14. Semantische Mashups auf Basis Vernetzter Daten

Sören Auer[1], Jens Lehmann[1] und Christian Bizer[2]

[1] Institut für Informatik (IfI), Universität Leipzig;
nachname@informatik.uni-leipzig.de
[2] Web-based Systems Group, Freie Universität Berlin;
chris@bizer.de

Zusammenfassung: Semantische Mashups sind Anwendungen, die vernetzte Daten aus mehreren Web-Datenquellen mittels standardisierter Datenformate und Zugriffsmechanismen nutzen. Der Artikel gibt einen Überblick über die Idee und Motivation der Vernetzung von Daten. Es werden verschiedene Architekturen und Ansätze zur Generierung von RDF-Daten aus bestehenden Web 2.0-Datenquellen, zur Vernetzung der extrahierten Daten sowie zur Veröffentlichung der Daten im Web anhand konkreter Beispiele diskutiert. Hierbei wird insbesondere auf Datenquellen, die aus sozialen Interaktionen hervorgegangen sind eingegangen. Anschließend wird ein Überblick über verschiedene, im Web frei zugängliche semantische Mashups gegeben und auf leichtgewichtige Inferenzansätze eingegangen, mittels derer sich die Funktionalität von semantischen Mashups weiter verbessern lässt.

Einleitung

Das Web wandelt sich zunehmend von einem Medium zur Veröffentlichung von Texten hin zu einem Medium zur Veröffentlichung von strukturierten Daten. Ein Beispiel für diese Entwicklung ist die zunehmende Verbreitung von Inhaltsformaten wie RSS, ATOM und Microformats, sowie Web-APIs, die Anfragen gegen Datenquellen wie Google, Yahoo! und Amazon ermöglichen.

Aufgrund der diversen Schnittstellen und Ergebnisformate, die derzeit von Web-APIs angeboten werden, ist die Integration von Daten aus mehreren Datenquellen nach wie vor mit einem relativ hohen Programmieraufwand verbunden.

In diesem Beitrag stellen wir das Konzept semantischer Mashups als Ansatz zur Integration von Daten aus unterschiedlichen Quellen vor.

Semantische Mashups sind Anwendungen, die vernetzte RDF-Daten aus mehreren Web-Datenquellen nutzen. Das Spektrum semantischer Mashups ist groß. Es reicht von generischen Daten-Browsern, über themenspezifische Portale bis hin zu Suchmaschinen, die expressive Anfragen gegen Web-Daten aus unterschiedlichen Quellen ermöglichen.

Im Konzept semantischer Mashups im Social Semantic Web spielen drei Elemente eine zentrale Rolle: (1) Vernetzte Daten (engl. Linked Data [7]) als Rahmenwerk für die Repräsentation und den Zugriff auf semantische Daten im Web, (2) DBpedia als ein Kristallisationskern für die Vernetzung von Daten aus unterschiedlichen Quellen sowie (3) leichtgewichtige Inferenzstrategien für die Integration und Strukturierung der Daten.

Im Gegensatz zu den diversen Schnittstellen und Ergebnisformaten, die derzeit von Web-APIs angeboten werden, bieten vernetzte Daten den Vorteil eines flexiblen, standardisierten Datenformats (RDF), eines standardisierten Zugriffsmechanismus (HTTP) sowie die Möglichkeit Verweise zwischen Daten in unterschiedlichen Datenquellen zu setzen. Anhand dieser Links und eines generischen Daten-Browsers können Nutzer von Datensätzen in einer Datenquelle zu Datensätzen in einer anderen Datenquelle navigieren. Verweise können auch von Suchmaschinen verwendet werden, um die Inhalte vernetzter Web-Datenquellen zu sammeln (mittels Web-Crawler) sowie expressive Abfrage- und Suchfunktionalitäten über die gesammelten Daten anzubieten.

Die Enzyklopädie Wikipedia beinhaltet Informationen zu einer sehr breiten Palette unterschiedlicher Themen. Das Projekt DBpedia extrahiert strukturierte Informationen aus Wikipedia und veröffentlicht diese Informationen als vernetzte Daten im Web. DBpedia bietet derzeit Informationen zu mehr als 1,95 Millionen ‚Dingen', inklusive 80.000 Personen, 70.000 Orten, 35.000 Musik-Alben, 12.000 Filmen. Durch die breite thematische Abdeckung lassen sich DBpedia-Daten mit Daten aus einer Vielzahl anderer Datenquellen verknüpfen. Hierdurch und durch die Vernetzung anderer Datenquellen untereinander entwickelt sich derzeit ein dezentrales Daten-Web. Dieses Daten-Web wuchs im Laufe des letzten Jahres sehr schnell und umfasste im Oktober 2007 mehr als 2 Milliarden Informationen (RDF Triples).

