Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Technische Informatik
und Kommunikationsnetze
Computer Engineering and
Networks Laboratory

# Strukturierung, Navigation und Suche in semantischen Netzen

**Semesterarbeit SA-2003-24**

**Mathias Lincke** (linckem@student.ethz.ch)
[D-BEPR]

**Eidgenössische Technische Hochschule (ETH)**
Departement Informationstechnologie und Elektrotechnik
Institut für Technische Informatik und Kommunikationsnetze
Zürich, Schweiz

Zürich, 04. Juli 2003

| | |
|---|---|
| Dozent: | Prof. Dr. B. Plattner |
| Betreuer: | P. Flury |

# Zusammenfassung

Das Internet hat in den letzten zehn Jahren für die Informationsbeschaffung bei der Arbeit und im Alltag grosse Bedeutung gewonnen. Gleichzeitig wird es wegen des raschen Wachstums im Informationsumfang zunehmend schwieriger gewünschte Informationen über Suchmaschinen zu finden. Um dieser Schwierigkeit entgegenzuwirken und um durch Erweiterungen des bestehenden Web höheren Nutzen zu ermöglichen, wurde in den letzten drei Jahren von Forschern das Konzept des Semantic Web entwickelt. Wichtigste Grundelemente des Semantic Web sind das vom World Wide Web Consortium (W3C) entwickelte Resource Description Framework (RDF) und das Konzept der Ontologien. Diese beiden Grundelemente ergänzen vorhandene Daten bzw. Datenquellen mit semantischen Metadaten.
In dieser Arbeit wird ein Konzept und ein Tool vorgestellt, welche die Nutzung solcher semantischer Metadaten möglich macht. Nutzung bedeutet hierbei einerseits die Erfassung und Speicherung dieser Daten in einer geeigneten Datenstruktur und andererseits die Datenanalyse, Informationssuche und Visualisierung.
Es wird gezeigt, dass sich zur Speicherung der semantischen Daten am besten eine Graphstruktur aus Knoten und gerichteten Kanten eignet. Die hier vorgestellte Graphstruktur ist eine Erweiterung zu den im Zusammenhang mit RDF üblicherweise benutzten Graphen. Die Erweiterung dient dem Zweck, dass bei einer zukünftigen „semantischen Applikation“ in Form eines „semantischen Browsers“ oder eines „semantischen Informationsagenten“ alle wichtigen Daten einer Webressource erfasst werden können und verfügbar sind. Zu diesen Daten gehören neben den eigentlichen RDF-Daten und den damit verbundenen Ontologien auch „Keywords“ und „Konzentrate“ von Information.
Als Abschluss und zur Überprüfung der konzeptionellen Arbeit wurde ein Tool namens GRASP (*Graph for the Analysis of Semantically enriched Polymorphous data*) entwickelt. Dieses in der Sprache Python programmierte Tool besitzt einfache Funktionen zur Nutzung von semantischen Metadaten. So können Daten in RDF-„Notation 3“-Syntax eingelesen, Suchoperationen und einfache Analyseoperationen durchgeführt und Output zwecks Visualisierung der Daten in einem fremden Tool generiert werden.

# Abstract

The internet has become in the last decade a very important tool for information retrieval at work as well as at leisure time. At the mean time it became increasingly difficult to find information using search machines, because the internet has grown rapidly. To fight this difficulty and to allow a higher level of service through the extension of the existing web the concept of the Semantic Web has been developed in the last three years. The core elements of the Semantic Web are the Resource Description Framework (RDF) developed by the World Wide Web Consortium (W3C) and the concept of ontologies. Those two core elements add semantic metadata to existing data or resources of data.
In this report a concept and a tool are presented, that make the use of such semantic metadata possible. Use in the sense of collecting and storing the data in a suitable data-structure on the one hand and the analysis, information retrieval and visualization on the other hand.
It will be shown, that a graph-structure consisting of nodes and directed edges is best for the storage of the semantic data. The graph-structure which is presented in this work is an extension to graphs frequently used to display examples of RDF-data. The extension aims

at providing a future “semantic application”, i.e. a “semantic browser” or “semantic information-agent”, the possibility to collect and to make available all important information of a webresource. “Keywords” and “Konzentrate” are part of this data besides RDF-data and the ontologies linked to it.
As completion of and in order to check the validity of the conceptual work a tool named GRASP (Graph for the Analysis of Semantically enriched Polymorphous data) was developed. The tool is written in the programming language Python and it has several simple functions to handle metadata. Data in “notation 3”-RDF-syntax can be put into the tool, search operations and statistic analysis can be done and finally an output can be generated for the purpose of visualization through another tool.

# Aufgabenstellung

Der vollständige Text der originalen Aufgabenstellung für die Semesterarbeit findet sich im Anhang. Der wichtigste Abschnitt ist hier jedoch zusätzlich zitiert.
„*Wie der Titel der Aufgabenstellung bereits andeutet, setzt sich diese Arbeit mit der Strukturierung von semantischen Netzen und mit der Suche und Navigation innerhalb solch strukturierter Netze auseinander.*
*In dieser Arbeit soll ein Model entwickelt werden, um Ressourcen mittels RDF Metadaten miteinander logisch zu verknüpfen. Der Student soll Konzepte entwickeln, um diese Verknüpfungen vorzunehmen. Dabei sollen semantische Graphen, vermutlich werden es Bäume sein, entstehen, durch die man navigieren und suchen kann.*
*Eine Visualisierung der Graphen soll je nach Aufwand ebenfalls umgesetzt werden. Die Visualisierung soll über ein Webinterface erfolgen können.*
*Die semantischen Graphen sollen sich vorerst auf eine einzige Ontologie (Wörterbuch) beziehen. Es sollen dennoch Überlegungen angestellt werden, um diese auf mehrere Ontologien zu erweitern.*
*Als letzte Aufgabe soll ein Interface in Form einer einfachen Suchmaschine implementiert werden. Dieses soll es ermöglichen, durch den semantischen Graphen zu navigieren und einfache Anfragen zu stellen. Dabei sollen die ’bidirektionalen’ Verknüpfungen der Ressourcen ausgeschöpft werden.*
*Die entwickelten Konzepte sollen in einer Implementation umgesetzt werden. Dabei nimmt der Student die Assoziation von RDF Metadaten und Ressourcen manuell vor. Es wird davon ausgegangen, dass die Ressourcen physikalisch verteilt sein werden und nur zum Teil über Links miteinander verknüpft sein werden. Ob das System den vollständigen Ressourcenpool für die Versuche kennen muss, ist Gegenstand der vorangegangenen konzeptionellen Arbeit.“*

# Einleitung

Der vorliegende Bericht ist in drei Hauptteile gegliedert.
Der Teil **Situationsanalyse** gibt dem Leser einen Eindruck zum thematischen Rahmen, in welchem die vorliegende Arbeit liegt. Probleme der heutigen Datenwelt des Internet werden aufgezeigt und das visionäre Konzept des Semantic Web wird vorgestellt. Die für diese Arbeit zentralen Begriffe RDF, Ontologie und Information werden gründlich erklärt. Weiter werden

bereits existierende Arbeiten im gleichen Themenumfeld erörtert und die Chancen zur Realisierung des Semantic Web beurteilt. Dieser erste Teil ist nur begrenzt technisch und behandelt stattdessen Fragen nach dem gesellschaftlichen und ökonomischen Nutzen und den Chancen des Semantic Web. Am Ende des ersten Teils entsteht so ein klares Bild zur Motivation und Problemstellung der vorliegenden Arbeit.
Der zweite Teil **Systembeschreibung und Lösungskonzept** ist die konzeptionelle Vorarbeit zur technischen Lösung. Zuerst werden der Untersuchungsgegenstand und dessen Systemumfeld genau beschrieben. Es wird weiter kurz beschrieben wie die eingelesenen Daten bezüglich ihrer Interkonnektivität aussehen werden. Die anschliessenden drei Abschnitte Datenmodell, Grundelemente und Verbindungen in Erweiterung zum RDF-Standard beschreiben die zentralen konzeptionellen Punkte dieser Arbeit. Es wird darin beschrieben wie die eingelesenen Daten modellhaft erfasst werden, welche Eigenschaften sie haben können und wie sie in Beziehung zu einander stehen können. Im abschliessenden Abschnitt werden mögliche Suchoperationen aufgelistet.
Der dritte Hauptteil befasst sich mit der **Technischen Lösung**. Zuerst werden die Anforderungen und Ziele des Tools beschrieben. Anschliessend folgen technische Betrachtungen zum zu erwartenden Rechenaufwand und der Interkonnektivität der Daten. Dann wird die eigentliche Datenstruktur und implementierte Funktionen vorgestellt. Es folgt ein Datenbeispiel, welches als Testfall verwendet wurde. Eine Evaluation bildet den Abschluss des Kapitels.
Im Anschluss an die drei Hauptteile findet sich ein abschliessender Ausblick und ein Anhang mit Verweisen auf verschiedene wichtige Informationsquellen

# Situationsanalyse

## *Die heutige Datenwelt – Situation und Trends*

„Das Informationszeitalter hat begonnen.“ Und: „Wir leben in einer Wissensgesellschaft.“ So oder ähnlich beschreiben in regelmässigen Abständen Wissenschaftler und Journalisten die Besonderheit unserer Zeit und Gesellschaft. Zwei Aspekte treten dabei in den Vordergrund: einerseits der technologische Aspekt, bei welchem beruhend auf den technischen Entwicklungen im Feld der Datenverarbeitung und Telekommunikation ein neuer wirtschaftlich-technologischer Zyklus eingeleitet wurde (vgl. dazu die berühmten Kondratieff-Zyklen), und andererseits ein gesellschaftlicher Aspekt, welcher den grundlegenden Wandel betont, den die Abwendung von der klassischen Industrie und die zunehmende Verfügbarkeit und Wichtigkeit von Informationen und Wissen für die Menschen mit sich bringt.
Betrachtet man die letzten zehn Jahre kann man wahrhaftig von revolutionären Umwälzungen sprechen, die das Internet und zunehmend auch die Mobilkommunikation in den entwickelten Ländern verursachte. Diese technischen Entwicklungen hatten und haben weltweit immens grossen Einfluss auf die Wirtschaft, die Kommunikation, die Informationsbeschaffung und den Alltag ganz allgemein.
Ein Massstab für die sich fortsetzende Entwicklung ist hierbei die wachsende Zahl registrierter Webpages (1998: ca. 4 Mio., 2001: ca. 28 Mio.) und von Internetusern (1998: 140 Mio. 2001: ca. 510 Mio.) (vgl. [d1]) Es ist zu erwarten, dass sich dieser Trend in den nächsten Jahren fortsetzt, da die Verfügbarkeit von ans Netz angeschlossenen Computern (auch in ärmeren Ländern) zunimmt. Weiter zeichnet sich ab, dass in den entwickelten Ländern der Zugang zum Netz vermehrt über mobile Geräte wie Handys und PDAs erfolgen wird und dass neuere Anwendungen wie vernetzte Navigationshilfen, Fernüberwachung oder –steuerung zu einer Erweiterung des Internets führen werden.

| Jahr | Zahl der Internetuser (Mio.) | Zahl der Domains (Mio.) |
|---|---|---|
| 2001 | 510 | 125 |
| 2000 | 370 | 93 |
| 1999 | 200 | 56 |
| 1998 | 150 | 36 |
| 1997 | 80 | 19 |

Table 1: Wachstum des Internet, Quelle: http://www.netvalley.com/intvalstat.html

Ein grosses Problem verursacht durch den rasanten Wachstums an auf dem Internet verfügbarer Information ist die Schwierigkeit diese zu finden. Die heute üblichen Suchmaschinen wie Google, Yahoo etc. decken mit ihren Indizes nur noch schätzungsweise 10 – 30 % des im Web verfügbaren Inhalts ab (vgl. [d2]). Dazu kommt hinzu, dass die referenzierten Inhalte nicht aktuell sind, da die Indexierung in zeitlichen Abständen von einigen Wochen bis Monaten erfolgt. Das Problem wird sich weiter zuspitzen und so das förderliche Potenzial schmälern, welches weltweit verfügbare Information zu allen erdenklichen Themen bietet.
Eine andere Seite des Problems riesiger Datenmengen findet man bei den Suchresultaten und der Suchmethodik. Ist die textbasierte Suche zu allgemein so liefern die Suchmaschinen kaum bewältigbare Mengen an Resultaten, aus denen sich nur schwer die gesuchte Information rausfiltern lässt. Andererseits ist es für den Benutzer schwierig seine Suche so spezifisch zu formulieren, dass er genau die Information findet, die er sucht, über die er aber gleichzeitig nichts genaues weiss, da er sie sonst ja nicht suchen würde. (Ein interessanter Ansatz Suchresultate maschinell nach „Themen" zu strukturieren findet sich bei Vivisimo, vgl. [d3].)
Bei der Diskussion der Problematik des explosiv wachsenden Web wird offensichtlich, dass dessen charakteristische Merkmale der geringen Hierarchisierung und der viel zitierten „Freiheit" mit fortschreitendem Alter zur Schwäche werden. Die Offenheit des Web fördert die Schaffung vieler Webpages und den Zugang unterschiedlichster Leute, verunmöglicht aber gleichzeitig eine Strukturierung der Inhalte, welche einen raschen und zielgerichteten Zugriff ermöglichen würde.
Der nächste Absatz zeigt mit welchen Ansätzen die Verfechter des sogenannten „Semantic Web" diesen Problemen entgegentreten wollen und wie sie die Zukunft des Internets sehen.

## *Die Vision des Semantic Web und deren Kernelemente*

*"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. … The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users."* Tim Berners-Lee, "The Semantic Web" (2001), [1]

Tim Berners Lee, der Begründer des World Wide Web, vertritt prononciert die Meinung, dass sich das Web zu einem Semantic Web wandeln muss, um in Zukunft maximalen Nutzen daraus gewinnen zu können. Der Begriff „Semantic" bezieht sich auf eine zusätzliche „semantische Schicht", welche den in Webdokumenten enthaltenen Informationen maschinenlesbare „Bedeutungs-Daten" oder anders gesagt „beschreibende Metadaten" beifügt (englisch: *Annotation of Metadata*). Diese Metadaten werden sodann mittels sogenannter „Ontologien" wiederum beschrieben bzw. in einen strukturierten, grösseren Kontext gestellt. Zu einem späteren Zeitpunkt sollen „intelligente Agenten" das Semantic Web durchstreifen und entsprechend ihren Funktionen menschlichen Nutzern oder anderen Agenten ihre Dienste zur Verfügung stellen.

In Bezug auf den Charakter der Webseiten soll ein eigentlicher Paradigmenwechsel stattfinden: Heute ist das Web eine Sammlung von Dokumenten, die mittels Browser von Menschen betrachtet werden können und als Nebenfunktion von Maschinen nach Text und Stichwörtern durchsucht werden können. Es steht folglich die „primitive“ grafische Anzeige von Information für den Menschen im Vordergrund - Informationen die in einem riesigen, ungeordneten elektronischen Kataloges gespeichert sind. Im Semantic Web soll die Information zusätzlich maschinell lesbar und begrenzt auch in ihrer Bedeutung für Maschinen „verständlich“ sein. Die Verarbeitung von Information sei es durch Maschinen oder sei es für die visuelle Anzeige für den Menschen steht dann im Vordergrund.
Die untenstehende Grafik gibt einen groben Überblick über die Komponenten des Semantic Web.

