Hochschule Bonn-Rhein-Sieg
University of Applied Sciences

Fachbereich Informatik
Department of Computer Science

Masterthesis

Master of Science in Computer Science

Semantische Suche in Wissensportalen

Konzeption und Evaluation der Erweiterung eines Suchframeworks um semantische Technologien

von

Thorsten Schäfer

Abgabetermin:
18. März 2013

Erstprüfer : Prof. Dr. Manfred Kaul
Hochschule Bonn-Rhein-Sieg
Fachbereich Informatik
Zweitprüfer : Prof. Dr. Sascha Alda
Hochschule Bonn-Rhein-Sieg
Fachbereich Informatik

Eidesstattliche Erklärung

Name:	Thorsten Schäfer
Adresse:	Kasparstr. 10
	50670 Köln

Ich versichere an Eides Statt, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Ort	Datum	Unterschrift

Inhaltsverzeichnis

Abbildungsverzeichnis

Listings

Tabellenverzeichnis

Abkürzungsverzeichnis

AJAX Asynchronous JavaScript and XML

API Application Programmers Interface

DIN Deutsche Institut für Normung

DISCO extracting DIstributionally related words using CO-occurrences

DLR Deutsche Zentrum für Luft- und Raumfahrt e.V.

Elib Electronic Library

FOAF Friend of a Friend

GUI Graphical User Interface

HTML HyperText Markup Language

IDF Inverse Document Frequency

IR Information Retrieval

IRI Internationalized Resource Identifier

ISO International Standardisation Organisation

KEA Keyphrase Extraction Algorithm

MCI Mensch-Computer-Interaktion

MVC Model View Contoller

NLP Natural Language Processing

OAI Open Archives Initiative

OAI-PMH OAI Protocol for Metadata Harvesting

ORM Objekt-Relationales-Mapping

OWL Web Ontology Language

RDF Resource Description Framework

RDFS RDF Schema

RFC Request for Comments

RIA Rich Internet Application

RIF Rule Interchange Format

Sail Storage And Inference Layer

SKOS Simple Knowledge Organisation System

SPARQL SPARQL Protocol And RDF Query Language

SVN Subversion

TF Term Frequency

TF-IDF Term Frequency-Inverse Document Frequency

UI User Interface

UML Unified Modeling Language

URI Uniform Ressource Identifier

W3C World Wide Web Consortium

WWW World Wide Web

XML eXtensibe Markup Language

1. Einleitung

1.1. Motivation & Problemstellung

Einhergehend mit der zunehmenden Digitalisierung nimmt die weltweit verfügbare Informationsmenge immer weiter zu. Das Auffinden von relevanten Informationen innerhalb dieser Informationsflut ist eine schwierige Aufgabe. Das prominenteste Beispiel für einen riesigen Datenspeicher ist das World Wide Web (WWW). Hier werden webbasierte Suchmaschinen wie Google dazu eingesetzt, relevante Informationen ausfindig zu machen. Diese Suchmaschinen verwenden dazu meist klassische Suchverfahren, die auf textueller und struktureller Ähnlichkeit basieren. Aufgrund von Mehrdeutigkeiten und unterschiedlichen Kontexten stoßen diese klassischen Verfahren jedoch oftmals an ihre Grenzen. Die Vollständigkeit eines Suchergebnisses, welches zum Beispiel mehr als eine Millionen Treffer liefert, ist nur noch schwer zu beurteilen. Semantische Technologien haben zum Ziel diesen Mangel zu beheben. Hier wird neben der textuellen und strukturellen Ähnlichkeit zusätzlich die Dimension der *Bedeutung* betrachtet. Ziel ist es dabei Informationen für eine Maschine interpretierbar zu machen und dadurch die Suchergebnisqualität zu verbessern. Bei der semantischen Suche wird also versucht, Informationen auch auf Basis ihrer Bedeutungen aufzufinden.

Das Deutsche Zentrum für Luft- und Raumfahrt e.V. (DLR) ist eine Forschungseinrichtung der Bundesrepublik Deutschland für Luft- und Raumfahrt, Energie und Verkehr. Das DLR beschäftigt deutschlandweit ca. 7.400 Mitarbeiter an insgesamt 16 Standorten. Die 32 Institute und Einrichtungen forschen in unterschiedlichsten Themengebieten wie beispielsweise Aerodynamik und Strömungstechnik, Lufttransportsysteme, Solarforschung oder Erdbeobachtung [DLR 2013]. Für die Durchführung größerer Projekte wie die Entwicklung von Flug- oder Raumfahrzeugen ist Wissen aus einer Vielzahl von Fachgebieten erforderlich. Oftmals ist es dazu notwendig, dass sich Wissenschaftler fachübergreifend in Themengebiete einarbeiten müssen. Im Rahmen dieser Einarbeitung führen diese Wissenschaftler Recherchen in fremden Fachbereichen durch.

Das DLR hat zu diesem Zweck das Wissensportal *KnowledgeFinder* entwickelt. Dieses Framework setzt klassische Suchverfahren zum Auffinden von Informationen in beliebigen Datenbeständen ein. Eine Anwendung des KnowledgeFinder ist beispielsweise die Suche innerhalb der DLR-Publikationsdatenbank. Wenn Wissenschaftler in fremden Fachbereichen recherchieren, dann fällt es ihnen aufgrund des oberflächlichen Einblicks oftmals schwer, zielgerichtet nach Informationen zu suchen. Sie kennen beispielsweise nicht sämtliche Fachtermini, die für eine spezifischere Suchanfrage notwendig wären. Daneben fehlen meist auch detaillierte Kenntnisse über Verknüpfungen zwischen den fachfremden Themen. Die im KnowledgeFinder eingesetzten klassischen Suchverfahren können bei diesen unspezifischen Suchanfragen nur bedingt beim Auffinden von relevanten Informationen helfen.

1.2. Zielsetzung

In der vorliegenden Arbeit soll untersucht werden, ob die Suchergebnisqualität des KnowledgeFinder durch den Einsatz semantischer Technologien verbessert werden kann. Innerhalb einer Machbarkeitsstudie soll dieses Framework dazu um semantische Suchverfahren erweitert werden. Diese Verfahren sollen die fachübergreifende Recherche von DLR-Wissenschaftlern erleichtern, indem sie ihnen helfen, passende Suchergebnisse in den entsprechenden Fachbereichen zu finden. Das Hauptaugenmerk der Arbeit liegt dabei auf der Konzeption. Um eine Evaluation durchführen zu können, soll weiterhin eine prototypische Implementierung stattfinden. Neben der Evaluation der Suchergebnisqualität soll die Relevanz semantischer Technologien im Kontext des DLR untersucht werden. Hier steht die Fragestellung der Umsetzbarkeit und des Mehrwerts dieser Technologien im Mittelpunkt. In diesem Zusammenhang soll zudem diskutiert werden, ob Gründe dafür gefunden werden können, warum sich semantische Technologien nicht auf breiter Basis durchgesetzt haben.

1.3. Vorgehensweise & Struktur der Arbeit

Um die zuvor genannten Fragestellungen beantworten zu können, wird ein beispielhafter Anwendungsfall herangezogen. Anhand dieses Falls wird im weiteren Verlauf der Arbeit eine Machbarkeitsstudie durchgeführt. Die vom KnowledgeFinder realisierte Suche in der Publikationsdatenbank des DLR wird zu diesem Zweck herangezogen und um semantische Suchfunktionalitäten erweitert. Die Veröffentlichungen aus dieser Datenbank decken alle Forschungsfelder des DLR ab. Dieser Anwendungsfall repräsentiert somit die Vielschichtigkeit und Komplexität dieser Forschungseinrichtung.

Die Arbeit besteht aus zwei Hauptteilen und insgesamt zehn Kapiteln. Teil I beschreibt zunächst die Grundlagen zur Durchführung der Machbarkeitsstudie. Zu Beginn wird in Kapitel 2 eine Einführung in die Thematik des Information Retrieval gegeben. Danach stellt Kapitel 3 die Grundlagen semantischer Technologien vor. Diese Technologien bilden die zentralen Bausteine zur Realisierung einer semantischen Suche. Hier werden zunächst verschiedene Modelle zur Repräsentation von semantischem Wissen dargelegt und anschließend die Semantic-Web-Standards als Rahmenwerk erörtert. Kapitel 4 beschreibt aufbauend auf den Grundlagen semantischer Technologien, was unter dem Prinzip der semantischen Suche zu verstehen ist. Neben einer Begriffsdefinition und dem Ziel wird eine allgemeine Architektur vorgestellt. Weiterhin wird ein Überblick über verschiedene Ansätze semantischer Suchverfahren verschafft. In Kapitel 5 erfolgt eine detaillierte Beschreibung des KnowledgeFinder-Wissensportal. In Teil II der Ausarbeitung wird die Durchführung der Machbarkeitsstudie vorgestellt und die darauf basierenden Ergebnisse präsentiert. Dazu findet in Kapitel 6 als konzeptionelle Grundlage der Studie eine Spezifikation der Anforderungen statt. Die Konzeption der semantischen Suche erfolgt in Kapitel 7. Ein zentraler Aspekt dieses Kapitels ist zunächst die Auswahl eines geeigneten Ansatzes zur semantischen Suche. Die Integration des ausgewählten Lösungsansatzes innerhalb des KnowledgeFinder wird anschließend entworfen und die Architektur des Zielsystems vorgestellt. Kapitel 8 erörtert die prototypische Implementierung des konzipierten Ansatzes. Aufbauend auf dieser Implementierung findet in Kapitel 9 die Evaluation des vorgestellten Konzepts statt. Zunächst steht hier die Suchergebnisqualität im Mittelpunkt der Untersuchung. Aufbauend auf dieser Analyse wird anschließend die Umsetzbarkeit und der Mehrwert semantischer Technologien im Kontext des DLR beleuchtet. In Kapitel 10 erfolgt zuletzt eine Zusammenfassung der Ergebnisse dieser Arbeit.

Teil I.

Grundlagen

2. Information Retrieval

Das Information Retrieval beschäftigt sich genauso wie das Themengebiet der semantischen Suche mit dem Auffinden von Informationen. Im Folgenden wird ein Überblick über die Grundlagen dieser Art der Informationssuche gegeben. Zunächst wird dazu in Abschnitt 2.1 eine Begriffsdefinition durchgeführt und das Ziel des Information Retrieval dargelegt. Anschließend findet eine Erläuterung des grundlegenden Aufbaus dieser Systeme statt. Danach werden unterschiedliche Modelle des klassischen Information Retrieval vorgestellt, nach denen diese Systeme arbeiten. In Abschnitt 2.4 findet eine Betrachtung des Prozesses der Informationssuche statt. Zuletzt werden die wichtigsten Evaluierungsparameter des Information Retrieval erläutert. Auf Grundlage dieser Darstellungen werden im weiteren Verlauf der vorliegenden Arbeit die Unterschiede und Gemeinsamkeiten zu dem Ansatz der semantischen Suche herausgestellt. Aufgrund der Tatsache, dass der KnowledgeFinder als Suchframework nach Information-Retrieval-Prinzipien arbeitet, sollen diese Ausführungen weiterhin zu einem besseren Gesamtverständnis beitragen.

2.1. Begriffsdefinition & Ziel

Zunächst ist im Kontext des Information Retrieval (IR) zu klären, was unter dem Begriff der *Information* zu verstehen ist. In diesem Zusammenhang müssen auch die Begriffe *Daten* und *Wissen* betrachtet werden. Unter Daten werden Elemente verstanden, deren Typ oder syntaktische Struktur bekannt ist. Daten werden zu Informationen, wenn sie semantisch interpretiert werden können. Informationen werden zu Wissen, wenn sie von Menschen aufgenommen, verarbeitet und genutzt werden können [Ferber 2003, S. 27]. In Anlehnung an Dengel kann also Wissen als *„Information, die in Aktion umgesetzt wird“* bezeichnet werden [Dengel 2012, S. 5]. Im IR werden computergestützte Verfahren dazu eingesetzt, Informationen in großen Datenmengen ausfindig und für den Menschen zugänglich zu machen [Hermans 2008, S. 23].

In der Literatur ist keine allgemein akzeptierte Begriffsdefinition des IR zu finden [Ferber 2003, S. 29]. Baeza-Yates und Ribeiro-Neto definieren die Aufgaben und Ziele aber wie folgt:

> *„Information Retrieval (IR) deals with the representation, storage, organization of, and access to information items. The representation and organization of the information items should provide the user with easy access to the information in which he is interested.“* [Baeza-Yates und Ribeiro-Neto 1999, S. 1]

Diese Definition lässt erkennen, dass für den effektiven Zugriff auf Informationen zunächst Themen wie die Repräsentation, die Speicherung und die Organisation der informationstragenden Einheiten zu betrachten sind. Erst nachdem entsprechende Verfahren angewandt wurden, kann der Nutzer seinen Informationsbedarf artikulieren und der Zugang zu Informationen erfolgen. Baeza-Yates und Ribeiro-Neto heben jedoch hervor, dass die Charakterisierung des Informationsbedarfs ein schwieriges Problem darstellt [Baeza-Yates und Ribeiro-Neto 1999, S. 1]. Dies liegt zum einen darin begründet, dass der Nutzer seinen Informationsbedarf nicht direkt an ein IR-System adressieren kann. So muss bei webbasierten Suchmaschinen

wie z.B. Google zunächst eine Transformation des Informationsbedarfs in eine Menge von Schlüsselwörtern stattfinden. Neben dieser Problematik spielen zum anderen Aspekte der Wissensverarbeitung eine Rolle. Nach Ferber haben IR-Systeme aufgrund ihrer Funktion eine wichtige Aufgabe im Prozess des Wissenstransfers [Ferber 2003, S. 30]. Hier dienen sie als Vermittler von Wissen zwischen den Menschen. Das in IR-Systemen vorliegende Wissen ist aber aufgrund beschränkter Abbildungsmöglichkeiten nicht vollständig. In diesem Zusammenhang spricht der Autor auch von der Vagheit des IR. Diese Vagheit trägt ebenfalls zur der Schwierigkeit der Informationsbedarfsdeckung bei. Diese Vagheit kann daneben auch zur Abgrenzung des IR zum *Data Retrieval* herangezogen werden. Beim Data Retrieval liegen wohlstrukturierte Daten vor. Zu einer präzisen Anfrage wird hier die Ergebnismenge geliefert, die exakt den Anfragebedingungen entspricht. Im Unterschied dazu arbeiten IR-Systeme auf unstrukturierten Inhalten in Form natürlichsprachlicher Texte. Ein IR-System muss diese Inhalte verarbeiten und ihre Relevanz gegenüber der unpräzisen Anfrage des Nutzers bewerten. Die Relevanzbestimmung stellt hierbei eine Kernproblematik des IR dar [Baeza-Yates und Ribeiro-Neto 1999, S. 2].

Insgesamt kann festgehalten werden, dass das Information Retrieval die Repräsentation, die Speicherung, die Organisation und das Auffinden von Informationen umfasst. Das Ziel eines IR-Systems besteht darin, den Informationsbedarf des Nutzers zu decken. Dabei soll der Anteil der relevanten Ergebnisse möglichst hoch und der Anteil irrelevanter Ergebnisse möglichst niedrig sein.

2.2. Aufbau eines Information-Retrieval-Systems

Abbildung 2.1 zeigt das grundlegende Schema eines Information-Retrieval-Systems. Auf der einen Seite sind die informationstragenden Objekte, die durchsucht werden sollen, und auf der anderen Seite der Informationsbedarf des Nutzers. Um eine Suche durchführen zu können, müssen sowohl die Objekte als auch der Informationsbedarf in eine entsprechende Repräsentationsform transformiert werden. Innerhalb des Suchvorgangs (siehe Kreis) führt das IR-System ein Vergleich der beiden Repräsentationen auf der Basis einer Relevanzbestimmung durch und liefert eine entsprechende Ergebnismenge an den Nutzer zurück.

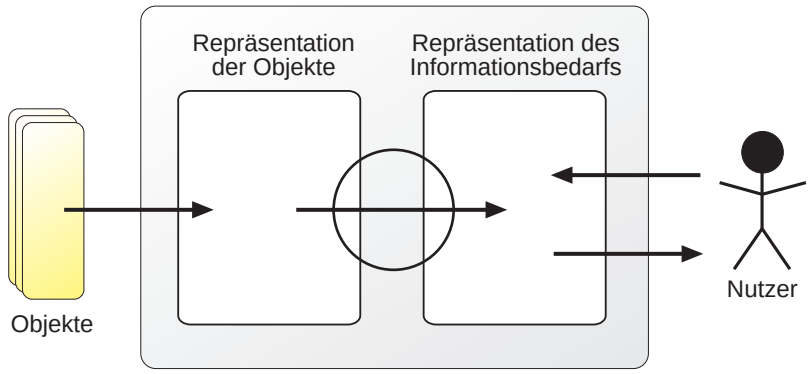


Abbildung 2.1.: Grundlegendes Schema eines IR-Systems (vgl. [Ferber 2003, S. 25]

Im weiteren Verlauf wird nun das klassische schlüsselwortbasierte IR betrachtet, da der KnowledgeFinder als Wissensportal nach diesem Prinzip arbeitet. Beim schlüsselwortbasierten IR liegen die in der Abbildung dargestellten Objekte in Form von Dokumenten vor. Bei diesem Ansatz wird jedes Dokument durch eine Menge von Schlüsselwörtern (engl. keywords)

beschrieben. Diese Schlüsselwörter werden auch als Indexterme bezeichnet [Baeza-Yates und Ribeiro-Neto 1999, S. 24]. Diese Indexterme dienen dazu, den Inhalt eines Dokuments zu beschreiben. Um innerhalb des IR-Systems ein effizientes Durchsuchen zu ermöglichen, werden diese Indexterme in einem sogenannten *invertierten Index* abgelegt. Der invertierte Index ist eine spezielle Speicherstruktur, in der eine Zuordnung der Indexterme zu deren Vorkommen innerhalb der Dokumente stattfindet. Um einen solchen Index aus den Volltexten der Dokumente zu erstellen, werden verschiedene lexikalische Methoden zur Textvorverarbeitung angewendet. Zu nennen sind hier unter anderem die *Stoppwort-Eliminierung*[1] und das sogenannte *Stemming*[2]. Im Kontext der zuvor dargestellten termbasierten Sichtweise wird häufig vom sogenannten *Bag-of-Words-Ansatz* des IR gesprochen [Cimiano 2006, S. 283].

2.3. Information-Retrieval-Modelle

Die Mechanismen zur Repräsentationsüberführung sowie zur Relevanzbestimmung sind abhängig vom zugrundeliegenden IR-Modell. Im Rahmen des klassischen IR finden das *boolesche Modell* und das *Vektorraummodell* am häufigsten Verwendung. Das boolesche Modell ist ein einfacher mengentheoretischer Ansatz. Hier wird zu jedem Indexterm lediglich dessen Dokumenten-Vorkommen vermerkt. Die Relevanzbestimmung findet anschließend auf Basis von binären Entscheidungen statt. Beim Vektorraummodell hingegen werden nicht-binäre Werte zu jedem Indexterm gespeichert und im Rahmen der Relevanzbestimmung zur Gewichtung herangezogen [Baeza-Yates und Ribeiro-Neto 1999, S. 21 ff.].

Folgende Parameter spielen im Rahmen der Gewichtung des Vektorraummodells eine entscheidende Rolle:

- **Termhäufigkeit (TF)**: Bei der Termhäufigkeit (engl. term frequency) wird davon ausgegangen, dass Terme, die innerhalb eines Dokuments oft vorkommen, eine höhere Bedeutung für deren inhaltliche Beschreibung besitzen als selten vorkommende Terme. Bei langen Texten ist die Häufigkeit von Termen größer als bei kurzen Texten. Deswegen wird zur Normalisierung der Termhäufigkeit das Vorkommen des häufigsten Terms mit hinzugezogen [Ferber 2003, S. 69].
- **Inverse Dokumentenhäufigkeit (IDF)**: Die inverse Dokumentenhäufigkeit (engl. inverse document frequency) dient zur Gewichtung eines Terms in Bezug zum gesamten Text-Korpus. Hier wird von der Annahme ausgegangen, dass Terme, die im gesamten Text-Korpus sehr häufig vorkommen, eine geringere Relevanz besitzen. Der IDF-Wert ergibt sich aus der Gesamtanzahl der Dokumente dividiert durch die Anzahl an Dokumenten, welche den Term enthalten [Baeza-Yates und Ribeiro-Neto 1999, S. 29].

Die Termhäufigkeit nimmt also eine lokale Gewichtung von Termen vor. Die inverse Dokumentenhäufigkeit hingegen dient der globalen Gewichtung der Terme. Werden diese beiden Häufigkeiten miteinander multipliziert, so ergibt sich die **TF-IDF-Gewichtung** (engl. term frequency-inverse document frequency). Die TF-IDF-Gewichtung ist nach Baeza-Yates und Ribeiro-Neto das am besten bekannte Schema zur Termgewichtung im Rahmen des Vektorraummodells [Baeza-Yates und Ribeiro-Neto 1999, S. 29].

[1] Bei der Stoppwort-Eliminierung werden beispielsweise bestimmte und unbestimmte Artikel sowie Konjunktionen entfernt.

[2] Beim Stemming findet eine Grundwortreduktion der Indexterme statt.

2.4. Information-Retrieval-Prozess

Abbildung 2.2 zeigt eine Übersicht über den Retrieval-Prozess des klassischen IR. Bevor der Text-Korpus durchsucht werden kann, muss innerhalb einer Offlineaktivität der invertierte Index erstellt werden. Dies geschieht mit Hilfe der bereits erwähnten Verfahren der Textvorverarbeitung. Der interne Aufbau des erzeugten Index ist abhängig vom gewählten IR-Modell. Im Anschluss an die Indizierung kann der Nutzer eine Informationssuche starten. Wie schon zuvor beschrieben, muss er dazu zunächst seinen Informationsbedarf innerhalb der Benutzerschnittstelle formulieren. Um eine logische Sicht der Nutzeranfrage zu erhalten, findet danach die gleiche Vorverarbeitung statt wie zuvor im Rahmen des Text-Korpus. In der Phase der Anfrageverarbeitung wird danach die vom System verarbeitbare Repräsentationsform der Anfrage erzeugt. Im Anschluss an diese Phase erfolgt die eigentliche Suche. Bevor die gefundenen Dokumente dem Nutzer präsentiert werden, findet eine Rangfolgenbildung (engl. ranking) statt. Dieses Ranking wird dabei auf Basis der Gewichtungen des IR-Modells durchgeführt. In einigen IR-Systemen hat der Nutzer an dieser Stelle die Möglichkeit, dem System ein Feedback zu den gefundenen Dokumenten zu geben. Diese Rückkopplung kann vom System dazu genutzt werden, eine neue Suche durchzuführen, deren Ergebnis relevanter ist als zuvor [Baeza-Yates und Ribeiro-Neto 1999, S. 9 f.].

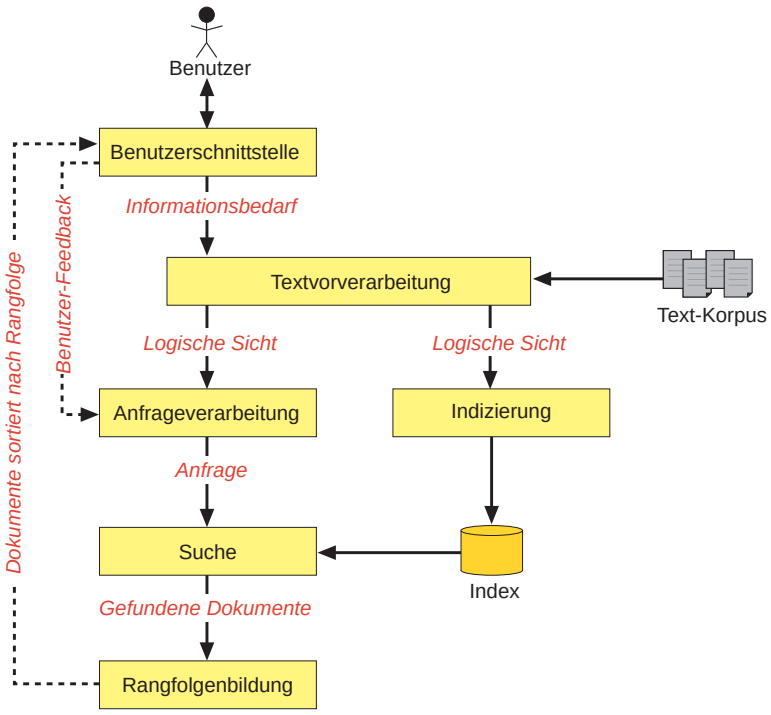


Abbildung 2.2.: Übersicht über den Retrieval-Prozess
(vgl. [Baeza-Yates und Ribeiro-Neto 1999, S. 10] u. [Sánchez 2009, S. 20]

2.5. Evaluation von Information-Retrieval-Systemen

Wie zuvor geschildert, besteht das Ziel von IR-Systemen darin, den Informationsbedarf des Nutzers zu decken. Es sollen möglichst nur relevante Ergebnisse geliefert werden. Im Rahmen der Evaluation von IR-Systemen wird genau diese Thematik untersucht. Die Frage, die in diesem Kontext zwangsläufig auftaucht: Wann genau ist ein Dokument für den Nutzer

relevant? Da dies ein Computersystem nicht entscheiden kann, findet die Evaluation von IR-System mit Hilfe von sogenannten *Testkollektionen* statt. Innerhalb dieser Kollektionen ist genau definiert, welche Dokumente für welche Informationsbedürfnisse relevant sind. Auf Basis dieser vordefinierten Relevanz kann dann eine Evaluation stattfinden. Die wichtigsten Parameter, die zur Bewertung herangezogen werden, sind die Genauigkeit (engl. precision) und die Vollständigkeit (engl. recall), die im Folgenden kurz dargestellt werden [Baeza-Yates und Ribeiro-Neto 1999, S. 75].

Gegeben ist eine Anfrage I, eine dazugehörige Menge an relevanten Dokumenten R. Sei $|R|$ die Anzahl relevanter Dokumente, dann erzeugt die Anfrage I die Ergebnismenge E. Wie in Abbildung 2.3 veranschaulicht, sei weiterhin $|Re|$ die Anzahl der Dokumente in der Schnittmenge von R und E. Dann ist Recall und Precision wie folgt definiert:

- **Recall** bezeichnet den Anteil an relevanten Dokumenten, die gefunden wurden:

$$\text{Recall} = \frac{|Re|}{|R|}$$

- **Precision** ergibt sich aus dem Anteil der gefundenen Dokumente, die relevant sind:

$$\text{Precision} = \frac{|Re|}{|E|}$$

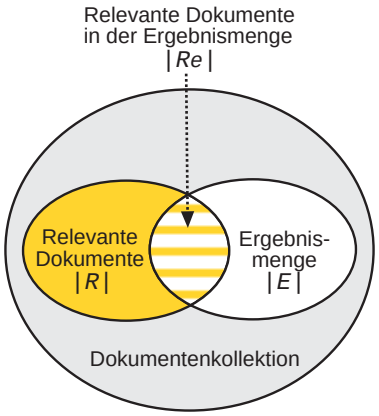


Abbildung 2.3.: Precision und Recall für eine gegebene Anfrage
(vgl. [Baeza-Yates und Ribeiro-Neto 1999, S. 75])

Die Werte für Recall und Precision liegen also immer zwischen 0 und 1. Je höher ihr Wert, desto besser die Retrieval-Performanz des betrachteten IR-Systems. Hervorzuheben ist, dass diese beiden Maße in der Praxis gegenläufig sind. Dies liegt darin begründet, dass für einen größeren Recall häufig allgemeinere Anfragen durchgeführt werden. Dies führt aber zwangsläufig zu einer Vergrößerung der Ergebnismenge E. Umgekehrt werden für eine bessere Precision spezifischere Anfragen abgesetzt, die zur Verkleinerung der Ergebnismenge führen [Ferber 2003, S. 87]. Liegt also der Recall bei 1, so tendiert die Precision gegen 0. In diesem Fall liefert das IR-System zwar alle relevanten Dokumente aus, die vergrößerte Ergebnismenge enthält aber auch sehr viele irrelevante Dokumente. Bei einer hohen Precision und einem geringen Recall findet das System zwar relevante Dokumente, aber aufgrund der spezifischeren Anfrage eben nicht alle [Sánchez 2009, S. 31 f.]. Dieser Zusammenhang wird meist mit Hilfe von *Precision-Recall-Diagrammen* veranschaulicht.

3. Semantische Technologien

Dieses Kapitel stellt die Grundlagen semantischer Technologien vor. Diese Technologien bilden die zentralen Bausteine zur Realisierung einer semantischen Suche. Dazu wird zunächst kurz auf den Begriff der Semantik im Allgemeinen eingegangen und dieser in den Kontext semantischer Technologien eingeordnet. Menschliches Wissen formal zu repräsentieren und dadurch maschinell verarbeitbar zu machen, ist eine wichtige Säule semantischer Technologien. Aufgrund dieser Tatsache werden im Anschluss daran in Abschnitt 3.2 verschiedene Modelle zur Repräsentation von semantischem Wissen vorgestellt und behandelt. Zuletzt werden Thematiken rund um den Begriff des Semantic Web erläutert. Neben der Idee und der Vision findet hier eine detaillierte Erläuterung der Semantic-Web-Standards statt. Diese Standards bilden einen weiteren wichtigen Eckpfeiler semantischer Technologien.

3.1. Semantik und semantische Technologien

Der Begriff *Semantik* bedeutet übersetzt *Lehre der Bedeutung*. Die Bedeutungslehre, als wissenschaftliches Teilgebiet der Linguistik, befasst sich mit dem Sinn und der Bedeutung von Sprachen. Sie untersucht die Zusammenhänge zwischen Objekten und ihren sprachlichen Begriffen bzw. Zeichen. Begriffe sind nach Dengel nichts anderes als Abstraktionen und Modelle der realen Welt, deren Sinn sich erst aus dem jeweiligen Kontext heraus ergibt [Dengel 2012, S. 10]. Hierbei ist es prinzipiell egal, ob es sich um ein gesprochenes Wort oder um niedergeschriebene Zeichen handelt. Dieser Zusammenhang wird oftmals mit Hilfe des sogenannten *semiotischen Dreiecks* verdeutlicht. Abbildung 3.1 zeigt dieses semiotische Dreieck in Anlehnung an Sowa [Sowa 2000].

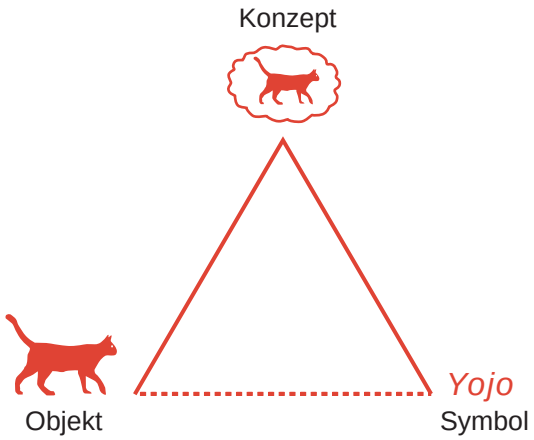


Abbildung 3.1.: Semiotisches Dreieck (vgl. [Sowa 2000])

In der rechten unteren Ecke des Dreiecks befinden sich die Symbole. Die Zeichen dieser Symbole folgen dabei meist einer bestimmten Syntax. Für sich alleinstehend haben sie jedoch keinerlei Bedeutung. Jemand, der die Katze mit dem Namen „Yojo“ nicht kennt, weiß auch nichts über die Semantik dieses Symbols. Die Semantik entsteht erst durch die Verbindung mit den beiden anderen Ecken des Dreieckes. Eine Verbindung kann aber nur dann erfolgen, wenn in unserem mentalen Modell ein Konzept vorhanden ist, welches diesem Symbol

entspricht. Das Konzept ist also ein Mediator zwischen den Objekten aus der realen Welt und den abstrakten Symbolen [Daconta et al. 2003, S. 209 f.]. Konzepte sorgen dafür, dass Symbole eine Semantik erhalten und interpretiert werden können.

Menschen sind, wenn sie über ausreichend Wissen verfügen, meistens selbstständig in der Lage, die Bedeutung von Symbolen zu erkennen. Computer besitzen diese Fähigkeit hingegen nicht. Sie sind nicht ohne weiteres in der Lage, die Bedeutung von Symbolen – in diesem Fall Daten – zu erkennen. Ein möglicher Ansatz besteht darin, die durch die Daten ausgedrückten Informationen mit beschreibenden Attributen anzureichern. Diese bedeutungstragenden Wörter werden unter dem Oberbegriff der Metadaten geführt. Die in den folgenden Abschnitten beschriebenen semantischen Technologien greifen diesen Ansatz auf und basieren auf solchen Metadaten-Infrastrukturen. Ihr Ziel besteht nicht nur darin, Informationen auf Basis ihrer Bedeutung besser auffindbar zu machen, sondern auch eine Interoperabilität zwischen verschiedenen Systemen herzustellen [Dengel 2012, S. 15]. Im Kontext semantischer Technologien werden dazu formale Semantiken eingesetzt. Formale Semantiken sind formale Beschreibungen der Bedeutung von künstlichen und natürlichen Sprachen. Die nachfolgenden beschriebenen Standards des Semantic Web beinhalten solche formalen Semantiken, um Informationen mit einer Bedeutung zu versehen.

3.2. Semantische Wissensrepräsentation

Im Rahmen dieser Ausarbeitung ist die Abbildung von Wissen in maschinenlesbarer Form ein zentraler Aspekt im Kontext der Realisierung eines semantischen Suchsystems. Für die Erstellung eines solchen Systems ist es notwendig, das Domänenwissen (oder zumindest eine Teilmenge davon) in ein Modell zu überführen, damit es innerhalb einer semantischen Suche genutzt werden kann. Die semantische Wissensrepräsentation (engl. semantic knowledge representation) beschäftigt sich genau mit dieser Thematik. Die Intention der Semantic Knowledge Representation besteht darin, geeignete Repräsentationsformen zur Abbildung von Wissen aus der realen Welt zur Verfügung zu stellen. Aus diesem Grund soll dieser Abschnitt einen Überblick darüber geben, welche möglichen Repräsentationsformen im Kontext semantischer Technologien eingesetzt werden. Ziel dabei ist es, ein grundlegendes Verständnis der verschiedenen Modelle zu vermitteln. Die vorgestellten Arten dienen im weiteren Verlauf als konzeptionelle Grundlage zur Erstellung eines semantischen Suchsystems.

3.2.1. Kontrollierte Vokabulare & Taxonomien

Kontrollierte Vokabulare bilden eine grundlegende Form der Repräsentation von Wissen. Diese sehr einfach gehaltenen Vokabulare dienen der Vermeidung von lexikalischen Mehrdeutigkeiten wie Homonyme oder Synonyme. Unter Synonymen werden unterschiedliche Wörter verstanden, die aber die gleiche Bedeutung besitzen. So ist beispielsweise das Wort „Ross“ ein Synonym für „Pferd“. Homonyme sind gleichlautende Wörter, welche aber verschiedene Begriffe umschreiben. So bezeichnet zum Beispiel das Wort „Golf“ sowohl ein Auto als auch eine Sportart [Lewandowski 2005, S. 112]. Zur Vermeidung von Homonymen besteht ein kontrolliertes Vokabular aus einem Wortschatz wie beispielsweise ein Index, Glossar oder ein Schlagwortkatalog. Innerhalb dieses Wortschatzes ist jedem Ausdruck eindeutig ein Begriff zugeordnet. Zur Vermeidung von Synonymen wird jedem Begriff eine eindeutige Bezeichnung (Deskriptor) zugeordnet. Mit Hilfe von kontrollierten Vokabularen wird eine Art Abkommen über die Semantik der Deskriptoren getroffen. Der Zweck von kontrollierten Vokabularen besteht darin, die Benutzung von nicht gewünschten Begriffen zu verhindern. Für komple-

xe Aufgabenstellungen ist diese Repräsentationsform aufgrund der fehlenden Struktur nicht geeignet. Allerdings liegen den im weiteren Verlauf beschriebenen Modellen oftmals kontrollierte Vokabulare zu Grunde. Dadurch soll die Homonym- und Synonymfreiheit sichergestellt werden [Pellegrini und Blumauer 2006, S. 362].

Eine *Taxonomie* ist eine erste strukturelle Art, um Informationen semantisch einzuordnen. Dazu findet innerhalb einer Taxonomie eine Zuordnung von Termen (oder auch Objekten) in eine vorgegebene Klassifikation statt. Dadurch werden diese Termen in Ober- und Unterklassen-Beziehung gesetzt. Die Klassen der Taxonomie sind hierarchisch organisiert und liegen in Form eines kontrollierten Vokabulars vor [Pellegrini und Blumauer 2006, S. 363]. Die Hierarchie bildet dabei eine Baumstruktur mit Knoten und Zweigen. Wie bei Baumstrukturen üblich, besitzt mit Ausnahme des Hauptknotens jeder Knoten nur einen Vorgänger. Die Terme des kontrollierten Vokabulars setzten sich aus den Konzepten eines Themengebietes zusammen [Dengel 2012, S. 48]. Klassischerweise sind Taxonomien in der Biologie und Medizin zur Einordnung von Lebewesen bzw. Krankheiten anzutreffen. Ein typischer Vertreter im informationstechnischen Umfeld ist die hierarchische Organisation von Dateiobjekten in einem Betriebssystem [Pellegrini und Blumauer 2006, S. 363]. Taxonomien haben den Nachteil, dass sie relativ unflexible Gebilde sind. Eine einmal festgelegte Klassifikation kann im Anschluss nicht mehr so leicht abgeändert werden. Als weitere Schwachstelle ist die stark eingeschränkte Möglichkeit der Relationsbildung zu nennen. Abbildung 3.2 zeigt einen beispielhaften Auszug aus der sogenannten *linnaeischen Taxonomie* zur Klassifizierung von Lebewesen. Zu sehen ist die Einordnung der menschlichen Spezies innerhalb dieses Klassifizierungssystems.

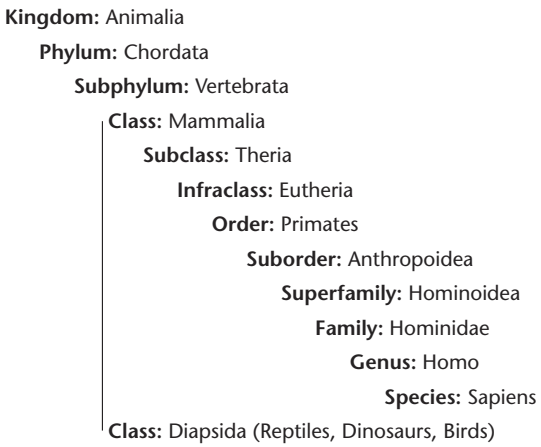


Abbildung 3.2.: Beispiel einer Taxonomie zur Klassifizierung von Lebenwesen [Daconta et al. 2003]

3.2.2. Thesauri

Thesauri bilden Konzepte einer Sprache oder eines Wissensgebietes ab und gehören wie Taxonomien zu der Kategorie der lexikalischen Modelle. Thesauri erweitern die hierarchischen Relationen von Taxonomien um eine Menge an nicht-hierarchischen Relationen. Aufgrund dieser Erweiterung besitzen sie eine höhere Ausdrucksfähigkeit. Die typischen Beziehungen eines Thesaurus sind wie folgt [Dengel 2012, S. 92]:

1. **Hierarchische Beziehungen:**
   a) Hypernym: Oberbegriff. Im Gegensatz zu einer Taxonomie sind in einem Thesaurus mehrere Oberbegriffe erlaubt

b) Hyponym: Unterbegriff

c) Meronym: Der Begriff ist Teilmenge eines anderen Begriffes

d) Holonym: Der Begriff enthält andere Begriffe

2. **Nicht-hierarchische Beziehungen:**

a) Synonym: Wörter mit ähnlicher Bedeutung

b) Antonym: Gegenwort. Wörter, die das Gegenteil ausdrücken

c) Assoziationen: Beziehungen zu anderen Begriffen

Allgemein kann zwischen linguistischen Thesauri und Thesauri zur Dokumentation unterschieden werden. Linguistische Thesauri bilden den Wortschatz einer Sprache oder einer Wissensdomäne ab. Thesauri zur Dokumentation dienen der effizienten Verschlagwortung von Dokumenten. Linguistische Thesauri werden häufig als Hilfsmittel sowohl im Indizierungs- als auch Suchprozesses des IR eingesetzt. Dengel nennt als möglichen Anwendungsfall beispielsweise die Erweiterung von Suchanfragen (engl. query expansion) [Dengel 2012, S. 93 f.]. Aufgrund der Querverbindung zwischen den Konzepten eines Wissensgebietes stellt ein Thesaurus daneben auch eine Art *semantische Landkarte* dar. Das heißt, dieses Repräsentationsmodell kann zu einem besseren Verständnis eines Wissensgebietes beitragen und somit als ontologische Wissensquelle dienen [Aitchison et al. 2000, S. 1]. Der Einsatz von Thesauri zur Repräsentation von Wissen hat den Nachteil, dass nur eine eingeschränkte Anzahl von Beziehungstypen zur Verfügung stehen. Wenn mehr als nur diese Typen benötigt werden, so empfiehlt sich die Verwendung einer Ontologie. Abbildung 3.3 zeigt einen beispielhaften Auszug aus dem WordNet-Thesaurus[1]. WordNet ist ein bekanntes Beispiel eines linguistischen Thesaurus. Hier werden alle Wörter der englischen Sprache in sogenannten *Synsets* organisiert. Ein Synset enthält eine Zusammenstellung von Termen, welche in einem bestimmten Kontext die gleiche Semantik besitzen. In dem hier dargestellten Beispiel werden die unterschiedlichen Bedeutungen des englischen Worts „car“ mit Hilfe von Synsets veranschaulicht.

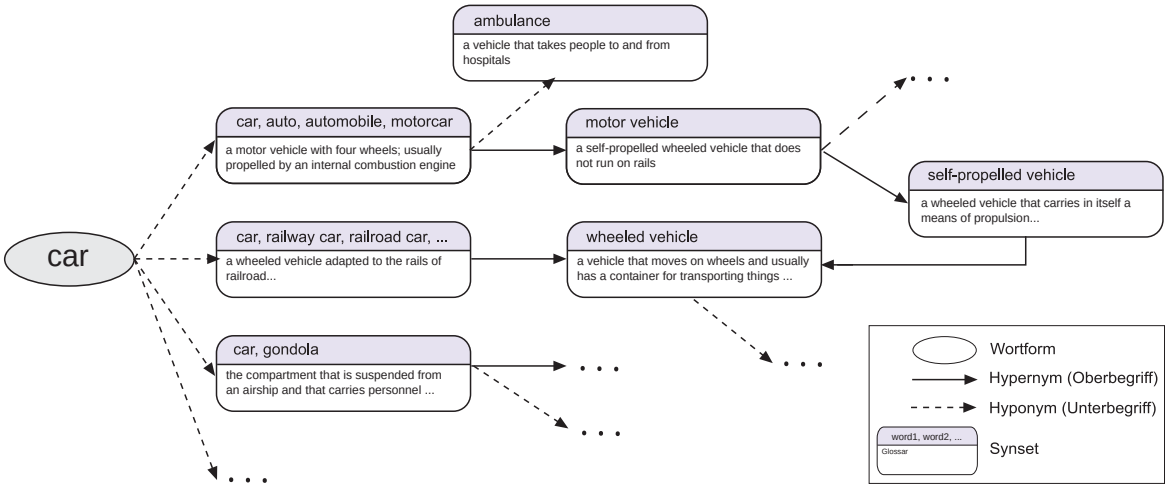


Abbildung 3.3.: Auszug aus dem WordNet-Thesaurus [Cimiano 2006, S. 56]

[1] http://wordnet.princeton.edu/ (Zugriffsdatum: 20.01.2013)

Zur Dokumentation von Thesauri existieren mehrere nationale sowie internationale Standardisierungen. Neben DIN-Normen[2] (Deutsche Institut für Normung) sowie ISO-Standards[3] (International Standardisation Organisation) ist hier vor allem das Simple Knowledge Organisation System (SKOS) zu nennen. SKOS ist ein auf dem Resource Description Framework (RDF) aufbauender Standard, der es ermöglicht, Thesauri in Semantic-Web-Applikationen verwenden zu können [Miles und Bechhofer 2009].

3.2.3. Topic Maps

Topic Maps bilden ein weiteres Modell zur semantischen Wissensrepräsentation, welche neben der Abbildung von Wissen die Möglichkeiten bieten, Wissen mit relevanten Informationen zu verknüpfen. Topic Maps ermöglichen eine themenzentrierte Sichtweise auf Informationen [Pepper und Moore 2010]. Das dazu notwendige Datenmodell ist im ISO-Standard 13250 festgehalten. Neben dem Datenmodell beschreibt der Standard auch eine Austauschsyntax, eine formale Semantik sowie eine grafische Notation von Topic Maps.

Die Kernkonzepte des Topic-Maps-Paradigmas sind wie folgt [Bouzid et al. 2012]:

- **Subject**: Ein Gegenstand, über den eine Aussage getätigt werden soll. Aussagegegenstände repräsentieren Konzepte aus der realen Welt.
- **Topic**: Aussagen über Gegenstände werden mit Hilfe von Topics getätigt. Topics sind somit Symbole für einen Aussagegegenstand. Das Integrationsmodell des Standards ermöglicht die Kollokation aller zu einem Subject zugehörigen Aussagen. Dies wird dadurch gewährleistet, dass jedes Topic nur ein einziges Subject repräsentieren darf [Dengel 2012, S. 95].
- **Name:** Der Name dient als Bezeichner des Aussagegegenstands.
- **Association** Mit Hilfe von Assoziationen können Beziehungen zwischen Topics hergestellt werden.
- **Occurrence**: Ausprägungen (engl. occurrences) dienen dazu, Verbindungen eines Topics mit den dazugehörigen Ressourcen zu realisieren.
- **Resource**: Eine Ressource stellt Informationen über einen Aussagegegenstand eines Topics zur Verfügung. Ressourcen können Dokumente, Webseiten etc. sein.

Abbildung 3.4 stellt die zuvor beschriebenen Konzepte der Topic-Map-Technologie anschaulich dar. Wie in dieser Darstellung verdeutlicht, ziehen Topic Maps eine strikte Trennung zwischen dem abgebildeten Wissen und den dazugehörigen Informationsquellen. Das durch die Verbindung unter den Topics entstehende Netz kann durch weitere Konzepte wie Gültigkeitsbereiche (engl. scopes), Typenbildung (engl. types) und Facetten (engl. facets) sogar noch weiter angereichert werden [Bouzid et al. 2012, S. 122]. Wie zu erkennen ist, sind Topic Maps in ihrer Ausdrucksfähigkeit mächtiger als die im vorherigen Abschnitt beschriebenen Thesauri.