Dieser Beitrag gliedert sich wie folgt: Wir stellen zunächst die Basis-Prinzipien vernetzter Daten vor (Abschn. 2), beschreiben am Beispiel von DBpedia, wie eine Fülle strukturierter Daten mittels sozialer Interaktionen gewonnen werden kann (Abschn. 3). Anschließend wird ein Überblick über verschiedene, im Web frei zugängliche semantische Mashups gegeben sowie auf Inferenzansätze eingegangen, mittels derer sich die Funktionalität der semantischen Mashups weiter verbessern lässt.

Vernetzte Daten im Web

Dieser Abschnitt erklärt die technischen Grundlagen vernetzter Daten und gibt einen Überblick über Werkzeuge zur Veröffentlichung vernetzter Daten im Web. Anschließend werden verschiedene, existierende Quellen vernetzter Daten vorgestellt.

Die Prinzipien vernetzter Daten

Das Konzept vernetzter Daten greift verschiedene Entwicklungen aus dem Bereich webbasierter Datenintegration auf und versucht die unterschiedlichen Entwicklungstendenzen auf einer gemeinsamen technologischen Basis zusammenzuführen. Das Konzept zielt darauf ab, es Informationsanbietern genau so einfach zu machen, strukturierte Daten im Web zu veröffentlichen und Datensätze in unterschiedlichen Datenquellen zu verknüpfen, wie sie heute klassische HTML-Dokumente im Web veröffentlichen und Verweise zwischen verschiedenen Dokumenten setzen.

Der Begriff *Vernetzte Daten* (engl. Linked Data) wurde von Tim Berners-Lee in [7] geprägt. Der Begriff bezieht sich auf eine Menge von Best-Practices zur Veröffentlichung und Verknüpfung von strukturierten Daten im Web. Grundannahme von Linked Data ist, dass der Wert und die Nützlichkeit von Daten steigen, je stärker sie mit Daten aus anderen Datenquellen verknüpft sind.

Die technischen Grundprinzipien vernetzter Daten bestehen darin,

1. das Resource Description Framework (RDF) [19] als universelles Datenmodell zur Veröffentlichung von strukturierten Daten im Web zu verwenden,
2. alle URIs, die in RDF-Graphen verwendet werden, über das Web dereferenzierbar zu machen, sowie
3. RDF-Verweise zwischen Daten in verschiedenen Datenquellen zu setzen.

Die Anwendung dieser beiden Prinzipien führt zur Entstehung eines Daten-Webs, eines offenen Informationsraums mit ähnlichen Eigenschaften wie denen des klassischen World Wide Webs.

Auf diesen Informationsraum lässt sich mittels generischer Daten-Browser zugreifen, ähnlich, wie auf das klassische Web mittels HTML-Browsern zugegriffen wird. Nutzer können sich Daten aus einer Quelle anzeigen lassen und anschließend anhand von RDF-Verweisen zu Daten in einer anderen Quelle navigieren. Ähnlich wie Suchmaschinen das klassische Web anhand von HTML-Verweisen crawlen, lassen sich anhand von RDF-Verweisen Daten aus verschiedenen Quellen zusammenführen.

RDF-Verweise [11] verknüpfen Datensätze aus unterschiedlichen Datenquellen. Sie repräsentieren typisierte Beziehungen zwischen den Datensätzen. So lässt sich beispielsweise mittels RDF-Verweisen ausdrücken, dass mehrere Datensätze in unterschiedlichen Datenquellen die gleiche Person beschreiben. Oder es lässt sich ausdrücken, dass eine Person, die in einer Datenquelle beschrieben wird, sich für ein Thema interessiert, über das es in einer anderen Datenquelle weitere Informationen gibt.

Der folgende RDF Graph besteht aus zwei RDF-Verweisen. Das Beispiel verwendet die Turtle Syntax [5].

```
1. # RDF link taken from Tim Berners-Lee's FOAF profile
2. <http://www.w3.org/People/Berners-Lee/card#i>
3. owl:sameAs
4. <http://www4.wiwiss.fu-
berlin.de/dblp/resource/person/100007>.
5.
6. # RDF link taken from Richard Cyganiaks's FOAF profile
7. <http://richard.cyganiak.de/foaf.rdf#cygri>
8. foaf:based_near
9. <http://dbpedia.org/resource/Berlin>.
```

Abb. 1. Beispiele für RDF-Verweise

Der erste RDF-Verweis (Zeile 2–4) drückt aus, dass die URI `http://www.w3.org/People/Berners-Lee/cardi` die gleiche Ressource identifiziert wie die URI `http://www4.wiwiss.fu-berlin.de/dblp/resource/person/100007`. Der zweite RDF-Verweis (Zeile 7–9) drückt aus, dass die URI `http://richard.cyganiak.de/foaf.rdfcygri` eine Ressource identifiziert, die in der Nähe einer anderen Ressource wohnt, welche mittels der URI `http://dbpedia.org/resource/Berlin` identifiziert wird.

