

SNS – Navigator

**Entwicklung einer grafischen
Navigationskomponente für ein semantisches
Netzwerk auf der Basis eines Web Services des
Umweltbundesamtes**

Diplomarbeit

im Fach Informationstechnik
Studiengang Informationswirtschaft
der
Fachhochschule Stuttgart –
Hochschule der Medien

Jörg Jochims

Erstprüfer:
Zweitprüfer:

Prof. Dr. Wolf-Fritz Riekert
Prof. Dr. Jörg Westbomke

Bearbeitungszeitraum: 1. April 2006 bis 27. Juli 2006

Stuttgart, Juli 2006

Kurzfassung

Gegenstand der vorliegenden Arbeit ist die Entwicklung und Präsentation einer grafischen Navigationskomponente, die miteinander vernetzte Begriffe aus der Umweltterminologie in Form einer interaktiven Grafik anzeigt und eine Navigation innerhalb dieser Begriffe ermöglicht. Die Webapplikation nutzt hierfür den semantischen Netzwerkdienst (SNS) des Umweltbundesamtes. Die im XML Topic Map-Format vorliegenden Datenbestände des Umweltbundesamtes werden auf Basis von Javascript (Ajax), SVG, XSLT und PHP ausgewertet und visualisiert. Es erfolgt zunächst eine kurze Einführung in die Themenkomplexe Web Services und Semantic Web, gefolgt von einer Beschreibung, wie sich Web Services in eine Webanwendung einbinden lassen und wie das bei der hier vorgestellten Webapplikation gelöst worden ist. Im nächsten Schritt wird auf die genormte Struktur und Elemente von Topic Maps eingegangen. Die Kapitel Konzeption und Realisierung beschreiben detailliert die funktionalen und gestalterischen Aspekte der vorliegenden Arbeit. Im Schlusskapitel wird anhand von verschiedenen Szenarien der Einsatz der Webapplikation dargestellt.

Schlagwörter: Semantic Web, Web Service, Topic Map, semantisches Netz, Umweltinformationen, Suchmethoden, Thesaurus, Internet, Navigationskomponente, Umweltbundesamt, Javascript, XML, XSLT, Ajax, SVG, PHP

Abstract

The subject of this thesis is the development and presentation of a interactive graphical system that shows networked environmental terms as an interactive icon with the added function of navigation. The application is written in PHP, SVG and Javascript(Ajax) based on a Web Service called Semantic Network Service (SNS) of the Federal Environmental Agency. The thesis gives an introduction of the topics Web Services and Semantic Web and how this parts work together and explains how Web Services could be included in a web application. In addition the thesis includes a description of all well-defined elements and structure of XML Topic Maps, which are evaluated by the web application. The chapter Concept and Execution describes the functional and creative aspects of this work. The chapter Result includes various scenarios describing how the application could be used by potential users.

Keywords: Semantic Web, Web Service, Topic Map, Semantic Network, environmental informations, search methods, thesaurus, internet, graphical system, Federal Environmental Agency, Javascript, XML, XSLT, Ajax, SVG, PHP

Inhaltsverzeichnis

Kurzfassung	2
Abstract	2
Inhaltsverzeichnis	3
Abbildungsverzeichnis	5
Tabellenverzeichnis	6
Abkürzungsverzeichnis	7
Vorwort	9
1 Überblick	10
2 Ausgangspunkt	12
3 Ziele	14
4 Technologien	15
4.1 Web Services	15
4.2 Semantische Netze und Semantic Web	19
4.3 Topic Maps	23
4.3.1 Der ISO-Standard 13250	24
4.3.2 Die Basiskonzepte von Topic Maps	24
4.3.3 Elemente einer XML Topic Map	27
4.4 Kommunikation mit dem Web Service	32
4.4.1 Funktionen des SNS Web Service	34
4.4.2 Web Services und Protokolle	35
4.4.3 SOAP vs. REST	36
4.5 Struktur der XML-Definitionen von SNS	37
4.6 Scalable Vector Graphics (SVG)	39
4.7 Javascript und Ajax	40
4.8 PHP	40
5 Anforderungen und technische Lösungsansätze	41
5.1 Use Cases	41
5.2 Empfehlungen des Umweltbundesamtes für die Einbindung des Services beim Nutzer	43
5.2.1 Axis	43
5.2.2 .NET Framework	44
5.3 Gewählter Lösungsansatz	44

6	Konzeption	45
6.1	Funktionaler Aufbau und Struktur der Webapplikation	45
6.1.1	Three-Tier-Architektur.....	45
6.1.2	Anwendungsarchitektur des SNS-Navigators.....	46
6.1.3	Einsatzbereiche der verschiedenen Webtechnologien	47
6.1.4	Proxy.....	47
6.1.5	Struktur der Ein- und Ausgabeseiten	48
6.1.6	Darstellung aller Komponenten der Anwendung	48
6.1.7	Die „Zufallssuche“	49
6.1.8	Model-View-Controller-Konzept.....	49
6.1.9	Aktivitätsdiagramme.....	49
6.2	Gestaltung der Webapplikation.....	52
6.2.1	Aufteilung der Benutzungsoberfläche in verschiedene Bereiche.....	52
6.2.2	Farblegenden der Topic- und Assoziationstypen.....	53
6.2.3	Anzeige aller gefundenen Topics als Liste	55
6.2.4	Grafische Anzeige der Topic Map.....	55
6.3	Technische Voraussetzungen.....	57
7	Realisierung	58
7.1	Web Service Anfrage mittels GET / POST Binding durch einen Proxy	58
7.2	XML-Transformation der zurückgelieferten Topic Map.....	59
7.3	Darstellung der SVG-Anzeige in einem I Frame	60
7.4	Realisierung der Statusanzeige mit Ajax	61
7.5	Entwurf der Icons (Topic types der SNS-Topic Map).....	64
7.6	Struktur der XML-Stylesheets	65
7.6.1	Listenanzeige aller gefundenen Topics (HTML-Ausgabe)	65
7.6.2	Grafikanzeige der XML Topic Map mit SVG	69
7.7	Interaktivität der SVG Grafik mit Javascript	76
8	Ergebnis.....	80
9	Zusammenfassung	88
	Anhang: Datenträger	90
	Quellenverzeichnis	91
	Erklärung	95

Abbildungsverzeichnis

Abbildung 1: Next Web Generation.....	11
Abbildung 2: Interoperabilität von Web Services	15
Abbildung 3: Geschichte des "distributed computing".....	16
Abbildung 4: Web Services	18
Abbildung 5: Darstellung eines semantischen Netzes	20
Abbildung 6: Semantisches Netz	21
Abbildung 7: Semantische Web Services als Symbiose von Semantik (Yin) und Syntax (Yang).....	23
Abbildung 8: Aufbau einer Topic Map	24
Abbildung 9: Topics.....	25
Abbildung 10: Topic types.....	25
Abbildung 11: Occurrences.....	26
Abbildung 12: Topic associations.....	26
Abbildung 13: Topic Map (Klassendiagramm)	28
Abbildung 14: Base Name im Inneren eines Scopes (Klassendiagramm).....	29
Abbildung 15: Occurrence (Klassendiagramm)	30
Abbildung 16: Assoziationen zwischen Topics (Klassendiagramm)	30
Abbildung 17: Distanztiefe der assoziierten Topics	34
Abbildung 18: Naturpark-Icon als SVG Grafik.....	39
Abbildung 19: Allgemeine Use Cases des SNS-Navigators	41
Abbildung 20: Erweitertes Use-Case-Diagramm des SNS-Navigators	42
Abbildung 21: Axis Client- und Server Engine	43
Abbildung 22: Systemkomponenten und Kontrollflüsse	46
Abbildung 23: Anwendungsarchitektur des SNS-Navigators	46
Abbildung 24: Darstellung und Gliederung aller Komponenten	48
Abbildung 25: UML - Aktivitätsdiagramm Legende	49
Abbildung 26: Schritt 1: Suchbegriff Eingabe.....	50
Abbildung 27: Schritt 2: Topic aus Liste auswählen.....	50
Abbildung 28: Schritt 3: Grafische Navigationskomponente	51
Abbildung 29: Aufteilung des Browserfensters in einzelne Bereiche	53
Abbildung 30: Topics in Kreisform vs. rechteckige Form	56
Abbildung 31: Bereiche der grafischen Navigationskomponente.....	56
Abbildung 32: Statusanzeige: Wartemodus	61
Abbildung 33: Statusanzeige: Ladevorgang	61
Abbildung 34: Statusanzeige: Ladevorgang abgeschlossen	62
Abbildung 35: Statusanzeige: Ladefehler	62
Abbildung 36: Topic Map Auszug (findTopics) mit Erklärungen	66
Abbildung 37: Screenshot der Listenanzeige.....	68
Abbildung 38: Topic Map Auszug (getPSI) mit Erklärungen	71
Abbildung 39: Einzelne Schritte der Positionierung der äußeren Topics	74
Abbildung 40: Screenshot der SVG-Anzeige	76
Abbildung 41: Szenario 1: Startseite.....	80

Abbildung 42: Szenario 1: Willkommensseite	81
Abbildung 43: Szenario 1: Umweltereignis Suchformular	81
Abbildung 44: Szenario 1: Fehlermeldung	81
Abbildung 45: Szenario 1: Listenanzeige	82
Abbildung 46: Szenario 1: Grafische Navigationskomponente Bsp. 1.....	82
Abbildung 47: Szenario 1: Auswahl eines äußeren Topics.....	83
Abbildung 48: Szenario 1: Auswahldialog für externe Links	83
Abbildung 49: Szenario 1: Externe Informationsressource	84
Abbildung 50: Szenario 1: Grafische Navigationskomponente Bsp. 2.....	84
Abbildung 51: Szenario 1: Grafische Navigationskomponente Bsp.3.....	85
Abbildung 52: Szenario 2: Listenanzeige	85
Abbildung 53: Szenario 2: Grafische Navigationskomponente Bsp.4.....	86
Abbildung 54: Szenario 3: Zufällig ausgewählte Topic Map	86
Abbildung 55: Szenario 3: Grafische Navigationskomponente Bsp.5.....	87
Abbildung 56: Szenario 3: Externe Informationsressource	87

Tabellenverzeichnis

Tabelle 1: Komplette Auflistung aller XTM-Elemente.....	27
Tabelle 2: Verbindungstypen	28
Tabelle 3: Zustände für das XMLHttpRequest-Objekt	61
Tabelle 4: Auflistung aller Topic Typen und die vom Autor entworfenen zugehörigen Icons	64

Abkürzungsverzeichnis

API	Application Programming Interface
ARPANET	Advanced Research Projects Agency Network
AXIS	Apache eXtensible Interaction System
CORBA	Common Object Request broker Architecture
CSS	Cascading Style Sheets
DCE	Distributed Computing Environment
FTP	File Transfer Protocol
gein®	German Environmental Information Network
HTML	Hypertext Markup Language
HTTP	Hyper Text Transfer Protocols
ISO	International Organization for Standardization
PSI	Published Subject Identifier
RDF	Resource Description Framework
REST	Representational State Transfer
RMI	Remote Method Invocation
SGML	Standard Generalized Markup Language
SMTP	Simple Mail Transfer Protocol
SNS	Semantischer Netzwerkservice
SOAP	Simple Object Access Protocol
SVG	Scalable Vector Graphics
UBA	Umweltbundesamt
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WSDL	Web Service Description Language
WWW	World Wide Web
XML	Extensible Markup Language

XSL	Extensible Stylesheet Language
XSLT	XSL Transformation
XTM	XML Topic Maps

Vorwort

Für Idee und Anstoß zu dieser Arbeit, Empfehlungen zum Aufbau und die rundum sehr gute Betreuung danke ich recht herzlich Herrn Prof. Dr. Wolf-Fritz Riekert.

Danken möchte ich auch Herrn Prof. Dr. Westbomke für die Übernahme des Amts des Zweitprüfers und die Hinweise bezüglich der gestalterischen Aspekte der Webapplikation.

Ein weiterer Dank geht an Herrn Hoffmann für die Bereitstellung des benötigten Web Space auf dem Studentenserver der Hochschule der Medien und die zusätzliche Installation von PHP-Erweiterungen, die für die Anwendung notwendig waren.

Ich bitte um Verständnis, das die vorliegende Arbeit aus Platz- und Übersichtlichkeitsgründen nicht auf alle Grundlagen der Informationstechnik eingehen kann, wie z.B. auf die Metasprache XML oder die Auszeichnungssprache HTML.

1 Überblick

Mit der zunehmenden Vernetzung und den expansiv steigenden Informationsquellen im World Wide Web sind nicht nur Informationen und Wissen zu fast allen Bereichen der menschlichen Existenz verfügbar, sondern es wird auch zusehends schwieriger aus diesem reichhaltigen Angebot die gesuchten Informationen zu isolieren und zu verwerten. Um so größer ist das Bestreben, dieser Situation durch ausgefeilte Suchmechanismen und -methoden gerecht zu werden. Momentan verwendet die Mehrheit der Internetsurfer Suchmaschinen wie z.B. Google, um zu den richtigen Informationsressourcen zu gelangen.

Zwar sind die aktuellen Suchmaschinen im Gegensatz zu vor einigen Jahren schon weiter entwickelt und beziehen z.B. auch die Verlinkung der Websites bei ihrem Ranking mit ein, aber die Gesamtsituation ist dennoch unbefriedigend. Ein häufig anzutreffendes Problem bei dem Einsatz von volltextbasierten Suchmaschinen ist, dass man adäquate Suchbegriffe benötigt, um gute Suchergebnisse zu erzielen. Das ist ein Dilemma, denn wie soll der Suchende einen Suchbegriff kennen, den er erst nach einer erfolgreichen Suche kennen kann? Ärgerlich ist ebenfalls, dass bei der Treffermenge homonyme¹ Begriffe auftauchen können (z.B. „Ajax“ in Bezug zur Web-Programmierung und die Fußballmannschaft Ajax Amsterdam) und so der gesuchte Begriff unter Umständen so weit von dem anderen Begriff verdrängt wurde, dass er erst sehr weit unten in der Trefferliste zu finden ist (Bsp.: Die Fußballmannschaft namens „Ajax“ erscheint bei google.com erst auf der zweiten Seite. Stand: 7. Mai 2006). Außerdem sprengt die gefundene Treffermenge häufig den von einem Menschen zu überblickenden Rahmen. Des Weiteren wäre es viel effizienter, wenn auch nichttextuelle Informationsressourcen wie z.B. Bilder, Videos und Musik einer gewissen Kategorie oder einem Begriff zugeordnet sind.

Viele der angesprochenen Probleme fußen auf der Tatsache, dass die bisherige Entwicklung der Internet-Technologie eher in einem syntaktischen Rahmen definiert worden ist und der Aspekt der Bedeutung (Semantik) nicht ausreichend berücksichtigt wurde.²

Eine mögliche Lösung für dieses Problem läge darin, die Inhalte des Netzes mit einer für Menschen und Maschinen verständlichen Metasprache zu erschließen.

Für diese „neue Generation des Internets“ werden in Fachkreisen aktuell die zwei vom World Wide Web Consortium (W3C) vorgeschlagenen Standards Web Services (s. Kap. 4.1) und Semantic Web (s. Kap. 4.2) diskutiert, die durch ihre Kombination von

¹ Homonyme sind gleichlautende, bzw. gleich geschriebene Wörter mit unterschiedlichen Bedeutungen. Beispiel: „Bank“ als Sitzmöbel und als Kreditinstitut.

² Vgl. Dostal, W. & Jeckle, M. (2004).

Automation (Web Services) und Semantik (Semantic Web) das neue Zeitalter des Internets herbeiführen sollen (s. untere Abbildung).

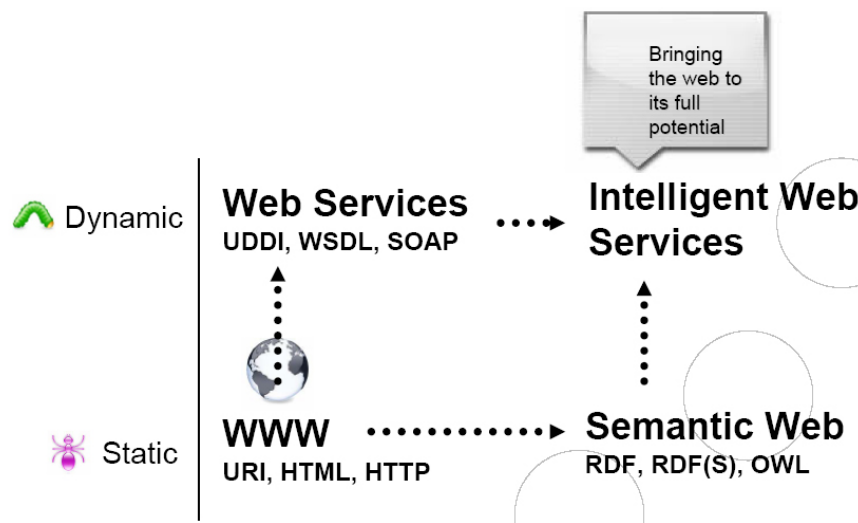


Abbildung 1: Next Web Generation³

Die ersten Schritte in Richtung dieses „Internets mit größerem Potential“ wurden schon durch zahlreiche Projekte begangen, unter ihnen auch der in dieser Arbeit vorgestellte Semantische Netzwerk Service (SNS) des Umweltbundesamtes.

³ Fensel, D. (2004).

2 Ausgangspunkt

Im Mai 2003 wurde im Auftrag des Umweltbundesamtes der semantische Netzwerkservice (SNS) für das Umweltinformationsnetz Deutschland erstellt.⁴ In der Kurzfassung des Forschungsberichts heißt es diesbezüglich:

„Aufbauend auf die Erfolge des Umweltinformationsnetz Deutschland (gein®, <http://www.gein.de>) mit der thesaurus-basierten, automatischen Verschlagwortung von Umweltinformation wurden die in gein® verwendeten Wortgut-Bausteine in eine übergreifende Topic Map integriert und weitergehend miteinander vernetzt. [...] Jeglicher Zugriff auf das Wortgut und die darauf basierenden Methoden konnte als Web Service bereitgestellt werden.“

Wie auch bei gein® integriert der Web Service die Thesauri „UBA-Umweltthesaurus UmThes®“, welcher sich über Umweltbegriffe erstreckt und den UBA-Geo-Thesaurus, der geographischen Angaben von Gemeinden, Bergen etc. unter Miteinbeziehung des sogenannten Umweltkalenders enthält, der die Funktion eines Thesaurus für Zeitangaben erfüllt. Eine wesentliche Motivation für die Erstellung des semantischen Netzwerkservice (SNS) war die Verbesserung der thesaurusgestützten Verschlagwortung von Internetseiten und die optimierte Integration zwischen Umwelt- und Geothesaurus (Thema und Raum) unter Berücksichtigung des Umweltkalenders (Umweltereignisse). Des Weiteren wurde das Problem der zu geringen Beachtung von Homonymen und zusammengesetzten Begriffen, das wie eingangs beschrieben auch bei herkömmlichen Suchmaschinen auftritt, durch den Einsatz einer Topic Map behoben.⁵

Ein weiterer, gewichtiger Grund für die Erstellung des semantischen Netzwerkdienstes war die Schaffung der Möglichkeit, die Daten auf verschiedensten Plattformen zu verteilen und anbieten zu können, unter Miteinbeziehung externer Informationsanbieter. Für dieses Vorhaben wurde auf die im Jahre 2003 bereits entwickelten und standardisierten Web Services zurückgegriffen.⁶

„Ziel des SNS-Vorhabens war es, die beiden genannten Ansätze zu vereinen und im Ergebnis Topic-Map-basierte semantische Dienste als Web Service anzubieten.“⁷

Der semantische Network Service bietet derzeit Zugriff auf rund 53.000 Topics (33.759 vernetzte Thesauruseinträge, 18.931 geographische Namen und Lagebeziehungen

⁴ Vgl. Umweltforschungsplan des Bundesministers für Umwelt, Naturschutz und Reaktorsicherheit – Forschungsbericht 201 11 612, Band 1, Titelblatt.

⁵ Vgl. Umweltforschungsplan des Bundesministers für Umwelt, Naturschutz und Reaktorsicherheit – Forschungsbericht 201 11 612, Band 1, S. 10.

⁶ Vgl. Umweltforschungsplan des Bundesministers für Umwelt, Naturschutz und Reaktorsicherheit – Forschungsbericht 201 11 612, Band 1, S. 11.

und 539 bedeutende umweltrelevante Ereignisse) und wird vom Umweltbundesamt bereit gestellt, um eine gemeinsame Fachsprache in der deutschen Umweltinformation zu fördern und diese Terminologie ebenso der Öffentlichkeit zugänglich zu machen.⁸

Neben dem Web Service-Zugang, der der Öffentlichkeit offen steht, hat das Umweltbundesamt auch eine Webanwendung erstellt, welche auf die beschriebene Umweltdatenbasis und die Web Services aufbaut und unter der URL <http://www.semantic-network.de> erreichbar ist.

⁷ S. Umweltforschungsplan des Bundesministers für Umwelt, Naturschutz und Reaktorsicherheit – Forschungsbericht 201 11 612, Band 1, S. 11.

⁸ Vgl. SNS (2006).

3 Ziele

Das grundsätzliche Ziel dieser Arbeit ist es aufzuzeigen, wie mit Hilfe einer Webapplikation ein Web Service genutzt werden kann, um die nach einer in der Webapplikation implementierten Suche zurück gelieferten Daten in grafisch ansprechender Form als eine Art semantisches Netz (Topic Map) darzustellen und eine effiziente Navigation in dieser interaktiven Grafik zu gewährleisten.

Der Hauptaugenmerk liegt demnach auf der grafischen Visualisierung einer Topic Map, die nicht nur die einzelnen Umweltbegriffe und ihre Beziehungen aufzeigt, sondern ebenfalls eine Navigation innerhalb des dargestellten Ausschnitts einer Topic Map ermöglicht. Zusätzliche Informationen, die mit einem Topic verknüpft sind, werden in ansprechender Art und Weise für den Nutzer aufbereitet. Externe Informationsressourcen, die einem Umweltbegriff zugeordnet sind, sollen über diese grafische Navigationskomponente vom Nutzer ebenfalls erschlossen werden können.

Aus diesen Anforderungen resultiert ein hohes Maß an Interaktivität und Übersichtlichkeit, die die Webanwendung bieten sollte.

Während die bereits bestehende Webapplikation des Umweltbundesamts (s. <http://www.semantic-network.de>) aus Sicht des Autors sich schwerpunktmäßig an einen fachspezifischen Nutzerkreis wie z.B. Mitarbeitern des Umweltbundesamtes oder Umweltexperten wendet und im Weiteren mehr einen Präsentationscharakter der dargebotenen Web Service - Funktionen besitzt, soll die hier vorgestellte Anwendung auch einen nichtfachkundigen Nutzerkreis ansprechen.

Dieser Anspruch kann nur unter Zuhilfenahme von gestalterischen Mitteln verwirklicht werden, die die Nutzbarkeit der hier vorgestellten Webapplikation unterstützen und vereinfachen.

Eine weitere Zielsetzung dieser Anwendung besteht demnach darin, die in ihr implementierten Suchfunktionen nach Begriffen aus der Umweltterminologie zu einem „spielerisch leicht zugänglichen Erlebnis“ zu machen.

Auf funktionaler Ebene ist es erklärtes Ziel, die Anwendung durch einzelne Komponenten, die eine definierte Rolle haben, überschaubar und geordnet zu konzipieren.

Am 12.6.2006 veröffentlichte das Umweltbundesamt die zweite Version des Semantische Netzwerkservices, welche über erweiterte Web Service-Funktionen verfügt, die jedoch in der bestehenden Webanwendung des UBA noch nicht implementiert worden sind (Stand: 24.7.06). Hieraus leitet sich eine weitere Zielsetzung für die zu entwickelnde Webapplikation ab, die im Gegensatz zu älteren Webanwendungen, die auf der Version 1.0 des SNS aufbauen, auch einen Teil der neuen Web Services anbieten möchte.

4 Technologien

In diesem Kapitel werden die Technologien, Standards und Softwarearchitekturen, die bei der vom Autor entwickelten und in dieser Arbeit vorgestellten Webapplikation eine besondere Rolle spielen und zur Anwendung kamen, beschrieben.

4.1 Web Services

Web Services sind ein relativ neuer Standard, welcher die Kommunikation zwischen Anwendungen über gewöhnliche Internetverbindungen, d.h. aufbauend auf Transportprotokollen wie HTTP, FTP oder SMTP bewerkstelligt. Der Hauptfokus liegt hierbei in der Interoperabilität (s. Abb. unten) und Plattform/Programmiersprachenunabhängigkeit, um verteilte Anwendungen in heterogenen Systemlandschaften ohne die zusätzliche Verteilung von Softwarekomponenten zu ermöglichen.⁹

Web Services sind speziell für die „Maschine-zu-Maschine-Kommunikation“ ausgelegt, da sie den automatisierten Datenaustausch zwischen Softwaresystemen unterstützen und Funktionen auf anderen Rechnern ausführen können.

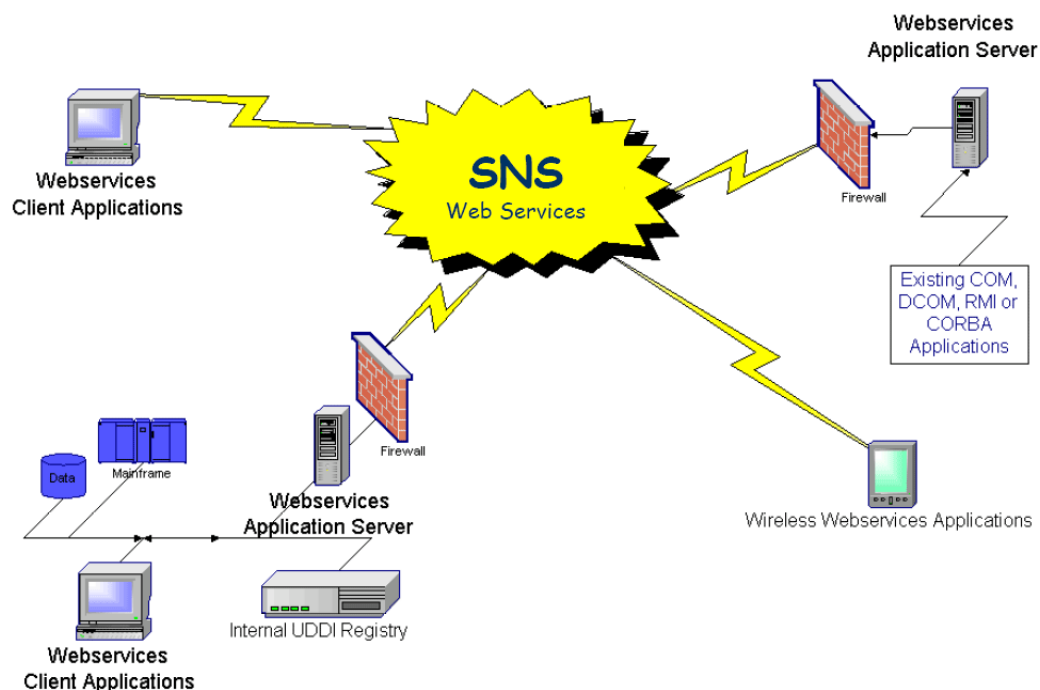


Abbildung 2: Interoperabilität von Web Services¹⁰

⁹ Vgl. Bandholtz, T. (2003), S. 39.

¹⁰ Bandholtz, T. (2003), S.39.

Web Services sind derzeit das populärste Konzept einer Service-orientierten Architektur (SOA), die als Kernelement den namensgebenden Service (Dienstleistung) hat. Wesentlich sind hierbei die vier Merkmale Verteilung, lose Kopplung, Standards und Prozessorientierung. **Verteilung** ist ein notwendiges Kriterium einer Architektur in einer immer stärker vernetzten Welt. Im Falle von SNS ist das die verteilte Nutzung von zentral installierten Thesaurus-Diensten. Eine **lose Kopplung** verringert das Anpassungsrisiko der Dienstnutzer bei einer sich immer schneller ändernden Umgebung, im Gegensatz zu den „fest verdrahteten“ Aufrufen älterer Applikationen. Allgemeingültige und freie **Standards** wie z.B. SOAP oder WSDL (s. Kap. 4.4.3) besitzen den Vorteil, dass man nicht von einem einzelnen Anbieter abhängig ist. **Prozessorientierung** sorgt für eine nicht zu große Komplexität des Services, da der Fokus primär auf den elementaren Leistungen liegt.¹¹

Ein ähnlicher Ansatz mit fast identischen Zielen kam schon zu Beginn der 90er Jahre mit dem sogenannten DCE (Distributed Computing Environment) Industriestandard auf. Allerdings wurde die Idee, eine einheitliche API auf allen Rechnern und Betriebssystemen einzusetzen, nie verwirklicht.¹²

Wie in der unteren Abbildung zu sehen ist, stehen Web Services momentan am Ende einer bereits vierzigjährigen Geschichte des „distributed computing“.

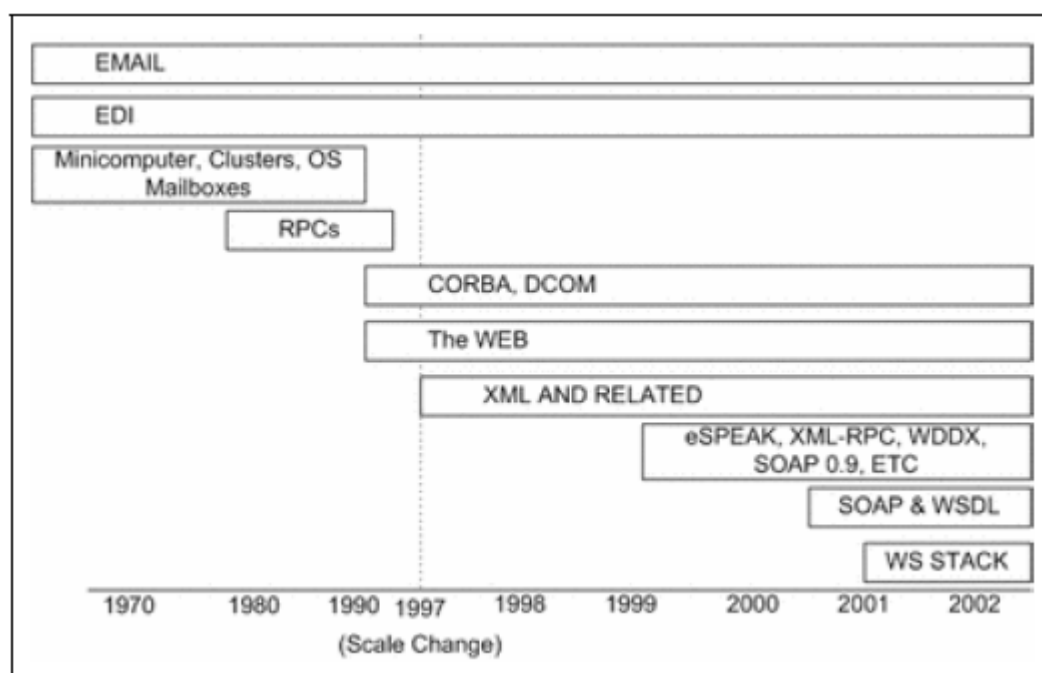


Abbildung 3: Geschichte des "distributed computing"¹³

¹¹ Vgl. Dostal, W. & Jeckle, M. (2004).