Aufgrund des Integrationsmodells bieten die Topic Maps die Möglichkeit der semantische Zusammenführung verschiedener Informationsressourcen. Einsatz finden Topic Maps im Bereich des Wissensmanagements, der Publikation hoch vernetzter Webinhalte und des E-Learnings [Dengel 2012, S. 102 f.].

[2] DIN 1463-1 1987 und DIN 1463-2 1993.
[3] ISO/IEC 2788 1986 und ISO/IEC 5964 1985.

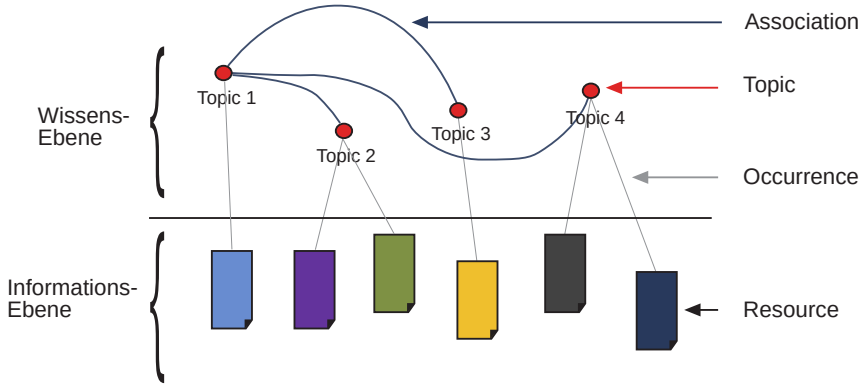


Abbildung 3.4.: Kernkonzepte des Topic-Maps-Paradigmas [Pepper und Moore 2010]

3.2.4. Ontologien

Der Begriff der *Ontologie* steht im Allgemeinen für die *Lehre vom Seienden*. Diese Disziplin der Philosophie sucht nach Möglichkeiten, die Realität korrekt und möglichst allgemeingültig zu beschreiben. Im Sinne der Informatik stammt die am häufigsten verwendete Definition des Begriffs der Ontologie von Gruber. Er versteht eine Ontologie als eine *„explizite Spezifizierung einer Konzeptualisierung“* [Gruber 1993]. Studer et al. erweiterten diese Definition um die Aspekte des Formalismus und der Gemeinsamkeit:

> *„An Ontology is a formal, explicit specification of a shared conceptualization“* [Studer et al. 1998]

Im Mittelpunkt dieser Definition steht die Explizitheit. Das darzustellende Wissen kann nur dann anderen Menschen zugänglich sein, wenn es *explizit* gemacht wurde. Mit Konzeptualisierung ist die Abbildung eines Wissensgebietes in ein abstraktes Modell gemeint. Dieses Modell muss in einer formalen Form vorliegen, damit es von einer Maschine interpretiert werden kann. Zudem sollte das Modell aus einem gemeinsamen Verständnis heraus entstanden sein. Das bedeutet: eine Ontologie kann nicht von einer Person alleine erschaffen werden, sondern sollte immer unter der Beteiligung und dem Einverständnis mehrerer Personen konstruiert werden [Hermans 2008, S. 50].

Eine Ontologie besteht im Allgemeinen aus den folgenden, teilweise schon zuvor beschriebenen Komponenten:

- **Klassen/Konzepte**: Klassen, die auch häufig als Konzepte bezeichnet werden, bilden verschiedene Begriffskategorien aus der betrachteten Domäne ab.
- **Relationen**: Relationen stellen Beziehungen zwischen den Klassen/Konzepten her.
- **Axiome**: Axiome sind Regeln über Konstellationen, die in einer Domäne gültig sind.
- **Instanzen**: Instanzen sind reale Objekte aus der betrachteten Domäne.

Abbildung 3.5 zeigt ein Beispielausschnitt einer Ontologie für Open-Source-Komponenten in Anlehnung an Hermans [2008]. Diese Ontologie beinhaltet die Konzepte *Person*, *Open-Source-Komponente*, *Programmiersprache* und die Spezialisierungen *Deklarative Sprache* und *Imperative Sprache*. Diese Konzepte sind mit Hilfe von Relationen in Beziehung zu einander gesetzt. Die Autorenschaft einer Personen bezüglich einer Open-Source-Komponente ist

beispielsweise durch die Relation *hat Autor* dargestellt. Ist eine Person ein Fachmann einer bestimmten Programmiersprache, so stehen diese beiden Konzepte über die Relation *beherrscht* in Beziehung zueinander. Konzepte werden über Attribute beschrieben. Jede Person hat beispielsweise einen Namen als Attribut. Die konkreten Ausprägungen eines Konzepts werden als Instanz bezeichnet. In diesem Fall ist *Jörg Müller* der Autor der Open-Source-Komponente *Freemind*, welche in Java implementiert ist. Anders als in objektorientierten Ansätzen wird eine Beziehung in einer Ontologie nicht in den Instanzen abgelegt, sondern eigenständig erzeugt [Hermans 2008, S. 51]. Die jeweiligen Relationen auf Instanzebene sind also eigenständige Instanzen der Verbindungen aus der Konzeptebene. Zusätzlich beinhaltet die dargestellte Ontologie zwei Axiome. Mit Hilfe dieser Axiome kann zusätzliches Wissen inferiert werden. Die erste Regel lässt die Schlussfolgerung zu, dass wenn eine Komponente x einen Autor y hat, dann ist y ein Fachmann für die Komponente x.

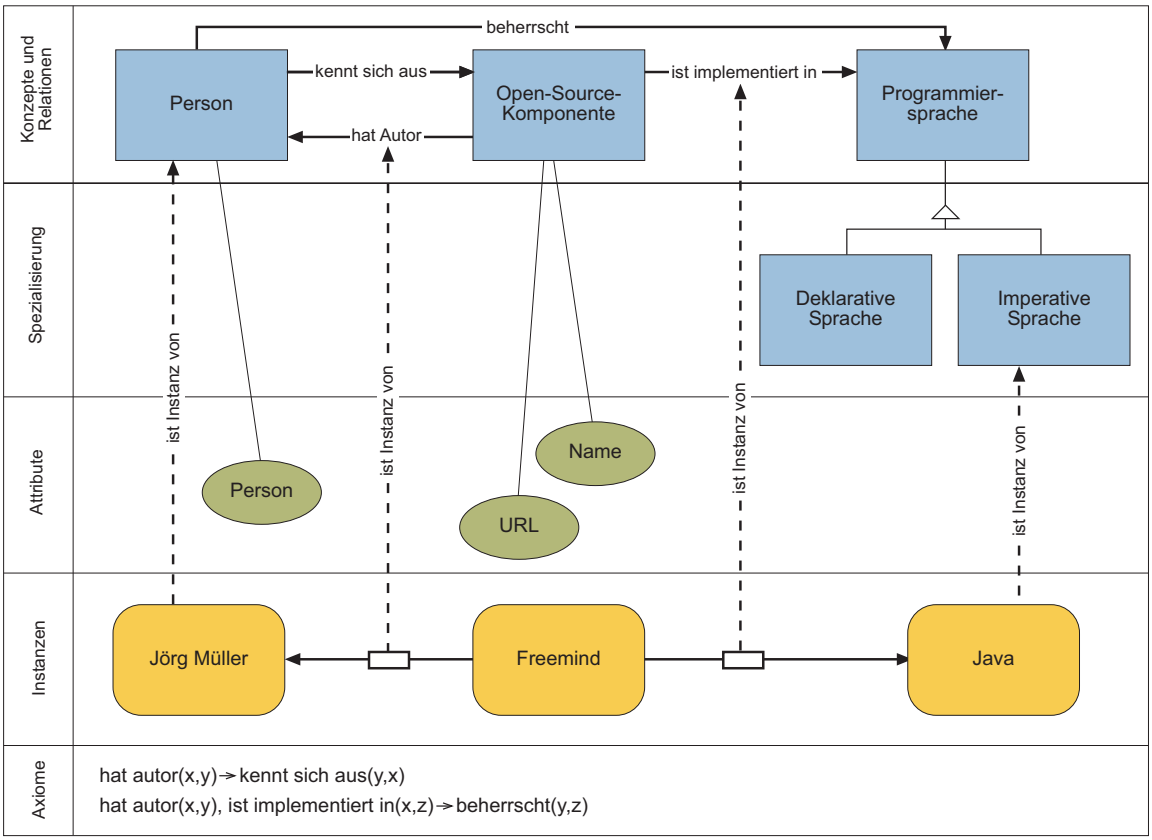


Abbildung 3.5.: Ausschnitt einer Ontologie für Open-Source-Komponenten [Hermans 2008, S. 51]

Die Anwendungsfelder von Ontologien sind nach Grunniger und Lee in den Bereichen der Kommunikation, dem computerbasierten Schließen sowie der Repräsentation und Wiederverwendung von Wissen zu sehen [Grunniger und Lee 2002]. Im Bereich der Kommunikation können Ontologien dazu eingesetzt werden, den strukturierten Austausch zweier Kommunikationsteilnehmer zu ermöglichen (z.B. in serviceorientierten Architekturen). Im Rahmen der Repräsentation und Wiederverwendung von Wissen dienen Ontologien zur expliziten und formalen Beschreibung eines Wissensgebietes [Hermans 2008, S. 50]. Aufgrund der höheren Ausdrucksstärke kann mit Hilfe einer Ontologie durch computerbasiertes Schließen neues Wissen geschlussfolgert werden. Die zuvor behandelten Modelle bieten diese Möglichkeit nicht [Miles und Bechhofer 2009]. Mit Blick auf das semiotische Dreieck (siehe Abschnitt 3.1),

verdeutlichen Daconta et al. ein weiteres wichtiges Unterscheidungsmerkmal, das Ontologien von den zuvor beschrieben Modellen abhebt [Daconta et al. 2003, S. 210]. So ist beispielsweise das Modell des Thesaurus auf der rechten Seite dieses Dreiecks einzuordnen. Dieses Modell verknüpft Symbole mit ihrer Bedeutung auf der Konzeptebene. Eine Ontologie hingegen hat zum Ziel, eine Verbindung zwischen der Konzeptebene und den Objekten aus der realen Welt zu schaffen. Ein prominenter Vertreter zur Dokumentation einer Ontologie ist die Ontologiesprache OWL. Eine genauere Betrachtung dieses Semantic-Web-Standards erfolgt im nachfolgenden Abschnitt.

3.3. Das Semantic Web

Dieser Abschnitt soll eine detaillierte Beschreibung der Ziele und der technologischen Grundlagen des Semantic Web geben. Dazu wird zunächst die Vision des Semantic Web geschildert. Anschließend werden die Standards, die auf Grundlage dieser Vision entwickelt wurden, erörtert. Hierzu wird in Abschnitt 3.3.2 der sogenannte *Semantic Web Stack* herangezogen. Dieser Stack veranschaulicht die Schichtenarchitektur des Semantic Web und zeigt das Zusammenspiel der einzelnen Semantic-Web-Standards. Zuletzt werden die für diese Ausarbeitung relevanten Standards im Detail erläutert.

3.3.1. Vision des Semantic Web

Im Jahre 2001 veröffentlichten Berners-Lee et al. einen damals revolutionären Artikel im Scientific American. In diesem Artikel wurde erstmals die Vision des Semantic Web wie folgt definiert:

> *„The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.“* [Berners-Lee et al. 2001, S. 25]

Dieser Auszug beinhaltet die grundsätzliche Idee des Semantic Web. Berners-Lee et al. sahen in ihm eine Erweiterung des bestehenden Web. In dieser Erweiterung sollen Informationen mit einer wohldefinierten Bedeutung, also Semantik, versehen werden. Mit Hilfe dieser wohldefinierten Bedeutung sollen die im Web vorhandenen Informationen durch Computer interpretierbar gemacht werden. Das Ziel der Autoren bestand darin, die Zusammenarbeit zwischen Mensch und Maschine zu verbessern. Dieser Ansatz steht also im Gegensatz zu dem momentan vorherrschenden Web, welches als globaler Dokumentenspeicher für den Zugriff durch Menschen konzipiert worden ist. In diesem Web steht im Mittelpunkt, Informationen für Menschen verfügbar zu machen und diese Informationen einfach und schnell zu veröffentlichen sowie miteinander verknüpfen zu können [Domingue et al. 2011, S.5]. Breitman et al. bezeichnen dieses Web auch als das *syntaktische Web*, in dem die Interpretation und Identifikation von relevanten Informationen erst durch den Menschen erfolgt [Breitman et al. 2007, S.4].

Die Anreicherung von Daten mit maschinenlesbaren Informationen und deren automatisierte Interpretation stellt ein grundlegendes Konzept des Semantic Web dar. Ein weiterer Grundbaustein ist die halb- bzw. vollautomatische Extraktion dieser Informationen aus semistrukturierten oder unstrukturierten Datenbeständen. Die Verteiltheit von Informationen in großen Mengen kann ebenso als Grundkonzept genannt werden. Wie schon im vorherigen Abschnitt 3.1 angedeutet, stellen diese Konzepte einzeln betrachtet keine Neuheit dar. Neu hingegen ist jedoch die Kombination dieser Konzepte [Pellegrini und Blumauer 2006, S. 2 f.].

3.3.2. Architektur des Semantic Web

Auf Basis der von Berners-Lee et al. beschriebenen Vision haben sich im Laufe der Zeit einige neue Standards herausgebildet. Diese Standards werden unter der Federführung des W3C-Standardisierungsgremiums (World Wide Web Consortium) als sogenannte Empfehlungen (engl. Recommendations) herausgegeben. Diese Empfehlungen definieren allgemein akzeptierte und herstellerunabhängige Normen und bilden einen wichtigen Bestandteil der semantischen Technologien. Sie sind die Basis zur Umsetzung der zuvor aufgezeigten Konzepte des Semantic Web. Die einzelnen Standards bauen dabei schichtartig aufeinander auf. Abbildung 3.6 zeigt diese Schichtenarchitektur des Semantic Web Stack.

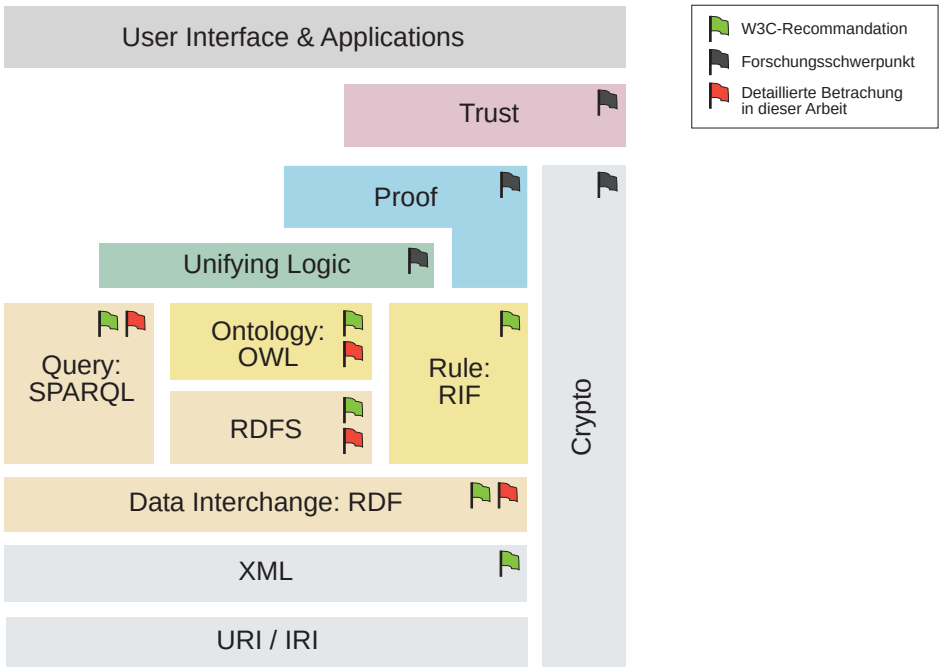


Abbildung 3.6.: Semantic Web Stack [W3C 2012]

Nicht alle Schichten bilden bis dato einen offiziellen Standard. Sie stellen aktuell noch Forschungsschwerpunkte dar. Diese Tatsache zeigt mit auf, dass die Vision des Semantic Web immer noch nicht vollständig Realität geworden ist. Die noch nicht standardisierten Schichten sind in der Abbildung 3.6 mit einer grauen Flagge gekennzeichnet. Verabschiedete W3C-Empfehlungen werden durch eine grüne Flagge hervorgehoben. Die im Rahmen dieser Arbeit detailliert betrachteten Schichten sind wiederum mit einer roten Flagge gekennzeichnet und werden in den nachfolgenden Abschnitten dargestellt. Die Rolle der nicht im Detail behandelten Schichten wird im Folgenden kurz aufgezeigt, um das Gesamtbild der Architektur zu vervollständigen.

Der *Uniform Ressource Identifier* (URI) bildet die unterste Schicht der Semantic-Web-Architektur. Dieser RFC-Standard (Request for Comments) stellt eine Basistechnologie des Stack dar und ist kein eigentlicher Bestandteil dessen. URIs sind Zeichenketten, die zur eindeutigen Identifikation beliebiger Ressourcen dienen. Eine Ressource kann entweder ein konkretes Element aus der realen Welt sein, oder aber ein abstraktes Objekt darstellen. Gerade im WWW werden URIs dazu verwendet, um Web-Seiten und die darin befindlichen Ressourcen, wie Dateien oder Bilder, zu adressieren [Hitzler et al. 2008, S. 26]. Der in der gleichen Schicht

dargestellte Standard des *Internationalized Resource Identifier* (IRI) ist eine Erweiterung des URI-Standards. Innerhalb eines IRI kann der vollständige Unicode-Zeichensatz zur eindeutigen Identifikation von Ressourcen verwendet werden.

In der zweiten Schicht ist die *eXtensibe Markup Language* (XML) angesiedelt. XML ist eine Auszeichnungs- und Metasprache, welche ein maschinenlesbares Dokumentenformat definiert. Ebenso wie die URI-Schicht ist XML eine Basistechnologie innerhalb des Semantic-Web-Architektur. Gerade im Kontext der Metadatenbeschreibung findet dieses Format häufig Verwendung[4]. Mit XML ist es nicht möglich eine Semantik maschinenlesbar zu kodieren, sie dient aber innerhalb der Architektur als syntaktische Grundlage von Sprachen aus höheren Schichten.

Neben den beiden zuvor erörterten Basistechnologien sind weitere Schichten vorgesehen. Die *RIF-Schicht* (Rule Interchange Format) definiert einheitliche Formate zum Austausch von Regeln und ist seit 2010 eine Empfehlung des W3C. Die *Unifying-Logic-Schicht* ist ein in der Entwicklung befindlicher Standard. Dieser Standard soll das Semantic Web um die Möglichkeit erweitern, Regeln zu definieren, welche von Anwendungen verarbeitet werden können. Die *Proof-Schicht* soll dazu dienen, die logischen Schlussfolgerungen der Maschine für den Menschen nachvollziehbar zu machen. *Trust* soll den Stack um Aspekte der Sicherheit wie Authentifizierung und Authentizität erweitern. Dies hat zum Ziel, die Informationen des Semantic Web auf ihre Gültigkeit und Vertrauenswürdigkeit hin überprüfen zu können. Hier sollen Technologien wie Digitale Signaturen und Kryptographie eingesetzt werden. Die Schicht *User Interface & Application*, als oberstes Element der dargestellten Architektur, nimmt eine Sonderrolle ein. Sie stellt die Verbindung zwischen den semantischen Technologien in den unteren Schichten und dem Nutzer her. Mit Hilfe von Benutzerschnittstellen und Anwendungen wird es dem Nutzer ermöglicht, semantische Anwendungen verwenden zu können. Diese Schicht ist nicht Bestandteil der eigentlichen W3C-Spezifikationsbestrebungen, sondern Ergebnis der Umsetzung dieser Standards in Form von Anwendungen.

3.3.3. Resource Description Framework

Modell und Syntax

Das Resource Description Framework ist eine Spezifikation des W3C, welches ursprünglich zur Anreicherung von Web-Dokumenten mit Metadaten vorgesehen war [Hitzler et al. 2008, S. 35]. RDF in der heutigen Form stellt ein Grundgerüst zur Beschreibung von beliebigen Ressourcen und deren Beziehungen zur Verfügung. RDF befindet sich in Schicht drei der Semantic-Web-Architektur. Die grundlegenden Konzepte des *RDF-Modells* sind *Ressourcen*, *Eigenschaften* und *Aussagen* [Antoniou und Van Harmelen 2004, S. 63 f.]:

- **Ressource:** Jedes Element, über das eine Aussage getätigt werden soll, ist eine Ressource. Jede Ressource muss durch ein URI eindeutig identifiziert werden können. Der URI muss dabei nicht zwingend auf die tatsächliche Ressource verweisen.
- **Eigenschaft:** Jede Ressource hat bestimmte Eigenschaften, die sie charakterisieren. Mittels einer Eigenschaft können Beziehungen (z.B. *geschrieben von*, *Alter* oder *Titel*) zwischen Ressourcen ausgedrückt werden. Eigenschaften werden ebenfalls mittels eines URI eindeutig identifiziert.

[4] Die Metadatenbeschreibungen der Dublin-Core-Initiative können beispielsweise mit Hilfe von XML dargestellt werden.

- **Aussage:** Mit Hilfe von Aussagen können physikalische oder abstrakte Objekte innerhalb einer Domäne beschrieben werden. Eine Aussage ist ein *RDF-Tripel*, das in der Form Subjekt-Prädikat-Objekt (*S-P-O*) definiert wird. Das Subjekt ist die Ressource, über die eine Aussage gemacht wird. Das Prädikat wird durch die Eigenschaft dargestellt. Das Objekt wiederum ist entweder eine Ressource oder ein Literal. Literale stellen einfache Datentypen innerhalb von RDF dar. Aussagen können selbst Ressourcen bilden, über die wiederum eine Aussage getroffen werden kann. Wenn eine Aussage über eine Aussage getätigt wird, so wird dies auch als *Reifikation* bezeichnet [Antoniou und Van Harmelen 2004, S. 67].

Im Folgenden soll ein Beispiel dafür gegeben werden, in welcher Art und Weise eine Aussage mit Hilfe der *RDF-Syntax* notiert werden kann. Dazu wird ein einfach gehaltenes Beispiel betrachtet [Hebeler et al. 2009, S. 69]:

> *Andrew knows Matt. Andrew's surname is Perez Lopez.*

Eine solche Aussage kann im Rahmen von RDF mit Hilfe verschiedener Notationsformen dargestellt werden. Die für den Menschen verständlichste Form der Notation von RDF-Aussagen ist die Visualisierung mit Hilfe eines gerichteten Graphen. Abbildung 3.7 zeigt eine solche Visualisierung der obigen Aussage mit Hilfe eines RDF-Graphen.

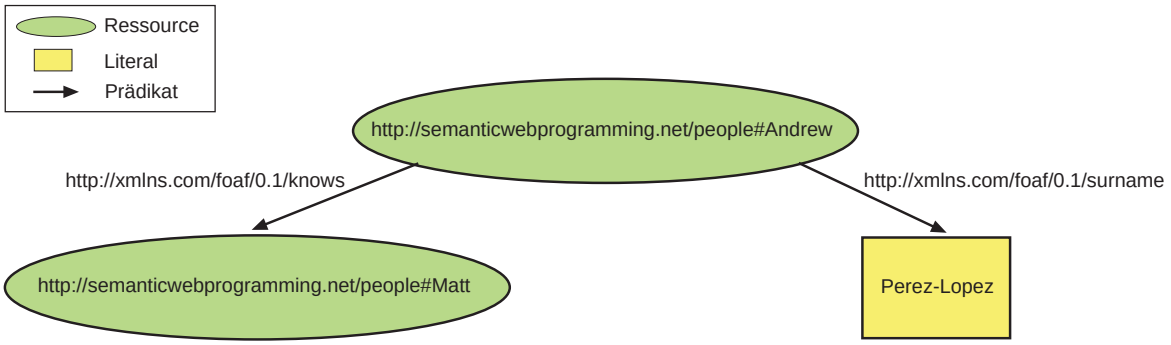


Abbildung 3.7.: RDF-Aussage in Form eines gerichteten Graphen [Hebeler et al. 2009]

Die in Abbildung 3.7 dargestellten Elemente folgen den üblichen Konventionen zur Visualisierung von RDF-Graphen. Ressourcen (hier *Matt* und *Andrew*) werden mit Hilfe eines Ovals dargestellt. Literale (hier *Perez-Lopez*) sind als Rechtecke abgebildet und Prädikate als gerichtete Kanten (*knows* und *surname*). Die URIs der Ressourcen und der Eigenschaften sind ebenfalls in der Abbildung aufgeführt. Häufig werden diese aufgrund der Übersichtlichkeit nicht dargestellt. Die Ressource *Andrew* kann beispielsweise eindeutig über die URI `http://semanticwebprogramming.net/people#Andrew` identifiziert werden. Das Prädikat zum Subjekt *Andrew* gibt eine Eigenschaft vom Typ *knows* aus dem FOAF-Namensraum[5] an. Das Objekt dieses Tripels ist in diesem Fall wiederum eine Ressource. Das Literal *Perez-Lopez* ist ein Beispiel für ein Literal in der Rolle des Objekts.

Neben der Visualisierung als Graph gibt es mehrere Möglichkeiten zur Serialisierung von RDF-Aussagen. So besteht die Möglichkeit, Aussagen als Ansammlung von RDF-Tripeln zu notieren. Zu nennen ist hier vor allem die moderne Turtle-Syntax[6]. Listing 3.1 zeigt wie das obige Beispiel mit Hilfe dieser Turtle-Syntax notiert werden kann.

[5] Das Friend of a Friend (FOAF) Projekt hat zum Ziel, ein maschinenlesbares Web von Dokumenten zu erzeugen, in dem Menschen und deren Beziehungen untereinander beschrieben werden.

[6] http://www.w3.org/TR/turtle/ (Zugriffsdatum: 07.01.2013)

```
1 @prefix foaf:   <http://xmlns.com/fof/0.1/> .
2 @prefix people: <http://semanticwebprogramming.net/people/> .
3
4 people:Andrew foaf:knows people:Matt .
5 people:Andrew foaf:surname "Perez-Lopez" .
```

Listing 3.1: RDF-Aussage in Turtle-Syntax [Hebeler et al. 2009]

Innerhalb der Turtle-Notation können eigene Namensräume (engl. namespaces) angegeben werden (Zeile 1 und 2). So wird mit Hilfe des `@prefix`-Befehls unter anderem der Namensraum des FOAF-Projekts deklariert. Die Kombination aus `prefix:name` wird auch als *qualifizierter Name* (engl. qualified name) bezeichnet [Hitzler et al. 2008, S. 41]. RDF-Aussagen werden in Turtle als Subjekt-Prädikat-Objekt innerhalb einer Zeile notiert und durch ein Leerzeichen voneinander getrennt (Zeile 4 bis 5). Ein Punkt terminiert jeweils eine einzelne Aussage.

Neben der für den Menschen gut lesbaren Tripel-Darstellung ist die am weitesten verbreitete Serialisierung von RDF-Aussagen die XML-basierte Notationsform. Dies liegt nach Hitzler et al. darin begründet, dass im Gegensatz zu tripelbasierten Notationen für XML in nahezu jeder Programmiersprache eine Vielzahl von Programmierbibliotheken existieren, welche das Verarbeiten von RDF-Aussagen erleichtern [Hitzler et al. 2008, S. 42]. Listing 3.2 zeigt die zuvor in Turtle notierten Aussagen in der XML-Notation.

```
1 <rdf:RDF
2     xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:foaf="http://xmlns.com/foaf/0.1/"
4     xmlns:people="http://semanticwebprogramming.net/people#">
5     <rdf:Description rdf:about="http://semanticwebprogramming.net/people#Andrew
          ">
6         <foaf:knows rdf:about="http://semanticwebprogramming.net/people#Matt"/>
7         <foaf:surname>Perez-Lopez</foaf:surname>
8     </rdf:Description>
9 </rdf:RDF>
```

Listing 3.2: RDF-Aussage in XML-Syntax [Hebeler et al. 2009]

Auch in Listing 3.2 werden zunächst mehrere Namensräume importiert (Zeile 2 bis 4). Hervorzuheben ist, dass die Definition der RDF-Syntax selbst ebenfalls über einen Namensraum referenziert wird. Innerhalb der Semantic-Web-Gemeinschaft besteht dabei die Konvention, dass dieser Namensraum immer mit `rdf` abgekürzt wird [Hebeler et al. 2009, S. 75]. Aussagen in der XML-Notation werden mit Hilfe des `<rdf:Description>`-Elements gruppiert. Das `rdf:about`-Element gibt das Subjekt des RDF-Tripels an. Das Kind-Element in Zeile 6 stellt wiederum das Prädikat und das dazugehörige Objekt dar. Zeile 7 zeigt die XML-Notation des Literals `<foaf:surname>`.

Neben den hier kurz angerissenen Elementen der RDF-Syntax existieren noch weitere Sprachelemente, mit deren Hilfe umfangreiche Strukturen abgebildet werden können. Auf eine vollständige Darstellung dieser Syntax wird an dieser Stelle verzichtet. Weiterführende Informationen sind in der entsprechenden W3C-Empfehlung zu finden [Klyne und Carroll 2004]. Mittels des zuvor behandelten RDF-Modells und der RDF-Syntax ist es möglich, komplexe Beschreibungen über Ressourcen einer Domäne mit Hilfe eines eigenen Vokabulars zu erstellen. Jedoch sind diese RDF-Beschreibungen für den Computer nichts weiter als eine Folge von Zeichen ohne feste Bedeutungen. RDF alleine reicht also nicht aus, um die Semantik einer Do-

mäne zu erfassen [Antoniou und Van Harmelen 2004, S. 80]. Dafür ist das im nachfolgenden Abschnitt beschriebene RDF Schema (RDFS) vorgesehen.

Semantik mit RDF Schema

RDF Schema ist in der vierten Schicht der Semantic-Web-Architektur zu finden (siehe Abbildung 3.6). RDF Schema ist dazu gedacht, semantische Zusammenhänge einer Domäne abzubilden. Das heißt, mit RDFS ist es möglich, terminologisches Wissen (auch Schemawissen genannt) über Begriffe aus einem Vokabular zu erfassen. Dieses Schemawissen wird dabei selber mittels eines speziellen RDF-Vokabulars spezifiziert. Durch diese Eigenschaft zählt RDFS zu den Wissensrepräsentationssprachen, mit deren Hilfe eine Ontologie abgebildet werden kann. Da die Mächtigkeit von RDFS begrenzt ist, bezeichnen Hitzler et al. RDFS als leichtgewichtige Ontologiesprache [Hitzler et al. 2008, S. 67].

Um Schemawissen ausdrücken zu können, beinhaltet RDFS vordefinierte Konzepte zur Spezifikation von Klassen, Ressourcen, Eigenschaftshierarchien und deren Zusammenhängen. Diese Konzepte können innerhalb eines RDFS-Dokuments zur Definition des jeweiligen Schemawissens genutzt werden. Mit Hilfe eines RDFS-Dokuments kann dann eine Typisierung von RDF-Ressourcen erfolgen. Diese kombinierte Verwendung von RDF zusammen mit RDFS wird häufig auch mit der Abkürzung RDF(S) veranschaulicht. Im Folgenden sollen einige Sprachelemente zur Definition eines RDF Schema beschrieben werden. Anzumerken ist, dass diese nur einen Teil des RDFS-Sprachumfangs darstellen. Eine vollständige Referenz ist in der entsprechenden W3C-Empfehlung festgehalten [Brickley und Guha 2004].

Unter anderem bietet RDFS folgende *Kern-Klassen*:

- `rdfs:Resource`: klassifiziert ein Element als Ressource
- `rdfs:Class`: dient zur expliziten Auszeichnung eines Elements als Klasse[7]
- `rdf:Property`: Mit Hilfe dieser Klasse wird eine Ressource ausgezeichnet, die eine Beziehung (Eigenschaft) zu anderen Elementen ausdrückt
- `rdf:Statement`: repräsentiert die Klasse aller reifizierten Aussagen

Daneben sind auch *Kern-Eigenschaften* definiert, mit denen *Beziehungen* ausgedrückt werden können. Dazu gehören beispielsweise:

- `rdf:type`: dient zur Zuordnung einer Ressource zu einer Klasse
- `rdfs:subClassOf`: zeichnet eine Klasse als Unterklasse einer anderen Klasse aus
- `rdfs:subPropertyOf`: erlaubt die Vererbung auf der Ebene von Eigenschaften

Um Eigenschaften von Ressourcen einzuschränken, sind weiterhin folgende *einschränkende Kern-Eigenschaften* vorgesehen:

- `rdfs:domain`: erlaubt die Einschränkung des Definitionsbereichs eines Subjekts innerhalb einer RDF-Aussage

[7] Diese explizite Auszeichnung ist wichtig, da eine Unterscheidung zwischen einer Instanz und einer Klasse sonst nicht möglich wäre.

- `rdfs:range`: typisiert das Objekt einer RDF-Aussage im Zusammenhang mit einem bestimmten Prädikat

Abbildung 3.8 zeigt beispielhaft die Verwendung einiger zuvor erläuterten Sprachelemente. Daneben veranschaulicht diese Grafik die unterschiedlichen Ebenen von RDF und RDFS. Das in RDFS spezifizierte Schemawissen ist in der oberen Hälfte der Abbildung dargestellt.

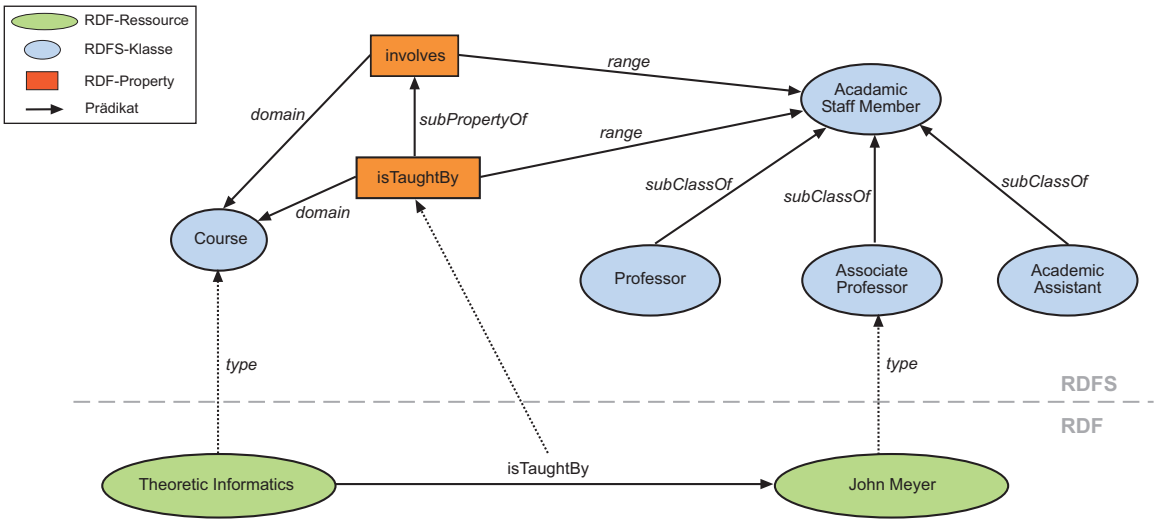


Abbildung 3.8.: Beispiel-Graph für RDF- und RDFS-Ebenen [Antoniou und Van Harmelen 2004]

Der Graph bildet (ausschnittsweise) die Domäne einer Universität mit ihren Vorlesungen ab. Dazu werden RDFS-Klassen und Properties definiert, die in der RDF-Ebene konkrete Instanzen typisieren bzw. als Prädikat eines Tripels genutzt werden. So werden aus einer abstrakten Vorlesung konkrete Veranstaltungen erzeugt. Die RDF-Ressource *Theoretic Informatics* ist beispielsweise vom Typ *Course* und somit als Veranstaltung typisiert. Die Eigenschaft *isTaugthBy* aus der RDFS-Ebene wird als Prädikat des dargestellten RDF-Tripels verwendet. Das Subjekt *John Meyer* wiederum ist als *Associate Professor* klassifiziert. Innerhalb der RDFS-Ebene wird die *isTaugthBy*-Eigenschaft auf Subjekte vom Typ *Course* beschränkt (mittels `rdfs:domain`). Gleichzeitig findet eine Eingrenzung des dazugehörigen Objekts der *isTaugthBy*-Eigenschaft auf die Klasse *Academic Staff Member* mittels `rdf:range` statt. Insgesamt ergibt dies die Bedeutung, dass nur Vorlesungen unterrichtet werden können und lediglich akademische Mitarbeiter diese Vorlesungen halten dürfen.

Insgesamt betrachtet liefert RDFS zwar bereits einige Sprachkonstrukte, um die Semantik einer Domäne ausdrücken zu können, trotzdem ist dessen Umfang beschränkt. Es ist beispielsweise nicht mögliche, negative Aussagen zu formulieren [Hitzler et al. 2008, S. 119]. Weiterhin erlaubt RDFS nur einfache Klassenhierarchien abzubilden. Mengenbeziehungen wie z.B. die explizite Disjunktion oder die Durchschnittsbildung von Klassen ist nicht möglich [Antoniou und Van Harmelen 2004, S. 111]. Diese und weitere Einschränkungen führten zur Einführung von OWL, welche im nachfolgenden Abschnitt erläutert wird.

3.3.4. Web Ontology Language

Die Ontologiesprache *Web Ontology Language* (OWL) wurde im Jahr 2004 vom W3C als Standard verabschiedet [van Harmelen und McGuinness 2004]. OWL ist eine Erweiterung

von RDFS und befindet sich im Schichtenmodell des Semantic Web oberhalb dieser Beschreibungssprachen. Um stärkere Ausdrucksmittel zur Verfügung zu haben, erweitert OWL zunächst das Vokabular von RDFS um neue Klassen und Prädikate. Ziel war es, Terminologien für komplexere Zusammenhänge bereitzustellen, eine bessere Einschränkungen von Eigenschaften und deren logischen Eigenheiten definieren zu können, sowie die Äquivalenz von Begrifflichkeiten zu ermöglichen. Dazu wird das RDF Schema unter anderen um Konstrukte erweitert, welche es durch

- Aufzählung von Instanzen,
- Durchschnittsbildung mit anderen Klassen (alles, was in Klasse A und B gleichzeitig ist),
- Vereinigung mit anderen Klassen (alles, was in Klasse A oder B ist),
- die Angabe von Kardinalitäten (z.B. n Instanzen aus Klasse A) und
- Komplementäroperationen (alles, was nicht in Klasse A ist)

ermöglichen, neue Klassen zu konstruieren [Pellegrini und Blumauer 2006, S. 83 ff.]. Weiterhin ist es möglich, Eigenschaften dahingehend einzuschränken, dass neben dem Domänen- und Wertebereich auch das Verhalten spezifiziert werden kann. Für ein Prädikat p des RDF-Tripels können folgende Verhalten angegeben werden:

- p ist transitiv: wenn $(a\ p\ b)$ und $(b\ p\ c)\ \Rightarrow\ (a\ p\ c)$
- p ist funktional: wenn $(a\ p\ b)$ und $(a\ p\ c)\ \Rightarrow\ b = c$
- p ist invers funktional: wenn $(a\ p\ b)$ und $(c\ p\ b)\ \Rightarrow\ a = c$
- p ist symmetrisch: $(a\ p\ b)\ \Rightarrow\ (b\ p\ a)$
- p ist invers zu q: $(a\ p\ b) \Rightarrow (b\ q\ a)$

Weiterhin unterscheidet OWL zwischen Objekt- und Datentyp-Eigenschaften. Objekteigenschaften (`owl:ObjectProperty`) stellen eine Verbindung von Objekten zu anderen Objekten her. Datentyp-Eigenschaften (`owl:DatatypeProperty`) sind Verbindungen zwischen Objekten und Datentypen [Antoniou und Van Harmelen 2004, S. 118].

Ontologiesprachen wie OWL beruhen auf formalen Semantiken. Diese formalen Semantiken ermöglichen das Schlussfolgern neuen Wissens. Im Fall von OWL bilden modelltheoretische Ansätze der Beschreibungslogik die Basis zur Interpretation der Sprachkonstrukte [Troncy et al. 2011, S. 102]. Grundlage von OWL ist dabei die Beschreibungslogik $\mathcal{SHOIN}(\mathbf{D})$. Eine hohe Ausdrucksstärke von Sprachen wie OWL führt zwangsläufig zu dem Problem einer hohen Komplexität des Schlussfolgerungsmechanismus. Je nach Mächtigkeit kann dies auch zur Unentscheidbarkeit einer Sprache führen. Aufgrund dieser Tatsache spielte bei der Standardisierung von OWL das Gleichgewicht zwischen Ausdrucksstärke und effizientem Schlussfolgern (und somit Skalierbarkeit) eine wichtige Rolle. Es musste also das Problem gelöst werden, eine hohe Ausdrucksmächtigkeit zu ermöglichen und dennoch effizient schlussfolgern zu können. Das W3C-Konsortium löste dieses Problem, indem es dem Anwender überlassen wird, je nach Praxisanforderung eine entsprechende Ausdrucksstärke zu nutzen [Hitzler et al. 2008, S. 125 f.]. OWL besteht dazu aus drei verschiedenen Teilsprachen: OWL Full, OWL DL und OWL Lite. Abbildung 3.9 zeigt die Teilmengenbeziehungen zwischen diesen Teilsprachen.

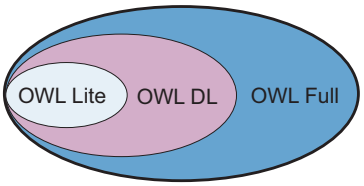


Abbildung 3.9.: Mengenbeziehungen der OWL-Teilsprachen

In *OWL Full* können alle OWL- sowie RDF(S)-Sprachelemente verwendet werden. Durch diesen mächtigen Sprachumfang ist das Schlussfolgern unentscheidbar. Es ist also möglich, dass ein Algorithmus, welcher auf einer OWL-Full-basierten Ontologie angewendet wird, keine Antwort zu einer Anfrage liefert. *OWL DL* hingegen ist entscheidbar. Die Abkürzung DL steht für *description logic*. Dies bringt die Verbindung von OWL und der Beschreibungslogik zum Ausdruck. Um die Entscheidbarkeit zu erhalten, erlaubt OWL DL nur bestimmte Elemente aus RDF(S) und schreibt eine strikte Typentrennung sowie -Deklaration vor. *OWL Lite* ist wiederum eine Teilmenge von OWL DL. In dieser Teilsprache existieren weitere Einschränkungen. Ziel von OWL Lite ist es, ein einfach zu implementierendes Sprachfragment zur Verfügung zu stellen. Da die Ausdrucksstärke von OWL Lite jedoch stark begrenzt ist, wird sie kaum in der Praxis verwendet. Die Laufzeit-Komplexität dieses Sprachfragments ist nochmals geringer [Hitzler et al. 2008, S. 125 ff.].

Die erste Version des OWL-Standards wurde im Jahre 2009 von der OWL2-Spezifikation abgelöst [Hitzler et al. 2012]. Die neue Spezifikation ist in erster Linie eine Erweiterung und Überarbeitung der zuvor beschriebenen Web Ontology Language. Anders als ihr Vorgänger, basiert OWL2 auf der Beschreibungslogik $\mathcal{SROIQ}(\mathbf{D})$. Weiterhin enthält diese Empfehlung auch mehrere neue Sprachprofile[8], die für unterschiedlichste Einsatzzwecke zugeschnitten sind [Hitzler et al. 2012].

3.3.5. SPARQL

Die zuvor beschriebenen Schichten des Semantic Web Stack beschäftigen sich mit der maschinenlesbaren Codierung von Informationen und deren Semantik. Die Informationen werden mit RDF(S) bzw. OWL formal abgebildet und können innerhalb einer Wissensbasis genutzt werden. Mit Hilfe von Schlussfolgerungsverfahren ist es möglich, neue Informationen zu gewinnen. Schlussfolgerungsverfahren alleine sind aber zur Informationsgewinnung unzureichend. Um komplexere Informationen auslesen zu können, bedarf es spezieller Anfragesprachen. Wie im Stack dargestellt, ist für diesen Zweck die *SPARQL Protocol And RDF Query Language* (SPARQL) vorgesehen [Hitzler et al. 2008, S. 201 f.].

SPARQL definiert die Syntax und Semantik von Anfragen über RDF-Aussagen. Daneben beinhaltet die Empfehlung auch eine Protokoll zur Übertragung von Anfragen[9] und eine Definition zur Ausgabeformatierung[10]. SPARQL ist eine graphbasierte Anfragesprache. Das bedeutet, dass sie sich auf RDF-Graphen als zugrundeliegendes Modell stützt. Um eine Anfrage durchzuführen, wird die zuvor beschriebene Turtle-Syntax verwendet. Listing 3.3 zeigt ein Beispiel einer SPARQL-Anfrage, die sich auf den zuvor dargestellten RDF-Graphen bezieht

[8] OWL2-EL, OWL2-QL & OWL2-RL
[9] SPARQL Protocol for RDF
[10] SPARQL Query Results XML Format

(siehe Abbildung 3.8).

```
1 PREFIX ex:  <http://example.org/>
2 SELECT ?Course
3 WHERE
4     {?Course ex:isTaughtBy ex:JohnMeyer .}
```

Listing 3.3: Beispiel einer einfachen SPARQL-Anfrage

Eine Anfrage besteht immer aus drei Elementen, die durch die Schlüsselwörter `PREFIX`, `SELECT` und `WHERE` markiert werden. `PREFIX` dient zur Angabe des Namensraumes (Zeile 1). Durch `SELECT` kann das Ausgabeformat gewählt werden (Zeile 2). Hier kann beispielsweise neben einer einfachen Tabelle auch eine RDF-Repräsentation des Ergebnisses angefordert werden. Mit Hilfe von `WHERE` in Zeile 3 wird die Anfrage spezifiziert. Innerhalb der geschweiften Klammer befindet sich anschließend das Graph-Muster in Form der Turtle-Syntax (Zeile 4). Neben den aufgeführten Sprachelementen gibt es noch eine Reihe weiterer Konstrukte zum Filtern der Anfragen und Modifizieren der Ausgabe. Die formale Semantik dieser Anfragen ist in der sogenannten *SPARQL-Algebra*[11] festgehalten. Mit Hilfe dieser Algebra wird das Ergebnis jeder denkbaren Anfrage spezifiziert [Hitzler et al. 2008, S. 202 ff.].