Gemäß der Prinzipien vernetzter Daten sollen alle URIs, die in RDF-Graphen verwendet werden, dereferenzierbar sein. Dies bedeutet, dass Web-Clients zu jeder URI mittels der HTTP-Operation GET weitere Informationen abrufen können. Sendet ein Web-Client zusammen mit seiner Anfrage den HTTP-Accept-Header `text/html`, sendet ihm der Server eine Repräsentation der Ressource im HTML-Format. Fragt der Client mittels des HTTP-Accept-Headers `application/rdf+xml` nach RDF-Daten, sendet ihm der Server Informationen über die Ressource im RDF/XML Format.

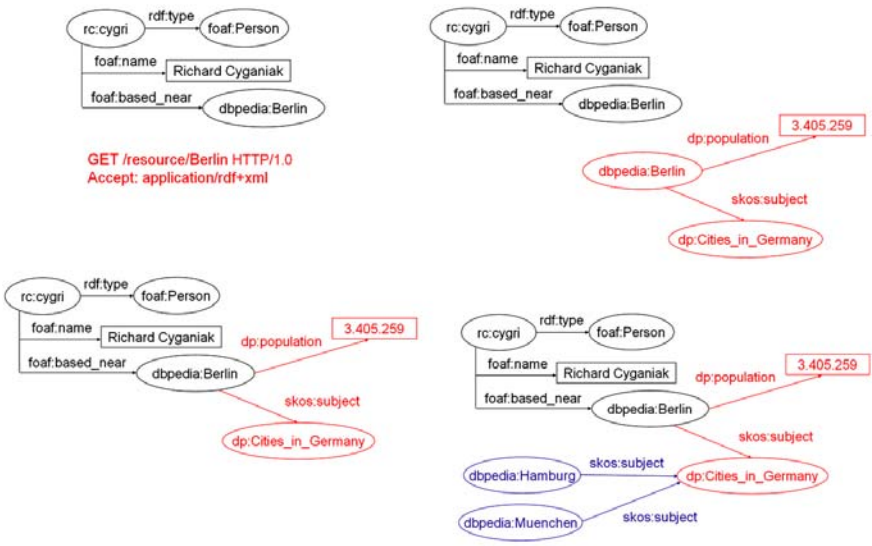

Abb. 2. Dereferenzierung von HTTP-URIs

Das folgende Beispiel illustriert, wie Web-Clients mittels URI-Dereferenzierung durchs Daten-Web navigieren. Interessiert sich der Benutzer eines Daten-Browsers beispielsweise dafür, was für eine Ressource mit der URI `http://richard.cyganiak.de/foaf.rdfcygri` identifiziert wird, weist er seinen Browser an, diese Ressource zu dereferenzieren. Als Antwort erhält er vom Server `http://richard.cyganiak.de` beispielsweise den in Abb. 2 links oben dargestellten RDF Graphen. Der Graph enthält die Information, dass es sich bei der Ressource um eine Person handelt, die Richard Cyganiak heißt. Interessiert sich der Benutzer darüber hinaus für den Ort, in dem Richard wohnt, dereferenziert er die URI `http://dbpedia.org/resource/Berlin` und bekommt vom Server `http://dbpedia.org` einen RDF-Graphen, der die Stadt Berlin beschreibt. Da sowohl der Graph über Richard als auch der Graph über Berlin die gleiche URI zur Identifikation der Stadt Berlin verwenden, fügen sich beide Graphen natürlich zusammen (Abb. 2 rechts oben und links unten). Der Graph, der Berlin beschreibt, beinhaltet die Information, dass Berlin zur Gruppe der deutschen Städte gehört. Interessiert sich der Nutzer für weitere deutsche Städte, dereferenziert er die URI `http://dbpedia.org/resource/Cities_in_Germany` und erhält eine Liste deutscher Städte, von der aus er zu weiteren Städten navigieren kann (Abb. 2 rechts unten).

Veröffentlichung vernetzter Daten im Web

Vernetzte Daten lassen sich im Web in Form von RDF/XML-Dateien, die auf einem Webserver abgelegt werden, veröffentlichen. Im Laufe des letzten Jahres wurden zusätzlich verschiedene Werkzeuge zur Veröffentlichung vernetzter Daten entwickelt, mit Hilfe derer sich Sichten auf die Inhalte relationaler Datenbanken und RDF-Stores im Web publizieren lassen. Beispiele derartiger Veröffentlichungs-Werkzeuge für relationale Datenbanken sind D2R-Server [8] und OpenLink Virtuoso [14]. Ein Tool, mit dem sich die Inhalte von RDF-Datenbanken publizieren lassen, ist Pubby [13].