¹² Vgl. Knuth, M. (2003), S.17.

¹³ Ogbujo, U. (2002).

Als Vorläufer des World Wide Web und den Web Services kam Ende der 60er Jahre als erstes elektronisches Austauschformat von Botschaften die E-Mail auf, die den Vorgänger des Internets, das ARPANET¹⁴, als Plattform nutzten. Für die ARPANET-Initiatoren war dies eine Überraschung, denn noch im Jahre 1967 sagte Lawrence Roberts, einer der ARPANET-Entwickler, ein Programm für den Austausch von Nachrichten unter den Netzwerkteilnehmern sei „*not an important motivation for a network of scientific computers*“¹⁵. Aber gerade dieser Standard bewirkte eine viel stärkere Nutzung des ARPANETS und begünstigte die Entwicklung des Internets.

Die in den 90er Jahren entwickelte Common Object Request Broker Architecture, kurz CORBA genannt, hatte als objektorientierte Middleware schon viel mit dem Grundgedanken von Web Services gemein, den sie vereinfachte das Erstellen von verteilten Anwendungen in heterogenen Umgebungen und ist außerdem nicht an eine bestimmte Programmiersprache gebunden. Ebenso wie bei Web Services sollten sich hierbei verschiedene Softwaresysteme austauschen können. Allerdings hatte CORBA den entscheidenden Nachteil, dass man sich bei der Interoperabilität auf die Basis eines gemeinsamen Protokolls konzentriert hatte, was sich erst später als Fehler herausgestellt hat.¹⁶ Zudem benötigt Corba einen offenen Port, der aus Sicherheitsgründen auf der Gegenseite geschlossen sein kann.¹⁷ Ähnliches gilt auch für RMI, eine javabasierte Middleware-Lösung mit ähnlicher Intention.

Web Services umgehen dieses Manko, indem sie auf allgemeingültige Standards wie das allgegenwärtigen HTTP-Internetprotokoll setzen und so den von Firewalls üblicherweise freigeschalteten Port 80 verwenden können.

Das Zwiebelschalenmodell¹⁸ von M. Jeckle (s. Abb. unten) veranschaulicht den Aufbau von Web Services. Hierbei wird der Web Service zunächst als klassische Dienstleistung definiert, die über das Internet erbracht wird. Als Standards werden hierbei neben der physischen Hardware die allgemeingültigen Transportprotokolle des Webs verwendet, wie z.B. HTTP, FTP oder SMTP. In der nächsten Stufe ist dann in der Regel XML als Vermittlungs- und Metasprache anzutreffen, um unter anderem die korrekte Darstellung des Dienstleistungsinhaltes zu gewährleisten.

¹⁴ Das ARPANET (Advanced Research Projects Agency Network) wurde ursprünglich im Auftrag der US-Luftwaffe ab 1962 von einer kleinen Forschergruppe unter der Leitung des Massachusetts Institute of Technology und des US-Verteidigungsministeriums entwickelt [vgl. computerhistory.org (2002)].

¹⁵ Peter, I. (2004).

¹⁶ Vgl. Knuth, M. (2003), S.17.

¹⁷ Ullenboom, C. (2005).

¹⁸ Jeckle, M. (2004).

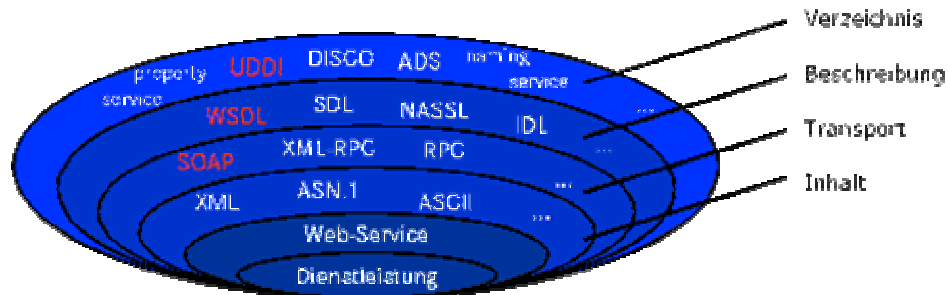


Abbildung 4: Web Services

Die nächste Schicht steht für den Kommunikationsaspekt und beinhaltet verschiedene potentielle Träger der Botschaft, wie z.B. SOAP oder HTTP-GET/POST-Bindings (eine detaillierte Beschreibung s. Kap. 4.4). Um die Natur der angebotenen Leistung dem Konsumenten näher zu bringen, ist eine Beschreibung des Services sinnvoll. Diesen Part übernimmt meist die Web Service Description Language (WSDL), die als frei zugängliche Datei auf dem Server des Services liegt, in der potentielle Nutzer die Abwicklungsmodalitäten wie etwa Abrufkonventionen oder zu übergebende Parameter einsehen können. Die äußerste Schale, die Verzeichnisschicht, stellt sicher, dass der angebotene Web Service gefunden werden kann, z.B. mittels Universal Description, Discovery and Integration (UDDI), einem zentralen Verzeichnisdienst, bzw. „Telefonbuch“ oder „Anbieterkatalog“ von Web Services.¹⁹

Der oben aufgezeigte Aufbau eines Web Service wurde vom World Wide Web Consortium (W3C) in der W3C Web Service Architecture²⁰ festgelegt und diente auch als Leitfaden bei der Erstellung des Semantic Network Service des Umweltbundesamtes.

In der Praxis finden Web Services zunehmend Verbreitung, wie z.B. bei dem Online-shop Amazon oder bei der Suchmaschine Google.²¹

Zusammenfassend betrachtet, können mit Hilfe von Web Services Funktionen auf einem fremden Rechner abgerufen werden, um z.B. auf dessen Wissensbasis zugreifen zu können, wie z.B. im Falle der vorliegenden Arbeit auf den seit über 20 Jahren gepflegten Umweltthesaurus UmThes®²² des Umweltbundesamtes.

¹⁹ Vgl. Jeckle, M. (2004).

²⁰ W3C (2002).

²¹ Vgl. Bandholtz, T. (2003), S. 39.

²² Umweltbundesamt.de (2005).

4.2 Semantische Netze und Semantic Web

“A semantic network or net is a graphic notation for representing knowledge in patterns of interconnected nodes and arcs. Computer implementations of semantic networks were first developed for artificial intelligence and machine translation, but earlier versions have long been used in philosophy, psychology, and linguistics.”²³

“The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.”²⁴

Die Begriffe „Semantisches Netz“ und „Semantic Web“ werden häufig synonym²⁵ verwendet und sind tatsächlich inhaltlich sehr stark miteinander verwandt. Während das Modell eines semantischen Netzes bedeutend älter ist und seinen Ursprung in der kognitiven Psychologie²⁶ hat, wurde der Begriff Semantic Web maßgeblich von dem WWW-Gründer Tim Berners-Lee in den Anfängen des Internets der 90er Jahre geprägt und sieht demnach eine Übertragung der Prinzipien eines semantischen Netzes auf das World Wide Web und andere Computernetzwerke vor.

Semantischen Netze, auch „Wissensnetze“ genannt, wurden bereits um 1900 von Psychologen entwickelt, als festgestellt wurde, dass im menschlichen Gehirn Begriffe miteinander verknüpft sind. Der Begriff „Semantisches Netz“ wurde zu dieser Zeit aber noch nicht benutzt, sondern maßgeblich von Ross Quillian geprägt, der das erste semantische Netz im Jahre 1967 vorgestellt hat. Seine Präsentation kann als ein Meilenstein in der Geschichte der künstlichen Intelligenz gewertet werden.²⁷

Lexikalisch-semantisch betrachtet stellen semantische Netze eine Weiterentwicklung von Taxonomien²⁸ und Thesauri²⁹ dar, da sie neben der durch Thesauri erzeugten Dar-

²³ Sowa, J.F. (2002).

²⁴ W3C (2001).

²⁵ Zwei Begriffe sind synonym, wenn sie verschiedene Bezeichnungen und dieselbe Definition aufweisen. Beispiel: Heirat und Eheschließung.

²⁶ Die kognitive Psychologie ist ein Teilgebiet der Psychologie und beschäftigt sich mit der Kognition, also der Informationsverarbeitung des Gehirns hinsichtlich Wahrnehmung, Gedächtnis, Aufmerksamkeit und Denken [vgl. Engineering Psychology Institute (2005)].

²⁷ Vgl. American Association for Artificial Intelligence (2002).

²⁸ Eine Taxonomie ist eine Hierarchie von Begriffen. Sie enthält Über- und Unterordnungsbeziehungen und kann auf diese Weise Vererbungsrelationen abbilden. Allerdings kann sie im Gegensatz zu Thesauri keine Beschreibung von Begriffen abbilden. Taxonomien sind in der Biologie weit verbreitet, wo sie zur Klassifikation von Pflanzen und Tieren genutzt werden. Eine weitere verbreitete Anwendung, die Taxonomien verwendet, ist das Dateisystem von Windows mit seiner Ordnerstruktur [vgl. Schmaltz, R. (2004), S.3; Ullrich, M./Maier, A./Angele, J. (2003)].

²⁹ Ein Thesaurus (altgriechisch. „thesauros“: Schatzhaus) ist ein systematisch geordnetes Verzeichnis von Wörtern. Er enthält eine möglichst vollständige Terminologie eines Fachgebietes inklusive Homonym-, Synonym- und Äquivalenzbeziehungen. Thesauri können ebenfalls

stellung von Synonymen und über- oder untergeordneten Begriffen ebenfalls Beziehungen (Relationen, Assoziationen) zwischen Begriffen aufzeigen können.

Man kann sich ein semantisches Netz als einen gerichteten Graph aus einer Menge von Knoten (die Begriffe repräsentieren) und einer Menge von gerichteten Kanten (die Beziehungen zwischen den Begriffen darstellen) vorstellen.³⁰ Die folgende Abbildung veranschaulicht dieses formale Modell von Begriffen und ihren Beziehungen.

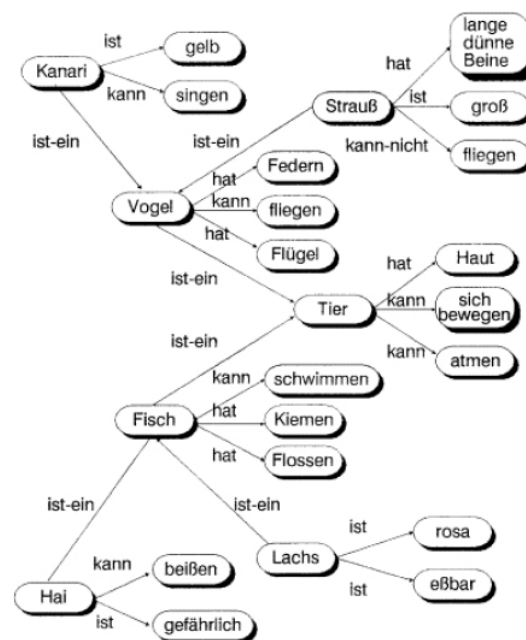


Abbildung 5: Darstellung eines semantischen Netzes³¹

Hervorzuheben sind hierbei folgende, meist binäre Relationen (Beziehungen) zwischen zwei Graphenknoten (Begriffen) unter weiteren lexikalisch-semantischen Relationen, die Roger Chaffin definiert hat.³²

- Vererbungsrelation: Ein „Kanari“ ist ein Unterbegriff von „Vogel“, welcher wiederum ein Unterbegriff von „Tier“ ist.
- Partitive Relation: „Federn“ sind ein Teil von „Vogel“ und ein „Fisch“ hat „Flossen“.
- Synonymie: „Heirat“ und „Eheschließung“ haben die gleiche Bedeutung.

hierarchische Relationen enthalten (vgl. Taxonomien). Eine bekannte Anwendung ist der in Microsoft Word integrierte Thesaurus für verschiedene Sprachen [vgl. Schmaltz, R. (2004), S.3; Ullrich, M./Maier, A./Angele, J. (2003)].

³⁰ Vgl. Sowa, J.F. (2002).

³¹ Nohr, H. (2005).

³² Chaffin, R. (1992).

Eine praktische Anwendung finden semantische Netze z.B. bei Clusteranalysen³³ im Bereich des informatrischen Data Mining³⁴, da sie zur visuellen Darstellung der aufgespürten Cluster geeignet sind.³⁵

Sie sind geradezu dafür prädestiniert, mit einer Software dynamisch erzeugt und dargestellt zu werden (s. untere Abb.).

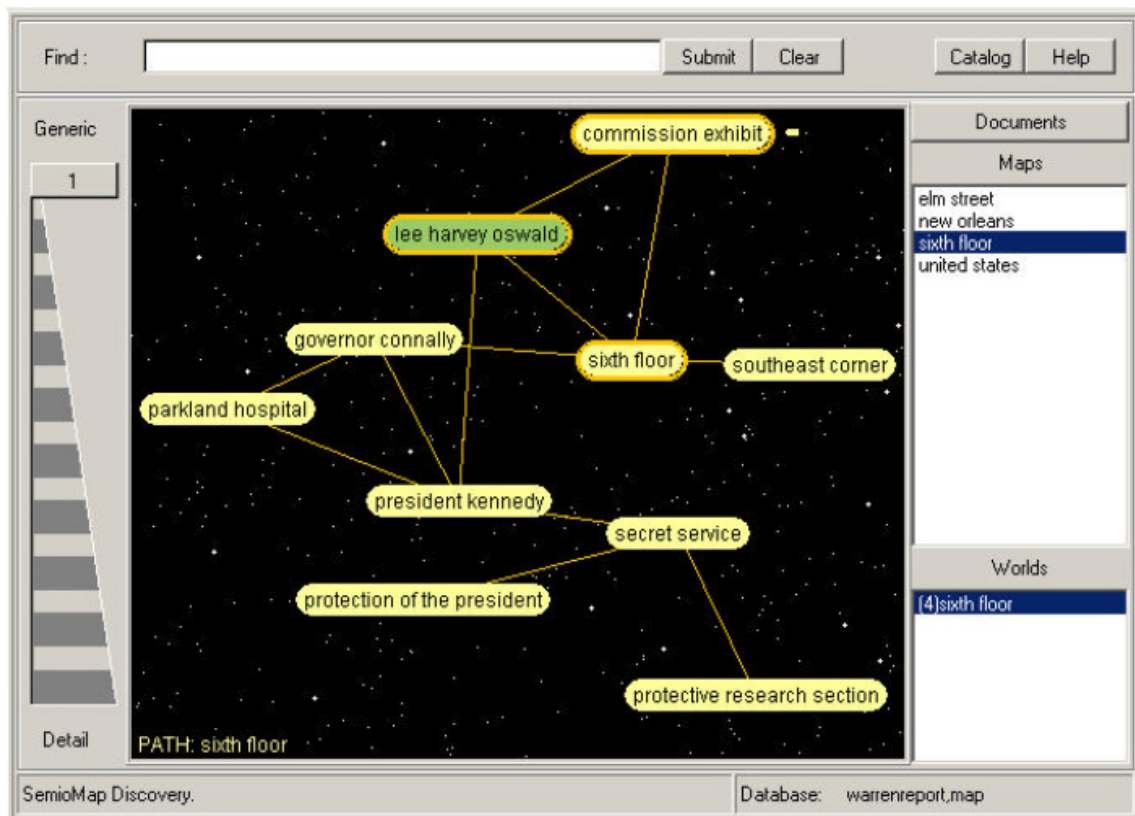


Abbildung 6: Semantisches Netz³⁶

Das **Semantic Web** hat seinen Ursprung im Bereich der künstlichen Intelligenz und der Wissensrepräsentation, welches die Informationen sowohl für Menschen als auch für Maschinen verständlich machen soll.³⁷

Hierbei sind die Erkenntnisse aus dem Bereich der kognitiven Psychologie entscheidend, „da bei der Beziehung zwischen Mensch und Computer Informationen ausgetauscht werden [...]“³⁸

³³ Die Clusteranalyse ist ein Verfahren der Datenanalyse für die Zusammenfassung von Gruppen (Cluster) ähnlicher Objekte aus einer Grundmenge von Objekten [vgl. Theus, M. (2004)].

³⁴ Unter Data Mining versteht man die Extraktion von implizit vorhandenem, nicht trivialen und nützlichen Wissens aus großen Datenbeständen [vgl. Bissantz, N. & Hagedorn, J. (1993)].

³⁵ Vgl. Stock, W.G. (2000), S.141.

³⁶ Entrieva Inc. (2006).

³⁷ Vgl. Maedche, A. & Motik, B. (2003).

³⁸ Vgl. Engineering Psychology Institute (2005).

Laut der richtungsweisenden Definition des Semantic Web von Tim Berners-Lee mit dem Wortlaut

*“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people work in cooperation”*³⁹

hat ein Semantic Web drei wesentliche Aufgaben:

- Das bereits bestehende präsentrationsorientierte Web soll in seiner jetzigen Form nicht abgelöst, sondern um semantische Anteile ergänzt werden.
- Die Bedeutung von Inhalten wird nach wie vor vom Menschen erfasst und nicht durch die künstliche Intelligenz von Maschinen.
- Primäres Ziel eines Semantic Web ist den Menschen bei der Nutzung des Webs zu unterstützen.⁴⁰

Dieses Konzept bzw. diese Vision eines Semantic Web erstreckt sich nicht nur über wissenschaftliche Forschungsbereiche, z.B. im Bereich der künstlichen Intelligenz, sondern soll in Zukunft eine breite öffentliche Akzeptanz bei „normalen“ WWW-Usern erhalten, indem sich das WWW zu einem großen Semantic Web weiterentwickelt.

Hierfür soll laut Tim Berners-Lee die vom World Wide Web Consortium (W3C) erarbeitete und standardisierte Auszeichnungssprache für Metadaten namens Resource Description Framework (RDF) zum Einsatz kommen. RDF wird sehr häufig zusammen mit dem Semantic Web genannt (vgl. Abb. 5). Wichtig ist zu wissen, dass keine Technik oder Spezifikation bei der Modellierung eines Semantic Web generell ausgeschlossen wird⁴¹. Die hier vorgestellte Arbeit zeigt auf, dass die Vision von einem Semantic Web zur Wissensrepräsentation gleichfalls mit dem vom W3C erarbeiteten XML Topic Map - Standard (s. Kap.4.3) umgesetzt werden kann. Der SNS-Entwickler Thomas Bandholtz bemerkt hierzu: *„Das Resource Description Framework (RDF) bietet zwar eine allgemeine Methode, um Ressourcen (Informationen) mit Metadaten zu verknüpfen, Topic Maps sind dagegen ein Behälter, um diese Metadaten selbst zu strukturieren.“*⁴²

Technisch betrachtet, stellen Semantic Webs einen Bestandteil der Service-orientierten Architektur (SOA) dar, neben dem ebenso wichtigen Bestandteil der Web Services (s. Kap. 4.1). Die untere Yin-Yang-Abbildung von Wolfgang Dostal bringt diese Symbiose zum Ausdruck.

³⁹ Berners-Lee, T. / Hendler, J. / Lassila, O. (2001).

⁴⁰ Vgl. Dostal, W. / Jeckle, M. / Meizer, I. / Zengler, Barbara (2004).

⁴¹ Vgl. W3C (2001).

⁴² Bandholtz, T. (2003), S.17.



Abbildung 7: Semantische Web Services als Symbiose von Semantik (Yin) und Syntax (Yang)⁴³

Kernaufgabe eines Semantic Web ist eine deutliche Vereinfachung der automatischen Datenbeschaffung aus verteilten Quellen, um somit automatische Integration von Daten und Anwendungen über Computernetzwerke zu ermöglichen. Dies wird durch zusätzliche strukturierte und von Maschinen lesbare Metadaten von Ressourcen ermöglicht.⁴⁴

4.3 Topic Maps

“Topic maps are a new ISO standard for describing knowledge structures and associating them with information resources. As such they constitute an enabling technology for knowledge management. Dubbed “the GPS of the information universe”, topic maps are also destined to provide powerful new ways of navigating large and interconnected corpora.”⁴⁵

Topic Maps stellen eine Variante von semantischen Netzen dar⁴⁶ und können als eine Art „Wissens-Landkarte“ angesehen werden, in welcher einzelne Themen (topics, wie z.B. Deskriptoren, geographische Namen, Ereignisse) durch Assoziationen miteinander vernetzt sind.⁴⁷

Sie repräsentieren somit einen inhaltlichen Index von Dokumenten, wobei die einzelnen Themengebiete (Topics) untereinander verbunden sind. Auf diesen Pfaden, bzw. Verbindungen von Topics kann in einer Topic Map navigiert werden. Dadurch ist es möglich, Themengebiete zu finden, ohne entsprechende fachspezifische Ausdrücke vorher kennen zu müssen (im Gegensatz zur Volltextsuche bei Suchmaschinen).⁴⁸

⁴³ Vgl. Dostal, W. & Jeckle, M. (2004).

⁴⁴ Vgl. Schmaltz, R. (2004), S.3.

⁴⁵ Pepper, S. (2000).

⁴⁶ Vgl. Schmaltz, R. (2004), S.5.

⁴⁷ Bandholtz, T. (2003), S.16.

⁴⁸ Vgl. Schmaltz, R. (2004), S.6.

4.3.1 Der ISO-Standard 13250

Ausgehend von der Entwicklung von Topic Maps der Davenport Gruppe in den frühen 90er Jahren⁴⁹ wurden die technischen und konzeptionellen Aspekte von Topic Maps von der ISO im Jahre 1999 standardisiert (ISO 13250), zunächst aufbauend auf der Terminologie von SGML⁵⁰ (Standard Generalized Markup Language).

Im folgenden Jahr wurde XML Topic Maps (XTM) als eine Erweiterung des ISO Standards 13250 veröffentlicht.

Die Hauptanwendungsgebiete von Topic Maps sind gemäß ISO 13250:

- Inhalte von Informationsobjekten als Topics darzustellen und zu vernetzen, um Navigationstools bereitzustellen.
- Die Erstellung von „virtuellen“ Informationsobjekten durch Topics der Topic Map.
- Das Filtern von Dokumentenbeständen in Bezug auf benutzerspezifische Ansichten, Mehrsprachigkeit, etc.
- Die Strukturierung von zumeist unstrukturierten Informationsobjekten, ohne die ursprünglichen Form zu verändern.⁵¹

4.3.2 Die Basiskonzepte von Topic Maps

Die drei Basiskonzepte einer Topic Map heißen *Topic*, *Occurrence* und *Association*⁵² (s. Abb. unten).

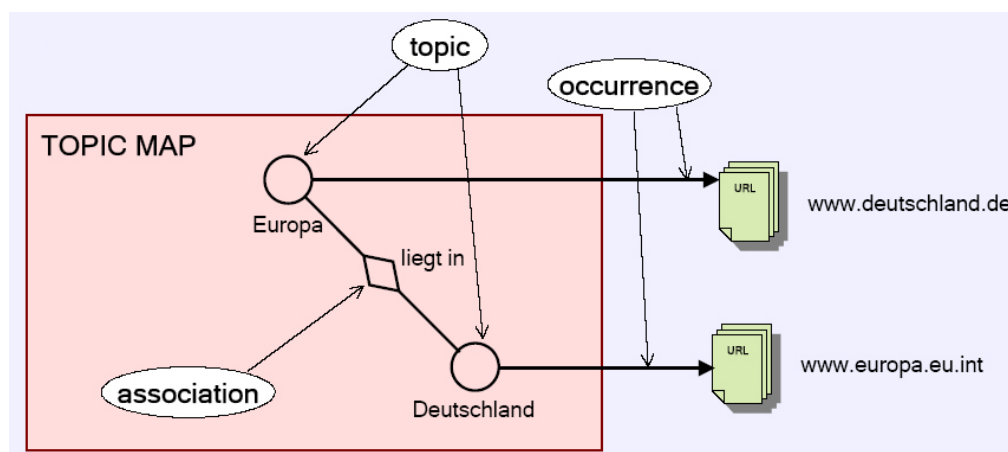


Abbildung 8: Aufbau einer Topic Map⁵³

⁴⁹ Vgl. Pepper, S. (2000).

⁵⁰ SGML ist eine komplexe Metasprache, die 1986 standardisiert wurde. Mit ihrer Hilfe können Auszeichnungs- oder Markup-Sprachen für Dokumente beschrieben werden [vgl. Hofmann, T. & Raithelhuber, U. (1998)].

⁵¹ Vgl. Schär, S. (2002).

⁵² Vgl. topicmaps.org (2001).

Topics bilden den zentralen Bestandteil der ISO 13250 und stehen für eine Art Ablage, die mit Inhalt gefüllt wird. Ein Topic kann ein beliebiges Objekt oder „Ding“ sein, das eine Bedeutung hat, wie z.B. Gegenstände, Personen, Orte jedoch ebenso Tätigkeiten, Relationen und abstrakte Begriffe.⁵⁴

Beispiele für Topics: „Stuttgart“, „Landschaft“, „Störfall“, „fließen“, „größer als“.

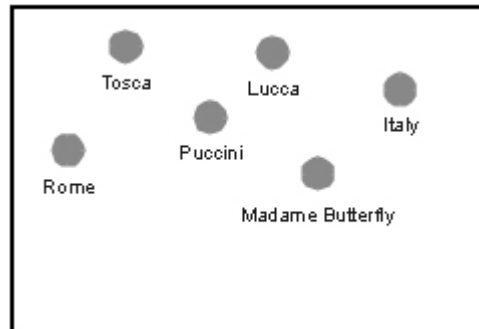


Abbildung 9: Topics⁵⁵

Um einen Topic genauer beschreiben zu können gibt es **Topic Types**.⁵⁶ Dieser Typ beschreibt, von welcher Art ein Topic ist und drückt somit eine „ist ein“-Relation aus.

Beispiele für Topic types: „Mauersegler“ ist vom Typ „Vogel“ oder „Zugspitze“ ist vom Typ „Berg“.

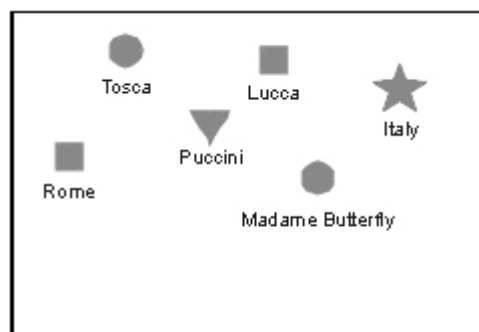


Abbildung 10: Topic types⁵⁷

Desweiteren gehört zu einem Topic mindestens ein Name, der **Topic name**.

Occurrences sind Verknüpfungen zu einer Informationsquelle, welche sich üblicherweise nicht in der Topic Map befindet. Bei der Zielressource der Verknüpfung kann es sich um beliebige Medien handeln, wie z.B. Videos, Musik, Präsentationen, gewöhnliche Texte, etc.⁵⁸

⁵³ Schär, S. (2002).

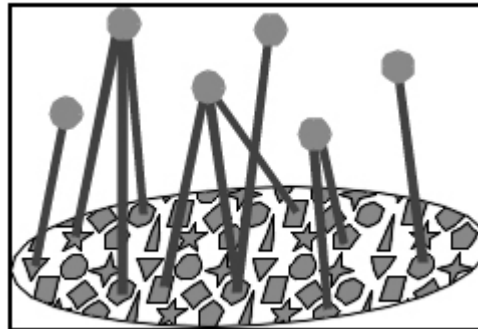
⁵⁴ Voegelé, T. (2002).

⁵⁵ Pepper, S. (2000).

⁵⁶ An dieser Stelle möchte der Autor daraufhin weisen, dass im Kapitel 7.5 dieser Arbeit eine Auflistung aller SNS-spezifischen Topic Typen enthalten ist.

⁵⁷ Pepper, S. (2000).

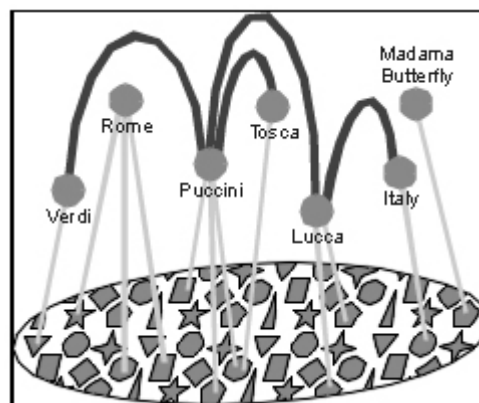
⁵⁸ Vgl. Pepper, S. (2000), S.11.

Abbildung 11: Occurrences⁵⁹

Bei den Occurrences gibt es ein optionales Typ-Element, um den Typ der Ressource näher zu beschreiben.

Mit den bisher vorgestellten Konzepten *Topic* und *Occurrences* von Topic Maps lassen sich zwar schon einfache Indexe erstellen und die Informationsressourcen der Topics organisieren, jedoch erfüllt es darüber hinaus keine weiteren Funktionen.⁶⁰

Das noch fehlende Schlüsselkonzept für die erweiterte, qualitative Funktionstüchtigkeit einer Topic Map sind die **Associations**, mit deren Hilfe die Beziehungen zwischen Topics modelliert werden können.

Abbildung 12: Topic associations⁶¹

Associations stellen Links zwischen zwei oder mehreren Topics dar. Ebenso wie bei den Konzepten *Topic* und *Occurrence* gibt es bei Associations einen Typ namens *Associations Type*, der die Art der Verbindung zwischen zwei Topics näher beschreibt, z.B. „Topic A liegt in Topic B“ oder „Topic A ist eine Zusammensetzung aus Topic B und Topic C“. Die Möglichkeit mit Hilfe von Associations verwandte Topics zusammenzuschließen, ist die Grundvoraussetzung für die Entwicklung anwendungsfreundlicher Navigationstools.⁶²

⁵⁹ Pepper, S. (2000).

⁶⁰ Pepper, S. (2000), S.12.

⁶¹ Pepper, S. (2000).

⁶² Vgl. Pepper (2000), S.13.

4.3.3 Elemente einer XML Topic Map

Die in diesem Kapitel vorgestellten Elemente einer Topic Map orientieren sich an dem XTM-Standard 1.0 der Arbeitsgruppe TopicMap.org⁶³, da die Topic Map von SNS auf diese Struktur und einige der in der unteren Liste aufgeführten Elemente zurückgreift. Das zu Grunde liegende XML Schema (xtn.xsd) von topicmaps.org (s. Anhang) wurde von SNS im Original belassen, bis auf das nachträgliche Einfügen eines Namensraums⁶⁴, der durch den XTM-Standard nicht vorgegeben ist, da dieser für die Serialisierung der Topic Maps entscheidend ist.