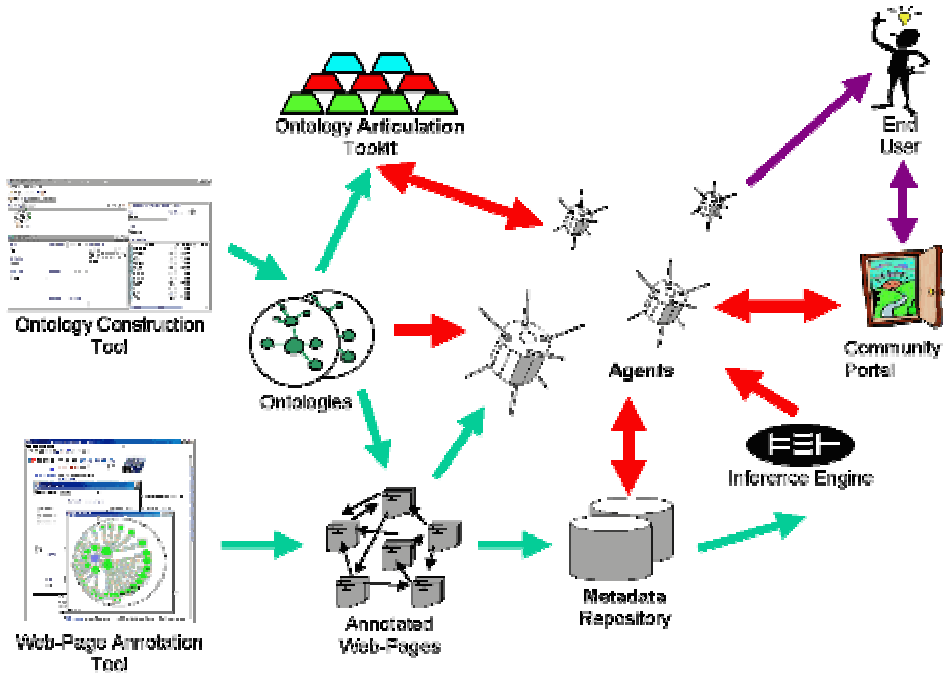


Figure 1: The Big Picture (of the Semantic Web), Quelle: www.semanticweb.org

In den nachfolgenden Abschnitten werden die beiden Kernelemente Ontologien und RDF-Metadaten genau erklärt. Die anderen Komponenten wie Agenten oder „Community Portal“ werden nicht genauer beschrieben.

## *RDF-Metadaten*

RDF steht für Resource Description Framework und ist ein vom W3C (World Wide Web Consortium) entwickelter Standard, um Daten-Ressourcen zu beschreiben. Da mit einer solchen Beschreibung Daten über Daten angelegt werden, spricht man häufig von Metadaten. Eine Ressource ist gemäss Spezifikation (vgl. [a3]) jegliche Art von Objekt, welches durch eine URI (Universal Resource Identifier) eindeutig beschreibbar ist. Standardmässig ist dies eine Webseite (Bsp. http://www.example.org), es kann aber auch ein Teil (*fragment*) einer Webseite, irgendein Textdokument, eine Grafik oder beispielsweise ein Audiofile sein. Wenn klar ist, welche

Ressource gemeint ist, wird in einer abgekürzten Schreibweise wird häufig nur der letzte Teil einer URI benutzt. Der Adressteil vor dem mit einem *crosshatch* (#) beginnenden sogenannten *fragment-identifier* wird bei Bedarf wieder hinzugefügt. Der Einfachheit halber und dem Thema der Arbeit entsprechend wird nachfolgend vorwiegend von Webpages gesprochen, wenn von Ressourcen die Rede ist.
Ein RDF-statement hat immer die Form eines Triples zusammengesetzt aus Subjekt, Prädikat und Objekt. Das Subjekt und das Objekt sind Ressourcen, wie sie oben beschrieben wurden, wobei das Objekt zusätzlich auch einfach ein Literal, sprich ein Textstring, sein kann. Das Prädikat drückt aus wie das Subjekt durch das Objekt beschrieben wird. Anders gesagt wird das Subjekt durch ein Objekt beschrieben, dessen Eigenschaften durch das Prädikat festgelegt sind. Schematisch können diese Triples einer Art dargestellt werden wie sie weiter unten einsehbar ist. Ressourcen werden durch Ovale, Literale durch Rechtecke und Prädikate durch verbindende Pfeile von Subjekt nach Objekt dargestellt. Im vorliegenden Beispiel wird gesagt, dass #Grasp.html (=Subjekt) den #Creator (=Prädikat) #Lincke.html (=Objekt) hat und den #Title (=Prädikat) „GRASP – Strukturierung von semantischen Daten“ (=Objekt) besitzt.