Ein weiterer Ansatz zur Veröffentlichung vernetzter Daten besteht in der Implementierung von Wrappern um existierende Anwendungen oder Web-APIs. Beispiele für Wrapper auf Anwendungsebene sind die SIOC-Exporter für WordPress, Drupal und phpBB [12]. Ein Beispiel eines Wrappers um Web-APIs ist das RDF Book-Mashup, das auf die Amazon und Google Base-API zugreift, um RDF-Daten über Bücher bereitzustellen [10].

Eine detaillierte Beschreibung der verschiedenen Techniken zum Publizieren vernetzter Daten im Web findet sich in [11].

Quellen vernetzter Daten im Web

Die Menge der im Oktober 2007 als vernetzte Daten im Web veröffentlichten Informationen wird auf über 2 Milliarden RDF-Triples geschätzt. Dieser Datenbestand ist mittels circa 3 Millionen RDF-Verweisen zwischen unterschiedlichen Datenquellen vernetzt.

Unterschiedliche Datenquellen veröffentlichen Informationen über Länder, Städte, Personen, Firmen, Bücher, Filme, Musik, wissenschaftliche Veröffentlichungen, Konferenzen, Projekte und Arbeitsgruppen, sowie statistische Daten über die Europäische Gemeinschaft und die Vereinigten Staaten.

Das Linking Open Data Projekt [26] der Semantic Web Education and Outreach Arbeitsgruppe des Word Wide Web Konsortiums führt ein Verzeichnis aller bekannten Quellen vernetzter Daten, die ihre Inhalte unter einer offenen Lizenz bereitstellen. Abbildung 3 gibt einen Überblick über einen Teil der vom Linking Open Data Projekt erfassten Datenquellen sowie über die Verknüpfungen zwischen Datensätzen in unterschiedlichen Datenquellen.

Die Datenquellen, die im Oktober 2007 neu in das Verzeichnis aufgenommen worden sind, sind in der Abbildung mit ‚NEW' gekennzeichnet. Die Datenquelle Wiki-Company bietet beispielsweise Informationen über circa 20.000 Firmen an. Der Flickr Wrappr ermöglicht den Zugriff auf die Fotodatenbank von Flickr[1] und generiert Bildersammlungen zu Dingen, die

[1] www.flickr.com/

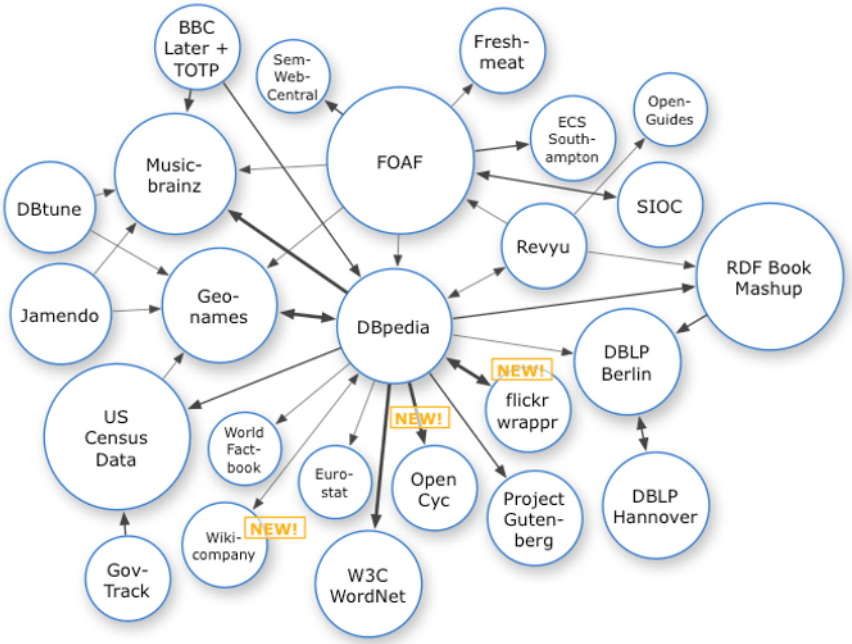

Abb. 3. Überblick über Quellen vernetzter Daten

in DBpedia beschrieben sind. Die Datenquelle Open-Cyc veröffentlicht einen Teil der Cyc-Ontologie[2] als vernetzte Daten.

Anhand von RDF-Verweisen können Nutzer zwischen Datensätzen in unterschiedlichen Datenquellen des Netzwerks navigieren. So ist es beispielsweise möglich, von Informationen über eine Person in einer FOAF-Datei zu Informationen über die Stadt, in der die Person wohnt, in DBpedia und anschließend zu weiteren Informationen über diese Stadt in der Geonames- sowie der US-Census-Datenbank zu navigieren. Alternativ lässt sich in DBpedia beispielsweise von der Stadt zu einer Liste von Musikgruppen, die aus dieser Stadt stammen, und anschließend zu detaillierten Informationen über die Alben dieser Bands in der MusicBrainz-Datenbank navigieren.

