameString>website</baseNameString>
      </baseName>
      <occurrence>
        <resourceData id="psi-website-description">
          Website specifies the nature of a resource as
information represented by a collection of web pages which are
hypertext documents created with markup languages containing text
and/or image.
        </resourceData>
      </occurrence>
    </topic>

    <topic id="any-content">
      <subjectIdentity>
        <subjectIndicatorRef xlink:href="#psi-any-content-
description"/>
      </subjectIdentity>
      <baseName>
        <scope>
          <topicRef xlink:href="#english"/>

```

```

        </scope>
        <baseNameString>any content</baseNameString>
    </baseName>
    <occurrence>
        <resourceData id="psi-any-content-description">
            Any content specifies the nature of a resource as
unspecified information.
        </resourceData>
    </occurrence>
</topic>

<!-- association types -->

<topic id="assigned-by">
    <subjectIdentity>
        <subjectIndicatorRef xlink:href="#psi-assigned-by-
description"/>
    </subjectIdentity>
    <baseName>
        <scope>
            <topicRef xlink:href="#english"/>
        </scope>
    <baseNameString>assigned-by</baseNameString>
</baseName>
    <occurrence>
        <resourceData id="psi-assigned-by-description">
            Assigned-by specifies the nature of relationships
represented by associations that represent relations between tagging
user and tag.
        </resourceData>
    </occurrence>
</topic>

<topic id="saved-by">
    <subjectIdentity>
        <subjectIndicatorRef xlink:href="#psi-saved-by-
description"/>
    </subjectIdentity>
    <baseName>
        <scope>
            <topicRef xlink:href="#english"/>
        </scope>
    <baseNameString>saved-by</baseNameString>
</baseName>
    <occurrence>
        <resourceData id="psi-saved-by-description">
            Saved-by specifies the nature of relationships
represented by associations that represent relations between tagging
user and taggable resource.
        </resourceData>
    </occurrence>
</topic>

<topic id="specified-by">
    <subjectIdentity>
        <subjectIndicatorRef xlink:href="#psi-specified-by-
description"/>
    </subjectIdentity>
    <baseName>
        <scope>
            <topicRef xlink:href="#english"/>
        </scope>
    <baseNameString>specified-by</baseNameString>
</baseName>
    <occurrence>
        <resourceData id="psi-specified-by-description">
            Specified-by specifies the nature of relationships

```


represented by associations that represent relations between tag and taggable resource.

```
        </resourceData>
    </occurrence>
</topic>

<!-- association role types -->

<topic id="user-to-tag">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="#psi-user-to-tag-
description"/>
  </subjectIdentity>
  <baseName>
    <scope>
      <topicRef xlink:href="#english"/>
    </scope>
  </baseNameString>user-to-tag</baseNameString>
</baseName>
<occurrence>
  <resourceData id="psi-user-to-tag-description">
    User-to-tag specifies the nature of the
participation of the role player "tagging user" in an association
specified by "assigned-by".
  </resourceData>
</occurrence>
</topic>

<topic id="tag-to-user">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="#psi-tag-to-user-
description"/>
  </subjectIdentity>
  <baseName>
    <scope>
      <topicRef xlink:href="#english"/>
    </scope>
  </baseNameString>tag-to-user</baseNameString>
</baseName>
<occurrence>
  <resourceData id="psi-tag-to-user-description">
    Tag-to-user specifies the nature of the
participation of the role player "tag" in an association specified by
"assigned-by".
  </resourceData>
</occurrence>
</topic>

<topic id="user-to-resource">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="#psi-user-to-
resource-description"/>
  </subjectIdentity>
  <baseName>
    <scope>
      <topicRef xlink:href="#english"/>
    </scope>
  </baseNameString>user-to-resource</baseNameString>
</baseName>
<occurrence>
  <resourceData id="psi-user-to-resource-
description">
    User-to-resource specifies the nature of the
participation of the role player "tagging user" in an association
specified by "saved-by".
  </resourceData>
</occurrence>
```

```

</topic>

<topic id="resource-to-user">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="#psi-resource-to-
user-description"/>
  </subjectIdentity>
  <baseName>
    <scope>
      <topicRef xlink:href="#english"/>
    </scope>
  <baseNameString>resource-to-user</baseNameString>
</baseName>
<occurrence>
  <resourceData id="psi-resource-to-user-
description">
    Resource-to-user specifies the nature of the
participation of the role player "taggable resource" in an association
specified by "saved-by".
  </resourceData>
</occurrence>
</topic>