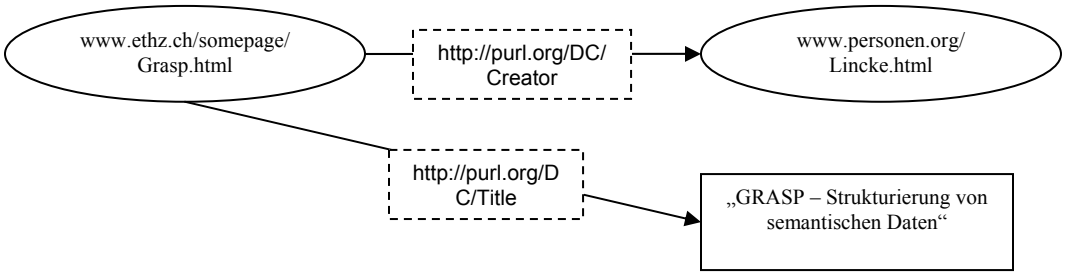


Figure 2: Schematische Darstellung eines einfachen RDF Beispiels

Dabei wird nun ersichtlich, dass auch das Prädikat ein URI besitzt. Dieser URI verweist auf einen Abschnitt einer Ontologie, in welchem der entsprechende Begriff definiert ist. (Ontologien werden im nachfolgenden Abschnitt beschrieben.)
Im Kontext einer Webpage wird RDF syntaktisch auf einfache Weise in XML integriert (XML ist die Nachfolgersprache von HTML). Das obige Beispiel sieht dann folgendermassen aus:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/DC/"
  <rdf:Description about="http://www.ethz.ch/somepage/grasp.html">
    <dc:Creator rdf:resource="www.personen.org/Lincke.html"/></dc:Creator>
    <dc:Title> GRASP – Strukturierung von semantischen Daten </dc:Title>
  </rdf:Description>
</rdf:RDF>
```

Die zweite und die dritte Zeile teilen dem interpretierenden Programm mit, dass ein RDF-Absatz folgt und bei welchen Webadressen die Ontologien zu den benutzten semantischen Begriffe (Creator, Title etc.) zu finden sind. In spitze Klammern eingeschlossene Tags mit Start- und Endmarkierungen werden benutzt.
Die Einfachheit der Syntax von RDF (subject, predicate und object) ist flexibel und mächtig. Diese Syntax erlaubt es sehr komplexe und individuell sehr unterschiedliche Zusammenhänge zu beschreiben. Gleichzeitig erzwingt der Bezug auf eine Ontologie die Benutzung oder Schaffung eines strukturierten Vokabulars. Es wird also eine Ressource nicht nur durch einen Titel beschrieben, sondern gleichzeitig wird darauf verwiesen wo eine Beschreibung des Begriffs

„Titel“ zu finden sei. Hierin liegt ein massgeblicher Unterschied von RDF zu den Metadaten, welche bereits in HTML vorgesehen sind. (Jene Metadaten können willkürlich gesetzt werden und sind deshalb weniger aussagekräftig.) Auf im Internet verfügbare Tutorials zu RDF wird im Anhang verwiesen.

## *Ontologien*

Das Wort Ontologie stammt ursprünglich aus der Philosophie. *Ontos* bedeutet auf Griechisch „sein“. Ontologie kann daher als „Lehre vom Sein“ bezeichnet werden. Dessen technische Bedeutung, welche erst später hinzukam, wird in diesem Abschnitt erläutert.
Im Rahmen des Semantic Web ist eine Ontologie ein Vokabular von Begriffen, das gebraucht wird, um einen Wissensbereich zu beschreiben. Die Begriffe werden typischerweise hierarchisch strukturiert und mittels logischer Ausdrücke zueinander in Beziehung gesetzt.
Eine Ontologie, welche im Zusammenhang mit e-commerce benutzt wird, würde zum Beispiel die Begriffe Kunde, Produkt, Transaktion, Kauf etc. definieren. Ein Stammkunde wäre in einem solchen Fall eine Unterkategorie von Kunde und Produkte könnten sich beispielsweise in Software und Hardware aufspalten. Auf einer mit Metadaten versehenen Website wäre dann ein reales Produkt X sozusagen als Instanz der Klasse Produkt der Ontologie Y beschrieben.
Eine Ontologie kann im Web sinnvollerweise nur verwendet werden, wenn sie an einer eindeutigen Adresse maschinenlesbar zugreifbar ist. (Hinweis: Synonym zu *ontology* werden im Englischen auch die Begriffe *namespace document*, *schema* oder etwas allgemeiner *set of (RDF-)properties* oder *(RDF-)vocabulary* verwendet.)
Prinzipiell beschreiben Ontologien reale oder abstrakte Dinge in ähnlicher Weise wie die natürliche menschliche Sprache. Im Gegensatz zu den natürlichen Sprachen sind Ontologien künstlich von Fachleuten erstellt um ein Themengebiet sachlich prägnant, logisch exakt und meist maschinell lesbar zu beschreiben. Der Nutzen von Ontologien liegt (ähnlich wie bei der objektorientierten Programmierung) vor allem in der Wiederverwendbarkeit, der Erweiterbarkeit und der Möglichkeit der Nutzung durch verschiedene Parteien.
Die erhöhte Struktur und Exaktheit, welche Ontologien in die Informationswelt des Semantic Web bringen sollen, wird dadurch abgeschwächt, dass es jedermann erlaubt ist seine Ontologie zu definieren und verfügbar zu machen. Dem dezentralen und freiheitlichen Prinzip des Internets wird also nicht abgeschworen. Dadurch kann es gut sein, dass zum gleichen Thema mehrere Ontologien existieren (und zusätzlich auch in mehreren Sprachen).
Als Beispiel könnte eine Ontologie die Begriffe Hochschule, Dozent, Student, Doktorand und Studentenvereinigung verwenden und eine andere Ontologie die Begriffe Universität, Professor, Diplomstudierender, Nachdiplomstudierender und Studentenorganisation. Dadurch entsteht ein sogenanntes Mapping-Problem (vgl. [4.1]): Ähnlichkeiten, Übereinstimmungen oder ganz allgemein die Relationen zwischen den Begriffen der verschiedenen Ontologien sind nur schwer herzustellen - besonders unter der Annahme, dass dies automatisch durch ein Computerprogramm geschehen sollte. Dieses Problem steht in Analogie zu der weiterhin vorhandenen Schwierigkeit mit Übersetzungsprogrammen Texte kontextgetreu in eine andere Sprache zu übersetzen.
Eine grobe Unterscheidung von zwei Arten von Ontologien kann nach dem Kriterium erfolgen, ob eine Ontologie spezifisch für eine Anwendung oder Fachgebiet geschaffen wurde oder ob sie sich an die natürliche Sprache anlehnt und damit etwas allgemeiner bleibt. Als Beispiel für eine Ontologie der zweiten Art sei auf [c2] verwiesen, wo mit Ontosaurus ein Vokabular von über 70'000 Begriffen der englischen Sprache in seiner Struktur einsehbar ist.
Das Konzept der Ontologien existierte schon bevor die Idee des Semantic Web aufkam. Es wurde beispielsweise bereits im symbolistischen Zweig der Forschung zur Künstlichen Intelligenz

verwendet. Ontologien sind generell gesagt ein Konzept zur Informationsverarbeitung bzw. zur Informationsspeicherung.

## *Der Informationsbegriff*

In diesem Abschnitt soll der in diesem Bericht vielfach benutzte Begriff der Information formal genauer beschrieben werden. Damit verknüpft ist ebenfalls der Begriff der Semantik.
In der Charakterisierung von Information spricht man häufig von den drei Dimensionen oder Aspekten der Information. (vgl. dazu Lyre [2])

- Die **Syntax** betrifft das Auftreten einzelner Informationseinheiten und ihre Beziehung untereinander.
- Die **Semantik** betrifft die Bedeutung der Informationseinheiten und ihrer Beziehungen untereinander.
- Die **Pragmatik** betrifft die Wirkung der Informationseinheiten und ihrer Beziehungen untereinander.

Interessant ist dabei die Verschränkung des semantischen Aspekts mit dem pragmatischen. Das (semantische) Verstehen von Information hat in seiner Wirkung (pragmatisch) immer die Erzeugung neuer Information zur Folge. (Beispiel: Person A fordert Person B auf das Fenster zu schliessen. B weiss in der Folge, dass A das Fenster geschlossen haben möchte.)
Weiter gilt, dass Semantik oder Bedeutung immer nur im Bezug auf bereits bestehende Semantik hergestellt werden kann. (vgl. [2]) (Ein Begriff wird mit anderen Begriffen erklärt, welche bereits verstanden sein müssen.)
Von informationsverarbeitenden technischen Systemen und von semantischen Daten zu sprechen ist darum etwas irreführend, da erstens Programme und Maschinen weit davon entfernt sind Intelligenz zu besitzen und da zweitens Semantik nicht von Information getrennt werden kann. Der Begriff der Semantik im Semantic Web rührt vermutlich daher, dass ein RDF-Prädikat im Kontext einer Ontologie genauer sozusagen „semantisch“ erklärt wird.
Die nachfolgende Abbildung zeigt eine Unterscheidung zwischen Daten, Information, Wissen und Weisheit. Diese Unterscheidet findet sich bei Bellinger (vgl. [d13]).

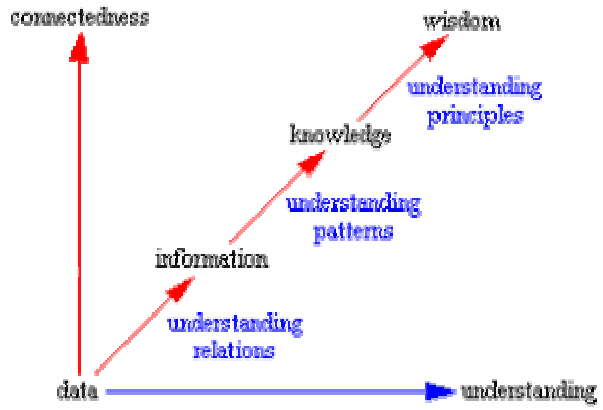


Figure 3: Daten, Information, Wissen und Weisheit

In einem technischen System kommen grundsätzlich nur Daten vor. Diese sind sozusagen neutral ohne Wertung und Bedeutung. Im Hinblick darauf wie der Mensch diese als Information

verstehen kann, darf jedoch auch von Information beispielsweise Information im Internet gesprochen werden.

## *Existierende Lösungen und Aktivitäten im Bereich des Semantic Web*

Es fällt auf, dass in Verbindung mit dem Semantic Web und dem Konzept von Ontologien Stimmen zu vernehmen sind, welche durch ihren Enthusiasmus an die Interneteuphorie und die geplatzte Dotcom-Blase Ende der neunziger Jahre erinnern. Analysten und Berater sprechen von einer „fundamentalen Triebkraft“ und „unaufhaltbaren nächsten Welle“. Sie bezeichnen Ontologie-Engineering als “a core knowledge modelling activity that will have definitive impact on a wide range of enterprise applications and knowledge integration in the next few years” [d10]. Weiter wird gesagt: “semantic technologies have the power to revolutionize the IT world”, und man spricht vom “hottest IT topic” [d9]. In diesem Abschnitt soll erörtert werden, welche Lösungen bereits realisiert wurden und an welchen Problemen und in welchen Bereichen konkret gearbeitet wird.
In der nicht-akademischen Welt konzentriert sich die Aufmerksamkeit auf die Frage wie mit dem semantischen oder ontologischen Ansatz Probleme gelöst, Geld gespart oder mehr Leistung erbracht werden kann. Gerade im Bereich des in den letzten Jahren vieldiskutierten Wissensmanagement haben Metadaten und Ontologien durchaus ihre Daseinsberechtigung. Man denke dabei beispielsweise an die Schwierigkeit die Unmengen an Dokumenten in einem Unternehmen sinnvoll zu administrieren oder das implizite Wissen der Mitarbeiter nutzbar zu machen. Die grosse Zahl gewichtiger Unternehmen, welche eine EU-Initiave (siehe [a2]) zur Förderung der Entwicklung und Anwendung von semantischen Technologien unterstützen, illustriert das Interesse für diese Technologien und wohl auch die Hoffnung, dass damit existierende Probleme gelöst werden können. Eine flüchtige Recherche zu durchgeführten, kommerziell orientierten Projekten zeigt jedoch, dass die realisierten Anwendungen weit davon entfernt sind revolutionär zu sein. Viel eher finden sich darunter Ansätze, welche den klassischen Versuchen zur Schaffung von Expertensystemen gleichen: ein wissensbasiertes System, um im Personalbereich geeignete Kandidaten für Aufgaben oder offene Stellen zu identifizieren, eine technische Datenbank, welche das Wissen von Experten für andere Mitarbeiter des Unternehmens verfügbar macht und eine Applikation zur inhaltsspezifischen Suche nach Dokumenten im Intranet einer Firma.
Einen guten generellen Überblick über bereits existierende Systeme und Lösungen finden sich auf der Ontoweb-Webseite der oben bereits erwähnten EU-Initiave [a2]. Die regelmässig aktualisierte „Technical Roadmap“ hält mit breitem Blickfeld die Entwicklungen im Feld der semantischen Technologien und Systeme fest. Es gibt fünf Hauptkategorien (in Klammern die Anzahl Einträge):

- Methods and methodologies for building ontologies (22)
- Ontology tools (59)
- Languages for building ontologies (17)
- Ontology-based applications (40)
- Semantic web services (9)

Die Kategorie “Ontology tools” (59) kann weiter aufgespalten werden:

- Environments for building ontologies (25)