DBpedia als Kristallisationskern für die Vernetzung von Datenquellen

Um Daten aus thematisch unterschiedlichen Datenquellen miteinander verknüpfen zu können, ist es sehr wichtig über Datensätze zu verfügen, die

[2] http://www.opencyc.org/

ein sehr breites Themenspektrum abdecken sowie Dinge aus unterschiedlichen Domänen miteinander in Beziehung setzen. Einer der größten Datensätze dieser Art wird zurzeit vom DBpedia-Projekt bereitgestellt. Die Bedeutung dieses Datensatzes für die Verknüpfung anderer Datenquellen verdeutlicht die zentrale Stellung von DBpedia in Abb. 3.

Die DBpedia Daten werden aus den Inhalten der Enzyklopädie Wikipedia extrahiert. Seit dem Beginn des Wikipedia-Projektes im Jahr 2001 hat sich Wikipedia zur umfassendsten Enzyklopädie und dem erfolgreichsten kollaborativen Gemeinschaftsprojekt im Internet entwickelt. Inzwischen gibt es Wikipedia-Versionen in mehr als 250 Sprachen. Im September 2004 überschritt der Umfang des Gesamtprojekts die Grenze von einer Million Artikel, mittlerweile sind es über 7,5 Millionen. Die deutschsprachige Wikipedia ist dabei eines der aktivsten und bestkoordinierten Teilprojekte. Sie enthält derzeit mehr als 655.000 Artikel, die englische Ausgabe umfasst über 2,07 Millionen (Stand: November 2007). Inzwischen ist Wikipedia eine der 10 am meisten besuchten Informationsangebote im Internet (vgl. alexa.com).

Eines der Wikipedia-Grundprinzipien ist die gemeinschaftliche Erstellung der Artikel. Dies resultiert in einer Reihe von Vor- und Nachteilen. Zu den Vorteilen gehört, dass Artikel oft einen Konsens repräsentieren, die Mitarbeit und Beteiligung angeregt wird und eine große Bandbreite von Themen umfassend und oft enorm aktuell abgedeckt wird. Folgende Nachteile des kollektiven Editierens haben sich ergeben: Manche (Rand-) Themen sind ungenau oder unvollständig dargestellt und dies ist für Leser nicht immer ersichtlich, einige Nutzer versuchen in Wikipedia einseitige oder werbende Darstellungen zu platzieren, die Inhalte sind nicht einheitlich strukturiert. Diese Nachteile wurden von der Wikipedia-Gemeinschaft inzwischen erkannt und es wird versucht Lösungsmöglichkeiten zu entwickeln: Stabile Artikelversionen sollen vor Vandalismus schützen, Artikel werden zunehmend kategorisiert und (z. B. mit Infoboxen) strukturiert, es gibt Wettbewerbe für die besten Beiträge und schlecht geschriebene Artikel und solche mit fehlenden Informationen werden entsprechend gekennzeichnet.

Das DBpedia-Projekt versucht nun strukturierte Informationen aus Wikipedia-Inhalten (z. B. Infoboxen) zu extrahieren. In diesen strukturierten Informationen liegt ein sehr großes Potential, das heute noch nicht genutzt wird. Durch die Extraktion von Informationen aus Wikipedia und deren Repräsentation mittels eines strukturierten Datenmodells lassen sich z. B. folgende Anwendungen realisieren:

- Es lassen sich komplexe Anfragen an Wikipedia stellen. Beispiele sind: „Welche deutschen Komponisten wurden im 18. Jahrhundert in Berlin geboren?“, „In welchen Filmen tritt Quentin Tarantino als Schauspieler auf?“ oder „Wer sind die Bürgermeister von Städten in den USA, die hö-

her als 1000m gelegen sind?“. Diese erweiterten Anfragemöglichkeiten revolutionieren den Zugriff auf Wikipedia-Inhalte für den Endnutzer und ermöglichen eine wesentlich spezifischere Nutzung dieser Wissensbasis.

- Über die gewonnenen strukturierten Informationen lässt sich die Konsistenz von Wikipedia, insbesondere auch die Konsistenz zwischen den verschiedenen Sprachversionen, überprüfen. Hiermit lässt sich die Qualität von Wikipedia und damit ihr Wert als eine der zentralen Wissensressourcen der Menschheit insgesamt verbessern.
- Aus Sicht der Wissensrepräsentation stellen die gewonnenen Daten eine der größten Ontologien dar. Diese Ontologie unterscheidet sich von bisherigen Ontologien darin, dass die in ihr definierten Konzepte ein tatsächliches ‚Community Agreement‘ darstellen und von der Gemeinschaft permanent aktualisiert werden.

Einen Überblick über die Struktur des DBpedia-Projektes gibt Abb. 4. Die DBpedia-Extraktion arbeitet auf der Basis der von Wikipedia veröffentlichten Datenexporte. In den folgenden Abschnitten stellen wir die DBpedia-Extraktion, die resultierenden Datenpakete als auch Beispiele von bestehenden DBpedia-Anwendungen vor. Detaillierte Informationen zu DBpedia finden sich in [2] und [3].