Tabelle 1: Komplette Auflistung aller XTM-Elemente⁶⁵

<topicRef>	Reference to a Topic element
<subjectIndicatorRef>	Reference to a Subject Indicator
<scope>	Reference to Topic(s) that comprise the Scope
<instanceOf>	Points to a Topic representing a class
<topicMap>	Topic Map document element
<topic>	Topic element
<subjectIdentity>	Subject reified by Topic
<baseName>	Base Name of a Topic
<baseNameString>	Base Name String container
<variant>	Alternate forms of Base Name
<variantName>	Container for Variant Name
<parameters>	Processing context for Variant
<association>	Topic Association
<member>	Member in Topic Association
<roleSpec>	Points to a Topic serving as an Association Role
<occurrence>	Resources regarded as an Occurrence
<resourceRef>	Reference to a Resource
<resourceData>	Container for Resource data
<mergeMap>	Merge with another Topic Map

⁶³ Topicmap.org (2001).

⁶⁴ Das Konzept der Namensräume dient dazu, die Namen einer Dokumentenklasse zusammenzufassen. Um möglichst eindeutige Namen zu erhalten, gibt man Namensräume in Form eines URI an [vgl. Erlenkötter, H. (2003), S.143].

⁶⁵ Vgl. topicmap.org (2001).

Zum besseren Verständnis werden die XTM-Elemente auch mit Hilfe der UML-Klassendiagramme der Arbeitsgruppe von topicmaps.org erklärt. Es gibt vier verschiedene Verbindungstypen zwischen Klassen und Instanzen von Klassen, die als spezielle Pfeile visualisiert worden sind, die an dieser Stelle kurz erläutert werden.

Tabelle 2: Verbindungstypen

—▷	Verbindung von einem Element zu seinem Subelement
—→	Eindeutige Zuordnung in eine Richtung. Ohne Pfeilkopf keine Richtungsvorgabe.
—◆	Strenges Abhängigkeitsverhältnis, bzw. Eigentum.
—◇	Anlage

Das Elternelement aller enthaltenen Topics in einer Topic Map heißt **<topicMap>**.

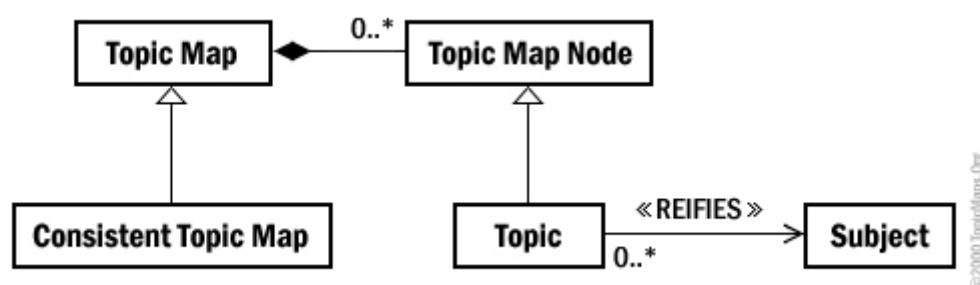


Abbildung 13: Topic Map (Klassendiagramm)⁶⁶

Es stellt das Dokumentenelement einer gültigen Topic Map dar und kann als Unter-elemente null oder mehr Topic Map Nodes (Topics, Assoziationen und andere Elemente) enthalten. Es ist mit einem eindeutigen Namensraum in Form einer URI versehen, im Falle der Topic Map von SNS mit dem Namensraum: „http://www.topicmaps.org/xtm/1.0/“ um somit kenntlich zu machen, dass die Topic Map den Spezifikationen von topicmaps.org unterliegt. Es ist für mehrere Topics innerhalb der Topic Map möglich, auf dieselbe Informationsressource (Subject) zuzugreifen.

Das Element **<topic>** umfasst den Namen und die Occurrences eines einzelnen Topics. Es muß immer mit einem eindeutigen ID-Attribut gekennzeichnet sein.

Das Kindelement **<instanceOf>** von **<topic>** ist optional und beschreibt den Topic Typ mit Hilfe des **<topicRef>** Elements und dessen xlink-Attribut mit einer eindeutigen URI des Topic Typs.

Das Element **<baseName>** schliesst das Unter-element **<baseNameString>** mit ein, wo der Name des Topics (als String) vermerkt ist.

⁶⁶ Topicmaps.org (2001).

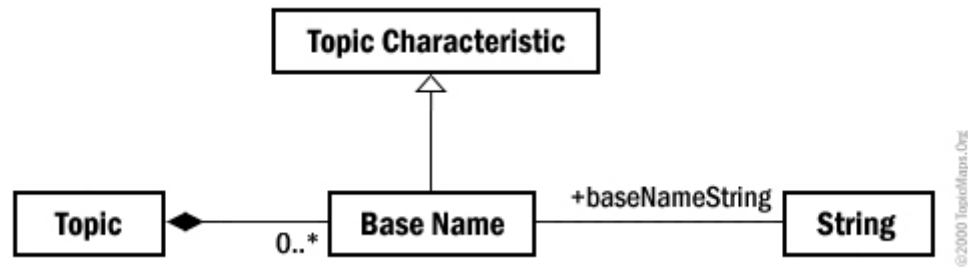


Abbildung 14: Base Name im Inneren eines Scopes (Klassendiagramm)⁶⁷

Das `<baseNameString>` Element kann von einem `<scope>` Element eingeschlossen sein, das z.B. die Sprache des Topic Namens kennzeichnet. Außerdem kann das Element `<baseName>` die Unterelemente `<variant>` mit den Unterelementen `<parameters>` und `<variantName>` für andere Bezeichnungen des Topics enthalten.

Zur Veranschaulichung folgt ein SNS-Topic Map-Auszug des Thesauruseintrags „Biene“:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<topicMapFragment xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="
http://www.semantic-network.de/xmlns/XTM/2005/2.0/">
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/">
<topic id="uba_thes_4911">
  <instanceOf><topicRef xlink:href="http://www.semantic-
network.de/xmlns/XTM/2005/2.0/sns-
classes_2.0.xtm#descriptorType"/></instanceOf>
  <baseName>
    <scope>
      <topicRef
xlink:href="http://www.topicmaps.org/xtm/1.0/language.xtm#de"/>
    </scope>
    <baseNameString>Biene</baseNameString>
  [...]
</baseName>
  <baseName>
    <scope>
      <topicRef
xlink:href="http://www.topicmaps.org/xtm/1.0/language.xtm#en"/>
    </scope>
    <baseNameString>bee</baseNameString>
    <variant>
      <parameters>
        <topicRef
xlink:href="http://www.topicmaps.org/xtm/1.0/core.xtm#variant"/>
      </parameters>
      <variantName>
        <resourceData>drone</resourceData>
      </variantName>
    </variant>
  </baseName>
</topic>
[...]
```

⁶⁷ Topicmaps.org (2001).

Bei einer genauen Betrachtung des obigen Topic Map-Auszug fällt das umschließende Elternelement `<topicMapFragment>` von `<topicMap>` auf. Was es damit auf sich hat, wird im Kap. 4.3. erläutert.

Occurrences sind ebenfalls Unterelemente des `<topic>`-Elements und werden mit dem Element **`<occurrence>`** ausgedrückt.

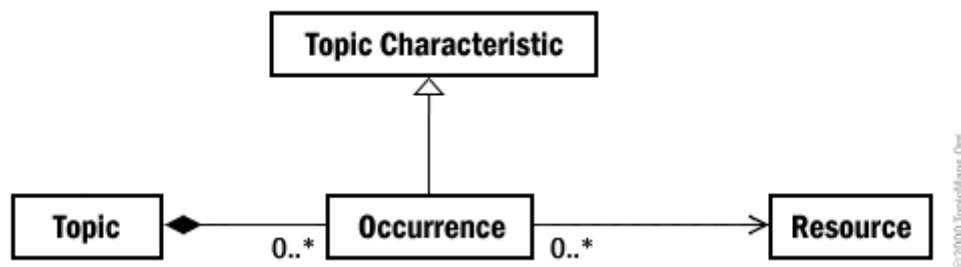


Abbildung 15: Occurrence (Klassendiagramm)⁶⁸

Als Unterelemente sind **`<resourceRef>`** (Verweis auf ein externes Dokument oder File) und **`<resourceData>`** (Beschreibung des Topics) definiert. Ein optionales `<scope>` Element kann hier erneut zum Einsatz kommen, um den Inhalt nach bestimmten Kriterien (z.B. Sprache) zu filtern. Das optionale `<instanceOf>`-Element spezifiziert die Klasse von der Occurrence eine Instanz bildet.

Das **`<association>`**-Element enthält die referenzierten Topics zu dem Basistopic.

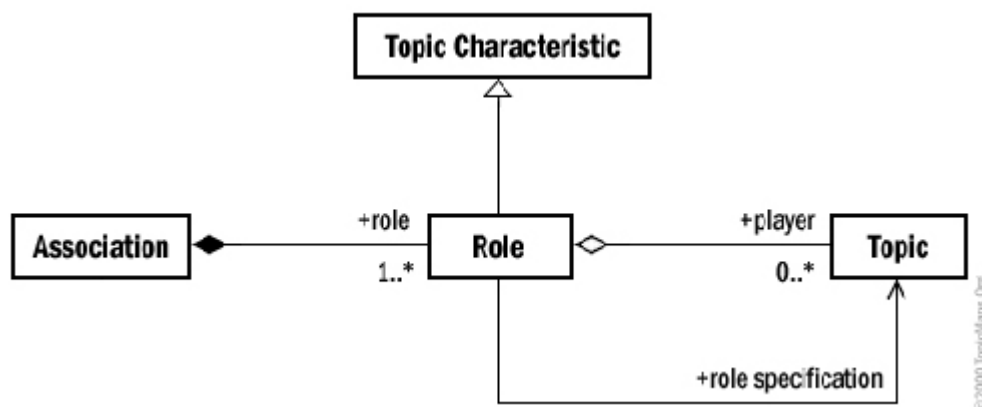


Abbildung 16: Assoziationen zwischen Topics (Klassendiagramm)⁶⁹

Hierbei wird mittels des Unterelements **`<member>`** der Typ der Verbindung innerhalb eines **`<roleSpec>`**-Elements und dem wiederum zugehörigen `<topicRef>`-Element

⁶⁸ Topicmaps.org (2001).

⁶⁹ Topicmaps.org (2001).

nebst xlink-Attribut verwendet. Es werden hintereinander beide miteinander verknüpften Topics mit Hilfe ihrer eindeutigen ID dargestellt. Beispielhaft dient hierfür ein Topic Map Auszug des SNS, bei dem die Topics „Biene“ und „Honig“ (beides Deskriptoren) miteinander assoziiert sind:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<topicMapFragment xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns="http://www.semantic-network.de/xmlns/XTM/2005/2.0/">
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/">
<topic id="uba_thes_4911">
[...]
  <baseNameString>Biene</baseNameString>
[...]
</topic>
<topic id="uba_thes_12883">
[...]
  <baseNameString>Honig</baseNameString>
[...]
</topic>
<association>
  <instanceOf>
    <topicRef xlink:href="http://www.semantic-
network.de/xmlns/XTM/2003/1.0/relatedTermsAssoc"/>
  </instanceOf>
  <member>
    <roleSpec><topicRef xlink:href="http://www.semantic-
network.de/xmlns/XTM/2005/2.0/sns-
classes_2.0.xtm#descriptorMember"></roleSpec>
    <topicRef xlink:href="uba_thes_4911"/>
  </member>
  <member>
    <roleSpec><topicRef xlink:href="http://www.semantic-
network.de/xmlns/XTM/2005/2.0/sns-
classes_2.0.xtm#descriptorMember"></roleSpec>
    <topicRef xlink:href="uba_thes_12883"/>
  </member>
</association>
</topicMap>
</topicMapFragment>
```

Der exemplarische und komplette Topic Map-Auszug des Thesauruseintrags „Biene“ von SNS befindet sich im Anhang dieser Arbeit. Die entsprechende Visualisierung dieser Topic Map wird in Kapitel 7.6.2 aufgezeigt.

4.4 Kommunikation mit dem Web Service

Wie im Kapitel 4.1 (Web Services) schon kurz angedeutet worden ist, gibt es verschiedene technische Möglichkeiten, um mit einem Web Service zu kommunizieren.

Um die von einem Web Service unterstützten Verbindungsmodalitäten und die zur Verfügung gestellten Dienste einzusehen, kann der Konsument des Dienstes einen Blick in das frei zugängliche WSDL-Dokument auf dem Server des Diensteanbieters werfen. Wie eine solche Datei aussieht und an welcher Stelle sich die interessanten Informationen bezüglich der Kommunikation mit dem Web Service und den bereitgestellten Diensten (im unteren Auszug fett hervorgehoben) verbergen, zeigt der folgende Auszug des WSDL-Dokuments vom Semantic Network Service des Umweltbundesamtes (Stand: Juni 2006), auch zu finden unter: <http://www.semantic-network.de/service-xtm-2.0/xtm/sns.wsd>. Das komplette Dokument befindet sich im Anhang dieser Arbeit.

```
<wsdl:definitions xmlns:sns="http://www.semantic-
network.de/xmlns/XTM/2005/2.0/"
[...]
targetNamespace="http://www.semantic-network.de/xmlns/XTM/2005/2.0/">
[...]
<!-- new webservice getSimilarTerms -->
  <wsdl:message name="getSimilarTermsInputMessage">
    <wsdl:part name="getSimilarTerms" element="sns:getSimilarTerms"/>
  </wsdl:message>
[...]
  <wsdl:message name="findEventsInputMessage">
    <wsdl:part name="findEvents" element="sns:findEvents"/>
  </wsdl:message>
[...]
  <wsdl:message name="anniversaryInputMessage">
    <wsdl:part name="anniversary" element="sns:anniversary"/>
  </wsdl:message>
[...]
  <wsdl:message name="findTopicsInputMessage">
    <wsdl:part name="findTopics" element="sns:findTopics"/>
  </wsdl:message>
[...]
  <wsdl:message name="getPSIInputMessage">
    <wsdl:part name="getPSI" element="sns:getPSI"/>
  </wsdl:message>
[...]
  <wsdl:message name="autoClassifyInputMessage">
    <wsdl:part name="autoClassify" element="sns:autoClassify"/>
  </wsdl:message>
[...]
  <wsdl:message name="getTypesInputMessage">
    <wsdl:part name="getTypes" element="sns:getTypes"/>
  </wsdl:message>
  <wsdl:message name="getTypesOutputMessage">
    <wsdl:part name="topicMap" element="sns:topicMapFragment"/>
  </wsdl:message>
  <wsdl:portType name="XTMEGetPortType">
[...]
  </wsdl:portType>
  <wsdl:binding name="HttpGetBinding" type="XTMEGetPortType">
    <http:binding verb="GET"/>
  </wsdl:binding>
[...]
```

```

</wsdl:binding>
<wsdl:binding name="HttpSoapBinding" type="XTMSoapPortType">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
[...]
```

```

</wsdl:binding>
<wsdl:service name="XTMservice">
  <wsdl:documentation>This Web Service permits Access to
SchlumbergerSema XTM</wsdl:documentation>
  <wsdl:port name="XTMGetPort" binding="HttpGetBinding">
    <!-- <http:address
location="http://213.144.28.228/service/xtm/command"/> -->
    <!-- <http:address location="http://uba-
web.imk.fraunhofer.de/service-xtm-2.0/xtm/command"/> -->
    <http:address location="http://www.semantic-network.de/service-
xtm-2.0/xtm/command"/>
    <!--http:address
location="http://localhost:8080/xtmwebbservice/xtm/command"/-->
    <!--http:address
location="http://roadrunner.sema.de/ubawebbservice/xtm/command"/-->
  </wsdl:port>
  <wsdl:port name="XTMSoapPort" binding="HttpSoapBinding">
    <!-- <soap:address
location="http://213.144.28.228/service/xtm/soap"/> -->
    <!-- <soap:address location="http://uba-
web.imk.fraunhofer.de/service-xtm-2.0/xtm/soap"/> -->
    <soap:address location="http://www.semantic-network.de/service-
xtm-2.0/xtm/soap"/>
    <!--soap:address
location="http://localhost:8080/xtmwebbservice/xtm/soap"/-->
    <!--soap:address
location="http://roadrunner.sema.de/ubawebbservice/xtm/soap"/-->
  </wsdl:port>
</wsdl:service>
<!-- ++++++ Type Definitions ++++++ -->
<!-- ++++++ Messages ++++++ -->
<!-- ++++++ PortTypes ++++++ -->
<!-- ++++++ Bindings ++++++ -->
<!-- GET/POST Binding -->
<!-- SOAP Binding -->
<!-- ++++++ Services ++++++ -->
</wsdl:definitions>

```

Wie zu sehen ist, zeigt die WSDL-Definition nicht nur die angebotenen Kommunikati-
onsschnittstellen (SOAP und GET/POST Bindings) auf, sondern ebenso die einzelnen
vom Service zur Verfügung gestellten Funktionsaufrufe (findTopics, getPSI, autoClassi-
fy, getTypes, getSimilarTerms, findEvents und anniversary.).

4.4.1 Funktionen des SNS Web Service

Mit dem Funktionsaufruf „**findTopics**“ wird in der Topic Map nach Topics mit den angegebenen Parametern gesucht. Die Rückgabe ist ein Topic Map-Fragment, welches die gefundenen Topics enthält, jedoch ohne Assoziationen. Die Topics erscheinen daher als flache Liste.⁷⁰

Der Name des Funktionsaufrufs „**getPSI**“ ist an die Published Subject Identifier (PSI) der Topic Map-Standardisierung (s. Kap. 4.3.1) angelehnt, welche vorschreibt, dass jedes Topic mit einer eindeutigen Webadresse (URI) definiert ist. Die HTTP-GET-Variante dieses Service implementiert für jedes in SNS enthaltene Topic diese eindeutige, gültige Webadresse, indem die ID des referenzierten Topics als Parameter übergeben wird. Die ID ist im Regelfall einer vorherigen Trefferliste des *findTopics*-Funktionsaufrufs entnommen. Unter Berücksichtigung der „topic characteristics“ (nach ISO 13250), wird nun eine vollständige Topic Map inklusive Assoziationen zurückgeliefert.⁷¹ Im SNS-Forschungsbericht vom Jahre 2003 wird noch auf den Parameter *distance* hingewiesen, mit dem gesteuert werden kann, wie tief die Assoziationen verfolgt werden können, um grafisch umgesetzt eine aus mehreren Ringen bestehende Topic Map darzustellen (s. untere Abb.). In der aktualisierten Anwenderdokumentation aus dem Jahre 2006 wird dieser Parameter als derzeit ungenutzt deklariert.⁷² Wie sich bei Versuchen des Autors bestätigte, konnte nur eine einfache grafische Anzeige mit einem Distance-Wert von 1 dargestellt werden.

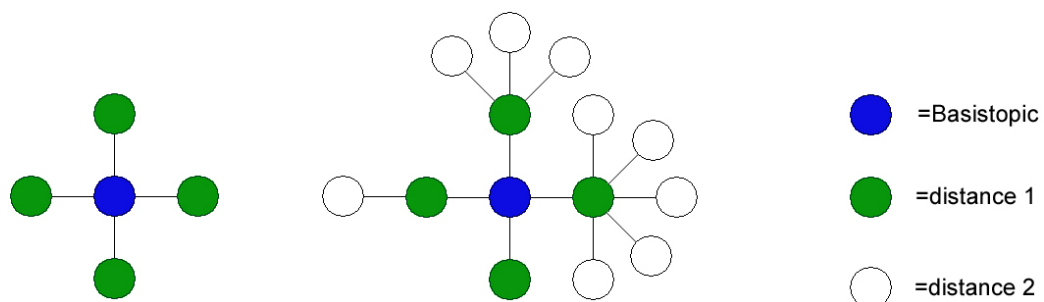


Abbildung 17: Distanztiefe der assoziierten Topics

Die Web Service Funktion namens „**Auto Classify**“ stellt die automatische Indexierung zur Verfügung. Der zu indexierende Text kann als *document* oder in Form einer *URL* übergeben werden. Die Rückgabe erfolgt wie bei dem „findTopics“-Service in Form einer flachen Liste von Topics, d.h. ohne Assoziationen.⁷³

⁷⁰ Vgl. Bandholtz, T. (2003), S.52.

⁷¹ Vgl. Bandholtz, T. (2003), S.53.

⁷² Vgl. Umweltbundesamt (2006), S.8.

⁷³ Vgl. Umweltbundesamt (2006), S.17.

Beim „**getTypes**“ – Request wird ein spezielles Fragment der SNS Topic Map übergeben, in welchem alle verwendeten Topic Typen und Assoziationsmuster beschrieben sind.⁷⁴

Mit der neuen Version 2.0 des SNS, die am 12.6.2006 veröffentlicht worden ist, werden noch drei weitere Web Service-Funktionen angeboten, mit den Bezeichnungen „getSimilarTerms“, „findEvents“ und „anniversary“.

Bei dem „**getSimilarTerms**“-Request werden syntaktisch oder semantisch ähnliche Begriffe zum Suchbegriff ausgegeben.⁷⁵

Mit der „**findEvents**“-Funktion können umweltbezogene Ereignisse aufgespürt werden, die in einem gewissen Zeitraum stattfanden, der von dem User vorher festgelegt worden ist.⁷⁶

„**anniversary**“ liefert Jahrestag-Umweltereignisse, ausgehend von einem zuvor festgelegten Datum.⁷⁷

4.4.2 Web Services und Protokolle

Die zwei wesentlichen Protokolle, die für die Kommunikation zwischen Web Services und dem Konsument zur Zeit eine Rolle spielen, sind das standardisierte XML-Protokoll SOAP und HTTP-GET/POST-Bindings auf der Basis von REST.

SOAP steht für Simple Object Access Protocol und stellt einen speziellen XML-Dialekt dar, der über das HTTP-Protokoll zwischen den Parteien übertragen wird. In der Spezifikation des W3C heißt es dazu: „*SOAP Version 1.2 is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment.*“⁷⁸

Bei SOAP wird der lokale Aufruf über einen Stub oder Proxy (s. Kap. 6.1.2) weiter über das Netz an die eigentliche Funktionsimplementierung weitergereicht, dort verarbeitet und abschließend werden die Ergebnisse an den Aufrufenden über den Stub/Proxy zurückgegeben.⁷⁹

Eine SOAP-Nachricht ist eine Art Briefumschlag oder Container (envelope), der ein optionales Header-Element und ein Body-Element enthalten kann. Im Head-Bereich befinden sich Metainformationen zur Nachricht und im Body-Bereich sind die eigentlichen Nutzdaten hinterlegt.⁸⁰ Ein SOAP-Request an den SNS Web Service könnte demnach wie folgt aussehen:

⁷⁴ Vgl. Umweltbundesamt (2006), S.7.

⁷⁵ Vgl. Umweltbundesamt (2006), S. 14.

⁷⁶ Vgl. Umweltbundesamt (2006), S. 12.

⁷⁷ Vgl. Umweltbundesamt (2006), S. 13.

⁷⁸ W3C (2003).

⁷⁹ Vgl. Knuth, M. (2003), S. 22.

⁸⁰ Vgl. Knuth, M. (2003), S.40.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <soapenv:Body>
    <sns:getPSI xmlns:sns="http://www.semantic-
network.de/service/xtm/soap">
      <getPSI>
        <user>xyz</user>
        <password>xyz</password>
        <id>uba_thes_4911</id>
      </getPSI>
    </sns:getPSI>
  </soapenv:Body>
</soapenv:Envelope>
```

REST ist eine Abkürzung für Representational State Transfer und steht für eine reine HTTP-basierte Lösung der Kommunikation mit dem Web Service, bei der die Übergabeparameter in einen HTTP-GET-Aufruf verpackt und als Resultat ein beliebiges XML-Dokument (z.B. eine Topic Map) zurückgeliefert wird⁸¹. Ein Beispiel für einen solchen Request, angelehnt an den Web Service von SNS, sähe dann so aus:

```
http://www.semantic-
network.de/service/xtm/command/getPSI?id=uba_thes_4911&user=x&password
=x
```

Als Resultat des oben aufgezeigten SOAP-Requests und HTTP-GET-Aufrufs käme vom Web Service dann eine Topic Map mit dem Basistopic bzw. Thesauruseintrag „Biene“ (s. Anhang) zurück.

4.4.3 SOAP vs. REST

Während Michael Knuth in seinem Buch „Web Services – Einführung und Übersicht“ die Möglichkeit mit HTTP-GET und HTTP-POST-Aufrufen zu arbeiten, als „offenes Hintertürchen“ beschreibt⁸², ist in Fachkreisen eine heftige Debatte über das Für und Wieder von SOAP und REST entbrannt.

REST erfüllt aus Sicht seiner Befürworter alle Kriterien eines Web Services, wie Verteilung, lose Kopplung und Plattformunabhängigkeit, da mittels einer HTTP-Botschaft alle Informationen übertragen werden können, die für eine verständliche Nachricht notwendig sind. Das wichtige Merkmal der Plattformunabhängigkeit bei Web Services spielt eine Schlüsselrolle bei den Verfechtern von REST. Paul Prescod äußert sich hierzu wie folgt: *“[...] ‘generality of interfaces’ is a key part of what makes REST advocates believe that it is a better basis for a Web Services framework than are the SOAP-based technologies. Any HTTP client can talk to any HTTP server with no further configuration. This is not true for SOAP. With SOAP, you must also know the method names,*

⁸¹ Vgl. Jeckle (2004).

⁸² Knuth, M. (2003), S.23.

*addressing model and procedural conventions of the particular service. This is because HTTP is an application protocol whereas SOAP is a protocol framework.*⁸³

SOAP-Befürworter sehen bei REST hingegen vor allem Schwächen in den Bereichen Sicherheit sowie Vertrauenswürdigkeit und beklagen eine fehlende Beschreibung des Dienstes, die bei einer SOAP-Lösung durch WSDL gewährleistet ist. Außerdem ist SOAP theoretisch nicht ausschließlich an das Transportprotokoll HTTP gebunden, sondern könnte auch auf der Basis von FTP oder SMTP eingesetzt werden.

Als Kompromiss der aufgezeigten Debatte unterstützen viele Web Services beide Konzepte, unter ihnen auch der SNS-Web Service.

4.5 Struktur der XML-Definitionen von SNS

*“We [the work group of SNS] found standardization quite developed in both the fields (Web Services and Topic Maps), but there was a gap about how to embed Topic Maps into a Web Service scenario. SNS had to close this gap to provide a working example, at state-of-the-art.”*⁸⁴

Wie die Auszüge der exemplarischen Topic Map und der WSDL-Datei von SNS in den vorhergehenden Kapiteln aufgezeigt haben, stehen diese beiden Komponenten in einer gewissen Beziehung zueinander. Der Grund dafür ist ein von der SNS-Arbeitsgruppe entwickeltes XML-Regelwerk, das sich aus mehreren XML Schema Definitionen zusammensetzt und die Lücke zwischen dem Topic Map-Standard und dem Web Service geschlossen hat.

In einer XML Schema Definition (XSD) werden die Regeln definiert, nach denen die Dokumente einer Klasse aufgebaut sein müssen. Das heißt, hier werden alle Elemente, die im referenzierten XML-Dokument vorkommen können, festgelegt. XSD unterscheidet hierbei zwischen einfachen und komplexen Typen. Einfache Typen enthalten nur Text, daher keine anderen Elemente. Elemente mit Kindelementen gehören zu den komplexen Typen und deswegen ist ein Dokumentenelement in der Regel immer ein komplexer Typ. Eine XML Schema Definition ist selbst ein XML-Dokument und wird in Dateien mit der Erweiterung .xsd gespeichert. Es ist der erklärte Nachfolger der Document Type Definition (DTD).⁸⁵

Die folgende Auflistung entstammt der englischen Dokumentation von SNS⁸⁶ und macht die Beziehung der einzelnen Komponenten deutlich:

SNS.WSDL – general definitions of services, bindings, and types (includes SNS.XSD)
SNS.XSD – definitions of specific request and response formats (includes XTM.XSD)
XTM.XSD – XML Topic Map interchange format (conforming ISO 13250).

⁸³ Prescod, P. (2002).

⁸⁴ SchlumbergerSema GmbH (2003).

⁸⁵ Vgl. Erlenkötter, H. (2003), S.140.

Die Funktionen des Web Service, welche in der WSDL-Datei aufgezeigt sind (s. Kap. 4.4.), werden als XML-Schema Definition im Definitionsbereich der Datentypen (*types*), die zum Austausch der *messages* benutzt werden, importiert, wie der folgende sns.wSDL-Auszug aufzeigt:

```
<wsdl:definitions xmlns:sns="http://www.semantic-
network.de/xmlns/XTM/2005/2.0/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://www.semantic-network.de/xmlns/XTM/2005/2.0/"
xmlns:ns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.semantic-network.de/xmlns/XTM/2005/2.0/">
  <wsdl:types>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <xs:import namespace="http://www.semantic-
network.de/xmlns/XTM/2005/2.0/" schemaLocation="sns.xsd"/>
    </xs:schema>
  </wsdl:types>
  [...]
</wsdl:definitions>
```

sns.xsd liefert demnach die Struktur der sechs Web Service-Funktionen findTopics, getPSI, autoClassify, findEvents, anniversary und getTopics. Nach einer Anfrage an den Web Service wird ein von sns.xsd definiertes Topic Map-Fragment zurückgeliefert, in dem sich ein passender Auszug der gesamten Topic Map von SNS mit dem durch die Suchparameter vorgegebenen Inhalt befindet. Die eigentliche Topic Map-Definition (xtn.xsd) wird wiederum von sns.xsd eingebettet, wie im folgenden Auszug von sns.xsd zu erkennen ist:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xtn="http://www.topicmaps.org/xtn/1.0/"
xmlns="http://www.semantic-network.de/xmlns/XTM/2005/2.0/"
xmlns:xlink="http://www.w3.org/1999/xlink"
elementFormDefault="qualified" targetNamespace="http://www.semantic-
network.de/xmlns/XTM/2005/2.0/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import schemaLocation="xtn.xsd"
namespace="http://www.topicmaps.org/xtn/1.0/" />
  <xs:element name="topicMapFragment">
    <xs:complexType>
      [...]
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Die standardisierte XML-Schema Definition **xtn.xsd** von topicmaps.org spezifiziert den Aufbau und die Elemente einer Topic Map (s. Kap. 4.3.3)

Im Anhang dieser Arbeit befinden sich alle wichtigen XML Dokumente von SNS in ausführlicher Form.

⁸⁶ SchlumbergerSema GmbH (2003).

4.6 Scalable Vector Graphics (SVG)

SVG ist ein XML-Standard zur Beschreibung zweidimensionaler Vektorgrafiken und neben den statischen Merkmalen, die auch schon von anderen Vektorgrafiken bekannt sind, sind es vor allem die dynamischen und im Web anwendbaren Eigenschaften von SVG, die es sehr interessant machen.⁸⁷ Für die in dieser Arbeit vorgestellte Visualisierung der Topic Map von SNS ist SVG hervorragend geeignet, da es ebenso wie der Topic Map-Standard auf XML basiert. Somit ist es mit einer XSL-Transformation möglich, die Topic Map in ein SVG Format umzuwandeln (s. Kap. 7.2.). Das folgende Beispiel zeigt einen SVG-Quelltext und im Anschluss die entsprechende Grafik.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
  "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg xmlns="http://www.w3.org/2000/svg" viewBox="-100 -100 200 200">
<title>Naturpark - Icon</title>

<circle x="0" y="0" r="50" style="fill: #04cc12; stroke: #203b54;
stroke-width: 6px" />

<path d="M -15,0 l -5,0 l -15,15 l 60,0 l 15,-15 l -55,0 M -15,10 l
0,-15 l -10,0 l 10,-20 l 10,20 l -10,0 M 10,5 l 0,-15 l -10,0 l 10,-20
l 10,20 l -10,0" style="fill: none; stroke: #ffffff; stroke-width:
6px"/>

<text font-size="29" style="fill: #ffffff;" text-anchor="middle" x="0"
y="42">P</text>
</svg>
```



Abbildung 18: Naturpark-Icon als SVG Grafik

Wie das obige Beispiel aufzeigt, werden bei SVG alle Bildeigenschaften als geometrische Formen (z.B. Linien, Rechtecke, Kreise, Ellipsen) in einem Koordinatensystem eingezeichnet. Diese Art der Bildbeschreibung führt zu einer großen Datenreduktion im Gegensatz zu Pixelformaten (z.B. JPEG, GIF, BMP) und hat außerdem den Vorteil einer beliebig skalierbaren Darstellung der Grafik („Zoomen“ ohne Qualitätsverlust).