- Tools for the merging and integration of ontologies (3)

- Ontology-based annotation tools (5)
- Ontology learning tools (7)
- Ontology evaluation tools (4)
- Ontology storage and querying tools (15)

Unter der Hauptkategorie „Ontology-based applications" (40) finden sich:

- Knowledge management (9)
- E-commerce (3)
- Natural language processing (13)
- Intelligent integration of information (3)
- Information retrieval (4)
- Semantic web portals and web communities (5)
- Education (3)

In der Kategorie „Ontology storage and querying tools" gibt es einige Arbeiten bzw. Tools, welche dem hier vorgelegten Tool ähnlich sind. Viele Arbeiten beschränken sich auf den Umgang mit Ontologien, einige jedoch schliessen wie hier in dieser Arbeit vorgesehen die semantisch annotierten Dokumente mit ein.
Auffallend ist bei der Durchsicht der verschiedenen Tools die häufige Referenz auf Datenbankenfunktionalitäten. Bei der Speicherung von RDF-Triples oder anderen Metadaten wie auch bei (SQL-ähnlichen) Suchanfragen bestehen Analogien zu den Datenbanken. Es kommt auch zum Ausdruck, dass RDF nicht der alleinige und unbestrittene Standard zur Erfassung von Metadaten ist. Ähnlich ausgerichtete Standards wie die Topic Maps oder formalisierte Sprachen zur Darstellung von logischen Zusammenhängen tauchen mehrfach auf. Kein fertiger „semantischer Browser" wurde gefunden.
Vier Arbeiten, welche die grösste Ähnlichkeit mit dem hier entwickelten Tool haben, werden hier herausgegriffen [a2]:

- Corese is a semantic search engine, compatible with RDF. Corese allows information retrieval from a set of documents semantically annotated by RDF annotations, relying on the conceptual vocabulary specified in an ontology. Several terms can be associated to a given concept (for example, according ot the language). Production rules can also be associated with the ontology, in order to complete the annotations. The Corese platform implements an RDF/RDFS processor based on Conceptual Graphs (CG). It enables the processing of RDF Schemas and RDF statements within the CG formalism. The graph matching algorithm, called projection, enables to retrieve RDF statements according to a query and hence implements a search engine. The projection operation takes advantage of the class and property type hierarchies. The engine exploits the ontology for information retrieval: desambiguisation, reasoning using the rules, ontology-guided query interface.

  [ http://www.inria.fr/acacia/soft/corese.html ]

- The ICS-FORTH RDFSuite, partially supported by EU projects C-Web (IST-1999-13479) and MesMuses (IST-2001- 26074), is a suite of tools for RDF metadata management, addressing the need of RDF metadata processing for large-scale Web-based applications. It consists of tools for parsing, validating, storing and querying RDF descriptions, namely the Validating RDF Parser (VRP), the RDF Schema Specific DataBase (RSSDB) and the RDF Query Language (RQL).

  [ http://139.91.183.30:9090/RDF/index.html ]

- The Inkling query engine was developed under partial funding from the Harmony and Imesh projects at the ILRT (Institute for Learning and Research Technology, University of Bristol) and can be used to create, query and display RDF documents. It is a Java? implementation of SquishQL created to be API and database-independent for testing the usefulness of SquishQL for comparatively small-scale projects. To ensure the validity of the input RDF data, it uses and upgrades the SiRPAC parser. Inkling can be used with almost any RDF database implementation written in Java (either in-memory or using some persistent storage) and uses the JDBC interfaces to make SquishQL queries.

[ http://swordfish.rdfweb.org/rdfquery/ ]

- Developed by the Hewlett-Packard Company, Jena is a collection of RDF tools written in Java that includes: a Java model/graph API, an RDF Parser (supporting an N-Triples filter), a query system based on RDQL, support classes for DAML+OIL ontologies and persistent/in-memory storage on BerkeleyDB or various other storage implementations. Due to its storage abstraction, Jena enables new storage subsystems to be integrated. To facilitate querying, Jena provides statement-centric methods for manipulating an RDF model as a set of RDF triples and resource-centric methods for manipulating an RDF model as a set of resources with properties, as well as built-in support for RDF containers. The current toolkit does not provide any inferencing mechanisms, since the query language used, i.e., RDQL, does not provide inference. [ http://www.hpl.hp.com/semweb/jena-top.html ]

Die nachfolgende Abbildung zeigt eine Aufteilung in verschiedene sich überlappende Forschungs- oder Themenbereiche:

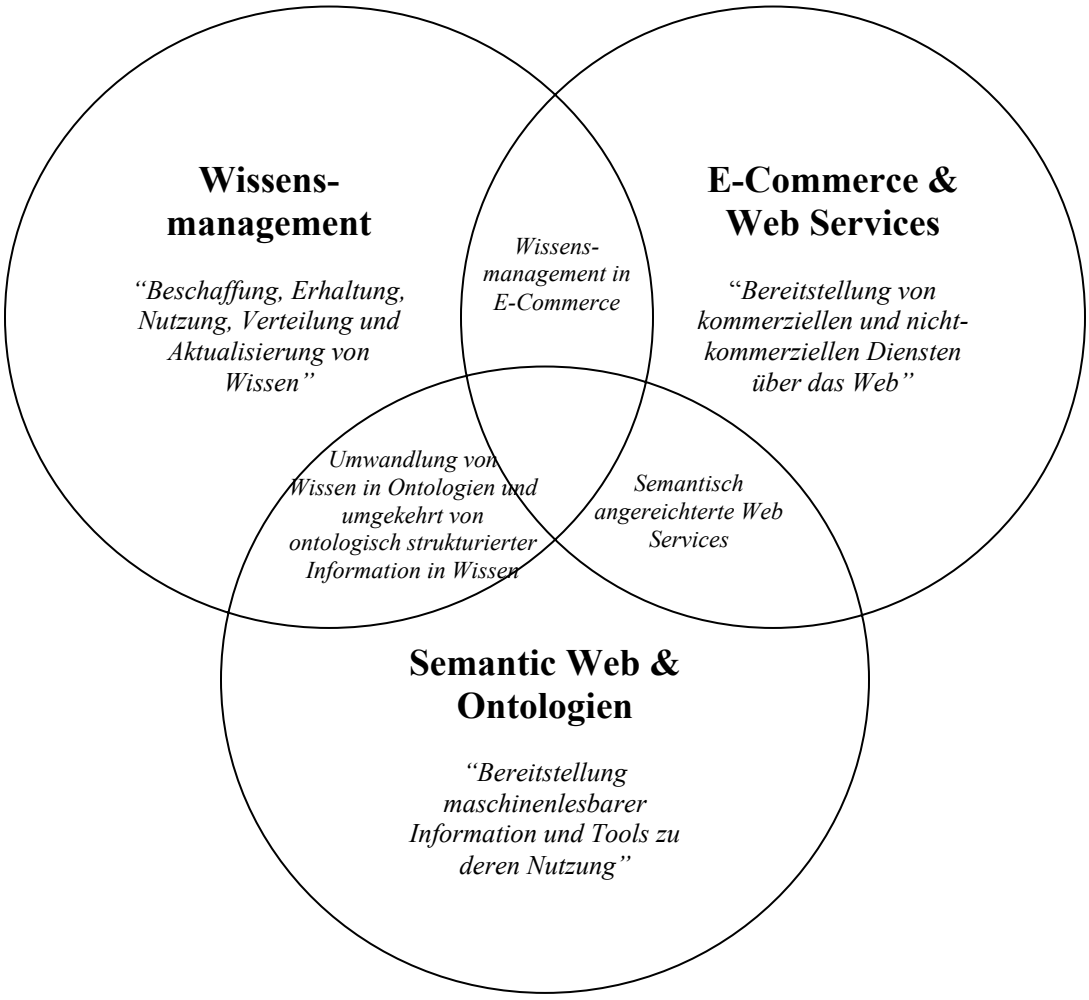


Figure 4: Themen- und Forschungsbereiche im Zusammenhang mit dem Semantic Web

Die Abbildung soll die Verwandtschaft des Themenbereichs „Semantic Web & Ontologien" zu den Bereichen „Wissensmanagement" und „E-Commerce & Web Services" aufzeigen.

Wissensmanagement beschäftigt sich wie der Name sagt mit Wissen. Im akademischen Zweig gibt es eine Verwandtschaft zum Forschungsbereich der künstlichen Intelligenz. Auf der anderen Seite ist Wissensmanagement in der Betriebswirtschaft ein vieldiskutiertes Thema. Im Informationszeitalter betrachten viele Firmen das Wissen als bedeutenden Erfolgsfaktor. Geeignete Lösungen und Prozesse entlang dem sogenannten Lebenszyklus des Wissens werden daher in den Firmen entwickelt und implementiert. Wissen ist wie schon erwähnt auch wichtigster Inhalt eines Expertensystems. (vgl. [4])
E-Commerce und Web Services sind Bereiche, welche dadurch angetrieben sind das Web kommerziell nutzbar zu machen und funktionell durch Web Services aufzuwerten. Einerseits gibt es da die Dienste, welche privaten Nutzern zur Verfügung gestellt werden. Andererseits besteht die Absicht zukünftig Geschäftsprozesse wie die Auftragsabwicklung, der Kauf, das Abrechnungswesen und weitere Prozesse zunehmend zu automatisieren. Anstelle proprietärer Systeme, die an der Schnittstelle zwischen den jeweiligen Geschäftspartnern angepasst werden müssten, könnte das Web standardisierte (d.h. neutrale) und global verfügbare Plattform für Dienste, Prozesse und Applikationen dienen. "Web services are rapidly becoming the enabling technology of today's e-business and e-commerce systems, and will soon transform the Web as it is now into a distributed computation and application framework." (siehe „Preface" von [5]).Viel Arbeit wird geleistet, um die Interoperabilität von Web Services zu ermöglichen. Ein semantischer Ansatz kann dabei sinnvoll sein, indem die Mehrfachverwendung und Erweiterungen erleichtert werden (vgl. [5]).
In Anlehnung an eine Internetquelle (vgl. [d15]) zeigt das nachfolgende Schema entlang einer darin vorgestellten Wertschöpfungskette den Stand der technologischen Entwicklungen des Semantic Web auf.

| Value chain | Description | Technological requirements | Maturity of development |
|---|---|---|---|
| **Knowledge creation** | Inferencing, contextualization, information broking | Ontology (metadata taxonomy + set of inference rules), automated dynamic linking of ontologies, automated inferencing | 25% |
| **Knowledge retrieval** | Tools for searching and finding of knowledge | Highly automated retrieval / navigation engines, Q&A repositories | 75% |
| **Collection of data** | Collection of knowledge and linking of content to knowledge models (annotation) | Automated knowledge collection / annotation engines | 50% |
| **Creation of ontologies** | Tool-based modeling of knowledge domains | Automated ontology creation tools | 25% |

Table 2: Wertschöpfungskette der semantischen Technologien,
Quelle: vgl. http://www.aifb.uni-karlsruhe.de/AIK/veranstaltungen/aik9/presentations/slides/020419FutureSemanticWeb.pdf

## *Zukunftschancen des Semantic Web und Schlussfolgerungen*

Die eingangs skizzierten Entwicklungen der heutigen Datenwelt und dabei insbesondere das rasante Wachstum des Internet zeigen die Notwendigkeit für neuartige Ansätze, um den Nutzen des Web als riesige Informationsquelle zu erhalten. In diesem Abschnitt soll kurz qualitativ

erörtert werden, welche Chancen für die Realisierung des Semantic Web bestehen, und einige kritische Punkte zur Sprache gebracht werden. Am Ende des Abschnitts werden aus den Erkenntnissen dieses Kapitels Schlussfolgerungen für diese Arbeit gezogen.
Eine der wohl wichtigsten Zweifel zu den Chancen des Semantic Web liegt in der Frage nach der Motivation für die Betreiber einer Website diese mit RDF- oder andersartigen Metadaten zu versehen. Dem Mehraufwand, welcher durch das Annotieren von Metadaten entsteht, steht kein direkter Nutzen gegenüber. Leider ist der Mehraufwand auch nicht beim Besuch einer Webseite sichtbar! Das Web funktioniert so wie es heute ist gut. Für den Betreiber einer Website existiert kein Leidensdruck etwas zu ändern. Als Entgegnung auf diesen Zweifel können folgende Argumente angeführt werden, welche jedoch vorerst spekulativ bleiben: (a) Benutzerfreundliche Funktionen zur Annotation von Metadaten können in Standard-Webeditoren integriert werden und so den Aufwand für die Webdesigner gering halten. (b) Semantische Daten können in Browsern oder zukünftigen Suchmaschinen sichtbar gemacht werden. Pioniere werden also dadurch belohnt, dass ihre Webseiten verstärkt sichtbar sind und über semantische Suchmaschinen gefunden werden. (c) Als qualitatives Argument ist beizufügen: Hätte jemand 1990 geglaubt, dass Millionen von Leuten bereit sind (meist unentgeltlich) Informationen öffentlich verfügbar zu machen?
Bezüglich technischer Probleme wurde das Mapping-Problem zwischen Ontologien mit ähnlichem Vokabular bereits erwähnt. Weiter ist anzufügen, dass das Vorhandensein semantischer Metadaten die Auffindbarkeit von Dokumenten im Internet noch keineswegs sicherstellt. Lösungsansätze dafür können Semantische Suchmaschinen mit unvermindert riesigen Indizes oder Webportale sein, welche einen semantischen Zugriff auf ein Themengebiet ermöglichen. Auch das noch unausgereifte Konzept der Agenten könnte hilfreich sein.
Andere technische Schwierigkeiten bestehen möglicherweise bezüglich der Anwendbarkeit von semantischen Metasprachen (wie RDF) auf Dokument oder Dokumentsammlungen und bezüglich der Handhabbarkeit dieser Metadaten.
Vorteilhaft erscheint die Art und Weise wie die semantischen Technologien in das bestehende Web integriert werden können und sollen. Dem dezentralen und nicht-hierarchischen Charakter wird Tribut gezollt und die Ontologien als höherwertige, strukturierte Begriffsschemata werden von wenigen dazu speziell fähigen Parteien erstellt werden. Im Weiteren bietet sich XML als geeignetes Transportgefäss für RDF-Daten an.
Als Stärken des Semantic Web treten ausserdem die bereits geleistete Vorarbeit im Bereich des Wissensmanagements und die vorhandenen Synergien zur Problematik des Wissensmanagements in Unternehmen hervor.