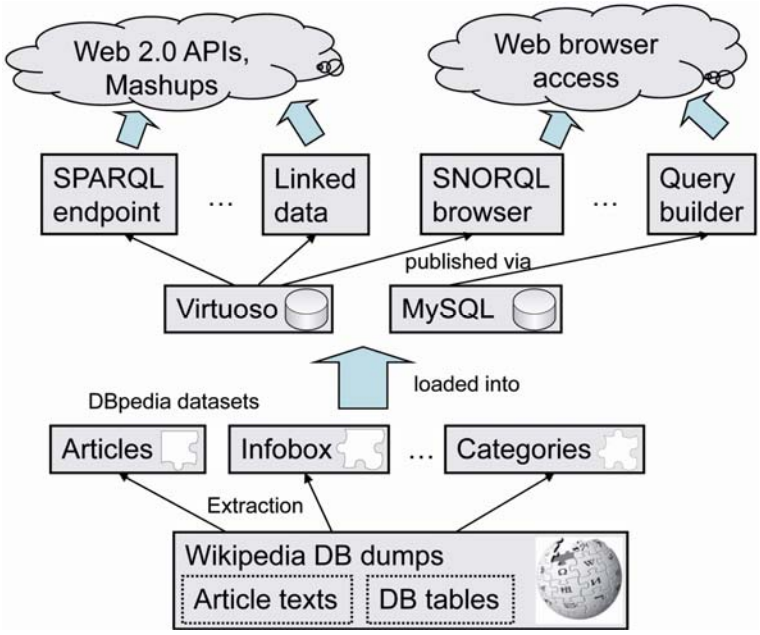

Abb. 4. Überblick über die einzelnen DBpedia-Komponenten

Extraktion

Wikipedia-Artikel bestehen zum größten Teil aus Freitext, enthalten aber auch verschiedene Arten strukturierter Informationen, wie z. B. Infobox-Templates, Kategorisierungen, Bilder, Geo-Koordinaten, Verweise zu externen Webseiten und zu Wikipedia-Editionen in anderen Sprachen.

Die Software hinter der Wikipedia-Webseite ist dabei Mediawiki[3]. In der Natur dieses Wiki-Systems liegt es dabei auch, dass alle Bearbeitungen, Verknüpfungen und Annotationen mit Meta-Daten innerhalb der Artikeltexte mit speziellen syntaktischen Konstrukten realisiert werden. Strukturierte Informationen können daher mittels einer Analyse und Verarbeitung der Artikel und der darin enthaltenen syntaktischen Konstrukte erreicht werden.

Da MediaWiki einige dieser Informationen intern selbst zur Erstellung der Benutzerschnittstelle nutzt, liegen einige extrahierte Informationen bereits in relationalen Datenbanktabellen vor. Exporte der zentralen Datenbank-Tabellen (inklusive der, welche den Artikelvolltext enthält) werden monatlich von Wikipedia im Netz bereitgestellt[4]. Basierend auf diesen Datenbankexporten, nutzt DBpedia im Moment zwei Arten der Extraktion semantischer Beziehungen: (1) Wir bilden relationale Beziehungen, die bereits in Form von Datenbank-Tabellen vorliegen, in RDF ab und (2) wir extrahieren zusätzliche Informationen direkt aus den Artikel-Texten und den Infobox-Vorlagen innerhalb dieser Texte.

Wir illustrieren die Extraktion von Semantik aus dem Artikel-Text anhand einer Wikipedia-Infobox-Vorlage. Abbildung 2 zeigt die Infobox-Vorlage (die im Quelltext eines Wikipedia-Artikels zu finden ist) und die daraus erstellte Ausgabe auf der Wikipedia-Seite zu der süd-koreanischen Stadt Busan. Der Infobox-Extraktions-Algorithmus entdeckt solche Vorlagen und erkennt deren Struktur mit Hilfe von Techniken der Mustererkennung. Er wählt signifikante Vorlagen aus, die dann verarbeitet und in RDF-Tripel konvertiert werden. Der Algorithmus bearbeitet die so extrahierten Daten nach, um die Qualität der Extraktion zu erhöhen. Beispielsweise werden Verweise zu anderen Artikeln erkannt und in passende URIs übersetzt, Maßeinheiten werden als entsprechende Datentypen zu RDF-Literalen hinzugefügt. Darüber hinaus erkennt der Algorithmus Listen von Objekten, die als RDF-Listen repräsentiert werden. Ein Auszug der aus der Wikipedia-Seite über Busan extrahierten RDF-Tripel ist in Abb. 6 dargestellt. Details zur Infobox-Extraktion (inklusive Angaben zur Datentypen-Erkennung, Heuristiken zur Datensäuberung und Generierung von