⁸⁷ Bader, H. (2004), S. 9.

Zur großen Freude des Autors werden die Grundfunktionen von SVG von den aktuellsten Webbrowsern unterstützt, wie z.B. von dem momentan sehr populären Webbrowser Firefox von Mozilla. Dies erleichtert die Anzeige im Webbrowser, da kein zusätzliches Plug-In mehr benötigt wird. Das trifft aber noch nicht auf den Internet Explorer zu, der erst ab der bald erscheinenden Version 7 eine native Unterstützung von SVG bieten soll. Die bereits erschienene Beta-Version des IE7 konnte hingegen noch keine SVG-Grafik anzeigen.

4.7 Javascript und Ajax

Javascript ist eine funktionale und clientseitige Skriptsprache zur interaktiven Belegung von XML-Derivaten oder ähnlichen Auszeichnungssprachen wie HTML.⁸⁸ Mit den so genannten *Events* kann auf Benutzeraktionen wie z.B. Mausklicks oder Bewegungen in Echtzeit reagiert werden. Für die in dieser Arbeit vorgestellten grafische Navigationskomponente ist Javascript unerlässlich, da es ermöglicht, in Echtzeit auf gewisse Elemente innerhalb des Dokuments zuzugreifen und diese auszuwerten, bzw. darzustellen (z.B. die Occurrences von Topics). Das zur Zeit sehr prominente Ajax-Konzept, das auf Javascript basiert, wurde ebenfalls vom Autor verwendet (s. Kap 7.4.).

Ajax steht für *Asynchronous JavaScript + XML* und schafft für Webanwendungen die Möglichkeit, im Hintergrund eine HTTP-Anfrage an einen Server zu schicken und die Rückgabe auszuwerten. Der Grundpfeiler von Ajax ist das XMLHttpRequest-Objekt, das den Datenaustausch mit einem Webserver ohne ein Neuladen und damit verbundenen Neuaufbau der Seite ermöglicht.⁸⁹

4.8 PHP

PHP ist eine der beliebtesten, serverseitigen Skriptsprachen und wurde vom Autor für die Server-Server-Kommunikation zwischen der vorliegenden Webapplikation und dem SNS-Web Service und für die Transformation und Auswertung der vom Web Service zurückgelieferten XML Topic Maps eingesetzt.

Die Entscheidung für PHP wurde durch die in der neuen PHP-Version 5 bereitgestellten und verbesserten XML-Funktionen und die Möglichkeit der einfachen Erstellung eines Proxys (s. Kap. 6.1.2) begünstigt.

⁸⁸ Vgl. Bader, H. (2004), S.249.

⁸⁹ Vgl. Wenz, C. (2006), S.14.

5 Anforderungen und technische Lösungsansätze

Als Voraussetzung für die Konzeption der Webanwendung werden in diesem Kapitel zunächst alle Anforderungen in Bezug zum Nutzungskontext aufgezeigt. Hierfür sind die UML-Anwendungsfalldiagramme (Use Cases) vorzüglich geeignet, da sie völlig unabhängig von einer technischen Implementierung die Anforderungen aus Sicht des Nutzers an die Webapplikation aufzeigen.

In dem darauf folgenden Unterkapitel 5.2. werden die verschiedenen technischen Lösungsansätze aufgezeigt und diskutiert, die für eine erfolgreiche Umsetzung der gestellten Anforderungen an die Webapplikation in Frage kommen.

Der vom Autor gewählte Lösungsansatz wird im Kapitel 5.3. erläutert.

5.1 Use Cases

Die folgenden UML-Anwendungsfalldiagramme beruhen auf dem UML Standard 2.0.

Der Nutzer eines Systems wird demnach als ein einfaches Strichmännchen dargestellt, die Anwendungsfälle als Ellipse, das System als ein rechteckiges Kästchen und die Zusammenhänge zwischen Use Cases und Nutzer als einfache Verbindungslinien.

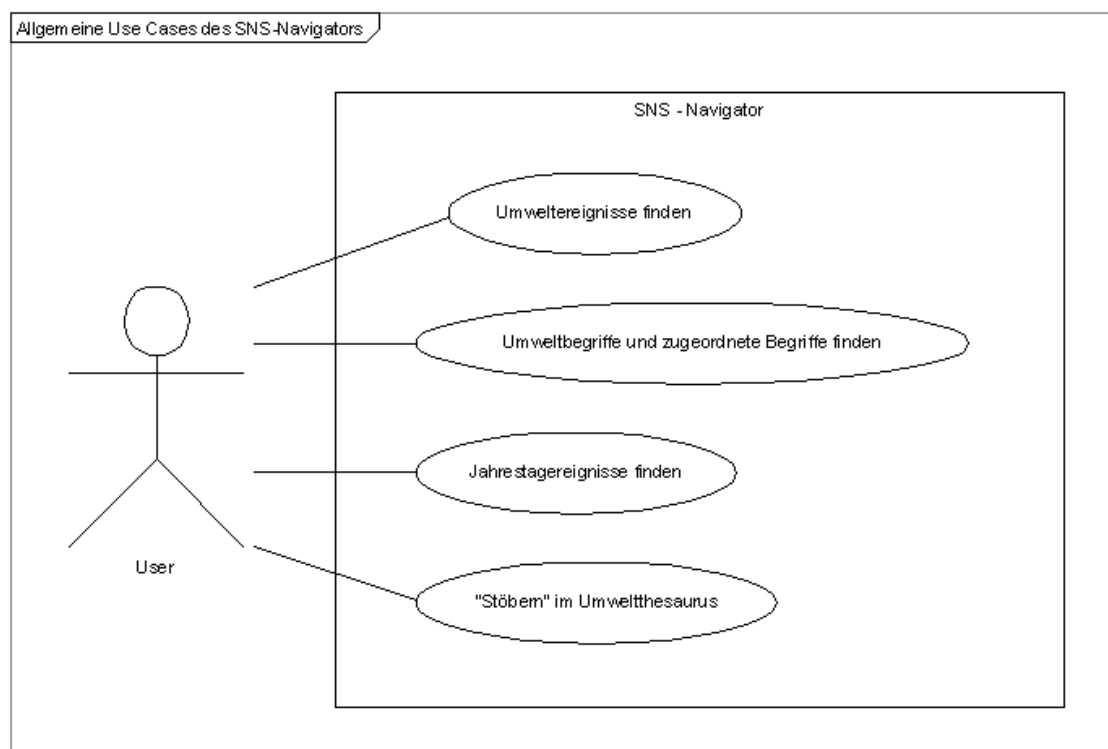


Abbildung 19: Allgemeine Use Cases des SNS-Navigators

Das obige Use-Case-Diagramm zeigt die wesentlichen Anwendungsfälle aus Sicht des Benutzers sowie den Informationsaustausch zwischen Benutzer und den Use Cases auf. Jedoch existiert ein weiteres wichtiges Use Case, auf dem der eigentliche Schwerpunkt dieser Arbeit liegt, die grafische Navigationskomponente, die in unterschiedlicher Weise an die bereits aufgezeigten Use Cases gekoppelt sind, wie in dem folgenden Diagramm zu sehen ist.

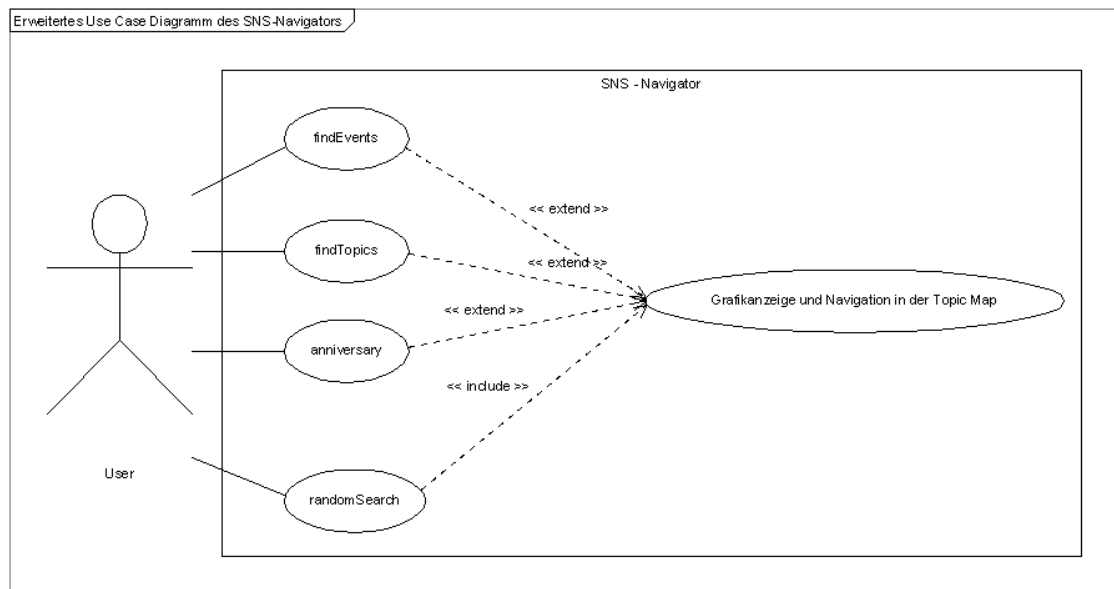


Abbildung 20: Erweitertes Use-Case-Diagramm des SNS-Navigators

In diesem erweiterten Anwendungsfalldiagramm hat der Autor die Namen der Use Cases an ihre eigentlichen Funktionsnamen angepasst, die SNS spezifiziert hat (s. Kap. 4.4.1.), mit Ausnahme von „randomSearch“, welcher eine Eigenentwicklung darstellt (s. Kap. 6.1.8). Der entscheidende Unterschied zu dem allgemeinen Anwendungsfalldiagramm liegt im Anwendungsfall „Grafikanzeige und Navigation in der Topic Map“. Dieser Use Case wird im Falle der Aktivierung von „findEvents“, „findTopics“ und „anniversary“ erst dann gestartet, wenn der Benutzer in der Suchergebnisliste explizit einen Topic auswählt, der durch eine eindeutige ID gekennzeichnet ist. Diese Art der Verbindung wird mit dem Schlüsselwort „extend“ ausgedrückt, womit kenntlich gemacht wird, dass der Use Case von dem die Verbindungslinie ausgeht, möglicherweise diesen Use Case erweitert, auf den die Verbindungslinie zeigt⁹⁰. Bei dem Use Case „randomSearch“ wird in jedem Fall nach dessen Ausführung die Grafikanzeige aktiviert, da das System einen zufälligen Topic aus dem Umweltthesaurus auswählt. Der Use Case „Grafikanzeige und Navigation in der Topic Map“ wird also immer dann ausgeführt, wenn explizit ein Topic vom Benutzer oder von dem System ausgewählt worden ist.

⁹⁰ Vgl. Schäling, B. (2005).

5.2 Empfehlungen des Umweltbundesamtes für die Einbindung des Services beim Nutzer

Die SNS-Arbeitsgruppe führte vor der Veröffentlichung ihres Netzwerkdienstes (s. Kap. 2) eine Reihe von Tests an bestehenden Plattformen durch, die auf die Kommunikation mit Web Services ausgelegt sind. Dabei stellten sich zwei technische Plattformen als tauglich heraus, die vom Umweltbundesamt empfohlen werden, um ihren Web Service anzusprechen.

5.2.1 Axis

Das Apache eXtensible Interaction System (Axis) ist eine Open-Source Implementierung des Web Service Standards SOAP (s. Kap. 4.4.2.) unter der Lizenz der Apache Software Foundation und „[...]hat sich zu einem wichtigen Tool und Framework für Web Services in der Java-Welt entwickelt“.⁹¹

Es wird in Form eines Web-Archivs angeboten und ist zur Installation in dem Servlet-Container Tomcat auf Webservern vorgesehen.

Es besteht aus einer Client- und Serverengine (s. untere Abb.) und ist somit sowohl für das Bereitstellen wie auch Abrufen eines Web Services geeignet.

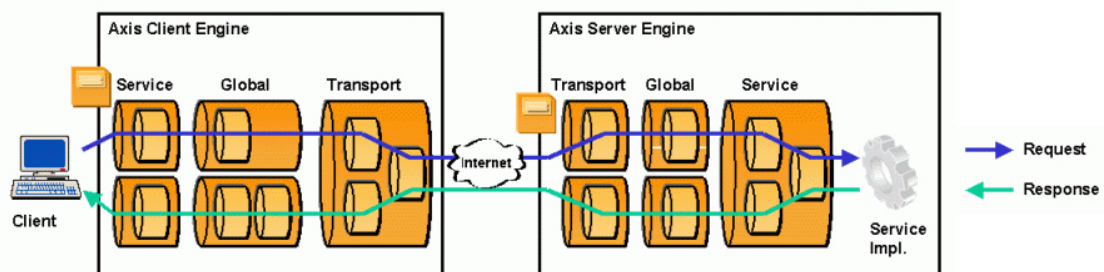


Abbildung 21: Axis Client- und Server Engine⁹²

Neben der Implementierung von SOAP unterstützt Axis ebenfalls die Web Service Description Language. (s. Kap. 4.4).

Mit Hilfe der WSDL werden Stubs (individuelle Java-Klassen) von diesem System generiert, welche den Zugriff auf den Web Service von Java aus erlauben, ohne dass die dazwischen stehende Kommunikation weiter beachtet werden muss. Allerdings stellte sich bei dem Test der SNS-Arbeitsgruppe heraus, dass die von Axis generierten Stubs

⁹¹ Bayer, T. (2003)

⁹² Bayer, T. (2003)

nicht korrekt funktionierten. Das hatte zur Folge, dass manuelle Bereinigungen der Stubs durchgeführt werden mussten.⁹³

Die Ursache für dieses Problem liegt in der mangelnden XML-Schema-Unterstützung (s. Kap. 4.5) von Axis, jedoch „[d]urch einige nachträgliche Anpassungen des SNS XML Schemas konnte dieses Problem [...] umgangen werden.“⁹⁴

5.2.2 .NET Framework

„.NET ist eine Plattform, welche die Entwicklung und Nutzung von XML-basierten Anwendungen, Prozessen und Websites als Dienste ermöglicht, die darauf ausgelegt sind, Informationen und Funktionen auf jeder Plattform und auf jedem Smart Device gemeinsam zu nutzen und zu kombinieren.“⁹⁵

Das .NET Framework enthält das sogenannte Web Service Development Kit, eine Klassenbibliothek von Microsoft.Net, die zur Erstellung von Webdiensten dient.⁹⁶

Das .NET Runtime Framework 1.0 arbeitete bei den Tests des Umweltbundesamt besser und war im Vergleich zu den anderen getesteten Plattformen die am weitesten fortgeschrittene Variante.⁹⁷

Ein Grund hierfür dürfte sein, dass der SOAP- und der WSDL-Standard verstärkt von dem Unternehmen Microsoft vorangetrieben worden ist, da es maßgeblich an der Entwicklung beider Standards beteiligt war.

5.3 Gewählter Lösungsansatz

Der Autor hat sich für die Entwicklung einer eigenen Webapplikation auf Basis von PHP (s. Kap. 4.8) entschieden, da Microsofts .NET-Plattform kostenpflichtig ist und das Open Source Projekt Axis zusammen mit dem SNS-Web Service bis dato nur mit den vom Umweltbundesamt manuell nachgearbeiteten Stubs komplett fehlerfrei arbeitet (s. Kap. 5.2.1.).

Aus Sicht des Autors hat PHP den großen Vorteil, dass es im Gegensatz zu den anderen beiden Plattformen auf fast jedem Server installiert ist, nicht so viele Ressourcen benötigt und außerdem über eine ausreichend große Anzahl von Funktionen sowie Routinen verfügt, um die Webapplikation mit den beschriebenen Anforderungen umzusetzen.

⁹³ Vgl. Bandholtz, T. (2003), S.47.

⁹⁴ Bandholtz, T. (2003), S.51.

⁹⁵ Pöhner G. & Renner, J. (2003), S.333.

⁹⁶ Vgl. Pöhner G. & Renner, J. (2003), S.352.

⁹⁷ Vgl. Bandholtz, T. (2003), S.50.

6 Konzeption

Die Webapplikation wird in einem Webbrowser angezeigt und besteht aus einem Frontend, das sich aus mehreren einzelnen Seiten zusammensetzt und einem für den Nutzer nicht sichtbaren Backend, in dem die serverseitigen Prozesse ablaufen und die Schnittstelle zum SNS-Web Service definiert ist.

Die gesamte Konzeption der Webapplikation baut auf zwei konzeptionellen Schwerpunkten auf, dem funktionalen und dem gestalterischen Aufbau des Systems.

Der funktionale Aufbau zeigt die Anwendungsarchitektur und die Applikationslogik auf.

Der gestalterische Aufbau beinhaltet die Oberflächengestaltung bzw. User-Schnittstelle zum System und hat somit einen entscheidenden Einfluss auf die Nutzerfreundlichkeit (Usability) der Anwendung und daher auf deren Erfolg beim Nutzerkreis.

6.1 Funktionaler Aufbau und Struktur der Webapplikation

Die funktionale Konzeption der Webanwendung erläutert zunächst die übergreifende Systemarchitektur, in der die Applikation eingebettet ist, die als höchste Instanz die Aufgaben und das Zusammenspiel der einzelnen Komponenten festlegt. Im nächsten Schritt werden die einzelnen Komponenten der Anwendung und die Kontrollflüsse und Schnittstellen innerhalb dieser Komponenten aufgezeigt.

6.1.1 Three-Tier-Architektur

Der SNS-Navigator ist Bestandteil einer Three-Tier-Architektur⁹⁸. Die erste Schicht ist der Webbrowser bzw. Client, der die Eingabe und Ausgabe der Daten über die Benutzungsoberfläche der Webapplikation steuert. Die zweite Schicht ist der SNS-Navigator, der als mittlere Schicht zwischen Webbrowser und dem Server des SNS als eine Art Broker fungiert. Der SNS-Server ist die dritte Schicht, der die Datenverwaltung und -haltung betreibt sowie den Web Service zur Verfügung stellt. Die folgende Abbildung veranschaulicht das Zusammenspiel dieser einzelnen Komponenten und die entsprechenden Kontrollflüsse:

⁹⁸ Three-Tier-Architektur = Dreischichtenarchitektur

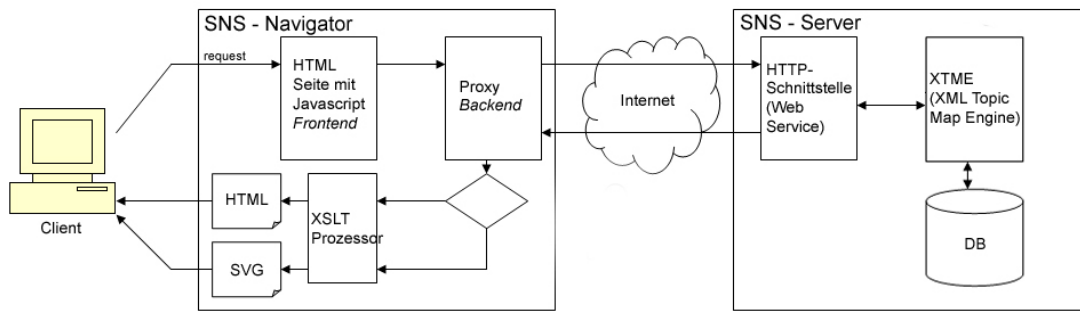


Abbildung 22: Systemkomponenten und Kontrollflüsse

Nachdem der Client eine Anfrage abgeschickt hat, werden die entsprechenden Parameter mittels Javascript an den Proxy weitergeleitet. Der Proxy leitet dann den Request zusammen mit den in ihm gekapselten Zugangsdaten mittels POST-Bindings über das Internet an den Web Service des SNS-Servers weiter. Dort wird die Anfrage an die XML Topic Map Engine⁹⁹ weitergeleitet, die die Zugangsdaten und den Funktionsaufruf mit den angegebenen Parametern überprüft, um anschließend Kontakt zur Datenbank aufzunehmen, die den entsprechenden Datensatz im XML-Format zurückgibt. Der Topic Map-Auszug wird dann über die HTTP-Schnittstelle an den Proxy des SNS-Navigators zurückgeschickt. Ein XSLT-Prozessor transformiert die eintreffende Topic Map im Proxy je nach Funktionsaufruf entweder als HTML-Tabelle oder als SVG-Grafik und gibt diese an den Client zurück.

6.1.2 Anwendungsarchitektur des SNS-Navigators

Der SNS-Navigator besitzt eine mehrschichtige Anwendungsarchitektur, die sich aus den drei Hauptkomponenten „Präsentation der Daten“, „Datenzugriff“ und „Datenverarbeitung“ zusammensetzt.

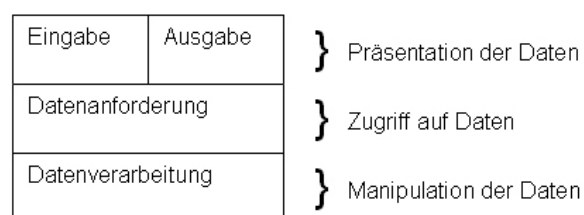


Abbildung 23: Anwendungsarchitektur des SNS-Navigators

⁹⁹ „Die XML Topic Map Engine (XTME) ist ein elementares Werkzeug zur Ablage, Navigation und Pflege individueller Topic Maps [...] Im Zentrum steht die Datenhaltung der Topic Map als XML-Dokument.“ [Schlumberger Sema GmbH (2002), S.14].

6.1.3 Einsatzbereiche der verschiedenen Webtechnologien

Die Anwendung basiert auf einem „Mix“ der Webtechnologien HTML, CSS, Javascript (Ajax), XML, XSLT, PHP und SVG.

In diesem Kapitel wird kurz dargestellt, in welchen Bereichen der Anwendung es sinnvoll ist, die oben genannten Webtechnologien einzusetzen.

Der Proxy wird auf Grund seiner Komplexität und seines hohen Funktionsumfangs mit PHP5 umgesetzt, dass mit der Bibliothek „libCurl“¹⁰⁰ die Funktionalitäten eines Proxys bietet und mit einer XSLT-Erweiterung die Transformation von XML-Dokumenten zur Verfügung stellt.

Alle Ein- und Ausgabeseiten der Benutzungsoberfläche bauen größtenteils auf der Auszeichnungssprache HTML auf, die das wichtigste Repräsentationsformat von Webseiten ist. Die Steuerung der Statusanzeige (s. Kap. 7.4) und die Überprüfung der Formularfelder sowie die Weiterleitung der Suchparameter an den Proxy werden mit Javascript-Funktionen abgebildet und Seiteninhalte, die immer identisch sind (Navigation) werden mit der PHP-Anweisung „include“ integriert.

Die Struktur der nach einer Suche zurückgelieferten HTML-Liste und der SVG-Grafik wird in XSL-Dateien gespeichert.

6.1.4 Proxy

Unter einem Proxy wird im Allgemeinen ein Dienstprogramm für Computernetze verstanden, das im Datenverkehr vermittelt.

Der Proxy des SNS-Navigators erfüllt darüber hinaus noch weitere Aufgaben:

- Speicherort der Zugangsdaten, die bei dem Datentransfer zwischen SNS-Navigator und SNS-Server mitgeliefert werden.
- Zwischenspeicher der eingetroffenen Topic Map.
- Transformation der zwischengespeicherten Topic Map in eine HTML-Tabelle (im Falle des Funktionsaufrufs „findTopics“, „findEvents“ und „anniversary“) oder SVG-Grafik (im Falle des Funktionsaufrufs „getPSI“).
- Weiterleitung der transformierten Topic Map an den Client.

Da die wichtigsten Funktionen der Anwendung innerhalb des Proxys enthalten sind, kann man diese Komponente als das „Herzstück“ des SNS-Navigators ansehen.

¹⁰⁰ Php.net (2006).

6.1.5 Struktur der Ein- und Ausgabeseiten

Jede Suchfunktion, die der SNS-Navigator bietet, wird als einzelne Seite abgebildet. Dazu gehören ebenfalls das Impressum und die Willkommenseite, die neben einem entsprechenden Inhalt über ein kleines Suchformular verfügen, mit dem der Nutzer sofort und ohne eine weitere Seite öffnen zu müssen, eine Suche nach Topics starten kann. Diese Seiten sind Bestandteil der Benutzungsoberfläche und liegen sowohl als deutsche als auch englische Versionen vor.

Es folgt eine Auflistung aller Seiten der Benutzungsoberfläche:

- Willkommenseite: index_de.php / index_en.php
- Suche nach Topics: suchetopics.php / findtopics.php
- Zufallssuche: suche.php / search.php
- Suche nach Umweltereignissen: sucheereignisse.php / findevents.php
- Suche nach Jahrestagsereignissen: jahrestag.php / anniversary.php
- Impressum: impressum.php / imprint.php

6.1.6 Darstellung aller Komponenten der Anwendung

Die folgende Abbildung präsentiert alle Komponenten der Anwendung. Aus Übersichtlichkeitsgründen wurde auf die englischsprachigen Komponenten verzichtet.

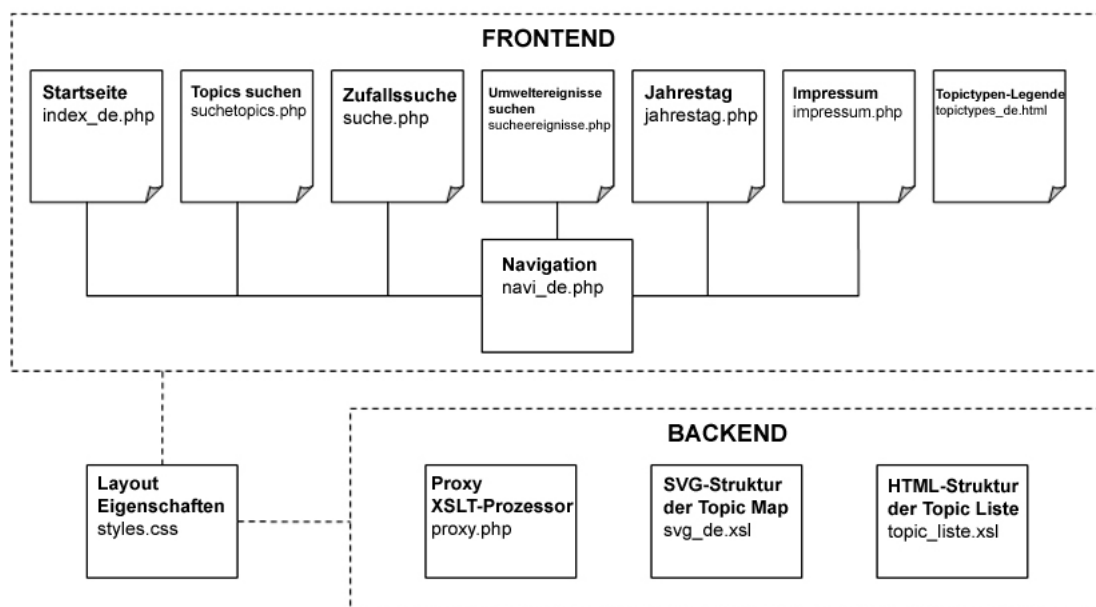


Abbildung 24: Darstellung und Gliederung aller Komponenten

6.1.7 Die „Zufallssuche“

Die „Zufallssuche“ ist eine Eigenkreation des Autors und soll potentielle Nutzer der Anwendung zum unbedarften „Stöbern“ in der Datenbasis des Umweltthesaurus einladen. Sie wählt beliebig eine Topic Map aus, ohne dass zuvor ein Suchbegriff eingegeben werden muss und gibt dem Nutzer somit einen ersten Eindruck von der Datenbasis und der Funktionsweise des Systems. Insbesondere ein fachfremder Nutzerkreis soll damit angesprochen und für weitere Suchanfragen inspiriert werden. Da die Identifikationsnummern der Umweltthesauruseinträge in aufsteigender Zahlenfolge geordnet sind (Bsp: uba_thes_**4911**, uba_thes_**30823**) kann mit einer Zufallszahl, die aus der Spanne aller Einträge gewonnen und an eine getPSI-Anfrage (s.Kap. 4.4.1) angehängt wird, diese Art einer zufälligen „Suche“ realisiert werden.

6.1.8 Model-View-Controller-Konzept

Das MVC-Konzept ist bei der vorliegenden Webapplikation in Reinform umgesetzt, da sich die Anwendung in die drei Einheiten Datenmodell (engl. **Model**), Präsentation (engl. **View**) und Programmsteuerung (engl. **Control**) aufteilt. Die XML-Datei von SNS steht für das Datenmodell, die XSL-Datei für die Präsentation und der Transformator für die Programmsteuerung. Die einzelnen Komponenten sind deutlich voneinander getrennt, was zu einer guten Strukturierung der Gesamtanwendung führt.¹⁰¹ Ein weiterer Vorteil dieses Modells besteht darin, dass die einzelnen Komponenten relativ problemlos ausgetauscht werden können, um die Anwendung an veränderte Anforderungen anzupassen.

6.1.9 Aktivitätsdiagramme

Die folgenden Aktivitätsdiagramme beschreiben die allgemeinen Abläufe der Anwendung und stellen die Reihenfolge, in der bestimmte Aktionen ausgeführt werden, dar. Der Ausgangspunkt der aufgezeigten Aktivitäten ist der Anwendungsfall „Umweltbegriffe und zugeordnete Begriffe finden“ (s. Kap. 5.1.), der der Web Service-Funktion „find-Topics“ (s. Kap. 4.4.1.) entspricht. Die grafischen Elemente der Aktivitätsdiagramme orientieren sich an dem UML-Standard 2.0 und werden in der folgenden Legende kurz erläutert:

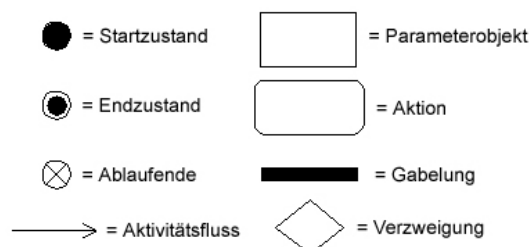


Abbildung 25: UML - Aktivitätsdiagramm Legende

¹⁰¹ Vgl. Bader, H. S.222.