In der Gesamtbeurteilung sind die Chancen für eine Erweiterung des Web in Richtung Semantic Web also durchwegs positiv. Aufgrund noch offener Fragen bzw. bestehender technischer Unklarheiten werden im nächsten Abschnitt angelehnt an die Aufgabenstellung Schlussfolgerungen für die vorliegende Arbeit gezogen.
Die Idee vom Semantic Web bietet gute Ansätze, um aus der Informations- und Dokumentensammlung des Internet zusätzlichen Nutzen zu gewinnen und um die sich durch dessen Wachstum zuspitzenden Probleme des Auffindens von Information zu entschärfen. Mit RDF steht ein durchdachter und breit akzeptierter Standard zum Verfassen von Metadaten zu Daten jeglicher Art zur Verfügung. Ontologien sind ein die Metadaten ergänzendes Konzept, welche als strukturierte Wissensspeicher Semantik bzw. Bedeutung beifügen.
Mit der vorliegenden Semesterarbeit soll das Wissen rund um RDF und Semantic Web vertieft werden. Neben einem Teil, der vorwiegend die Informationsbeschaffung zum aktuellen Stand der Entwicklungen zum Ziel hat, wird in einem zweiten eher praktisch orientierten Teil ein Tool entwickelt und implementiert. Dieses Tool soll wie in der Aufgabenstellung erwähnt semantische Netze erfassen und strukturieren. Diese Struktur, die wahrscheinlich die Form eines Graphen haben wird, soll die Navigation darin und die Suche nach Daten ermöglichen.

Neben diesen konkreten Zielen soll die Arbeit möglicherweise vorhandene Schwierigkeiten in der Arbeit mit solchen semantischen Netzen aufzeigen und als Grundlage für nachfolgende Arbeiten zum Thema Semantic Web dienen.

# Systembeschreibung und Lösungskonzept

## *Der Untersuchungsgegenstand und dessen systemische Abgrenzung*

In diesem Abschnitt soll der untersuchte Gegenstand genau definiert werden sowie gegenüber dem umgebenden System präzise abgegrenzt werden. Damit soll klargestellt werden, was betrachtet wird und dann gegebenenfalls in der praktischen Umsetzung beeinflusst wird.

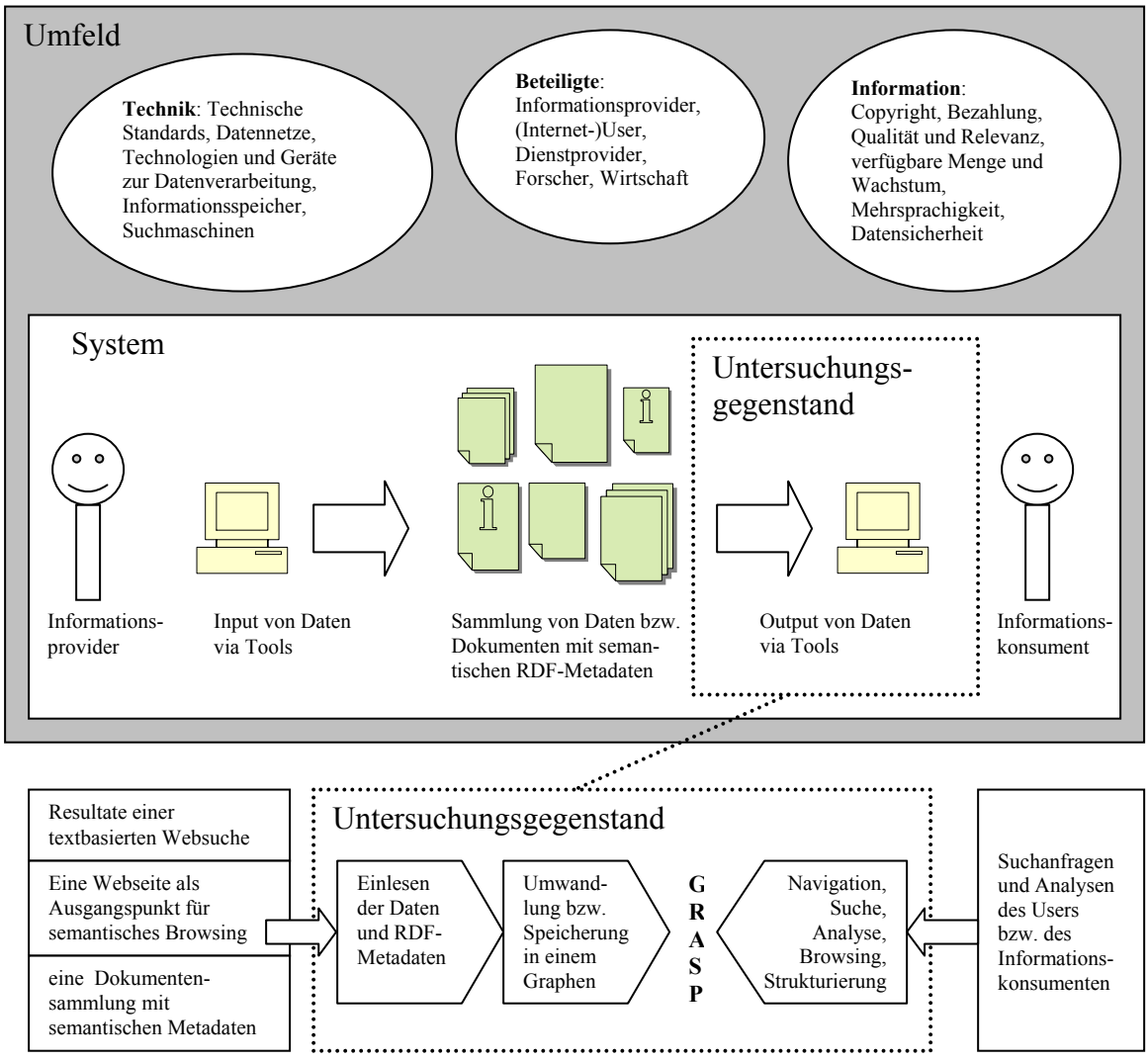


Figure 5: Untersuchungsgegenstand und Systemumfeld

Die obige Darstellung illustriert das Umfeld, das System und den Untersuchungsgegenstand. Im Umfeld finden sich Einflussgrössen der drei Hauptkategorien Technik, Beteiligte und Information. Das System gleicht einer Box mit Daten bzw. Dokumenten, welche mit semantischen Metadaten ergänzt sind. (Diese Daten oder Dokumente sind in der weiteren Betrachtung vorwiegend mit RDF annotierte Webseiten.) In die Box findet von der Seite der Informationsprovider (mittels Annotationstools, Ontologieerstellungstools, Webeditoren etc.) der Input von Daten statt. Auf der anderen Seite werden von Informationskonsumenten Daten ausgelesen (mittels Browser etc.).
Der Output-Vorgang ist der zu untersuchende und zu verändernde Gegenstand dieser Arbeit. Die groben Schritte dieses Vorgangs sind im unteren Teil der Darstellung gezeigt: Semantische Metadaten werden eingelesen und daraufhin in einem Graphen oder einer anderen Struktur gespeichert. Auf diesem Graphen führt der Informationskonsument Navigations-, Such-, Analyse-, Browsing- oder Strukturierungsprozesse durch.
Auf der Inputseite ist es wichtig die Unterscheidung zwischen einem vorgegebenen Set an Daten und einem Startpunkt innerhalb vernetzter Daten zu machen. Im ersten Fall wird das Datenset eingelesen und anschliessend analysiert oder weiterverarbeitet. Im zweiten Fall wird in einer Art von Browsing, welches den Links zu vernetzten Daten folgt, der Graph schrittweise erweitert. In dieser Arbeit soll der erste Fall der Analyse und Verarbeitung eines bestehenden Sets an Daten im Vordergrund stehen. Der zweite Fall ist etwas komplizierter, ist jedoch als Erweiterung in einer späteren Arbeit realisierbar.

## *Szenarien zur Interkonnektivität der Inputdaten*

Es ist zu erwarten, dass je nach Typ der eingelesenen (Web-) Daten unterschiedliche Graphen resultieren werden. Dabei spielt die Häufigkeit von Verknüpfungen eine grosse Rolle. In der Art von Szenarien sind in der nachfolgenden Tabelle fünf Typen von Datensammlungen definiert.

| **Typ** | **Beschreibung** |
|---|---|
| Island | Ein Island-Typ von Datensammlung hat kaum Links zu Webressourcen mit anderer URI. In den meisten Fällen wird eine solche Island eine Webseite mit nur wenigen Subseiten sein. Oft sind diese Seiten privaten Ursprungs mit wenig wichtiger Information und wenig semantischen Metadaten.<br>Am ehesten wird es bei diesem Typ Links nach aussen zu einer Ontologie geben, jedoch kaum Links, welche von aussen auf diese Datensammlung verweisen. |
| Site | Eine Datensammlung des Typs Site ist die klassische Art von Website. Weblinks, RDF-Links zu anderen Datensammlungen und semantische Links – alle Arten sind hier in durchschnittlicher Anzahl zu finden. Kleine Organisationen und Firmen betreiben in den meisten Fällen solche Websites. |
| Friends | Datensammlung vom Typ Friends sind eine Gruppe verschiedener durch RDF- und Internetlinks verbundener Webressourcen. Durch die bei ihnen behandelten Themen oder die gemeinsamen ontologischen Begriffe bilden sie eng vernetzte Graphen. |
| Portal | Eine Datensammlung vom Typ Portal hat sehr viele nach aussen führende Verbindungen. Oft gehören diese mit dem Portal verknüpften Webressourcen zu einem Themengebiet. Es kann jedoch auch Fälle geben, in denen die Verknüpfungen thematisch mehr oder weniger willkürlich sind. |

Table 3: Webpage Typen und ihre Interkonnektivität

## *Datenmodell*

In diesem Abschnitt wird der Frage nachgegangen, wie die eingelesenen Daten aussehen. Modellhaft wird beschrieben, welche Eigenschaften diese Daten besitzen und wie sie verknüpft sind.

Die grafische Schematisierung von RDF-Triples als Ovale (bzw. Rechtecken) verbunden durch Pfeile ist leicht verständlich. Die gleiche Art zur grafischen Repräsentation wird auch hier verwendet (vgl. Absatz zu RDF). Anhand der nachfolgenden Grafik wird das Datenmodell erklärt, welches zusätzlich zu den eigentlichen RDF-Daten weitere Ebenen enthält.

Die RDF-Daten und andere denkbare Metadaten werden auf der Ebene „Metadaten" erfasst. Diese Metadaten beschreiben Ressourcen und ihre Beziehungen zueinander. Als Prädikate (grafisch Pfeile) dienen ontologische Begriffe, welche durch Ontologien umschrieben werden.

Die darunterliegende zweite Ebene trägt die Bezeichnung „ontologische Begriffe". Auf dieser Ebene sind die Ontologien eingetragen, welche zur Umschreibung der Metadaten benötigt werden. Die einzelnen ontologischen Begriffe nehmen den Platz der Ressourcen innerhalb der Metadaten ein. Sie können als mit anderen Begriffen verbundene Elemente verstanden werden, welche über (durch Pfeile dargestellte) logische Relationen mit diesen verbunden sind. Es existiert so ein ontologisches Netzwerk, welches in seiner Art im Gegensatz zu den Metadaten jedoch eher hierarchisch strukturiert ist. Es muss erwähnt werden, dass die logischen Relationen in Ontologien in ihrer Komplexität oftmals über einfache `Subclass_of` Beziehungen hinausgehen. (Im dargestellten Beispiel gibt es nur `Subclass_of` Beziehungen.) Solche komplexeren Relationen, wie beispielsweise ein Ausdruck, der besagt, dass eine Klasse als Sub-Klasse ausschliesslich die Begriffe einer von zwei Sub-Klassen besitzen kann (eine sogenannte *exhaustive subclass partition*), können oftmals nur im Umweg über Zwischenelemente realisiert werden. Solche indirekten Formulierungen sind auch in RDF nicht ungewöhnlich, wie später noch gezeigt wird.

Als dritte Ebene bezieht das entwickelte Datenmodell „Keywords" als zusätzliche Elemente mit in den Graphen ein. Diese Erweiterung ist dadurch begründet, dass angenommen werden muss, dass semantische Metadaten nur eine grobe Umschreibung der in den beschriebenen Daten liegenden Information sind und dass es sein kann, dass diese Metadaten nur bruchstückhaft angefügt werden. Im Beispiel kommen die Keywords „Gershwin" und „Porgy" vor, welche mit der Ressource vom Typ Konzert verbunden sind.

Um also die Vorteile der textbasierten Suche, wie wir sie heute kennen und nutzen, in einem semantisch orientierten System zu integrieren, soll die Ebene der Keywords die Möglichkeit bieten die Auftretenshäufigkeit von einzelnen Wörtern in Ressourcen bzw. Textdokumenten zu erfassen. Diese Zusatzinformation wird bei der Informationssuche einen wichtigen Mehrwert gegenüber einzig auf semantischen Metadaten abgestützten Systemen bieten. Solche Keywords können nicht vollständig durch ontologische Begriffe abgedeckt werden, da sie erwartungsgemäss auch Eigennamen oder seltene Begriffe enthalten werden, die nicht formal in einer Ontologie beschrieben sein werden.

Eine letzte vierte Ebene (zuoberst in der Abbildung) integriert weitere zusätzliche Elemente ins Datenmodell, welche hier als „Konzentrate" bezeichnet werden. Die Idee ist, dass jedes informationsverarbeitende System am Ende für die Nutzung durch einen Menschen geschaffen ist. Der Mensch kann und will Information jedoch nicht aus vielen kleinen Teilen zusammensetzen. In Analogie zu einem Inhaltsverzeichnis eines Buches wünscht er in der Regel konzentrierte Zusammenfassungen von Information, um nur bei Bedarf genauere Details einzusehen. Das Ziel eines Tools zur Analyse eines semantischen Netzwerkes wird es deshalb sein in einem letzten voraussichtlich sehr schwierigen Schritt eine grosse Datenmenge thematisch zu strukturieren (*clustering*). Solche strukturierte Information kann dann in Form der hier vorgestellten Konzentrate verfügbar gemacht werden. Es ist einsichtig, dass solche Konzentrate nicht vollständig durch ontologische Begriffe abgedeckt werden können. Sie können komplizierte

Zusammenhänge beinhalten und so vielleicht am ehesten als Kombination aus Keywords und ontologischen Begriffen bezeichnet werden. Ein Beispiel zusätzlich zu dem der Abbildung könnte sein: „Bücher von deutschen Nachkriegsautoren, die in ihrer assoziativen Erzähltechnik stilistisch mit jenen von Virginia Woolf verwandt sind“.
Die folgende Abbildung zeigt ein Beispiel mit einer Handvoll Elementen, welche über alle Ebenen hinweg miteinander verbunden sind.

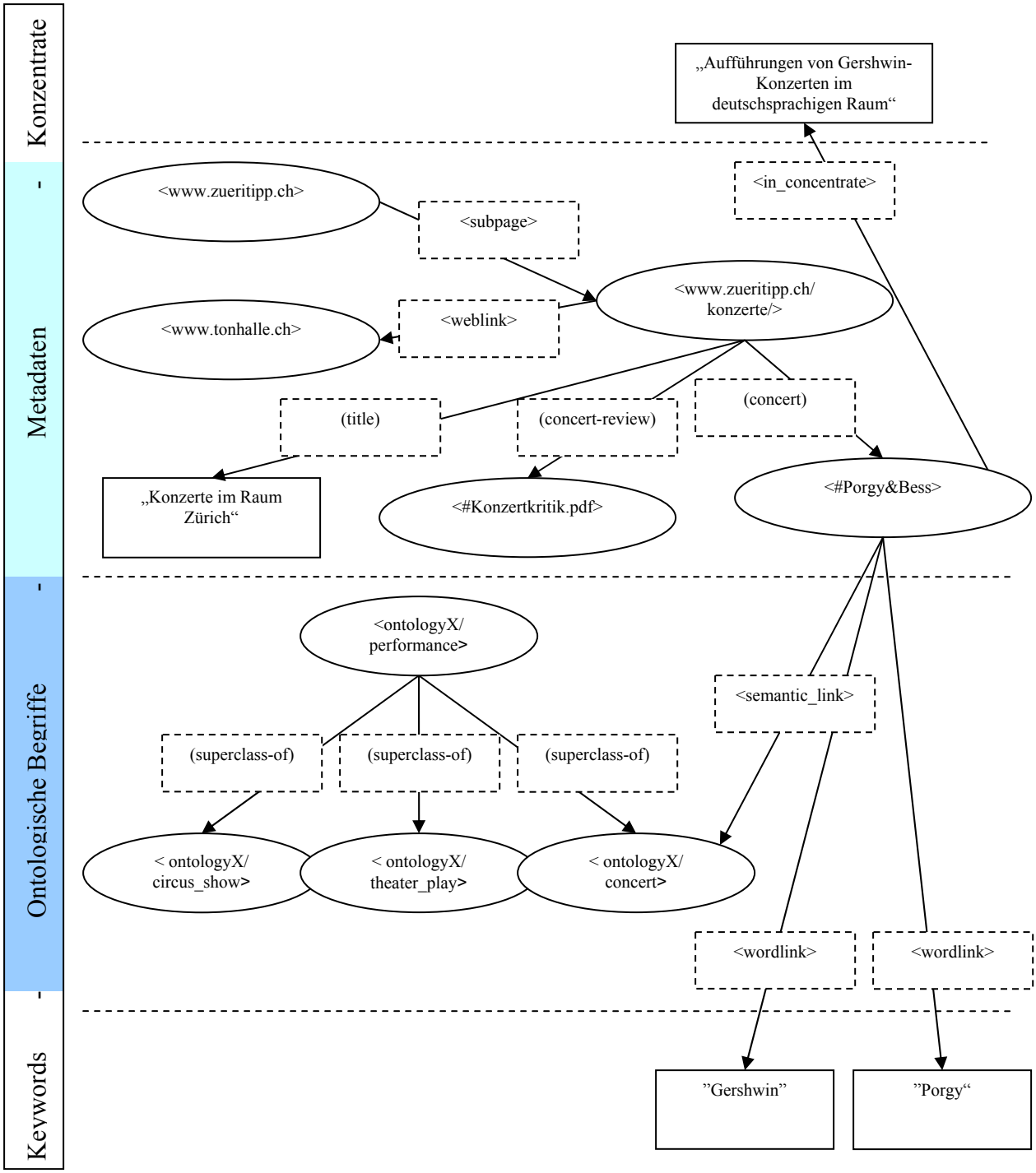


Figure 6: Verschiedene Ebenen des Datenmodells

Das Datenmodell mit den vier verschiedenartigen Elementkategorien oder Ebenen ist darauf ausgerichtet vollständig zu sein in dem Sinne, dass keine wichtige Information vergessen geht. Für das später vorgestellte semantische Tool stehen der Einfachheit halber und wegen des Interesses für semantische Daten vorläufig die beiden Ebenen „Metadaten“ und „ontologische Begriffe“ im Zentrum der Aufmerksamkeit.