[11] http://www.w3.org/TR/rdf-sparql-query/#sparqlAlgebra (Zugriffsdatum: 20.01.2013)

4. Semantische Suche

Die in den vorherigen Kapiteln beschriebenen Repräsentationsarten zur Abbildung semantischen Wissens sowie die Standards des Semantic Web bilden die technologischen Grundlagen dafür, Informationen für Maschinen interpretierbar zu machen. In diesem Kapitel wird nun dargelegt, was unter dem Prinzip der semantischen Suche zu verstehen ist und welche Rolle diese Technologien bei dieser Art der Informationssuche spielen. Zunächst wird dazu eine Definition des Begriffs der semantischen Suche gegeben und das Ziel dieses Suchansatzes im Vergleich zum IR herausgestellt. Danach findet die Darstellung einer allgemeinen Architektur statt. Zuletzt wird ein Überblick über semantische Suchverfahren gegeben. Dazu werden unterschiedliche Kategorisierungen vorgestellt, um aufzuzeigen, welche Ansätze zur semantischen Suche in der Literatur zu finden sind. Diese Darstellungen dienen anschließend im Rahmen der Konzeption als Grundlage zur Erstellung einer semantischen Suche.

4.1. Begriffsdefinition & Ziel

Der Begriff der semantischen Suche wird in Anlehnung an Dengel wie folgt definiert:

> *„Die semantische Suche beschreibt einen Suchprozess, in dem in einer beliebigen Phase der Suche formale Semantiken verwendet werden.“* [Dengel 2012, S. 232]

Mit Suchphasen sind die einzelnen Phasen des klassischen IR gemeint (siehe Abschnitt 2.4). Wenn in einer dieser Phasen formale Semantik zum Einsatz kommt, so kann von einer semantischen Suche gesprochen werden. Abbildung 4.1 gibt hierbei eine Einordnung verschiedener Standardformate zur Repräsentation von Informationen und deren Grad an formaler Semantik.

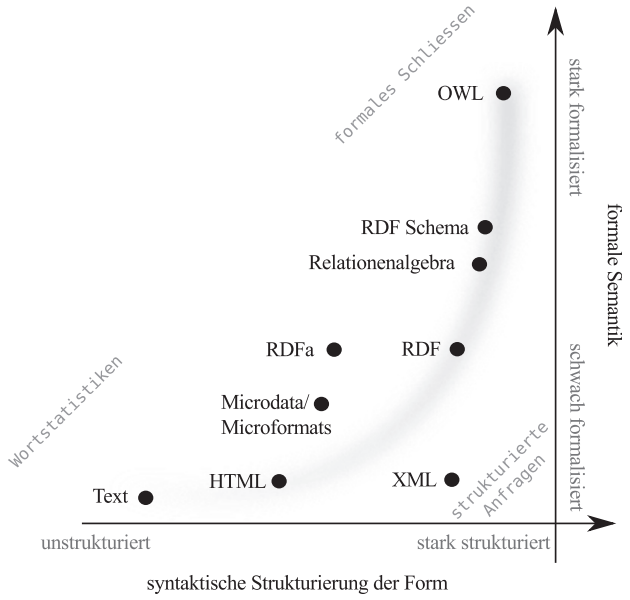


Abbildung 4.1.: Grad der formalen Semantik gängiger Formate zur Informationsrepräsentation [Dengel 2012, S. 206]

Wie bereits beschrieben, variiert der Grad der semantischen Ausdrucksstärke je nach Format. Die in Abschnitt 3.3.2 aufgezeigten Standards des Semantic Web wie RDFS oder OWL sind Vertreter für Formate, die einen hohen Grad an formaler Semantik besitzen. OWL bietet dabei die am stärksten formalisierte Semantik. Mit dieser Ontologiesprache können nicht nur einfache Informationen sondern auch komplexe Wissensstrukturen abgebildet werden. Auszeichnungssprachen wie HTML und XML haben zwar eine syntaktische Strukturierung, besitzen aber nur sehr wenige Informationen über die Bedeutung ihrer Daten. Der jeweilige Abbildungsstandard ist für das formale Schließen umso besser geeignet, je stärker sein Grad an formaler Semantik ist.

Aus der Sichtweise des Nutzers entspricht das Ziel der semantischen Suche dem Ziel klassischer IR-Systeme. Ziel ist es den Informationsbedarf des Nutzers zu decken. Es soll zu einem bestimmten Problem eine Lösung bereitgestellt werden. Der Nutzer verfügt dazu über ein implizites mentales Modell des Problems und leitet hieraus den zur Lösung beitragenden Informationsbedarf ab. Durch die Formulierung der Anfrage an das Suchsystem wird das mentale Modell in der Phase der Anfragestellung expliziert. Abbildung 4.2 veranschaulicht den zuvor erläuterten Ablauf und stellt die Unterschiede der traditionellen Schlüsselwortsuche des IR im Vergleich zur semantischen Suche dar.

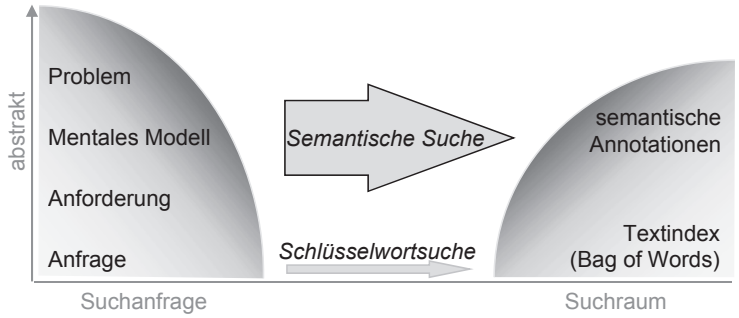


Abbildung 4.2.: Schlüsselwortsuche vs. semantische Suche [Dengel 2012, S. 233]

Wie in Abschnitt 2.4 herausgestellt, wird im Bag-of-Words-Ansatz des IR ein Dokument als eine ungeordnete Menge von Wörtern angesehen. Um über diese Daten effizient Suchen zu können, werden diese in den für die Suche optimierten Index überführt. Die Schlüsselwortsuche führt in der Suchphase ein einfaches syntaktisches Pattern-Matching durch. Es wird überprüft, ob die Suchworte im Textindex vorhanden sind. Im Rahmen des syntaktischen Pattern-Matching treten nun Probleme der lexikalischen und strukturellen Mehrdeutigkeiten auf. Mit strukturellen Mehrdeutigkeiten werden Mehrdeutigkeiten bezeichnet, die aufgrund einer bestimmten Satzstruktur nicht klar hervorgehen. Die Verfahren des klassischen IR sind nicht in der Lage diese Mehrdeutigkeiten zu erkennen. Lexikalische und strukturelle Mehrdeutigkeiten bilden nach Dengel das Grundproblem des schlüsselwortbasierten Ansatzes [Dengel 2012, S. 233]. Im Rahmen der semantischen Suche wird nun versucht, dieses Problem aufzulösen, indem den zu durchsuchenden Termen mit Hilfe von semantischen Anreicherungen eine maschinenlesbare Bedeutung zugewiesen wird. Dadurch wird eine höhere Abstraktionsebene erreicht und somit eine Annäherung an das mentale Modell des Benutzers vollzogen. Die semantische Suche hat also insgesamt zum Ziel, durch den Einsatz semantischer Technologien und der damit verbundenen Annäherung an das mentale Modell die Suchergebnisqualität zu verbessern [Dengel 2012, S. 233].

4.2. Architektur

Im Folgenden Abschnitt sollen die allgemeine Architektur eines semantischen Suchsystems dargestellt und die wichtigsten Komponenten kurz erläutert werden. Abbildung 4.3 zeigt schematisch diesen Aufbau in Anlehnung an Dengel [Dengel 2012, S. 243]. Grundsätzlich unterscheidet der Autor innerhalb der Architektur zwischen Off- und Online-Teil. Dies liegt darin begründet, dass die formalen Semantiken der Wissensbasis bereits zum Suchzeitpunkt vorliegen müssen. Um an dieses Wissen zu gelangen, können unterschiedliche Wege beschritten werden. Die Erfassung kann beispielsweise vollständig manuell erfolgen. Das heißt, ein Domänenexperte befüllt die Wissensbasis mit entsprechendem Domänenwissen. Daneben ist es möglich, Inhalte aus vorhandenen Datenquellen zu extrahieren. Die Art der Datenquellen ist hierbei beliebig. Es sind unstrukturierte, semistrukturierte oder strukturierte Daten möglich. Je nach Quellenart kommen dann in der Phase der Verarbeitung Mapping- und Annotationsmethoden, statistische Verfahren oder Methoden des Natural Language Processing (NLP) zum Einsatz. Mit NLP-Methoden können beispielsweise Entitäten und Relationen extrahiert werden.

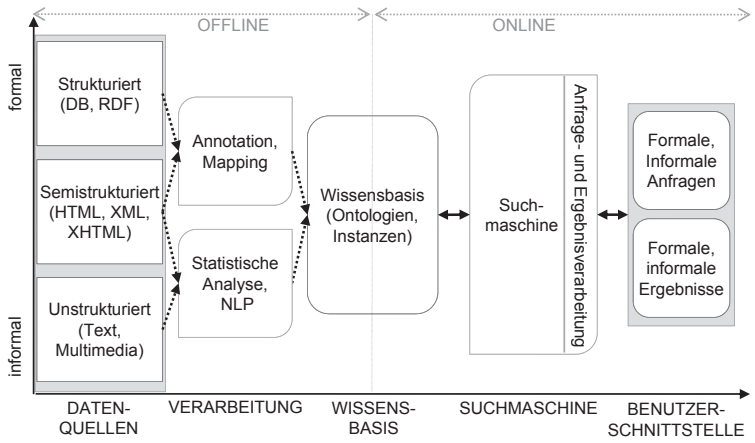


Abbildung 4.3.: Architektur semantischer Suchsysteme [Dengel 2012, S. 243]

Der Online-Teil eines semantischen Suchsystems beinhaltet Komponenten, die zur Beantwortung der Suchanfrage benötigt werden. Hier ist zunächst der eigentliche Suchraum zu nennen, der alle Daten (Objekte) enthält, über die gesucht werden soll. Die Ausprägung des Suchraums wird von den Objekten, deren Repräsentationsform und durch den gewählten Suchansatz bestimmt. Mögliche Repräsentationsformen können beispielsweise die in Abschnitt 3.2 genannten Thesauri, Topic Maps oder Ontologien sein. Neben dem Suchraum ist im Online-Teil ebenfalls die *Wissensbasis* anzutreffen. Bei einer reinen Faktensuche entspricht die Wissensbasis dem Suchraum. Daneben kann die Wissensbasis als Ergänzung zu einem möglichen Textindex eingesetzt werden. In diesem Fall wird die Wissensbasis zur Anreicherung des Dokumenten-Suchraumes mit Informationen herangezogen. Je nach Herangehensweise des semantischen Suchansatzes bildet also die Wissensbasis im Online-Teil den gesamten Suchraum oder ist lediglich ein Teil davon. Eine zentrale Rolle innerhalb des Online-Teils nimmt die *Suchmaschine* ein. Diese Komponente beinhaltet den eigentlichen Suchalgorithmus und die Anfragevor- sowie Nachbereitung. Über die *Benutzerschnittstelle* können sowohl formale als auch informale Anfragen an die Suchmaschine gestellt werden. Dies ist wiederum vom gewählten Suchansatz abhängig. Die Suchmaschine durchsucht die Wissensbasis bzw. den Suchraum und erzeugt eine zur Anfrage passende Ergebnismenge. Die Repräsentationsform der Ergebnisse (formal,

informal) wird von der Art des Suchraumes bestimmt. Sucht der Nutzer über eine Menge an Dokumenten, so ist dies beispielsweise ein einfaches Dokumenten-Listing. Wird mit Hilfe einer formalen Anfrage nach bestimmten Entitäten gesucht, so ist eine entsprechende formale Repräsentationsform zu wählen.

4.3. Überblick über semantische Suchverfahren

Nachdem in den vorherigen Abschnitten der Begriff und das Ziel der semantischen Suche definiert sowie eine allgemeine Architektur vorgestellt wurde, soll im Folgenden dargelegt werden, welche semantischer Suchverfahren in der Literatur anzutreffen sind. Diese Verfahren werden in Form unterschiedlicher Kategorien vorgestellt. Dengel hebt hervor, dass es in diesem Kontext verschiedenste Aspekte gibt, nach denen semantische Suchansätze kategorisiert werden können [Dengel 2012, S. 234]. Dies begründen Tran und Mika damit, dass eine hohe Anzahl von unterschiedlichen Verfahren anzutreffen ist [Tran und Mika 2012, S. 2 f.]. Den Grund dafür sehen die Autoren in der starken Fragmentierung der Wissenschaft auf diesem Themengebiet. So beschäftigen sich beispielsweise Forschungsgruppen aus dem Bereich der Datenbanken, des IR und des Semantic Web mit der Thematik der semantischen Suche. Dies hat, je nach Forschungsschwerpunkt, zu unterschiedlichsten Ansätzen geführt. Um einen umfassenden Überblick über diese Ansätze geben zu können, werden im Folgenden zwei Arten der Kategorisierung vorgestellt. Zum einen die Kategorisierung nach Dengel, die darauf basiert, welche Art der Hilfestellung der Benutzer bei der Suchanfrage durch die Benutzerschnittstelle erfährt [Dengel 2012]. Zum anderen die Kategorisierung nach Tran und Mika [Tran und Mika 2012]. Hier wird eine Einordnung anhand mehrerer Aspekte durchgeführt und fünf verschiedene Ansätze zur semantischen Suche identifiziert.

4.3.1. Kategorisierung nach Dengel

Dengel führt eine Kategorisierung semantischer Suchansätze mit Blick auf die Benutzerschnittstelle ein. Diese Herangehensweise wird damit begründet, dass der zugrundeliegende semantische Suchansatz in einer starken Abhängigkeit zu der Art der Anfragestellung steht. Je präziser die Anfrageformulierung, desto geringer sind die zu erwartenden lexikalischen und strukturellen Mehrdeutigkeiten. Dies hat einen weniger komplexen Suchalgorithmus zur Folge und führt zu einer höheren Genauigkeit des Suchergebnisses [Dengel 2012, S. 235]. Abbildung 4.4 veranschaulicht diesen Zusammenhang.

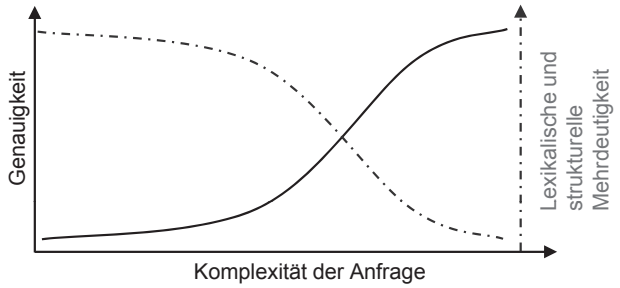


Abbildung 4.4.: Genauigkeit der Suche vs. Komplexität der Anfrage bedingt durch die lexikalische und strukturelle Mehrdeutigkeit [Dengel 2012, S. 235]

Dengel schließt also in Abhängigkeit der Benutzerschnittstelle auf das jeweilige Hauptaugenmerk des Suchansatzes. Der Autor identifiziert die folgenden Kategorien semantischer Suchansätze [Dengel 2012, S. 238 ff.]:

- **Formularbasierte Suche**: Die formularbasierte Suche ist eine semantische Suche bei, der vorhandene Metadaten per Formular angezeigt und abgefragt werden können. Daneben kann ein Suchbegriff in natürlicher Sprache angegeben werden. Da die Semantik bereits bei der Eingabe bekannt ist, sind keine komplexen Suchalgorithmen notwendig. Der Schwerpunkt dieses Ansatzes liegt in der Anfragestellung über das Graphical User Interface (GUI). Als eines der ersten formularbasierten semantischen Suchmaschinen ist SHOE[1] zu nennen.

- **Suchansätze mit RDF-basierten Anfragesprachen**: RDF-basierte Suchmaschinen besitzen ebenfalls ihren Schwerpunkt in der Anfrageformulierung. Hier wird die Anfrage mit Hilfe einer formalen RDF-basierten Sprache formuliert. Da dafür Wissen über den Aufbau der dahinter liegenden Ontologie notwendig ist, werden hier häufig Hilfestellungen bzgl. der Ontologie-Strukturen bereitgestellt.

- **Facettierte Suche**: Die facettierte Suche ermöglicht dem Nutzer die Selektion von Informationen anhand bestimmter Kategorien (Facetten). Dadurch kann der Suchraum iterativ verfeinert werden. Die Facetten setzen sich dabei aus Attribut-Werte-Paaren vorhandener Metadaten zusammen. Im Fall der semantischen facettierten Suche bilden sich die Facettenmengen aus den Werten der RDF-Prädikate der Ressourcen.

- **Semantikbasierte Schlüsselwortsuche**: Bei der semantikbasierten Schlüsselwortsuche wird die klassische Schlüsselwortsuche durch Einbeziehung verfügbarer semantischer Daten ergänzt. Dieser Ansatz ist nach Dengel für den Nutzer transparent, da ihm die Schlüsselwortsuche vertraut ist. Dadurch braucht er kein weiteres Wissen zur Nutzung dieser Art der semantischen Suche. Im Wesentlichen sind dazu zwei Schritte notwendig. Zunächst werden die zur Abfrage passenden Konzepte in der Wissensbasis bestimmt und danach verwandte Instanzen gesucht. Die Suchmaschine *SIG.MA*[2] ist ein Vertreter dieses Suchansatzes.

- **Question-Answering-Tools**: Question-Answering-Systeme sind semantische Suchmaschinen, die natürlichsprachlich formulierte Anfragen des Benutzers beantworten können. Hier werden linguistische Techniken des NLP zur Analyse und Überführung der Anfrage eingesetzt. Die verwendeten Wissensbasen sind aufwendig konstruiert und stellen detailreiches Wissen zur Verfügung. Ein Beispiel für diese Art der semantischen Suche ist das von IBM entwickelte Computersystem *Watson*[3].

- **Schlüsselwortsuche mit semantischer Nachverarbeitung**: Diese Art der semantischen Suche führt eine herkömmliche Schlüsselwortsuche durch und bereitet das Ergebnis semantisch auf.

- **Semantikbasierte intelligente Visualisierung**: Systeme mit semantikbasierter intelligenter Visualisierung nutzen intelligente Visualisierungstechniken, um den Nutzer während des Suchprozesses zu unterstützen. Die verwendeten Darstellungen werden dabei mit Hilfe von formalen Semantiken erzeugt. Abbildung 4.5 zeigt beispielweise die Suchergebnissdarstellung der Suchmaschine *eyePlorer* für den Suchbegriff „Bit“ [eyePlorer 2013].

[1] http://www.cs.umd.edu/projects/plus/SHOE/search (Zugriffsdatum: 19.01.2013)
[2] http://sig.ma (Zugriffsdatum: 19.01.2013)
[3] http://www-05.ibm.com/de/pov/watson/index.html (Zugriffsdatum: 19.01.2013)

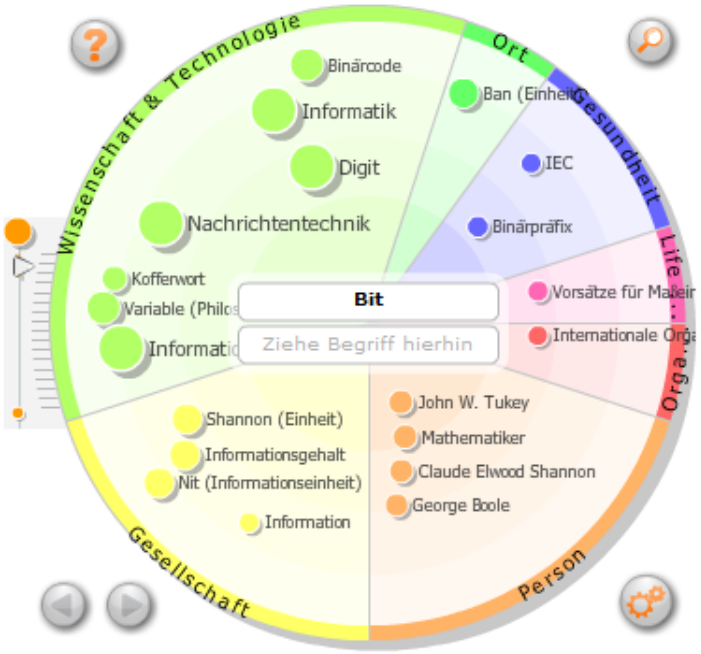


Abbildung 4.5.: Beipiel einer intelligenten Visualisierungstechnik der Suchmaschine eyePlorer [eyePlorer 2013]

4.3.2. Kategorisierung nach Tran und Mika

Um weitere Ansätze zur semantischen Suche darzustellen, wird im Folgenden die Kategorisierung nach Tran und Mika erläutert [Tran und Mika 2012]. Neben der Gemeinsamkeit, dass alle Ansätze formale Semantiken innerhalb des Suchprozesses verwenden (siehe Definition), führen die Autoren eine Charakterisierung anhand mehrerer Kriterien durch. Diese Kriterien sollen im weiteren Verlauf kurz erläutert werden. Im Anschluss daran werden die von Tran und Mika ermittelten Ansätze zur semantischen Suche vorgestellt.

Folgende Aspekte haben die Autoren in ihrer Analyse untersucht und zur Charakterisierung herangezogen:

- Art des Informationsbedürfnisses
- Art der Anfrageformulierung (Query Paradigma)
- Art des verwendeten semantischen Modells
- Repräsentationsform der Informationsquellen (Daten)
- Art des verwendeten Rahmenwerks zur Interpretation der Daten und Anfragen

Eine Hauptmotivation zur Nutzung semantischer Suchsysteme besteht darin, mehr als nur einfache *Informationsbedürfnisse des Nutzers* zu befriedigen [Tran und Mika 2012, S. 4]. Das heißt: neben der klassischen Suche nach Informationen in Dokumenten können semantische Suchverfahren andere Informationsbedürfnisse decken. Dazu zählt beispielsweise die *Entitätssuche*, welche dazu genutzt werden kann, Entitäten wie Personen oder Unternehmen aufzufinden. Im Gegensatz dazu dient die bereits erwähnte *Faktensuche* zur Suche nach Informationen über Entitäten. Zu nennen wäre hier beispielsweise die Telefonnummer einer

bestimmten Person. Die *relationale Suche* geht noch einen Schritt weiter und bezieht Verbindungen zwischen Entitäten mit in die Suche ein. Somit ist eine Suchabfrage über diese Verbindungen hinweg möglich.

In Bezug auf die verwendete *Anfrageformulierung* sind Parallelen zur der im vorherigen Abschnitt beschriebenen Kategorisierung nach Dengel zu sehen. Die Art der Anfragestellung bildet in diesem Fall aber nur einen von fünf Aspekten der vorgenommenen Kategorisierung. Genannt werden folgende bereits bekannte Ausprägungen der Benutzerschnittstelle: Die *Schlüsselwortsuche*, die *explorative Suche* und die *facettierte Suche* als deren Spezialfall, sowie Interfaces, welche eine *natürlichsprachliche Anfrageformulierung* erlauben.

Welches *semantische Modell* innerhalb der Wissensbasis zugrunde liegt, ist ein weiteres Kriterium im Rahmen der Charakterisierung. Ein semantisches Modell bestimmt auf konzeptueller Ebene den Aufbau der semantischen Daten und Metadaten. Die in Abschnitt 3.2 beschriebenen lexikalischen Modelle, wie Taxonomien oder Thesauri dienen dazu, die Bedeutung von Wörtern als Konzepte abzubilden. Ontologien dagegen erfassen real-weltliches Wissen.

Die *Repräsentationsform der Informationsquellen* wird in semantische Daten und Rohdaten unterschieden werden. Mit Rohdaten sind die in Abbildung 4.3 aufgeführten unstrukturierten sowie semistrukturierten Datenquellen gemeint. Semantische Daten liegen in strukturierter Form, wie zum Beispiel SKOS, RDF oder OWL vor. Semantische Daten beschreiben Objekte und deren Beziehung aus der realen Welt. Werden Rohdaten semantisch annotiert, so bezeichnen Tran und Mika dies als semantische Metadaten.

Das verwendete *Rahmenwerk zur Interpretation der Daten und Anfragen* ist ein weiteres Unterscheidungsmerkmal semantischer Suchsysteme. Als eine Kernaufgabe von Suchsystemen, welche auf unstrukturierten Daten arbeiten, sehen die Autoren die semantische Vorverarbeitung der Daten sowie der Anfrage [Tran und Mika 2012, S. 9]. Ziel ist es, eine reichhaltigere Repräsentation und ein besseres Verständnis des Eingabetextes zu erhalten. Dazu ist es notwendig, Entitäten, Konzepte und Relationen zu extrahieren. Wie schon beschrieben, steht dieser Ansatz im Kontrast zum klassischen IR und dem Bag-of-Words-Ansatz (siehe Abbildung 4.2). Hier kommen statistische Methoden und NLP-Techniken zum Einsatz. Die Autoren nennen in diesem Kontext weitere NLP-Techniken wie das Part-of-Speech-Tagging[4] und die Wortsinn-Disambiguierung[5]. Neben der Interpretation der Daten und der Anfragen gibt es unterschiedliche Ansätze zur Anfrageverarbeitung und zur Ergebnispräsentation.

Auf Grundlage der zuvor beschriebenen Aspekte haben Tran und Mika folgende fünf Ansätze der semantischen Suche identifiziert:

- **Konzeptbasiertes Dokumenten-Retrieval**: Diese Art der semantischen Suche erlaubt eine Schlüsselwortsuche über Dokumente als Informationsquelle. Der Suche liegen dabei Konzepte in Form von semantischen Daten zugrunde, die aus Schlüsselwörtern und Dokumenten extrahiert wurden. Dazu werden leichtgewichtige lexikalische Modelle wie beispielsweise Thesauri eingesetzt. Diese Modelle werden dazu verwendet, die Semantik der Suchanfrage und der Dokumente interpretieren zu können. Die durch die lexikalischen Modelle abgebildeten Konzepte werden dazu genutzt, die Suchanfrage zu verfeinern oder die Relevanz der Suchergebnisse zu verbessern. Oftmals unterstützen diese Systeme auch die

[4] Beim Part-of-Speech-Tagging findet eine syntaktische Zerlegung eines Satzes in einzelne Wortformen statt.
[5] Die Wortsinn-Disambiguierung versucht die Bedeutung von Wörtern innerhalb eines Satzes zu ermitteln.

explorative Suche, mit deren Hilfe der Nutzer den Suchraum iterativ verfeinern kann. Das konzeptbasierte Dokumenten-Retrieval bezeichnen Tran und Mika als die klassische Art der semantischen Suche [Tran und Mika 2012, S. 16]. Dieser Typ findet häufig in Kombination mit dem Bag-of-Word-Ansatz des IR Anwendung. Die Internetsuchmaschine Hakia[6] ist ein prominentes Beispiel für diesen Ansatz.

- **Annotationsbasiertes Dokumenten-Retrieval**: Der Ansatz nutzt die bereits erwähnten Methoden zur Informationsextraktion. Diese Technik hat nach Tran und Mika in der neuesten Zeit deutliche Fortschritte gemacht. Wie zuvor bilden Dokumente den Suchraum dieses Ansatzes. Diese Dokumente werden semantisch vorprozessiert und danach unter Berücksichtigung der semantischen Annotationen durchsucht. Das von IBM entwickelte System AVATAR[7] ist beispielsweise ein Kandidat des annotationsbasierten Dokumenten-Retrievals.

- **Entitätssuche**: Im Gegensatz zu den beiden zuvor beschriebenen Ansätzen wird bei der Entitätssuche nicht nach Dokumenten, sondern nach Entitäten gesucht. Dazu zählt die Entitätssuche innerhalb eines annotierten Dokumentenbestandes, als auch die Entitätssuche innerhalb einer Wissensbasis. Microsofts EntityCube[8] ist ein Vertreter dieses Suchansatzes. Dieses System extrahiert textuelle Informationen und ermöglicht eine entitätsbezogene Schlüsselwortsuche.

- **Relationale Schlüsselwortsuche**: Die relationale Schlüsselwortsuche umfasst alle Ansätze, bei denen eine schlüsselwortbasierte Suche über semantische Daten stattfindet. Der Fokus dieses Ansatzes liegt dabei darauf, passend zur Anfrage komplexe Subgraphen eines semantischen Modells zu ermitteln. SemSearchPro[9] beispielsweise übersetzt Schlüsselwort-Anfragen zu SPARQL-Anfragen mit Hilfe von semantischen Modellen. Die SPARQL-Anfragen werden dazu genutzt, um nach Beziehungen zwischen den Entitäten suchen zu können.

- **Relationale natürlichsprachliche Suche**: Im Gegensatz zu der relationalen Schlüsselwortsuche liegt bei diesem Ansatz die Anfrage in natürlicher Sprache vor. Ziel ist es, diese Anfrage zu interpretieren und innerhalb einer Wissensbasis eine Antwort auf die natürlichsprachliche formulierte Anfrage zu finden. Wolfram Alpha[10] ist eine Suchmaschine, welche diese Art der Suche ermöglicht.

Die Autoren heben abschließend hervor, dass die Auswahl eines Ansatzes zur semantischen Suche vom jeweiligen Kontext abhängt [Tran und Mika 2012, S. 19]. Entscheidend bei der Auswahl ist vor allem, welches Informationsbedürfnis zu befriedigen ist. Daneben sollte beachtet werden, welches semantische Modell zu Grunde liegt und welche Informationsquellen-Art zu durchsuchen ist.

[6] http://hakia.com (Zugriffsdatum: 18.01.2013)
[7] http://www.almaden.ibm.com/cs/projects/avatar (Zugriffsdatum: 20.01.2013)
[8] http://entitycube.research.microsoft.com (Zugriffsdatum: 19.01.2013)
[9] http://www.websemanticsjournal.org/index.php/ps/article/view/236/234 (Zugriffsdatum: 19.01.2013)
[10] http://www.wolframalpha.com (Zugriffsdatum: 19.01.2013)

5. KnowledgeFinder: Das Wissensportal des DLR

Das Ziel dieser Arbeit darin besteht den KnowledgeFinder um semantische Suchfunktionalitäten zu erweitern. Aus diesem Grund soll im folgenden Kapitel das KnowledgeFinder-Wissensportal vorgestellt werden. Zunächst wird dazu der Einsatzzweck dieser Software im Kontext des DLR erläutert. Danach erfolgt die Darstellung der Architektur und der Funktionsweise dieses Software-Systems. Anschließend wird die Benutzerschnittstelle anhand eines Beispiels erörtert. Zuletzt werden in Abschnitt 5.4 die im Einsatz befindlichen Technologien kurz aufgeführt.

5.1. Kontext & Einsatzzweck

Der KnowledgeFinder wurde innerhalb einer Projektaktivität der Einrichtung *Simulation und Softwaretechnik* des DLR entwickelt. Die in Java geschriebene Software wird aktuell in verschiedenen Projekten der Einrichtung zur Unterstützung des Wissensmanagements eingesetzt. Das Wissensmanagement im Allgemeinen umschreibt dabei Aktivitäten eines Unternehmens, welche auf einen *„verbesserten organisationsspezifischen Umgang mit internem sowie externem Wissen abzielen“* [Cissek 2010, S. 111]. Diese Aktivitäten sollen Mitarbeitern dabei helfen, das eigene Know-how zu explizieren und in einer strukturierten Form zu dokumentieren. Ziel ist nicht nur die Archivierung, sondern auch das Wiederfinden und die Wiederverwendung von Wissen in passenden Situationen [Dengel 2012, S. 83]. Gerade innerhalb eines forschungsorientierten Betriebes, wie im DLR, hat das Wissensmanagement eine wichtige Bedeutung. Hier besteht aufgrund einer ständigen Fluktuation von Mitarbeitern die Gefahr, dass mit deren Weggang auch Wissen abhanden kommt. In diesen Kontext wird der KnowledgeFinder als Wissensportal dazu eingesetzt, Informationen und somit Wissen auf einfache Weise wiederzufinden. Wissensportale dienen dabei als zentrale Plattform im Wissensmanagement und können als eine Art Wegweiser durch die Informationen eines Unternehmens angesehen werden [Cissek 2010, S. 138].

Als technologisches Hilfsmittel zum Auffinden von Wissen hat der KnowledgeFinder verschiedene Einsatzzwecke. Wie bereits in Kapitel 1 erwähnt, ist die in der Machbarkeitsstudie zu betrachtende Suche innerhalb der Publikationsdatenbank Elib (Electronic Library) eine Anwendung. Elib ist eine Datenbank mit ca. 69.000 Publikationen[1] des DLR aus dem Zeitraum von 1990 bis heute [Elib 2012]. Da die eigenen Suchfunktionalitäten dieser Publikationsdatenbank sehr eingeschränkt sind, wurde im Rahmen einer Diplomarbeit ein alternativer Zugriff auf die Veröffentlichungen mit Hilfe des KnowledgeFinder realisiert [Juchmes 2011]. Ziel war es insbesondere eine verbesserte Benutzerschnittstelle zur Verfügung zu stellen, mit der sowohl einfache als auch komplexe Suchanfragen durchgeführt werden können. Die zuvor beschriebene Ausprägung des KnowledgeFinder wird als *Elib-Portal* bezeichnet. Neben dem Elib-Portal wird der KnowledgeFinder aber auch als Suchframework über beliebige Datenbestände eingesetzt. Im *Monitor-Portal*[2] werden beispielsweise statistische Daten des Flughafenwesens gesammelt, aufbereitet und dargestellt. Der KnowledgeFinder indiziert diesen Datenbestand und ermöglicht so das komfortable Durchsuchen dieser Daten.

[1] Stand vom 29.11.2012.

[2] http://monitorportal.dlr.de (Zugriffsdatum: 10.01.2013)

5.2. Architektur & Funktionsweise

Abbildung 5.1 stellt die Systemarchitektur des Suchframework anschaulich dar. Dieses Suchframework arbeitet nach dem schlüsselwortbasierten Ansatz des klassischen IR (siehe Kapitel 2). Der KnowledgeFinder kann sowohl Metadaten als auch Volltexte verarbeiten. Die Ausprägung der Metadaten ist abhängig von der Datenanbindung. Zu jedem Metadatentyp existiert ein eigenes *Metadatenmodell*. Dieses flexible Modell repräsentiert den Aufbau der Metadaten und wird zur Verarbeitung der Metadaten während der Indexierung und des Suchvorganges herangezogen. Aufgrund der Flexibilität ist es möglich, das Wissensportal auf unterschiedliche Projektanforderungen anzupassen. Die Anbindung des KnowledgeFinder kann auf zwei verschiedene Arten erfolgen. Zum einen gibt es die Möglichkeit ein Subversion Repository als Datenquelle zu nutzen (blaue Linie). Zum anderen existiert eine Schnittstelle zum Import der Elib-Datenbestände (rote Linie).

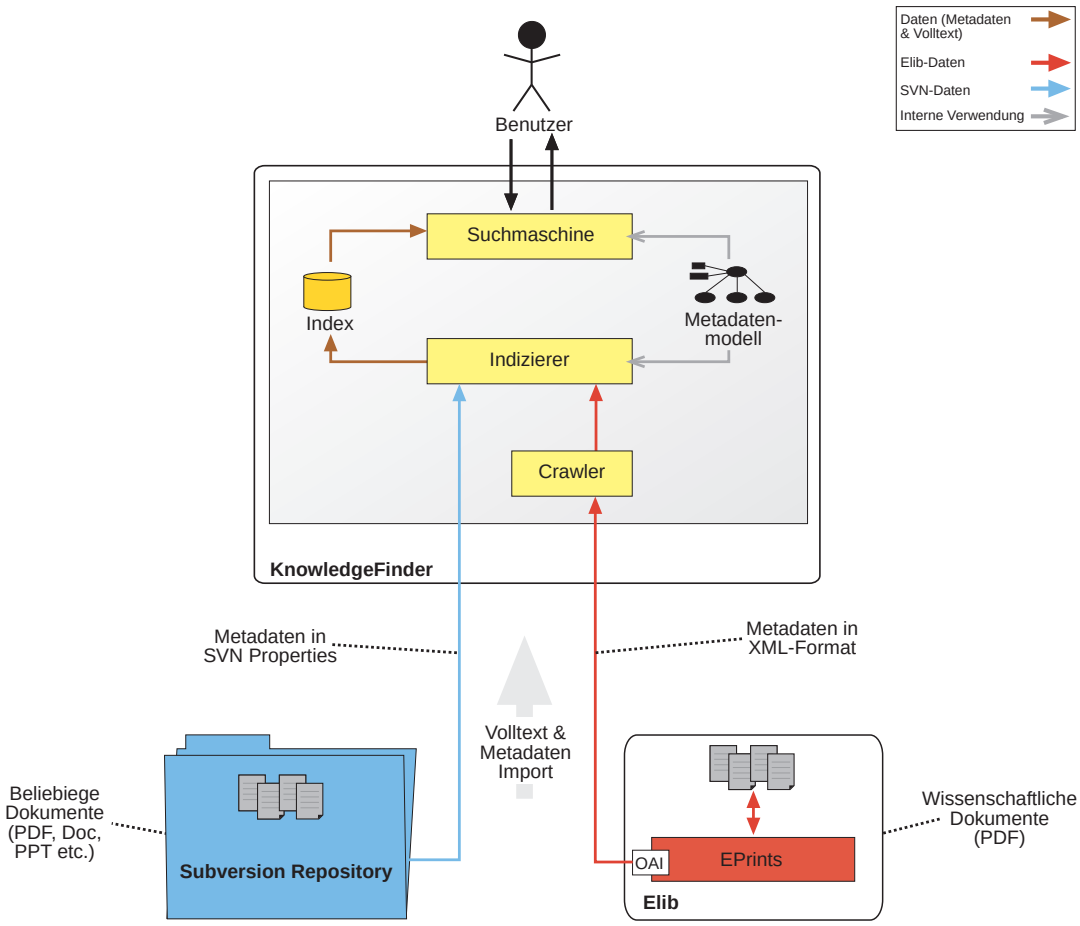


Abbildung 5.1.: KnowledgeFinder-Systemarchitektur

Ist der KnowledgeFinder an Subversion (SVN) angebunden, so liegen die Metadaten der einzelnen Dokumente innerhalb einer SVN-Eigenschaft (engl. property) vor. Zur Pflege dieser Metadaten wird eine vom DLR entwickelte Datenmanagement-Software eingesetzt. Im Rahmen der Indizierung importiert der KnowledgeFinder die Daten des SVN Repositories und führt eine Indizierung sowohl der Metadaten als der Volltexte durch. Der Dokumententyp der Volltexte im SVN Repository ist dabei beliebig. Es werden alle bekannten Dateiformate wie PDF, Word, PowerPoint oder Excel unterstützt.

Die Anbindung des KnowledgeFinder an die Elib-Datenbestände soll als nächstes erörtert werden. Zur Verwaltung der Elib-Dokumente wird die Publikationsverwaltungs-Software *EPrints*[3] eingesetzt (siehe Abbildung 5.1). EPrints stellt eine sogenannte *OAI-Schnittstelle* (Open Archives Initiative) zur Verfügung. Die Elib-Metadaten werden über diese spezielle Schnittstelle abgerufen und liegen in diesem Fall im Dublin-Core-Format vor. Das OAI-Interface implementiert dazu das OAI Protocol for Metadata Harvesting[4] (OAI-PMH). Das Suchframework nutzt zum Sammeln der Elib-Metadaten einen eigenen *Crawler*. Anders als die restlichen Komponenten des KnowledgeFinder, ist der Crawler in der Programmiersprache Python geschrieben. Der Crawler spricht die OAI-Schnittstelle an und ruft die Metadaten der Publikationen über das OAI-Protokoll ab. Jeder Metadatensatz hat zusätzlich zu den Metadatenattributen ein Verweis zu den Volltext-Dokumenten. Der Crawler wertet diese Verweise aus und importiert die dazugehörigen Dokumente. Im Fall von Elib handelt es sich bei den Volltexten ausschließlich um wissenschaftliche Dokumente im PDF-Format. Pro Elib-Eintrag wird ein Metadatensatz vom Crawler abgerufen, im KnowledgeFinder abgelegt und für die Indizierung vorbereitet. Das Crawling der Elib-Publikationen ist aufgrund der enormen Datenmenge eine Offline-Aktivität, welche typischerweise einmal am Tag ausgeführt wird. Im Anschluss an das Crawling können daraufhin die Elib-Daten indiziert werden.

Der *Indizierer* verarbeitet nun die Metadaten und die Volltexte aus den jeweiligen Datenquellen und schreibt diese in einen invertierten *Index*. Im Rahmen der Metadatenindizierung kommt, wie schon erwähnt, das Metadatenmodell zum Einsatz. Jedes Metadatenattribut wird in der Indizierungsphase einzeln betrachtet und als eigenes Datum im Index abgelegt. Dadurch ist eine schnelle Filterung der Datensätze nach Metadatenattributen wie Autor oder Kategorie möglich. Dieses Vorgehen ermöglicht innerhalb der Benutzerschnittstelle die Darstellung von Facettenfilter basierend auf den Metadatenattributen. Der Indizierer verrichtet seine Arbeit ebenfalls als Offline-Aktivität.

Nach der Indizierung kann die *Suchmaschine* auf den Index zugreifen und die jeweiligen Datenbestände durchsuchen. Dazu führt diese Komponente neben der eigentlichen Suche auch die Interaktion mit dem *Benutzer* durch. Hier finden also die Verarbeitung der Suchanfrage, die Relevanzbestimmung und die Ergebnisdarstellung statt (vgl. Abschnitt 2.2). Um die Suche durchführen zu können, transformiert diese Komponente zunächst den Informationsbedarf des Nutzers in eine entsprechende Repräsentationsform. Anschließend wird eine Suche und das darauffolgende Ranking durchgeführt. Zuletzt präsentiert die Suchmaschine die Ergebnismenge innerhalb der Benutzeroberfläche. Zur Generierung der Facetten greift die Suchmaschine ebenfalls auf das Metadatenmodell zu. Um die Suchmaschinen-Komponente möglichst flexibel einsetzen zu können, ist sie kompatibel zum Java-Portlet-Standard JSR 286 implementiert [JSR286 2008]. Durch die Implementierung als Portlet ist es möglich, den KnowledgeFinder in unterschiedlichen Portaltypen einsetzten zu können.

[3] EPrints ist eine frei verfügbare Publikationsverwaltungs-Software, welche an der Universität Southampton entwickelt wird [EPrints 2012].

[4] Das OAI Protocol for Metadata Harvesting beschreibt ein Verfahren zum Abrufen von großen Literaturbeständen [OAI-PMH 2012].

5.3. Benutzerschnittstelle

Der KnowlodgeFinder besitzt eine anpassungsfähige Benutzerschnittstelle. Je nach Einsatzzweck existieren unterschiedliche Ausprägungen dieser Schnittstelle. Da die durchzuführende Machbarkeitsstudie am Beispiel des Elib-Portals erfolgen soll, wird im Folgenden das Interface dieses Portals vorgestellt.

Abbildung 5.2 zeigt die Benutzerschnittstelle des Elib-Portals. Sowohl in der rechten als auch in der linken Randspalte befinden sich die auf den Metadatenattributen beruhenden Facetten. Die linke Randspalte beinhaltet eine *Facette zur Navigation*. Diese Facette bietet dem Benutzer die Möglichkeit, ein bestimmtes Institut des DLR auszuwählen.