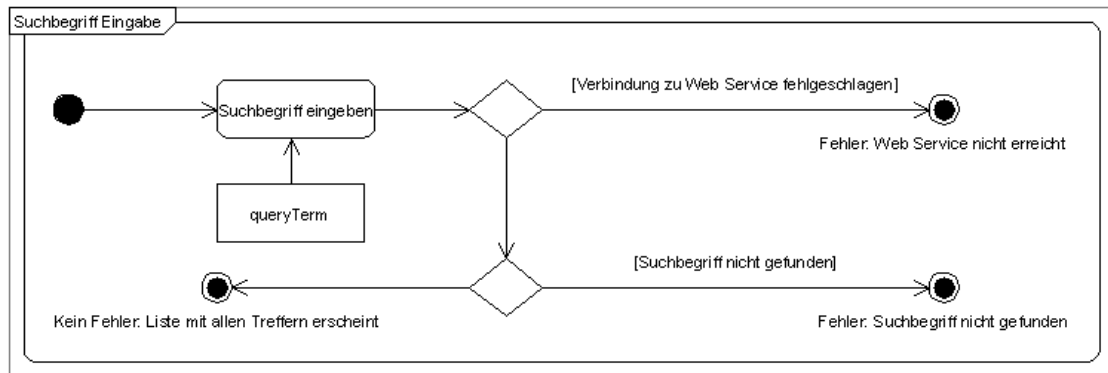


Abbildung 26: Schritt 1: Suchbegriff Eingabe

Wenn der Nutzer einen Suchbegriff eingibt wird die entsprechende Zeichenkette als Inhalt des Parameters „queryTerm“ zusammen mit der Suchfunktion „findTopics“ durch den Proxy an den SNS-Web Service geschickt. Die Statusanzeige springt auf „Datensätze werden geladen“. Es können bei dieser Aktivität zwei Fehler auftreten: Zum Einen, dass der Web Service aus technischen Gründen (inaktive Internetverbindung oder Serverausfall) nicht erreicht werden kann oder zum Anderen, dass die Suche kein Ergebnis liefert. In beiden Fällen wird eine entsprechende Fehlermeldung ausgegeben. Im günstigeren Fall wird vom System eine Liste mit allen Topics, die passend zur Suchanfrage gefunden wurden, generiert.

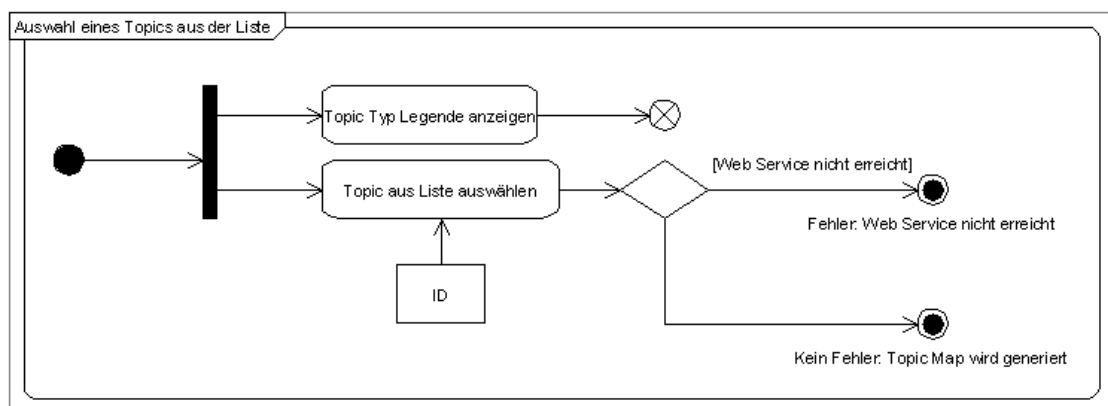


Abbildung 27: Schritt 2: Topic aus Liste auswählen

Der Nutzer kann nun einen Topic aus der Liste auswählen und sich die Topic Typ-Legende in einem separaten Fenster anzeigen lassen. Nach Auswahl eines Topics wird die entsprechende ID als Parameter innerhalb des Funktionsaufrufs „getPSI“ an den Server von SNS weitergegeben. Wenn kein Fehler auftritt, wird die grafische Navigationskomponente generiert.

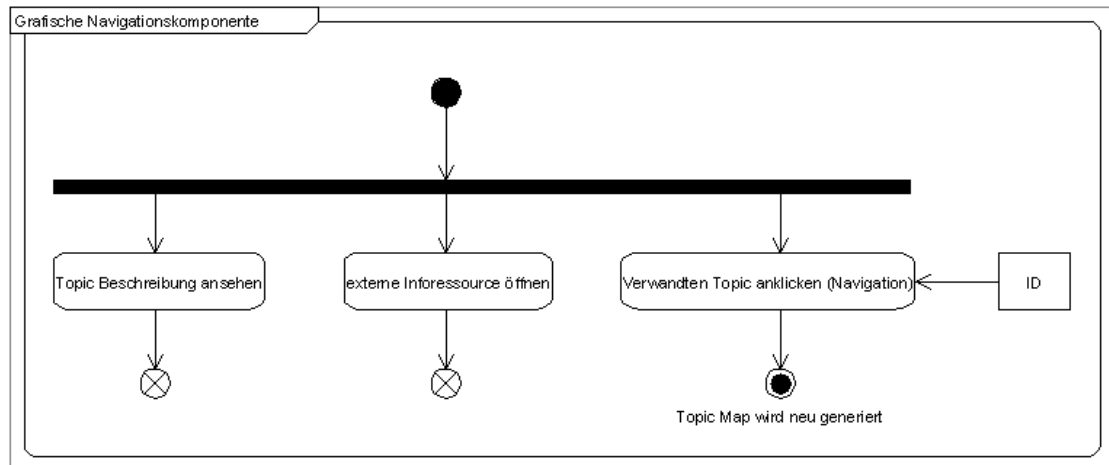


Abbildung 28: Schritt 3: Grafische Navigationskomponente

Die möglichen Aktivitäten bei der grafischen Navigationskomponente sind:

- **Topic Beschreibung ansehen:** Dies kann der Nutzer bewerkstelligen, indem er mit dem Mauszeiger über einen Topic fährt und die entsprechenden Zusatzinformationen in der Infobox aufnimmt.
- **Externe Informationsressource öffnen:** Wird ausgeführt, wenn der Nutzer auf den Link in der Infobox klickt oder auf einen Topic. In letzterem Fall erscheint ein Dialogfenster, bei dem nachgefragt wird, ob die externe Informationsressource in einem neuen Browserfenster geöffnet werden soll.
- Durch **Anklicken eines äußeren Topics** wird die Topic Map wieder neu generiert, auf Basis der mitgelieferten ID des ausgewählten Topics.

6.2 Gestaltung der Webapplikation

Da die Webapplikation in einem Webbrowser angezeigt wird, muss bei der Gestaltung der Benutzungsoberfläche darauf geachtet werden, dass die Darstellung in nahezu jedem Browser-Typ gleich aussieht. Die Konzeption der Gestaltung orientiert sich an einer Bildschirmauflösung von 1024 x 768 Pixeln, da diese die gängigste Einstellung auf den Computern eines durchschnittlichen Nutzer ist und zur Zeit vorausgesetzt werden kann. Außerdem muss darauf geachtet werden, dass die Schriftgröße nicht zu klein ist und ein ausreichender Kontrast zum Hintergrund gegeben ist, damit die Informationen vom Nutzer besser erfasst werden können. Die entsprechenden Layout-Eigenschaften werden als Cascading Style Sheets in eine separate Datei ausgelagert, um nicht jede einzelne Seite des Systems bei einer Änderung des Layouts anpassen zu müssen. Die Übersichtlichkeit wird durch eine logische Einteilung der verschiedenen Bereiche und dem Vermeiden von Informationsüberfluss gewahrt, der z.B. durch zu viele Auswahlmöglichkeiten bei der Hauptnavigation entstehen könnte.

6.2.1 Aufteilung der Benutzungsoberfläche in verschiedene Bereiche

Es gibt einzelne Bereiche der gesamten Benutzungsoberfläche innerhalb des Browser-fensters, die statischen und dynamischen Charakter haben. Der statische Bereich befindet sich an der obersten Stelle des Fensters, und wird somit von dem Nutzer als erstes wahrgenommen, da er es von anderen Medien (z.B. Büchern, Zeitschriften) gewöhnt ist in der Richtung von oben nach unten Informationen aufzunehmen. Dieser Bereich beinhaltet einen Schriftzug der Anwendung („SNS – Navigator“), ein zugehöriges Logo (kleine Visualisierung einer fiktiven Topic Map), ein Sprachauswahlmenü (Deutsch und Englisch), die Navigation zu den einzelnen Suchfunktionen, sowie zu der Willkommensseite und einem Impressum.

Die in dem statischen Bereich eingebettete Navigation erlaubt es dem Nutzer an jedem beliebigen Punkt in der Anwendung zu einem anderen Bereich zu wechseln.

Da der statische Bereich einen großen Einfluss auf das Gesamtbild der Anwendung hat, drückt er durch seine Farbgebung, das Logo und ein entsprechendes Hintergrundbild, dass beispielsweise eine Landschaftsaufnahme eines Kohlekraftwerks mit darauf zulaufenden Strommasten darstellt, den inhaltlichen Schwerpunkt der Anwendung aus, der auf Umweltinformationen liegt.

Von hoher Priorität ist die Statusanzeige, die dem Nutzer ein aktuelles Feedback darüber gibt, was die Anwendung augenblicklich von ihm erwartet (z.B. eine Aufforderung zur Suche), was sie gerade macht (z.B. Laden von Datensätzen) oder das Anzeigen eines Fehlers, wenn etwas nicht funktioniert hat. Die Statusanzeige hat ihrer Natur entsprechend einen dynamischen Charakter und steht für den Nutzer deutlich hervorgehoben direkt unter dem statischen Bereich auf der linken Seite, um somit ihre wichtige Funktion zu unterstreichen.

Der dynamische Inhaltsbereich nimmt den größten Raum ein und füllt die verbleibende Fensterfläche aus. In diesem Bereich werden die Suchergebnisse und die grafische Navigationskomponente dargestellt. Die folgende Abbildung veranschaulicht zusammenfassend die Aufteilung der einzelnen Bereiche im Browserfenster:



Abbildung 29: Aufteilung des Browserfensters in einzelne Bereiche

6.2.2 Farblegenden der Topic- und Assoziationstypen

Um dem Nutzer die zugehörige Hauptgruppe eines Topics sofort kenntlich zu machen, hat sich der Autor dazu entschlossen, eigene Farbschemata zu entwickeln.

Hierbei grenzen sich die 6 Hauptgruppen farblich voneinander ab, denen insgesamt 34 verschiedenen Topic-Typen untergeordnet sind. Im Folgenden wird die Farbauswahl der Hauptgruppen aufgezeigt und begründet:

- **Ereignisse** werden in der Signalfarbe **Rot** dargestellt, um deren Alleinstellungsmerkmal zu verdeutlichen und die Aufmerksamkeit des Nutzers auf sich zu ziehen. Aus Sicht des Autors haben die Umweltereignisse den größten informativen Charakter aller Topic Typen, da sie häufig mit einer Beschreibung und zusätzlichen externen Informationsressourcen aufwarten.
- Topics der Gruppe **Ort -> Land** werden durch einen **Ockerton** gekennzeichnet, der zumeist mit Erde, Land und Boden assoziiert werden kann.

- Topics der Gruppe **Ort -> Geschützter Bereich** sind durch die Farbe **Grün** spezifiziert, da Grün im Allgemeinen mit Biologie und Wäldern in Verbindung gebracht wird und diese Farbe bei der Gestaltung von Schildern von Naturschutzgebieten zum Einsatz kommt, die in der Regel grün umrahmt sind.
- Topics der Gruppe **Ort -> Gewässer** haben die Farbe **Blau**, da Gewässer auch auf handelsüblichen Landkarten oder in geographischen Informationssystemen blau abgebildet werden und Wasser in der visuellen Wahrnehmung eines Menschen einen bläulichen Farbton besitzt.
- Topics der Gruppe **Ort -> Verwaltungseinheit** bekamen vom Autor einen **violetten** Farbton zugeteilt, um sich von den anderen Farbtönen abzuheben. Eine sich auf eine Spezifikation stützende Zuordnung eines bestimmten Farbtons war nicht möglich, da es keine gemeingültige Norm gibt, wie z.B. Städte auf Landkarten abgebildet werden. Der Autor hat zu diesem Zweck mehrere Atlanten geprüft und musste feststellen, dass hierbei eine gewisse Willkür vorherrscht.
- Den **Thesauruseinträgen** wurde vom Autor ein **Grauton** zugeordnet, der die Neutralität, die einem solchen Wörterbucheintrag inne steckt, verdeutlichen soll. Im weiteren Sinne sind alle Thesauruseinträge und ihre Assoziationen im Graustufenbereich angelegt.

Die Assoziationstypen werden nur innerhalb der grafischen Navigationskomponente dargestellt und lehnen sich in ihrer Farbgebung an das Topic Typ – Farbschema an:

- **Oberbegriff** - Verbindungen werden **schwarz** dargestellt um sich im Graustufenbereich der Thesauruseinträge zu integrieren.
- **Unterbegriff** – Verbindungen werden **weiß** dargestellt, um ihre konträre Position zu Oberbegriff-Verbindungen zu verdeutlichen
- **Deskriptor** – Verbindungen haben die Farbe **Dunkelgrau**, angelehnt an die Thesauruseinträge.
- **Synonym** – Verbindungen werden **hellgrau** dargestellt um sich im Graustufenbereich der Thesauruseinträge zu integrieren.
- **Zusammensetzungen**¹⁰² werden **gelb** dargestellt, damit keine Verwechslungsgefahr zu einem bereits bestehenden Farbschema besteht.
- **Räumliche Überschneidungen** sind **violett**, da sie immer in Verbindung mit Orten und häufig mit Verwaltungseinheiten (Gemeinden, Kreise) auftreten.
- Eine **Ereignis Zusammengehörigkeit** bekommt die Farbe **Rot** zugeteilt, da diese Verbindung nur zwischen Ereignissen auftreten kann.

¹⁰² Mit Zusammensetzungen sind im Sinne von SNS Wörterzusammensetzungen gemeint. Bsp: „Bienenweide“ ist eine Zusammensetzung der Wörter „Biene“ und „Weide“.

6.2.3 Anzeige aller gefundenen Topics als Liste

Wenn der Nutzer eine Suchanfrage aufgegeben hat, bekommt er im günstigen Fall eine Liste aller gefundenen Topics zurück. Diese wird als HTML-Tabelle dargestellt, wobei immer nur zwanzig Treffer pro Seite angezeigt werden, damit der Nutzer den Überblick bewahren kann und ein Scrollen der Seite vermieden wird. Bei einer Treffermenge von mehr als zwanzig Treffern wird die Blätterfunktion aktiviert. Ebenso wird die Gesamttrefferanzahl angezeigt, sowie eine Beschreibung des Topic Typs, wenn der Nutzer einen Topic mit dem Mauszeiger überfährt. Auf die zusätzliche Anzeige der Topic Typen-Legende wurde verzichtet, in der alle 34 Icons aufgezählt werden, um der Gefahr einer Informationsflut entgegen zu wirken. Der Nutzer kann sich aber diese Topic Typen-Legende in einem zusätzlichen Browser-Fenster anzeigen lassen, indem er einen entsprechenden Link aktiviert.

Jeder Topic wird mit seinem Namen angezeigt und dem Datum seines Eintrags, sofern dieses vorhanden ist.

6.2.4 Grafische Anzeige der Topic Map

Wenn der Nutzer aus der Trefferliste einen speziellen Topic ausgewählt hat, d.h. wenn das System eine „getPSI“-Anfrage registriert, wird eine grafische Darstellung der entsprechenden Topic Map generiert.

Die gestalterische Konzeption dieser grafischen Navigationskomponente ist in hohem Maße abhängig von der zugrunde liegenden XML-Struktur des Topic Map-Ausschnitts vom Semantischen Netzwerkdienst. Nur die in ihr enthaltenen Informationen können grafisch dargestellt werden.

Die grafische Navigationskomponente kann einer grafischen Benutzungsoberfläche, einer GUI (**G**raphical **U**ser **I**nterface) gleichgesetzt werden, die dem Nutzer die Interaktion mit dem System über grafische, metaphorartige Elemente unter Verwendung eines Zeigegeräts (wie der Maus) erlaubt. Es gilt demnach folgende Elemente in sinnvoller Verbindung zueinander darzustellen:

- **Knoten/Topics** werden als Icons abgebildet, die Aufschluss über den Topic Typ geben. Ein darüber liegender Schriftzug beschreibt den Titel des Topics. Der ausgewählte Topic (im Weiteren „Basistopic“ genannt) liegt im Mittelpunkt der Topic Map und die mit diesem Topic assoziierten Topics befinden sich in einer stabilen Kreisbahn im gleichen Abstand zueinander um den Basistopic.
- **Kanten/Assoziationen** werden als direkte Verbindungslinien zwischen den Topics dargestellt, die ebenfalls durch ein Farbschema die Art ihrer Verbindung aufzeigen.
- Titel, Topic Typ, Beschreibung und externe Informationsressourcen des Topics, der gerade mit dem Mauszeiger überfahren wird, werden außerhalb der Topic Map in einer separaten **Infobox** angezeigt. Dort befindet sich auch eine Legende der verschiedenen Assoziationstypen.

Als geometrische Figur für die Topics/Icons bietet sich der Kreis an, da nur bei einem Kreis gewährleistet ist, dass die Kanten/Verbindungslinien bei gleicher Länge sich ohne einen optisch als unsauber empfundenen Versatz an die Knoten anheften können. Folgende Abbildung verdeutlicht dieses Prinzip:

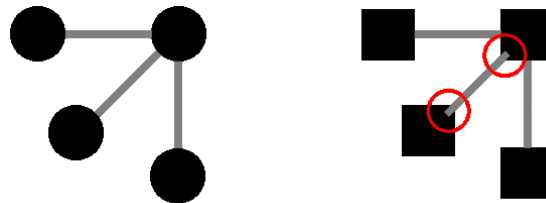


Abbildung 30: Topics in Kreisform vs. rechteckige Form

Die kreisförmige Anordnung der Topic Map bietet aus Sicht des Autors den weiteren Vorteil, dass die dargestellte Topic Map in ihrer Gesamtheit in einem exakten Quadrat eingebettet ist und da die Bildschirmauflösung in der Regel ein liegendes Rechteck im Verhältnis 1.33 (Breite) zu 1 (Höhe) ist, kann der restliche verfügbare Platz für die Infobox genutzt werden. Die folgende Abbildung veranschaulicht diese Bildschirmaufteilung in Topic Map (Quadrat) und Infobox (Rechteck):

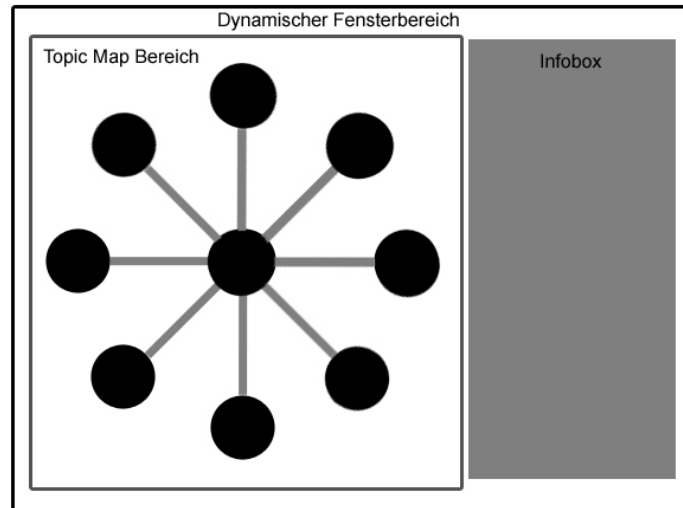


Abbildung 31: Bereiche der grafischen Navigationskomponente

Neben der dynamischen Darstellung der Topic Map gibt es interaktive grafische Veränderungen, wenn der Nutzer den Mauszeiger über einen Topic bewegt. In diesem Fall wird der ausgewählte Topic größer dargestellt und die entsprechenden Informationen werden in der Infobox angezeigt. Befindet sich der Mauszeiger nicht über einem Topic werden die Informationen des Basistopics angezeigt. Klickt der Benutzer auf einen äußeren Topic, dann wird dieser zum Basistopic und befindet sich in der Mitte der neu generierten Topic Map. Sollte der ausgewählte Topic über eine externe Informations-

ressource verfügen, öffnet sich zunächst ein Dialogfenster, bei dem der Nutzer entscheiden kann, ob er die externe Informationsressource zusätzlich in einem neuen Browserfenster öffnen möchte.

Ein grundlegendes Problem dieser Anzeigeform einer Topic Map ist die Darstellung von mehr als zwanzig Topics, da sich in diesem Fall die äußeren Knoten überlagern und somit nicht mehr in ihrer Vollständigkeit wahrgenommen werden können. Die Ursache liegt in der großen Gesamtmenge der Topics, wodurch sich der Gradabstand der Knoten verringert. Der Nutzer kann in diesem Fall über jedes einzelne äußere Topic mit dem Mauszeiger gleiten, um die zugehörigen Informationen in der Infobox aufnehmen zu können.

6.3 Technische Voraussetzungen

Der Nutzer benötigt für die korrekte Darstellung und volle Funktionalität des SNS-Navigators einen SVG-fähigen Webbrowser, bei dem Javascript aktiviert ist. Zur Zeit der Erstellung dieser Arbeit erfüllte der Firefox-Browser ab Version 1.5. von Mozilla diese Voraussetzungen.

Der weit verbreitete Internet Explorer bietet noch keine native SVG-Unterstützung, d.h. es muss ein zusätzliches Plug-In installiert werden, im Idealfall das SVG-Plug-In des Unternehmens Adobe ab Version 3. Dieses Plug-In verfügt gegenüber der nativen SVG-Unterstützung des Firefox über einige Spezialfunktionen, wie z.B. eine beliebige Skalierung der SVG-Grafik.

Für die Konzeption der Anwendung geht nun hervor, dass auf der Willkommenseite des SNS-Navigators diese Systemanforderungen vermerkt und gegebenenfalls an geänderte Bedingungen angepasst werden.

7 Realisierung

Auf der Grundlage der oben beschriebenen Konzeption erfolgte die Programmierung der Anwendung zunächst auf dem Computer des Autors in Verbund mit einem lokalen Webserver. Im nächsten Schritt wurde die fertige Anwendung auf den Studentenserver „reloaded.iuk.hdm-stuttgart.de“ übertragen. Auf „reloaded“ ist PHP 5 mit der zusätzlichen PHP-Bibliothek „libCurl“¹⁰³ installiert, welche die Proxy-Funktionen beinhaltet.

7.1 Web Service Anfrage mittels GET / POST Binding durch einen Proxy

Wie in der Konzeption aufgezeigt worden ist, werden die Suchparameter, die der Nutzer in einem Formular des Frontends definiert hat, mittels GET an den Proxy (Backend) der Anwendung gesendet, wo diese aufgefangen werden und als POST Binding mit Hilfe der PHP-Bibliothek „libcurl“ an den Server von SNS weitergeliefert werden.

Folgender Programmausschnitt zeigt diesen Prozess auf:

```
[...]

// Parameter der Suchformulare auffangen

$id = $_GET["id"]; // Id eines Topic

$path = $_GET["path"]; // Einschränkung auf Topic Typen
$queryTerm = $_GET["queryTerm"]; // Zeichenkette, nach der gesucht
werden soll
$ignoreCase = "true"; // true: Groß-/Kleinschreibung in queryTerm
ignorieren
$lang = $_GET["lang"]; // Sprache
$searchType = $_GET["searchType"]; // exact, beginsWith, contains -->
geforderte Vorkommen von queryTerm in den Topicdaten.
$fields = $_GET["fields"]; // Topic Attribute, in denen gesucht werden
soll
$offset = $_GET["offset"]; // "Blätter"-Funktion in der
Gesamtergebnisliste

[...]

$user = "xyz"; // User
$password = "xyz"; // Passwort

if ($queryTerm != "") { // entsprechende Parameter für die Web Service
- Funktion "findTopics"
$function = "findTopics";
$POSTfields =
"path=".$path."&queryTerm=".$queryTerm."&ignoreCase=".$ignoreCase."&lang=".$lang."&searchType=".$searchType."&fields=".$fields."&offset=".$offset."&user=".$user."&password=".$password;
```

¹⁰³ Vgl. PHP.net (2006).

```

}
[...]
```

```

$url = "http://www.semantic-network.de/service-xtm-
2.0/xtm/command/". $function; // URL des Web Service für GET-Anfragen,
siehe: http://www.semantic-network.de/service-xtm-2.0/xtm/sns.wsdl
$ch = curl_init(); // curl handler initialisieren

[...]
```

```

curl_setopt($ch, CURLOPT_URL, $url); // URL welche angesprochen werden
soll (hier vom Web Service des UBA)
curl_setopt($ch, CURLOPT_FAILONERROR, 1);
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1); // redirects erlauben
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // return als eine
variable
curl_setopt($ch, CURLOPT_TIMEOUT, 60); // Time Out nach 61s
curl_setopt($ch, CURLOPT_POST, 1); // POST Methode setzen
curl_setopt($ch, CURLOPT_POSTFIELDS, $POSTfields); // entsprechende
Parameter anhängen

[...]
```

```

$result = curl_exec($ch);

curl_close($ch);
```

An dieser Stelle möchte der Autor darauf hinweisen, dass diese Form der Datenweitergabe ebenso aus Sicherheitsgründen notwendig ist, da die modernen Browser einen direkten Request mit dem XMLHttpRequest-Objekt von Ajax (s. Kap. 4.7) an einen fremden Server unterbinden.

7.2 XML-Transformation der zurückgelieferten Topic Map

Je nach Suchfunktion wird das zurückgelieferte XML-Dokument entweder in eine HTML-Tabelle oder in eine SVG-Grafik umgewandelt. Dies geschieht mit der in PHP5 implementierten XSLT-Erweiterung, welche die benötigte Funktionalität zur Verfügung stellt. Das folgende Code-Fragment zeigt den Ablauf der Transformation auf:

```

$document = DOMDocument::loadXML($result);
$stylesheet = DOMDocument::load('topic_list_de.xsl');
$processor = new XSLTProcessor;
$processor->importStylesheet($stylesheet);
print $processor->transformToDoc($document)->saveXML();
```

Zuerst wird das XML-Dokument entgegengenommen, im nächsten Schritt wird der entsprechende XML-Stylesheet geladen, dann wird ein XSLT-Prozessor erzeugt, die Transformation wird durchgeführt und im Anschluß wird das neue transformierte Dokument ausgegeben (in diesem Fall die flache Liste der gefundenen Topics als HTML-Tabelle).

7.3 Darstellung der SVG-Anzeige in einem I Frame

Die Darstellung der grafischen Navigationskomponente in der HTML-Seite des Frontends gestaltet sich etwas umfangreicher im Gegensatz zur transformierten HTML-Ausgabe.

Dr. Thomas Meinike hat in seinem Artikel „Know-how: Statische / dynamische SVG-Grafiken“¹⁰⁴ aufgezeigt, dass es mit Hilfe eines I Frame bzw. Object-Elements möglich ist, eine SVG-Grafik in einer HTML-Seite einzubetten. Die Anwendung vollzieht nach dem Erhalt der Topic Map die folgenden Schritte:

1. Die Transformation der Topic Map wird durchgeführt.

```
$document = DOMDocument::loadXML($result);
$stylesheet = DOMDocument::load('svg_de.xsl');
$processor = new XSLTProcessor;
$processor->importStylesheet($stylesheet);
$svgfile = $processor->transformToDoc($document)->saveXML();
```

2. Die SVG-Grafik wird als separate Datei „anzeige.svg“ auf dem Server gespeichert.

```
// SVG-Datei (über)schreiben

$Datei = fopen("anzeige.svg", "w");
flock($Datei,2);
fwrite($Datei, $svgfile);
flock($Datei,3);
fclose($Datei);
```

3. Die SVG-Datei wird in einem I Frame ausgegeben.

```
// SVG-Datei Ausgabe in einem I Frame

print "<iframe id='dynsvg' src='anzeige.svg' width='83%'
height='83%' frameborder='0' align='left'>";
print "  <p>Zur Anzeige der Grafik wird der <a
href=\"http://www.adobe.com/svg/viewer/install/main.html\">
SVG Viewer von Adobe</a> ben&ouml;tigt!</p>\n";
print "</iframe>\n";
```

¹⁰⁴ Vgl. Meinike, T. (2002).

7.4 Realisierung der Statusanzeige mit Ajax

Für die Generierung der Statusanzeige in den Skripten des Frontends hat sich der Autor eine Funktion des XMLHttpRequest-Objekts (s. Kap. 4.7) zu Nutze gemacht. „Diese Funktion – auch Callback-Funktion genannt – wird jedes Mal aufgerufen, wenn sich der Zustand der http-Anfrage ändert. Das XMLHttpRequest-Objekt definiert dazu fünf verschiedene Zustände.“¹⁰⁵

Tabelle 3: Zustände für das XMLHttpRequest-Objekt¹⁰⁶

Zustand	Beschreibung
0	Nicht initialisiert
1	Lädt gerade
2	Fertig geladen
3	Wartet auf Rückgabe
4	Vollständig

Von Bedeutung für die Anwendung sind die Zustände 1 und 4.

Die Callback-Funktion überprüft zunächst ob eine Suchanfrage vorliegt. Sollte keine Suchanfrage vorliegen wird die Statusanzeige wie folgt dargestellt:

Status: Warte auf Suchanfrage ?

Abbildung 32: Statusanzeige: Wartemodus

Sobald der Nutzer eine Suchanfrage aufgibt und die Daten geladen werden (Zustand 1), springt die Statusanzeige um und es erscheint folgende Ausgabe:

Status: Daten werden geladen

Abbildung 33: Statusanzeige: Ladevorgang

Um dem Nutzer in diesem Moment zu suggerieren, dass tatsächlich etwas im Hintergrund der Anwendung passiert, führt das grün-weiße Icon eine Drehung aus. Es handelt sich hierbei um ein animiertes Gif-Bild.

¹⁰⁵ Wenz, C. (2006), S.22.

¹⁰⁶ Wenz, C. (2006), S.22.