Eine Erweiterung zur Beschränkung auf RDF-Metadaten liegt neben den erwähnten angefügten Ebenen auch durch gewisse Verbindungen vor. Dies sind hauptsächlich Verbindungen, welche über Grenzen zwischen den Ebenen hinweg führen. Weiter finden sich aber auch spezielle Verbindungen innerhalb einer Ebene. Im angeführten Beispiel sind dies die Verbindungen `<weblink>` und `<subpage>`. Diese erfassen nicht-semantische Verknüpfungen zwischen Webressourcen: im ersten Fall ein klassischer Internetlink und im zweiten Fall die logische Verbindung von Seiten innerhalb desselben Domains (oder Verzeichnisses). Verbindungen und Beziehungen, die eine Ausnahme zu einfachen RDF-Beispielen bilden, werden in zwei eigenen Abschnitten beschrieben.
Bereits das einfache dargestellte Beispiel zeigt wie schnell ein Graph unübersichtlich werden kann. Der Aspekt der Visualisierungsmöglichkeiten ist deshalb in seiner Wichtigkeit für das Tool nicht zu unterschätzen. (Existierende Arbeiten zur Visualisierung von Information im Zusammenhang mit dem Semantic Web sind in [3] ausführlich vorgestellt.)

## *Grundelemente*

In Anlehnung an das Datenmodell sollen die Daten in einem Graphen aus Knoten und gerichteten Kanten erfasst werden (engl.: *nodes and directed edges*). Die Knoten repräsentieren Ressourcen im Sinne des RDF-Standards, ontologische bzw. semantische Begriffe wie sie innerhalb einer Ontologie definiert werden oder Text unterschiedlicher Art (RDF-Literal, Keywords, Konzentrat). Die Knoten werden nach der Ebene, in welcher sie gemäss dem Datenmodell liegen kategorisiert. Der Text, welcher zu einem Knoten oder einer Kante gehört wird fortan als „Label“ bezeichnet. Bei einer Ressource besteht dieses Label aus dem URI. In den anderen Fällen ist dies ein beschreibender Textstring. Gleichlautende ontologische Begriffe verschiedener Ontologien werden durch einen vorangestellten eindeutigen Identifier der Ontologie gegeneinander abgegrenzt.
Die Richtung der Kanten ist nicht im Sinne einer Einbahnstrasse unidirektional, sondern ist einzig für die korrekte Interpretation der Bedeutung des Prädikats oder allgemeiner der Verbindung zwischen zwei Elementen nötig. Es wird damit also die Subjekt-Objekt-Relation erfasst (bspw. Subjekt ist `superclass_of` Objekt). Eine Navigation oder Suche in die der Pfeilrichtung entgegengesetzte Richtung soll ebenfalls möglich sein. Dies bedeutet, dass ein Knoten „wissen“ muss, welche anderen Knoten eine Verbindung zu ihm haben.
Neben dem Label können Knoten und Kanten später durch weitere noch zu definierende Eigenschaften wie Verbindungsstärke oder Kategorie zusätzlich charakterisiert werden. Für die vorläufige Lösung ist dies jedoch nicht vorgesehen.

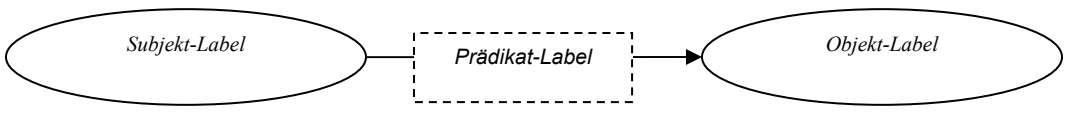


Figure 7: einfaches Beispiel einer Verbindung zwischen zwei Knoten

## *Verbindungen in Erweiterung zum RDF-Standard*

Im Datenmodell wurde bereits erwähnt, dass gewisse Verbindungen als eine Erweiterung zu den in RDF vorkommenden Kanten vorgesehen sind. Beim RDF Normalfall wird eine Ressource durch eine andere Ressource beschrieben, also beispielsweise Ressource A hat den Autor beschrieben durch Ressource B.
Die „erweiterten" Verbindungen sind alle Fälle, welche Elemente aus zwei verschiedenen Ebenen verbinden. Daneben gibt es aber auch solche Verbindungen innerhalb einer Ebene. Nachfolgend sind solche Verbindungen beschrieben und mit Beispielen illustriert. Für alle diese Verbindungen muss ein eigenes kleines ontologisches Vokabular existieren, damit sie gleichwertig zu RDF-Prädikaten behandelt werden können.

**Von Webpage zu Webpage:**
Neben den semantischen RDF-Links werden auch die traditionellen Internetlinks wichtige Informationen über eine Datensammlung liefern und als "Türen nach aussen" zur Erweiterung der Datensammlung dienen. Diese klassischen Internetlinks sind normalerweise nicht durch RDF-Metadaten beschrieben. Sie müssen daher separat erfasst werden.

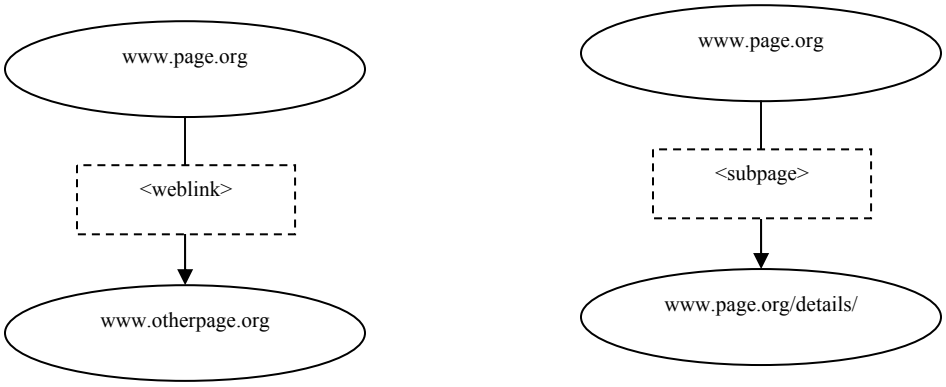


Figure 8: Webpage zu Webpage

**Von ontologischem Begriff zu ontologischem Begriff:**
Innerhalb der Ontologien werden die ontologischen Begriffe über logische Relationen zueinander in Beziehung gesetzt. Ganz einfache Verbindungen wie hier im Beispiel `superclass_of` sind möglich. Daneben sind aber viele andere logische Relationen denkbar, die weit komplizierter sein können.

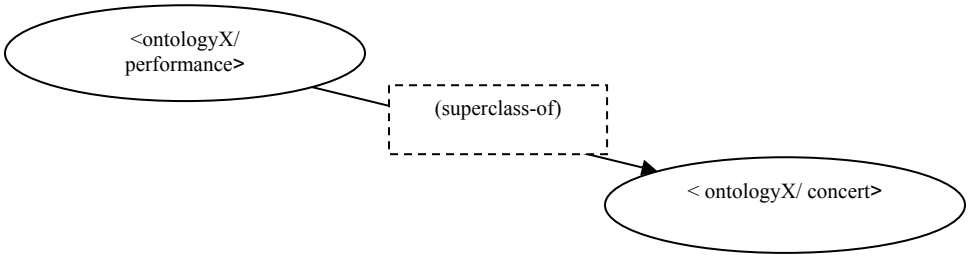


Figure 9: ontologischer Begriff zu ontologischer Begriff

**Von Konzentrat zu Konzentrat oder von Keyword zu Keyword:**
Verbindungen zwischen Konzentraten oder Keywords werden einen assoziativen Charakter haben. Eine solche Verbindung kann die Nähe zwischen zwei Themen oder Begriffen auf eine „fuzzy“ Art ausdrücken. Dies würde die strenge Struktur einer Ontologie ergänzen. Ähnlich wie in neuronalen Netzwerken könnte eine zusätzliche Kanteneigenschaften die Nähe zwischen zwei Elementen oder anders gesagt die Stärke der Verbindung erfassen.

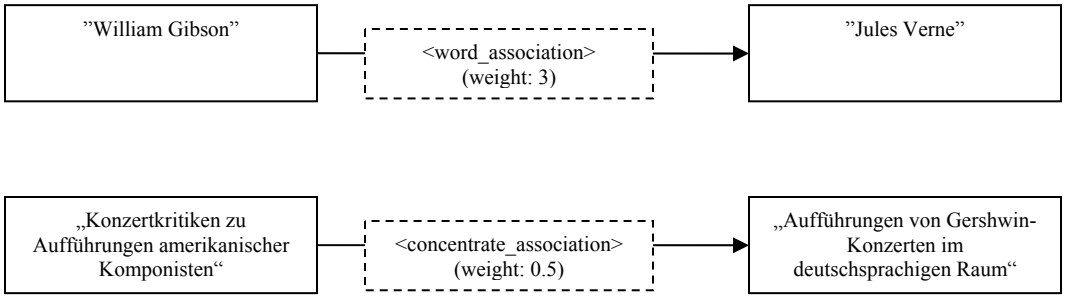


Figure 10: Von Konzentrat zu Konzentrat oder von Keyword zu Keyword

**Von Webressource zu Keyword:**
Um die Stärken der textbasierten Suche ins Lösungskonzept zu integrieren wurden bereits im Datenmodell die Keyword-Elemente vorgestellt. Das Auftreten eines Suchbegriffes in einem Dokument soll damit erfasst werden können. Nachfolgend sind zwei verschiedene Varianten aufgezeigt, wie die Zusatzinformation über die Häufigkeit des Auftretens eines Begriffs festgehalten werden kann.
Da es keinen Sinn macht alle in einem Dokument vorkommenden Wörter als Keywords einzuscannen, muss eine Methode angewandt werden, um eine Auswahl der erfassten Wörter zu treffen. Die Auswahl könnte vom Benutzer bestimmte Suchwörter einschliessen. Oder die Wörter könnten solche sein, welche in verschiedenen Dokumenten gleichzeitig vorkommen. Es ist auch denkbar Eigennamen zu identifizieren, welche nicht in einem Wörterbuch der benutzten Sprache vorkommen.

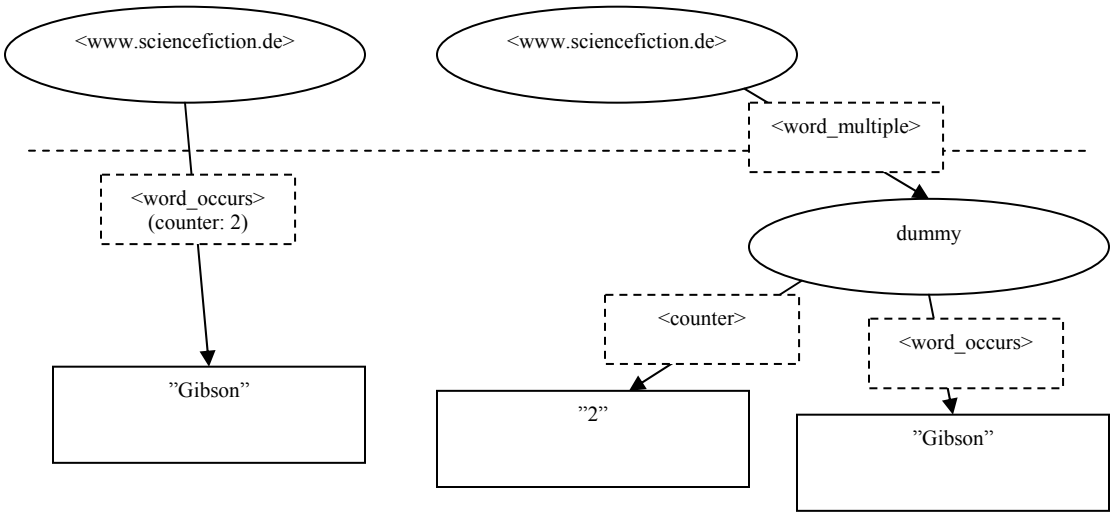


Figure 11: Von Webressource zu Keyword

**Von Ressource zu ontologischem Begriff:**
Die bedeutsame Verbindung von einer Ressource zu einem ontologischen Begriff kann mit einem speziellen Link mit dem Label `semantic_link` erfasst werden. Inhärent existiert diese Verbindung bereits, da jedes RDF-Prädikat einen Verweis auf eine Ontologie enthält. Eine explizit erfasste Verbindung mit dem Label `semantic_link` einerseits zum Zwecke der Visualisierung nötig und unterstreicht andererseits die im Datenmodell kommunizierte Idee eines Netzwerks aus gleichwertigen Elementen, welche über verschiedene Ebenen hinweg miteinander verbunden sind.

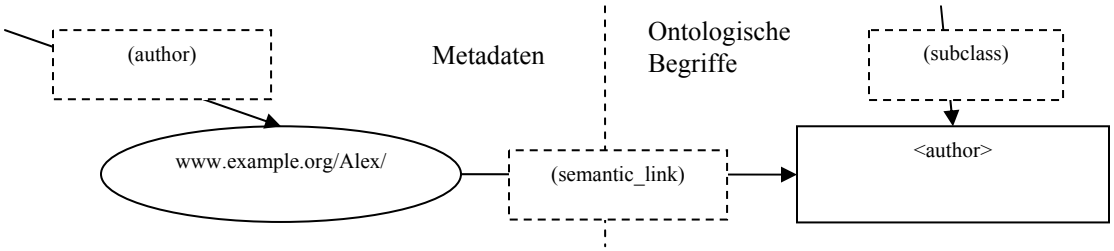


Figure 12: Von Ressource zu ontologischem Begriff

**Von Ressource zu Konzentrat:**
Im Zusammenhang mit der bereits im Datenmodell besprochenen „konzentrierten Information" soll eine spezielle Verbindung `in_concentrate` die Einordnung des Inhalts einer Ressource in ein Konzentrat beschreiben. Die Erfassung eines solchen Informations-Konzentrates wird maschinell nur sehr schwer realisierbar sein. Die Idee vom Konzentrat hat deshalb utopischen Charakter. Diese Idee wird hier dazu auf Papier gebracht, damit die Bedürfnisse des menschlichen Nutzers nach konzentrierter, zusammengefasster Information bei einer technischen Ingenieurlösung nicht vergessen gehen.

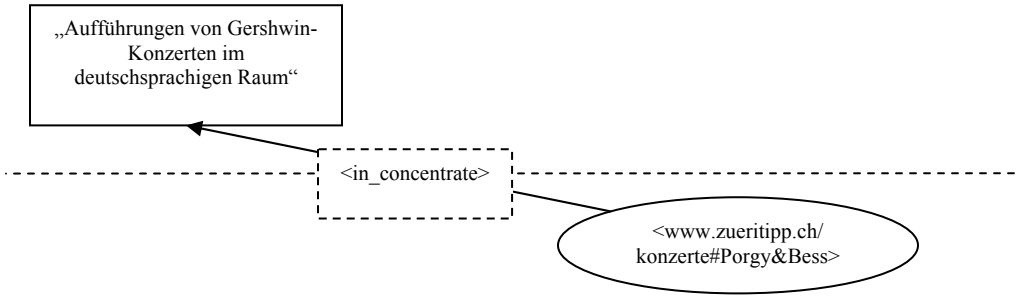


Figure 13: Von Ressource zu Konzentrat

## *Ungewöhnliche Knoten-Kanten-Beziehungen*

In diesem Abschnitt werden ungewöhnliche Beziehungen vorgestellt, welche mehrere Knoten und Kanten einschliessen. Solche speziellen Verhältnisse innerhalb des Graphen sind bei der Implementation des Tools zu berücksichtigen, damit nicht aufgrund von falschen Annahmen Fehler auftreten.

**Loops:**
Innerhalb des Graphs werden Loops vorkommen. Häufig wird dies der Fall sein, wenn man der entgegengesetzten Richtung einer gerichteten Kante folgt. Das untenstehende Beispiel illustriert jedoch, dass dies auch der Pfeilrichtung folgend möglich ist. Loops müssen im Speziellen bei der Iteration durch den Graphen berücksichtigt werden. Es soll keine endlosen Iterationsschleifen geben!

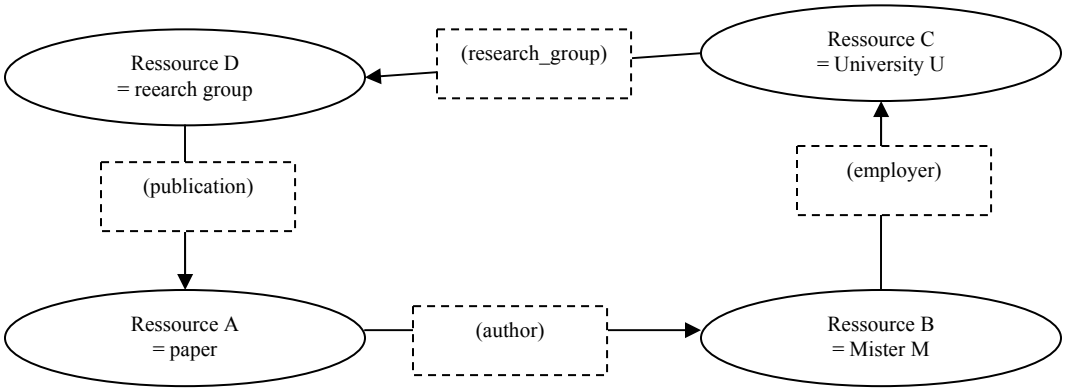


Figure 14: Loop

**Reifikation (*reification*):**
Reifikation wird definiert als "*treatment of an analytic or abstract relationship as though it were a concrete entity.*" (s. [d16]) Etwas spezifischer auf den Kontext von RDF angewandt heisst es: *"Reification is a mechanism to record statements without asserting them."* (s. [d17])
Als Erklärung dienen die nachfolgenden Illustrationen. Eine Ressource D wird über ein Prädikat P durch eine zweite Ressource C beschrieben.

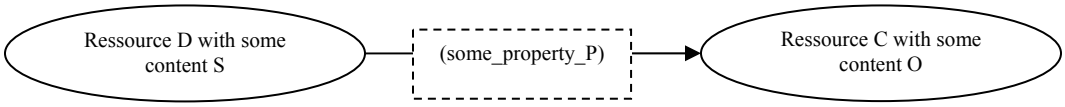


Figure 15: Reification

Beschreibt nun eine andere Ressource A diesen Zusammenhang, so liegt eine Aussage über eine Aussage vor, welche als Reifikation bezeichnet wird. Die dazu verwendete Schreibweise ist weiter unten ersichtlich. Diese Reifikation wird technisch kaum Schwierigkeiten verursachen.

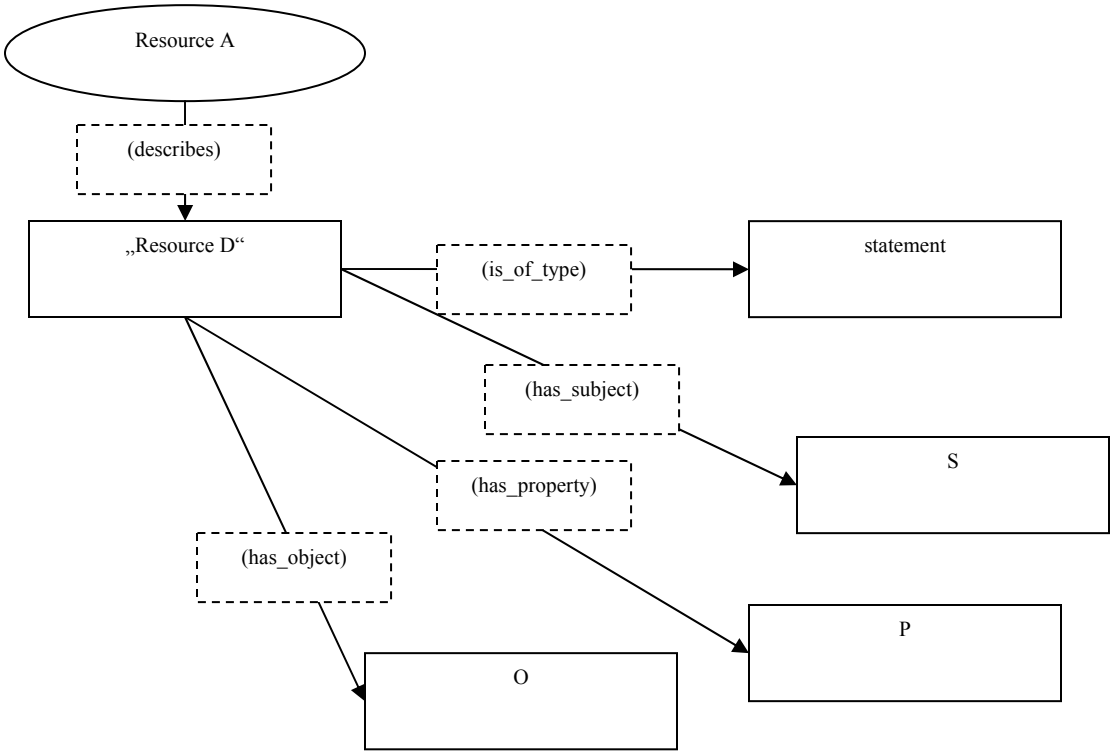


Figure 16: Reification

**Aussage über ein Prädikat (*property*):**
Ein Beispiel von Reifikation, welches häufiger vorkommt als das oben vorgestellte, ist eine Aussage über ein Prädikat (d.h. eine Kante). Das nachfolgende Beispiel zeigt, dass eine Kante indirekt beschrieben werden kann, indem darauf verwiesen wird, dass ein Knoten eine Kante mit einem Prädikat XY besitzt. Eine einfachere Schreibweise ist über einen `semantic_link` möglich, falls direkt auf die Ontologie verwiesen werden kann (vgl. Abschnitt zu Verbindungen in Erweiterung zum RDF-Standard).

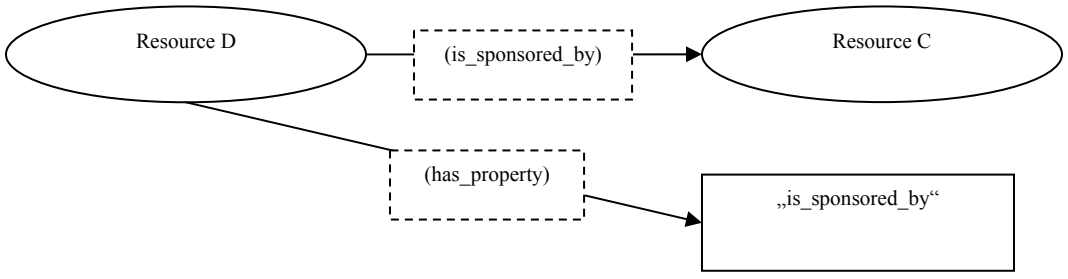


Figure 17: Aussage über ein Prädikat

**Mehrfachverwendung eines Prädikats (*grouped properties or multiple values*):**
In RDF ist vorgesehen, dass ein Prädikat für mehrere Objekte - also für eine Objektgruppe - benutzt werden kann. Dazu gibt es drei vordefinierte Gruppentypen (oftmals *container* genannt):

*bag*, *sequence* und *alternatives* (vgl. [b2]). Da eine Kante nicht gleichzeitig auf mehrere Objekte verweisen kann, werden ein Umweg und Identifikationsnummern benutzt.

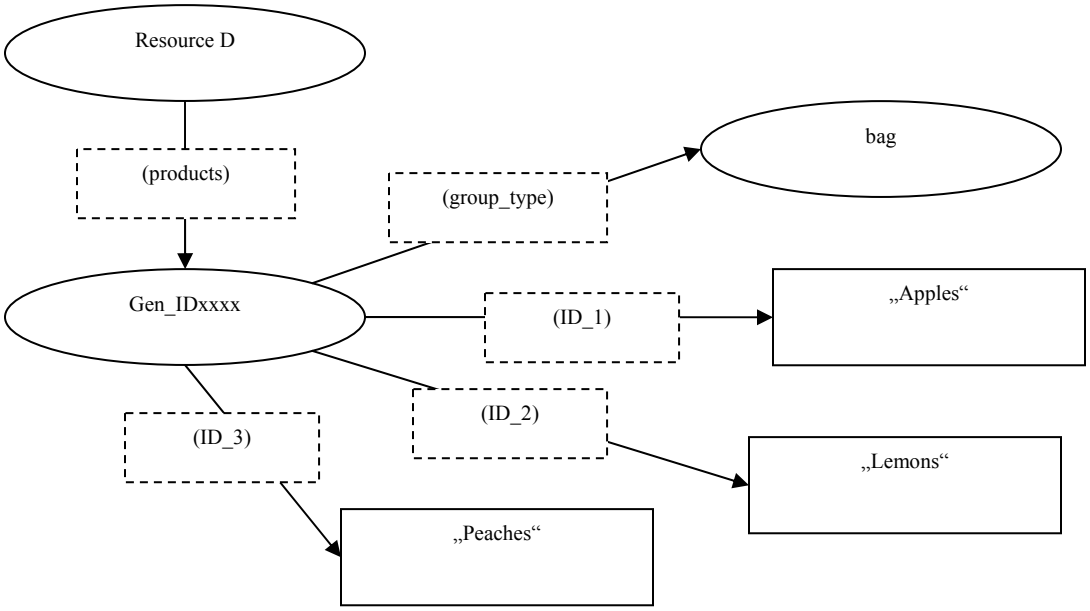


Figure 18: Mehrfachverwendung eines Prädikats

**Parallele Kanten:**
Es ist möglich, dass parallele Kanten vorkommen, so wie sie im unten dargestellten Beispiel vorliegen. Beim Auffinden von Verbindungen zwischen zwei Knoten ist dieser Fall zu berücksichtigen.

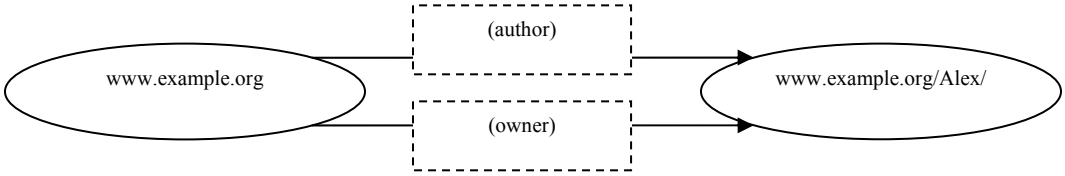


Figure 19: Parallele Kanten

## *Suchoperationen*

Es wird nie ein Problem sein einen spezifischen Knoten zu finden, sofern die betrachtete Menge an eingelesenen Daten nicht sehr gross ist. Andere Suchanfragen können jedoch komplizierter sein. In diesem Abschnitt werden einige der zu erwartende Besonderheiten vorgestellt, welche berücksichtigt werden sollten, damit Suchanfragen richtig konzipiert und ausgeführt werden bzw. die Suchresultate korrekt interpretiert werden.
Generell werden die Genauigkeit und der informatorische Wert von Suchresultaten stark davon abhängen, wie spezifisch die Suchanfrage ist bzw. wie genau der Benutzer weiss, nach was er sucht und was er ungefähr im Graph vorfinden wird. (Dies paradox!) Eine angepasste und gut durchdachte Suchmethodologie wird deshalb im Zusammenhang mit semantischen Suchanfragen von grosser Wichtigkeit sein.