[3] http://www.mediawiki.org
[4] http://download.wikimedia.org/

```
{{infobox City Korea|
  full_name=Busan Metropolitan City|
  image=[[Image:Haeundaebeachbusan.jpg|
    250px|Haeundae Beach, Busan]]|
  rr=Busan Gwangyeoksi|
  mr=Pusan Kwangyŏksi|
  hangul=부산 광역시|
  hanja=釜山廣域市|
  short_name=Busan (Pusan; 부산; 釜山)|
  population=3,635,389 ...|
  area=763.46 km²|
  government=[[Metropolitan cities of
    South Korea|Metropolitan City]]|
  divisions=15 wards (Gu),
    <br>1 county (Gun)|
  region=[[Yeongnam]]|
  dialect=[[Gyeongsang Dialect|
    Gyeongsang]]|
  map=[[Image:Busan map.png|Map of
    South Korea highlighting the city]]|
 }}
```

Busan Metropolitan City

Korean name	
Revised Romanization	Busan Gwangyeoksi
McCune-Reischauer	Pusan Kwangyŏksi
Hangul	부산 광역시
Hanja	釜山廣域市
Short name	Busan (Pusan; 부산; 釜山)

Abb. 5. Beispiel einer Wikipedia-Infobox-Vorlage und der erstellten Web-Ausgabe (Ausschnitt)

Busan	full_name	„Busan Metropolitan City“
Busan	image	Haeundaebeachbusan.jpg
Busan	rr	„Busan Gwangyeoksi“
Busan	mr	„Pusan KwangyÅksi“
Busan	short	„Busan (Pusan;...)“
Busan	population	„3,657,840...“
Busan	area	„763.46 km2“
Busan	government	Metropolitan_cities_of_South_Korea
Busan	divisions	„15 wards (Gu), 1 county (Gun)“
Busan	region	Yeongnam
Busan	dialect	Gyeongsang_Dialect
Busan	map	Busan_map.png

Abb. 6. Auszug der aus der Wikipedia-Seite über Busan extrahierten RDF-Tripel

URIs) enthält die Publikation [3]. Alle Extraktionsalgorithmen sind in der Skriptsprache PHP implementiert und unter einer Open-Source-Lizenz veröffentlicht[5].

[5] http://sf.net/projects/dbpedia

Die DBpedia-Wissensbasis

Die DBpedia-Wissensbasis umfasst derzeit Informationen über mehr als 1,95 Millionen ‚Dinge', inklusive 80.000 Personen, 70.000 Orte, 35.000 Musik-Alben, 12.000 Filme. Sie enthält 657.000 Verweise zu Bildern, 1.600.000 Verweise zu relevanten externen Webseiten, 180.000 Verweise zu anderen RDF-Daten, 207.000 Wikipedia-Kategorien und 75.000 YAGO-Kategorien (siehe [28]). DBpedia-Konzepte sind darüberhinaus durch Kurz- und Langzusammenfassungen in 13 Sprachen beschrieben. Insgesamt besteht die DBpedia-Wissensbasis aus ca. 103 Millionen RDF-Tripeln.

Jede der 1,95 Millionen Ressourcen in den DBpedia-Datenpaketen ist durch einen eindeutigen URI-Bezeichner der Form `http://dbpedia.org/resource/Name` identifiziert. `Name` ist dabei vom Titel des jeweiligen Wikipedia-Artikels abgeleitet, der sich auch in den Web-Adressen der Wikipedia-Artikel widerspiegelt (z. B. `http://en.wikipedia.org/wiki/Name`). Damit ist jede DBpedia-Ressource direkt mit dem entsprechenden englischsprachigen Wikipedia-Artikel verküpft. Dies resultiert in einer Reihe von vorteilhaften Eigenschaften der DBpedia-URI-Bezeichner:

- Es wird ein breites Spektrum enzyklopädischer Themen abgedeckt.
- Die Bezeichner sind Ergebnis eines Gemeinschaftskonsens.
- Es existieren klare Regeln für deren Management.
- Es existieren eine umfassende textuelle Beschreibung der Konzepte und Verweise zu einer maßgeblichen Webseite (der entsprechenden Wikipedia-Seite).

Die DBpedia-Wissensbasis wird in drei verschiedenen Formen über das Web zugänglich gemacht:

- *Vernetzte Daten:* Die Wissensbasis wird in Form vernetzter Daten veröffentlicht. Dies bedeutet, dass jede DBpedia-URI über das HTTP-Protokoll dereferenzierbar ist.
- *SPARQL-Endpoint:* Der DBpedia-SPARQL-Endpoint ermöglicht es Client-Anwendungen, Anfragen an DBpedia über das SPARQL-Protokoll zu stellen. Zusätzlich zur Standard-konformen SPARQL-Funktionalität, werden einige Funktionen bereitgestellt, die sich als besonders nützlich zur Generierung spezifischer Benutzerschnittstellen erwiesen haben. Dazu gehört eine Volltext-Suche über RDF-Literale, Aggregat-Funktionen zur statistischen Auswertung insbesondere zum Zählen von Anfrageergebnissen. Der SPARQL-Endpunkt wird durch einen Virtuoso Universal Server[6] bereitgestellt.