Wenn die Daten erfolgreich geladen wurden (Zustand 4), erscheint die folgende Anzeige:



Abbildung 34: Statusanzeige: Ladevorgang abgeschlossen

Sollte vom Proxy statt eines Suchergebnisses eine Fehlermeldung zurückkommen, gibt die Statusanzeige die folgende Meldung aus:



Abbildung 35: Statusanzeige: Ladefehler

Der entsprechende Quellcode befindet sich im Head-Bereich der Frontendskripte und sieht folgendermaßen aus:

```
<head>

<script language="JavaScript"
  type="text/javascript">
// <![CDATA[

// Abfangen der Parameter
var id = "<?php echo($_GET['id']); ?>";
var path = "<?php echo($_GET['path']); ?>";
var queryTerm = "<?php echo($_GET['queryTerm']); ?>";
var ignoreCase = "<?php echo($_GET['ignoreCase']); ?>";
var lang = "de";
var searchType = "<?php echo($_GET['searchType']); ?>";
var fields = "<?php echo($_GET['fields']); ?>";
var offset = "<?php echo($_GET['offset']); ?>";

var XMLHTTP = null;

if (window.XMLHttpRequest) {
  XMLHTTP = new XMLHttpRequest();
} else if (window.ActiveXObject) {
  try {
    XMLHTTP =
      new ActiveXObject("Msxml2.XMLHTTP");
  } catch (ex) {
    try {
      XMLHTTP =
        new ActiveXObject("Microsoft.XMLHTTP");
    } catch (ex) {
    }
  }
}

function DatenAusgeben() {

  if (XMLHTTP.readyState == 1) {    // Lädt gerade
    document.form.status.value = ['Daten werden geladen'];
    document.images.bild.src="icons/ladesymbol.gif";
  }
}
```

```

    if (XMLHTTP.readyState == 4) { // Vollständig geladen
        var d = document.getElementById("Daten");
        var e = XMLHTTP.responseText;
        var Ergebnis = e.search(/Der Web Service konnte leider nicht
erreicht werden.+/);
        if (Ergebnis != -1) {
            d.innerHTML += XMLHTTP.responseText;
            document.form.status.value = ['Ladefehler'];
            document.images.bild.src="icons/aktion.gif";
        } else {
            d.innerHTML += XMLHTTP.responseText;
            document.form.status.value = ['Datensätze geladen'];
            document.images.bild.src="icons/haken.gif"; }
        }
    }
}

window.onload = function() {

    if(id != "" || queryTerm != "" )    // Testen, ob eine Suchanfrage
vorliegt
    {

        XMLHTTP.open("GET", "proxy.php?path=" + path + "&queryTerm=" +
queryTerm + "&ignoreCase=" + ignoreCase + "&lang=" + lang +
"&searchType=" + searchType + "&fields=" + fields + "&offset=" +
offset + "&id=" + id);
        XMLHTTP.onreadystatechange = DatenAusgeben;
        XMLHTTP.send(null);

    } else {
document.form.status.value = ['Warte auf Suchanfrage'];
    }

}

function zeigeLegende() {
F1 = window.open("topicTypes_de.html", "Legende",
"width=390px,height=410px,left=400px,top=250px");
}

function ShowImg(link) {
    document.images.icon.src=link;
}

function ShowDescription(txt) {
    document.desc.desc.value = [txt];
}

// ]]>
</script>

<title>SNS - Navigator...Topics suchen</title>
</head>

```

Im obersten Bereich des JavaScript-Codes werden die entsprechenden Suchparameter abgefangen, die der Nutzer durch seine Eingaben in den Formularfeldern definiert.

Im darauf folgenden Block wird das XMLHttpRequest-Objekt instanziiert.

Dann werden die einzelnen Funktionen definiert, die bei den entsprechenden Ladezuständen aktiviert werden.

7.5 Entwurf der Icons (Topic types der SNS-Topic Map)

Der Autor hat auf Basis der in der Konzeption festgelegten Farbschemen kreisförmige Icons im SVG-Format entwickelt. Die folgende Tabelle zeigt das Ergebnis.

Tabelle 4: Auflistung aller Topic Typen und die vom Autor entworfenen zugehörigen Icons

SNS – Topic Typ - Legende			
Ereignis -> Konferenz		Ort -> Land -> Insel	
Ereignis -> Konvention		Ort -> Land -> Landschaft	
Ereignis -> Katastrophe		Ort -> Land -> Berg	
Ereignis -> Richtlinie		Ort -> Land -> Gebirge	
Ereignis -> Geschichte		Ort -> Land -> Naturraum	
Ereignis -> Störfall		Ort -> Geschützter Bereich -> Biosphäre	
Ereignis -> Aktion		Ort -> Geschützter Bereich -> Nationalpark	
Ereignis -> Gründung		Ort -> Geschützter Bereich -> Naturpark	
Ereignis -> Gesetz		Ort -> Geschützter Bereich -> Schutzgebiet	
Ereignis -> Schiffsunfall		Ort -> Gewässer -> Kanal	
Ereignis -> Natur des Jahres		Ort -> Gewässer -> See	
Ereignis -> Publikation		Ort -> Gewässer -> Stausee	
Ort -> Verwaltungseinheit -> Gemeinde		Ort -> Gewässer -> Fluss	
Ort -> Verwaltungseinheit -> Kreis		Ort -> Gewässer -> Meer	
Ort -> Verwaltungseinheit -> Staat		Thesauruseintrag -> Einzelwort	
Ort -> Verwaltungseinheit -> Bundesland		Thesauruseintrag -> Deskriptor	
Ort -> Land -> Wassereinzugsgebiet		Thesauruseintrag -> Nichtdeskriptor	

7.6 Struktur der XML-Stylesheets

Dieses Kapitel zeigt die Struktur der vom Autor entwickelten XML-Stylesheets auf, die für die Transformation der XML-Topic Map in ein neues Ausgabeformat notwendig sind, d.h. HTML bei den Web Service Funktionen „findTopics“, „findEvents“ und „anniversary“ und SVG bei „getPSI“). Der Kopfbereich der beiden XML-Stylesheets ist identisch:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:tmf="http://www.semantic-network.de/xmlns/XTM/2005/2.0/"
  xmlns:tm="http://www.topicmaps.org/xtm/1.0/">
```

Ein Stylesheet ist ein wohlgeformtes Dokument, daher beginnt es mit einer XML-Deklaration. Dann werden im Dokumentenelement stylesheet die XSLT-Elemente mit ihrem eigenen Namensraum als Attribute angegeben. Für den späteren Zugriff auf Daten innerhalb der Elemente der Topic Map ist es notwendig, den entsprechenden Namensraum anzugeben.

7.6.1 Listenanzeige aller gefundenen Topics (HTML-Ausgabe)

Bei der HTML-Ausgabe lautet die nächste Zeile:

```
<xsl:output method="html" />
```

Das Element output legt die Ausgabestruktur fest, in diesem Falle HTML.

```
<xsl:template match="/">
  <html>
    <body><br /><br />
      <table border="0"> <!-- Rahmentabelle -->
        <tr><td align="center" valign="top">
[...]
```

Innerhalb des template-Elements wird die Struktur festgelegt, d.h. wie das Dokument in der Ausgabe aussehen soll. Es wird in diesem Bereich also eine Vorlage definiert, die mit den Daten des Originals gefüllt wird. Die Daten werden aber im Gegensatz zur Erstellung einer HTML-Seite nicht eingegeben, sondern durch spezielle Anweisungen aus dem Original übernommen. Zur besseren Veranschaulichung, aus welchen Bereichen der Topic Map das Stylesheet die entsprechenden Daten übernimmt, wird zunächst ein Teilausschnitt einer Topic Map von SNS aufgezeigt, die bei einer findTopics - Suchanfrage nach dem Begriff „Biene“ erscheint:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<topicMapFragment
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="http://www.semantic-network.de/xmlns/XTM/2005/2.0/">
  <listExcerpt totalSize='19' pageSize='20' offset='0' /> --> Gesamtanzahl der Suchergebnisse.
                                                                Wichtig für die Blätterfunktion.
  <requestParameters>password=XYZ&user=XYZ&searchType=contains&lang=de&
  queryTerm=Biene&ignoreCase=true&isActive=true&fields=names&path=/ } Suchparameter
  </requestParameters>

  <topicMap xmlns="http://www.topicmaps.org/xtm/1.0/">

    <topic id="uba_thes_4911"> --> ID des Topics. Wichtig für die "getPSI"-Anfrage.

      <instanceOf>
        <topicRef xlink:href="http://www.semantic-network.de/xmlns/
          XTM/2005/2.0/sns-classes_2.0.xtm#descriptorType"/> } Topic - Typ Angabe
      </instanceOf>
      <baseName>
        <scope>
          <topicRef xlink:href="http://www.topicmaps.org/xtm/1.0/language.xtm#de"/> --> Sprachenangabe
        </scope>
        <baseNameString>Biene</baseNameString> --> Titel des Topics
      </baseName>

      [...]

    </topic>
  </topicMap>
</topicMapFragment>

```

Abbildung 36: Topic Map Auszug (findTopics) mit Erklärungen

In dem XML-Stylesheet wird nun definiert, an welcher Stelle sich die wichtigen Daten befinden.

```

<xsl:for-each select="tmf:topicMapFragment/tm:topicMap/tm:topic">

  <tr>
    <xsl:variable name="id" select="@id" />

    <!-- Topic Typen Ausgabe (Alle Topic Typs sind entnommen aus der
    Anwenderdokumentation von SNS) -->

    <xsl:variable name="topictyp">http://www.semantic-
    network.de/xmlns/XTM/2005/2.0/sns-classes_2.0.xtm#</xsl:variable>

    <td class="liste">
      <xsl:choose>

        <xsl:when
          test="tm:instanceOf/tm:topicRef/@xlink:href=concat($topictyp,
          'conferenceType')">
          <a id="link1" name="link1" href="#"
          onmouseover="ShowImg('icons/konferenz.gif');ShowDescription('=
          Ereignisse -> Konferenz');"></a> </xsl:when>
          [...]

```

Zunächst wird eine for-each-Schleife initialisiert, die festlegt, dass alle Topics der Topic Map als Bestandteil eines Topic Map-Fragments durchlaufen werden, wobei die Anweisungen innerhalb des Elements einmal pro Knoten ausgeführt werden.

Dann wird eine Variable definiert, die das Attribut eines Topics, seine ID, auffängt.

Um unter der Anzahl von verschiedenen Topic Typen die richtige Auswahl zu treffen, wird `<xsl:choose>` verwendet. In den folgenden `<xsl:when>` Anweisungen wird

jeweils festgelegt, welches Bild und welcher Javascript-Parameter bei einem Topic-Typ ausgegeben werden soll. Aus Übersichtlichkeitsgründen werden nicht alle 34 Auswahlblöcke an dieser Stelle aufgezeigt. Das Teilergebnis der bis zu diesem Punkt erläuterten Anweisungen wäre demnach die Darstellung der ersten Tabellenspalte mit den entsprechenden Icons der gefundenen Topics, die beim Darübergleiten der Maus in einer zweiten Tabelle ausgegeben werden.

```

</xsl:choose>
</td>
<!-- Topic - Namen Ausgabe -->
<td class="liste">

<xsl:for-each select="tm:baseName">

  <xsl:choose>
    <xsl:when
test="tm:scope/tm:topicRef/@xlink:href='http://www.topicmaps.org/xtm/1
.0/language.xtm#de'">
      <xsl:choose>

        <xsl:when
test="..../tm:instanceOf/tm:topicRef/@xlink:href=concat($topictyp,
'conferenceType')">
          <a>
            <xsl:attribute name="href">
              suche.php?id=<xsl:value-of select="$id" />
            </xsl:attribute>

            <xsl:attribute name="onmouseover">
              ShowImg('icons/konferenz.gif');ShowDescription('=
Ereignisse -> Konferenz');
            </xsl:attribute>
            <xsl:value-of select="tm:baseNameString" />
          </a>
        </xsl:when>
      </xsl:choose>
    </td>
  </xsl:for-each>
</td>
</tr>
</table>

```

Im oben gezeigten Abschnitt des XML-Stylesheets wird der Name des Topics ausgegeben. Hierfür muss eine for-each-Schleife innerhalb des Elements baseName durchgeführt werden, um eine Selektion der ausgewählten Sprache zu gewährleisten. Gleichzeitig ist der dargestellte Titel auch ein Hyperlink, der bei Aktivierung die getPSI-Funktion startet. Das Javascript-Event onmouseover wird bei dem Überfahren des Topic Titels aktiviert, das eine Ausgabe der entsprechenden Deutung des Topic Typs in einer separaten Tabelle auf der Seite zur Folge hat.

```

</table> <!-- Ende der Tabelle im 1. Feld der Rahmentabelle -->

</td><td align="center" valign="top">

  <table border="1" bordercolor="#041c17" frame="border" rules="none"
style="background-color: #516134; border-collapse: collapse;">
    <tr><th colspan="2" class="liste">Topic - Typ:</th>
    </tr><tr>
      <td class="liste"></td>
      <td class="liste"><form name="desc"><input type="text"
name="desc" size="51" class="topictyp" value="Zur Anzeige den
Mauszeiger über einen Topic bewegen" /></form></td>
    </tr><tr>

```

```

        <td></td><td><a href="javascript:zeigeLegende()">Zeige Legende
mit allen Topic Typen</a></td>
    </tr>
</table>

</td></tr></table> <!-- Ende Rahmentabelle -->
</body>
</html>

</xsl:template>
</xsl:stylesheet>

```

Gegen Ende des XML-Stylesheets wird die 2. Tabelle generiert, in der die entsprechenden Zusatzinformationen zur Art des Topic Typs beim Überfahren eines Topics mittels Javascript angezeigt werden.

Das Ausgabedokument sieht im Falle dieses Beispiels dann folgendermaßen aus:

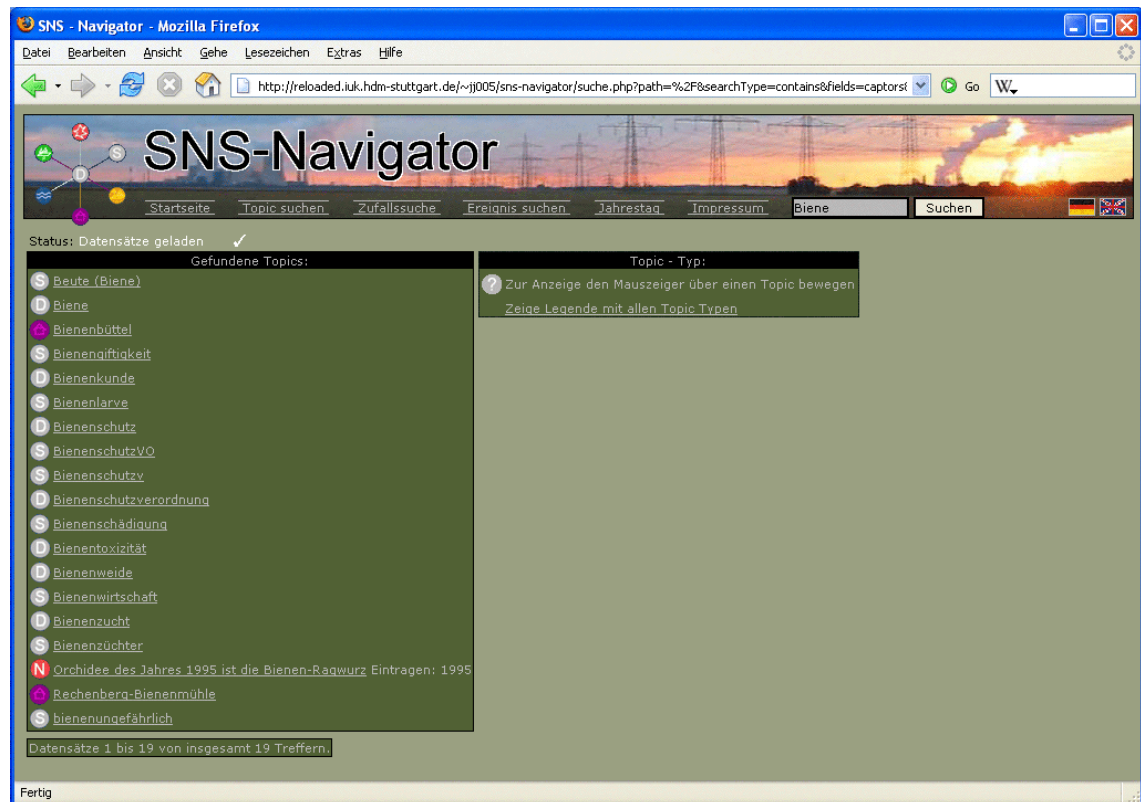


Abbildung 37: Screenshot der Listenanzeige

7.6.2 Grafikanzeige der XML Topic Map mit SVG

Das XML-Stylesheet `svg_de.xsl` legt die repräsentative Ausgabe der Topic Map als SVG-Grafik fest und definiert XML im Head-Bereich statt HTML als Ausgabeformat, da SVG ein Substandard von XML ist.

```
<xsl:output
  method="xml"
  doctype-public="-//W3C//DTD SVG 1.0//EN"
  doctype-system="http://www.w3.org/TR/2001/REC-SVG-
20010904/DTD/svg10.dtd"
  encoding="ISO-8859-1" version="1.0"
  media-type="image/svg+xml" indent="yes" />

<xsl:template match="/">
```

Mit dem Attribut `media-type` wird dem XSLT-Prozessor mitgeteilt, dass es sich bei dem Ergebnisdokument um eine SVG-Grafik handelt.

Die nächste Zeile legt mittels `viewBox` das sichtbare Koordinatensystem der SVG-Grafik fest.

```
<svg viewBox="-340 -340 1060 680" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" onload="Init(evt)">
```

In dem folgenden `defs`-Element werden alle Elemente definiert, die nicht sofort gezeichnet, sondern für einen späteren Gebrauch vorbereitet werden. Das sind in diesem Fall alle Javascript-Funktionen, die eine interaktive grafische Veränderung herbeiführen und die Darstellung der Informationen in der Infobox steuern. Mit dem oben aufgezeigten Ladezustand-Event `onLoad` wird bewirkt, dass die Javascript-Funktionen erst in Kraft treten, wenn die SVG-Grafik vollständig geladen wurde.

```
<defs>
<script xlink:href="js_funktionen_svg.js" type="text/javascript"/>
</defs>
```

Nun werden die statischen Inhalte der SVG – Grafik gezeichnet, darunter fallen z.B. die Hintergrundfarbe und der Rahmen der Infobox.

```
<!-- Hintergrundfarbe -->
<rect x="-1000" y="-1000" width="3000" height="2000" style="fill:
#9AA081" />

<!-- Infobox -->
<rect x="350" y="-295" width="380" height="590" style="fill:
#516134; stroke: #000000; stroke-width: 2px" />

<!-- Statischer Inhalt von Infobox (Überschriften und Assoziationstyp-
Legende) -->

  <text x="360" y="-270" font-size="14" fill="#bab9b9">Topic -
Typ</text>
  <line x1="355" y1="-268" x2="700" y2="-268"
style="stroke:#000000;stroke-width:1;opacity:0.1" />
[...]
```

Im nächsten Schritt erfolgt die Darstellung der Kanten. Sie werden vor den Knoten gezeichnet, damit die Knoten die Kanten überlappen und nicht umgekehrt.

```
<!-- 1. Durchlauf für die Darstellung der Kanten -->

<xsl:for-each select="tmf:topicMapFragment/tm:topicMap/tm:topic">
```

```

<xsl:variable name="alle" select="last()" /> <!-- Anzahl aller Topics
-->
<xsl:variable name="ta" select="360 div ($alle - 1)" /> <!--
Berechnung des Gradabstands der äußeren Topics -->
  <xsl:variable name="id" select="@id" /> <!-- Id pro Topic für
Verlinkung -->
  <xsl:variable name="pos" select="position()" /> <!-- Derzeitige
Topic-Nummer in der Schleife -->
<xsl:variable name="um" select="($pos - 1) * $ta" /> <!-- Addierung
des Gradabstands pro Topic -->

  <xsl:choose>
    <xsl:when test="$pos=1"> </xsl:when>
<xsl:otherwise>

  <xsl:for-each select="../tm:association/tm:member">

    <xsl:variable name="topicid" select="tm:topicRef/@xlink:href" />

    <xsl:if test="$topicid=$id">

<!-- Darstellung der Kanten -->

      <xsl:variable name="assoc"
select="tm:roleSpec/tm:topicRef/@xlink:href" />

      <xsl:choose>
        <xsl:when test="$assoc='http://www.semantic-
network.de/xmlns/XTM/2005/2.0/sns-classes_2.0.xtm#widerTermMember'">
<!-- Oberbegriff -->
<line y1="50" x2="0" y2="245" transform="rotate({$um})"
style="stroke:#000000;stroke-width:6" />
        </xsl:when>
      </xsl:choose>
    </xsl:if>
  </xsl:for-each>
</xsl:otherwise>
</xsl:choose>

```

Für die Darstellung der Kanten wird zunächst die Gesamtanzahl aller Knoten benötigt, um den erforderlichen Gradabstand und die Anzahl der Kanten zu definieren. Der Ablauf gestaltet sich folgendermaßen:

- `last()` liefert die Anzahl aller Knoten des Dokuments. Der Knoten in der Mitte wird abgezogen (-1).
- Der Gradabstand der Kanten wird berechnet, indem 360 Grad durch die Anzahl der äußeren Knoten geteilt wird
- Der Gradabstand wird bei jeder neu generierten Kante hinzuaddiert, bzw. das Koordinatensystem wird mit `rotate` um den Gradabstand weitergedreht, bevor die Kante gezeichnet wird.
- Mit einer `<xsl:choose>` Anweisung wird gewährleistet, dass der mittlere Topic ausgespart (Position 1) wird und die oberen Anweisungen sich nur auf die Kanten auswirken, die als Verbinder zu den äußeren Knoten fungieren.

Dieser Ablauf kommt ebenso im Stylesheet bei der anschließenden Darstellung der äußeren Knoten zum Einsatz.

Im letzten Schritt des obigen Code-Ausschnitts werden die Kanten, je nach ihrem Assoziationstyp, der in der Topic Map innerhalb des <association>-Elements und dort unter <member> festgelegt ist (s. untere Abb.), ihrem Farbschema entsprechend angezeigt. Da im <association>-Element beide Assoziationstypen aus Sicht der jeweiligen verbundenen Topics abgebildet sind, muss gewährleistet sein, dass der richtige Verbindungstyp erfasst wird. Dies geschieht mit der <xsl:if>-Anweisung, die überprüft, ob die zuvor in einer Variable gespeicherte ID des Topics mit der ID im <member>-Bereich identisch ist. Um diesen Sachverhalt besser nachvollziehen zu können, wird ein exemplarischer Ausschnitt einer XML-Topic Map des Begriffs „Biene“ aufgezeigt:

```
[...]
<association>
  <instanceOf>
    <topicRef xlink:href="http://www.semantic-network.de/xmlns/
      XTM/2005/2.0/sns-classes_2.0.xtm#narrowerTermAssoc"/>
    </instanceOf>
    <member>
      <roleSpec><topicRef xlink:href="http://www.semantic-network.de/xmlns/
        XTM/2005/2.0/sns-classes_2.0.xtm#widerTermMember"/></roleSpec> --> Oberbegriff - Verbindung
      <topicRef xlink:href="uba_thes_600381"/> --> ID des äußeren Topics "Hautflügler" (Oberbegriff von "Biene")
    </member>
    <member>
      <roleSpec><topicRef xlink:href="http://www.semantic-network.de/xmlns/
        XTM/2005/2.0/sns-classes_2.0.xtm#narrowerTermMember"/></roleSpec> --> Unterbegriff - Verbindung
      <topicRef xlink:href="uba_thes_4911"/> --> ID des Basistopics "Biene" (Unterbegriff von "Hautflügler")
    </member>
  </association>
[...]
```

Abbildung 38: Topic Map Auszug (getPSI) mit Erklärungen

Nach der Zeichnung aller Kanten wird der mittlere Topic abgebildet.

```
<!-- 2. Durchlauf für die Anzeige der Topics -->

<xsl:for-each select="tmf:topicMapFragment/tm:topicMap/tm:topic">

[ Errechnung des Gradabstandes - siehe oben ]

<!-- Topic in der Mitte der Map -->
  <xsl:choose>

    <xsl:when test="$pos=1">

      <xsl:variable name="topicname">
        <xsl:variable name="string"> <!-- Einfaches Hochkomma
entfernen für Javascript-Funktion -->
        <xsl:value-of select="tm:baseName/tm:baseNameString" />
      </xsl:variable>
      <xsl:variable name="hochkomma">'</xsl:variable>
      <xsl:value-of select="translate($string, $hochkomma, ' `')"/>
    </xsl:variable>
    <xsl:variable name="thema">
      <xsl:variable name="string"> <!-- Einfaches Hochkomma
entfernen für Javascript-Funktion -->
      <xsl:value-of select="tm:occurrence/tm:resourceData" />
    </xsl:variable>
```

```

        <xsl:variable name="hochkomma">'</xsl:variable>
        <xsl:value-of select="translate($string, $hochkomma, ' `')"/>

    </xsl:variable>

    <xsl:variable name="extlink">
        <xsl:value-of
select="tm:occurrence/tm:resourceRef/@xlink:href" />
    </xsl:variable>

    <xsl:variable name="topictyp">http://www.semantic-
network.de/xmlns/XTM/2005/2.0/sns-classes_2.0.xtm#</xsl:variable>

    <!-- Variable "t_typ" für die Einblendung des Topic Typs -->

    <xsl:variable name="t_typ">
        <xsl:choose>
            <xsl:when
test="tm:instanceOf/tm:topicRef/@xlink:href=concat($topictyp,
'conferenceType')">
                <xsl:text>tt1</xsl:text>
            </xsl:when>
        </xsl:choose>
    </xsl:variable>

```

Hierbei wird zunächst der innere Topic anhand von `position()` ermittelt und gezeichnet. Die entsprechenden Informationen zu dem Topic wie Name, Typ und Beschreibung werden in Variablen gespeichert, um sie später in der Infobox darstellen zu können.

Da ein einfaches Hochkommata bei der Überlieferung an die Javascript-Funktion hinderlich ist, wird es durch ein Apostroph ersetzt.

```

    <!-- / Variable "t_typ" -->

<!-- Leere Bereiche für Beschreibung des mittleren Topic, wenn Mouse
sich nicht über einem anderen Topic befindet -->
<g
onmouseover="ShowTopicName('{ $topicname }');ShowThema('{ $thema }');ShowT
opicTyp('{ $t_typ }');ShowExtLink('{ $extlink }');"
onmousemove="ShowTopicName('{ $topicname }');ShowThema('{ $thema }');ShowT
opicTyp('{ $t_typ }');ShowExtLink('{ $extlink }');"
onmouseout="HideTopicTyp('{ $t_typ }');">

<rect x="295" y="-340" width="400" height="660" style="opacity:0.0" />
<circle cx="0" cy="0" r="200" style="opacity:0.0" />
</g>

<!-- Link zu externer Ressource -->
    <a target="_blank">
        <xsl:attribute name="xlink:href">
            <xsl:value-of select="tm:occurrence/tm:resourceRef/@xlink:href" />
        </xsl:attribute>
        <text id="ext_link" x="360" y="160" style="visibility: hidden">dyn.
Text</text>
    </a>

```

In dem oberen Abschnitt werden innerhalb des Containerelements `<g>` alle Attribute festgelegt, die in den in ihm enthaltenen Bereichen Gültigkeit besitzen. Die Bereiche sind in diesem Fall ein durchsichtiges Rechteck, dass über der Infobox liegt und ein Kreis dessen Mittelpunkt der mittlere Knoten ist und dessen Radius so lang ist wie eine

Kante, d.h. er kommt nicht in Berührung mit den äußeren Knoten. Wenn der Mauszeiger über diesen Bereichen liegt, wird der Inhalt des mittleren Knotens in der Infobox ausgegeben mit Hilfe der Javascript-Funktionen ShowTopicName, ShowThema, ShowTopicTyp und ShowExtLink (s. auch Kap. 7.7). Außerdem kann der Nutzer einen evtl. vorhandenen externen Link auch über die Infobox aktivieren.

```
<g id="g{$pos}"
onmouseover="ShowTopicName('{ $topicname}');ShowThema('{ $thema}');ShowExtLink('{ $extlink}');SetScale('g{$pos}',1.4);ShowTopicTyp('{ $t_typ}');"
onclick="ExtLink('{ $extlink}')"
onmouseout="HideTopicName();HideThema();HideExtLink();SetScale('g{$pos}',1);HideTopicTyp('{ $t_typ}');">

<!-- Topic Typen Ausgabe (Alle Topic Typs sind entnommen aus der
Anwenderdokumentation von SNS) -->

<xsl:choose>

<!-- Ereignisse #f1454d | #ffffff -->

    <xsl:when
test="tm:instanceOf/tm:topicRef/@xlink:href=concat($topictyp,
'conferenceType')">
        <!-- Konferenz --> <circle cx="0" cy="0" r="50" style="fill:
#f1454d; stroke: #767D84; stroke-width: 6px" /><circle cx="0" cy="-30"
r="10" style="fill: none; stroke: #ffffff; stroke-width: 6px" /><path
d="M 0,-20 l 0,20 l -25,-25 m 50,0 l -25,25 l 0,15 l -20,20 m 40,0 l -
20,-20" style="fill: none; stroke: #ffffff; stroke-width: 6px"/>
        <text font-size="11" style="fill: #000000" text-anchor="middle"
x="0" y="0" transform="translate(0 4)"><xsl:value-of
select="tm:baseName/tm:baseNameString" /></text>
    </xsl:when>
[...]
```

Das sich im oberen Abschnitt öffnende <g>-Containerelement beinhaltet den Knoten in der Mitte der Topic Map. Zunächst bekommt dieser Basistopic eine eindeutige ID zugeteilt, mit Hilfe seiner eindeutigen Position im XML-Dokument. Diese ID ist notwendig, damit die Javascript-Funktionen auf dieses eindeutig gekennzeichnete Element zugreifen können.

Die äußeren Topics sind in dem <otherwise> Element eingeschlossen, da sie nicht die 1.Position in der Topic Map haben und somit nicht in der Mitte dargestellt werden.

```
</xsl:choose>
</g>
</xsl:when>
<!-- Äußere Topics -->
<xsl:otherwise>
[ Speichern der Topic - Informationen (Name, Topic Typ, Occurrences) ]
[ und anschliessende Topic Typ Ermittlung s.o. ]
[...]
```

Es folgt die Darstellung der äußeren Topics. An dieser Stelle möchte der Autor auf seine Lösung des Problems der richtigen Positionierung der äußeren Knoten aufmerksam machen.

```
<g id="g{$pos}"
onmouseover="ShowTopicName('{ $topicname}');ShowThema('{ $thema}');ShowExtLink('{ $extlink}');SetTranslate('g{$pos}', '{ $sum}',0,265,1.4);ShowTopicTyp('{ $t_typ}');"
onclick="ExtLink('{ $extlink}')">
```

```
onmouseout="HideTopicName();HideThema();HideExtLink();SetTranslate('g{
$pos}','{$sum}',0,245,1);HideTopicTyp('{$t_typ}');"
transform="rotate({$sum}) translate(0 245) rotate(-{$sum})" >
[...]
```

Der interessante Teil ist im vorherigen Abschnitt fett hervorgehoben und zeigt auf, wie die ordnungsgemäße Positionierung der äußeren Knoten funktioniert.