<topic id="tag-to-resource">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="#psi-tag-to-
resource-description"/>
  </subjectIdentity>
  <baseName>
    <scope>
      <topicRef xlink:href="#english"/>
    </scope>
  <baseNameString>tag-to-resource</baseNameString>
</baseName>
<occurrence>
  <resourceData id="psi-tag-to-resource-description">
    Tag-to-resource specifies the nature of the
participation of the role player "tag" in an association specified by
"specified-by".
  </resourceData>
</occurrence>
</topic>

<topic id="resource-to-tag">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="#psi-resource-to-
tag-description"/>
  </subjectIdentity>
  <baseName>
    <scope>
      <topicRef xlink:href="#english"/>
    </scope>
  <baseNameString>resource-to-tag</baseNameString>
</baseName>
<occurrence>
  <resourceData id="psi-resource-to-tag-description">
    Resource-to-tag specifies the nature of the
participation of the role player "taggable resource" in an association
specified by "specified-by".
  </resourceData>
</occurrence>
</topic>
</topicMap>

```

Anhang J – Spezifikation des Topic Maps-Tagging-Modells in XTM 1.0

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE topicMap PUBLIC "-//TopicMaps.Org//DTD XML Topic Map (XTM)
1.0//EN" "http://topicmaps.org/xtm/1.0/xtml.dtd">
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/"
xmlns:xlink="http://www.w3.org/1999/xlink" id="topicmap-1">
  <topic id="topic-1">
    <subjectIdentity>
      <subjectIndicatorRef
xlink:href="http://www.topicmaps.org/xtm/1.0/language.xtm#en"/>
    </subjectIdentity>
    <baseName id="topicname-1">
      <baseNameString>english</baseNameString>
    </baseName>
  </topic>
  <topic id="topic-2">
    <subjectIdentity>
      <subjectIndicatorRef xlink:href="#topicmap-1"/>
    </subjectIdentity>
    <baseName id="topicname-2">
      <scope>
        <topicRef xlink:href="#topic-1"/>
      </scope>
      <baseNameString>Topic Maps-Tagging-Model</baseNameString>
    </baseName>
    <occurrence id="occurrence-1">
      <instanceOf>
        <topicRef xlink:href="#topic-3"/>
      </instanceOf>
      <resourceData>Johannes Schmidt, HAW Hamburg</resourceData>
    </occurrence>
    <occurrence id="occurrence-2">
      <instanceOf>
        <topicRef xlink:href="#topic-4"/>
      </instanceOf>
      <resourceData>2006-11-05</resourceData>
    </occurrence>
    <occurrence id="occurrence-3">
      <instanceOf>
        <topicRef xlink:href="#topic-5"/>
      </instanceOf>
      <resourceData>The Topic Maps-Tagging-Model is an implementation
of the conceptual tagging model. The conceptual tagging model
represents the tagging context: the domain of discourse which can be
described by !users that assign tags to resources!</resourceData>
    </occurrence>
  </topic>
  <topic id="topic-3">
    <subjectIdentity>
      <subjectIndicatorRef
xlink:href="http://purl.org/dc/elements/1.1/creator"/>
    </subjectIdentity>
    <baseName id="topicname-3">
      <baseNameString>Creator</baseNameString>
    </baseName>
  </topic>
  <topic id="topic-4">
    <subjectIdentity>
      <subjectIndicatorRef
xlink:href="http://purl.org/dc/elements/1.1/date"/>
    </subjectIdentity>
    <baseName id="topicname-4">
      <baseNameString>Date</baseNameString>
    </baseName>
  </topic>
  <topic id="topic-5">
    <subjectIdentity>
      <subjectIndicatorRef
xlink:href="http://purl.org/dc/elements/1.1/date"/>
    </subjectIdentity>
    <baseName id="topicname-5">
      <baseNameString>Date</baseNameString>
    </baseName>
  </topic>
</topicMap>
```

```

</topic>
<topic id="topic-5">
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://purl.org/dc/elements/1.1/description"/>
  </subjectIdentity>
  <baseName id="topicname-5">
    <baseNameString>Description</baseNameString>
  </baseName>
</topic>
<topic id="topic-6">
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.topicmaps.org/iso13250/glossary/topic-type"/>
  </subjectIdentity>
  <baseName id="topicname-6">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>Topic Type</baseNameString>
  </baseName>
</topic>
<topic id="topic-7">
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.topicmaps.org/iso13250/glossary/association-
type"/>
  </subjectIdentity>
  <baseName id="topicname-7">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>Association Type</baseNameString>
  </baseName>
</topic>
<topic id="topic-8">
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.topicmaps.org/iso13250/glossary/association-
role-type"/>
  </subjectIdentity>
  <baseName id="topicname-8">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>Association Role Type</baseNameString>
  </baseName>
</topic>
<topic id="topic-9">
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.topicmaps.org/iso13250/glossary/occurrence-
type"/>
  </subjectIdentity>
  <baseName id="topicname-9">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>Occurrence Type</baseNameString>
  </baseName>
</topic>
<topic id="topic-10">
  <instanceOf>
    <topicRef xlink:href="#topic-6"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef

```

```

xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tag-
description"/>
  </subjectIdentity>
  <baseName id="topicname-10">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>tag</baseNameString>
  </baseName>
</topic>
<topic id="topic-11">
  <instanceOf>
    <topicRef xlink:href="#topic-6"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-taggable-
resource-description"/>
  </subjectIdentity>
  <baseName id="topicname-11">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>taggable resource</baseNameString>
  </baseName>
</topic>
<topic id="topic-12">
  <instanceOf>
    <topicRef xlink:href="#topic-6"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tagging-user-
description"/>
  </subjectIdentity>
  <baseName id="topicname-12">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>tagging user</baseNameString>
  </baseName>
</topic>
<topic id="topic-13">
  <instanceOf>
    <topicRef xlink:href="#topic-6"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-audio-
description"/>
  </subjectIdentity>
  <baseName id="topicname-13">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>audio</baseNameString>
  </baseName>
</topic>
<topic id="topic-14">
  <instanceOf>
    <topicRef xlink:href="#topic-6"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-video-
description"/>
  </subjectIdentity>

```

```

    <baseName id="topicname-14">
      <scope>
        <topicRef xlink:href="#topic-1"/>
      </scope>
      <baseNameString>video</baseNameString>
    </baseName>
  </topic>
  <topic id="topic-15">
    <instanceOf>
      <topicRef xlink:href="#topic-6"/>
    </instanceOf>
    <subjectIdentity>
      <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-image-
description"/>
    </subjectIdentity>
    <baseName id="topicname-15">
      <scope>
        <topicRef xlink:href="#topic-1"/>
      </scope>
      <baseNameString>image</baseNameString>
    </baseName>
  </topic>
  <topic id="topic-16">
    <instanceOf>
      <topicRef xlink:href="#topic-6"/>
    </instanceOf>
    <subjectIdentity>
      <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-any-content-
description"/>
    </subjectIdentity>
    <baseName id="topicname-16">
      <scope>
        <topicRef xlink:href="#topic-1"/>
      </scope>
      <baseNameString>any content</baseNameString>
    </baseName>
  </topic>
  <topic id="topic-17">
    <instanceOf>
      <topicRef xlink:href="#topic-6"/>
    </instanceOf>
    <subjectIdentity>
      <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-website-
description"/>
    </subjectIdentity>
    <baseName id="topicname-17">
      <scope>
        <topicRef xlink:href="#topic-1"/>
      </scope>
      <baseNameString>website</baseNameString>
    </baseName>
  </topic>
  <topic id="topic-18">
    <instanceOf>
      <topicRef xlink:href="#topic-7"/>
    </instanceOf>
    <subjectIdentity>
      <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-assigned-by-
description"/>
    </subjectIdentity>
    <baseName id="topicname-18">
      <scope>
        <topicRef xlink:href="#topic-1"/>