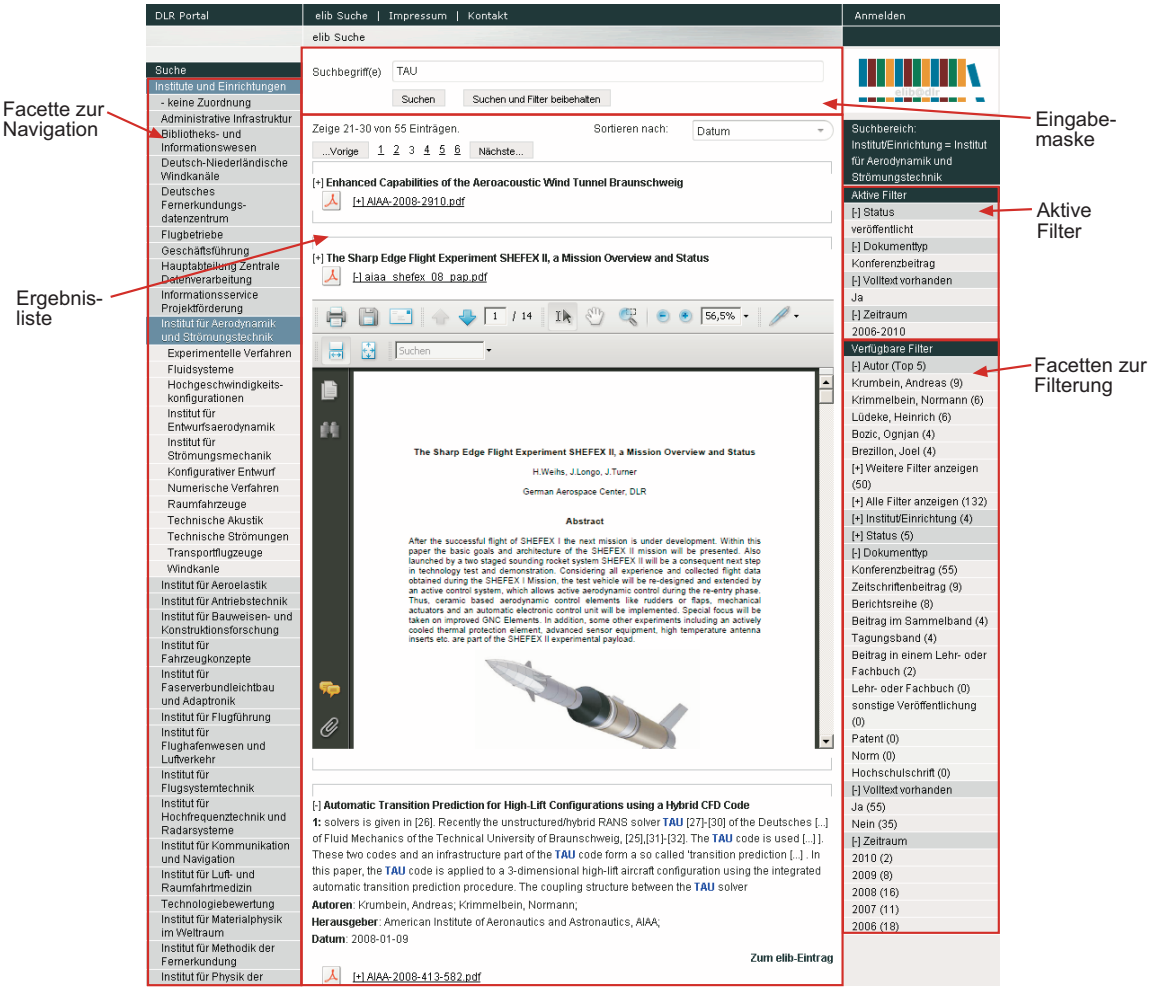


Abbildung 5.2.: Die Benutzerschnittstelle des Elib-Portals

Im Gegensatz zur linken Randspalte hat der Benutzer in der rechten Spalte die Möglichkeit, eine Filterung anhand mehrerer Facetten gleichzeitig durchzuführen. Neben den *Facetten zur Filterung* werden in der rechten Hälfte auch alle *aktiven Filter* angezeigt. Im dargestellten Beispiel findet eine Filterung nach allen veröffentlichten Konferenzbeiträgen mit Volltexten aus dem Zeitraum von 2008 bis 2010 statt. Die Facetten der beiden Randspalten werden innerhalb der Suchanfrage konjugiert. In der Hauptspalte ist die *Eingabemaske* und die *Ergebnisliste* zu sehen. Wie bei Google & anderen Suchmaschinen üblich, kann der Benutzer

innerhalb der Eingabemaske seine Suchbegriffe in Form von Schlüsselwörtern eingeben. Die Filterung durch die Facetten bezieht sich dabei immer auf die eingegebenen Suchbegriffe. Die Ergebnisliste wird unterhalb der Eingabemaske dargestellt. Um die Übersichtlichkeit zu gewährleisten, ist die Anzahl der Treffer pro Seite auf 10 begrenzt. Mit Hilfe einer Seitennavigationsleiste kann der Benutzer durch die Ergebnisliste navigieren und zu weiteren Treffern gelangen. Daneben besteht die Möglichkeit, die Sortierung der Ergebnisliste zu ändern. Unterhalb der zuvor beschriebenen Elemente werden die eigentlichen Suchtreffer präsentiert. Jeder Elib-Eintrag wird mit den vorhandenen Metadaten, wie beispielsweise Titel, Autor, Herausgeber oder Datum, dargestellt. Innerhalb jedes Eintrages findet eine Hervorhebung der gefundenen Schlüsselwörter statt. Ist ein Volltext zu einem Eintrag vorhanden, so kann der Nutzer das Dokument direkt im Browser betrachten.

5.4. Verwendete Technologien

In diesem Abschnitt werden die im KnowledgeFinder verwendeten Technologien kurz vorgestellt. Neben dieser Betrachtung findet eine Zuordnung dieser Technologien zu den einzelnen Komponenten des KnowledgeFinder statt.

Der KnowledgeFinder setzt die nachfolgenden Software-Bibliotheken bzw. -Frameworks ein:

- **Apache Lucene**: Lucene ist eine quelloffene Java-Such-API (Application Programmers Interface), die unter der Apache Software Foundation entwickelt wird [Apache Lucene 2012]. Lucene unterstützt die wesentlichen Vorgänge eines IR-Systems. Dazu gehören die Index-Erstellung sowie die Suche innerhalb eines Indexes. Alle weiteren Aspekte, wie das Crawling oder eine Benutzerschnittstelle, sind selbstständig zu realisieren. Lucene setzt eine Kombination aus dem booleschen Modell und dem Vektorraummodell ein. Zur Suche wird zunächst der Ansatz des booleschen Modells genutzt. Innerhalb des Ranking werden danach die Gewichtungen des Vektorraummodells herangezogen [Apache Lucene 2012]. Im KnowledgeFinder wird Lucene innerhalb der Indizierungs- und Suchmaschinen-Komponente in der Version 3.0 eingesetzt.

- **Liferay**: Liferay ist eine Open-Source-Portalsoftware. Diese Software unterstützt den zuvor erwähnten Portlet-Standard JSR 286 [Liferay 2012]. Die Suchmaschinen-Komponente wird als Portlet innerhalb von Liferay bereitgestellt und steht dem Benutzer dadurch als Web-Anwendung zur Verfügung. Verwendung findet die Version 6.0.4 in Kombination mit einem Tomcat-Server 6.0.26.

- **Vaadin:** Dieses Java-Framework dient der Erstellung von Rich Internet Applications[5] (RIA) [Vaadin 2012]. Vaadin hat eine serverseitige Architektur, die AJAX (Asynchronous JavaScript and XML) zur Kommunikation zwischen Client und Server einsetzt. Die Darstellung der Benutzerschnittstelle wird mit Widgets des Google Web Toolkit[6] realisiert. Das Vaadin-Framework kommt in der Suchmaschinen-Komponente zur Umsetzung der GUI zum Einsatz (Version 6.3.3).

[5] Das Konzept der Rich Internet Applications beschreibt im Allgemeinen Web-Anwendungen, welche reichhaltigere Benutzerschnittstellen und Interaktionsmöglichkeiten besitzen als herkömmliche HTML-basierte Web-Applikationen

[6] https://developers.google.com/web-toolkit (Zugriffsdatum: 18.03.2013)

Teil II.

Durchführung & Ergebnisse

6. Anforderungen

Wie in Kapitel 1 dargelegt, soll zur Beantwortung der Fragestellung des Mehrwerts semantischer Technologien ein beispielhafter Anwendungsfall herangezogen werden. Grundlage dieser Machbarkeitsstudie ist die Publikationsdatenbank des DLR. Der KnowledgeFinder in Form des Elib-Portals ermöglicht das komfortable Durchsuchen dieses Datenbestandes. Als konzeptionelle Grundlage der Studie werden in diesem Kapitel die Anforderungen an die Erweiterung des Elib-Portals um eine semantische Suche spezifiziert. Die Ermittlung von formalisierten Anforderungen mit Hilfe z.B. einer Use-Case-Analyse nach Cockburn [Cockburn 2007] oder des taskbasierten Ansatzes nach Lauesen [Lauesen 2003] ist im Rahmen der durchgeführten Machbarkeitsstudie nicht praktikabel. Dies liegt darin begründet, dass auf Seiten der Stakeholder nur vage Vorstellungen über den Umfang der gewünschten Erweiterungen vorhanden sind. Für eine formale Dokumentation liegen also nicht genügend Detailinformationen vor. Stattdessen werden die Anforderungen mit Hilfe einer Ziel-Szenario-Analyse nach Pohl ermittelt [Pohl 2008], da konkrete Ziele bzgl. dem was die Erweiterungen leisten soll durchaus vorhanden sind. Zunächst werden dazu die identifizierten Ziele bzgl. der Integration einer semantischen Suche dargestellt und anschließend mit Hilfe informeller Szenarien beschrieben. Diese Szenarien repräsentieren im Rahmen dieser Ausarbeitung den funktionalen Teil der Anforderungen. Zuletzt erfolgt die Herleitung von Qualitätsanforderungen sowie Rahmenbedingungen aus den zuvor aufgeführten Szenarien.

6.1. Ziel-Szenario-Analyse

Mit Hilfe einer Ziel-Szenario-Analyse werden Ziele eines Projektes strukturiert erfasst und mittels Szenarien veranschaulicht. Zur Zielmodellierung wird ein sogenannter *erweiterter Und-Oder-Baum* eingesetzt. Mit diesem Baum ist es möglich, eine Zielhierarchie eines Projektes zu erstellen und Abhängigkeiten sowie Konflikte zwischen den Zielen zu ermitteln. Die identifizierten Ziele werden dazu in Ober- und Unterziele zerlegt. Die anschließende Veranschaulichung der Ziele mittels informeller Szenarien dient der Dokumentation des konkreten Mehrwertes des geplanten Systems. Die Szenarien beschreiben die Schritte, welche zur Zielerfüllung notwendig sind. Daneben enthalten Szenarien wichtige Kontextinformationen wie beispielsweise relevante Benutzerrollen.

6.1.1. Ziele

Die in Abbildung 6.1 dargestellte Zielmodellierung zeigt die identifizierten Ziele bzgl. der Integration einer semantischen Suche. Diese Ziele wurden in Zusammenarbeit mit Mitarbeitern des DLR erarbeitet. Zur Gewinnung wurden mehrere Brainstormings durchgeführt, die durch die Kreativitätstechnik des Mind Mapping unterstützt wurden.

Das Hauptziel der Integration einer semantischen Suchfunktionalität beschreibt das Ziel Z-1. Die ca. 69.000 Publikationen des DLR sollen mit Hilfe des Elib-Portals semantisch durchsucht werden können. Dies soll die Suche von Wissenschaftlern in fachfremden Themengebieten erleichtern. Unterhalb dieses Oberzieles konnten weitere Ziele identifiziert werden. Zunächst ist es wichtig, dass die durchgeführten Erweiterungen für den Nutzer transparent sind (Z-1.1).

Der Nutzer soll also das Elib-Portal wie gewohnt verwenden können und die Erweiterungen sollen unbemerkt im Hintergrund ihren Dienst verrichten. Das bedeutet, dass die Erweiterungen innerhalb der momentanen Benutzerschnittstelle des Portals stattfinden. Aufgrund dieser Tatsache besteht eine Abhängigkeit zum Ziel der Darstellung in der vorhandenen GUI (Z-1.2). Dieses Ziel hat wiederum zwei Unterziele. Z-1.2.1 beschreibt die Erweiterung der Benutzerschnittstelle um eine Autovervollständigung. Die momentane Sucheingabemaske soll also um Mechanismen erweitert werden, die dem Benutzer bei der Eingabe passende Suchbegriffe automatisch vorschlagen. Diese Begriffe sollen auch semantisch ähnliche Begriffe umfassen. Ein weiteres Unterziel von Z-1.2.2 ist die Nutzung von Facetten. Die Facetten des Elib-Portals sollen dazu genutzt werden, um zum momentan eingegebenen Suchbegriff semantisch verwandte Elemente anzuzeigen. Dies könnten zum Beispiel ein Hinweis wie „Das könnte Sie auch interessieren“ oder die Darstellung ähnlicher Konzepte umfassen. Ein weiteres Ziel ist die Unterstützung der schlüsselwortbasierten Suchanfrage (Z-1.3). Der momentane Suchmechanismus des KnowledgeFinder soll also nicht geändert werden. Es wird nicht verlangt, dass der Benutzer eine spezielle Anfragesprache wie SPARQL o.ä. beherrschen muss. Zuletzt ist das Ziel der Unterstützung der explorativen Suche in Z-1.4 aufgeführt. Die semantischen Erweiterungen sollen die explorative Suche unterstützen und dem Nutzer iterativ dabei helfen, seinen Informationsbedarf decken zu können. Die Unterstützung der explorativen Suche spielt gerade im Zusammenhang mit der zuvor beschriebenen fachübergreifenden Suche von Wissenschaftlern eine bedeutende Rolle. Zur Unterstützung dieser Suche sollen insbesondere Zusammenhänge zwischen verschiedenen DLR-Themengebieten aufgedeckt werden.

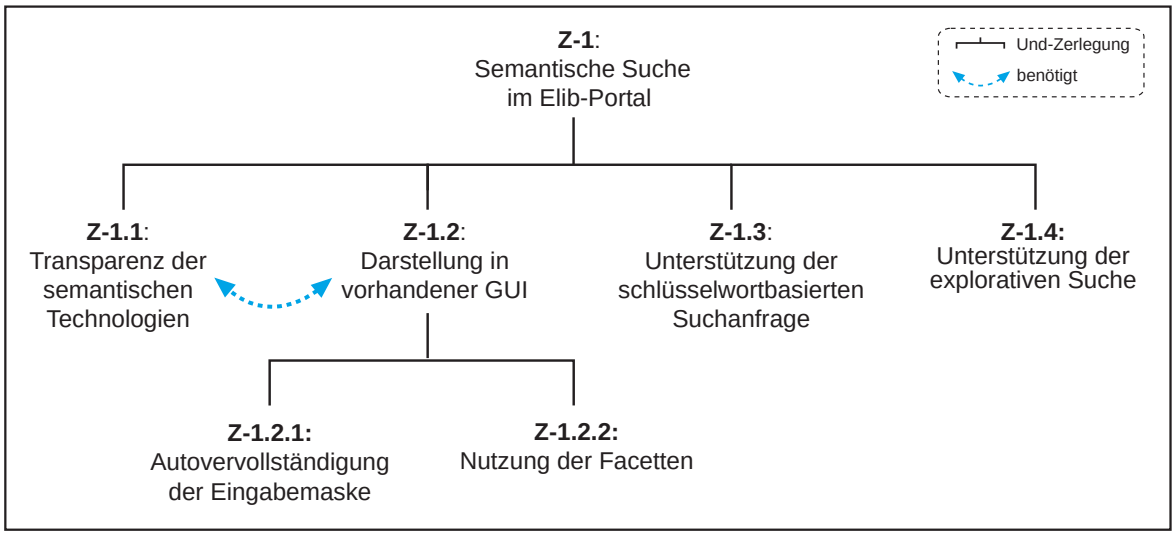


Abbildung 6.1.: Zielmodellierung mit erweitertem Und-Oder-Baum

6.1.2. Szenarien & Benutzerrolle

In den Tabellen 6.1 und 6.2 sind Szenarien aufgeführt, welche jeweils die Erfüllung der Ziele aus dem vorherigen Abschnitt beschreiben. Diese Szenarien sind optative Szenarien, welche den angestrebten Soll-Zustand eines Systems schildern. Im Anschluss an die Szenarien findet eine Darstellung der darin identifizierten Benutzerrolle statt. Die Szenarien und die Benutzerrolle dienen im nachfolgenden Kapitel der Konzeption als Anforderungsgrundlage.

Tabelle 6.1.: Szenario S-1: Unterstützung des wissenschaftlichen Mitarbeiters in der explorativen Suche

Szenario-ID:	S-1
Szenario-Name:	Unterstützung des wissenschaftlichen Mitarbeiters in der explorativen Suche
Ziele:	Z-1.1: Transparenz der semantischen Technologien, Z-1.2: Darstellung in vorhandener GUI, Z-1.2.2: Nutzung der Facetten, Z-1.3: Unterstützung der schlüsselwortbasierten Suchanfrage, Z-1.4: Unterstützung der explorativen Suche
Szenario-Schritte:	Max Mustermann ist wissenschaftlicher Mitarbeiter beim DLR. Im Rahmen eines Projektes möchte er DLR-weit nach Veröffentlichungen zum Thema Erneuerbare Energien suchen. Dazu nutzt er die Suchfunktion des Elib-Portals. Innerhalb der Eingabemaske des Portals trägt er dazu den Suchbegriff „Erneuerbare Energien“ ein. Er bestätigt das Abschicken des Formulars durch Drücken des Such-Buttons und erhält eine Liste von Suchergebnissen. Bei der Durchsicht dieser Ergebnisse stellt er fest, dass das System neben ihm bekannten Facetten auch eine Facette mit dem Titel „Verwandte Themengebiete“ zur Verfügung stellt. Innerhalb dieser Facette bekommt Max Mustermann den Eintrag „solarthermisches Kraftwerk“ angezeigt. Da Erneuerbare Energien nicht das Fachgebiet von Max Mustermann sind, freut er sich über die Vorschläge des Portals. Nachdem er alle für ihn relevanten Treffer der Ergebnisliste zum Thema Erneuerbare Energien abgearbeitet hat, klickt er auf den Eintrag „solarthermisches Kraftwerk“ in der Facette. Das System präsentiert ihm daraufhin eine Ergebnisliste mit Treffern zu diesem Suchbegriff. Diese Ergebnisse helfen ihm sehr bei seiner weiteren Arbeit, da er hier Veröffentlichungen von hoher Relevanz findet.

Tabelle 6.2.: Szenario S-2: Autovervollständigung der Sucheingabe

Szenario-ID:	S-2
Szenario-Name:	Autovervollständigung der Sucheingabe
Ziele:	Z-1.1: Transparenz der semantischen Technologien / Z-1.2: Darstellung in vorhandener GUI / Z-1.2.1: Autovervollständigung

Tabelle 6.2.: (Fortsetzung)

Szenario-Schritte:	Die wissenschaftliche Mitarbeiterin Irene Mustermann nutzt zur Suche nach wissenschaftlichen Dokumenten aus vergangenen DLR-Projekten das Elib-Portal. Ihre Suche zielt darauf ab, Veröffentlichungen aus der Vergangenheit zum Thema Fluglärm zu finden. Sie tippt in die Eingabemaske des Portals die Buchstaben „Flugl“ ein. Während der Eingabe dieser Buchstaben schlägt ihr das System eine Liste von Begriffen vor. Diese Liste beinhaltet nicht nur lexikalisch ähnliche Begriffe wie beispielweise „Fluglärmbelästigung“ sondern auch weitere Begriffe. Zum einen sind dies Begriffe mit ähnlicher Bedeutung wie z.B. „Aeroacoustic“ und Begriffe aus dem Themengebiet der Fluglärmforschung wie beispielsweise „Air Traffic Management“. Irene Mustermann findet die Autovervollständigung des Portals sehr hilfreich, da sie nicht alle Begriffe rund um die Thematik der Fluglärmforschung kennt. Sie klickt auf den Eintrag „Aeroacoustic“ und bekommt eine entsprechende Ergebnisliste zu diesem Suchbegriff angezeigt. Daraufhin stellt sie fest, dass diese Liste für ihre Recherche relevante Dokumente enthält.

Aus den zuvor geschilderten Szenarien und den darin enthaltenen Akteuren Max und Irene Mustermann lässt sich die in Tabelle 6.3 aufgeführte Benutzerrolle *wissenschaftlicher Mitarbeiter* ableiten. Diese Rolle charakterisiert einen DLR-Mitarbeiter, der eine fachübergreifende Recherche nach Informationen durchführt. Ein solcher Mitarbeiter hat nur einen oberflächlichen Einblick in die ihm fremden Fachbereiche. Im weiteren Verlauf dieser Ausarbeitung soll die zu konzipierende semantische Suche diese Benutzerrolle beim Auffinden von relevanten Informationen unterstützen.

Tabelle 6.3.: Benutzerrolle: Wissenschaftlicher Mitarbeiter

Rollenname:	Wissenschaftlicher Mitarbeiter
Aufgaben:	Fachübergreifende Suche von Informationen
Erfolgskriterien:	Auffinden von relevanten Informationen
Kommunikationspartner:	Elib-Portal
Innovationsgrad:	Hohe Internetaffinität
Typisches Benutzerprofil:	Erfahrung in der Literaturrecherche. Keine detaillierten Kenntnisse über alle Begrifflichkeiten in den fremden Fachbereichen. Keine tiefgehenden Kenntnisse über Verknüpfungen zwischen fachfremden Themengebieten. Sucht explorativ nach relevanten Informationen.

6.2. Qualitätsanforderungen & Rahmenbedingungen

Aus den Szenarien S-1 und S-2 werden in diesem Abschnitt die Qualitätsanforderungen und Rahmenbedingungen extrahiert. Qualitätsanforderungen werden häufig auch als nichtfunktionale Anforderungen bezeichnet. Diese Art der Anforderungen beziehen sich zum einen auf

Merkmale zur Laufzeit eines Systems wie z.B. Performanz oder Sicherheit und zum anderen auf Merkmale der Software-Erstellung wie Wartbarkeit oder Portabilität. Rahmenbedingungen (engl. constraints) sind Einschränkungen organisatorischer oder technologischer Art, die direkten Einfluss auf die Entwicklung eines Systems nehmen. Die Dokumentation dieser Einschränkungen dient dazu, eine vollständige Sichtweise auf ein zu entwickelndes Software-System zu erhalten.

Qualitätsanforderungen können den genannten Szenarien nicht entnommen werden. Zwar wären Merkmale wie eine bestimmte Performanz der Suche wünschenswert, jedoch steht diese Anforderungsart nicht im Fokus der durchgeführten Machbarkeitsstudie. Gerade im Rahmen einer forschenden Studie können diese Qualitätsparameter nicht abgesehen werden und spielen eine eher untergeordnete Rolle.

Weiterhin konnte den Szenarien die in C-1 aufgeführte Rahmenbedingung entnommen werden.

- **C-1: Basistechnologie Java**
  Da der KnowledgeFinder in der Programmiersprache Java implementiert ist, sind die Erweiterungen dieses Suchframeworks um semantische Technologien ebenfalls in dieser Programmiersprache umzusetzen.

7. Konzeption und Architekturentwurf der semantischen Suche

In diesem Kapitel findet die Konzeption der semantischen Suche statt. Aufbauend auf den in Kapitel 3 beschriebenen Grundlagen semantischer Technologien und den in Kapitel 4 genannten Aspekten semantischer Suchverfahren, wird dazu zunächst evaluiert, welcher Ansatz im Kontext des DLR geeignet ist. Hierbei spielen die in Kapitel 6 erhobenen Anforderungen eine zentrale Rolle. Diese Anforderungen werden zu Beginn des Kapitels zunächst analysiert, um darauf aufbauend einen geeigneten Ansatz zur semantischen Suche auswählen zu können. Die Integration des ausgewählten Lösungsansatzes innerhalb des KnowledgeFinder wird anschließend in Abschnitt 7.2.4 konzipiert und eine Architektur des Zielsystems vorgestellt.

7.1. Auswahl eines Ansatzes zur semantischen Suche

Die Auswahl eines Ansatzes zur semantischen Suche kann aus verschiedensten Blickwinkeln erfolgen. Die hier durchgeführten Untersuchungen finden auf Grundlage der von Tran und Mika eingeführten Aspekte zur Charakterisierung semantischer Suchverfahren statt (siehe Abschnitt 4.3.2). Die einzelnen Aspekte werden im Folgenden aufgeführt und im Kontext des KnowledgeFinder sowie der festgelegten Anforderungen analysiert. Auf Grundlage dieser Betrachtungen wird dann ein Ansatz zur semantischen Suche hergeleitet.

Zunächst ist zu untersuchen, welche *Art des Informationsbedürfnisses* durch die zu erstellende semantische Suche befriedigt werden soll. Da das Elib-Portal dazu dient, wissenschaftliche Veröffentlichungen zu durchsuchen, ist hier das klassische Informationsbedürfnis der Dokumentensuche anzutreffen. Auch im Zuge der Einführung einer semantischen Suche bleibt dieses Bedürfnis bestehen. Das bedeutet: hier ist weder eine Entitätssuche, noch eine Faktensuche oder gar eine relationale Suche notwendig. Das KnowledgeFinder-System ist also mit Blick auf das zu befriedigende Informationsbedürfnis im Sinne eines semantischen Dokumenten-Retrieval-Systems zu erweitern.

Einen weiteren Aspekt bildet die *Art der Anfrageformulierung*, welche durch die Benutzerschnittstelle zur Verfügung gestellt werden soll. Wie in Ziel Z-1.3 festgehalten, soll hier die schlüsselwortbasierte Suchanfrage unterstützt werden. Diese Art der Schnittstelle verbirgt die dahinter liegenden semantischen Funktionalitäten. Der Benutzer benötigt kein weiteres Wissen zur Nutzung der semantischen Suche. Dadurch kann gleichzeitig das Ziel der Transparenz der semantischen Technologien (Z-1.1) erfüllt werden. Neben der Schlüsselwortsuche wird die Nutzung der Facetten (Z-1.2.2) sowie die Unterstützung der explorativen Suche (Z-1.4) gefordert. Aufgrund dieser Tatsache ist die Schlüsselwortsuche durch eine facettierte Suche auf Seiten des User Interface (UI) zu ergänzen.

Eine Kernfrage bei der Realisierung einer semantischen Suche ist die Auswahl eines *semantischen Modells*. Die in Abschnitt 3.2 dargestellten Repräsentationsformen dienen dabei als Entscheidungsgrundlage. Zentrale Aspekte, die bei der Auswahl im weiteren Verlauf in Betracht gezogen werden, umfassen:

- die formale Aussagemächtigkeit,

- den Berechnungsaufwand und
- den Erstellungsaufwand

dieser Modelle. Die Aussagemächtigkeit der vorgestellten Modelle ist gleichzusetzen mit ihrer semantischen Reichhaltigkeit. Je mehr semantisches Wissen abgebildet werden kann, desto mächtiger ist das entsprechende Modell. Die in Abbildung 7.1 dargestellte *semantische Treppe* veranschaulicht die semantische Reichhaltigkeit der behandelten Modelle.

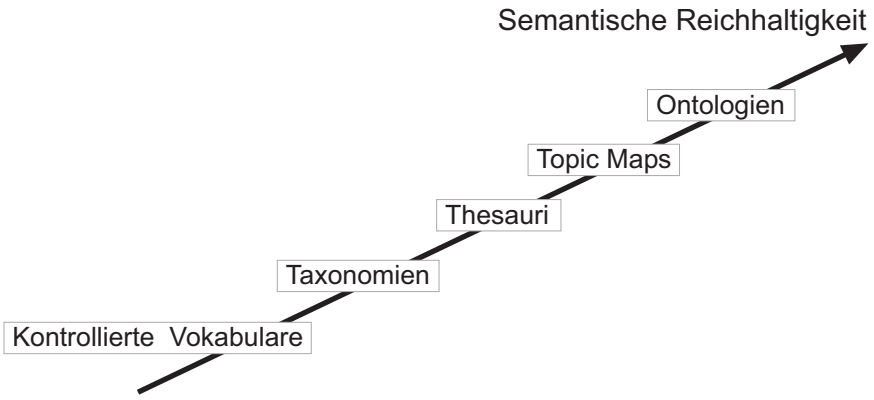


Abbildung 7.1.: Semantische Treppe
(vgl. [Pellegrini und Blumauer 2006] u. [Ullrich et al. 2004])

Lexikalische Modelle wie kontrollierte Vokabulare, Taxonomien und Thesauri haben die geringste semantische Reichhaltigkeit. Aufgrund der Tatsache, dass ein Thesaurus auch nichthierarchische Beziehungen abbilden kann, ist seine Ausdrucksfähigkeit unter diesen drei Modellen am stärksten. Topic Maps haben im Gegensatz zu Thesauri keine fest vorgegebenen Beziehungstypen und verfügen über weitreichendere Konzepte wie beispielsweise Typenbildung und Gültigkeitsbereiche. Aufgrund dieser Flexibilität bietet dieses Modell nochmals eine höhere semantische Reichhaltigkeit. Ontologien haben die größte Ausdrucksfähigkeit. Mit Ontologien können sehr komplexe Wissensgebiete abgebildet werden. Dafür ist der Berechnungs- und Erstellungsaufwand dieser Repräsentationsform deutlich höher als bei den lexikalischen Modellen. Wie im Rahmen der Semantic-Web-Standards erläutert, ist beispielsweise der volle Sprachumfang der Ontologiesprache OWL nicht entscheidbar (siehe Abschnitt 3.3.4).

Der Erstellungsaufwand spielt bei der Auswahl eines semantischen Modells eine zentrale Rolle. Wie und mit welchem Aufwand kann das real-weltliche Wissen in ein entsprechendes formales Modell überführt werden? In diesem Kontext spricht Cimiano auch vom sogenannten *Flaschenhals der Wissensakquirierung* (engl. knowledge acquisition bottleneck) [Cimiano 2006, S. 4]. Die Überführung einer komplexen Domäne in ein formales Modell stellt eine schwierige und sehr zeitaufwändige Aufgabe dar. Sie ist der Flaschenhals bei der Erstellung semantischer Modelle. Im Rahmen dieser Machbarkeitsstudie liegt eine solch komplexe Domäne in Form des DLR mit seinen unterschiedlichsten Forschungsgebieten vor. So besteht beispielsweise im Forschungsgebiet der Flugzeug-Aerodynamik die Herausforderung darin, effiziente numerische Algorithmen zu entwickeln, welche die Wirklichkeit möglichst detailgetreu abbilden. Im Forschungsfeld der Erdbeobachtung hingegen gilt es riesige stetig anfallende Datenmengen zu speichern sowie auszuwerten.

Mit Blick auf die in Abbildung 7.1 veranschaulichte Reichhaltigkeit der verschiedenen semantischen Modelle wäre die Überführung dieser Strukturen in eine Ontologie erstrebenswert. Ontologien sind im Kontext der semantischen Suche die Repräsentationsform, aus der am meisten Information gewonnen werden kann. Aufgrund ihrer Eigenschaften können nicht nur komplexe Themengebiete abgebildet werden, sondern auch neues Wissen durch Inferenz gewonnen werden. Daneben stellt eine Ontologie im Rahmen des hier angestrebten semantischen Dokumenten-Retrieval das am besten geeignete Modell zur semantischen Anreicherung des Dokumenten-Suchraumes mit Informationen dar. In Verbindung mit entsprechenden Mapping- und Annotations-Methoden kann eine Ontologie dazu genutzt werden, Entitäten und Relationen in semi- bzw. unstrukturierten Daten zu erfassen. Sie spielt somit eine wichtige Rolle bei der semantischen Anreicherung des Dokumenten-Suchraumes (vgl. Abschnitt 4.2).

Bezüglich des Erstellungsaufwandes steht im Folgenden die Frage im Mittelpunkt, ob und wie es möglich ist, die komplexen und vielschichtigen Strukturen des DLR in eine Ontologie überführen zu können. Nach Cimiano liegt dabei die Herausforderung darin, die gesamte Domäne abzubilden und gleichzeitig das Modell so abstrakt wie nur möglich zu halten [Cimiano 2006, S. 4 f.]. Daneben ist das Schaffen eines gemeinsamen Verständnisses ebenfalls eine große Hürde bei der Erstellung einer Ontologie (siehe auch Abschnitt 3.2.4). Der Ansatz der automatisierten Erstellung von Ontologien stellt in diesem Kontext eine ideale Lösung dar. Das Ziel dieser Herangehensweise ist das automatisierte Erlernen von Ontologien auf der Grundlage von semi- und unstrukturierten Daten. Dieser Ansatz geht davon aus, dass je größer die zugrundeliegende Datenbasis ist, desto höher ist auch die Abdeckung der betrachteten Domäne. Unter der Annahme, dass die Inhalte der Datenbasis von verschiedenen Autoren stammen, ist nach Cimiano auch ein hohes gemeinsames Verständnis des entstehenden Modells zu erwarten [Cimiano 2006, S. 5]. Mit Blick auf die hier behandelte Domäne wäre also der Ansatz einer automatisierten Erstellung der gangbare Weg. Eine automatisierte Wissensextraktion würde neben dem Problem des Knowledge Acquisition Bottleneck auch dem ständigen Wandel dieser Domäne Rechnung tragen. Gerade forschungsorientierte Gebiete sind stetigen Änderungen unterlegen. Es werden laufend neue Forschungsfelder erschlossen und neue Themengebiete behandelt. Dieser Wandel müsste auch in dem zugrundeliegenden semantischen Modell vollzogen werden, da sonst mit nicht aktuellen Modellen gearbeitet werden würde. Eine Automatisierung ermöglicht also nicht nur die vereinfachte Erstellung eines semantischen Modells, sondern auch die nachgelagerte Pflege und Anpassung.

Es ist also zu untersuchen, ob eine vollständige Automatisierung oder zumindest eine Teilautomatisierung der Erstellung einer Ontologie des DLR möglich ist. In der Literatur existieren eine Vielzahl von Ansätzen wie beispielsweise die lexiko-syntaktische Analyse, Methoden der Information-Extraktion, Clustering-Verfahren oder Kookkurrenz-Analysen [Wei et al. 2008]. Diese Ansätze beruhen alle auf der Annahme, dass die Bedeutung von Begriffen aus der Art und Weise ihrer Verwendung hervorgeht [Cimiano 2006, S. 311]. Cimiano hebt hervor, dass solche Verfahren hohe Fehleranfälligkeiten besitzen. Diese Tatsache hat zur Konsequenz, dass zwangsläufig die Notwendigkeit besteht, eine automatisiert erzeugte Ontologie nachgelagert durch einen Menschen zu validieren und gegebenenfalls anzupassen [Cimiano 2006, S. 6]. Bedini und Nguyen bestätigen in ihrer Untersuchung verschiedener automatisierter Systeme die Fehleranfälligkeit dieser Verfahren. Sie kommen zu dem Schluss, dass keines dieser Systeme brauchbare Ergebnisse liefert [Bedini und Nguyen 2007]. Die Einordnung dieses Problems von Feigenbaum aus dem Jahre 2003 unterstreicht die hier beschriebene Tatsache ebenfalls.

Er bezeichnet die Gewinnung von Wissen aus Texten als eine der größten Herausforderungen der Informatik [Feigenbaum 2003]. Es bleibt also die Überlegung einer möglichen Teilautomatisierung der Ontologie-Erstellung. Bei diesem Ansatz werden zunächst automatisierte Verfahren zur Ontologie-Erstellung angewendet und die erzeugten Modelle anschließend einer menschlichen Überprüfung unterzogen. In diesem Zusammenhang ist jedoch fraglich, von wem die notwendigen Validierungsmaßnahmen durchgeführt werden sollten. Aufgrund der hohen Komplexität der Domäne sind für eine Validierung tiefe Einblicke in die jeweiligen Themengebiete der einzelnen DLR-Institute erforderlich. Ohne dieses Wissen ist es nicht möglich zu entscheiden, ob das erzeugte Modell sinnvoll ist oder nicht. Auch der Ansatz, die Themengebiete einzeln zu betrachten und von den jeweiligen Experten kontrollieren zu lassen, würde hier keine Lösung bringen, da eine Zusammenführung von Teil-Ontologien ebenfalls keine triviale Aufgabe darstellt (vgl. [Euzenat und Shvaiko 2013]). Dengel hebt hervor, dass logikorientierte Ansätze zur Zusammenführung von Ontologien zwar logisch korrekte Ergebnisse liefern, jedoch ist ihre Umsetzung eine sehr komplexe und wissensintensive Aufgabe [Dengel 2012, S. 138 f.]. Im Gegensatz dazu, sind heuristische Verfahren zwar einfacher anwendbar, jedoch kann deren Korrektheit nicht generell gewährleistet werden. Auch eine Teilautomatisierung der Ontologie-Erstellung ist mit Blick auf die zuvor beschriebenen Problematiken der Ontologie-Validierung und -Zusammenführung nicht sinnvoll.

Es kann also festgehalten werden, dass aufgrund des Knowledge Acquisition Bottleneck und der damit verbundenen Problematiken Ontologien trotz ihrer hohen semantischen Reichhaltigkeit im Kontext der hier betrachteten Domäne nicht geeignet sind. Somit ist eine Repräsentationsform mit geringerer Aussagemächtigkeit zu wählen. Mit Blick auf die noch vorhandenen Modelle ist die Frage zu klären, ob die einfacheren lexikalischen Modelle oder die mächtigeren Topic Maps Verwendung finden sollten. Wie schon in vorangegangen Passagen erläutert, sind kontrollierte Vokabulare aufgrund ihrer fehlenden Struktur für komplexe Aufgabenstellungen nicht geeignet. Auch Taxonomien sind mit Blick auf die Anforderungen nicht das Mittel der Wahl, da sie nur eine eingeschränkte Möglichkeit zur Relationsbildung besitzen. Die Abbildung der untereinander stark vernetzten DLR-Themengebiete innerhalb einer Baumstruktur ist als nicht sinnvoll zu erachten. Verbleiben also die Modelle des Thesaurus und der Topic Maps. In diesem Zusammenhang sind folgende Abwägungen relevant, die im Anschluss diskutiert werden:

- Mit welcher Repräsentationsform können die Anforderungen am besten umgesetzt werden?
- Welches Modell ist im Kontext des semantischen Dokumenten-Retrieval am besten geeignet?
- Wie praktikabel ist die Erstellung dieser Modelle?

Mit Blick auf die in Kapitel 6 festgelegten Anforderungen ist zu untersuchen, welche Repräsentationsform für die Umsetzung zweckmäßiger ist. Für die in den Szenarien S-1 und S-2 beschriebene Vorschlags-Funktion ist das leichtgewichtige Modell des Thesaurus wie geschaffen. Die Abbildung verwandter Themengebiete wäre zwar auch mit Hilfe einer Topic Map möglich, aber ein Thesaurus bietet hier genau die gewünschten Funktionalitäten. Ein Themengebiet könnte dabei als Konzept innerhalb des Thesaurus abgebildet werden. Die Verbindungen zwischen den einzelnen Themengebieten könnten mit Hilfe hierarchischer und nicht hierarchischer Beziehungen in diesem Modell ausgedrückt werden. Dadurch würde eine semantische Landkarte der DLR-Themengebiete entstehen. Weiterhin gilt es zu eruieren welches Modell zur Befriedigung des Informationsbedürfnisses vorzuziehen ist. Im Kontext des

semantischen Dokumenten-Retrieval bieten Topic Maps den Vorteil der direkten Verknüpfung von Wissen mit dazugehörigen Dokumenten. Thesauri bieten diese Verknüpfungsmöglichkeit nicht, da sie losgelöst von der zugrundeliegenden Datenbasis sind. Hier wäre also ein Matching zwischen verwandten Konzepten und deren Vorkommen im Dokumenten-Suchraum notwendig.

Zuletzt ist abzuwägen, ob anwendbare Ansätze zur Erstellung der beiden diskutierten Modelle vorhanden sind. Aufgrund der zuvor erörterten Problematiken bzgl. des Erstellungsaufwands ist auch im Fall der weniger mächtigen Modelle zumindest eine Teilautomatisierung der Wissensabbildung notwendig. Die Notwendigkeit einer Teilautomatisierung bildet demnach im gegebenen Kontext einen zentralen Baustein zur Erstellung eines semantischen Suchsystems. Für die im Rahmen der Teilautomatisierung durchzuführende automatische Konstruktion eines Thesaurus sind im Wesentlichen drei Schritte notwendig. Zunächst müssen relevante Terme aus einem Text-Korpus gewonnen werden. Im nächsten Schritt ist eine Ähnlichkeitsbestimmung durchzuführen und zuletzt eine Hierarchiebildung zwischen diesen Termen notwendig [Meusel 2009, S. 21]. Bei der automatischen Konstruktion von Topic Maps sind prinzipiell ähnliche Vorgehensweisen erforderlich. Hier besteht allerdings die Herausforderung darin, die Art der Assoziationen zu bestimmen. Auch zur automatischen Konstruktion von Thesauri und Topic Maps kommen einige der schon zuvor genannten Verfahren zum Einsatz. Zu nennen sind hier lexiko-syntaktische Analysen, Clustering-Verfahren und Kookkurrenz-Analysen (vgl. [Manning et al. 2009, S. 192], [Ellouze et al. 2012] und [Meusel 2009, S. 24]). Aufgrund der Einfachheit des Thesaurus-Modells kann davon ausgegangen werden, dass die Fehleranfälligkeit dieser Konstruktionsmethoden geringer ausfällt als bei den abstrakteren Topic Maps. Gerade mit Blick auf die Anforderungen ist das Thesaurus-Modell das passende Mittel der Wahl. Es bietet entsprechende Strukturen zur Abbildung der Themenlandschaften. Auch mit Blick auf eine nachgelagerte Evaluierung ist diese Wissensrepräsentationsform im Kontext dieser Ausarbeitungen vorzuziehen. Die Pflege eines Thesaurus ist aufgrund der klar definierten Beziehungstypen einfacher als bei Topic Maps. Dies ist ein weiterer Vorteil dieses Modells. Insgesamt kann also festgehalten werden, dass der Thesaurus als zugrundeliegendes semantisches Modell zur Wissensrepräsentation am besten geeignet ist.

Tabelle 7.1 fasst die Ergebnisse der zuvor durchgeführten Bewertung der verschiedenen Modelle nochmals zusammen. Das ausgewählte Modell des Thesaurus ist zur besseren Unterscheidung hervorgehoben.

Tabelle 7.1.: Ergebnisse der Bewertung semantischer Modelle

Modell	**Aussage-mächtig-keit**	**Berech-nungs-aufw.**	**Erstel-lungs-aufw.**	**Umsetzbarkeit Anforderungen**	**Doku-menten-Retrieval**
Kontr. Vokabulare	– –	++	++	– –	– –
Taxonomien	–	++	++	– –	– –
Thesauri	o	++	++	++	+
Topic Maps	+	+	+	+	++
Ontologien	++	o	– –	+	++

Schema zur Bewertung der Eignung:
– – sehr schlecht – schlecht o eingeschränkt + gut ++ sehr gut

Zuletzt sind die Aspekte der *Repräsentationsform der Informationsquellen* und der *Art des verwendeten Rahmenwerks zur Interpretation der Daten und Anfragen* zu behandeln. Die Art der Informationsquellen ergibt sich aus den im KnowledgeFinder vorliegenden Daten. Im Fall des Elib-Portals handelt es sich dabei um wissenschaftliche Dokumente, die mit Hilfe von Metadaten beschrieben sind. Das bedeutet, die Informationsquellen umfassen sowohl strukturierte als auch unstrukturierte Arten. Das verwendete Rahmenwerk zur Dateninterpretation und Anfrageverarbeitung ist abhängig vom gewählten semantischen Modell, da je nach Typ entsprechende Verfahren einzusetzen sind. Wie zuvor herausgestellt, wird hier ein Thesaurus genutzt werden. Das bedeutet, es sind Methoden zur automatischen Konstruktion dieser Modelle einzusetzen.

Zusammenfassend kann aufgrund der zuvor durchgeführten Untersuchungen geschlussfolgert werden, dass die Erweiterung des KnowledgeFinder nach dem Prinzip des konzeptbasierten Dokumenten-Retrieval erfolgt (siehe Abschnitt 4.3.2). Dieses Prinzip wird im Folgenden auch als *konzeptbasierte semantische Suche* bezeichnet. Im weiteren Verlauf wird also ein System entworfen, dass eine Schlüsselwortsuche von Dokumenten auf Basis von Konzepten und deren Relationen ermöglichen soll. Das leichtgewichtige lexikalische Modell des Thesaurus soll dabei zur Abbildung des Wissens herangezogen werden und als Wissensbasis innerhalb des semantischen Suchsystems eingesetzt werden. Die Konzeption dieser semantischen Suche wird im nachfolgenden Abschnitt detailliert dargestellt.

7.2. Konzeption einer konzeptbasierten semantischen Suche

Der im vorherigen Abschnitt hergeleitete Ansatz der konzeptbasierten semantischen Suche wird im Folgenden auf den vorliegenden Anwendungsfall übertragen und die dafür notwendigen Erweiterungen werden entworfen. Die Grundüberlegungen sind dabei wie folgt: Zunächst ist die Überführung von Domänenwissen in die gewählte Repräsentationsform des Thesaurus notwendig. Hier ist eine Teilautomatisierung dieser Aufgabe zu konzipieren. Innerhalb dieser Teilautomatisierung müssen Konzepte und deren semantische Beziehungen aus den Elib-Daten extrahiert und in einen Thesaurus überführt werden. Das Ziel der Extraktion besteht dabei darin, dass sich die Begrifflichkeiten der DLR-Themengebiete innerhalb dieser Konzepte widerspiegeln. Die so identifizierten Konzepte sollen nachgelagert einer manuellen Bearbeitung unterzogen werden können. Die Konzepte und deren Beziehungen werden durch den KnowledgeFinder als semantische Hilfsmittel bei der explorativen Suche zur Verfügung gestellt werden. Wie in den Anforderungen beschrieben, besteht die Hilfestellung dabei darin, die durch den Thesaurus aufgespannte semantische Themenlandkarte kontextabhängig an der Benutzeroberfläche anzuzeigen. Das heißt, es werden Querbezüge wie Oberbegriffe, Unterbegriffe oder Synonyme zur der vom Nutzer durchgeführten Schlüsselwortsuche innerhalb des Thesaurus identifiziert und in der Benutzeroberfläche dargestellt. Die Darstellung der Hilfestellungen soll zum einen mit Hilfe der schon vorhandenen Facetten und zum anderen in Form einer Autovervollständigung in der GUI erfolgen (siehe Szenario S-1 & S-2 in Kapitel 6).

Um das zuvor aufgezeigte Gesamtkonzept umsetzen zu können, findet im Folgenden eine Bottom-Up-Beschreibung statt. Zunächst werden einzelne Teilkonzepte erarbeitet und anschließend im Gesamtbild dargestellt. Die Teilkonzepte befassen sich dabei mit folgenden Thematiken:

1. **Format zur Repräsentation der Thesaurus-Inhalte**
   Um eine semantische Suche nach dem Prinzip des konzeptbasierten Dokumenten-Retrieval

zur ermöglichen, ist entsprechenden Format zur Repräsentation der Thesaurus-Inhalte notwendig.

2. **Teilautomatisierte Konstruktion des Thesaurus**
   Im Rahmen der teilautomatisierten Konstruktion des Thesaurus sind Überlegungen anzustellen, wie die Informationsquellen des Elib-Systems semantisch aufbereitet und als Konzepte innerhalb des Thesaurus abgelegt werden können. Daneben gilt es ein Verfahren zur Ermittlung von Beziehungen zwischen diesen Konzepten zu entwerfen. Zur Nachbearbeitung des erstellten Thesaurus ist zudem eine Benutzerschnittstelle notwendig.

3. **Schlüsselwortbasierte Suche nach Konzepten**
   Da auf Seiten der Anfrageformulierung die Schlüsselwortsuche unterstützt werden soll, sind Verfahren, zum effizienten schlüsselwortbasierten Durchsuchen der Thesaurus-Inhalte zu konzipieren.

Eine detaillierte Beschreibung dieser Teilkonzepte findet in den nachfolgenden Abschnitten statt. Danach werden diese Teilkonzepte in Abschnitt 7.2.4 in den Gesamtkontext des KnowledgeFinder eingeordnet. Hier wird erörtert, welche Erweiterungen und Komponenten zur Realisierung der Teilkonzepte notwendig sind. Weiterhin findet eine Schilderung des Zusammenspiels der einzelnen Komponenten statt. Zuletzt wird in Abschnitt 7.2.5 dargelegt, welche Erweiterungen auf Seiten der Benutzeroberfläche des Elib-Portals durchzuführen sind.