Es werden nicht etwa die Zielkoordinaten ausgerechnet, wo der äußere Topic positioniert werden soll, sondern das ganze Koordinatensystem wird gedreht, bevor die Verschiebung von 245 Bildpunkten, ausgehend vom Mittelpunkt der Topic Map, stattfindet.

Dieses Verfahren erspart dem System größere Rechenprozeduren, die bei einer Errechnung von genauen Zielkoordinaten die Folge wäre.

Die folgende Abbildung zeigt die einzelnen Schritte der Transformation auf:

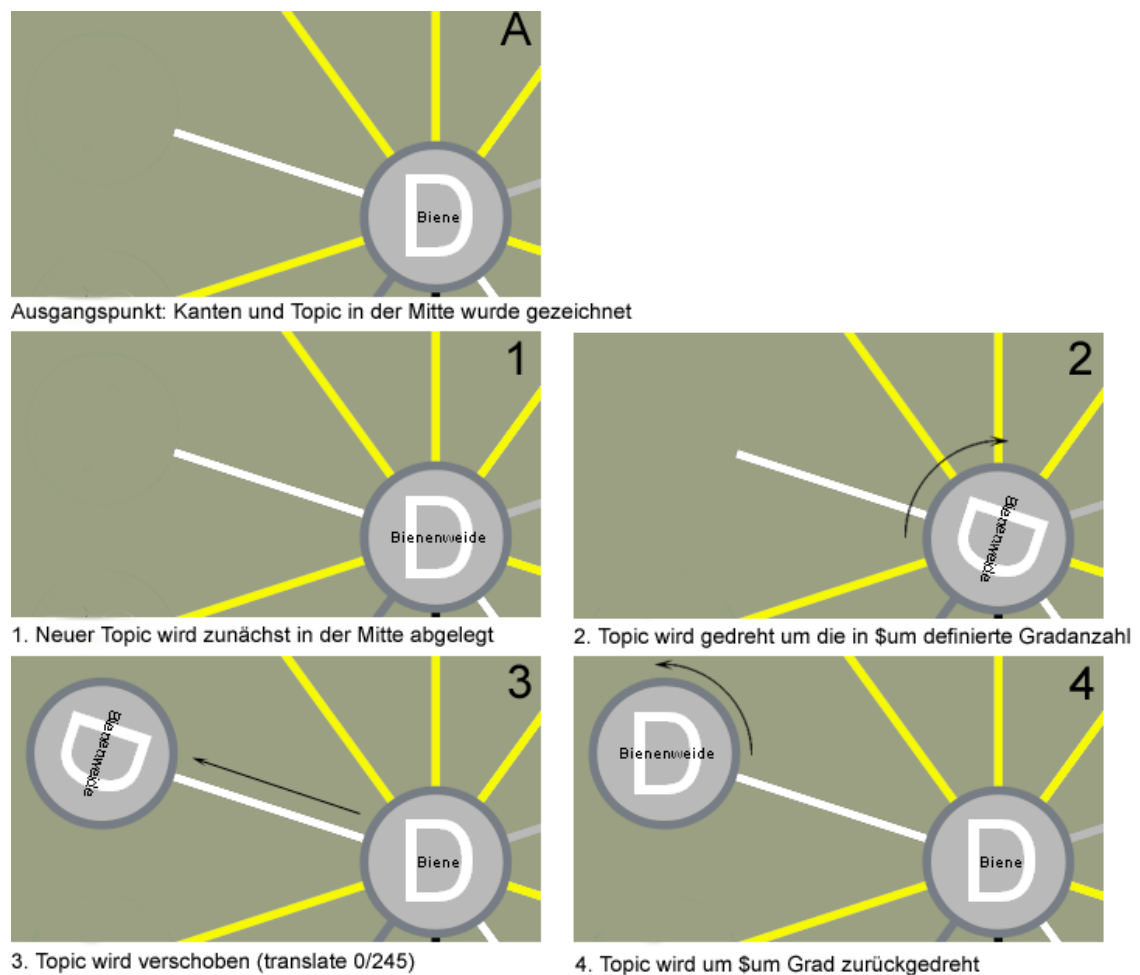


Abbildung 39: Einzelne Schritte der Positionierung der äußeren Topics

Die Knoten werden, nach dem sie an der richtigen Stelle platziert worden sind (mit Hilfe einer Drehung um \$sum Grad und einer anschließenden Verschiebung des Objekts um 245 Punkten), mittels `rotate(-{$sum})` wieder „zurückgedreht“ um eine waagerech-

te Positionierung zu gewährleisten. Wie an diesem Beispiel deutlich wird, ist das Koordinatensystem in SVG nicht fest fixiert.

Im nächsten Abschnitt des XML-Stylesheets werden die Bereiche definiert, an denen die Textausgabe erfolgen soll. Diese Bereiche müssen ebenfalls mit einer eindeutigen ID gekennzeichnet sein, damit Javascript die Ausgabe sinnvoll steuern kann.

```

        </xsl:choose>
      </g>
    </a>
    </xsl:otherwise>

  </xsl:choose>
</xsl:for-each>

<!-- Dynamischer Inhalt von Infobox -->

  <text id="topic_name" x="360" y="-180" style="visibility:
hidden">dyn. Text</text>

  <text id="thema1" x="360" y="-130" style="visibility: hidden">
dyn.Text
  <tspan id="thema2" dy="12" x="360" style="visibility: hidden">dyn.
Text</tspan>
  <tspan id="thema3" dy="12" x="360" style="visibility: hidden">dyn.
Text</tspan>
  <tspan id="thema4" dy="12" x="360" style="visibility: hidden">dyn.
Text</tspan>
[...]
```

Texte können in einem Vektorgrafikformat wie SVG naturgemäß nicht sehr gut formatiert werden, z.B. sind automatische Zeilenumbrüche nur schwer umsetzbar. Aus diesem Grund hat sich der Autor dazu entschlossen, die Beschreibungstexte der Topics in einzelne Teilstings zu zerlegen. Die entsprechende Funktion ist in dem referenzierten Javascript-Code enthalten (s. Kap. 7.7.)

Der letzte Teil des Stylesheets platziert die entsprechenden Topic Typen – Icons, die auch in der Infobox erscheinen an der dafür festgelegten Stelle.

```

  <text id="topic_entry" x="260" y="-100" style="visibility:
hidden">dyn. Text</text>

  <!-- Topic Types -->

    <g id="tt1" transform="translate(375 -245) scale(0.3 0.3)"
style="visibility: hidden">
      <!-- Konferenz -->
      <circle cx="0" cy="0" r="50" style="fill: #f1454d; stroke: #767D84;
stroke-width: 6px" /><circle cx="0" cy="-30" r="10" style="fill: none;
stroke: #ffffff; stroke-width: 6px" /><path d="M 0,-20 l 0,20 l -25,-
25 m 50,0 l -25,25 l 0,15 l -20,20 m 40,0 l -20,-20" style="fill:
none; stroke: #ffffff; stroke-width: 6px"/>
      <rect x="70" y="-40" width="1000" height="100" style="fill: #516134;"
/>
      <text text-anchor="right" x="0" y="0" font-size="45" fill="#f1454d"
transform="translate(75 18)">= Ereignis -> Konferenz</text>
    </g>
  [...]
  <!-- / Topic Types -->
<!-- / Dynamischer Inhalt von Infobox -->
```



```

function ShowTopicName(txt)
{
    var elem;

    elem=svgdoc.getElementById("topic_name");
    elem.childNodes.item(0).data=txt;

    elem.setAttribute("style","fill: #ffffff; font-size: 14px;
visibility: visible");

    // Fix fuer Corel SVG Viewer - nochmaliges Zuweisen des
    Textinhalts
    elem.childNodes.item(0).data=txt;
}

function HideTopicName()
{
    var elem;
    elem=svgdoc.getElementById("topic_name");
    elem.setAttribute("style","visibility: hidden");
}

```

Die Funktion ShowTopicName erwartet einen Parameter txt, der in dem XML-Stylesheet aus dem ursprünglichen Dokument ausgelesen wurde. Es wird der Bereich aufgespürt (ID: topic_name), der mit dem Text gefüllt werden soll. Mit dem SVG Stilattribut visibility wird der Text sichtbar gemacht. Sobald der Nutzer seinen Mauszeiger von dem Topic wegbewegt wird die Funktion HideTopicName() aktiv und der Titel des entsprechenden Topics verschwindet. Das gleiche Prinzip gilt auch für die folgenden Funktionen:

```

function ShowExtLink(txt)
{
    var elem;

    elem=svgdoc.getElementById("ext_link");
    elem.childNodes.item(0).data=txt;

    elem.setAttribute("style","fill: #ffffff; font-size: 12px;
visibility: visible");

    // Fix fuer Corel SVG Viewer - nochmaliges Zuweisen des
    Textinhalts
    elem.childNodes.item(0).data=txt;
}

function HideExtLink()
{
    var elem;
    elem=svgdoc.getElementById("ext_link");
    elem.setAttribute("style","visibility: hidden");
}

function ShowTopicTyp(txt)
{
    var elem;

    elem=svgdoc.getElementById(txt);

    elem.setAttribute("style","visibility: visible");
}

```

```

}

function HideTopicTyp(txt)
{
    var elem;
    elem=svgdoc.getElementById(txt);
    elem.setAttribute("style","visibility: hidden");
}

```

Bei der folgenden Funktion `ShowThema(txt)` wird der eintreffende Text in Teilstrings zerlegt und an der entsprechenden Stelle im Zieldokument ausgegeben.

```

function ShowThema(txt)
{
    var elem;
    var TextLaenge = txt.length;

    if (TextLaenge > 60) {
        txt1 = txt.substring(0, 60);
        txt2 = txt.substring(60, 120);
[...]
        txt19 = txt.substring(1080, 1140);
        txt20 = txt.substring(1140, 1200);

        elem1=svgdoc.getElementById("thema1");
        elem1.childNodes.item(0).data=txt1;
        elem2=svgdoc.getElementById("thema2");
        elem2.childNodes.item(0).data=txt2;
[...]
        elem19=svgdoc.getElementById("thema19");
        elem19.childNodes.item(0).data=txt19;
        elem20=svgdoc.getElementById("thema20");
        elem20.childNodes.item(0).data=txt20;
        elem1.setAttribute("style","fill: #ffffff; font-size: 12px;
visibility: visible");
        elem2.setAttribute("style","fill: #ffffff; font-size: 12px;
visibility: visible");
[...]
        elem19.setAttribute("style","fill: #ffffff; font-size: 12px;
visibility: visible");
        elem20.setAttribute("style","fill: #ffffff; font-size: 12px;
visibility: visible");
    }

    else {

        elem=svgdoc.getElementById("thema1");
        elem.childNodes.item(0).data=txt;

        elem.setAttribute("style","fill: #ffffff; font-size: 12px;
visibility: visible");

        // Fix fuer Corel SVG Viewer - nochmaliges Zuweisen des
        Textinhalts
        elem.childNodes.item(0).data=txt;
    } }

function HideThema()
{

```

```

var elem;
elem1=svgdoc.getElementById("thema1");
elem1.setAttribute("style","visibility: hidden");
elem2=svgdoc.getElementById("thema2");
[...]
elem19=svgdoc.getElementById("thema19");
elem19.setAttribute("style","visibility: hidden");
elem20=svgdoc.getElementById("thema20");
elem20.setAttribute("style","visibility: hidden");
}

```

Die folgenden Funktionen sind verantwortlich für die Anzeigebereichsänderung, also die Größenveränderung des Topics, wenn der Nutzer ihn mit dem Mauszeiger überfährt.

```

function SetTranslate(evt, rotate, dx, dy, scale)
{
    obj=svgdoc.getElementById(evt);

    var y;
    y = "0";

    var rotateback;
    rotateback ="-"+rotate;

    obj.setAttribute("transform","translate("+dx+", "+y+")
rotate("+rotate+") translate("+dx+", "+dy+") rotate("+rotateback+")
scale("+scale+") ");
}

function SetScale(evt, scale)
{
    obj=svgdoc.getElementById(evt);

    obj.setAttribute("transform","scale("+scale+)");
}

```

Die letzte Funktion in dem Javascript-Dokument öffnet das Dialogfenster, wenn zu dem angeklickten Topic ein externer Link vorhanden ist.

```

function ExtLink(extlink)
{
    if (extlink == "") {
        // Kein externer Link vorhanden
    } else {
        Check = confirm("Zu dem ausgewaehlten Topic ist ein externer Link
vorhanden. Soll dieser geoeffnet werden?");
        if (Check == true) {
            F1 = window.open(extlink, extlink);
            // top.location=extlink;

        }
        if (Check == false) { }
    }
}

```

8 Ergebnis

Wie die Webapplikation in der Praxis eingesetzt werden kann, wird anhand von drei Szenarien mit verschiedenen fiktiven Benutzern aufgezeigt.

1. Szenario:

Ein Nutzer namens Herr X tippt die URL **<http://www.sns-navigator.de>** in seinem Browser ein. Er verwendet den Firefox V1.5 von Mozilla.

Es erscheint zunächst die Startseite:

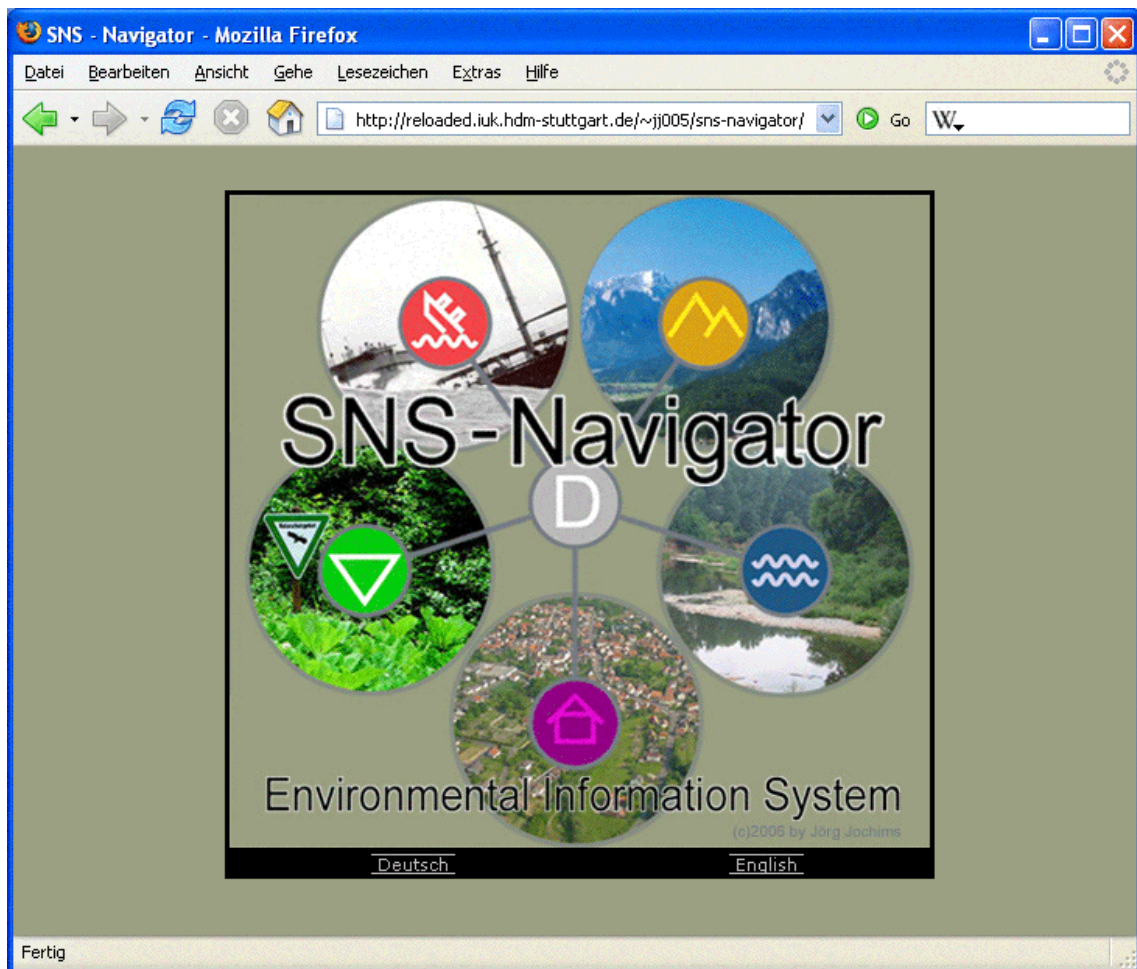


Abbildung 41: Szenario 1: Startseite

Der Nutzer wählt die deutsche Version aus und gelangt auf die Willkommenseite `index_de.php` des SNS-Navigators.

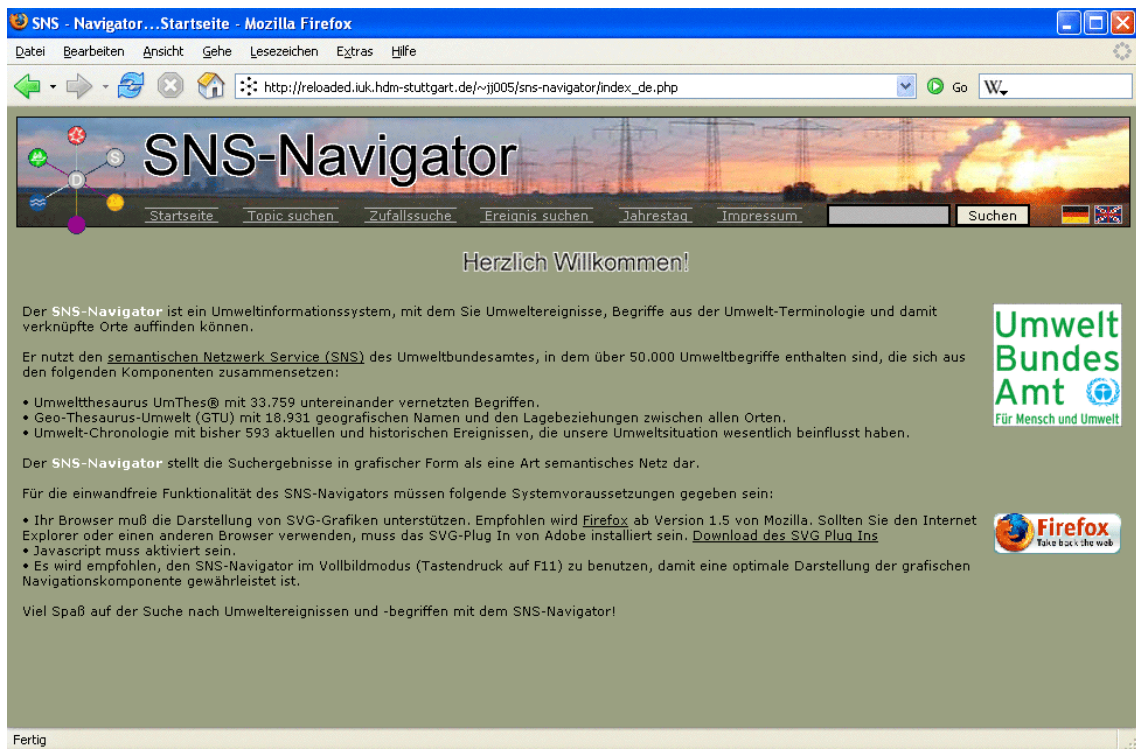


Abbildung 42: Szenario 1: Willkommensseite

Herr X stellt befriedigt fest, dass er die erforderlichen Systemvoraussetzungen erfüllt, da er den Firefox V1.5 benutzt, der über eine native SVG-Unterstützung verfügt.

Er wechselt daraufhin in den empfohlenen Vollbildmodus und wählt den Link „Ereignis suchen“ aus, da er ein ausgeprägtes Interesse an Umweltereignissen hat, die in der letzten Zeit stattgefunden haben.



Abbildung 43: Szenario 1: Umweltereignis Suchformular

Der Nutzer gibt den Zeitraum vom 1.1.2005 bis 31.7.2006 an. Er vertippt sich aber bei der zweiten Jahreszahl und schreibt „200p“ statt „2006“.



Abbildung 44: Szenario 1: Fehlermeldung

Das System generiert die Fehlermeldung „Jahreszahl eingeben“ und der Nutzer korrigiert daraufhin seine Eingabe und startet die Suche neu.

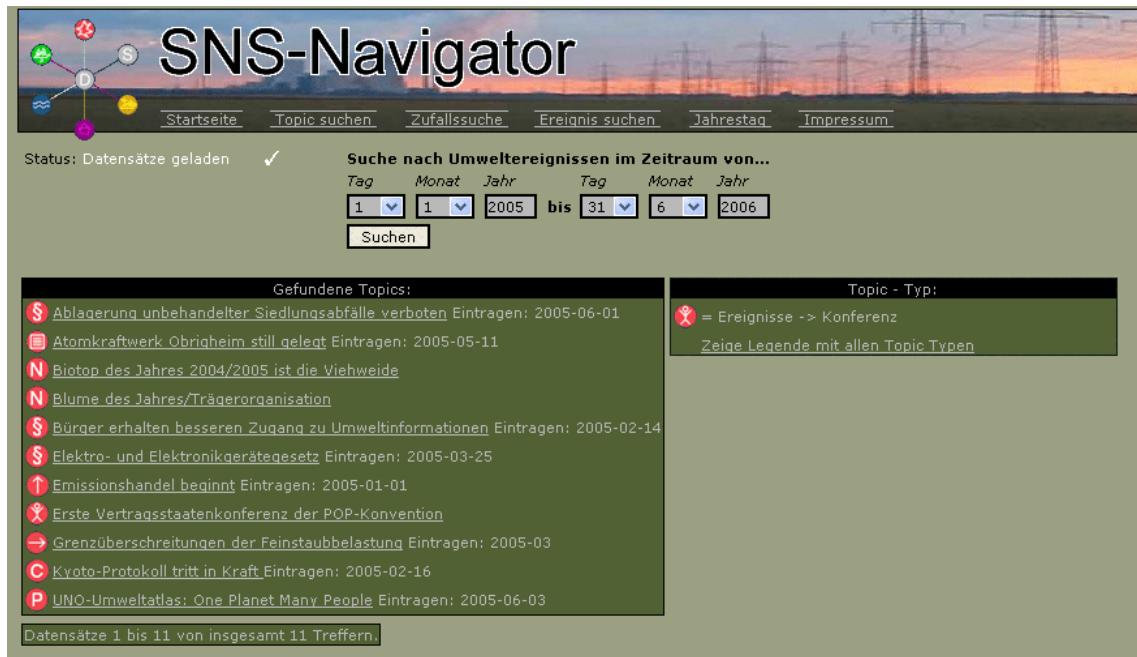


Abbildung 45: Szenario 1: Listenanzeige

Der Nutzer wird sofort auf den Eintrag „Atomkraftwerk Obrigheim still gelegt“ aufmerksam und klickt darauf.

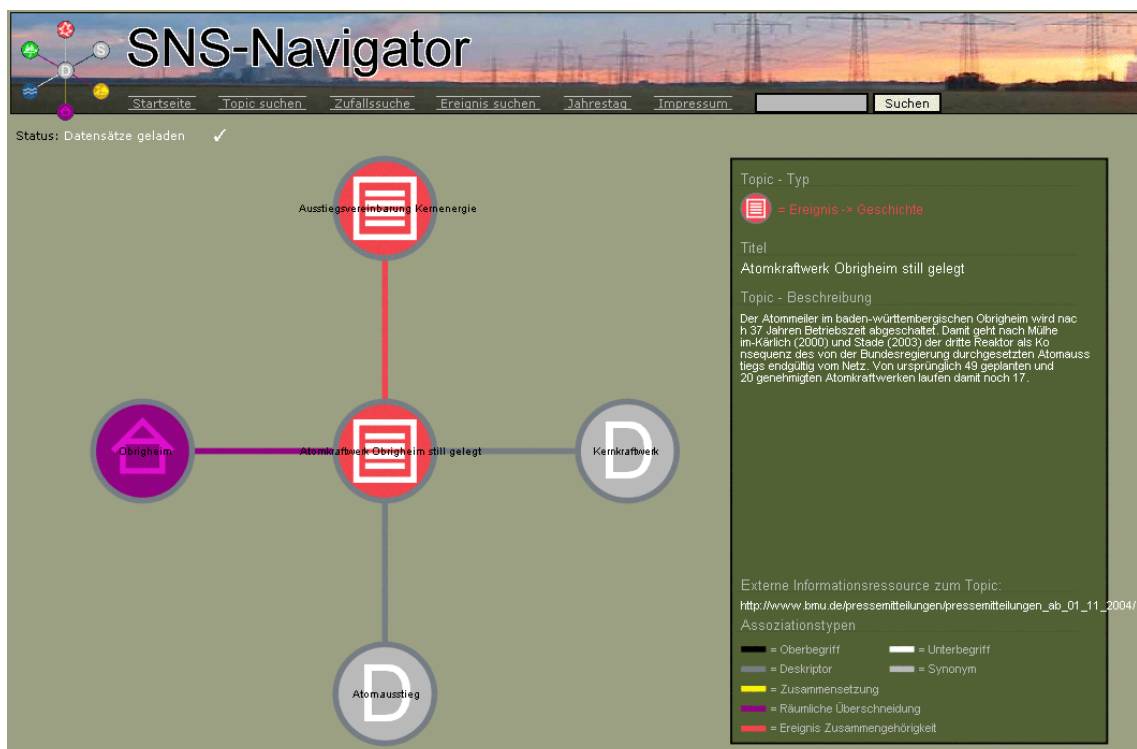


Abbildung 46: Szenario 1: Grafische Navigationskomponente Bsp. 1

Der Nutzer ist nun sehr interessiert an dem mit dem Basistopic verknüpften Ereignis namens „Ausstiegsvereinbarung Kernenergie“ und überfährt es mit seiner Maus.



Abbildung 47: Szenario 1: Auswahl eines äußeren Topics

Das Icon vergrößert sich und Herr X kann die entsprechende Beschreibung des Topics der Infobox auf der rechten Seite entnehmen. Er klickt nun mit der Maus auf den Topic.

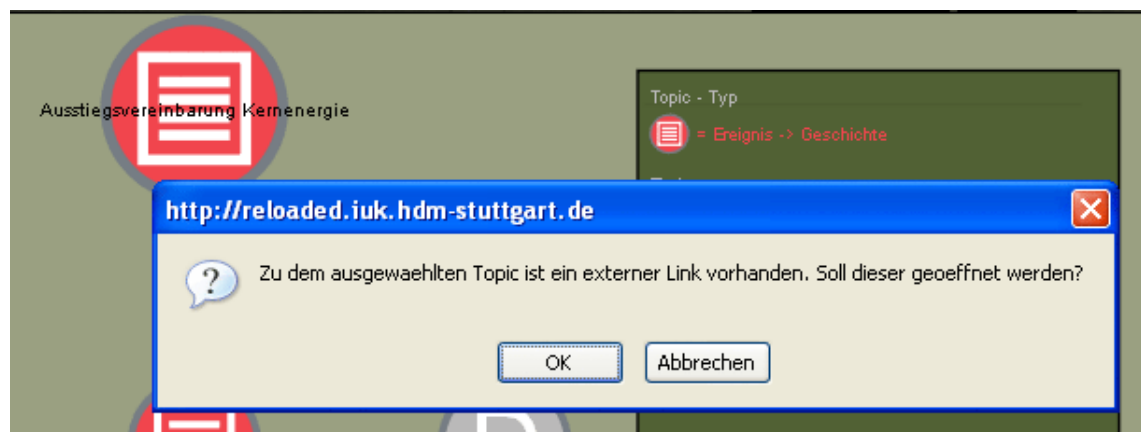


Abbildung 48: Szenario 1: Auswahldialog für externe Links

Es erscheint ein Dialogfenster. Der Nutzer klickt auf „OK“.



Abbildung 49: Szenario 1: Externe Informationsressource

In einem neuen Browserfenster wird die entsprechende externe Informationsressource aufgezeigt und der zuvor ausgewählte Topic befindet sich in der Mitte der neu generierten Topic Map.

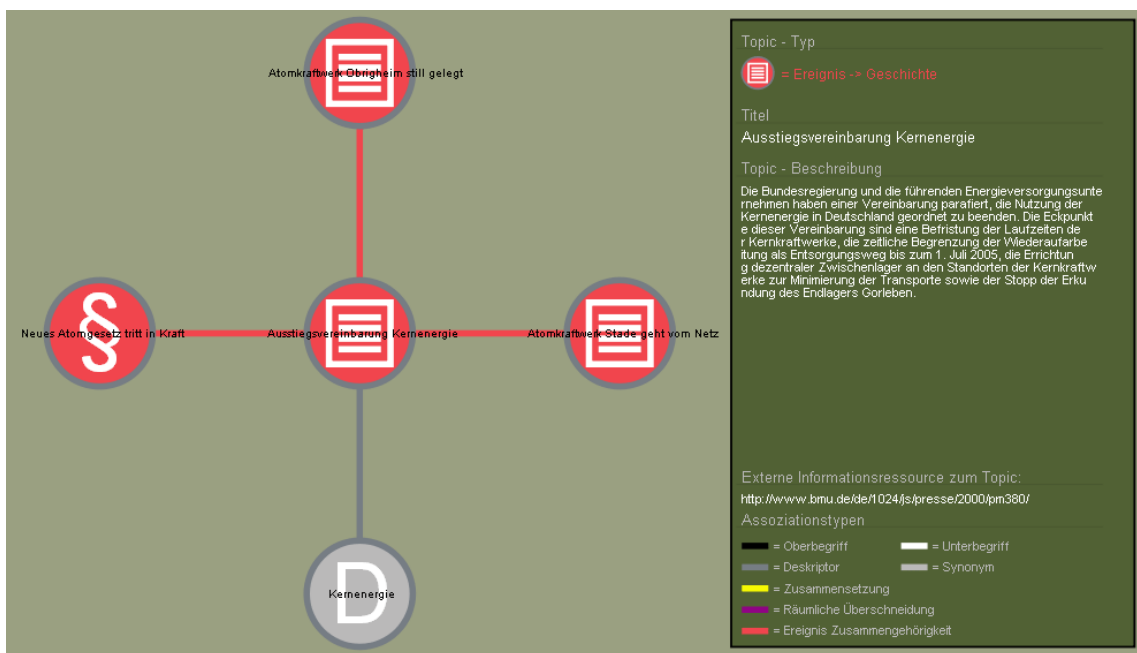


Abbildung 50: Szenario 1: Grafische Navigationskomponente Bsp. 2

Herr X interessiert sich auch für das Ereignis „Grenzüberschreitungen der Feinstaubbelastung“ (das er in der Liste gesehen hat), denn er möchte herausfinden, in welchen Städten, außerhalb seiner Heimatstadt Stuttgart, ebenfalls noch Grenzüberschreitungen aufgetreten sind.