```

```

        </scope>
        <baseNameString>assigned-by</baseNameString>
    </baseName>
</topic>
<topic id="topic-19">
    <instanceOf>
        <topicRef xlink:href="#topic-7"/>
    </instanceOf>
    <subjectIdentity>
        <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-saved-by-
description"/>
        </subjectIdentity>
        <baseName id="topicname-19">
            <scope>
                <topicRef xlink:href="#topic-1"/>
            </scope>
            <baseNameString>saved-by</baseNameString>
        </baseName>
    </topic>
<topic id="topic-20">
    <instanceOf>
        <topicRef xlink:href="#topic-7"/>
    </instanceOf>
    <subjectIdentity>
        <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-specified-by-
description"/>
        </subjectIdentity>
        <baseName id="topicname-20">
            <scope>
                <topicRef xlink:href="#topic-1"/>
            </scope>
            <baseNameString>specified-by</baseNameString>
        </baseName>
    </topic>
<topic id="topic-21">
    <instanceOf>
        <topicRef xlink:href="#topic-8"/>
    </instanceOf>
    <subjectIdentity>
        <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-user-to-tag-
description"/>
        </subjectIdentity>
        <baseName id="topicname-21">
            <scope>
                <topicRef xlink:href="#topic-1"/>
            </scope>
            <baseNameString>user-to-tag</baseNameString>
        </baseName>
    </topic>
<topic id="topic-22">
    <instanceOf>
        <topicRef xlink:href="#topic-8"/>
    </instanceOf>
    <subjectIdentity>
        <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tag-to-user-
description"/>
        </subjectIdentity>
        <baseName id="topicname-22">
            <scope>
                <topicRef xlink:href="#topic-1"/>
            </scope>
            <baseNameString>tag-to-user</baseNameString>
        </baseName>

```

```

</topic>
<topic id="topic-23">
  <instanceOf>
    <topicRef xlink:href="#topic-8"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-user-to-
resource-description"/>
  </subjectIdentity>
  <baseName id="topicname-23">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>user-to-resource</baseNameString>
  </baseName>
</topic>
<topic id="topic-24">
  <instanceOf>
    <topicRef xlink:href="#topic-8"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-resource-to-
user-description"/>
  </subjectIdentity>
  <baseName id="topicname-24">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>resource-to-user</baseNameString>
  </baseName>
</topic>
<topic id="topic-25">
  <instanceOf>
    <topicRef xlink:href="#topic-8"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-tag-to-
resource-description"/>
  </subjectIdentity>
  <baseName id="topicname-25">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>tag-to-resource</baseNameString>
  </baseName>
</topic>
<topic id="topic-26">
  <instanceOf>
    <topicRef xlink:href="#topic-8"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef
xlink:href="http://psi.t8d.de/tmodel/tmodelpsi.xtm#psi-resource-to-
tag-description"/>
  </subjectIdentity>
  <baseName id="topicname-26">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>resource-to-tag</baseNameString>
  </baseName>
</topic>
<topic id="topic-27">
  <instanceOf>

```



```

    <topicRef xlink:href="#topic-9"/>
  </instanceOf>
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="http://www.w3.org/TR/2004/REC-
xmlschema-2-20041028/datatypes.html#date"/>
  </subjectIdentity>
  <baseName id="topicname-27">
    <scope>
      <topicRef xlink:href="#topic-1"/>
    </scope>
    <baseNameString>date</baseNameString>
  </baseName>
</topic>
<association id="association-1">
  <instanceOf>
    <topicRef xlink:href="#topic-18"/>
  </instanceOf>
  <member>
    <roleSpec>
      <topicRef xlink:href="#topic-22"/>
    </roleSpec>
    <topicRef xlink:href="#topic-10"/>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#topic-21"/>
    </roleSpec>
    <topicRef xlink:href="#topic-12"/>
  </member>
</association>
<association id="association-2">
  <instanceOf>
    <topicRef xlink:href="#topic-19"/>
  </instanceOf>
  <member>
    <roleSpec>
      <topicRef xlink:href="#topic-24"/>
    </roleSpec>
    <topicRef xlink:href="#topic-11"/>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#topic-23"/>
    </roleSpec>
    <topicRef xlink:href="#topic-12"/>
  </member>
</association>
<association id="association-3">
  <instanceOf>
    <topicRef xlink:href="#topic-20"/>
  </instanceOf>
  <member>
    <roleSpec>
      <topicRef xlink:href="#topic-25"/>
    </roleSpec>
    <topicRef xlink:href="#topic-10"/>
  </member>
  <member>
    <roleSpec>
      <topicRef xlink:href="#topic-26"/>
    </roleSpec>
    <topicRef xlink:href="#topic-11"/>
  </member>
</association>
</topicMap>

```

Anhang K – Installationshilfe für die PHPTMAPI-Implementierung (QuaaxTM)

Installationsvoraussetzungen:

- PHP 5.x
- MySQL mit InnoDB

Im folgenden alle Beispiele mit `http://localhost/`

1. Kopieren Sie das Verzeichnis `quaaxtm` aus dem Verzeichnis `quaaxtm` der beigelegten CD-ROM in das Document-Root-Verzeichnis Ihres Webservers.
HINWEIS: Sie können das System im Verzeichnis `src` beliebig positionieren, wenn Sie die include-Pfade der Klassen im Verzeichnis `http://localhost/quaaxtm/src/phptmapi/core1.1` und unter 5. entsprechend anpassen. Beachten Sie in diesem Fall auch den Pfad in der Datei `TopicMapSystemFactoryImpl` (siehe 2.) sowie die Pfade in den u. g. Bsp.
2. Rufen Sie im Browser die Datei `http://localhost/quaaxtm/tmodel/documentroot.php` und ersetzen Sie in der Datei `http://localhost/quaaxtm/META/factory/TopicMapSystemFactoryImpl [DOCUMENT_ROOT]` durch den im Browser angezeigten Pfad.
3. Erstellen Sie die Datenbank mit der Datei `quaaxtm_db_build.sql` aus dem Verzeichnis `quaaxtm/quaaxtm_db_build` der beigelegten CD-ROM.
4. Editieren Sie die Datei `http://localhost/quaaxtm/src/config/config.php` gemäß den Einstellungen Ihrer Datenbank.
5. Rufen Sie im Browser die Datei `http://localhost/quaaxtm/tmodel/system_runttest.php`, um die Betriebsbereitschaft festzustellen.

Dateien `http://localhost/quaaxtm/tmodel/build_tmodel_basic.php` und `http://localhost/quaaxtm/tmodel/build_tmodel_assocs.php` implementieren das Topic Maps-Tagging-Modell

Datei `http://localhost/quaaxtm/tmodel/tmodel/serialize_tmodel.php` exportiert jeweils XTM 1.0 (Browser – Ansicht – Quelltext), auch für die Beispiele:

Dateien `http://localhost/quaaxtm/tmodel/tmodel/use_tmodel_1.php` und `http://localhost/quaaxtm/tmodel/tmodel/use_tmodel_2.php` sind die Anwendungsbeispiele

Datei `http://localhost/quaaxtm/tmodel/delete_tmodel.php` leert die Datenbank

Anhang L – E-Mail von Graham Moore über die Topic Maps-E-Mailliste zum weiteren Vorgehen bzgl. TMWSI

Betreff: RE: [topicmapmail] Topic Map Web Services
Von: "Graham Moore" <graham.moore@networkedplanet.com>
Datum: Thu, 19 Oct 2006 19:48:59 +0200
An: "'Stefan Lischke'" <lischke@novacom.net>, <topicmapmail@infoloom.com>

Hi,

I brought up at the ISO meeting about undertaking a work item there to define some requirements, survey current approaches and then get something together that maybe some of us could agree on.

However, it was decided that this was not a thing for ISO to do and should instead be done in the community much the same way that TMAPI got done. Given this, I am creating a new project on Sourceforge called Topic Maps Web Service Interface (tmwsi) where I shall post an initial requirements document.

Having people submit their current thinking and any available documentation to the process would be very useful at this stage.

Subsequently, I plan to post a straw man proposal that tries to capture the powerful aspects of the different services I have been exposed to (such as, TMCORE TMWS, TMRAP, RDFNet API, SNAPI, the work by Robert Barta and that of Robert Cerny)

I would also encourage any lurkers on this list who are interested in this subject to get a SF account and get involved. Kal Ahmed, Lars Marius and I (with a good deal of community contribution) managed an OK job on TMAPI, and I hope we can match or better that effort. It will certainly be quicker and less painful than ISO.

I will post the SF project link as soon as the nice people there let me have it. Let's go do it!

Cheers,

Graham

--

Graham Moore, Founder, NetworkedPlanet
Editor XTM 1.0, ISO13250 (TopicMaps) -2,-3, TMCL
e: graham.moore@networkedplanet.com
w: www.networkedplanet.com
m: +47 45271713
p: +44 (0)1865 811131

**Anhang M – CD-ROM, enthält PHPTMAPI 1.0, PHPTMAPI-
Implementierung (QuaaxTM), XTM-Beispiele**

Eidesstattliche Versicherung

Ich versichere, die vorliegende Arbeit selbständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangabe kenntlich gemacht.

Hamburg, 15.12.2006

Johannes Schmidt