7.2.1. Format zur Repräsentation der Thesaurus-Inhalte

Es existieren verschiedene Standardisierungen, mit denen ein Thesaurus abgebildet werden kann. Im betrachteten Kontext spielen mehrere Faktoren eine entscheidende Rolle. Zum einen ist relevant, inwieweit die jeweiligen Formate für den Einsatzzweck der semantischen Suche geeignet sind. Zum anderen ist die Möglichkeit der Persistierung dieser Datenformate zu betrachteten. Mit Blick auf diese beiden Faktoren soll das vom W3C herausgegebene SKOS-Format zur Abbildung des Thesaurus genutzt werden (siehe Abschnitt 3.2.2). Der Vorteil dieses Formates liegt im Vergleich zu anderen Standardisierungen darin, dass SKOS auf den Semantic-Web-Standards aufbaut. Das heißt, diese W3C-Empfehlung dient dazu, Informationen maschinenlesbar zu machen. Gleichzeitig garantiert die Verwendung der Semantic-Web-Standards als Basistechnologie ein hohes Maß an Interoperabilität [Miles und Bechhofer 2009]. Wie bei Thesauri üblich, wird Wissen innerhalb des SKOS-Datenmodells mit Hilfe von Konzepten repräsentiert. Ein SKOS-Konzept kann dabei mit Hilfe eines URI eindeutig identifiziert werden. Für die unterschiedlichen Relationstypen des Thesaurus-Modells stellt die W3C-Empfehlungen eigene RDF Properties bereit (siehe Abschnitt 3.3). So wird die Beziehung zu einem Oberbegriff beispielsweise mit der `<skos:broader>`-Relation ausgedrückt. Konzepte und deren Eigenschaften werden mit Hilfe von RDF-Tripeln ausgedrückt. Diese Art der Datenrepräsentation ermöglicht eine einfache Persistierung eines SKOS-basierten Thesaurus mit Hilfe eines *RDF Triple Store*. Neben der Speicherung von RDF-Daten bieten Triple Stores Schnittstellen zur graphbasierten Abfrage ihrer Inhalte mit Hilfe von SPARQL. Aufgrund der zuvor herausgestellten Eigenschaften von SKOS ist dieser Standard optimal für die Umsetzung der hier konzipierten semantischen Suche geeignet. Zur Persistierung der Thesaurus-Inhalte ist ein Triple Store als zugrundeliegender Datenspeicher vorzusehen.

7.2.2. Teilautomatisierte Erstellung des Thesaurus

In diesem Abschnitt wird die teilautomatisierte Erstellung des Thesaurus dargestellt. Das Ziel besteht darin, Konzepte und deren semantische Beziehungen aus den Elib-Daten zu extra-

hieren und in einen SKOS-Thesaurus zu überführen. Dazu sind mehrere Schritte notwendig. Zunächst müssen relevante Konzepte innerhalb der Informationsquellen identifiziert werden. Der zweite Schritt besteht darin, Beziehungen zwischen den Konzepten zu ermitteln. Im letzten Schritt sollte der so erzeugte Thesaurus einer Validierung und Anpassung unterzogen werden können. Für die ersten beiden Schritte sind geeignete Ansätze anzuwenden. Damit eine Validierung durchgeführt werden kann, ist eine Benutzerschnittstelle zur Bearbeitung des Thesaurus notwendig. Im Folgenden soll auf die einzelnen Schritte detailliert eingegangen werden. Anzumerken ist, dass es im Kontext dieser Ausarbeitung nicht möglich ist, sämtliche Verfahren der automatisierten Erstellung von Thesauri zu untersuchen und gegeneinander abzuwägen.

Extraktion relevanter Konzepte

Bei der Extraktion relevanter Konzepte geht es im weiteren Verlauf darum, zu ermitteln, mit welchen Thematiken sich die einzelnen Elib-Dokumente beschäftigen. Diese Thematiken werden als Konzepte der betrachteten Domäne aufgefasst und im Thesaurus abgebildet. Liegt beispielsweise ein wissenschaftliches Paper zum Thema Erneuerbare Energien vor, so wird dieses Thema als Konzept im Thesaurus abgelegt. Im ersten Schritt besteht also die Aufgabe darin, die Inhalte und somit die Thematiken der einzelnen Elib-Dokumente zu extrahieren. Dabei wird zunächst die Tatsache ausgenutzt, dass im zugrundeliegenden Elib-System ein Metadatum zur Vergabe von Schlüsselwörtern vorgesehen ist. Dieses in Elib mit „Stichwörter“ bezeichnete Attribut wird durch den Verfasser einer wissenschaftlichen Ausarbeitung gefüllt. Schlüsselwörter sind dazu vorgesehen, den Inhalt eines Elib-Eintrages mit umgangssprachlichen Begriffen zu beschreiben. Die Schlüsselwörter spiegeln somit die Themen, von denen die Einträge handeln, wider. Abbildung 7.2 zeigt beispielhaft die Schlüsselwörter eines Elib-Eintrages[1] (siehe rote Umrandung).

Begutachtungsstatus :	referierte Publikation (nicht in Journal Citation Report)
Stichwörter:	Solar concentrating system, central receiver, shielding design, ray tracing, flux mapping, optical errors
Veranstaltungstitel:	ANZSES 2006

Abbildung 7.2.: Elib-Eintrag mit vom Nutzer vergebenen Schlüsselwörtern

Die Schlüsselwörter der Elib-Einträge können also als Konzepte aufgefasst werden. Unter Schlüsselwörtern werden an dieser Stelle nicht nur einzelne Terme, sondern auch Phrasen, also zusammengesetzte Wörter (engl. keyphrases), verstanden.

Um einen Überblick über die Vergabe der Schlüsselwörter zu erhalten, wurde des Weiteren eine Analyse durchgeführt. Ziel war es, zu ermitteln, wie viele Einträge Schlüsselwörter besitzen und wie viele Schlüsselwörter pro Eintrag durchschnittlich vergeben sind. Tabelle 7.2 zeigt das Ergebnis dieser Analyse[2].

[1] Der Eintrag ist unter http://elib.dlr.de/53306 abrufbar. (Zugriffsdatum: 12.01.2013)
[2] Grundlage der Analyse war der Elib-Datenbestand vom 29.11.2012.

Tabelle 7.2.: Schlüsselwort-Abdeckung des Elib-Datenbestands

#[3] Einträge Gesamt	68.925
# Einträge mit Schlüsselwörtern	42.213
# Schlüsselwörter	175.510
Ø Schlüsselwörter pro Eintrag	≈ 4

Der Gesamtprozentsatz der Einträge mit Schlüsselwörtern ist mit 61,24 % recht gering. Dokumente, welche Schlüsselwörter besitzen, haben im Durchschnitt ca. 4 vergebene Schlüsselwörter. Mit Blick auf die Komplexität der behandelten Themengebiete ist auch dieser Wert als relativ niedrig einzuschätzen. Aufgrund dieser Tatsachen besteht neben der Verwendung der vom Nutzer vergebenen Schlüsselwörter die Notwendigkeit, ein automatisiertes Verfahren zur Extraktion von Schlüsselwörtern aus den Elib-Dokumenten zu entwickeln. Für dieses Verfahren soll das Java Open Source Framework KEA genutzt werden. KEA steht für Keyphrase Extraction Algorithm und wurde an der Universität von Waikato entwickelt [KEA 2013]. Da das Framework ein maschinelles Lernverfahren einsetzt, unterscheidet der Algorithmus grundsätzlich zwei Phasen: eine Trainingsphase sowie eine Extraktionsphase. Bevor die eigentliche Extraktion von Schlüsselwörtern stattfinden kann, ist zunächst eine Trainingsphase notwendig. Anschließend kann in der Extraktionsphase die Ermittlung von neuen Schlüsselwörtern aus einem Text-Korpus stattfinden. Aufgrund dieses Lernverfahrens kann das Framework in beliebigen Anwendungsfällen genutzt werden. Um die Eignung von KEA im Rahmen dieser Machbarkeitsstudie zu untermauern, wurden Test-Extraktionen auf der Basis von Elib-Daten durchgeführt. Die Ergebnisse der Test-Extraktionen wurden anschließend mit den nutzervergebenen Schlüsselwörtern verglichen. Listing 7.1 zeigt ein Beispiel dieser Test-Extraktionen. Die hier gewonnenen Schlüsselwörter beziehen sich auf den zuvor in Abbildung 7.2 beschriebenen Elib-Eintrag. Wie im Vergleich damit zu erkennen ist, beinhaltet die automatische Extraktion einen Großteil der vom Nutzer vergebenen Schlüsselwörter.

```
1  solar
2  ray tracing
3  flux mapping
4  central receiver
5  flux
6  ray
7  design
8  shielding design
9  optical errors
10 design and research
```

Listing 7.1: Schlüsselwörter einer KEA-Test-Extraktion

Die Funktionsweise des KEA-Frameworks soll im Folgenden kurz aufgezeigt werden. Die Hauptschritte des Algorithmus sind in Abbildung 7.3 schematisch dargestellt. Zur Identifikation von möglichen Schlüsselwörtern nutzt KEA im Rahmen der Extraktion von Kandidaten lexikalische Methoden des IR (siehe Kapitel 2). Hier wird unter anderem der Eingabetext von Sonderzeichen befreit und verschiedene Stemming-Verfahren angewendet [Witten et al. 1999]. Für jeden möglichen Kandidaten werden anschließend mehrere Eigenschaften wie beispielsweise die TF-IDF-Gewichtung oder das erste Auftreten berechnet. Findet eine Trainingsphase statt, so wird mit einem Satz von Beispieldokumenten und einem maschinellen Lernverfahren

[3] # = Anzahl

ein domänenspezifisches Modell erzeugt. Dieses Modell wird in der anschließenden Extraktionsphase zur automatischen Bestimmung von Schlüsselwörtern genutzt. Für die Trainingsphase sind Dokumente mit manuell vergebenen Schlüsselwörtern notwendig. Auf Grundlage des trainierten Modells kann in der Extraktionsphase auf einem globalen Text-Korpus gearbeitet werden. Hier werden die zuvor berechneten Eigenschaften der Kandidaten mit dem erlernten Modell verglichen und die Wahrscheinlichkeit für das Vorliegen eines Schlüsselwortes bestimmt.

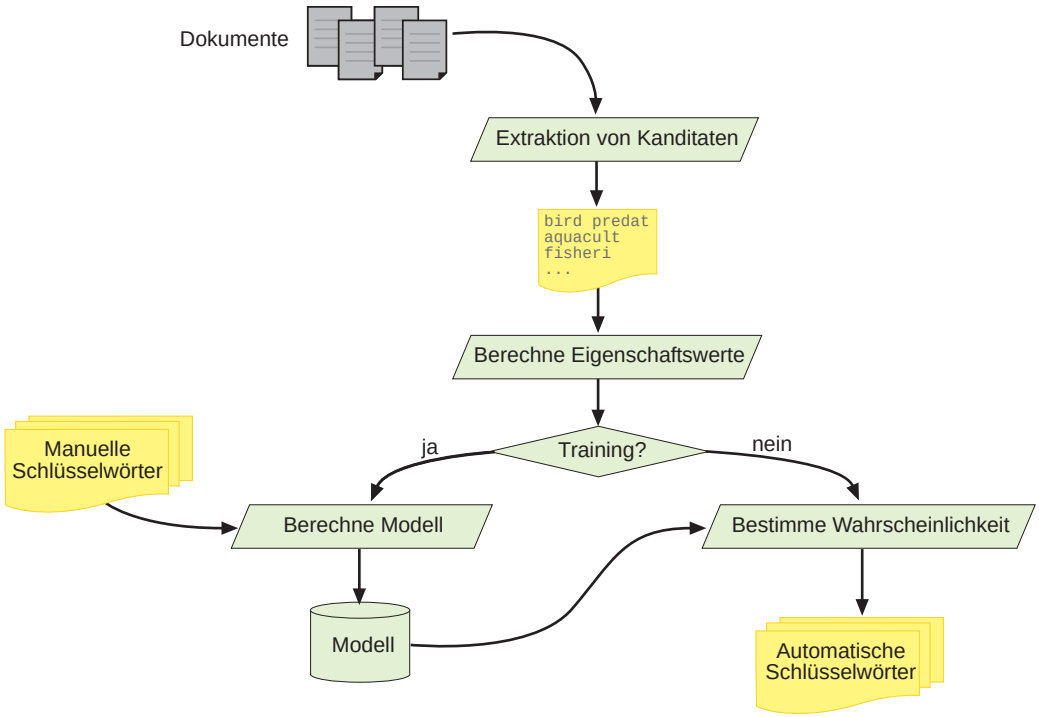


Abbildung 7.3.: Hauptschritte des KEA-Algorithmus (vgl. [KEA 2013])

Da sowohl die Trainings- als auch die Extraktionsphase des KEA-Frameworks aufgrund der Stemming-Verfahren sprachabhängig sind, ist im Rahmen der Extraktion zudem ein Verfahren zur Spracherkennung notwendig. Hierzu soll die Methode der N-Gramm-Statistik eingesetzt werden. Bei diesem Verfahren wird die Wahrscheinlichkeit für das Vorliegen einer bestimmten Sprache mit Hilfe sogenannter *N-Gramme* ermittelt. Ein N-Gramm umfasst dabei die Menge aller Wörter der Länge N. Die N-Gramm-Häufigkeit variiert dabei je nach Sprache [Dengel 2012, S. 209].

Bestimmung von Relationen

Als nächstes ist ein Verfahren zur Bestimmung von Relationen zwischen den als Konzepte aufgefassten Schlüsselwörtern zu erarbeiten. Hier wird der Ansatz der Kookkurrenz-Analyse verfolgt. Bei diesem Ansatz wird von der vereinfachten Annahme ausgegangen, dass Wörter, die häufig miteinander in einem Text in Erscheinung treten, in irgendeiner Art Beziehung zueinander stehen [Manning et al. 2009, S. 192]. Dieses Auftreten wird als Kookkurrenz zweier Terme bezeichnet. Die Kookkurrenz-Analyse ist ein Verfahren der Computerlinguistik und dient der Ermittlung dieser Eigenschaft. Treten Wörter statistisch gesehen häufiger innerhalb

eines Textes miteinander auf, so spricht man auch von einer Kollokation [Pellegrini und Blumauer 2006, S. 445]. Mit Hilfe einer Kookkurrenz-Analyse kann also festgestellt werden, ob eine Kollokation vorliegt. Nach der sogenannten *Distributionellen Hypothese* kann in diesem Fall von einer semantischen Ähnlichkeit zwischen Wörtern ausgegangen werden [Manning et al. 2009, S. 44]. Im Folgenden wird dieses Verfahren dazu genutzt, um Beziehungen zwischen den Schlüsselwörtern zu ermittelten. Da mit Hilfe einer Kookkurrenz-Analyse keine Unterscheidung bzgl. der Art der Beziehung getätigt werden kann, findet eine Beschränkung auf die Thesaurus-Beziehung der Assoziation statt. Schlüsselwörter, zwischen denen aufgrund ihrer Kollokation eine semantische Ähnlichkeit besteht, werden innerhalb des Thesaurus als verwandt markiert. Im Rahmen von SKOS entspricht dies dem RDF-Prädikat `<skos:related>`.

Bei der Relationsbestimmung soll die Open-Source-Anwendung DISCO eingesetzt werden [DISCO 2013]. Die Abkürzung DISCO steht für „extracting DIstributionally related words using CO-occurrences“. Diese Java-Anwendung erlaubt eine Ähnlichkeitsbestimmung zwischen Wörtern auf Basis ihrer Kookkurrenz. Dazu verfügt dieses Tool über eine im Voraus berechnete Kollokations-Datenbank in Form eines invertierten Indexes. Bei der Ähnlichkeitsbestimmung zweier Wörter werden Wort-Metriken aus diesen Datenbanken ausgelesen und die Ähnlichkeit auf Basis eines informations-theoretischen Maßes bestimmt [Kolb 2008]. Das Ergebnis einer solchen Berechnung kann zwischen 0 (keine Ähnlichkeit) und 1 (sehr hohe Ähnlichkeit) liegen. Da Kollokationen sprachabhängig sind, sind für verschiedene Sprachen einzelne Datenbanken vorhanden. So bieten die Tool-Autoren eine englischsprachige Datenbank an, die auf der Basis von ca. 300.000 englischen Wikipedia-Artikeln generiert wurde [Kolb 2008]. Für die deutsche Sprache ist ebenfalls eine Datenbank vorhanden, welche auf deutschen Wikipedia-Artikeln beruht. Hier wurde laut Angabe der Autoren die Übersetzungsfunktion des Wikipedia-Systems ausgenutzt. Die Tatsache, dass DISCO sowohl für die englische als auch die deutsche Sprache geeignet ist, ist ein wichtiges Kriterium im Kontext des Elib-Datenbestands. Auch das Vorhandensein von vorberechneten Datenbanken ist als Vorteil anzusehen. Dies hat auf Seiten des KnowledgeFinder eine erhebliche Reduktion des Berechnungsaufwandes zur Bestimmung der Relationen zur Konsequenz. Weiterhin kann, im Gegensatz zu anderen Ähnlichkeitsbestimmungs-Tools, bei DISCO die zugrundeliegende Datenbank lokal genutzt werden. Oftmals stehen hier lediglich webbasierte APIs zur Verfügung. Aufgrund der Sensibilität der DLR-Daten ist die Nutzung solcher Anwendungen nicht möglich.

Da es vorkommen kann, dass Schlüsselwörter nicht durch die DISCO-Datenbanken abgedeckt werden, ist es möglich, dass die Kookkurrenz-Analyse nicht in jedem Fall anwendbar ist. In diesem Fall soll eine einfache syntaktische Ähnlichkeitsbestimmung zwischen den Zeichenketten der betrachteten Schlüsselwörter stattfinden. Hierzu soll auf die Open-Source-Bibliothek SimMetrics zurückgegriffen werden [SimMetrics 2013]. Diese Bibliothek ermöglicht die Berechnung einer Vielzahl verschiedener Ähnlichkeitsmetriken wie die Levenshtein-Distanz[4] oder die Jaro-Winkler-Distanz[5]. Für jede der zuvor beschriebenen Ähnlichkeitsbestimmungen sollen Schwellwerte eingeführt werden. Dies soll verhindern, dass Relationen mit geringer Relevanz in den Thesaurus aufgenommen werden. Weiterhin sollen die Ähnlichkeitswerte mit in den Thesaurus überführt werden, um eine eventuelle Selektion von verwandten Thesaurus-

[4] Die Levenshtein-Distanz gibt die minimale Anzahl von Editieroperationen an, die notwendig sind um eine Zeichenkette in eine andere Zeichenkette zu transformieren [Dengel 2012, S. 144]

[5] Die Jaro-Winkler-Distanz basiert auf einer präfixbasierten Ähnlichkeitsbestimmung [SimMetrics 2013].

Konzepten auf Basis dieser Werte durchführen zu können. Der in Abbildung 7.4 dargestellte Ablauf verdeutlicht nochmals den Gesamtablauf der Relationsbestimmung. Zunächst wird auf der Basis von DISCO überprüft, ob eine Kollokation zwischen den Schlüsselwörtern A und B vorzufinden ist. Liegt der Wert dieser Überprüfung unter einem Schwellwert, so wird die Ähnlichkeit auf Basis eines Zeichenkettenvergleiches bestimmt. Wenn eines der beiden Verfahren ein Ergebnis größer der festgelegten Schwellwerte liefert, wird die Relation $A \sim B$ und der dazugehörige Ähnlichkeitswert im Triple Store des Thesaurus abgespeichert. Kollokationen werden hierbei bevorzugt behandelt, da die Qualität dieses statistischen Verfahrens höher einzustufen ist als der auf Distanz-Metriken beruhende Zeichenkettenvergleich.

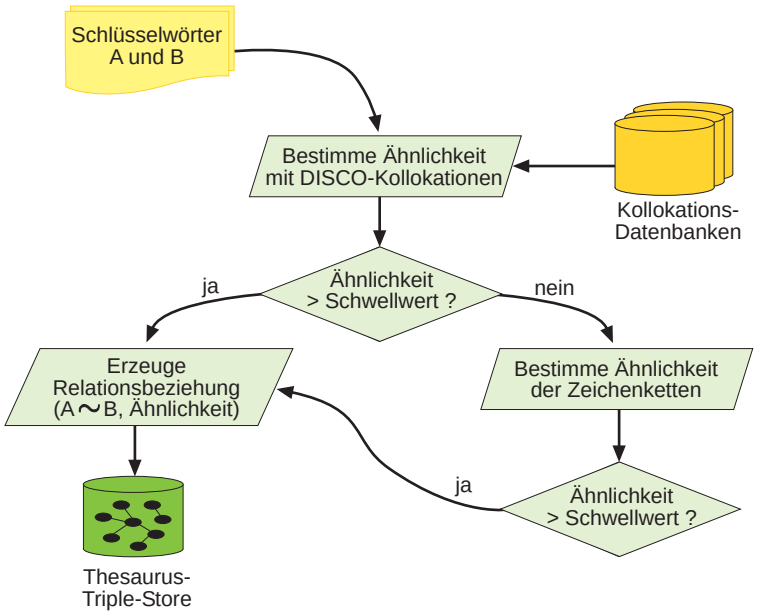


Abbildung 7.4.: Ablauf der Relationsbestimmung

Benutzerschnittstelle zur Bearbeitung des Thesaurus

Um den automatisiert erzeugten Thesaurus validieren und verbessern zu können, wird im Folgenden eine Benutzerschnittstelle zur Bearbeitung der Thesaurus-Inhalte konzipiert. Die einfachste Art der Bearbeitung dieser Inhalte bestünde in einem simplen Export der RDF-Daten. Die exportierten Daten könnten mit einem beliebigen Text-Editor bearbeitet und anschließenden wieder reimportiert werden. Da diese Art der Bearbeitung der Thesaurus-Inhalte nicht sonderlich komfortabel ist, wäre eine bedienungsfreundlichere Schnittstelle wünschenswert.

Da bereits eine Reihe von fertigen SKOS-Editoren existiert, ist die Umsetzung einer vollständig selbst entwickelten Lösung nicht zielführend. Hier soll die Open-Source-Software *SKOSjs* eingesetzt werden [SKOSjs 2013]. SKOSjs ist ein in JavaScript geschriebenes Bearbeitungs-Programm für SKOS-basierte Daten. Die Software ist kompatibel zum SPARQL 1.1 Protokoll[6]. Dieses Protokoll beschreibt eine Menge von Operationen, die zwischen einem SPARQL Client und Server zur Abfrage sowie Bearbeitung von RDF-Inhalten durchgeführt werden können. Abbildung 7.5 zeigt ein Beispiel der SKOSjs-Benutzerschnittstelle. Im linken Be-

[6] http://www.w3.org/TR/sparql11-protocol/

reich des Interface werden alle verfügbaren Konzepte des Thesaurus in einer Baumhierarchie dargestellt. Die Konzepte können einzeln ausgewählt und in der rechten Hälfte des Interface bearbeitet werden. Hier können Eigenschaften wie das präferierte Label (`<skos:prefLabel>`) und auch Beziehungen wie Ober- oder Unterkonzepte festgelegt werden.

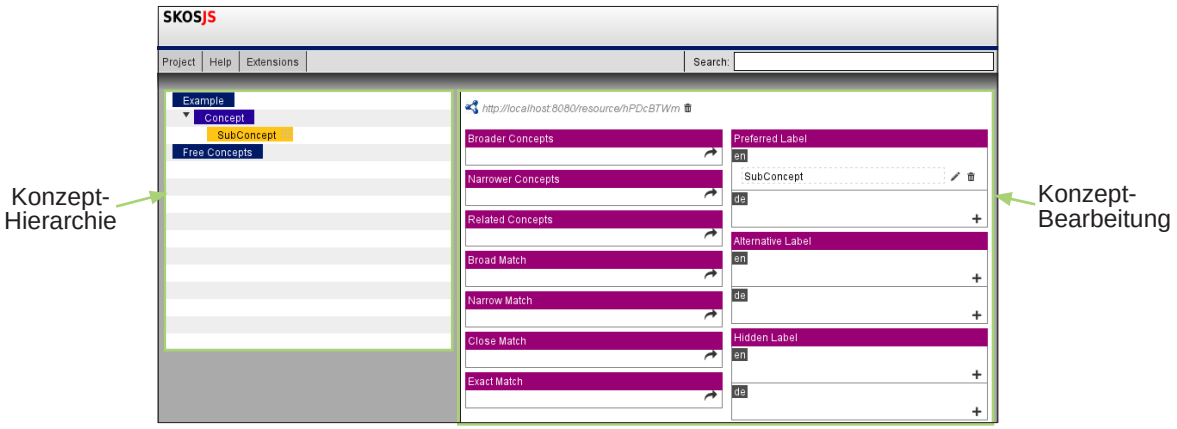


Abbildung 7.5.: SKOSjs-Benutzerschnittstelle

7.2.3. Schlüsselwortbasierte Suche nach Konzepten

Wie in Abschnitt 7.1 herausgestellt, soll das zu entwerfende semantische Suchsystem eine Schlüsselwortsuche ermöglichen. Der Thesaurus ist das zugrundeliegende semantische Modell, dass im Kontext der Suche zur Auflösung von Mehrdeutigkeiten sowie zur Abbildung von Beziehungen zwischen den DLR-Themengebieten eingesetzt werden soll. Diese Themen werden im Thesaurus durch Konzepte repräsentiert. Zu den vom Nutzer eingegebenen Suchbegriffen müssen also passende Konzepte innerhalb des Thesaurus bestimmt werden. Der Thesaurus ist also Wissensbasis und Suchraum zugleich. Die Suchfunktionalität ist für die Darstellung verwandter Themen mit Hilfe der Facetten (Szenario S-1) und für die Autovervollständigung (Szenario S-2) von großer Bedeutung. Um eine solche Suche zur ermöglichen, wird im Folgenden das Prinzip der schlüsselwortbasierten semantischen Suche angewendet. Bei diesem Prinzip werden mehrere Ansätze miteinander vereint. Zum einen der graphbasierte Ansatz der semantischen Technologien und zum anderen der Bag-of-Word-Ansatz des IR. Eine Kombination dieser Ansätze ist notwendig, da eine Schlüsselwortsuche, die mit Hilfe einer graphbasierten Abfragesprache durchgeführt wird, häufig zu schlechten Suchergebnissen führt (vgl. [Pérez-Agüera et al. 2010]). Aufgrund dieser Tatsache wird bei der hier beschriebenen Schlüsselwortsuche das Ranking-Modell des IR herangezogen. Diese Herangehensweise ermöglicht das effektive Durchsuchen und Finden von Thesaurus-Konzepten. Zunächst wird dazu das Prinzip der schlüsselwortbasierten semantischen Suche auf den hier betrachteten Anwendungsfall übertragen. Anschließend wird erläutert, wie diese Suchfunktionen innerhalb der Facetten und der Autovervollständigung genutzt werden sollen. Die hier verwendeten Ansätze der schlüsselwortbasierten semantischen Suche sind dabei an Wang et al. [2009], Cheng und Qu [2009] und Pérez-Agüera et al. [2010] angelehnt.

Neben dem zuvor erwähnten Triple Store (siehe Abschnitt 7.2.1) ist zunächst ein invertierter Index notwendig. Dieser Index bildet die Grundlage zur Anwendung der IR-Methoden. Die Thesaurus-Konzepte sollen dazu nicht nur im Triple Store abgelegt werden, sondern auch innerhalb dieses Indexes. Dadurch können die Konzepte mit Hilfe von IR-Methoden durchsucht

und eine Rangfolgenbildung durchgeführt werden. Hierzu müssen die als RDF-Tripel vorliegenden Konzepte für den Index entsprechend aufbereitet werden. An dieser Stelle ist eine Transformation aus der objektorientierten Sichtweise des Semantic Web in die dokumentenorientierte Sichtweise des IR notwendig. Bei schlüsselwortbasierten semantischen Suchsystemen wird dazu aus jedem Objekt ein entsprechendes virtuelles Dokument konstruiert [Cheng und Qu 2009]. Die virtuellen Dokumente beinhalten dabei die textuellen Informationen der Objekte. Üblicherweise besteht nach Cheng und Qu ein virtuelles Dokument aus folgenden Feldern:

- **Lokaler Anteil der Objekt-URI**: Der lokale Anteil des Identifier ist innerhalb eines Namensraumes eindeutig. In semantischen Suchsystemen dient er zur Identifizierung der Objekte im Index.
- **Literale des Objekts**: Für die Überführung in ein virtuelles Dokument werden die Literale verwendet, welche für den Menschen lesbare Namen oder Beschreibungen enthalten.

Auf Basis eines so konstruierten virtuellen Dokuments kann der invertierte Index erstellt werden und eine schlüsselwortbasierte Suche stattfinden. Für die Konzepte des Thesaurus müssen also Eigenschaften bestimmt werden, die innerhalb eines virtuellen Dokuments verwendet werden sollen. Mit Blick auf den SKOS-Standard werden dazu folgende Eigenschaften der Konzepte zur Überführung ausgewählt:

- **Objekt-URI**: Neben dem lokalen Anteil wird auch der Namensraum zur eindeutigen Identifizierung eines Konzepts im Index verwendet. Dies ermöglicht den eventuellen Import anderer Thesaurus-Inhalte, ohne dass dadurch Namensraum-Konflikte entstehen.
- `<skos:prefLabel>`: Das präferierte Label eines SKOS-Konzepts. Dieses RDF-Literal liefert die wichtigste textuelle Beschreibung für eine Schlüsselwortsuche.
- `<skos:altLabel>`: Mit Hilfe dieser Eigenschaft werden Synonyme der SKOS-Konzepte repräsentiert. Auch diese Eigenschaft stellt eine bedeutende textuelle Beschreibung im Rahmen der Suche nach Konzepten dar.

Die zuvor bestimmten Elemente repräsentieren also ein SKOS-Konzept innerhalb des invertierten Indexes. Der SKOS-Standard enthält daneben zwar noch eine Reihe weiterer Literale (z.B. `<skos:note>`). Diese sind für die hier beschriebene Suche nach Konzepten aber nicht relevant.

Das generelle Prinzip zum Auffinden von Konzepten ist wie folgt: Zunächst wird ein *syntaktisches Matching* auf Basis des invertierten Indexes durchgeführt. Ein syntaktisches Matching bezeichnet dabei im Rahmen semantischer Suchsysteme einen Suchvorgang, bei dem der textuelle Inhalt von RDF-Ressourcen mit dem der Nutzeranfrage verglichen wird [Hildebrand et al. 2007]. Die vom Nutzer eingegebenen Schlüsselwörter werden also an dieser Stelle mit den textuellen Eigenschaften der Konzepte im Index verglichen. Die so gefundenen Konzepte werden danach einem Ranking unterzogen. Für ein effektives Ranking-Verfahren ist es notwendig, dass für die Felder des virtuellen Dokuments ein Relevanzfaktor festgelegt wird. Dieser Faktor dient dazu, das Ranking der Suchergebnisse zu beeinflussen. Objekt-Eigenschaften mit wichtigeren textuellen Informationen erhalten so einen höheren Stellenwert innerhalb der Rangfolgenbildung. Im nächsten Schritt wird ein *semantisches Matching* durchgeführt. Von einem semantischen Matching wird gesprochen, wenn bei der Suche RDF-Graphstrukturen und die dahinter liegende Semantik mit berücksichtigt werden [Hildebrand et al. 2007]. Das

bedeutet, an dieser Stelle werden die semantischen Verbindungen der gefundenen Konzepte innerhalb des RDF-Graphen des Thesaurus ermittelt, ausgelesen und aufbereitet. Die konkreten Strategien für das zuvor beschriebene Prinzip des Auffindens von Konzepten sind abhängig vom jeweiligen Einsatzkontext. Eine Erläuterung dieser Suchstrategien erfolgt in den nachfolgenden Abschnitten.

Da innerhalb des KnowledgeFinder bereits Lucene als Suchtechnologie Verwendung findet, soll diese Software ebenfalls zur Implementierung der hier beschriebenen Suchstrategien genutzt werden.

Suchstrategien im Rahmen der facettierten Suche

Innerhalb der Facetten sollen Querbezüge zu der vom Nutzer durchgeführten Suche dargestellt werden (siehe Szenario S-1). Dazu müssen zunächst zum eingegebenen Suchbegriff passende Konzepte im Thesaurus bestimmt werden. Im ersten Schritt wird dazu versucht mit Hilfe der Rangfolgenbildung, Konzepte mit der höchsten Relevanz zu ermitteln. Hier findet zunächst ein exaktes syntaktisches Matching auf Basis der zuvor bestimmten virtuellen Dokumente statt. Exakt bedeutet in diesem Zusammenhang, dass der Suchbegriff und die Felder des virtuellen Dokuments genau übereinstimmen müssen. Liefert diese Suche kein Ergebnis, so wird im nächsten Schritt die von Lucene unterstützte unscharfe Suche durchgeführt. Wäre beispielsweise das Konzept „Assistenzsystem“ nicht vorhanden, so würde die unscharfe Suche beispielsweise das Konzept „Assistenzsicherheitssystem“ finden. Liefert auch die unscharfe Anfrage keine Treffer, so wird eine präfixbasierte Suche gestartet. Hier würde beispielsweise zum Begriff „Assistenzsystem“ auch das Konzept „Assistenzsystem-Richtlinien“ gefunden. Aus der Menge der gefundenen Konzepte wird anschließend das in der Rangfolge am höchsten bewertete Konzept ausgewählt. Für das so gefundene Konzept wird danach das semantische Matching ausgeführt. Die folgenden Verbindungen sollen dabei im Triple Store ermittelt werden:

- Synonyme (`<skos:altLabel>`)
- Oberkonzepte (`<skos:broader>`)
- Unterkonzepte (`<skos:narrower>`)
- Verwandte Konzepte (`<skos:related>`)

Sofern keine menschliche Validierung erfolgt ist, besteht der Thesaurus aus einer Ansammlung von Konzepten, die lediglich ein präferiertes Label und 0 bis n verbundene Elemente besitzen. Das heißt, auf Basis dieser Daten können Synonyme und Ober- sowie Unterkonzepte nicht ermittelt werden. Ziel ist es aber, auch mit diesem eingeschränkten Thesaurus semantische Hilfestellungen zu geben. Aufgrund der automatisierten Erstellung ist in diesem Fall die Qualität der verwandten Themengebiete zu untersuchen.

Suchstrategien im Rahmen der Autovervollständigung

Wie in Szenario S-2 beschrieben, soll die Autovervollständigungs-Funktion dem Nutzer während der Eingabe eines Suchbegriffs eine Liste von Vorschlägen unterbreiten. Diese Vorschläge sollen dabei nicht nur lexikalisch verwandte Begriffe enthalten, sondern auch Vorschläge, die mit diesem Begriff in Beziehung stehen. Auch die Autovervollständigung soll also die im Thesaurus abgebildeten Themengebiete und deren Beziehungen nutzen. Das heißt, hier wird

nicht der sonst übliche Ansatz dieser Vorschlagssysteme gewählt. Vorschlagssysteme, wie sie bei Suchmaschinen wie Google eingesetzt werden, basieren auf der Auswertung vom Nutzungsstatistiken[7] (vgl. [Bar-Yossef und Kraus 2011] und [Kastrinakis und Tzitzikas 2010]). Das bedeutet, dem Nutzer werden passend zu Eingabe die zuvor durchgeführten Suchanfragen anderer Nutzer vorgeschlagen. Dieser Ansatz wird nicht verfolgt, da zum einen für das Elib-Portal keine auswertbaren Statistiken vorliegen. Zum anderen steht diese Vorgehensweise im Widerspruch zu dem Ziel, Beziehungen zwischen Themen aufzeigen zu wollen. Da aber die Thesaurus-Inhalte aus den Elib-Daten extrahiert werden, wird davon ausgegangen, dass eine daraus generierte Vorschlagsliste ebenso hilfreich ist.

Im Gegensatz zu den zuvor untersuchten Facetten müssen im Fall der Autovervollständigung mehrere Konzepte ermittelt werden. Hierzu wird mit Hilfe des invertierten Indexes eine Menge von relevanten Konzepten per syntaktischen Matching gesucht. Hier wird eine Präfixsuche durchgeführt. Dadurch können alle Konzepte gefunden werden, welche die eingegebenen Zeichen als Präfix enthalten. Wie bereits erwähnt, werden danach mit Hilfe des semantischen Matching die Beziehungen aus dem Triple Store geladen. Die Sortierung innerhalb der Autovervollständigungsliste findet auf Basis der Relevanz der gefundenen Konzepte statt. Das zuvor beschriebene Suchverfahren muss dabei nach jedem eingegebenen Zeichen erfolgen. Dies ist notwendig, um eine ständige Aktualisierung der Ergebnisse passend zur Eingabe des Nutzers sicherzustellen.

7.2.4. Architekturüberblick und Integration in den KnowledgeFinder

Zur Umsetzung einer konzeptbasierten semantischen Suche wurde in den vorherigen Abschnitten mehrere Teilkonzepte erarbeitet. Zunächst fand die Festlegung eines geeigneten Formats zur Speicherung der Thesaurus-Inhalte statt. Anschließend stand die automatisierte Erstellung des Thesaurus im Mittelpunkt. Danach wurde diskutiert, wie eine schlüsselwortbasierte Suche nach Konzepten des Thesaurus durchgeführt werden kann. Im Folgenden wird nun erläutert, wie diese Ansätze und Verfahren in das Gesamtbild des KnowledgeFinder integriert werden sollen. Dazu wird ein Überblick über die resultierende Gesamtarchitektur gegeben und herausgestellt welche Erweiterungen und Komponenten zur Realisierung der genannten Teilkonzepte notwendig sind. Abbildung 7.6 stellt diese Gesamtarchitektur anschaulich dar. Diese Darstellung zeigt den Aufbau des KnowledgeFinder erweitert um die Komponenten, welche für die hier konzipierte semantische Suche notwendig sind. Die Bereiche, in denen Erweiterungen stattfinden, sind durch eine gestrichelte Linie hervorgehoben. Da die SVN-Datenanbindung für den Elib-Anwendungsfall keine Rolle spielt, ist dieser Systemteil nicht dargestellt (vgl. Abbildung 5.1).

Um die im Elib-System eingetragenen Schlüsselwörter abrufen zu können, ist zunächst die *Crawler-Komponente* anzupassen. Da die OAI-Schnittstelle des Elib-Systems die Schlüsselwörter nicht zur Verfügung stellt, ist eine alternative Lösung notwendig. Eine Anpassung auf Schnittstellenseite kann nicht durchgeführt werden, da auf die zugrundeliegende EPrints-Software nicht zugegriffen werden kann. Aufgrund der Tatsache, dass innerhalb der Elib-Benutzerschnittstelle diese Schlüsselwörter ausgegeben werden, besteht die Lösung darin, den Crawler um einen Mechanismus zum Abrufen und Parsen von HTML-Seiten zu ergänzen.

Die *Extractor-Komponente* hat die Schlüsselfunktion im Rahmen der semantischen Aufbe-

[7] Dieses Verfahren wurde erstmalig von Amazon eingesetzt und als Patent angemeldet.

reitung der Informationsquellen. Diese Komponente führt dabei mehrere Aufgaben durch. Zunächst sind die nutzervergebenen Schlüsselwörter für die Verwendung im Thesaurus aufzubereiten. Daneben ist der Extraktionsprozess zur automatisierten Gewinnung von Schlüsselwörtern sowie das Verfahren zur Relationsbestimmung zu integrieren. Hierzu soll wie folgt vorgegangen werden: In einem ersten Schritt werden die nutzervergebenen Schlüsselwörter innerhalb der Trainingsphase zur Erzeugung eines KEA-Modells benutzt. Dazu werden Elib-Einträge, bei denen Schlüsselwörter vorhanden sind, zufällig für das Training ausgewählt. Nachdem ein entsprechendes KEA-Modell erzeugt wurde, kann in einem nächsten Schritt die automatisierte Extraktion der Schlüsselwörter über den gesamten Elib-Datenbestand stattfinden. Die so gewonnenen Schlüsselwörter werden dann mit den nutzervergebenen Schlüsselwörtern zusammengeführt. Im letzten Schritt kann die Relationsbestimmung nach dem zuvor erarbeiteten Verfahren zwischen den n häufigsten Schlüsselwörtern stattfinden. Aufgrund der aufwändigen Relationsbestimmung und der resultierenden quadratischen Zeitkomplexität des Verfahrens ist zuvor ein entsprechender Wert für n festzulegen.

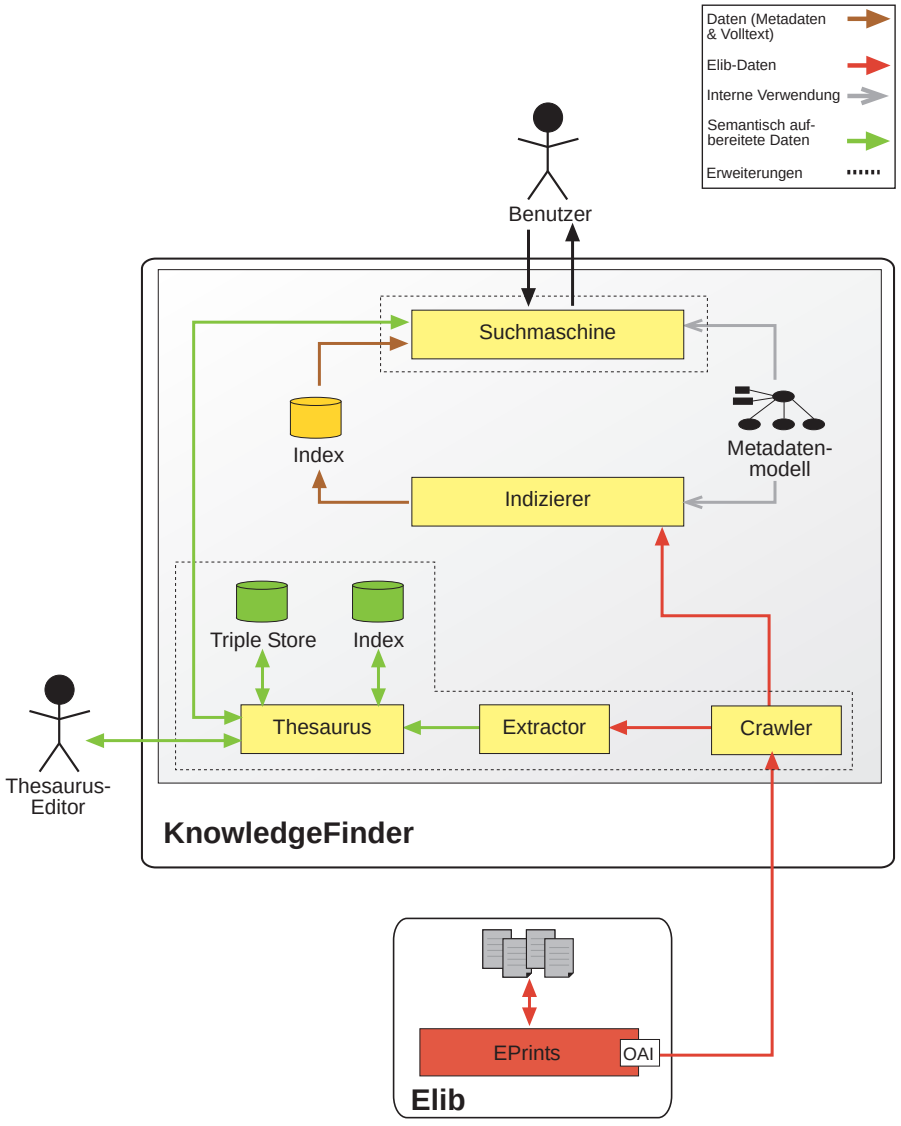


Abbildung 7.6.: Überblick über die resultierende Gesamtarchitektur des KnowledgeFinder

Die Schlüsselwörter und ihre Relationen werden anschließend innerhalb der *Thesaurus-Komponente* als SKOS-Konzepte persistiert. Dieser Systemteil bietet dazu Schnittstellen an und repräsentiert das Bindeglied zwischen den semantischen Daten und der Suchmaschinen-Komponente. Im Rahmen der Persistierung muss dafür gesorgt werden, dass Konzepte nicht doppelt eingetragen werden. Für diese Überprüfung sind das präferierte Label und die Synonyme eines Konzepts relevant. Ebenfalls ist es notwendig, neu eingetragene Konzepte zu kennzeichnen. Dadurch ist eine Trennung zwischen automatisiert hinzugefügten und bereits bearbeiteten Elementen möglich. Aufgrund ihrer Bindeglied-Funktion stellt die Thesaurus-Komponente neben den Import-Schnittstellen auch die semantischen Suchfunktionalitäten zur Verfügung. Das heißt, hier werden die zuvor erläuterten Verfahren zur Suche nach Konzepten integriert (siehe Abschnitt 7.2.3). Die Komponente verfügt dazu zum einen über den Triple Store zur Speicherung und Abfrage der SKOS-basierten Thesaurus-Inhalte. Zum anderen beinhaltet die Komponente den invertierten Index. In diesem Index sind die SKOS-Konzepte als virtuelle Dokumente abgebildet. Zur Durchführung der verschiedenen Suchstrategien (Facette bzw. Autovervollständigung) sind ebenfalls entsprechende Schnittstellen vorhanden. Für den Zweck der Validierung und Bearbeitung der Thesaurus-Inhalte durch einen Benutzer (*Thesaurus-Editor*) ist weiterhin ein User Interface vorgesehen. Hier ist die SKOSjs-Benutzerschnittstelle in Kombination mit einem SPARQL Endpoint zu integrieren.

7.2.5. Erweiterungen der Benutzerschnittstelle

In diesem Abschnitt werden zuletzt die Erweiterungen innerhalb der Benutzerschnittstelle des Elib-Portals konzipiert. Hier ist einerseits die in Szenario S-1 aufgeführte facettierte Darstellung eines Thesaurus-Konzepts zu betrachten und andererseits das in Szenario S-2 beschriebene GUI-Element der Autovervollständigung zu entwerfen. Zu diesem Zweck wurden grafische Entwürfe entwickelt, die im Folgenden vorgestellt werden. Die Entwicklung dieser Entwürfe fand auf Basis der Anforderungsszenarien statt.