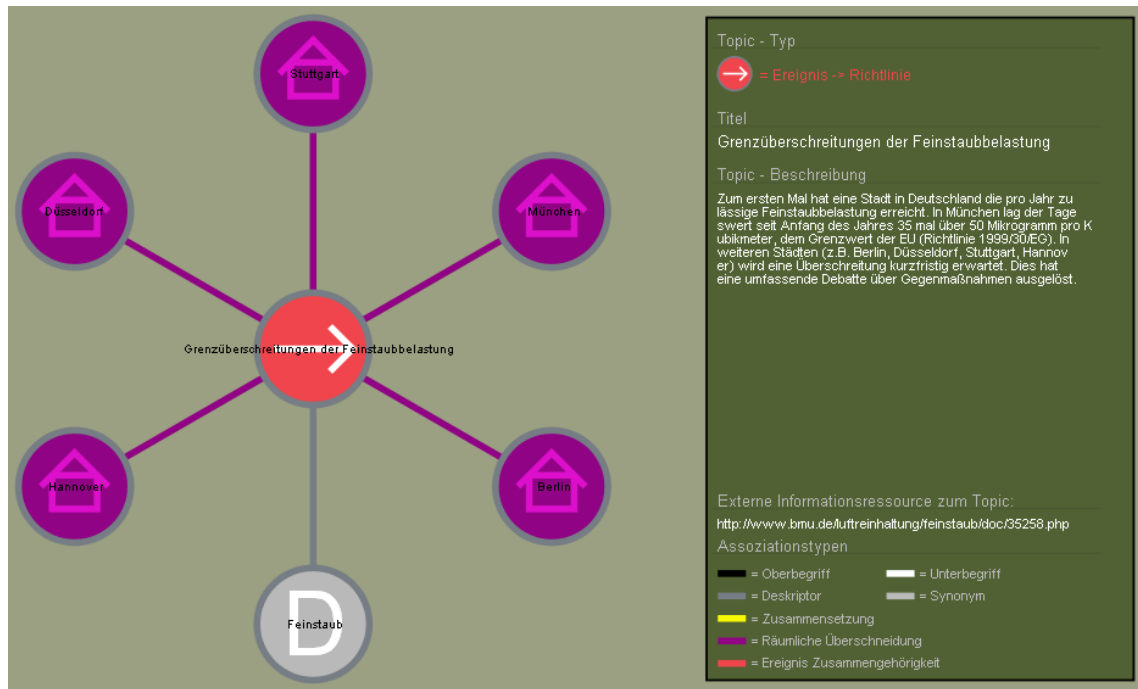


Abbildung 51: Szenario 1: Grafische Navigationskomponente Bsp.3

2. Szenario:

Ein anderer Nutzer namens Frau Y benutzt den Internet Explorer mit dem SVG Plug In von Adobe und hofft mittels des SNS-Navigators herauszufinden, was man unter „Pops“ versteht. Sie meint, diesen Begriff schon mal im Zusammenhang mit einer Umweltgefährdung gehört zu haben. Da sie es eilig hat, gibt sie den Begriff direkt auf der Willkommenseite in dem oberen Schnellsuchformular ein und erhält folgende Liste:

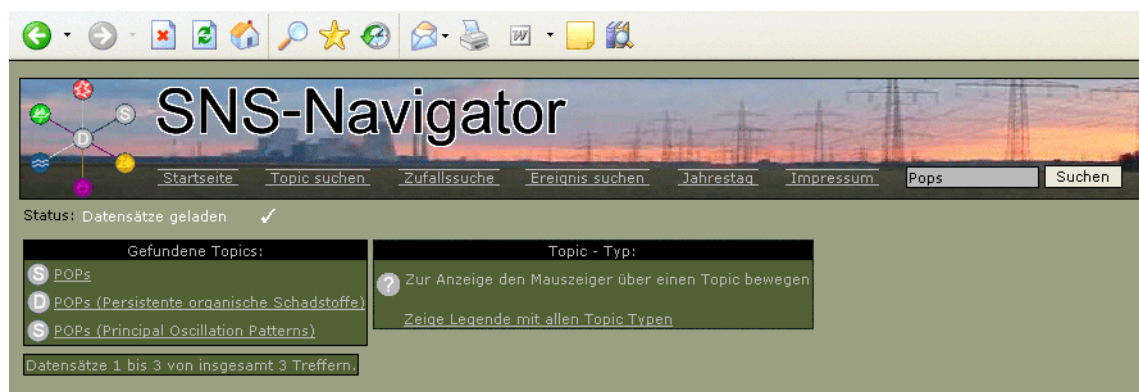


Abbildung 52: Szenario 2: Listenanzeige

Frau Y wählt daraufhin den zweiten Eintrag „POPs (Persistente organische Schadstoffe)“ in der Liste aus.

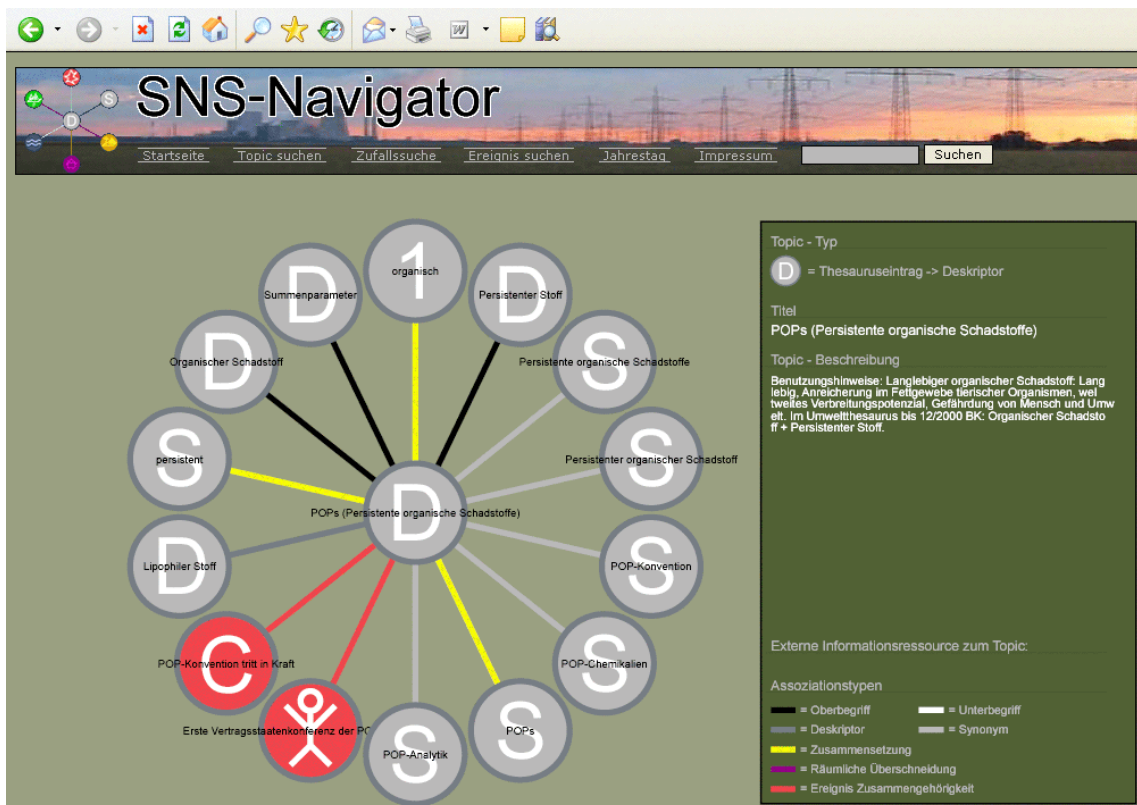


Abbildung 53: Szenario 2: Grafische Navigationskomponente Bsp.4

Dank der Beschreibung des Begriffs weiß Frau Y jetzt, um was es sich bei „Pops“ handelt.

3. Szenario:

Ein Schüler namens „Z“ gerät durch Zufall beim Surfen auf die Seite des SNS-Navigators und erfasst dessen Sinn zuerst nicht vollständig. Er klickt aus diesem Grund erstmal auf „Zufallssuche“ um zu sehen, was daraufhin passiert.

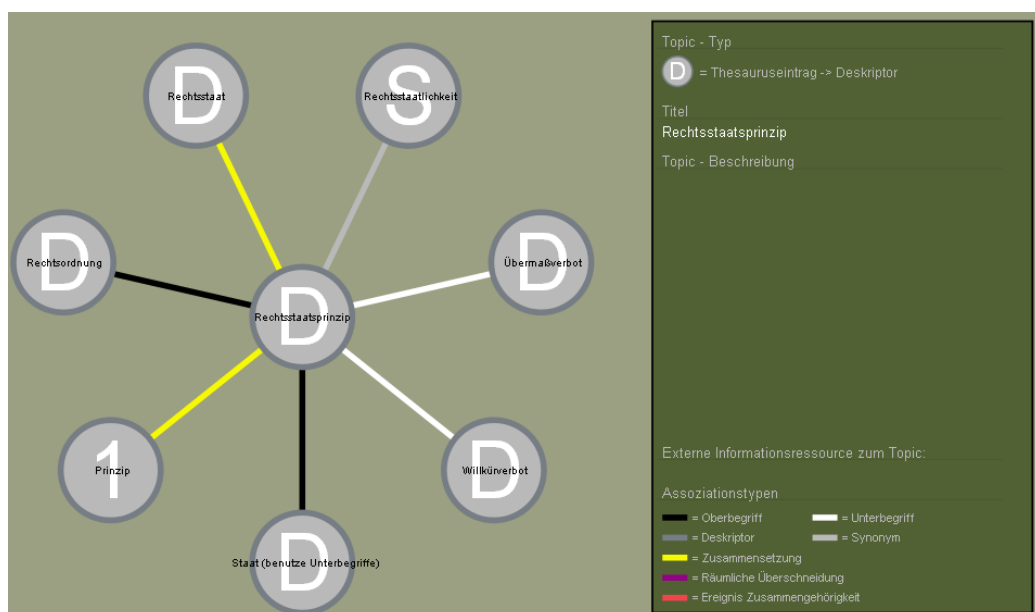


Abbildung 54: Szenario 3: Zufällig ausgewählte Topic Map

Der Schüler fängt nun an sich auf spielerische Art und Weise mit dem System zu beschäftigen, indem er die Navigation ausprobiert und sich durch mehrere Topic Maps „durchklickt“. Plötzlich fällt ihm ein, dass sein Biologielehrer ihm heute aufgetragen hat, herauszufinden, zu welcher Sorte ein Baum namens Esche gehört. Er gibt also „Esche“ in dem kleinen Suchfeld über der grafischen Navigationskomponente ein und wählt aus der Trefferliste den Eintrag „Esche“ aus. Folgende Anzeige erscheint:

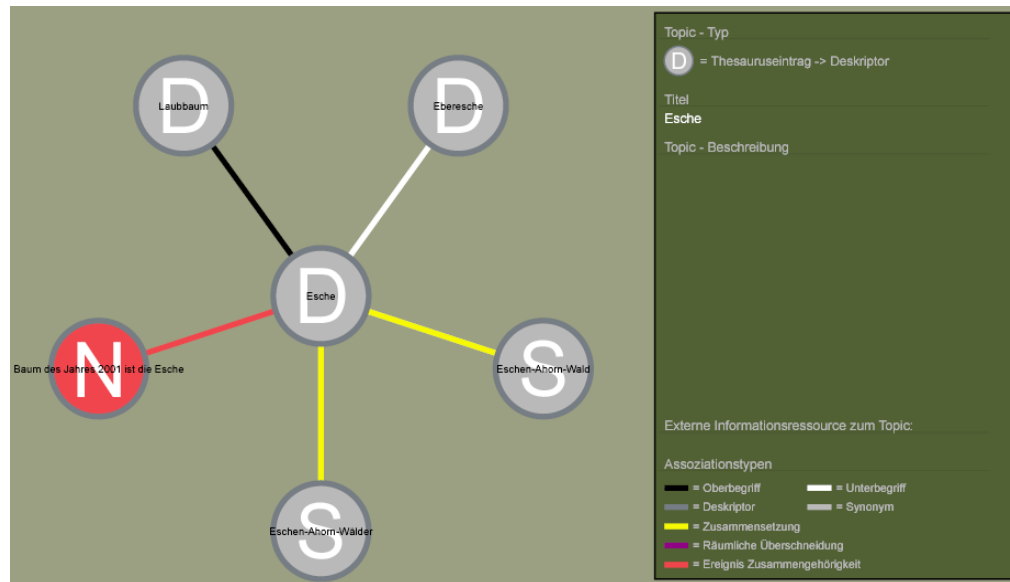


Abbildung 55: Szenario 3: Grafische Navigationskomponente Bsp.5

Der Schüler freut sich, da er nun weiß, dass die Esche ein Laubbaum ist. Gespannt klickt er auf den markanten roten Topic mit der Bezeichnung „Baum des Jahres 2001 ist die Esche“.

Er entscheidet sich in dem Auswahldialog dafür, die externe Informationsressource zu öffnen und erhält folgende Seite in einem neuen Browserfenster:



Abbildung 56: Szenario 3: Externe Informationsressource

Voller Stolz druckt der Schüler die Seite aus und beeindruckt seinen Lehrer am nächsten Tag mit seinen neu erworbenen Botanik-Kenntnissen.

9 Zusammenfassung

Nach einem einleitenden Überblick über die heutige Situation der Informationsrecherche im Internet, erläuterte diese Arbeit die Geschichte des semantischen Netzwerk Service – kurz SNS –, auf dessen Basis die in dieser Arbeit vorgestellte Webapplikation namens „SNS-Navigator“ entwickelt worden ist.

Als Ziele dieser Anwendung wurde eine graphische Visualisierung der Umweltdaten in Form eines semantischen Netzes, eine Navigation innerhalb der Suchergebnisse, einfache Benutzerführung sowie Interaktivität und Übersichtlichkeit festgelegt.

Die Darstellung der Technologien gab zunächst einen Überblick über das Feld von existierenden Technologien und Standards, die für das Verständnis der hier vorgestellten Webanwendung notwendig waren.

Hierunter fielen an erster Stelle die Web Services, die zur Zeit „State of the art“ sind, wenn es um das Zusammenspiel verteilter Anwendungen in einer heterogenen Systemlandschaft (wie z.B. dem Internet) geht. Sie sollen zusammen mit dem Semantic Web eine synergetische Einheit bilden, um die Informationsrecherche, die Aufbereitung der Suchergebnisse und schlussendlich die Entwicklung des ganzen World Wide Web voranzutreiben und mit mehr Bedeutungsinhalten zu füllen.

Auf semantische Netzen und den zugehörigen Topic Maps lag der Schwerpunkt bei den Technologien, denn der SNS-Navigator wertet die XML-Topic Maps des SNS aus und generiert auf deren Basis eine grafische Navigationskomponente, die für sich genommen eine Art semantisches Netz darstellt.

Das Alleinstellungsmerkmal der Topic Maps im Gegensatz zu anderen Formen der Wissensrepräsentation begründet sich durch die Assoziationen zu anderen Topics, womit semantische Bezüge nicht nur innerhalb der jeweiligen Kategorie möglich sind. Dies ist aus Sicht des Autors der entscheidende Faktor für die Mächtigkeit von Topic Maps.

Im Anschluss wurde auf die Kommunikationsaspekte bei Web Services eingegangen, insbesondere auf Kommunikationsstandards, wie z.B. SOAP und GET/POST-Bindings.

Die Darstellung der Struktur der XML-Definitionen von SNS zeigte ein XML-Regelwerk auf, welches das Zusammenspiel und die Struktur der Web Services und der Topic Maps definiert.

Im Weiteren beschrieb die Arbeit die Anforderungen an die Webapplikation aus Sicht der Nutzer anhand von Use Case-Diagrammen und die verschiedenen technischen Lösungen einer Implementierung, sowie der vom Autor eingeschlagene Lösungsweg.

Der Hauptteil dieser Arbeit zeigte die Darstellung der Konzeption und Realisierung des SNS-Navigators auf. Hier waren aufschlussreiche, technische und gestalterische Lösungen zu finden, mit deren Hilfe der Autor diese Anwendung umgesetzt hat. Einen

Schwerpunkt legte der Autor hierbei auf die Präsentation und Erläuterung des Aufbaus der XML-Stylesheets, in denen viele funktionale und gestalterische Aspekte der Webapplikation enthalten sind.

Eine Besonderheit dieser Arbeit stellt die Anzeige der grafischen Navigationskomponente mit dem Vektorgrafikformat SVG dar, da dieses Format im WWW noch nicht sehr häufig anzutreffen ist und der Autor einen kleinen Beitrag dazu leisten wollte, dieses Grafikformat etwas bekannter zu machen. Darüber hinaus bot sich dieses Format vorzüglich an, da es ebenso wie der Topic Map Standard auf XML basiert, ähnlich wie HTML eine Verlinkung unterstützt und mit Javascript kombiniert werden kann für eine interaktive Belegung der grafischen Navigationskomponente. Gerade in Verbindung mit dem XML Topic Map Standard, dessen Vorzüge vielleicht sonst nicht so leicht sichtbar wären, kann SVG seine Stärken ausspielen.

Anhand von drei Szenarien wurde unter „Ergebnis“ der mögliche praktische Einsatz der Anwendung erläutert.

Ein Anliegen der Arbeit war es mit dem SNS-Navigator auch einen nichtfachkundigen Nutzerkreis anzusprechen und bei diesem ein Interesse am Auffinden und Suchen von Begriffen aus dem Umweltthesaurus zu wecken. Um das zu erreichen, wird die vom Autor entwickelte „Zufallssuche“ angeboten. Mit dieser Suchfunktion kann auch ohne Angabe eines Suchbegriffs ein Topic aus dem Umweltthesaurus als Topic Map dargestellt werden und vermittelt dem unerfahrenen Nutzer somit einen ersten Eindruck von der Datenbasis des Umweltthesaurus und dessen Visualisierung. Diese Funktion soll der Inspiration des Nutzers für seine weiteren Suchanfragen dienen. Insbesondere Topic Maps besitzen den Reiz, dass interessante Informationen durch ein „unbedarftes Stöbern“ seitens des Nutzers in den Suchergebnissen aufgefunden werden können.

Die in dieser Arbeit vorgestellte Webapplikation namens „SNS-Navigator“ ist unter der URL „<http://www.sns-navigator.de>“ im Internet erreichbar und stellt zusammenfassend einen Informationsdienst für Umweltdaten dar. Sie verfügt über verschiedene Suchmöglichkeiten, wie eine Topic-Suche für erfahrene Rechercheure mit entsprechenden Zusatzeinstellungen, eine Schnellsuche, eine gezielte Suche nach Umweltereignissen in einem gewissen Zeitraum, die zuvor beschriebene Zufallssuche, bei der wahllos ein Begriff aus dem Umweltthesaurus des Umweltbundesamtes herausgegriffen wird und die Suche nach Jahrestag-Ereignissen. Zusätzlich wird die Webapplikation zweisprachig in deutscher und englischer Fassung angeboten. Mit Hilfe der vom Autor selbst entworfenen Farblegenden und Icons kann der Informationsinhalt vom Konsument schneller erfasst werden.

Anhang: Datenträger

Die beigefügte CD Rom enthält:

- Diese Diplomarbeit als PDF-Datei
- Die Anwenderdokumentation V 1.5. (12.6.06) von SNS, auf deren Grundlage die Webapplikation entwickelt worden ist
- Den kompletten Quellcode aller Skripte und Komponenten der Anwendung
- Die wichtigsten XML-Dokumente von SNS (XML-Schema-Definitionen, WSDL-Datei und Topic Map-Beispiele)
- Eine ReadMe-Textdatei, die den Inhalt der CD kurz erläutert.

Quellenverzeichnis

American Association for Artificial Intelligence (2002): Brief History of Artificial Intelligence. <http://www.aaai.org/AITopics/bbhist.html> (Datum des Zugriffs: 6. Mai 2006).

Bader, H. (2004): SVG Reporting – Vektorgrafiken im Web einsetzen. Frankfurt: Software & Support Verlag GmbH, 2004

Bandholtz, T. (2003): Erstellung eines semantischen Netzwerkdienstes (SNS) für das Umweltinformationsnetz Deutschland – German Environmental Information Network (gein®) – Band 1. SchlumbergerSema GmbH, Köln. Im Auftrag des Umweltbundesamtes. <http://www.semantic-network.de/sns-AB-2003-06-06.pdf> (Datum des Zugriffs: 10. März 2006).

Bayer, T. (2003): Apache Axis - Architektur und Erweiterbarkeit. <http://www.oio.de/public/xml/axis-extensibility.htm> (Datum des Zugriffs: 15. Mai 2006)

Berners-Lee, T. / Hendler, J. / Lassila, O. (2001): The Semantic Web. Scientific America 2001

http://www.sciamdigital.com/index.cfm?fa=Products.ViewIssuePreview&ARTICLEID_C HAR=5704DA3F-F381-40DD-ABB2-52EC0A2A748 (Datum des Zugriffs: 2. Mai 2006)

Bissantz, N. & Hagedorn, J. (1993): Data Mining (Datenmustererkennung), in: Wirtschaftsinformatik: 5/1993, S.481

Chaffin, R. (1992): The concept of a semantic Relation, in: Adrienne Lehrer u.a. (Hrsg.): Frames, Fields and contrasts. New essays in semantic and lexical organisation, Erlbaum, Hillsdale, N.J. 1992, ISBN 0-8058-1089-7, S. 253-288

Computerhistory.org (2006): Internet History. http://www.computerhistory.org/exhibits/internet_history (Datum des Zugriffs: 6. Mai 2006).

Dostal, W. & Jeckle, M. (2004): Semantik, Oden einer Service-orientierten Architektur. http://www.sigs.de/publications/js/2004/01/dostal_JS_01_04.pdf . (Datum des Zugriffs: 2. Mai 2006).

Dostal, W. & Jeckle, M. / Melzer, I. / Zengler, B. (2004): Semantic Web. http://www.sigs.de/publications/os/2004/05/dostal_melzer_OS_05_04.pdf . (Datum des Zugriffs: 2. Mai 2006).

Engineering Psychology Institute (2005): Grundlagenforschung und cognitive Psychologie. <http://www.ep-i.ch/m3/m32.php> (Datum des Zugriffs: 6. Mai 2006).

Entrieva Inc. (2006): Entrieva – ClickSenseTM. <http://www.entrieva.com/entrieva/semiomap.htm> (Datum des Zugriffs: 7. Mai 2006).

- Erlenkötter, H. (2003):** XML – Extensible Markup Language von Anfang an. Hamburg: Rowohlt Taschenbuch Verlag GmbH, 2003
- Fensel, D. (2004):** The Semantic Web – Capitalising on growing Semantic Web expertise.
http://europa.eu.int/information_society/istevent/2004/cf/document.cfm?doc_id=1289
(Datum des Zugriffs: 7. Mai 2006)
- Hofmann, T. & Raitelhuber, U. (1998):** SGML/XML. <http://th-o.de/sgml/sgmlv.htm>
(Datum des Zugriffs: 8. Mai 2006)
- Jeckle, M. (2004):** Web Services. <http://www.jeckle.de/webServices/index.html>. (Datum des Zugriffs: 3. Mai 2006).
- Knuth, M. (2003):** Web Services – Einführung und Übersicht. 2. erweiterte Auflage. Frankfurt: Software & Support Verlag GmbH, 2003
- Kunze, J. (2006):** Morphologie. <http://www2.hu-berlin.de/compling/Lehrstuhl/Skripte/Morphologie/index.html> (Datum des Zugriffs: 17.7.2006)
- Maedche, M. & Motik, B. (2003):** Repräsentations- und Anfragesprachen für Ontologien – Eine Übersicht. http://www.fzi.de/KCMS/kcms_file.php?action=link&id=145. (Datum des Zugriffs: 2. Mai 2006).
- Meinike, T. (2002):** Know-how: Statische/ dynamische SVG-Grafiken .
<http://www.vnunet.de/praxis/netzwerke/article20020610509.aspx> (Datum des Zugriffs: 13. Juli 2006)
- Nohr, H. (2005):** Wissensrepräsentation, SS 05. Foliensatz. Unveröffentlicht. Hochschule der Medien Stuttgart.
- Ogbuji, U. (2002):** The Past, Present and Future of Web Services - part 1.
[http://www.webservices.org/categories/enterprise/strategy_architecture/the_past_present_and_future_of_web_services_part_1/\(go\)/Articles](http://www.webservices.org/categories/enterprise/strategy_architecture/the_past_present_and_future_of_web_services_part_1/(go)/Articles) . (Datum des Zugriffs: 3. Mai 2006).
- Pepper, S. (2000):** The TAO of Topic Maps - Finding the Way in the Age of Infoglut.
<http://xml.coverpages.org/pepperTAOofTopicMaps.pdf> (Datum des Zugriffs: 8. Mai 2006)
- Peter, I. (2004):** Beginnings of the Internet.
<http://www.nethistory.info/History%20of%20the%20Internet/beginnings.html> , Fassung von 2004. (Datum des Zugriffs: 4. Mai 2006)
- PHP.net (2006):** cURL, Client URL Bibliothek-Funktionen.
<http://de.php.net/manual/de/ref.curl.php> (Datum des Zugriffs: 13. Juli 2006).
- Pöhner G. & Renner, J. (2003):** Die Microsoft .NET-Plattform. In: Knuth, M. (Hrsg.): Web Services – Einführung und Übersicht. 2. erweiterte Auflage. Frankfurt: Software & Support Verlag GmbH, 2003

Prescod, P. (2002): Roots of the REST/SOAP Debate.

http://www.prescod.net/rest/rest_vs_soap_overview/ (Datum des Zugriffs: 13. Mai 2006)

Schäling, B. (2005): Der moderne Softwareentwicklungsprozess mit UML.

<http://www.highscore.de/uml/> (Datum des Zugriffs: 7. Juli 2006).

Schär, S. (2000): Topic Maps - GPS of the information universe? [http://www-](http://www-mmt.inf.tu-dres-den.de/lehre/Sommersemester_02/Hauptseminar/vortraege/Praesentation_Schaer.pdf)

[mmt.inf.tu-](http://www-mmt.inf.tu-dres-den.de/lehre/Sommersemester_02/Hauptseminar/vortraege/Praesentation_Schaer.pdf)

[dres-](http://www-mmt.inf.tu-dres-den.de/lehre/Sommersemester_02/Hauptseminar/vortraege/Praesentation_Schaer.pdf)

[den.de/lehre/Sommersemester_02/Hauptseminar/vortraege/Praesentation_Schaer.pdf](http://www-mmt.inf.tu-dres-den.de/lehre/Sommersemester_02/Hauptseminar/vortraege/Praesentation_Schaer.pdf)

(Datum des Zugriffs: 8. Mai 2006)

SchlumbergerSema GmbH (2002): Feinspezifikation. [http://www.semantic-](http://www.semantic-network.de/sns-AB-Feinspez10-a.pdf)

[network.de/sns-AB-Feinspez10-a.pdf](http://www.semantic-network.de/sns-AB-Feinspez10-a.pdf) (Datum des Zugriffs: 10. März 2006).

SchlumbergerSema GmbH (2003): Topic Maps, Web Services, and the Semantic

Web Semantic Network Services (SNS) – a case study. [http://www.semantic-](http://www.semantic-network.de/about-sns-xtm.pdf)

[network.de/about-sns-xtm.pdf](http://www.semantic-network.de/about-sns-xtm.pdf) (Datum des Zugriffs: 10. März 2006)

Schmaltz, R. (2004): Semantic Web Technologien für das Wissensmanagement. Ar-

beitsbericht Nr. 1/2004, in: Matthias Schumann (Hrsg.). Institut für Wirtschaftsinforma-

tik, Georg-August-Universität Göttingen. [http://www.wi2.wiso.uni-](http://www.wi2.wiso.uni-goettingen.de/getfile?DateilD=466)

[goettingen.de/getfile?DateilD=466](http://www.wi2.wiso.uni-goettingen.de/getfile?DateilD=466) (Datum des Zugriffs: 7. Mai 2006).

SNS (2006): SNS – Semantischer Netzwerk Service. [http://www.semantic-](http://www.semantic-network.de/doc_intro.html?lang=de)

[network.de/doc_intro.html?lang=de](http://www.semantic-network.de/doc_intro.html?lang=de) (Datum des Zugriffs: 10. März 2006)

Sowa, J.F. (2002): Semantic Networks. <http://www.jfsowa.com/pubs/semnet.htm> (Da-

tum des Zugriffs: 6. Mai 2006).

Stock, W.G. (2000): Informationswirtschaft – Management externen Wissens. Mün-

chen, Wien: Oldenbourg Wissenschaftsverlag GmbH, 2000

Theus, M. (2004): Cluster Analyse. [http://stats.math.uni-](http://stats.math.uni-augsburg.de/lehre/SS04/CA1.pdf)

[augsburg.de/lehre/SS04/CA1.pdf](http://stats.math.uni-augsburg.de/lehre/SS04/CA1.pdf) (Datum des Zugriffs: 15. Mai 2006)

Topicmaps.org (2001): XML Topic Maps (XTM) 1.0 - TopicMaps.Org Specification.

<http://www.topicmaps.org/xtm/1.0/> (Datum des Zugriffs: 8. Mai 2006)

Ullenboom, C. (2005): Java ist auch eine Insel – Kapitel 18 Verteilte Programmierung mit RMI und SOAP.

http://www.galileocomputing.de/openbook/javainsel5/javainsel18_014.htm (Datum des Zugriffs: 16. Mai 2006)

Ullrich, M. / Maier, A. / Angele, J. (2003): Taxonomie, Thesaurus, Topic Map, Ontologie - ein Vergleich. ontoprise® GmbH.

http://www.ontoprise.de/content/e5/e69/e208/TaxonomieThesaurusTopicMapOntologie_v13_ger.pdf (Datum des Zugriffs: 7. Mai 2006)

Umweltbundesamt (2006): SNS Anwenderdokumentation - Im Auftrag des Umweltbundesamtes, Dessau – Version 1.5, 12.06.2006. <http://www.semantic-network.de/sns-ws.pdf> (Datum des Zugriffs: 14. Juni 2006)

Umweltbundesamt.de (2005): Datenbanken des Umweltbundesamtes - Umweltthesaurus. <http://www.umweltbundesamt.de/uba-datenbanken/thes.htm> (Datum des Zugriffs: 6. Mai 2006).

Vogele, T. (2002): HyTime und Topic Maps. Oberseminar 2002 Video- und Audiokompression, Visualisierung und Animation am Lehrstuhl für Informatik II der Universität Würzburg. <http://www2.informatik.uni-wuerzburg.de/mitarbeiter/holger/lehre/osss02/vogele/vortrag.pdf> (Datum des Zugriffs: 8. Mai 2006).

W3C (2001): Semantic Web. <http://www.w3.org/2001/sw/>. (Datum des Zugriffs: 6. Mai 2006).

W3C (2002): Web Services. <http://www.w3.org/2002/ws/>. (Datum des Zugriffs: 6. Mai 2006).

W3C (2003): SOAP Version 1.2 Part 2: Adjuncts. <http://www.w3.org/TR/soap12-part2/>. (Datum des Zugriffs: 12. Mai 2006).

Wenz, C. (2006): Ajax – schnell + kompakt. Frankfurt: Software & Support Verlag GmbH, 2006

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ort, Datum

Unterschrift