**Die kürzeste Verbindung entspricht nicht der besten bzw. aussagekräftigsten Verbindung:**
Innerhalb des Graphen muss die kürzeste Verbindung zwischen zwei Knoten nicht unbedingt die beste Verbindung bezüglich ihres informatorischen Wertes sein. Die aussagekräftigste Verbindung zwischen zwei Knoten ist im Graphen also nicht inhärent vorgegeben, sondern existiert als solche nur nach der Beurteilung des Informationskonsumenten. Theoretisch sind beliebig lange „aussagekräftige“ Verbindungen möglich, die alle eingelesenen Knoten eines Graphen miteinschliessen.
Im nachfolgenden Schema wird beispielhaft die Suche nach einer Verbindung zwischen einer Webseite über ein Konzert und eine andere Webseite mit CD-Verkauf illustriert. Die gesuchte CD zum Konzert befindet sich auf der längeren Verbindung.

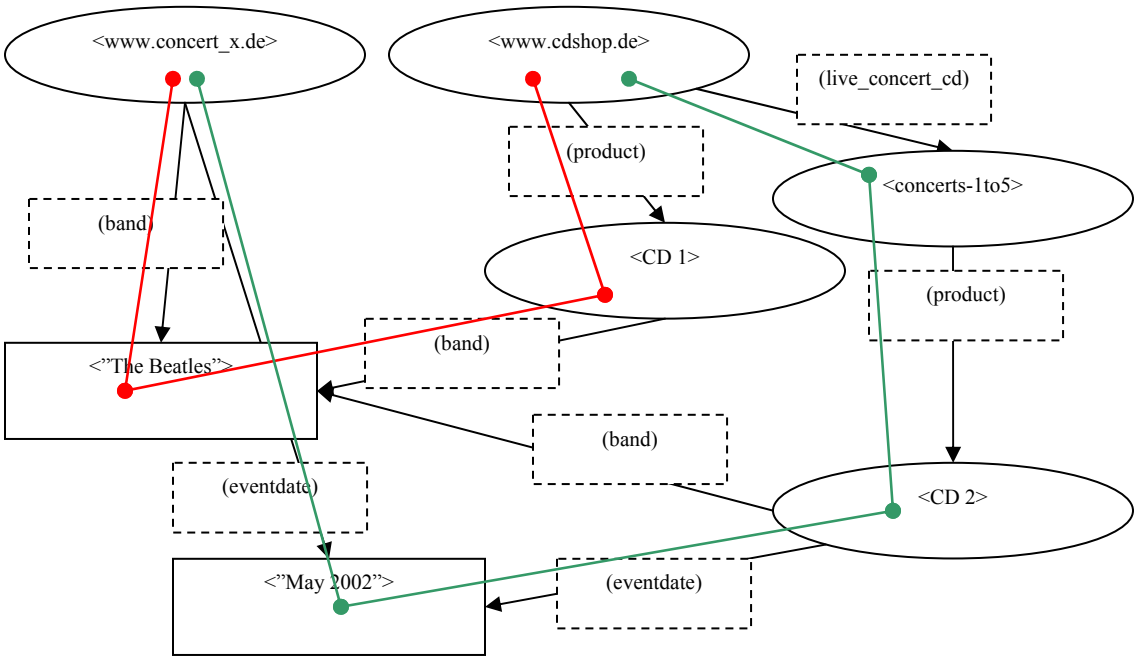


Figure 20: Aussagekräftige Verbindung

**Teilgebiete:**
Um die Suche auf bestimmte Teilgebiete innerhalb eines Graphen zu beschränken, müssen Grenzen zwischen solchen Gebieten erkannt werden. Solche Teilgebiete können der Adressbereich (via URI) einer Webressource, eine Ontologie oder eine Ebene bzw. Kategorie innerhalb des Datenmodells sein.
Hinweise auf den Wechsel in ein anderes Gebiet können Knoten einer andersartigen Kategorie, spezielle Kanten (wie `weblink` oder `semantic_link`), der Wechsel der Stammadresse im URI oder die wechselnde (Pfeil-)Richtung auf einer Kante sein.

**Semantische Instanzen:**
Eine sehr sinnvolle Suchanfrage betrifft die Suche nach den Instanzen sprich realen Webobjekten zu einem ontologischen Begriff. In einem solchen Fall ist es wichtig die verschiedenen (hierarchischen) Verallgemeinerungsebenen innerhalb einer Ontologie zu beachten. Im unten vorgestellten Beispiel sollten bei der Suche nach „*concert*“-Instanzen auch Instanzen der Sub-Klasse „*jazz-concert*“ eingeschlossen werden.

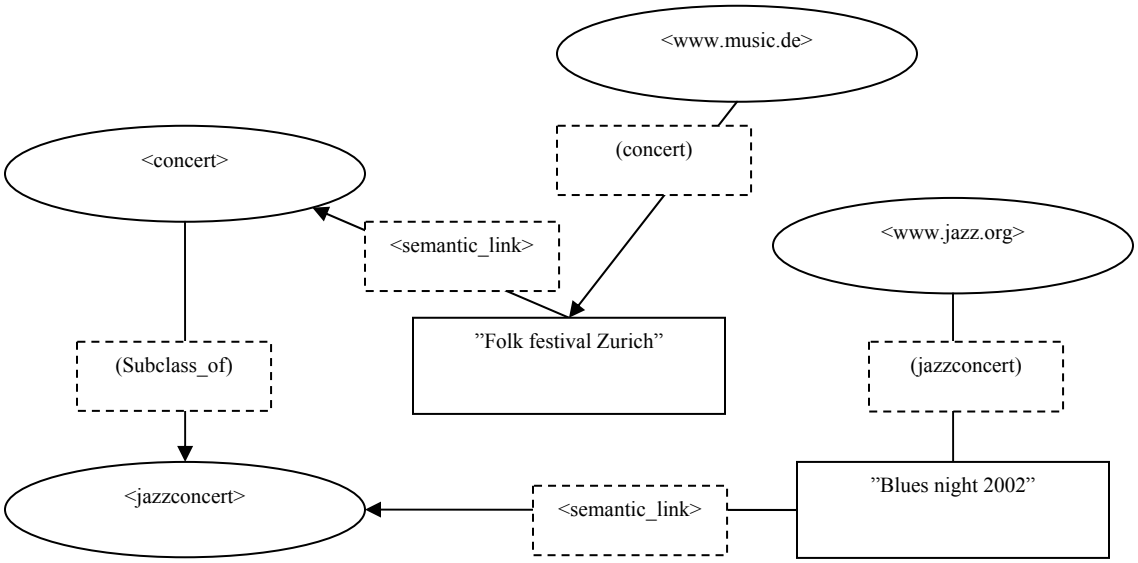


Figure 21: Semantische Instanzen

## Suchanfragen:

Nachfolgend ist eine Liste möglicher sinnvoller Suchanfragen gegeben. Die Liste soll zeigen, welche Suchoperationen durch das semantische Tool ermöglicht werden müssen.

- Kommt der Knoten x oder der Knoten y im Graph vor?
- Wie gross ist die Distanz zwischen Knoten n und m?
- Welche Kanten und Knoten liegen zwischen n und m?
- Auf welche Literale verweisen die Prädikate o, p und q?
- Welcher ontologische Begriff hat am meisten Instanzen?
- Welche Ressource hat am meisten Verbindungen zu anderen Ressourcen?
- Welche Verbindungen führen aus dem Graph heraus?
- Welche Ressource ist mit dem Konzentrat k verbunden?
- Welche Ressourcen haben am meisten gemeinsame Prädikate?
- Welche Prädikate kommen mehrmals vor?
- Wie viele getrennte Graphen wurden eingelesen?
- Wie viele Knoten hat der Graph g?

# Technische Lösung

## *Anforderungen und Ziele*

In den vorangegangenen Kapiteln des Berichtes wurden die allgemeine Situation, das betrachtete System und ein dazupassendes Datenmodell sowie ein geeignetes Lösungskonzept beschrieben. Zusammengefasst ist das Ziel der hier entwickelten technischen Lösung die Schaffung eines Tools, welches semantische Metadaten (unter Benutzung des RDF-Standards) in einer als Graph modellierten Datenstruktur erfasst und auf dieser Datenstruktur verschiedene Operationen wie Suchanfragen, Navigation, Analyse, Browsing und Strukturierung ermöglicht.
Wichtige Anforderungen entstehen aus der Annahme, dass das Tool in späteren Arbeiten weiterentwickelt werden soll. So besteht der Anspruch, dass das Tool konzeptionell und in seiner reellen Ausführung leicht zu verstehen ist und dass es leicht erweitert und ergänzt werden kann (Stichwort: objektorientierte Programmierung). Eine gute Dokumentation ist Teil dieser Anforderungen.
Die Tatsache, dass als Input für ein ausgereiftes Tool grosse Datenmengen zu erwarten sind, erfordert die Skalierbarkeit der technischen Lösung.
Das Python-Tool hat den Namen GRASP erhalten. Dies einerseits in Anlehnung an das englische Verb *grasp* (= begreifen, erfassen, verstehen) und andererseits als Umschreibung für *Graph for the Analysis of Semantically enriched Polymorphous data*.

## *Betrachtungen zum Datenumfang, zur Interkonnektivität und zum Rechenaufwand*

Der Umfang der Daten bei der „semantischen“ Erfassung von Webressourcen wird schnell sehr gross. Der Graph wird auf der Ebene der Metadaten ungeordnet sein, das heisst es gibt keine Hierarchie mit einem „Startknoten“ und also keine Baumstruktur mit einer Wurzel (Ontologien sind hingegen häufig hierarchisch). Es ist aus praktischen Überlegungen ausschliessbar, dass die Knoten total miteinander verknüpft (ein solcher Fall hätte zur Folge, dass auf n Knoten (n*(n-1))/2 Kanten kommen würden. Als grobe Abschätzung ist zu erwarten, dass ein Knoten eins bis zehn Kanten zu anderen Knoten besitzen wird. Der Graph wird gesamthaft betrachtet Bereiche besitzen, welche intern stark verknüpft und nach aussen nur wenige Verbindungen besitzt (vgl. dazu der Abschnitt „Szenarien zur Verknüpfung der Inputdaten“). Es ist auch möglich, dass beim Einlesen eines vorgegebenen Sets an Daten Teilgraphen entstehen, das heisst Graphen, welche gar nicht mit dem Rest der Daten verbunden sind.
Die technische Lösung muss so gestaltet sein, dass für häufige Operationen der Rechenaufwand möglichst gering ist. Eine der häufigsten Operationen wird die Suche nach Knoten sein. Dies kommt daher, dass einerseits bei der Erweiterung bzw. dem schrittweisen Einlesen und verknüpfen von Knoten das bereits Vorhandensein eines Knotens geprüft und für vorhandene Kanten die Verbindung zu den entsprechenden Knoten geschaffen werden muss und da andererseits die meisten denkbaren Suchanfragen die Suche nach Knoten einschliessen werden. Im Datenmodell kann die Suche nach einem Knoten das Traversieren aller im Graphen vorhandener n Knoten bedingen (unter der Berücksichtigung bereits besuchter Knoten). In der programmierten Datenstruktur sind Optimierungen für das rasche Auffinden eines Knotens möglich.

## *Die Datenstruktur*

Eine Übersicht zur Datenstruktur des Python-Programms `grasp.py` ist in der nachfolgenden Abbildung gegeben.

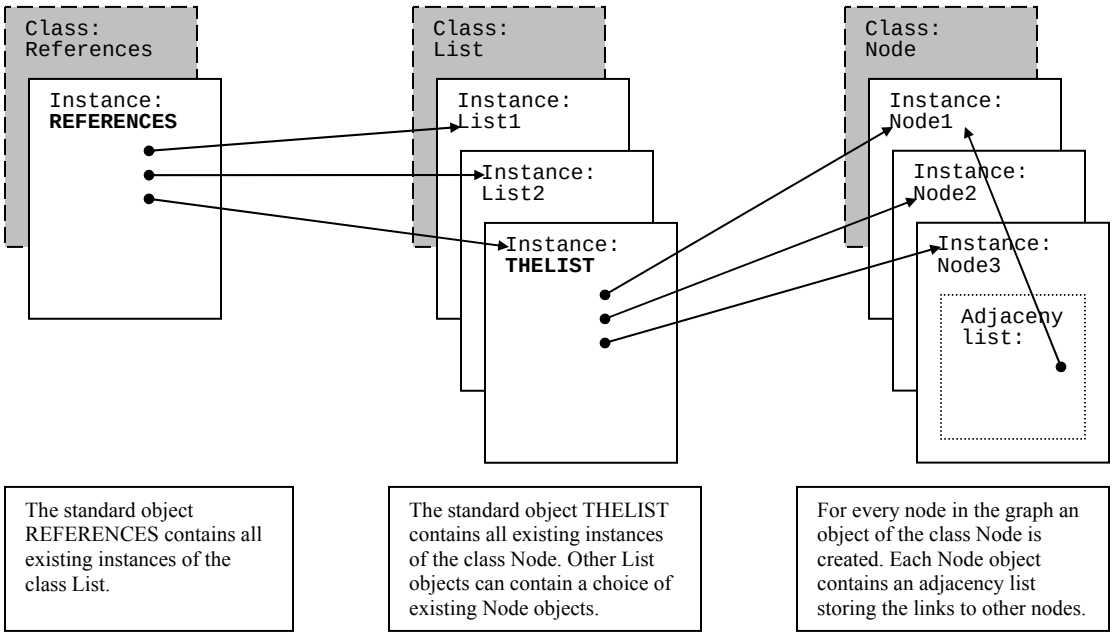


Figure 22: Datenstruktur

Es existieren drei Klassen von Objekten, jede mit klasseneigenen Funktionen und Attributen ausgestattet. Beispielhaft sind in der Abbildung die Klassen und mögliche Instanzen mit grauen respektive weissen Kästchen dargestellt. Referenzen auf andere Objekte sind wiederum beispielhaft durch Pfeile angedeutet.
Das kleinste Objekt ist ein `Node` bzw. Knoten. Für jeden Knoten des Graphen wird ein neues Knotenobjekt mit eindeutiger Identifikationsnummer kreiert. Jedes Knotenobjekt besitzt eine Adjazenzliste, welche die Information über via Kanten (*edges*) direkt verbundene Knoten speichert. Sowohl eingehende als auch wegführende Kanten (Pfeile) werden in dieser Liste gespeichert, wobei ein Code die beiden Fälle unterscheidbar erfasst.
Knoten können in Objekten der Klasse `List` zu Gruppen zusammengefasst werden. Die Gesamtheit aller Knoten ist in einem standardmässig programmierten Listen-Objekt mit der Bezeichnung `THELIST` gespeichert. Alle anderen Listen enthalten also Teilmengen der in `THELIST` enthaltenen bzw. referenzierten Knoten (minimal 1 Knoten). Auf diesen Teilmengen können viele verschiedene Operationen ausgeführt werden ohne alle Daten in die Operation einschliessen zu müssen.
Zwecks Zugriff auf die Listen-Objekte existiert ein standardmässig programmiertes Objekt der Klasse `References` mit der Bezeichnung `REFERENCES`. Nur diese eine Instanz ist für die Zwecke des Tools nötig.

## *Funktionen*

Im Rahmen dieser Arbeit wurden die wichtigsten grundlegenden Funktionen programmiert, welche erste Tests auf manuell bereitgestellten Testdaten ermöglichen und so Schlussfolgerungen zum Nutzen, zu Schwächen und zu möglichen weiteren Entwicklungsschritten des Tools erlauben. In diesem Abschnitt werden die Funktionen erläutert, welche für die Benutzung des Tools notwendig sind. Daneben existieren programmintern weitere Funktionen.

**Einlesefunktion `data_in(x)`:**
Mit dieser Funktion werden die RDF-Metadaten eingelesen und zu einem Graphen verknüpft. Es können so auch bereits eingelesene Daten mit weiteren Daten ergänzt oder erweitert werden.
Als Input dient ein Textfile mit RDF-Daten, welche in der sogenannten *Notation 3* verfasst sind (siehe dazu [b4]). Diese N3-Syntax ist eine im Gegensatz zur in XML integrierten Notation eine für den menschlichen Gebrauch vereinfachte Schreibweise, welche jedoch trotzdem alle Elemente enthält. In der Einlesefunktion enthalten ist ein Parser, welcher die N3-Syntax für die Zwecke des Programms aufschlüsselt und interpretiert. Dieser Parser erfordert die Berücksichtigung einiger Einschränkungen zur Form des N3-Dateninputs (Details dazu sind im Programmcode vermerkt). Das Textfile wird zuerst zeilenweise eingelesen und aufgeschlüsselt. Danach werden die einzelnen Elemente der RDF-statement, das heisst die Subjekt-Prädikat-Objekt-Triples, in den Graphen eingelesen.
Falls erneut Daten beispielsweise aus einem zweiten Textfile eingelesen werden, so werden bereits vorhandene Knoten nicht ein zweites Mal kreiert.
Als Abschluss der Einlesefunktion werden dem Benutzer grundlegende Informationen zum Graphen wie beispielsweise die Zahl der Knoten etc. angezeigt.

**Knoten-Suchfunktion `find_node(x)`:**
Mit dieser Funktion kann nach eingelesenen Knoten gesucht werden. Als Input dient ein Textstring. Für jeden eingelesenen Knoten, in dessen Label dieser Textstring enthalten ist, werden dessen Identifikationsnummer und das Label angezeigt.

**Verbindungs-Suchfunktion `find_connection(x)`:**
Mit dieser Funktion kann nach Verbindungen zwischen zwei Knoten gesucht werden. Als Input dienen zwei Identifikationsnummern von Knoten, zwischen denen Verbindungen gesucht werden sollen. Besteht eine Verbindung so wird eine Liste all derjenigen Knoten kreiert, die Teil einer solchen Verbindung sind. Als pragmatischer Ansatz werden dabei per Standard nur solche Verbindungen berücksichtigt, deren Distanz nicht mehr als zwei Schritte (*hops*) grösser ist als diejenige der kürzesten Verbindung. Ohne Beschränkung der Verbindungslänge sind alle Knoten eines Graphen Teil einer Verbindung. Die Spanne der Verbindungen, welche bei einer Suche erfasst werden sollen, kann selbst bestimmt werden.
Der Iterationsalgorithmus entspricht einer *breath first search* (vgl. dazu [d11]). Die Liste der so identifizierten Verbindungsknoten kann für weitere Bearbeitungsschritte benutzt werden, also beispielsweise über die Ausgabefunktion betrachtet werden.