Facettierte Darstellung eines Thesaurus-Konzepts

Der in Abbildung 7.7 dargestellte Entwurf veranschaulicht die facettierte Darstellung eines gefundenen Thesaurus-Konzepts. In dem dargestellten Fall hat der Nutzer beispielsweise nach „solar energy“ gesucht. Mit Hilfe der Thesaurus-Komponente konnte dazu ein passendes Konzept innerhalb des Thesaurus ermittelt werden. Der allgemeine Aufbau dieser Konzept-Facette entspricht den schon bekannten Facetten des Elib-Portals (siehe Abschnitt 5.3). Der Titel „Das könnte Sie auch interessieren“ soll auf die besondere Art der Hilfestellung dieser Schnittstelle hinweisen. Die durch das semantische Matching ermittelten Konzept-Eigenschaften werden jeweils in einer eigenen Gruppe angezeigt (siehe Abschnitt 7.2.3). Aufgrund der Tatsache, dass Konzepte die Themengebiete des DLR repräsentieren, ist die Beschriftung der Gruppen dementsprechend gewählt. Wie von den Facetten des Elib-Portals bekannt, wird immer nur eine gewisse Anzahl an Elementen innerhalb einer Gruppe dargestellt. Mit Hilfe von entsprechenden Schaltflächen kann eine vollständige Liste angezeigt werden. Innerhalb einer Gruppe sind die verknüpften Konzepte absteigend nach der Anzahl der im Elib-Index gefundenen Treffer sortiert. Bei automatisch ermittelten Beziehungen wäre an dieser Stelle auch eine andere Sortierung denkbar. So könnte innerhalb der verwandten Themen auch nach dem Ähnlichkeitswert sortiert werden (siehe Abschnitt 7.2.2). Im Rahmen der Evaluation ist zu untersuchen, inwiefern sich unterschiedliche Sortierungen auf die Ergebnisqualität auswirken. Hervorzuheben ist, dass bei der Auswahl eines Eintrages durch

den Nutzer keine Filterung der Ergebnisse stattfindet. Stattdessen wird eine neue Suche nach Elib-Dokumenten, die dieses Thema enthalten, gestartet. Dieses Verhalten steht im Gegensatz zu dem üblichen Verhalten der Facetten. Eine Filterung der Suchergebnisse würde jedoch zu einer weiteren Einschränkung des Suchraumes führen. Da aber die explorative Suche unterstützt werden soll, ist eine weitere Einschränkung des Suchraumes nicht erwünscht. Vielmehr soll es dem Nutzer ermöglicht werden, neue Themen und Querbezüge zu entdecken, die ihm vorher nicht bewusst oder bekannt waren.

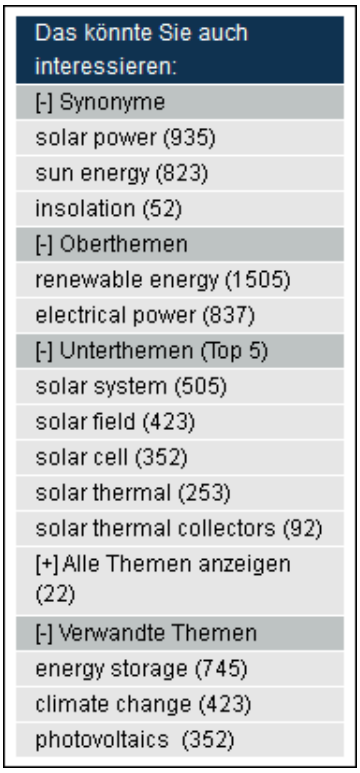


Abbildung 7.7.: Entwurf der facettierten Darstellung eines Thesaurus-Konzepts

Darstellung der Autovervollständigung

Die Darstellung der Autovervollständigung innerhalb der Benutzerschnittstelle ist in Abbildung 7.8 zu sehen. In dem hier aufgeführten Beispiel hat der Nutzer den Suchbegriff „Solar“ eingegeben. Aufgrund der Übersichtlichkeit ist die maximale Anzahl an dargestellten Konzepten auf 5 beschränkt. Die einzelnen Einträge der Liste sind optisch durch eine Trennlinie voneinander abgesetzt. Die Darstellung eines Konzepts erfolgt dabei in Anlehnung an Arias et al. [Arias et al. 2008]. Für jedes Konzept werden nicht nur das präferierte Label, sondern auch Oberbegriffe, Unterbegriffe und Assoziationen angezeigt. Lediglich Synonyme werden an dieser Stelle nicht aufgeführt, da sie für die Hilfestellungen der Autovervollständigungs-Funktion eine untergeordnete Rolle spielen. Anders als von Arias et al. beschrieben, ist zum besseren Verständnis der dargestellten Beziehungen jeweils ein Bezeichner vorangestellt. Die Bezeichner sind hierbei einheitlich zur Konzept-Facette gewählt (vgl. Abbildung 7.7). Sind die Einträge einer Eigenschaft zu lang, so wird am Ende der Zeile abgeschnitten. Wie abgebildet, wird ein Eintrag grau hinterlegt, wenn der Nutzer mit der Maus über diesen fährt. Klickt der Nutzer darauf, so übernimmt der KnowledgeFinder das präferierte Label des ausgewählten Konzepts als Suchbegriff und führt eine schlüsselwortbasierte Suche durch. Die

Benutzeroberfläche des Elib-Portals ist also wie beschrieben anzupassen und die entsprechende Schnittstelle der Thesaurus-Komponente zu nutzen. Wie zuvor erläutert, stellt diese Komponente ein Interface zur Durchführung der Autovervollständigungs-Suche bereit. Die Integration eines solchen Service innerhalb einer Benutzerschnittstelle wird üblicherweise mit Hilfe von AJAX auf der Basis einer Client-Server-Kommunikation realisiert. Im Kontext des KnowledgeFinder kann Vaadin als RIA-Framework zur Anbindung dieses Dienstes genutzt werden. Dieses Framework kapselt diese Art der Kommunikation bereits (siehe auch Abschnitt 5.4).

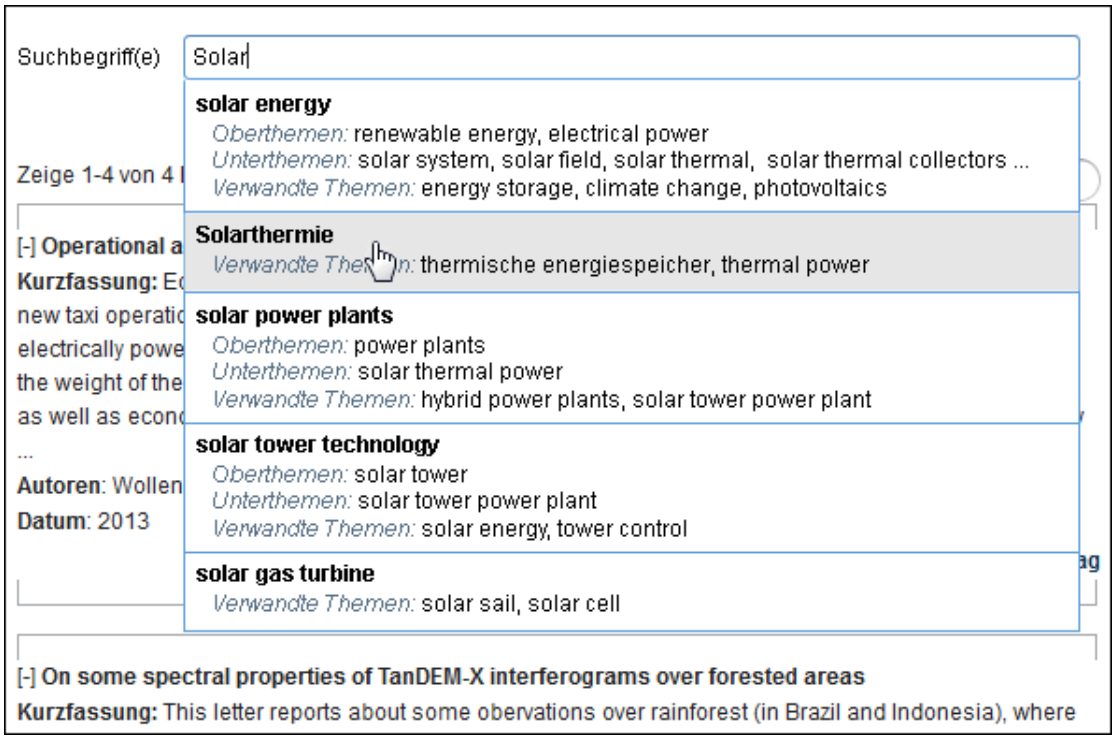


Abbildung 7.8.: Entwurf der Autovervollständigung

8. Prototypische Implementierung der semantischen Suche

In diesem Kapitel wird die prototypische Implementierung der konzeptbasierten semantischen Suche vorgestellt. Diese Implementierung dient anschließend als Basis für die Evaluation des vorgestellten Konzepts. Zunächst werden dazu die im vorherigen Kapitel erläuterten Komponenten und deren Schnittstellen aus softwaretechnischer Sicht beschrieben. Der Aufbau und die Funktionsweise der einzelnen Subsysteme wird danach detailliert dargestellt. Zuletzt werden die notwendigen Änderungen zur Integration der semantischen Suche in den eigentlichen Such- und Darstellungsprozess der Suchmaschinen-Komponente erörtert. Da es sich um eine prototypische Implementierung handelt, wurden lediglich Grundfunktionalitäten umgesetzt. Welche Funktionalitäten nicht oder nur teilweise implementiert wurden, wird in den jeweiligen Unterkapiteln aufgeführt.

8.1. Systemüberblick

Dieser Abschnitt soll einen Systemüberblick über die Komponenten zur Realisierung der semantischen Suche und deren Schnittstellen zu den vorhandenen KnowledgeFinder-Komponenten geben. Wie innerhalb der Konzeption aufgezeigt, sind zur Umsetzung der semantischen Suche die Extractor- sowie die Thesaurus-Komponente als zentrale Bausteine vorgesehen. Abbildung 8.1 zeigt die Integration dieser beiden Softwarebausteine in den Kontext des KnowledgeFinder mit Hilfe eines UML-Komponentendiagramms (Unified Modeling Language). Die hier aufgeführte Blackbox-Darstellung soll zunächst einen Systemüberblick vermitteln. Ziel ist es, die Schnittstellen sowie das Zusammenspiel zwischen den relevanten Komponenten zu veranschaulichen. Die neu erstellten Komponenten sind zur besseren Unterscheidung durch einen lilafarbigen Hintergrund von den ursprünglichen KnowledgeFinder-Komponenten abgehoben. Anzumerken ist, dass dieses Diagramm aus Gründen der Übersichtlichkeit nur die Komponenten darstellt, die im Rahmen der semantischen Suche eine Rolle spielen.

Die Crawler-Komponente wird innerhalb der Abbildung durch den Softwarebaustein `Crawler` repräsentiert. Die von dieser Komponente abgerufenen Elib-Dokumente werden von der Extractor-Komponente (`Extractor`) weiterverarbeitet. Dieses Element hat die Aufgabe, aus den Elib-Dokumenten Konzepte und Relationen zu extrahieren und diese im Thesaurus abzulegen. Dazu benötigt dieses Software-Element eine Schnittstelle vom Typ `Thesaurus`. Diese zentrale Schnittstelle stellt die Thesaurus-Komponente (`Thesaurus`) zur Verfügung. Das Interface beinhaltet Operationen zum Speichern und Suchen von Thesaurus-Inhalten. Von der Suchmaschinen-Komponente (`Search`) wird diese Schnittstelle zur semantischen Suche von Konzepten herangezogen. Ein detaillierter Aufbau sowie die Funktionsweise der zuvor genannten Komponenten wird in den nachfolgenden Abschnitten gegeben.

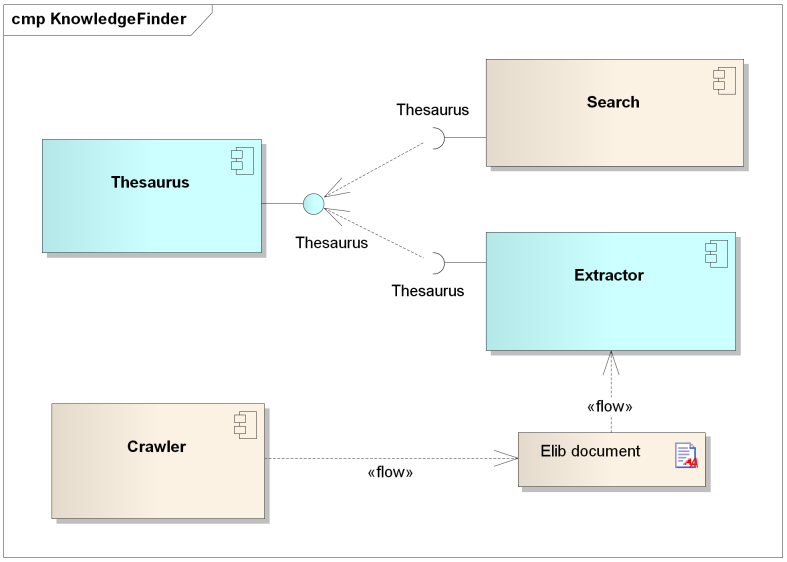


Abbildung 8.1.: KnowledgeFinder-Komponentendiagramm (Ausschnitt)

8.2. Die Crawler-Komponente

Wie zuvor in Abschnitt 7.2.2 herausgestellt, ist im Rahmen der Extraktion von Konzepten zunächst die `Crawler`-Komponente anzupassen. Diese Anpassung ist eine Grundvoraussetzung für die teilautomatisierte Erstellung des Thesaurus. Das Ziel besteht darin, die nutzervergebenen Schlüsselwörter der Elib-Einträge für die Erstellung des Thesaurus zu verwenden. Ein direktes Abrufen dieser Schlüsselwörter über die OAI-Schnittstelle ist nicht möglich. Die Umsetzung des identifizierten Lösungsweges wird im Folgenden kurz dargestellt werden.

Die Grundidee der hier durchgeführten Anpassungen der `Crawler`-Komponente besteht darin, die HTML-Seiten der Elib-Einträge abzurufen und entsprechend zu verarbeiten. Zu diesem Zweck wird die Python-Bibliothek *LXML*[1] genutzt, die das Verarbeiten von XML- und HTML-Dokumenten ermöglicht. Das Parsen der Schlüsselwörter erfolgt auf Basis der W3C-Empfehlung *XPath*[2]. Diese Sprache ermöglicht das direkte Adressieren von Knoten innerhalb eines XML-Dokuments. Die Knoten werden dabei mit Hilfe eines sogenannten XPath-Ausdrucks referenziert. Die `Crawler`-Komponente wurde in der Art und Weise angepasst, dass eine beliebige Anzahl von XPath-Ausdrücken zur Extraktion von Elementen angegeben werden kann. Der folgende Ausdruck dient zur Adressierung und Extraktion der nutzervergebenen Schlüsselwörter eines Elib-Eintrages:

```
//th[contains(.,'Stichw')]/parent::tr/child::td/text()
```

Mit diesem XPath-Ausdruck wird zunächst der HTML-Tabellen-Kopf innerhalb einer Elib-Webseite angesprochen, der die Zeichenkette „Stichw“ enthält[3]. Danach wird entlang des XML-Baumes die dazugehörige Tabellen-Zelle ermittelt und der textuelle Inhalt des Knotens ausgelesen. Die so gewonnenen Schlüsselwörter werden als zusätzliche Attribute an den Metadatensatz eines Elib-Eintrags angehangen und innerhalb des KnowledgeFinder gespeichert.

[1] http://lxml.de
[2] http://www.w3.org/TR/xpath
[3] Aufgrund von Verarbeitungsproblemen der LXML-Bibliothek im Zusammenhang mit Sonderzeichen, wird ein eindeutiges Präfix zu Adressierung eingesetzt.

Listing 8.1 zeigt einen beispielhaften Ausschnitt eines solchen Datensatzes[4].

```
1 <additionalMetadata>
2     <keywords>Solar concentrating system, central receiver, shielding design,
          ray tracing, flux mapping, optical errors</keywords>
3 </additionalMetadata>
```

Listing 8.1: Ausschnitt eines Elib-Metadatensatzes mit nutzervergebenen Schlüsselwörtern

8.3. Die Thesaurus-Komponente

Die `Thesaurus`-Komponente nimmt die zentrale Rolle im Rahmen der durchgeführten Erweiterungen ein. Sie stellt die semantischen Suchfunktionalitäten bereit und ist das Bindeglied zwischen den semantisch aufbereiteten Informationsquellen und der `Search`-Komponente. Aufgrund dieser zentralen Funktion wird die Realisierung dieser Komponente als nächstes beleuchtet. Zuerst findet dazu eine Erörterung der Umsetzung des Triple Store statt. Anschließend wird in Abschnitt 8.3.2 die interne Struktur sowie die Funktionsweise der Komponente dargestellt.

8.3.1. Realisierung des Triple Store

Der Triple Store dient der `Thesaurus`-Komponente zur Verwaltung der Thesaurus-Daten. Innerhalb der hier vorgestellten prototypischen Implementierung wird das *Sesame Framework* zur Verarbeitung und Speicherung der SKOS-Konzepte verwendet [Sesame 2012]. Dieses Framework ermöglicht das Verarbeiten, Speichern, Schlussfolgern und Abfragen von RDF-Daten. Außerdem werden mehrere APIs zur Anbindung von Triple Stores bereitgestellt. Die `Thesaurus`-Komponente nutzt verschiedene Technologien des Framework, die im weiteren Verlauf beschrieben werden.

Auf unterster Ebene einer Sesame-Anwendung ist das *RDF-Modell* angesiedelt, welches Basisschnittstellen für den Zugriff auf RDF-Entitäten bereitstellt. Darüber befindet sich die *Storage And Inference Layer* (Sail) API. Hier werden grundlegende Funktionalitäten für RDF Stores und Inferenz-Systeme bereitgestellt. Sesame bietet eine Reihe von unterschiedlichen Store-Varianten an. Oberhalb der Sail-Ebene befindet sich die *Repository API*. Diese API stellt High-Level-Schnittstellen für den Zugriff auf RDF-Daten bereit. Auch hier sind verschiedene Implementierungen verfügbar. So wird beispielsweise zwischen lokalen und entfernten Repositories unterschieden [Sesame 2012]. Im Rahmen des KnowledgeFinder wird ein nativer Store genutzt (*NativeStore*). Der native Speicher ist ein einfacher dateibasierter Triple Store. Diese Variante hat den Vorteil, den Entwicklungs- und Installationsaufwand des KnowledgeFinder möglichst gering zu halten. Bei der Verwendung eines speicherbasierten Store müsste beispielsweise sichergestellt werden, dass in Produktivumgebungen genügend Hauptspeicher zur Verfügung steht. Weiterhin erfolgt ein lokaler Zugriff mit Hilfe eines *SailRepository*. Die Verwendung eines entfernten *HTTP Repository* würde ebenfalls zu zusätzlichen Abhängigkeiten im Rahmen der Entwicklung und Installation zur Konsequenz haben.

Neben den zuvor aufgeführten Elementen wird das *Sesame Elmo Persistence Framework* eingesetzt [Elmo 2008]. Dieses Framework realisiert ein Objekt-Relationales-Mapping (ORM)

[4] Dieses Beispiel bezieht sich auf den bereits zuvor verwendeten Elib-Eintrag aus Abbildung 7.2.

zwischen Java-Objekten und RDF-Ressourcen. Dieses Prinzip erlaubt den einfachen objektorientierten Zugriff auf RDF-Ressourcen des Triple Store. Neben der Möglichkeit eigene sogenannte *JavaBeans* zu erzeugen, stellt das Persistenz-Framework diverse vorgefertigte Mappings bereit. Unter anderem wird ein Interface für das Mapping von SKOS-Konzepten zur Verfügung gestellt. Listing 8.2 zeigt einen kurzen Auszug dieser Schnittstellen-Definition. Wie zu sehen ist, findet die Zuordnung zwischen RDF-Ressourcen und Objekten mit Hilfe von URI-Annotationen statt (Zeile 3, 7 und 8). Diese Schnittstellen-Definition wird von der `Thesaurus`-Komponente zum Mapping von SKOS-Konzepten verwendet.

```
1  package org.openrdf.concepts.skos.core;
2
3  @rdf("http://www.w3.org/2004/02/skos/core#Concept")
4  public interface Concept extends Resource {
5
6      /** Broader concepts are typically rendered as parents in a concept
           hierarchy (tree). */
7      @rdf({"http://www.w3.org/2004/02/skos/core#broader", "http://www.w3.org
           /2004/02/skos/core#semanticRelation"})
8      @inverseOf({"http://www.w3.org/2004/02/skos/core#narrower"})
9      public abstract Set<Concept> getSkosBroaders();
10
11     ...
12 }
```

Listing 8.2: Quellcode-Auszug der SKOS-`Concept`-Schnittstellen-Definition [Elmo 2008]

Insgesamt ergibt sich somit der in Abbildung 8.2 schematisch dargestellte Aufbau des Thesaurus-Triple-Store. Das Elmo Framework als ORM-Mapper abstrahiert den Zugriff durch die `Thesaurus`-Komponente. Das Speichern und Auslesen der SKOS-Konzepte kann dadurch objektorientiert erfolgen. Darunter befindet sich ein SailRepository in Kombination mit dem nativen Store und dem Sesame-RDF-Modell.

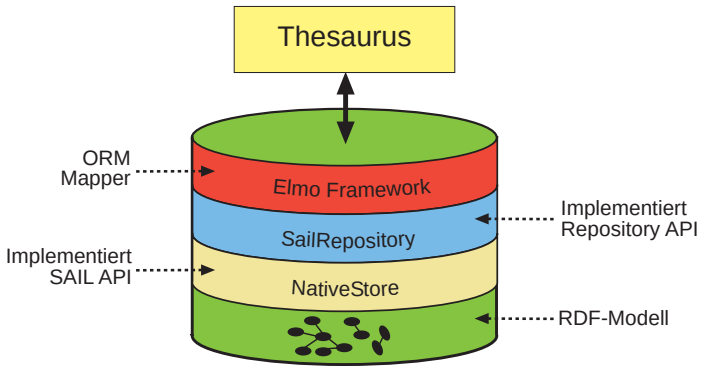


Abbildung 8.2.: Aufbau des Thesaurus-Triple-Store

8.3.2. Aufbau & Funktionsweise

In diesem Abschnitt soll der Aufbau und die Funktionsweise der `Thesaurus`-Komponente erläutert werden. Dazu wird zunächst die interne Struktur der Komponenten aufgezeigt. Danach wird der statische Aufbau mit Hilfe eines Klassendiagrammes dargestellt und die Funktionsweise der wichtigsten Klassen aufgezeigt.

Abbildung 8.3 zeigt eine Whitebox-Darstellung der Bestandteile der `Thesaurus`-Komponente. Die nach außen hin zur Verfügung gestellte `Thesaurus`-Schnittstelle wird durch die Subkomponente `Thesaurus-Core` realisiert. Diese Subkomponente ist für die Verwaltung und Suche von Thesaurus-Inhalten verantwortlich. Daneben findet hier auch die Anbindung des Triple Store und des invertierten Indexes statt. Der `Thesaurus-Core` ist das einzige Subelement der `Thesaurus`-Komponente. Wie in Abschnitt 7.2.4 beschrieben, waren hier ursprünglich weitere Subkomponenten vorgesehen. Die konzipierte Anbindung des SKOSjs-Editors wurde jedoch nicht umgesetzt. Wider Erwarten konnte kein SPARQL Endpoint zur Anbindung der SKOSjs-Benutzerschnittstelle integriert werden. Die Bearbeitung der Thesaurus-Inhalte in der hier vorgestellten Implementierung ist lediglich über einen Import-Export-Mechanismus möglich. Aufgrund dieses Mechanismus ist die Kennzeichnung von neu extrahierten Konzepten ebenfalls nicht realisiert worden.

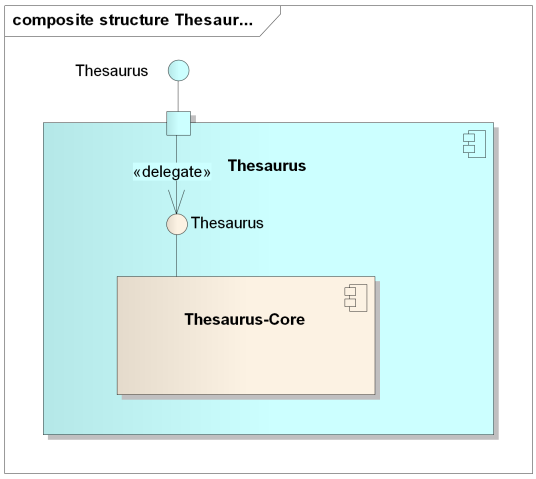


Abbildung 8.3.: Kompositionsstrukturdiagramm der `Thesaurus`-Komponente

Abbildung 8.4 zeigt die statische Struktur der `Thesaurus`-Komponente mit Hilfe eines Klassendiagrammes. Diese Darstellung beschränkt sich dabei auf die wesentlichen Elemente. Funktionsparameter, Rückgabetypen sowie Setter- und Getter-Methoden sind aufgrund der Übersichtlichkeit nicht aufgeführt. Die zuvor dargestellte `Thesaurus-Core`-Subkomponente setzt sich aus den Unterpaketen `elib.thesaurus.core.api` und `elib.thesaurus.core.impl` zusammen. Daneben ist das Paket `org.openrdf` mit mehreren Unterpaketen abgebildet, welche das zuvor beschriebenen Sesame Framework repräsentieren. Aufgrund der zentralen Funktion der `Thesaurus`-Komponente, soll der Sinn und Zweck der wichtigsten Klassen im Folgenden erörtert werden.

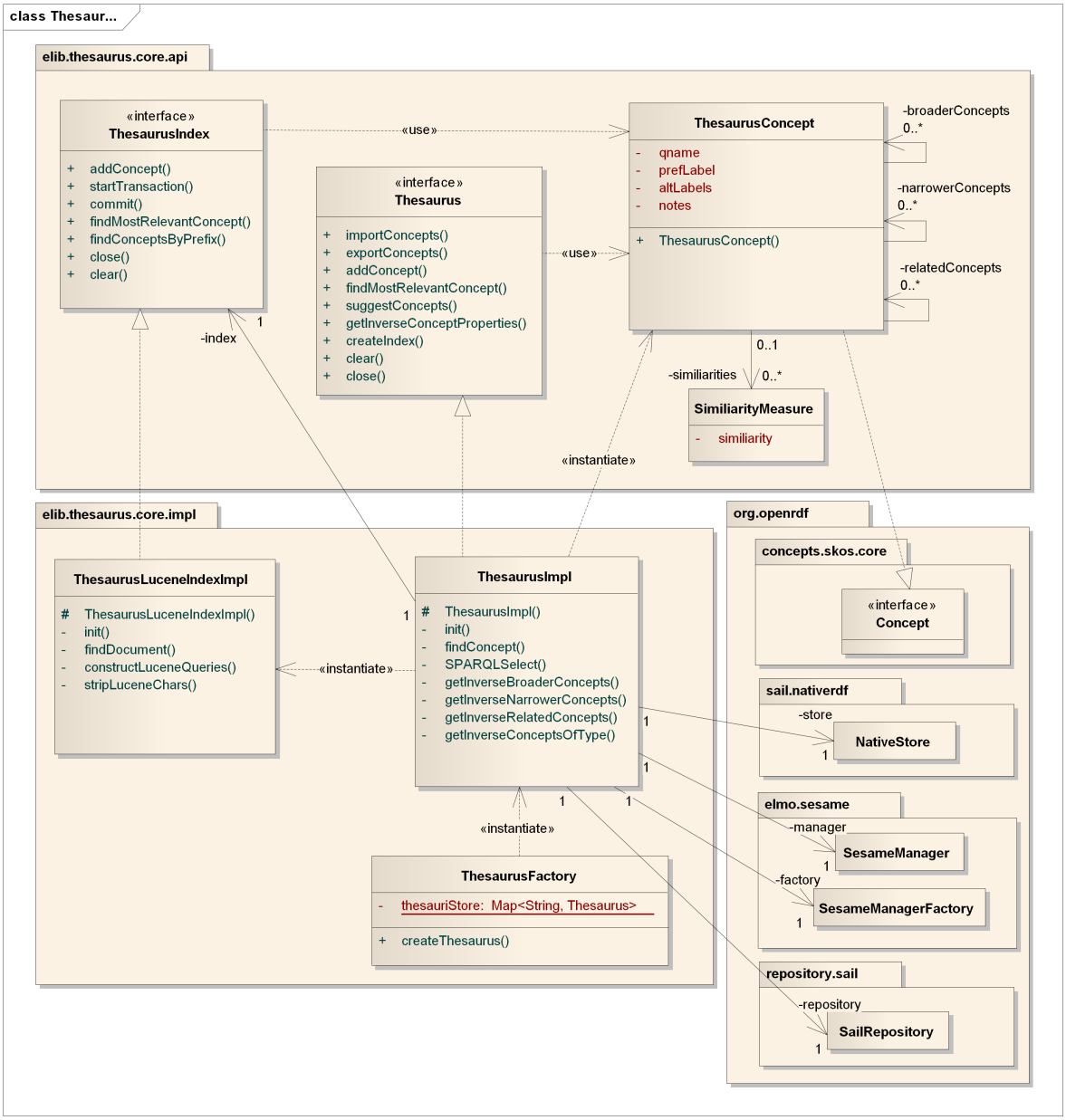


Abbildung 8.4.: Klassendiagramm der `Thesaurus`-Komponente

Zunächst werden die Klassen des Package `elib.thesaurus.core.api` betrachtet. Die Rollen der hier dargestellten Klassen-Definitionen sind folgendermaßen:

- `ThesaurusConcept`: repräsentiert ein Konzept des Thesaurus. Dazu implementiert diese Klasse die zuvor beschriebene `Concept`-Schnittstelle des Elmo Framework (siehe Listing 8.2). Dadurch können die Konzepte des Thesausus mit einfachen Mitteln ausgelesen, geändert und persistiert werden. Wie abgebildet, realisiert die `ThesaurusConcept`-Klasse nur grundlegende Eigenschaften dieser Schnittstellen-Definition. Der qualifizierte Name (`qname`) repräsentiert den vollständigen URI des Konzepts. Die Attribute `prefLabel` und `altLabels` entsprechen den jeweiligen Eigenschaften im SKOS-Standard. Daneben sind Attribute für die verschiedenen hierarchischen und nicht-hierarchischen Beziehungen vorgesehen. Ein `ThesaurusConcept`-Objekt kann Oberkonzepte, Unterkonzepte und verwandte

Konzepte besitzen. Die Variable `similiarities` repräsentiert die Ähnlichkeitswerte verwandter Konzepte, welche im Rahmen der Relationsbestimmung ermittelt werden. Diese Werte werden durch die Klasse `SimiliarityMeasure` gekapselt. Um die Werte im Triple Store speichern zu können, werden sie in Form von SKOS-Notizen abgelegt.

- `Thesaurus`: beinhaltet die Schnittstellen-Definitionen der nach außen hin zur Verfügung gestellten `Thesaurus`-Schnittstelle. Die hervorzuhebenden Methoden sind wie folgt:
  - `addConcept()`: dient zum Abspeichern von `ThesaurusConcept`-Instanzen im Triple Store. Die Persistierung erfolgt mit Hilfe des Elmo-ORM-Mechanismus.
  - `findMostRelevantConcept()`: führt die semantische Suche nach einem zum Suchbegriff passenden Konzept durch. Das Auffinden wird nach dem Prinzip des syntaktischen und semantischen Matching durchgeführt (siehe Abschnitt 7.2.3). Der qualifizierte Name dient bei diesem Vorgang als Primärschlüssel zur Identifizierung von Konzepten im Index sowie im Triple Store.
  - `suggestConcept()`: ermittelt im Rahmen der Autovervollständigung zu einer beliebigen Zeichenkette eine definierte Anzahl von Konzepten. Diese semantische Suche wird ebenfalls nach dem Prinzip des syntaktischen und semantischen Matching vollzogen. Die Konzepte werden im Rahmen des syntaktischen Matching mit Hilfe der präfixbasierten Suche ermittelt. Hier findet auch das IR-Ranking-Verfahren statt. Synonyme eines Konzepts haben dabei eine geringere Relevanz als das präferierte Label. Das semantische Matching ermittelt anschließend die Beziehungen der gefundenen Konzepte im Triple Store.
  - `createIndex()`: überführt die Triple-Store-Inhalte in den Thesaurus-Index.
- `ThesaurusIndex`: definiert Schnittstellen für den Zugriff auf den Thesaurus-Index. Die Verwendung einer Schnittstelle dient hier der Abstraktion von einer konkreten Implementierung. Dies ermöglicht neben Lucene auch die Verwendung anderer IR-Anwendungen. Dieses Interface ist nach außen hin nicht nutzbar, da Objekte der implementierenden Klasse nur innerhalb des Pakets erzeugt werden können. Die wichtigsten Methoden zur Verwaltung des Thesaurus-Indexes sind wie folgt:
  - `addConcept()`: überführt ein `ThesaurusConcept`-Objekt in den Index. Diese Überführung findet nach dem in Abschnitt 7.2.3 konzipierten Mechanismus statt. Listing 8.3 zeigt beispielhaft den Quellcode zur Überführung des präferierten Labels. Zunächst findet die Erzeugung eines Lucene-Feldes für diese Konzept-Eigenschaft statt (Zeile 1). Danach wird in Zeile 3 durch Einstellen eines Boost-Faktors die Relevanz des Labels für das IR-Ranking-Verfahren erhöht. Zuletzt wird das zuvor erzeugte Feld dem virtuellen Dokument hinzugefügt (Zeile 4). Dieses Vorgehen findet in ähnlicher Weise für alle in den Index zu überführenden Konzept-Eigenschaften statt.

```
1 Field prefLabelField = new Field(PREFLABEL_FIELD, concept.getPrefLabel().
      toLowerCase(), Field.Store.YES, Field.Index.ANALYZED);
2 // Adjust boost for higher ranking priority
3 prefLabelField.setBoost(2f);
4 document.add(prefLabelField);
```

Listing 8.3: Quellcode-Auszug der `addConcept()`-Methode

  - `startTransaction()` & `commit()`: dienen zur Definition von Transaktionsgrenzen im Rahmen der vollständigen Indizierung der Triple-Store-Inhalte. Dieses Vorgehen stellt sicher, dass das Indizieren innerhalb einer Transaktion erfolgt.

  - `findMostRelevantConcept()`: ist das Pendant zur gleichnamigen Funktion der `Thesaurus`-Schnittstelle, welches die syntaktische Suche im Index realisiert. Zur Suche nach Konzepten für die facettierte Darstellung wird diese Methode benötigt.
  - `findConceptsByPrefix()`: führt die syntaktische präfixbasierte Suche innerhalb des Indexes durch. Diese Funktion wird zur Realisierung der Autovervollständigung benötigt.

Die Klassen des Pakets `elib.thesaurus.core.impl` realisieren die zuvor erörterten Schnittstellen-Definitionen:

- `ThesaurusImpl`: implementiert sämtliche Funktionalitäten der `Thesaurus`-Schnittstelle. Dazu verfügt die `ThesaurusImpl`-Klasse über Attribute zur Anbindung des Sesame Triple Store sowie des Thesaurus-Indexes. Der `NativeStore` und das `SailRepository` repräsentieren die API-Implementierungen des Triple Store. Das ORM Mapping wird durch die `SesameManager`- und `SesameManagerFactory`-Attribute realisiert.

  Neben der Realisierung der `Thesaurus`-Schnittstellen beinhaltet die `ThesaurusImpl`-Klasse weitere Methoden. Herauszustellen sind unter anderem folgende private Funktionen:

  - `findConcept()`: realisiert das semantische Matching zur Ermittlung eines Konzepts und dessen Beziehungen im Triple Store. Dazu wird dem `SesameManager` der qualifizierte Name des zuvor per syntaktischen Matching gefundenen Konzepts übergeben.
  - `SPARQLSelect()`: ermöglicht die Abfrage des Triple Store mit Hilfe von SPARQL. An dieser Stelle findet der Zugriff auf den RDF-Datenspeicher nicht mittels des ORM-Mapper statt, sondern direkt über das `SailRepository`. Dieser Workaround ist notwendig, da das Persistenz-Framework RDF-Prädikate mit inverser Eigenschaft nicht korrekt auflöst. Daher werden inverse Beziehungen mit Hilfe einer graphbasierten Abfrage ermittelt. Die `SPARQLSelect()`-Methode wird beispielsweise von der `getInverseNarrowerConcepts()`-Methoden dazu genutzt, inverse Konzeptbeziehungen zu finden. Zur Ermittlung der vollständigen Oberkonzeptmenge werden beispielsweise alle RDF-Ressourcen gesucht, welche als Oberkonzept mit `<skos:narrower>` auf ein Konzept verweisen. Die Prädikate `<skos:broader>` und `<skos:narrower>` sind invers zueinander. Das in Listing 8.4 aufgeführte Beispiel veranschaulicht die dazu abgesetzte SPARQL-Anfrage.

```
1 PREFIX skos: <http://www.w3.org/2004/02/skos/core>
2 SELECT ?subject
3 WHERE {
4     ?subject <skos:narrower> ?object .
5     FILTER (?object = <http://elib.dlr.de/concept/#c_8189>)
6 }
```

Listing 8.4: SPARQL-Anfrage zur Ermittlung von Oberkonzept-Beziehungen

  Diese Anfrage sucht mit Hilfe eines Graph-Musters im `WHERE`-Anteil nach allen Subjekten, die mit dem Prädikat <skos:narrower> auf ein Objekt verweisen (Zeile 4). Der Filter in Zeile 5 dient zur Selektion des betrachteten Konzepts. Die gefundenen Subjekte werden zuletzt mit Hilfe von `SELECT` ausgewählt (Zeile 2).

- `ThesaurusLuceneIndexImpl`: realisiert einen Index auf der Basis von Lucene. Hier werden die zuvor beschriebenen Funktionalitäten der `ThesaurusIndex`-Schnittstelle implementiert.

- `ThesaurusFactory`: stellt als Fabrik-Klasse sicher, dass zu einer physischen Datenspeichergruppe (Triple Store und Index) nur jeweils ein `ThesaurusImpl`-Objekt existiert.

8.4. Die Extractor-Komponente

In diesem Abschnitt wird der Aufbau und die Funktionsweise der `Extractor`-Komponente, welche die Schlüsselrolle bei der semantischen Vorverarbeitung der Informationsquellen einnimmt, genauer betrachtet. Abbildung 8.5 zeigt die interne Struktur dieses Software-Elements mit Hilfe eines Kompositionsstrukturdiagramms. Die eigentliche Kernfunktionalität wird durch die Subkomponente `Extractor-Core` realisiert. Diese Komponente benötigt das zuvor beschriebene `Thesaurus`-Interface. Daneben beinhaltet die `Extractor`-Komponente drei weitere Subsysteme. Diese stellen dem `Extractor-Core` wichtige Funktionalitäten bereit. Die `Kea`-Subkomponente beinhaltet das KEA-Framework zur Schlüsselwortextraktion. Die `Disco`- und `SimMetrics`-Subsysteme werden im Rahmen der Relationsbestimmung herangezogen.

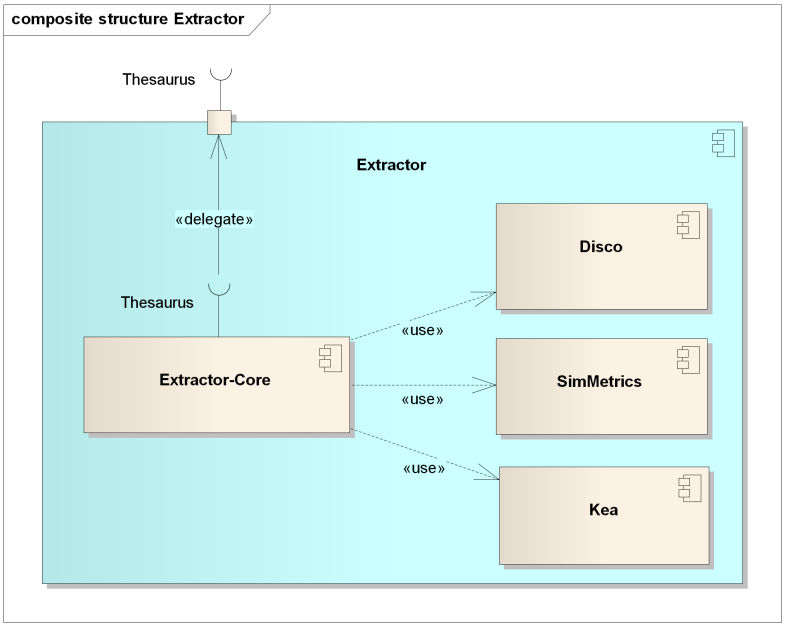


Abbildung 8.5.: Kompositionsstrukturdiagramm der `Extractor`-Komponente

Abbildung 8.6 zeigt die statische Struktur der `Extractor`-Komponente. Die zuvor beschriebene `Extractor-Core`-Subkomponente wird durch das Paket `elib.thesaurus.extractor.core` realisiert. Daneben ist das Paket `elib.thesaurus.extractor.helper` vorzufinden, das verschiedene Hilfsklassen beinhaltet. Unter anderem sind hier mehrere Hilfsklassen zur Dateiverarbeitung angesiedelt. Das Subpackage `model` enthält Klassen zur Abbildung von Schlüsselwörtern und Elib-Einträgen auf Java-Objekte. Die `Disco`- und `SimMetrics`-Subsysteme werden jeweils von den gleichlautenden Paketen repräsentiert. Im Rahmen der Schlüsselwortextraktion sowie der Relationsbestimmung bewerkstelligen die aufgeführten Klassen folgende Teilaufgaben:

- `ModelTrainer`: führt die KEA-Trainingsphase durch. Dazu wählt diese Klasse zunächst zufällig eine Anzahl von Elib-Einträgen mit nutzervergebenen Schlüsselwörtern aus. Jeder Eintrag wird dabei mit Hilfe der `ElibItem`-Klasse repräsentiert. Bei der Erzeugung von Elib-Objekten findet die Spracherkennung auf der Basis der N-Gramm-Statistik statt. Mit Hilfe der zufällig ausgewählten Schlüsselwörter wird anschließend eine Trainingsphase zur Generierung der sprachabhängigen KEA-Modelle gestartet.

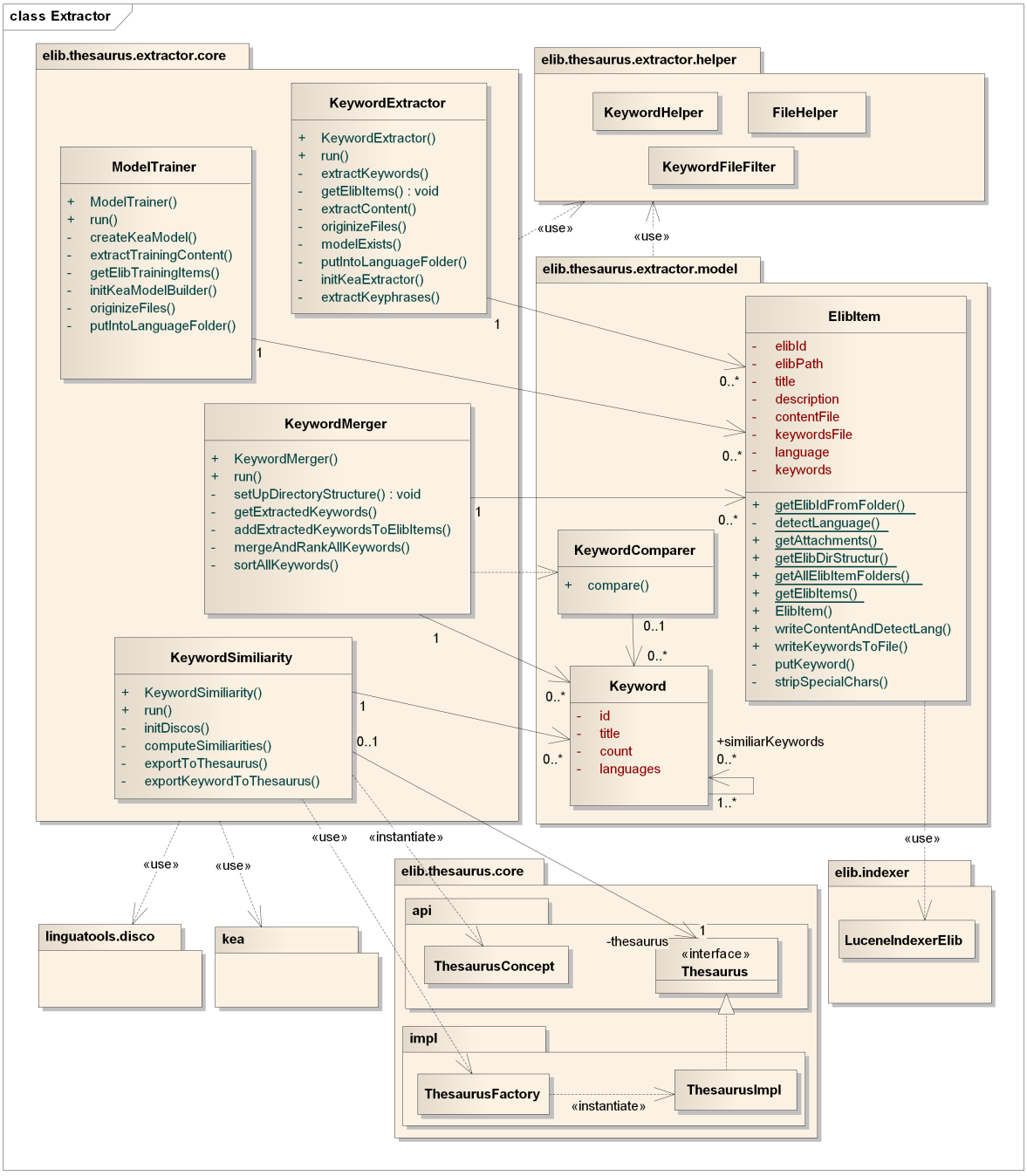


Abbildung 8.6.: Klassendiagramm der `Extractor`-Komponente

- `KeywordExtractor`: realisiert die Extraktionsphase des KEA-Algorithmus. Auf Basis des zuvor erzeugten Modells werden hier alle Elib-Einträge durchlaufen und neue Schlüsselwörter extrahiert. Dieses Verfahren ist insgesamt sehr aufwändig, da für alle 69.000 Elib-Einträge zunächst der Inhalt aus den Originaldateien ausgelesen und deren Sprache bestimmt werden muss. Hierzu findet ebenfalls die `ElibItem`-Klasse Verwendung. Für die Weiterverarbeitung durch die `Kea`-Komponente wird anschließend der Volltext-Inhalt der Elib-Dokumente als Datei gespeichert. Zuletzt findet innerhalb der `extractKeywords()`-Methode die eigentliche Extraktion der Schlüsselwörter statt.