**Statistikfunktion `statistics(x)`:**
Mit dieser Funktion können einfache Informationen über ein Listenobjekt angezeigt werde. Dies sind die Anzahl enthaltener Knoten, die Anzahl interner Kanten und die Anzahl nach aussen gehender Kanten. Als Input wird ein Listen-Objekt übergeben.

**Ausgabefunktion `data_out(x)`:**

Mit dieser Funktion werden die Daten eines Listen-Objekts in ein Outputfile geschrieben. Als Input wird ein Listen-Objekt übergeben. Ein Outputfile mit der Bezeichnung `output.n3` wird geschrieben beziehungsweise überschrieben. Dieses File enthält in N3-Syntax zeilenweise die RDF-Triples zu allen Knoten und internen Kanten, welche im Listen-Objekt referenziert sind. Da das Listen-Objekt auch nur einen Teil aller Knoten enthalten kann, bietet sich so die Visualisierung von Teilgraphen an.
Auch möglicherweise vorhanden Deklarationen von Abkürzungen (*prefixes*) in der Notation 3-Schreibweise (vgl. dazu [b4]) werden in das Outputfile geschrieben.
Hauptzweck der Ausgabefunktion ist die Nutzung des Outputfiles durch das vom W3C (World Wide Web Consortium) entwickelte *IsaViz*-Visualisierungtool. Der Import in jenes Tools erfolgt im Menü über `File`, dann `Import`, dann `Replace` und dann `Notation 3 from file` sowie anschliessend die Wahl des zu importierenden Files über das erscheinende Menü. Die nächste Abbildung zeigt einen *Screenshot* des *IsaViz* Programms.

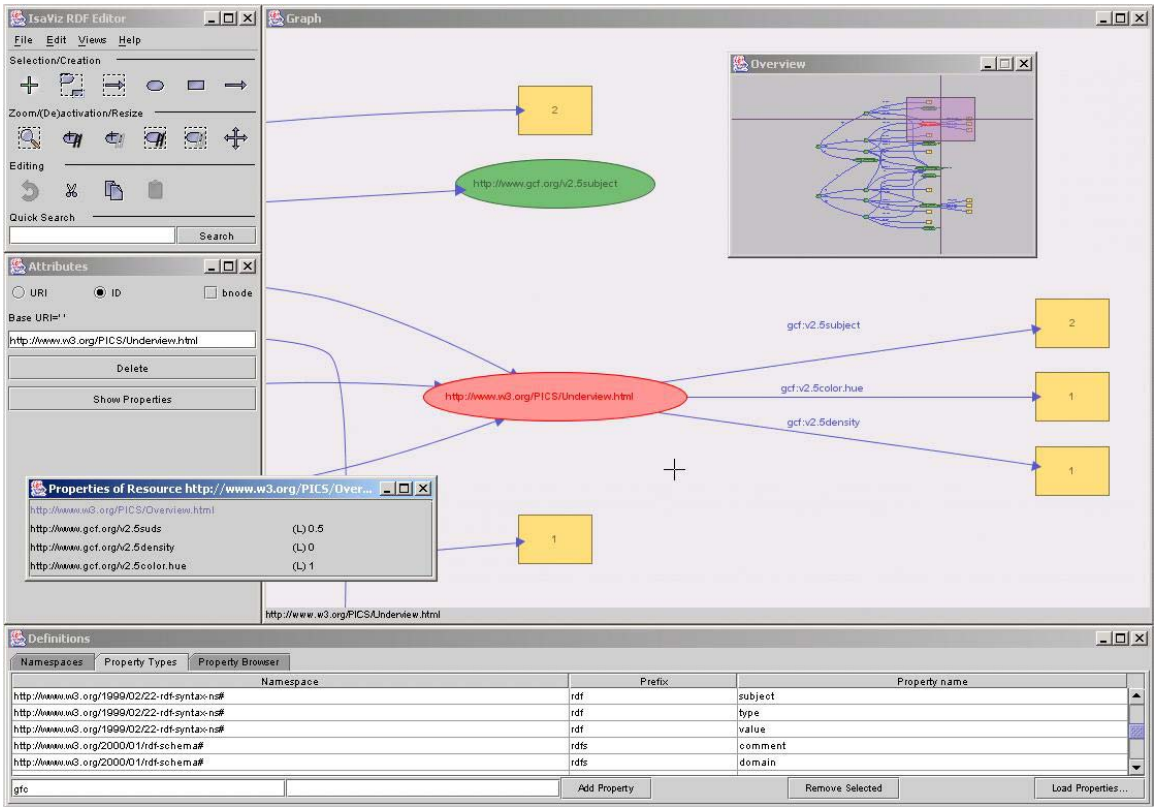

Figure 23: Screenshot des IsaViz Visualisierungstools

## Testdaten

Um das Tool zu testen und seine Funktionen veranschaulichen zu können, wurde ein Set an Testdaten erstellt. Diese Testdaten wurden manuell in Form von N3-Inputfiles angefertigt, wobei neben eigentlichen Metadaten und auch dazupassende ontologische Daten erstellt wurden. Das Testbeispiel stellt einen Konzertveranstaltungskalender dar, wie man ihn häufig bei Online-Zeitungen (bsp. Züritipp) findet. Dieses Beispiel wurde gewählt, da einerseits Konzerte gut geeignet sind, um Verbindungen zu Komponisten, Bands, Musikaufnahmen etc. zu modellieren und andererseits da die zeitliche Dynamik eines Veranstaltungskalenders das Auffinden der darin enthaltenen Daten erschwert, da eine herkömmliche Suchmaschine seinen Suchindex nicht genug

häufig aktualisiert. Für solche dynamischen Datenressourcen ist daher der semantische Ansatz besonders geeignet, da zumindest die Kategorien Konzert, Theaterstück, Lokalität, Aufführungsdatum als (semantische) Kategorien fix bestehen bleiben.
Im gewählten Beispiel wurde in Anlehnung an www.zueritipp.ch davon ausgegangen, dass der Zugriff auf die Daten über die Kategorien Konzerttitel, Aufführungslokal oder Aufführungsdatum erfolgen kann. Was wie unten ersichtlich als Tabelle sehr einfach daherkommt, ist im Graph-Modell der RDF-Daten bereits ein kleines Netzwerk von Knoten und Kanten. Liest man zusätzlich die ontologischen Daten (in Anlehnung an Ontosaurus [c2] manuell erstellt) mit den Definitionen der vorkommenden Begriffe ein, so entsteht schon ein unübersichtliches Gewirr an Knoten und Verbindungen.

| Konzerttitel | Aufführungslokal | Aufführungsdatum |
|---|---|---|
| Porgy & Bess | Tonhalle | 19.06.2003 |
| Blues Night | Moods | 25.07.2003 |
| Avalon Trio | The Club | 27.06.2003 |
| Bach Serenade | Predigerkirche | 05.07.2003 |

Table 4: Konzertkalenderdaten

## *Evaluation*

Anbei werden die Stärken und Schwächen der technischen Lösung, also des GRASP-Tools, aufgelistet.

Stärken:

- Alle wichtigen Funktionen vorhanden
- Im Aufbau einfach strukturiert, leicht verständlich
- Visualisierung von Daten über *IsaViz* möglich
- Erste Testdaten vorhanden

Schwächen:

- Konzeptionell im Datenmodell vorgesehene Unterscheidung der verschiedenen Datenkategorien noch nicht ausgereift bzw. implementiert
- Die Einlesefunktion (Parsing) für N3-Inputdaten ist nicht benutzerfreundlich: Fehler zu detektieren, die durch syntaktisch fehlerhafte Inputdaten entstehen, ist aufwändig
- Begrenzter Umfang der Testdaten und der verfügbaren Funktionen

Gesamthaft gesehen wird durch das GRASP-Tool sowie die dazugehörige konzeptionelle Arbeit und Dokumentation der Grundstein für weitere Arbeiten zum Thema Semantic Web und *Volatile Content Search* gelegt.
Das nächste Kapitel zeigt im Ausblick, dass es noch offene Fragen zu klären gilt. Eine fundierte Beurteilung zum Nutzen und der Anwendbarkeit des semantischen Ansatzes kann erst nach weiteren Untersuchungen mit umfangreicheren Testdaten gemacht werden.

# Offene Fragen, mögliche nächste Schritte und Ausblick

Vorläufig wurde nur mit einfachen Beispielen und kleinen Datenmengen gearbeitet. Sind semantische Metadaten auch bei grösseren Datenmenge handhabbar und ist RDF auch für kompliziertere Beispiele geeignet?
In der hier erarbeiteten Lösung werden die Daten über das *IsaViz* Tool visualisiert. Schon bei begrenztem Umfang der Daten wird der Graph unübersichtlich. Kann die Visualisierung verbessert werden, um den raschen Zugriff auf Daten und Informationen für den Menschen zu erleichtern?
In GRASP sind die Suchmöglichkeiten sehr beschränkt. Ausserdem ist absehbar, dass bei semantischen Metadaten die herkömmliche textbasierte Suche ungeeignet ist. Gibt es eine an RDF angepasste Suchmethodik, welche dem Benutzer ermöglicht rasch und zielgerichtet Informationen zu finden?
Da grosse Datenmengen für den Menschen nur schwer handhabbar sind, muss es das Ziel sein solche Daten auf geeignete Weise zu strukturieren (*clustering*). Gibt es geeignete Möglichkeiten, um semantische Metadaten so zu strukturieren, dass sich ein Mensch rasch einen Überblick verschaffen kann?
Es ist klar, dass ein zukünftiges Tool zur Erfassung und Analyse von semantischen Metadaten direkt ans Web angeschlossen sein muss. Als nächster Entwicklungsschritt ist es deshalb denkbar, das Tool so zu erweitern, dass RDF-Daten direkt in der XML-Schreibweise eingelesen und verarbeitet werden können sowie dass über eine Webanbindung Daten direkt vom Web geladen werden können.
Es bleibt wichtig die Aktivitäten in der Forschung zu beobachten, um nicht unnötig Energien in die Lösung von Problemen zu stecken, die bereits gelöst wurden. Im Abschnitt „Existierende Lösungen und Aktivitäten im Bereich des Semantic Web“ wurde auf das sehr aktive Geschehen an der Forschungsfront verwiesen und es wurden vier ähnlich wie GRASP ausgerichtete Tools herausgegriffen.
Als Ergebnis der vielen Arbeiten und Beiträge entsteht vielleicht schon in den nächsten Jahren ein „Sweb“, das mit neuen semantischen Funktionen ausgestattet ist. Hoffentlich kann damit die Effizienz bei der Informationssuche und Datenverarbeitung gesteigert werden und hoffentlich steht damit den Menschen weiterhin ein Werkzeug zur Verfügung, welches die Neugier und den Spass im Umgang mit Wissen fördert!

# Anhang

## *Wichtige Internetquellen*

Allgemeine Informationen:

- [a1] http://www.semanticweb.org - The Semantic Web Community Portal
- [a2] http://ontoweb.aifb.uni-karlsruhe.de/ - Ontoweb, a European Union founded project about Ontology-based information exchange for knowledge management and electronic commerce
- [a3] http://www.w3.org/RDF/ - Resource Description Framework (RDF) / W3C Semantic Web Activity

RDF-Tutorials:

- [b1] http://www710.univ-lyon1.fr/~champin/rdf-tutorial/ - RDF-Tutorial
- [b2] http://www.dstc.edu.au/Research/Projects/rdf/RDF-Idiot.html - An Idiot's Guide to the Resource Description Framework
- [b3] http://www.ilrt.bris.ac.uk/discovery/rdf/resources/ - Dave Beckett's Resource Description Framework (RDF) Resource Guide
- [b4] http://www.w3.org/2000/10/swap/Primer - Primer: Getting into RDF & Semantic Web using N3 (Notation 3)

Ontologien:

- [c1] http://www.ontology.org/main/papers/madrid-tutorials.html - Tutorials and Introductions to Ontologies, Ontological Engineering and their Application
- [c2] http://mozart.isi.edu:8003/sensus2/ - SENSUS Ontosaurus Ontology
- [c3] http://www-ksl.stanford.edu/knowledge-sharing/ontologies/html/frame-ontology/index.html - Theory to the FRAME-ONTOLOGY
- [c4] http://www.w3.org/2000/01/rdf-schema - W3C RDF-schema
- [c5] http://dublincore.org/2003/03/24/dces# - Dublincore ontology

Andere interessante Internetquellen im Rahmen des Themas:

- [d1] http://www.netvalley.com/intvalstat.html - History of Internet and WWW
- [d2] http://searchengineshowdown.com - Search Engine News
- [d3] http://www.vivisimo.com – Vivisimo Document Clustering
- [d4] http://www.w3.org/RDF/Validator/ - W3C RDF Validation Service
- [d5] http://www.w3.org/2001/11/IsaViz/ - IsaViz: A Visual Authoring Tool for RDF
- [d6] http://www.topicmap.com/topicmap/resources.html - Resources on Topic Maps
- [d7] http://www.ontopia.net/topicmaps/materials/rdf.html - Ten Theses on Topic Maps and RDF
- [d8] http://www.ontoknowledge.org/index.shtml - Content-driven Knowledge-Management through Evolving Ontologies
- [d9] http://www.ontoprise.de/company/publications - Ontoprise GmbH - publications
- [d10] http://www.topquadrant.com/ - Knowledge Systems Architects

- [d11] http://www.pri.univie.ac.at/~schiki/unterlagen/GZ1/Kapitel8/ppframe.htm - Präsentation zu Graphen
- [d12] http://www.ontoknowledge.org/oil/downl/IEEE00.pdf - The Semantic Web – on the repectives Roles of XML and RDF
- [d13] http://www.outsights.com/systems/dikw/dikw.htm - Bellinger: Data, Information, Knowledge and Wisdom
- [d14] http://review.software.ibm.com/developer/library/tutorial-prog/overview.html - XML Tutorial for Programmers
- [d15] http://www.aifb.uni-karlsruhe.de/AIK/veranstaltungen/aik9/presentations/slides/020419FutureSemanticWeb.pdf - Presentation "The Future of the Semantic Web"
- [d16] http://pespmc1.vub.ac.be/ASC/REIFICATION.html - on Reification
- [d17] http://ioctl.org/rdf/usementionmyarse - on Reification

## *Literaturverzeichnis*


[1] Berners-Lee, Timm (2001), "The Semantic Web", in Scientific American

[2] Lyre, Holger (2002), „Informationstheorie: eine philosophisch-naturwissenschaftliche Einführung", Wilhelm Fink Verlag, München

[3] V. Geroimenko und Ch. Chen (Eds.) (2003), „Visualizing the semantic web: XML-based internet and information visualization", Springer-Verlag, London

[4] Gomez and Benjamins (Eds.) (2002), "Knowledge Engineering and Knowledge Management" (Papers of the 13th EKAW Conference 2002), Springer-Verlag, Berlin

[4.1] Maedche, Motik, Silva and Volz, "MAFRA – A Mapping FRAmework for distributed Ontologies"

[5] Bussler, Hull et al. (Eds.) (2002), "Web Services, E-Business, and the Semantic Web" (Papers of the 2002 CaiSE Workshop), Springer-Verlag, Berlin


## *Liste der elektronischen Files*

- grasp.py – in Python programmiertes zentrales Tool zur Nutzung semantischer Metadaten
- grasp.n3 – ontologisches Vokabular mit eigenen Begriffsdefinitionen wie `semantic_link`, `concentrate`, etc. (in RDF-„Notation 3")
- onto_concert.n3 – Schema mit Begriffen wie `concert`, `date` und `title`; erstellt in Anlehnung an die Ontosaurus Ontologie
- zhtipp.n3 – Metadaten Beispiel mit Konzertkalenderinformationen („Züritipp")

## *Angefügte Dokumente*

- Ursprüngliche Aufgabenstellung