- `KeywordMerger`: übernimmt die Aufgabe der Zusammenführung (engl. merging) der beiden Schlüsselwortmengen. Dazu werden alle Elib-Einträge durchlaufen und zunächst pro Eintrag ein Merging der Schlüsselwortmengen vollzogen. Im Anschluss daran wird eine globale Zusammenführung durchgeführt (`mergeAndRankAllKeywords()`). Dazu werden gleichlautende Schlüsselwörter innerhalb eines `Keyword`-Objekts zusammengeführt. Die `Keyword`-Klasse beinhaltet verschiedene Attribute zur Abbildung eines Schlüsselworts. Jedes Schlüsselwort hat zunächst einen eindeutigen Identifier. Dieser Identifier dient im Rahmen der anschließenden Überführung in den Thesaurus zur Generierung eines URI. Daneben ist eine Liste von Verweisen zu ähnlichen Schlüsselwörtern vorhanden. Weiterhin gibt es ein Attribut zum Zählen des Schlüsselworts-Vorkommens, das zur Auswahl der n häufigsten Schlüsselwörter herangezogen wird. Aufgrund der Tatsache, dass ein Schlüsselwort in mehreren Sprachen verwendet werden kann, ist weiterhin eine Liste, in der alle relevanten Sprachen abgelegt werden, vorgesehen. Die Sprache eines Schlüsselworts ist im Rahmen der Relationsbestimmung von Bedeutung. Bei der globalen Zusammenführung werden alle Schlüsselwörter auf Basis eines String-Vergleiches innerhalb einer `Keyword`-Objekt-Liste zusammengeführt. Nach dem globalen Merging wird diese Liste nach dem Vorkommen der Schlüsselwörter mit Hilfe der `KeywordComparer`-Klasse sortiert. Zuletzt findet die Reduktion der Schlüsselwortmenge statt. Hier werden die n häufigsten Schlüsselwörter zur Weiterverarbeitung ausgewählt.

- `KeywordSimiliarity`: ist für die Relationsbestimmung sowie die Überführung der Konzepte in den Thesaurus verantwortlich. Dazu werden alle `Keyword`-Objekte schrittweise durchlaufen und die Ähnlichkeit eines jeden Schlüsselwortes mit allen anderen Schlüsselwörtern berechnet (`computeSimiliarities()`). Die Berechnung findet nach dem in Abbildung 7.4 veranschaulichten Prinzip statt. Listing 8.5 zeigt den entsprechenden Quellcode-Ausschnitt innerhalb der betrachteten Methode. Zunächst wird mit Hilfe der `Disco`-Komponente die Ähnlichkeit zwischen zwei Schlüsselwörtern bestimmt (Zeile 4). Der hier berechnete Ähnlichkeitswert zwischen `kwStr1` und `kwStr2` basiert also auf der Kookkurrenz-Analyse. Für die verschiedenen Sprachen sind jeweils eigene `DISCO`-Instanzen vorgesehen, die in einer Java HashMap verwaltet werden. Liegt der berechnete Wert oberhalb des vorgegebenen Schwellwertes, wird eine Relation zwischen den beiden Schlüsselwörtern inklusive dem Ähnlichkeitswert vermerkt (Zeile 9). Andernfalls wird eine Ahnlichkeitsbestimmung auf Basis eines Zeichenkettenvergleichs mit Hilfe der `SimMetrics`-Komponente durchgeführt (Zeile 11 bis 15). Für diese Berechnung wird die Jaro-Winkler-Distanz genutzt, da mit dieser Metrik innerhalb von Testläufen die besten Ergebnisse erzielt wurden. Nachdem die Relationsbestimmung stattgefunden hat, werden als nächstes alle Schlüsselwörter und deren Beziehungen als Konzepte in Thesaurus überführt (`exportToThesaurus()`). Dazu wird pro Schlüsselwort ein `ThesaurusConcept`-Objekt erzeugt. Um die Konzepte persistieren zu können, verfügt die `KeywordSimiliarity`-Klasse über eine Referenz vom Typ `Thesaurus`. Mit Hilfe dieser Referenz können die Konzepte im Triple Store der `Thesaurus`-Komponente abgelegt werden. Als letzter Schritt wird danach die Überführung der Triple-Store-Inhalte in den invertierten Indexes angestoßen.

```
1  float similarity = 0;
2  if (!kwStr1.equals(kwStr2) && !kwStr1.isEmpty() && !kwStr2.isEmpty()) {
3      try {
4          similarity = this.discos.get(keyword.getLanguage()).
               firstOrderSimilarity(kwStr1, kwStr2);
5      } catch (IOException e) {
6          LOGGER.info("Error while computing firstOrderSimilarity between " +
               kwStr1 + " and " + kwStr2);
7      }
8      if (similarity > DISCO_SIMILARITY_THRESHOLD) {
9          keyword.addSimiliarKeyword(keywordToCompare, similarity);
10     } else {
11         AbstractStringMetric metric = new JaroWinkler();
12         similarity = metric.getSimilarity(kwStr1, kwStr2);
13         if (similarity > JARO_WINKLER_SIMILARITY_THRESHOLD) {
14             keyword.addSimiliarKeyword(keywordToCompare, similarity);
15         }
16     }
17 }
```

Listing 8.5: Quellcode-Auszug zur Relationsbestimmung aus der `computeSimiliarities()`-Methode

8.5. Integration der semantischen Suche in die Search-Komponente

Zuletzt wird nachfolgend die Integration der semantischen Suchfunktionalität in den Such- und Darstellungsprozess der `Search`-Komponente aufgeführt. Zunächst wird dazu erörtert, wie diese Komponente die `Thesaurus`-Schnittstelle innerhalb der Schlüsselwortsuche des KnowledgeFinder verwendet. Zuletzt findet eine Beschreibung der Suchergebnisverarbeitung sowie -anzeige innerhalb der GUI statt.

8.5.1. Integration in den Suchprozess

Das Klassendiagramm in Abbildung 8.7 zeigt Details der internen Beschaffenheit der `Search`-Komponente. Für eine bessere Unterscheidung sind die Klassen unterschiedlich farblich gekennzeichnet. Lila Klassen sind im Rahmen der Integration der semantischen Suche neu hinzugefügt und gelb markierte Klassen angepasst worden.

Aufgrund der vielfältigen Einsatzzwecke des KnowledgeFinder sind die Pakete der `Search`-Komponente modulartig aufgebaut. Wie den Paketnamen zu entnehmen, ist diese Komponente weiterhin nach dem MVC-Prinzip (Model View Contoller) strukturiert. Die im Paket `common.search` aufgeführten Klassen sind dabei Basisklassen, die viele Grundfunktionalitäten zur Suche und Ergebnisdarstellung bereitstellen. Das `elib.search`-Paket repräsentiert die spezifischen Klassen des Elib-Portals. Diese Klassen sind Spezialisierungen der Basisklassen und nutzen sowie erweitern deren Funktionsumfang. Mit Hilfe dieser Vererbungshierarchie ist es möglich, den KnowledgeFinder aufbauend auf Grundfunktionen an die Eigenheiten der jeweiligen Anwendungsfälle anzupassen. Neben den Paketen der `Search`-Komponente sind Klassen der `Thesaurus-Core`-Komponente zu sehen.

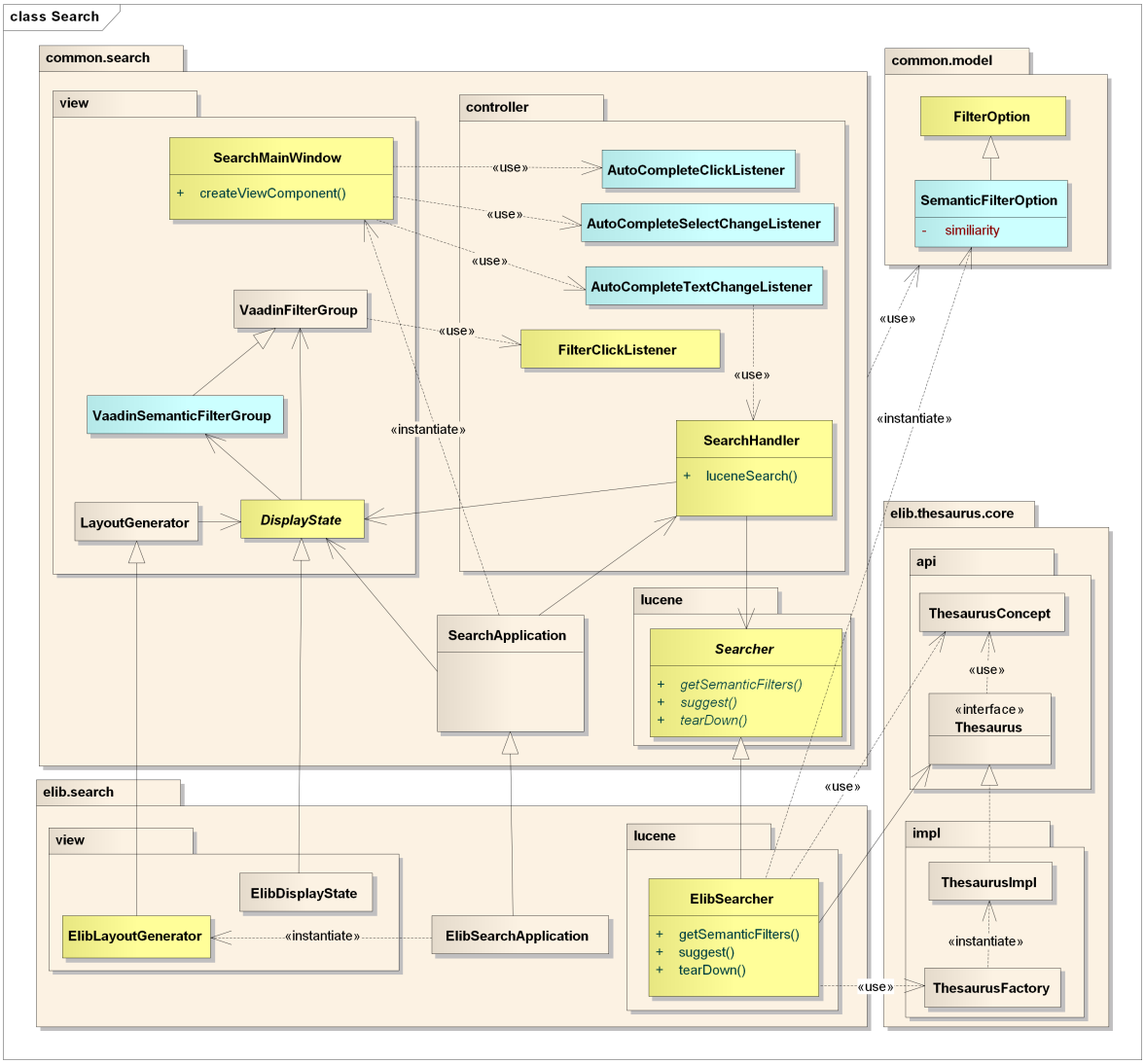


Abbildung 8.7.: Klassendiagramm der `Search`-Komponente

Zur Integration der semantischen Suche in den ursprünglichen Suchprozess ist hauptsächlich die Anpassung an der `ElibSearcher`-Klasse zu erwähnen. Diese Klasse ist der Dreh- und Angelpunkt innerhalb der Schlüsselwortsuche des Elib-Portals. Diese Klasse führt die Schlüsselwortsuche im Index aus und bereitet die Ergebnisse entsprechend auf. Aufgrund dieser zentralen Rolle verfügt die `ElibSearcher`-Klasse über ein `Thesaurus`-Attribut, mit dem die semantische Suche durchgeführt werden kann. Die Funktion `getSemanticFilters()` dient zur Erzeugung von semantischen Filtern auf Basis des zum Suchbegriff gefundenen Konzepts. Diese Filter werden in der Konzept-Facette zur Anzeige gebracht. Zum Auffinden eines Konzepts wird die `findMostRelevantConcept()`-Methode des `Thesaurus` genutzt. Neben der Suche von Konzepten für die facettierte Darstellung, integriert der `ElibSearcher` die Autovervollständigungs-Suche. Hier ist anzumerken, dass nicht der vollständige Funktionsumfang implementiert wurde. Die Integration beschränkt sich auf eine einfache Variante, in der lediglich das präferierte Label der ermittelten Konzepte weiterverarbeitet wird. Die semantischen Beziehungen werden durch die Autovervollständigung nicht verwendet und dargestellt (vgl. Szenario S-2). Hervorzuheben ist, dass diese Beziehungen jedoch nachgelagert durch die facettierte Darstellung eines Konzepts dem Benutzer zur Verfügung gestellt werden. Listing 8.6

zeigt den entsprechenden Quellcode der zuständigen `suggest()`-Methode der `ElibSearcher`-Klasse. Diese Methode reicht zunächst die Funktionsaufrufe an das Pendant des `Thesaurus` weiter (Zeile 3). Danach wird die Auswahlliste der Autovervollständigung auf Basis der präferierten Konzept-Label zusammengesetzt (Zeile 4 bis 7).

```
1 public List<String> suggest(String searchInput) {
2     List<ThesaurusConcept> concepts = thesaurus.suggestConcepts(searchInput,
          SearchMainWindow.MAX_AUTOCOMPLETE_ITEMS);
3     List<String> prefLabels = new Vector<String>();
4     for(ThesaurusConcept c: concepts) {
5         prefLabels.add(c.getPrefLabel());
6     }
7     return prefLabels;
8 }
```

Listing 8.6: Quellcode-Auszug der `suggest()`-Methode

8.5.2. Integration in die Benutzerschnittstelle

Neben der Einbindung der semantischen Suchfunktionen der `Thesaurus`-Komponente in den Suchprozess, mussten weiterhin verschiedene Erweiterungen innerhalb der View- und Controller-Klassen vollzogen werden. Diese Anpassungen hatten zum Ziel, die zuvor in Abschnitt 7.2.5 entworfenen Erweiterungen der Benutzerschnittstelle umzusetzen. Zunächst werden die zentralen Schritte zur Realisierung der Konzept-Facette aufgezeigt und anschließend die Integration der Autovervollständigung erörtert.

Für die facettierte Darstellung eines Thesaurus-Konzepts innerhalb der Elib-Portal-GUI sind folgende Anpassungen von Bedeutung:

- `SemanticFilterOption`: stellt einen semantischen Filter auf der Basis der Konzepte-Eigenschaften dar (präferiertes Label, Synonyme, Ober- sowie Unterkonzepte etc.). Die Sortierung der Filter erfolgt standardmäßig nach der Anzahl gefundener Treffer im Index. Für jeden Filter wird dazu eine eigene Suchanfrage zur Ermittlung der Trefferanzahl abgesetzt. Das Attribut `similiarity` speichert das zuvor beschriebene Ähnlichkeitsmaß verwandter Konzepte, das ebenfalls zur Sortierung herangezogen werden kann.

- `VaadinSemanticFilterGroup`: ist für die facettierte Darstellung eines Konzepts innerhalb der GUI zuständig. Die Klasse ist eine Spezialisierung der `VaadinFilterGroup`-Klasse. In dieser Klasse sind VAADIN-spezifische Implementierungen zur Anzeige von Filtern in der Benutzeroberfläche vorgesehen.

- `FilterClickListener`: diese Controller-Klasse reagiert auf Filter-Click-Events innerhalb der GUI. Zur Umsetzung des spezifischen Verhaltens der Konzept-Facette ist hier die Ereignisbehandlung angepasst worden.

Zur Realisierung der Autovervollständigung innerhalb der UI sind folgende Anpassungen hervorzuheben:

- `SearchMainWindow`: repräsentiert das Hauptfenster der Benutzerschnittstelle. Hier wird die Suchergebnisliste zur Anzeige gebracht. Dieses Hauptfenster wurde mit Hilfe des VAADIN Plugin *Overlays*[5] erweitert. Das Overlays Plugin ermöglicht das flexible Positionieren von

[5] http://vaadin.com/directory#addon/overlays (Zugriffsdatum 28.02.2013)

UI-Elementen relativ zu anderen UI-Elementen. Innerhalb eines solchen Overlay wurde eine Auswahllisten-Element integriert. Diese Auswahlliste wird durch einen Datencontainer befüllt. Um diesen Datencontainer abhängig von der Nutzereingabe zu aktualisieren, sind mehrere Listener-Klassen im `controller`-Paket vorgesehen. Abbildung 8.8 zeigt ein Beispiel für das Aussehen der implementierten Autovervollständigung innerhalb der GUI des Elib-Portals. Im dargestellten Fall hat der Nutzer den Suchbegriff „Flugl“ in die Eingabemaske eingetippt.

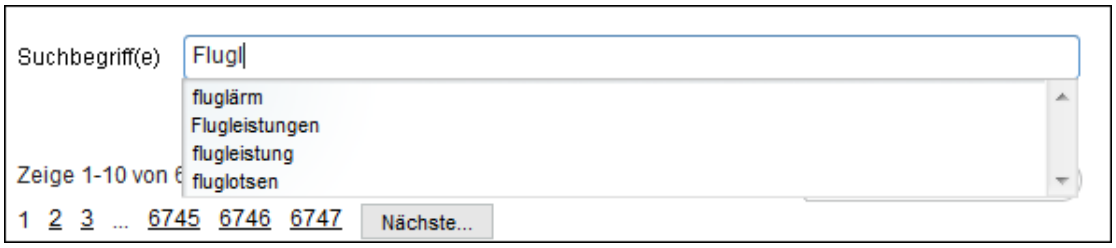


Abbildung 8.8.: Beispiel der Autovervollständigung für die Suchanfrage „Flugl“

- `AutoCompleteTextChangeListener`: ist der wichtigste Listener im Rahmen der Autovervollständigung. Das Sequenzdiagramm in Abbildung 8.9 stellt den im Folgenden beschriebenen Vorgang zur Aktualisierung der Autovervollständigung schematisch dar. Der Listener wird aktiv, sobald der Nutzer innerhalb der Eingabemaske einen Suchbegriff eintippt (Schritt 1.0 bis 1.1). Ist der Suchbegriff länger als zwei Zeichen, so wird zunächst der Datencontainer geleert (Schritt 1.2). Im Anschluss daran ruft der Listener die `suggest()`-Methode des `ElibSearcher` auf (Schritt 1.3). An dieser Stelle wird die `Thesaurus`-Instanz angesprochen. Diese Instanz sucht nach entsprechenden Konzepten innerhalb des Thesaurus-Index und ermittelt die semantischen Beziehungen im Triple Store (Schritt 1.4 bis 1.5). Die Label der gefundenen Konzepte werden anschließend in den Datencontainer geladen und das Overlay-Element innerhalb der Benutzeroberfläche angezeigt (Schritt 1.6 bis 1.8).

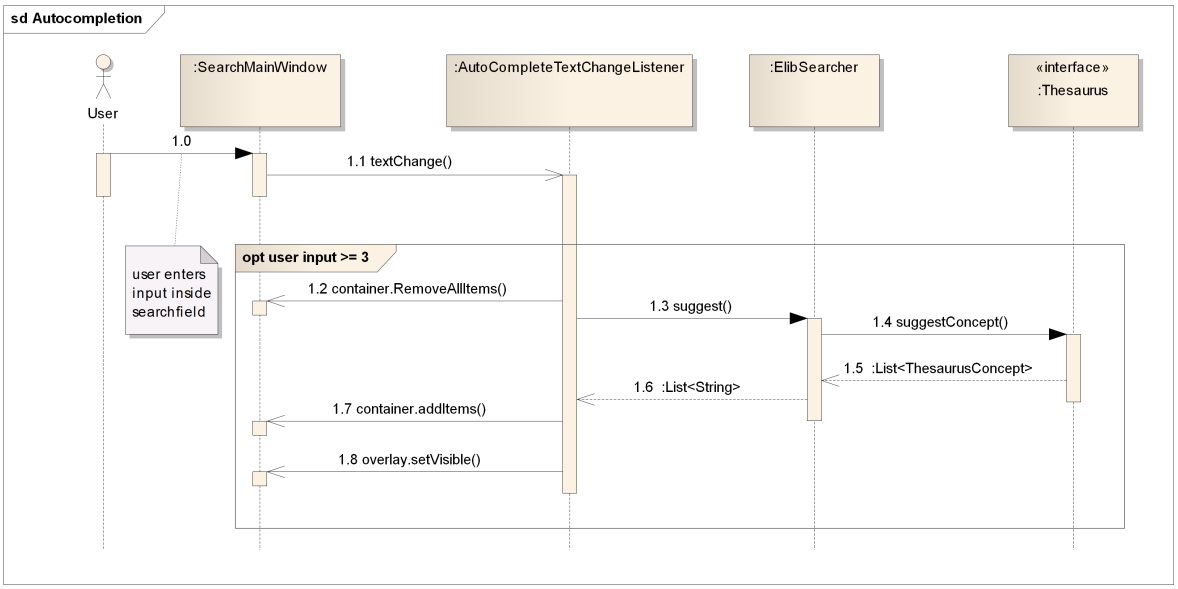


Abbildung 8.9.: Sequenzdiagramm zur Aktualisierung der Autovervollständigung

9. Evaluation

Nachdem in Kapitel 7 eine semantische Suche nach dem Prinzip des konzeptbasierten Dokumenten-Retrieval hergeleitet und konzipiert wurde, wird in diesem Kapitel die Evaluation des vorgestellten Konzepts vollzogen. Um eine Bewertung durchführen zu können, wird die zuvor in Kapitel 8 vorgestellte prototypische Implementierung herangezogen. Zunächst steht die Suchergebnisqualität im Mittelpunkt der Untersuchung. Hier wird betrachtet, ob durch die Integration der semantischen Suche innerhalb des KnowledgeFinder eine Verbesserung der Suchergebnisqualität erreicht werden kann. Aufbauend auf dieser Analyse wird anschließend in Abschnitt 9.2 die Umsetzbarkeit und der Mehrwert semantischer Technologien im Kontext des DLR beleuchtet.

9.1. Evaluation der semantischen Suche

In diesem Abschnitt wird die zuvor entwickelte und implementierte semantische Suche bewertet. Das Ziel besteht darin zu ermitteln, welchen Einfluss die durchgeführten Erweiterungen auf die Suchergebnisqualität des KnowledgeFinder haben. Mit Blick auf die in Abschnitt 1.1 beschriebene Problemstellung ist dabei zu evaluieren, inwiefern es dem fachübergreifend suchenden Wissenschaftler erleichtert wird, relevante Informationen aus anderen Fachbereichen zu finden. Zunächst wird dazu das verwendete Vorgehen im Rahmen dieser Evaluation erörtert. Im Anschluss daran werden die Ergebnisse der Untersuchung vorgestellt und diskutiert.

9.1.1. Vorgehen zur Bewertung der Suchergebnisqualität

Wie in Kapitel 2 ausgeführt, werden zur Evaluation von IR-Systemen die Parameter Precision und Recall herangezogen. Auf Basis von Testkollektionen werden Werte für diese Parameter bestimmt und aufbauend auf diesen Werten eine Bewertung des IR-Systems durchgeführt. Diese Herangehensweise ist im Rahmen der hier durchgeführten Evaluation nicht anwendbar. Dies ist darin begründet, dass die entwickelte semantische Suche nicht direkt in die eigentliche Schlüsselwortsuche des KnowledgeFinder integriert ist. Stattdessen wird sie innerhalb der Konzept-Facette und der Autovervollständigung zur Bereitstellung von Benutzerunterstützungen verwendet. Die semantische Suche hat also keinen direkten Einfluss auf die Suchergebnisse. Eine Evaluation auf der Basis von Precision und Recall ist hier nicht möglich, da diese Parameter sich lediglich darauf beziehen, ob relevante Dokumente in der Ergebnismenge auftauchen oder nicht. Die Hilfestellungen durch die Facetten und die Autovervollständigung würden von dieser Evaluierungsmethode nicht erfasst.

Das zuvor beschriebene Problem stellt dabei ein allgemeines Problem der Evaluation semantischer Suchsysteme dar. Anders als in der IR Community existiert im Kontext semantischer Suchsysteme kein gemeinsames Verständnis darüber, wie und mit welchen Parameter diese Systeme evaluiert und miteinander verglichen werden können (vgl. [Hildebrand et al. 2007], [Sánchez 2009, S.107] und [Tran et al. 2011]). Diesen Mangel begründen Hildebrand et al. mit der starken Fragmentierung der Wissenschaft auf diesem Themengebiet und der daraus resultierenden Anzahl unterschiedlicher Ansätze zur semantischen Suche (vgl. auch Abschnitt 4.3). Tran et al. heben hervor, dass die Schaffung einer gemeinsamen Basis zur Evaluation

und zum Vergleich semantischer Suchsysteme eine der großen Herausforderungen in diesem Forschungsbereich ist [Tran et al. 2011]. Sánchez weißt darauf hin, dass vorhandene Ansätze meist auf Methoden basieren, die den Endbenutzer im Rahmen der Evaluation mit einbeziehen [Sánchez 2009, S. 107]. Das heißt, hier werden nutzerorientierte Verfahren der Mensch-Computer-Interaktion (MCI) eingesetzt.

Mit Blick auf die zuvor beschriebene Art der Verwendung der semantischen Suche im KnowledgeFinder wäre eine Evaluation unter der Zuhilfenahme eines Nutzertests eine sinnvolle Alternative. Hier könnte auf der Basis eines Vorher-Nachher-Vergleichs die (subjektiv empfundene) Suchergebnisqualität des KnowledgeFinder mittels einer Nutzerbefragung erhoben und untersucht werden. Die Vorbereitung, Durchführung und Nachbereitung von solchen Nutzerbefragungen sind jedoch sehr aufwändig und im Rahmen dieser Machbarkeitsstudie nicht durchführbar. Stattdessen soll im Folgenden die Suchergebnisqualität anhand einzelner Stichproben untersucht und bewertet werden. Im Fokus steht dabei, inwiefern dem fachübergreifend recherchierenden Wissenschaftler das Auffinden von Informationen erleichtert wird. Die Bewertung, ob eine Verbesserung vorliegt oder nicht, erfolgt aus der subjektiven Sichtweise des Autors. Die Stichproben orientieren sich dabei an den in Kapitel 6 beschriebenen Szenarien. Um repräsentative Stichproben durchführen zu können, basieren die herangezogenen Suchanfragen zudem auf einer Logfile-Analyse des Elib-Systems. Diese Logfiles stammen aus dem Zeitraum von Juni bis Dezember 2012 und beinhalten ca. 36.000 Suchanfragen. Diese Suchanfragen wurden extrahiert, bereinigt und nach ihrer Häufigkeit sortiert. Tabelle 9.1 zeigt die 5 häufigsten Suchanfragen an das Elib-System[1].

Tabelle 9.1.: Fünf häufigsten Suchanfragen an das Elib-System aus dem Zeitraum Juni bis Dezember 2012

Suchanfrage	**Anzahl**	**Häufigkeit**
aero	1.708	4,68 %
rocket	180	0,49 %
smartphone sensor	53	0,14 %
sensor algorithm	52	0,14 %
methane	48	0,13 %

Neben der Orientierung an den Anforderungsszenarien und der Verwendung von Stichproben, sollen die Untersuchungen weiterhin auf der Grundlage von zwei verschiedenen Thesauri stattfinden. Zum einen ein automatisiert erzeugter Thesaurus, der lediglich aus einer Ansammlung von extrahierten Themen besteht. Zum anderen ein Thesaurus, der innerhalb eines kleinen Themenausschnitts validiert und angepasst wurde. In diesem Thesaurus sind für eine gewisse Anzahl an Themen auch Synonyme, Oberthemen und Unterthemen abgebildet. In diesem Zusammenhang soll untersucht werden, inwiefern sich die Beschaffenheit der verschiedenen Thesauri auf die semantische Suche und die Suchergebnisqualität auswirkt.

9.1.2. Bewertung der Suchergebnisqualität

Im Folgenden soll eine Bewertung der Suchergebnisqualität nach dem zuvor geschilderten Vorgehen durchgeführt werden. Zunächst wird dazu die Suchergebnisqualität der semantischen Suche auf Basis des automatisiert erzeugten Thesaurus beleuchtet. Danach findet eine Untersuchung auf Grundlage des angepassten Thesaurus statt.

[1] Eine vollständige Liste aller Suchanfragen ist im Anhang A.1 zu finden.

Automatisierter Thesaurus

Um einem automatisierten Thesaurus zu erhalten, wurden mit der Extraktion-Komponente mehrere Testläufe auf Basis der Elib-Daten durchgeführt[2]. Die Ergebnisse der Erstellung sollen im Folgenden kurz vorgestellt und untersucht werden. Aufbauend auf dieser Untersuchung wird für die anschließende Betrachtung der Suchergebnisqualität ein geeigneter Thesaurus ausgewählt.

Tabelle 9.2 zeigt zunächst die Resultate der Gewinnung von zusätzlichen Schlüsselwörtern in Form eines Vorher-Nachher-Vergleiches. Der Gesamtprozentsatz der Einträge mit Schlüsselwörtern konnte mit Hilfe der Extraktions-Komponente um ca. 8% auf insgesamt 69,69% gesteigert werden. Diese recht geringe Verbesserung ist damit zu erklären, dass ca. 60% aller Elib-Einträge weder Beschreibungen noch Volltext-Dokumente enthalten. Auf Basis von kurzen Zeichenketten, wie z.B. der Titel einer Elib-Publikation, ist jedoch eine Bestimmung von zusätzlichen Schlüsselwörtern mit Hilfe des KEA-Algorithmus nur begrenzt möglich. Wie weiterhin in Tabelle 9.2 dargestellt, konnte die durchschnittliche Anzahl von Schlüsselwörtern pro Elib-Eintrag mehr als verdoppelt werden. Hier ist eine erhebliche Steigerung durch das verwendete Verfahren festzustellen.

Tabelle 9.2.: Vorher-Nachher-Vergleich der Schlüsselwort-Abdeckung des Elib-Datenbestands

	Vorher	**Nachher**
# Einträge mit Schlüsselwörtern	42.213	48.037
# Schlüsselwörter	175.510	482.401
Ø Schlüsselwörter pro Eintrag	≈ 4	≈ 10

Tabelle 9.3 fasst die Ergebnisse der anschließend durchgeführten Relationsbestimmungen zusammen. Zunächst wurde ein Testlauf mit einem minimalen Vorkommen von 10 Schlüsselwörtern durchgeführt. Dieser Wert wurde danach schrittweise verringert und die Qualität der erzeugten Thesauri untersucht. Die Anzahl gewonnener Konzepte sowie deren Relationen verhält sich ungefähr umgekehrt proportional zum Schlüsselwortvorkommen. Wie erwartet, ist die Zeitkomplexität der Relationsbestimmung $\mathcal{O}(n^2)$ in Abhängigkeit zur Konzept-Anzahl. Für die nachfolgende Untersuchung wird als Datenbasis der farblich hervorgehobene Thesaurus Nr. 3 ausgewählt. Diese Auswahl wurde getroffen, da die Thesauri Nr. 1 und Nr. 2 eine zu geringe Anzahl an Konzepten beinhalten. Das heißt, diese Thesauri werden als nicht repräsentativ genug bezüglich des Elib-Datenbestands angesehen. Der Thesaurus Nr. 4 enthält zwar die meisten Konzepte. Aufgrund von Stichproben wurde hier jedoch ein hoher Anteil unbrauchbarer Konzepte festgestellt. Listing 9.1 zeigt zwei Beispiele solcher unbrauchbaren Konzepte in Form von XML-basierten RDF-Aussagen.

Tabelle 9.3.: Ergebnisse der Relationsbestimmung

Nr.	**# Min. Vorkommen**	**# Konzepte**	**# Relationen**	**Laufzeit in h**
1.	10	7.062	3.821	8,30
2.	7	10.144	5.945	10,42
3.	5	14.702	9.101	34,35
4.	3	18.984	12.135	61,32

[2]Die erzeugten Thesauri sind in Anhang A.1 aufgeführt.

```
<rdf:Description rdf:about="http://elib.dlr.de/concept/#c_91343">
    <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
    <prefLabel xmlns="http://www.w3.org/2004/02/skos/core#">31a</prefLabel>
</rdf:Description>

<rdf:Description rdf:about="http://elib.dlr.de/concept/#c_47522">
    <rdf:type rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
    <prefLabel xmlns="http://www.w3.org/2004/02/skos/core#">2nd</prefLabel>
</rdf:Description>
```

Listing 9.1: Beispiele unbrauchbarer Thesaurus-Konzepte

Zunächst wird die in Szenario S-2 beschriebene Autovervollständigung auf Basis des automatisiert erstellten Thesaurus untersucht. Diese Erweiterung ist der erste Berührungspunkt des Nutzers mit der semantischen Suche des Elib-Portals. Abbildung 9.1 zeigt die Vorschläge der Autovervollständigung für die am häufigsten durchgeführte Elib-Suchanfrage „aero“.

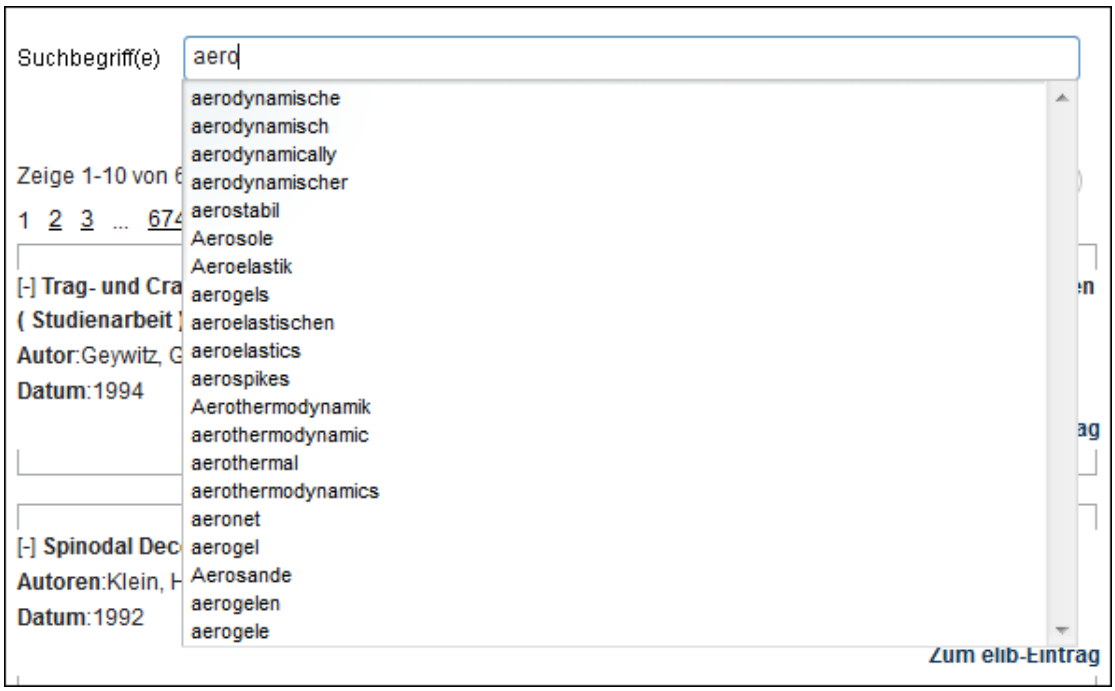


Abbildung 9.1.: Beispiel der Autovervollständigung für die Suchanfrage „aero“

Wie im Szenario gefordert, schlägt die Autovervollständigung automatisch eine Vielzahl von Themen zur Suchanfrage vor. Im Vergleich zur ursprünglichen Benutzeroberfläche des Elib-Portals, ist diese Unterstützung durch die Autovervollständigung als Verbesserung zu werten. Mit Blick auf die in den Anforderungen beschriebene Benutzerrolle, kann gesagt werden, dass diese Hilfestellung das fachübergreifende Suchen deutlich vereinfacht. Mögliche Themengebiete werden dem Wissenschaftler bereits während der Artikulierung seines Informationsbedarfes präsentiert. Insbesondere bei einem nur oberflächlichen Einblick in fremde Fachbereiche, ist diese Unterstützung im Rahmen einer explorativen Suche als positiv zu bewerten (vgl. auch Ziel Z-1.4). Anzumerken ist, dass neben vielen hilfreichen Einträgen auch einige nicht brauchbare Einträge vorgeschlagen werden (z.B. „aerodynamische“ oder „aerodynamischer“). Diese Einträge wurden fälschlicherweise im Rahmen der Extraktion als Thema identifiziert. Hier zeigt sich der Bedarf der Nachbearbeitung einer solchen Automatisierung. Die unterschiedliche Groß- und Kleinschreibung der Bezeichnungen sind ebenfalls

Beispiele für den Nachbarbeitungsbedarf. Anders als in Szenario S-2 erläutert, unterbreitet die Autovervollständigungs-Funktion nur Themen, die das eingegebene Suchwort enthalten. Semantische Beziehungen werden nicht präsentiert, da der geforderte Funktionsumfang der Autovervollständigung nicht vollständig realisiert wurde. Es ist anzunehmen, dass durch das Aufzeigen semantischer Beziehungen der Mehrwert der Autovervollständigung nochmals gesteigert werden könnte. Problematisch ist die eingeschränkte Anzahl präsentierter Themen innerhalb der Auswahlliste. Im dargestellten Fall ist diese Anzahl auf 20 begrenzt. Durch diese Begrenzung werden weitere evtl. relevante Themen nicht angezeigt. Beispielsweise fehlt ein Eintrag zum Thema „Aerodynamik“. Gerade wenn das Suchwort wie bei „aero“ nur aus wenigen Zeichen besteht, tritt dieses Problem auf.

Abbildung 9.2 zeigt ein Beispiel für einen anschließenden Suchvorgang durch Auswahl eines Eintrages aus der Autovervollständigung. Im dargestellten Fall wurde der Eintrag „Aeroelastik“ selektiert und insgesamt 59 Einträge durch die Schlüsselwortsuche des KnowledgeFinder gefunden. Wie in Szenario S-1 verlangt, ermittelt die gleichzeitig aktivierte semantische Suche der Thesaurus-Komponente die dazu verwandten Themen. Im dargestellten Fall wird „Aeroelasticity“ – die englische Übersetzung von Aeroelastik – vorgeschlagen. Hier zeigt sich ein weiterer Zugewinn im Vergleich zum ursprünglichen System. Ein domänenfremder Wissenschaftler muss in diesem Fall nicht selber nach der englischsprachigen Entsprechung zu Aeroelastik recherchieren. Der automatisiert erstellte Thesaurus bildet diesen Zusammenhang ab und die semantische Suche findet und präsentiert ihn innerhalb der Benutzeroberfläche. Durch diese Art der Hilfestellung ist es möglich, dem explorativ suchenden Wissenschaftler weitere eventuell relevante Dokumente zugänglich zu machen (vgl. Szenario S-1).

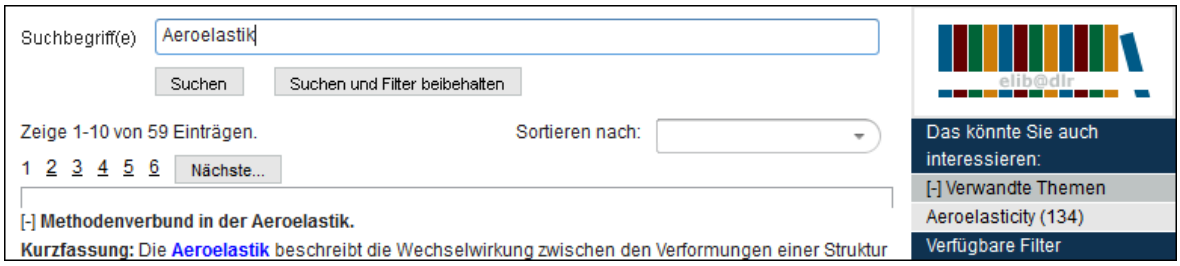


Abbildung 9.2.: Suchergebnisse zur Suchanfrage „Aeroelastik“

Es gibt aber auch Fälle, in denen die Autovervollständigung keine hilfreiche Unterstützung im Rahmen der Informationssuche liefert. Abbildung 9.3 zeigt ein solches Beispiel. Bei der Eingabe der Suchanfrage „smartphone sensor“ werden durch die Autovervollständigung die dargestellten Themen vorgeschlagen. Darunter ist jedoch kein Eintrag zum Thema „Smartphone“. Mit Blick auf die Elib-Daten ist aber festzustellen, dass Dokumente zu dieser Thematik durchaus gefunden werden können. Das bedeutet, dass der Nutzer an dieser Stelle eigentlich einen Vorschlag durch die Autovervollständigung erwarten sollte. Hier liegt eine mangelhafte Unterstützung dieser Erweiterung vor. Der Grund dafür liegt in der Filterung dieser Thematik im Rahmen der Extraktion. Hier zeigt sich wiederum, dass die Konfiguration der automatisierten Thesaurus-Erstellung immer eine schwierige Abwägung zwischen einer hohen Themenabdeckung und der Filterung von unbrauchbaren Einträgen ist. Hier ist zu entscheiden, welcher Datenbestand als repräsentativ anzusehen ist.

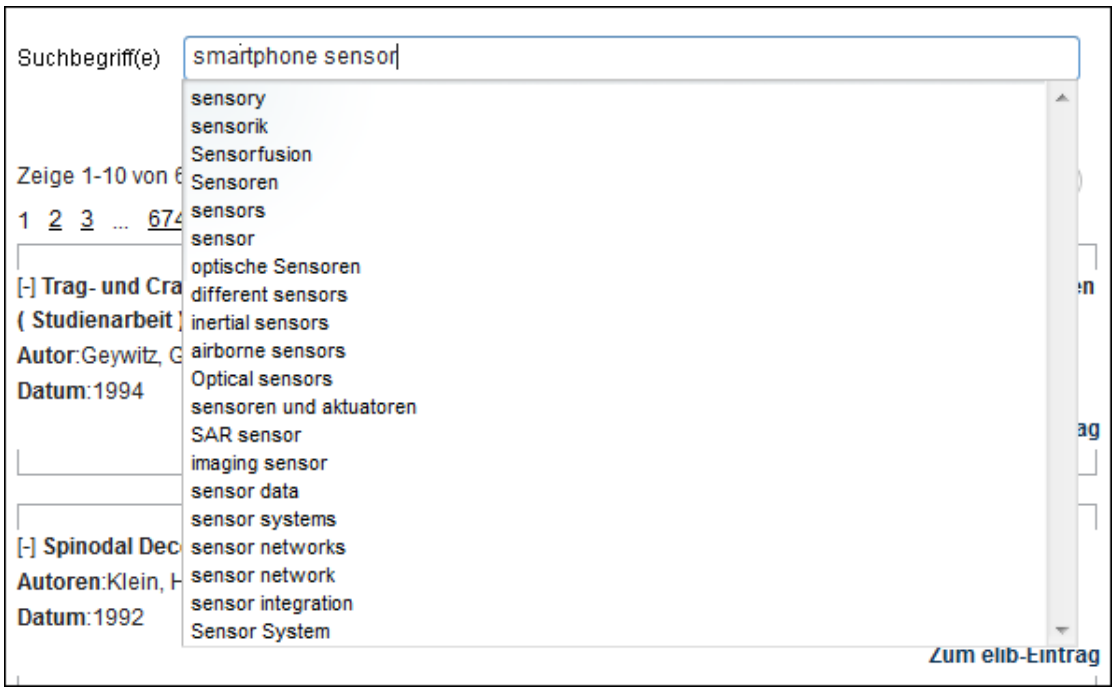
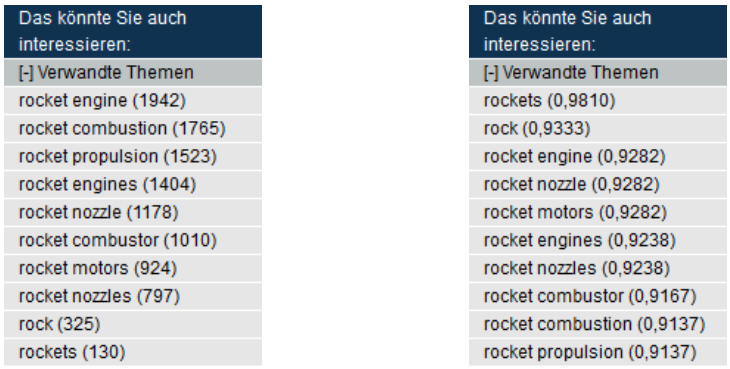


Abbildung 9.3.: Beispiel der Autovervollständigung für die Suchanfrage „smartphone sensor“

Als nächstes soll nun die Hilfestellung der in Szenario S-1 aufgeführten Konzept-Facette und die damit verbundene semantische Suche genauer betrachtet werden. Abbildung 9.4 zeigt zwei unterschiedliche Varianten dieser Darstellung zu der Suchanfrage „rocket“. Zum einen eine Sortierung nach der Anzahl der Treffer (a) und zum anderen eine Sortierung nach dem Ähnlichkeitsmaß (b). Anzumerken ist, dass für eine bessere Darstellung der Werte eine Rundung durchgeführt worden ist. Für die Suchanfrage „rocket“ schlägt die semantische Suche, wie im Szenario S-1 festgehalten, eine Liste von verwandten Themen vor. Auch diese Art der Hilfestellung ist ein Mehrgewinn im Rahmen der explorativen Suche. Der wissenschaftliche Mitarbeiter erhält so einen ersten Einblick in verknüpfte Themengebiete. Bei einer genaueren Betrachtung der dargestellten Vorschläge ist zu erkennen, dass es sich um Unterthemen zum eingegebenen Suchbegriff handelt. Daneben sind auch einzelne nicht korrekt zugeordnete Themen in der Facette aufgeführt (z.B. „rock“). Diese Zusammenhänge können jedoch durch das Extraktions-Verfahren nicht ermittelt werden. Hier wäre die Ermittlung von Beziehungstypen und die Verringerung der Fehleranfälligkeit erstrebenswert. Mit Blick auf die unterschiedlichen Sortierungen kann schwer beurteilt werden, welche Sortierungsart hilfreicher ist. Eine Kombination aus der Suchtreffer-Anzahl und dem Ähnlichkeitsmaß wäre eine mögliche Alternative. Hier könnte beispielsweise mit einer Farbgebung in Abhängigkeit zur Ähnlichkeit als grafisches Hilfsmittel gearbeitet werden.

Mit Blick auf die zuvor beschriebenen beispielhaften Stichproben, ist zu erkennen, dass die entwickelte semantische Suche auf Basis des automatisierten Thesaurus dem fachübergreifend suchenden Wissenschaftler bereits nützliche Hilfestellungen bereitstellt. Im Vergleich zu den Szenarien zeigt sich zudem, dass die semantische Suche dem geforderten Funktionsumfang zumindest in Teilen gerecht wird. Aufgrund der automatisierten Erstellung ist eine gewisse Fehlerquote vorhanden, die nicht außer Acht gelassen werden kann. Insgesamt kann aber gesagt werden, dass die Hilfestellungen der semantischen Suche das Auffinden von Informationen vereinfachen und zu einer Verbesserung der Suchergebnisqualität beitragen.



(a) Sortiert nach Treffer

(b) Sortiert nach Ähnlichkeitsmaß

Abbildung 9.4.: Facettierte Darstellung eines Thesaurus-Konzepts für die Suchanfrage „rocket“

Angepasster Thesaurus

Nachdem die automatisiert erzeugte Variante des Thesaurus untersucht worden ist, wird in diesem Abschnitt ein manuell angepasster Thesaurus als Datengrundlage verwendet. Hier wird ein zuvor in Abschnitt 7.2.5 dargestelltes Beispiel aufgegriffen. Der in diesem Abschnitt veranschaulichte Ausschnitt des Themengebiets der Solarenergie wurde dazu teilweise in einem Thesaurus überführt (siehe Abbildung 7.7). Der Mehrwert der semantischen Suchfunktionen im Kontext der fachübergreifenden Recherche wird mittels diesem Thesaurus anhand mehrerer beispielhafter Suchanfragen untersucht und im Vergleich zur automatisiert erzeugten Variante betrachtet. Dabei wird im weiteren Verlauf nur die Konzept-Facette (Szenario S-1) betrachtet, da die Autovervollständigung von der verbesserten Datengrundlage keinen Gebrauch machen würde.

Abbildung 9.5 zeigt die Suchergebnisse zur Suchanfrage „renewable energy“[3]. Hier wird dem Nutzer innerhalb der Facette das Unterthema „solar energy“ vorgeschlagen. Im Vergleich zum zuvor untersuchten Fall, kann durch den angepassten Thesaurus die in Abschnitt 7.2.5 beschriebene Themenhierarchie aufgelöst werden. Hier zeigt sich der volle Funktionsumfang der entwickelten semantischen Suche. Mit Blick auf die betrachtete Benutzerrolle sind die durch die Facette dargestellten Querbezüge wertvolle Unterstützungen im Rahmen der Informationssuche (vgl. Szenario S-1). Im abgebildeten Fall erfährt der fachübergreifend suchende Wissenschaftler, dass im Bereich der Erneuerbaren Energien auch Forschungsaktivitäten zum Thema der Solarenergie stattfinden.

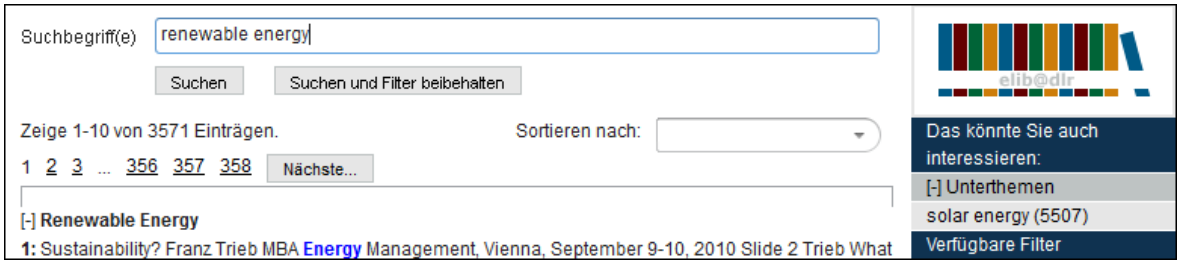


Abbildung 9.5.: Suchergebnisse zur Suchanfrage „renewable energy“

[3]Innerhalb des Elib-Logfile ist „renewable energy“ mit 0,0822% auf Platz Nr. 12 aller Suchanfragen.

Wird der Facetten-Eintrag „solar energy“ selektiert, so zeigt der KnowledgeFinder die in Abbildung 9.6 dargestellte Benutzeroberfläche. Für das Solarenergie-Themengebiet werden in dieser Oberfläche neben Synonymen auch eine Vielzahl an Oberthemen, Unterthemen und verwandte Themen innerhalb der Facette präsentiert. Die gegenseitige Verlinkung zwischen Ober- und Unterthemen ermöglicht ein beliebiges Erkunden der DLR-Themenlandschaft entlang von Hierarchien. Dadurch können innerhalb einer explorativen Suche vollständig neue Themengebiete erschlossen werden, die domänenfremden Personen sonst verschlossen bleiben würden. Im Vergleich zur vorherigen Version des Thesaurus, die nur ein unstrukturiertes Erschließen von verwandten Themengebieten ermöglicht, ist hier also nochmals eine Verbesserung der Unterstützung des Suchvorgangs zu vermerken. Aufgrund der Tatsache, dass die präsentierten Verbindungen nicht innerhalb einer Automatisierung erzeugt wurden, ist diese Unterstützung weiterhin als verlässlicher einzustufen als zuvor. Auch die Hilfestellung durch die angezeigten verwandten Themen ist im Vergleich ebenfalls gesteigert worden. Die dargestellte Verbindung der Solarenergie-Forschung mit dem Themengebiet des Klimawandels („climate change“) ist ein Beispiel hierfür. Solche Verbindungen können dem fachübergreifend suchenden Forscher wichtige Zusammenhänge erschließen. Ein automatisiertes Verfahren ist nur selten in der Lage derartige Querbezüge herzustellen.

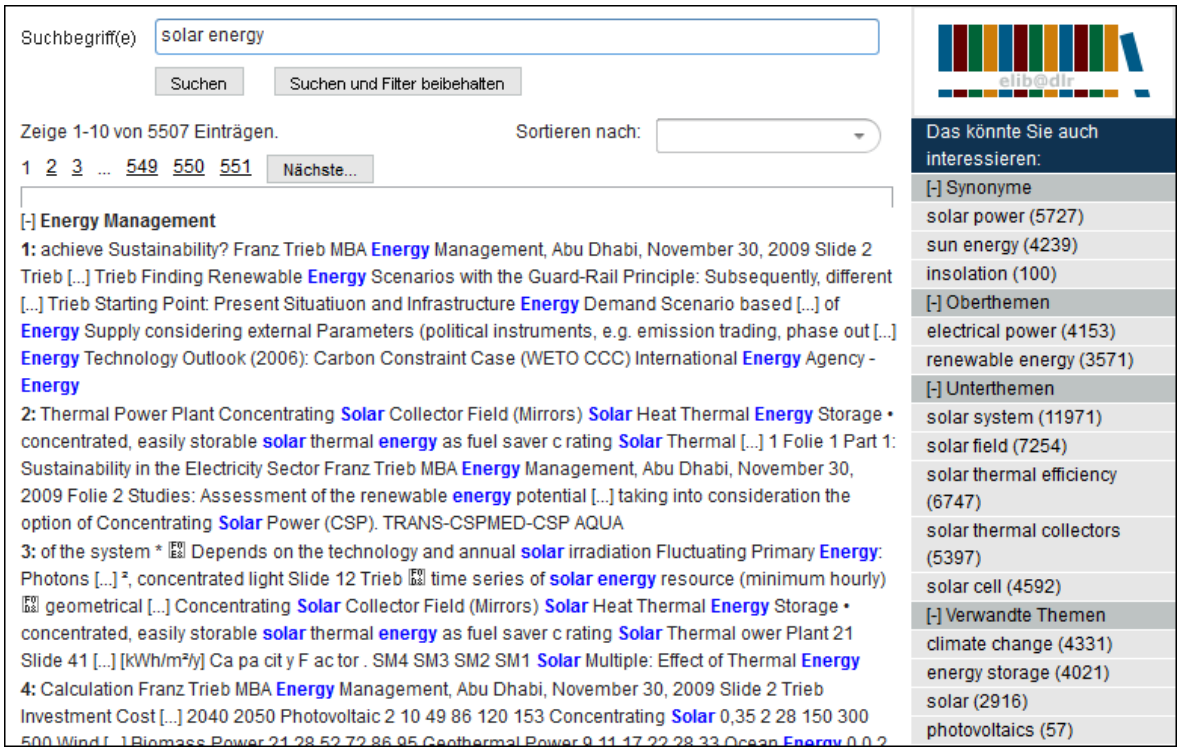


Abbildung 9.6.: Suchergebnisse zur Suchanfrage „solar energy“

Das Beispiel in Abbildung 9.7 zeigt wiederum die schon zuvor geschilderte gegenseitige Verknüpfung zwischen den Themengebieten. Sucht der Wissenschaftler nach dem Begriff „climate change“, so wird ihm die Querverbindung zum Forschungsfeld der Solarenergie ebenfalls dargestellt.

Insgesamt kann festgehalten werden, dass die Hilfestellungen der semantischen Suche auf Basis eines angepassten Thesaurus weiter gesteigert werden konnte. Die Unterstützung des wissenschaftlichen Mitarbeiters bei seiner Rechercheaktivitäten ist nochmals verbessert wor-

den. Durch das Aufzeigen von Themenhierarchien geht diese Unterstützung sogar über den in Szenario S-1 geforderten Funktionsumfang hinaus. Die Reichhaltigkeit und Verlässlichkeit der präsentierten semantischen Zusammenhänge ist zudem deutlich höher, als bei der automatisiert erzeugten Datengrundlage. Diese beiden Aspekte führen nochmals zu einer Erleichterung der Informationssuche und können als weitere Steigerung der Ergebnisqualität gewertet werden. Hier zeigt sich wiederum, dass eine nachgelagerte Validierung und Anpassung der Thesaurus-Inhalte unabdingbar ist, um das volle Potenzial der semantischen Suche des KnowledgeFinder ausschöpfen zu können.

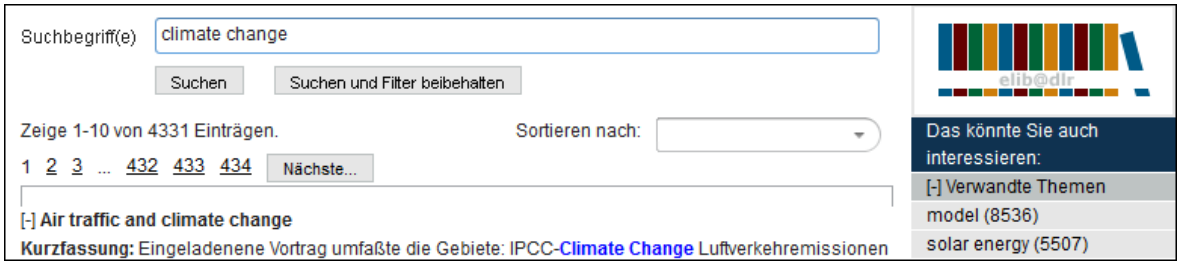


Abbildung 9.7.: Suchergebnisse zur Suchanfrage „climate change“

9.2. Semantische Technologien im Kontext des DLR

Abschließend findet in diesem Abschnitt eine kritische Betrachtung der Relevanz semantischer Technologien im Kontext des DLR statt. Hierzu dient die zuvor dargestellte Machbarkeitsstudie als Grundlage. Zunächst wird dazu eine Bewertung der eingesetzten Methoden sowie Technologien durchgeführt. Daran anschließend findet eine Einschätzung der Umsetzbarkeit und des Mehrwerts semantischer Technologien im gegebenen Kontext statt.

9.2.1. Bewertung der eingesetzten Methoden & Technologien

Zunächst sollen die beiden Hauptansätze zur Realisierung der semantischen Suche betrachtet werden. Der erste Hauptansatz im Rahmen der Machbarkeitsstudie bestand in der *teilautomatisierten Erstellung* eines zugrundeliegenden semantischen Modells. Ein solches Modell dient zur Abbildung von Domänenwissen und wird als Wissensbasis von jedem semantischen Suchsystem zwingend vorausgesetzt. Die Erkenntnis ist hierbei, dass aufgrund des Knowledge Acquisition Bottleneck eine Teilautomatisierung unabdingbar ist. Eine manuelle Wissensüberführung der komplexen DLR-Domäne ist im gegebenen Kontext nicht realisierbar. Eine manuelle Überführung wäre nur bei einer entsprechenden Einschränkung der Domäne oder einer Abstraktionsniveau-Erhöhung des resultierenden Modells möglich.

Der ausgewählte *Ansatz zur semantischen Suche* bildet den zweiten zentralen Eckpfeiler innerhalb dieser Ausarbeitungen. Der hier verfolgte Ansatz des konzeptbasierten Dokumenten-Retrieval wurde unter Berücksichtigung der Anforderungen mit Hilfe mehrerer Kriterien hergeleitet. Das konzeptbasierte Dokumenten-Retrieval ist eine relativ einfache Form der semantischen Suche. Hier werden keine Schlussfolgerungsverfahren zur Gewinnung von neuem Wissen eingesetzt, sondern leichtgewichtige lexikalische Modelle unterstützend verwendet. In Fall des Elib-Portals diente ein Thesaurus zur Repräsentation des Domänenwissens. Die im Thesaurus-Modell enthaltene Semantik beschränkt sich auf die Abbildung von Konzepten einer Sprache oder eines Wissensgebietes. Das heißt, Symbole werden mit ihrer Bedeutung

auf der Konzeptebene verknüpft (vgl. Abschnitt 3.1). Der Thesaurus ist im betrachteten Anwendungsfall zur Unterstützung der explorativen Suchen eingesetzt worden. Die durch die Konzepte abgebildeten Themengebiete wurden kontextabhängig in der Benutzeroberfläche des Elib-Portals präsentiert. Die Semantik des Thesaurus-Modells wurde somit nicht direkt zur Interpretation der Suchanfrage oder zur Relevanz-Verbesserung genutzt, sondern als Hilfestellungen in der GUI verwendet. Das bedeutet auch, dass die entwickelte semantische Suche die Grundprobleme des schlüsselwortbasierten IR-Ansatzes nicht direkt auf der Ebene der Anfrageverarbeitung oder Suchergebnispräsentation behebt. Mehrdeutigkeiten werden also nicht aufgelöst, sondern dem Nutzer präsentiert. Mit Blick auf die zuvor durchgeführte Evaluation der Suchergebnisqualität kann trotz dieser andersartigen Verwendung festgehalten werden, dass der Ansatz des konzeptbasierten Dokumenten-Retrieval erfolgreich auf den Anwendungsfall übertragen werden konnte. Aufgrund der hierdurch realisierten Benutzerunterstützung, kann zudem von einer Annäherung an das mentale Modell des Benutzers gesprochen werden (vgl. Abbildung 4.2). Allerdings ist herauszustellen, dass die vollen Möglichkeiten semantischer Technologien bei der gewählten Herangehensweise nicht vollends ausgeschöpft werden. Das annotationsbasierte Dokumenten-Retrieval in dem Ontologien und Schlußfolgerungsverfahren eingesetzt werden, wäre hier sicherlich eine Alternative. Aufgrund der Schwierigkeit der automatisierten Ontologie-Erstellung wurde dieser Ansatz jedoch nicht ausgewählt. Durch den Einsatz semantisch reichhaltigerer Ontologien, könnte jedoch nochmals ein höheres Abstraktionsniveau erreicht und dadurch eine weitere Annäherung an das mentale Modell des Nutzers vollzogen werden.

Neben den beiden zuvor beschriebenen Hauptansätzen sind weitere verwendete Methoden und Technologien in Abhängigkeit vom Einsatzzweck zu nennen. Diese sollen im Folgenden einer kritischen Betrachtung unterzogen werden.

- Methoden und Technologien zur Repräsentation der Thesaurus-Inhalte
  - SKOS-Standard und RDF Triple Store
- Methoden und Technologien zur automatisierten Erstellung eines Thesaurus:
  - Extraktion von Schlüsselwörtern
  - Relationsbestimmung
- Methoden und Technologien zur Suche im Thesaurus:
  - Schlüsselwortbasiertes Suchen nach Konzepten

Der *SKOS-Standard* in Kombination mit einem *RDF Triple Store* dienten zur Repräsentation der Thesaurus-Inhalte. Der SKOS-Standard wurde im Rahmen der durchgeführten Erweiterungen nicht im vollen Umfang genutzt. SKOS definiert zwar keine Ontologiesprache in dem hier verstandenen Sinne, nutzt aber einige Sprachkonstrukte von OWL wie z.B. `owl:inverseOf`. Die Triple-Store-Varianten des Sesame Framework bieten nicht die Möglichkeit der Inferenz auf Basis von OWL-Semantik. Aufgrund dieses Mangels war der in Abschnitt 8.3.2 erörterte Workaround zur Ermittlung inverser Konzept-Beziehungen notwendig. Hier wäre ein leistungsfähigerer Triple Store wünschenswert, der das Schlussfolgern auf Basis von OWL-Semantik ermöglicht und somit diesen Workaround erübrigt. Weiterhin ist festzuhalten, dass das Elmo Framework als ORM-Mapper zwischen den RDF-Daten des Triple Store und den Java-Objekten zu einer erheblichen Erleichterung der Entwicklung führte.

Im Rahmen der *Schlüsselwort-Extraktion* wurde das KEA-Framework verwendet. Die hier genutzten maschinellen Lernverfahren ermöglichen die sprachabhängige Erzeugung von Modellen. Mit diesen Modellen sowie mit Hilfe lexikalischer Methoden des IR konnten anschließend erfolgreich Schlüsselwörter aus dem Elib-Datenbestand gewonnen werden. Mit Blick auf die zuvor in Tabelle 9.2 dargestellten Ergebnisse, sind hier jedoch noch erhebliche Verbesserungsmaßnahmen zur Erhöhung der Schlüsselwortabdeckung notwendig. Ziel sollte es sein auch die restlichen 30% aller Elib-Einträge durch dieses Verfahren zu erfassen. Auch ist die Qualität der Extraktionen noch deutlich zu erhöhen. Zwar liegen keine genauen Statistiken vor, jedoch konnte stichprobenhaft festgestellt werden, dass der Anteil unbrauchbarer Schlüsselwörter mit zunehmender Menge steigt. Idealerweise sollten also möglichst nur brauchbare Schlüsselwörter erfasst und gleichzeitig alle Störeinflüsse herausgefiltert werden.

Die *Relationsbestimmung* fand zum einen auf Basis einer Kookkurrenz-Analyse mit Hilfe der Open-Source-Anwendung DISCO und zum anderen mit Hilfe einer einfachen Ähnlichkeitsbestimmung zwischen Zeichenketten statt. Die Nutzung der vorgefertigten DISCO-Datenbanken zur Ermittlung einer Kollokation bietet zwar den Vorteil des geringen Rechenaufwandes, jedoch beruhen diese Datenbanken auf domänenfremden Inhalten. Dies führt zu dem Problem, dass nicht in jedem Fall eine Ähnlichkeitsbestimmung mit Hilfe der Kookkurrenz-Analyse durchgeführt werden kann. Tabelle 9.4 verdeutlicht dieses Problem anhand von Zahlen. Im Durchschnitt sind rund 54% der Thesaurus-Relationen aufgrund einer Kollokation ermittelt worden. Es kann davon ausgegangen werden, dass durch eine Erhöhung des Kollokations-Anteils, die Qualität der Thesaurus-Beziehungen gesteigert werden kann. Gleichzeitig würde auch die Fehleranfälligkeit der Relationsbestimmung verringert werden. Dieses Ziel könnte durch die Erzeugung einer eigenen Kollokations-Datenbank auf Basis der Elib-Dokumente erreicht werden. Weiterhin würde bei der Verwendung einer eigenen Datenbank auch die Quantität der Relationen erhöht.

Tabelle 9.4.: Ursprung der Relationen in Thesaurus Nr. 1 bis 4

Nr.	# Relationen	# Kollokation	# Zeichenkettenvergleich
1.	(100%) 3.821	(54,38%) 2.078	(45,62%) 1.743
2.	(100%) 5.945	(54,08%) 3.215	(45,92%) 2.730
3.	(100%) 9.101	(53,53%) 4.872	(46,47%) 4.229
4.	(100%) 12.135	(53,69%) 6.515	(46,31%) 5.620

Zuletzt sollen die Methoden und Technologien, die im Kontext des *schlüsselwortbasierten Suchens nach Konzepten* eingesetzt wurden, beleuchtet werden. Hier wurde der graphbasierte Ansatz der semantischen Technologien und der Bag-of-Word-Ansatz des IR miteinander vereint. Das hier genutzte syntakische Matching diente dazu, passende Konzepte zu finden. Dieses Vorgehen hat zum einen den Vorteil, dass ein Ranking mit Hilfe von IR-Methoden durchgeführt werden kann. Zum anderen ist ein effizientes Suchen auf Basis des invertierten Indexes möglich. Innerhalb der Anforderungen wurden zwar keine Qualitätsanforderungen bzgl. der Performanz festgelegt, trotzdem profitierte gerade die Autovervollständigungs-Funktion von der Effizienz des invertierten Indexes. Hier wäre interessant zu untersuchen, inwiefern eine reine graphbasierte Lösung ein effizientes Aktualisieren der Autovervollständigung ermöglichen würde. Mit Blick auf das syntaktische Matching ist jedoch auch noch Anpassungsbedarf zu vermerken. Wie im Fall der Suchanfrage „smartphone sensor“ geschildert, liefert dieses Matching bei nicht vorhandenen Thesaurus-Konzepten keine sinnvollen Ergebnisse (siehe Abschnitt 9.1.2). Hier ist die Präfixsuche der Autovervollständigung zu verbessern. Mit Hilfe des semantischen Matching konnten die Verbindungen der gefundenen Konzepte innerhalb des

RDF-Graphen ermittelt werden. Es zeigte sich ebenfalls, dass im Rahmen der Autovervollständigung erheblicher Verbesserungsbedarf besteht. An dieser Stelle fehlt ein graphbasiertes Verfahren zur intelligenten Zusammenführung gefundener Konzepte.

9.2.2. Umsetzbarkeit und Mehrwert

Die Frage der Umsetzbarkeit ist eng mit der Problematik des Knowledge Acquisition Bottleneck verbunden. Diese Problematik war der limitierende Faktor der durchgeführten Machbarkeitsstudie. Die vollständige Abbildung einer Domäne in Form einer Ontologie wäre zwar erstrebenswert, um aufbauend auf dieser reichhaltigen Repräsentationsform eine semantische Suche erstellen zu können. Jedoch wurde im Rahmen der durchgeführten Untersuchungen festgestellt, dass dieses Ziel nicht erreicht werden kann. Die Wissensakquise innerhalb einer komplexen Domäne, wie die des DLR, kann weder von einer Person alleine noch vollständig automatisiert erfolgen. Der mögliche Ansatz einer teilautomatisierten Ontologie-Erstellung wurde aus gleichem Grund verworfen. Der hergeleitete Ansatz des konzeptbasierten Dokumenten-Retrieval auf Basis eines Thesaurus ist demzufolge ein Kompromiss zwischen Nutzen und Machbarkeit.

Bei Betrachtung der Umsetzbarkeit des gewählten Ansatzes, sind wiederum viele Hürden genommen worden. Die Kernproblematik die es zu lösen galt, bestand in der automatisierten Bestimmung von semantischen Ähnlichkeiten zwischen Konzepten, um darauf aufbauend Beziehungen festlegen zu können. In diesem Zusammenhang war es zwar nicht möglich unterschiedliche Verfahren gegeneinander abzuwägen, jedoch kann gesagt werden, dass eine solche Problemstellung immer nur mit einer gewissen Fehleranfälligkeit zu lösen ist. Hier kommt wiederum die von Feigenbaum durchgeführte Einordnung dieses Problems zum tragen: Die Gewinnung von Wissen aus Texten ist (immer noch) eine der größten Herausforderungen der Informatik [Feigenbaum 2003]. Aus diesem Grund ist eine nachträgliche Validierung und Anpassung des erzeugten semantischen Modells unumgänglich. Auch diese nachträgliche Validierung und Anpassung ist wiederum mit erheblichem Aufwand verbunden. Von wem diese Aufgabe durchgeführt werden soll, ist zudem keine einfach zu beantwortende Frage.

In Bezug auf die technische Realisierung der eigentlichen semantischen Suche kann gesagt werden, dass viele ausgereifte Anwendungen und Hilfsmittel vorhanden sind, die eine Umsetzung erheblich vereinfachen. Zu nennen sind beispielsweise graphbasierte Datenbanken, Abbildungsstandards, ORM-Mapper, Reasoner oder RDF-Programmierschnittstellen. Von dieser Arbeit kann jedoch nicht abschließend beantwortet werden, ob die zuvor erläuterten Probleme der Umsetzbarkeit, Gründe dafür sind, dass sich semantische Technologien nicht auf breiter Basis durchgesetzt haben. Diese Probleme stellen aber mit Sicherheit große Hürden dar, die bei der Realisierung einer semantischen Suche zu überwinden sind.

Im Rahmen dieser Betrachtungen konnte eine Verbesserung der Suchergebnisqualität festgestellt werden. Somit ist durch den Einsatz dieser Technologien auch ein gewisser Mehrwert entstanden. Im betrachteten Anwendungsfall besteht der konkrete Mehrwert in der Unterstützung der fachübergreifenden Informationssuche eines Wissenschaftlers. Die entwickelte semantische Suche erleichtert diesen Personen das Auffinden von relevanten Informationen.

10. Zusammenfassung & Ausblick

Die durchgeführten Untersuchungen haben gezeigt, dass die Suchergebnisqualität des KnowledgeFinder durch den Einsatz von semantischen Technologien verbessert werden kann. Hierdurch ist es möglich, die fachübergreifende Informationssuche von DLR-Wissenschaftlern zu unterstützen und zu vereinfachen. Um zu diesem Ergebnis zu gelangen wurde eine Machbarkeitsstudie durchgeführt. Als Anwendungsfall wurde das Elib-Portal des KnowledgeFinder herangezogen. Innerhalb der Studie wurde dieses Portal um semantische Suchverfahren erweitert. Dazu fand zunächst die Auswahl eines geeigneten Ansatzes zur semantischen Suche auf der Basis zuvor festgelegter Anforderungen statt. Im Mittelpunkt dieser Analyse stand die Auswahl eines semantischen Modells zur Abbildung des Domänenwissens. Diese Abwägungen ergaben unter anderem, dass im gegebenen Kontext aufgrund des Knowledge Acquisition Bottleneck die teilautomatisierte Erstellung eines Modells das Mittel der Wahl ist. Weiterhin wurde festgestellt, dass die Wissensabbildung mit Hilfe einer Ontologie nicht zielführend ist. Aufgrund ihrer Aussagemächtigkeit ist die Verwendung dieses Modells zwar erstrebenswert, jedoch kann eine Teilautomatisierung wegen der schwierigen nachgelagerten Ontologie-Validierung nicht durchgeführt werden. Stattdessen wurde der Thesaurus als zugrundeliegendes semantisches Modell ausgewählt. Ausschlaggebend für diese Entscheidung war die hervorragende Eignung des Thesaurus in Bezug auf die Anforderungen und die weniger komplexe nachgelagerten Validierung. Basierend auf diesem Modell wurde das konzeptbasierte Dokumenten-Retrieval als Ansatz zur semantischen Suche hergeleitet. Im weiteren Verlauf der Arbeit wurde eine Schlüsselwortsuche über Dokumente auf der Basis von Thesaurus-Konzepten entwickelt. Die Grundidee des konzipierten Ansatzes besteht darin, die DLR-Themengebiete automatisiert aus den Informationsquellen zu erfassen und im Thesaurus abzubilden. Die semantischen Verknüpfungen zwischen den Themengebieten werden dem Wissenschaftler passend zur Suchanfrage präsentiert. Das entworfene Konzept wurde anschließend auf Grundlage einer prototypischen Implementierung evaluiert. Mit Hilfe von repräsentativen Stichproben konnte hier gezeigt werden, dass die entwickelte semantische Suche die fachübergreifende Recherche von DLR-Wissenschaftlern vereinfacht. Bei dieser Untersuchung wurde zudem festgestellt, dass eine nachgelagerte Anpassung des automatisiert erstellten Thesaurus unabdingbar ist. Die gesteigerte Qualität der angezeigten semantischen Zusammenhänge führt zu einer weiteren Vereinfachung der Informationssuche. Insgesamt ist durch den Einsatz semantischer Technologien ein konkreter Mehrwert entstanden. In Bezug auf die Semantic-Web-Vision von Tim Burners Lee kann somit durchaus von einer Verbesserung der Zusammenarbeit zwischen Mensch und Maschine gesprochen werden.

Die Evaluation ergab zudem, dass das vorgestellte Konzept an einige Stellen noch Verbesserungsbedarf aufweist. Hervorzuheben ist zum einen das Verfahren zur Relationsbestimmung. Für eine effektivere Ermittlung von Beziehungen wäre hier der Einsatz einer Kookkurrenz-Analyse auf Basis von DLR-Daten erstrebenswert. Zum anderen fehlen zur Realisierung des vollen Funktionsumfangs der Autovervollständigung graphbasierte Verfahren zur intelligenten Zusammenführung von Thesaurus-Konzepten. Der Mehrwert semantischer Technologien könnte zudem durch die Hinzunahme von nutzerorientierten Ansätzen noch erhöht werden. So wäre eine adaptive semantische Suche denkbar, in der semantische Beziehungen nicht nur algorithmisch, sondern auch aufgrund von Nutzerverhalten erzeugt oder angepasst werden.

11. Literaturverzeichnis

[**Aitchison et al. 2000**] Aitchison, Jean ; Gilchrist, Alan ; Bawden, David: *Thesaurus construction and use: a practical guide*. 4. London : Aslib IMI, 2000. – ISBN 9780851424460

[**Antoniou und Van Harmelen 2004**] Antoniou, G. ; Van Harmelen, F.: *A Semantic Web Primer*. Cambridge (Massachusetts) : Mit Press, 2004 (Cooperative Information Systems). – ISBN 9780262012102

[**Apache Lucene 2012**] Apache Software Foundation: *Apache Lucene*. 2012. – URL `http://lucene.apache.org/`. – Zugriffsdatum: 26.12.2012

[**Arias et al. 2008**] Arias, Mario ; Cantera, José M. ; Vegas, Jesús ; Fuente, Pablo de la ; Alonso, Jorge C. ; Bernardo, Guido G. ; Llamas, César ; Zubizarreta, Álvaro: Context-Based Personalization for Mobile Web Search. In: *PersDB 2008, 2nd International Workshop on Personalized Access, Profile Management, and Context Awareness*, 2008, S. 33–39

[**Baeza-Yates und Ribeiro-Neto 1999**] Baeza-Yates, Ricardo ; Ribeiro-Neto, Berthier: *Modern information retrieval*. Harlow : Addison-Wesley, 1999. – ISBN 9780201398298

[**Bar-Yossef und Kraus 2011**] Bar-Yossef, Ziv ; Kraus, Naama: Context-sensitive query auto-completion. In: *Proceedings of the 20th international conference on World wide web*. New York, NY, USA : ACM, 2011 (WWW '11), S. 107–116. – ISBN 9781450306324

[**Bedini und Nguyen 2007**] Bedini, Ivan ; Nguyen, Benjamin: Automatic Ontology Generation: State of the Art. In: *PRiSM Laboratory Technical Report. University of Versailles* (2007)

[**Berners-Lee et al. 2001**] Berners-Lee, Tim ; Hendler, James ; Lassila, Ora: The Semantic Web: A New Form of Web Content That Is Meaningful to Computers Will Unleash a Revolution of New Possibilities. In: *Scientific American* May 2001 Issue (2001), S. 29–37

[**Bouzid et al. 2012**] Bouzid, Sara ; Cauvet, Corine ; Pinaton, Jacques: A survey of semantic web standards to representing knowledge in problem solving situations. In: Mahmod, Ramlan (Hrsg.) ; Abdullah, Rusli (Hrsg.) ; Abdullah, Lili N. (Hrsg.) ; Sembok, Tengku Mohd T. (Hrsg.) ; Smeaton, Alan F. (Hrsg.) ; Crestani, Fabio (Hrsg.) ; Doraisamy, Shyamala (Hrsg.) ; Kadir, Rabiah A. (Hrsg.) ; Ismail, Mahamod (Hrsg.): *CAMP*, IEEE, 2012, S. 121–125. – ISBN 9781467310918

[**Breitman et al. 2007**] Breitman, K.K. ; Casanova, M.A. ; Truszkowski, W. ; Aeronautics, United States. N. ; Administration, Space: *Semantic Web: Concepts, Technologies and Applications*. London : Springer, 2007 (Nasa Monographs in Systems and Software Engineering). – ISBN 9781846285813

[**Brickley und Guha 2004**] Brickley, Dan ; Guha, R.V.: *RDF Vocabulary Description Language 1.0: RDF Schema.* 2004. – URL `http://www.w3.org/TR/rdf-schema/`. – Zugriffsdatum: 12.01.2013

[**Cheng und Qu 2009**] Cheng, Gong ; Qu, Yuzhong: Searching Linked Objects with Falcons: Approach, Implementation and Evaluation. In: *Int. J. Semantic Web Inf. Syst.* 5 (2009), Nr. 3, S. 49–70

[**Cimiano 2006**] Cimiano, Philipp: *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications.* Secaucus, NJ, USA : Springer-Verlag New York, 2006. – ISBN 0387306323

[**Cissek 2010**] Cissek, Peter: *Strategische Unternehmensplanung in einer Data Warehouse-Umgebung unterstützt durch ein Wissensmanagementsystem.* Düsseldorf, Carl von Ossietzky Universität Oldenburg, Dissertation, 2010

[**Cockburn 2007**] Cockburn, A.: *Use Cases effektiv erstellen.* Heidelberg : mitp-Verlag, 2007. – ISBN 9783826617966

[**Daconta et al. 2003**] Daconta, M.C. ; Obrst, L.J. ; Smith, K.T.: *The Semantic Web: a guide to the future of XML, Web services, and knowledge management.* Indianapolis : Wiley Pub., 2003 (Programming, software development). – ISBN 9780471432579

[**Dengel 2012**] Dengel, A.: *Semantische Technologien: Grundlagen. Konzepte. Anwendungen.* Heidelberg : Spektrum Akademischer Verlag, 2012. – ISBN 9783827426635

[**DISCO 2013**] Peter Kolb & Prochazkova GbR: *DISCO - semantische Ähnlichkeit zwischen Wörtern abfragen.* 2013. – URL `http://www.linguatools.de/disco/disco.html`. – Zugriffsdatum: 04.02.2013

[**DLR 2013**] Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR): *DLR Portal.* 2013. – URL `http://www.dlr.de`. – Zugriffsdatum: 13.03.2013

[**Domingue et al. 2011**] Domingue, J. ; Fensel, D. ; Hendler, J.A.: *Handbook of Semantic Web Technologies.* Heidelberg : Springer, 2011 (Computer science Bd. 1). – ISBN 9783540929123

[**Elib 2012**] Deutsche Zentrum für Luft- und Raumfahrt e.V.: *Electronic Library.* 2012. – URL `http://elib.dlr.de`. – Zugriffsdatum: 26.12.2012

[**Ellouze et al. 2012**] Ellouze, Nebrasse ; Lammari, Nadira ; Métais, Elisabeth: CITOM: An incremental construction of multilingual topic maps. In: *Data and Knowledge Engineering* 74 (2012), April, S. 46–62

[**Elmo 2008**] Aduna: *Elmo.* 2008. – URL `http://www.openrdf.org/doc/elmo/1.4/`. – Zugriffsdatum: 23.02.2013

[**EPrints 2012**] Universität Southampton (Electronics and Computer Science Department): *GNU EPrints.* 2012. – URL `http://www.eprints.org/`. – Zugriffsdatum: 27.12.2012

[**Euzenat und Shvaiko 2013**] Euzenat, J. ; Shvaiko, P.: Ontology Matching: State of the Art and Future Challenges. In: *Knowledge and Data Engineering, IEEE Transactions on* 25 (2013), jan., Nr. 1, S. 158 –176

[**eyePlorer 2013**] eyePlorer GmbH: *eyePlorer*. 2013. – URL `http://de.vionto.com/show/`. – Zugriffsdatum: 09.01.2013

[**Feigenbaum 2003**] Feigenbaum, E. A.: Some challenges and grand challenges for computational intelligence. In: *Journal of the ACM* 50 (2003), Nr. 1, S. 32–40

[**Ferber 2003**] Ferber, Reginald: *Information Retrieval: Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. Heidelberg : dpunkt-Verl., 2003. – ISBN 3898642135

[**Gruber 1993**] Gruber, Tom: A Translation Approach to Portable Ontologies Specifications. In: *Knowledge Acquisition* 5 (1993), Nr. 2, S. 199–220

[**Grunniger und Lee 2002**] Grunniger, M. ; Lee, J.: Ontology - application and design. In: *Commun. ACM* 45 (2002), Nr. 2, S. 39–41

[**van Harmelen und McGuinness 2004**] Harmelen, Frank van ; McGuinness, Deborah L.: *OWL Web Ontology Language Overview*. 2004. – URL `http://www.w3.org/TR/owl-features/`. – Zugriffsdatum: 12.02.2013

[**Hebeler et al. 2009**] Hebeler, John ; Fisher, Matthew ; Blace, Ryan ; Perez-Lopez, Andrew ; Dean, Mike: *Semantic Web Programming*. Indianapolis : Wiley, 2009. – ISBN 9780470418017

[**Hermans 2008**] Hermans, Jan: *Ontologiebasiertes Information Retrieval für das Wissensmanagement*. Berlin : Logos Verlag Berlin, 2008. – ISBN 3832520635

[**Hildebrand et al. 2007**] Hildebrand, M. ; Ossenbruggen, J.R. van ; Hardman, L.: An analysis of search-based user interaction on the Semantic Web / CWI. 2007. – Forschungsbericht

[**Hitzler et al. 2008**] Hitzler, P. ; Krötzsch, M. ; Rudolph, S. ; Sure, Y.: *Semantic Web: Grundlagen*. London : Springer London, Limited, 2008. – ISBN 9783540339946

[**Hitzler et al. 2012**] Hitzler, Pascal ; Krötzsch, Markus ; Parsia, Bijan ; Patel-Schneider, Peter F. ; Rudolph, Sebastian: *OWL 2 Web Ontology Language Primer (Second Edition)*. 2012. – URL `http://www.w3.org/TR/owl2-primer/`. – Zugriffsdatum: 12.02.2013

[**JSR286 2008**] Java Specification Requests: *JSR 286: Portlet Specification 2.0*. 2008. – URL `http://www.jcp.org/en/jsr/detail?id=286`. – Zugriffsdatum: 14.10.2012

[**Juchmes 2011**] Juchmes, Matthias: *Konzeption und Entwicklung einer facettierten Abfrage-Anwendung am Beispiel der Publikationsdatenbank des Deutschen Zentrums für Luft- und Raumfahrt*. Kaiserslautern, Fachhochschule Kaiserslautern, Diplomarbeit, 2011

[**Kastrinakis und Tzitzikas 2010**] Kastrinakis, Dimitrios ; Tzitzikas, Yannis: Advancing search query autocompletion services with more and better suggestions. In: *Proceedings of the 10th international conference on Web engineering*. Berlin, Heidelberg : Springer-Verlag, 2010 (ICWE'10), S. 35–49. – ISBN 9783642-139109

[**KEA 2013**] Frank, Eibe ; Medelyan, Olena: *Keyword Extraction Algorithm*. 2013. – URL `http://www.nzdl.org/Kea/index.html`. – Zugriffsdatum: 04.02.2013

[**Klyne und Carroll 2004**] Klyne, Graham ; Carroll, Jeremy J.: *Resource Description Framework (RDF): Concepts and Abstract Syntax*. 2004. – URL `http://www.w3.org/TR/rdf-concepts/`. – Zugriffsdatum: 07.01.2013

[**Kolb 2008**] Kolb, Peter: DISCO: A Multilingual Database of Distributionally Similar Words. In: Storrer, Angelika (Hrsg.) ; Geyken, Alexander (Hrsg.) ; Siebert, Alexander (Hrsg.) ; Würzner, Kay-Michael (Hrsg.): *KONVENS 2008 – Ergänzungsband: Textressourcen und lexikalisches Wissen*, 2008, S. 37–44

[**Lauesen 2003**] Lauesen, Søren: Task Descriptions as Functional Requirements. In: *IEEE Software* 20 (2003), Nr. 2, S. 58–65

[**Lewandowski 2005**] Lewandowski, Dirk: *Web Information Retrieval - Technologien zur Informationssuche im Internet*. Wiesbaden : Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis e.V., DGI, 2005. – ISBN 9783925474552

[**Liferay 2012**] Liferay Inc.: *Liferay - Enterprise open source portal and collaboration software*. 2012. – URL `http://www.liferay.com`. – Zugriffsdatum: 26.12.2012

[**Manning et al. 2009**] Manning, C.D. ; Raghavan, P. ; Schütze, H.: *Introduction to Information Retrieval*. Cambridge, England : Cambridge University Press, 2009. – ISBN 9780521865715

[**Meusel 2009**] Meusel, Robert: *Text-Mining for Semi-Automatic Thesaurus Enhancement*, University Mannheim, Diplomarbeit, 2009

[**Miles und Bechhofer 2009**] Miles, Alistair ; Bechhofer, Sean: *SKOS Simple Knowledge Organization System Reference*. 2009. – URL `http://www.w3.org/TR/skos-reference/`. – Zugriffsdatum: 02.02.2013

[**OAI-PMH 2012**] Open Archives Initiative: *The Open Archives Initiative Protocol for Metadata Harvesting*. 2012. – URL `http://www.openarchives.org/OAI/openarchivesprotocol.html`. – Zugriffsdatum: 26.12.2012

[**Pellegrini und Blumauer 2006**] Pellegrini, T. ; Blumauer, A.: *Semantic Web: Wege Zur Vernetzten Wissensgesellschaft*. Heidelberg : Springer, 2006 (X.MEDIA.PRESS Series). – ISBN 9783540293248

[**Pepper und Moore 2010**] Pepper, S. ; Moore, G.: Topic maps. In: *Encyclopedia of Library and Information Sciences,* (2010), S. 1–19

[**Pérez-Agüera et al. 2010**] Pérez-Agüera, José R. ; Arroyo, Javier ; Greenberg, Jane ; Iglesias, Joaquin P. ; Fresno, Victor: Using BM25F for semantic search. In: *Proceedings of the 3rd International Semantic Search Workshop*. New York, NY, USA : ACM, 2010 (SEMSEARCH '10), S. 2:1–2:8. – ISBN 9781450301305

[**Pohl 2008**] Pohl, Klaus: *Requirements Engineering: Grundlagen, Prinzipien,Techniken*. Bd. 2., korrigierte Auflage. 2. Heidelberg : dpunkt.Verlag GmbH, 2008. – ISBN 3898645509

[**Sánchez 2009**] Sánchez, Miriam F.: *Semantically enhanced Information Retrieval: an ontology-based approach*. Madrid, Universidad Autonoma de Madrid, Dissertation, Jan 2009

[**Sesame 2012**] Aduna: *OpenRDF.org: home of Sesame.* 2012. – URL `http://www.openrdf.org/`. – Zugriffsdatum: 23.02.2013

[**SimMetrics 2013**] Chapman, Sam: *SimMetrics.* 2013. – URL `http://sourceforge.net/projects/simmetrics/`. – Zugriffsdatum: 05.02.2013

[**SKOSjs 2013**] Kurz, Thomas: *SKOSjs Homepage.* 2013. – URL `https://github.com/tkurz/skosjs`. – Zugriffsdatum: 06.02.2013

[**Sowa 2000**] Sowa, John F.: Ontology, Metadata, and Semiotics. In: *Proceedings of the Linguistic on Conceptual Structures: Logical Linguistic, and Computational Issues.* London, UK, UK : Springer-Verlag, 2000 (ICCS '00), S. 55–81. – ISBN 354067859X

[**Studer et al. 1998**] Studer, R. ; Benjamins, V. R. ; Fensel, D.: Knowledge Engineering: Principles and Methods. In: *Data and Knowledge Engineering* 25 (1998), Nr. 1-2, S. 161–197

[**Tran und Mika 2012**] Tran, Thanh ; Mika, Peter: *Semantic Search - Systems, Concepts, Methods and the Communities behind It.* 2012. – URL `http://sites.google.com/site/kimducthanh/publication/semsearch-survey.pdf`. – Zugriffsdatum: 29.01.2013

[**Tran et al. 2011**] Tran, Thanh ; Mika, Peter ; Wang, Haofen ; Grobelnik, Marko: SemSearch'11: the 4th semantic search workshop. In: *Proceedings of the 20th international conference companion on World wide web.* New York, NY, USA : ACM, 2011 (WWW '11), S. 315–316. – ISBN 9781450306379

[**Troncy et al. 2011**] Troncy, R. ; Huet, B. ; Schenk, S.: *Multimedia Semantics: Metadata, Analysis and Interaction.* Wiley, 2011. – ISBN 9781119970620

[**Ullrich et al. 2004**] Ullrich, M. ; Maier, A. ; Angele, J.: *Taxonomie, Thesaurus, Topic Map, Ontologie – ein Vergleich.* 2004. – URL `http://www.ullri.ch/download/Ontologien/ttto13.pdf`. – Zugriffsdatum: 04.01.2013

[**Vaadin 2012**] Vaadin Ltd.: *Vaadin.* 2012. – URL `http://www.vaadin.com`. – Zugriffsdatum: 21.11.2012

[**W3C 2012**] W3C: *Semantic Web Homepage.* 2012. – URL `http://www.w3.org/2001/sw/`. – Zugriffsdatum: 14.10.2012

[**Wang et al. 2009**] Wang, Haofen ; Liu, Qiaoling ; Penin, Thomas ; Fu, Linyun ; Zhang, Lei ; Tran, Thanh ; Yu, Yong ; Pan, Yue: Semplore: A scalable IR approach to search the Web of Data. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 7 (2009), Nr. 3, S. 177 – 188

[**Wei et al. 2008**] Wei, W. ; Barnaghi, P.M. ; Bargiela, A.: Search with meanings: an overview of semantic search systems. In: *Int. J. Communications of SIWN* 3 (2008), S. 76–82

[**Witten et al. 1999**] Witten, Ian H. ; Paynter, Gordon W. ; Frank, Eibe ; Gutwin, Carl ; Nevill-Manning, Craig G.: KEA: Practical Automatic Keyphrase Extraction. In: *ACM DL*, ACM, 1999, S. 254–256

A. Anhang

A.1. DVD

Auf der beigefügten DVD befinden sich folgende Inhalte:

- **Exposé der Masterthesis** (expose.pdf)
- **Masterthesis** (thesis.pdf)
- **Literaturquellen**
- **Logfile-Analyse**
- **Thesauri**
- **Quellcode**