[6] http://virtuoso.openlinksw.com

- *RDF-Datenpakete:* Zusätzlich wird die DBpedia Wissenbasis auch in Form mehrerer Datenpakete, die jeweils zwischen einer und 60 Millionen RDF-Triples enthalten, zum Download angeboten.

DBpedia-Benutzerschnittstellen

Im Folgenden werden verschiedene Benutzerschnittstellen vorgestellt, über die sich die DBpedia-Wissensbasis erforschen und abfragen lässt.

Graph-Pattern-Builder

Verglichen mit anderen Semantic Web-Wissensbasen, die derzeit verfügbar sind, haben die DBpedia-Datenpakete eine andere Struktur. DBpedia enthält eine Fülle von relativ ungenau definierten Schema-Elementen, insbesondere RDF-Properties. Darüber hinaus beinhalten die DBpedia-Daten eine enorme Menge an Informationen zu diesem relativ vagen Informationsschema. Für einen Anwender ist es daher sehr schwer zu erkennen, welche Objekte und Properties in Anfragen verwendet werden können. Bestehende Werkzeuge fokussieren zudem meist auf große Mengen in nur einer der beiden Informationskategorien: Daten- oder Schemainformationen. Um Anwender trotzdem zu befähigen, diese Fülle an Informationen zu erschließen, müssen neue, alternative Benutzerschnittstellen entwickelt werden.

Eine solche neue Benutzerschnittstelle für große und inhomogen strukturierte Daten ist der Graph-Pattern-Builder. Anwender können mit ihm die Wissensbasis mittels Graph-Pattern bestehend aus mehreren Triple-Patterns anfragen. Ein Web-Formular erlaubt die Eingabe der Triple-Patterns. Für jedes Triple-Pattern existieren drei Formularfelder, in welche Variablen, Objektbezeichner oder Filteroperatoren für Subjekte, Prädikate oder Objekte eines Triples eingetragen werden können. Während Nutzer Objektbezeichner in die entsprechenden Formularfelder eintragen, wird im Hintergrund (per AJAX-Autovervollständigung) in der Wissensbasis nach passenden Objekten gesucht und diese werden dem Nutzer zur Auswahl angeboten. Die passenden Objekte sind dabei nicht beliebige, in denen der eingegebene Suchbegriff auftritt, sondern die komplette Suchanfrage wird mit dem entsprechenden Suchbegriff ausgeführt, und nur solche passenden Resultate werden angeboten, für die letztendlich auch Suchergebnisse für den kompletten Graph-Pattern existieren. Dies ermöglicht den Benutzern Suchanfragen zu stellen, ohne die genaue Stuktur der Wissensbasis zu kennen und trotzdem relevante Ergebnisse zu bekommen. Abbildung 7 zeigt den Graph-Pattern-Builder.

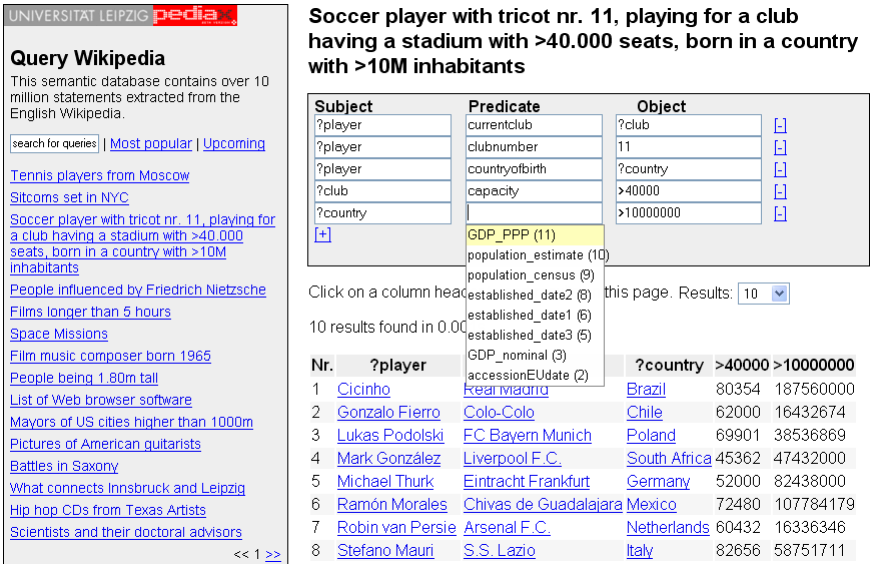

Abb. 7. Formularbasierter Graph Pattern Builder für inhomogen strukturierte Wissensbasen wie z. B. DBpedia

DBpedia Relationship Finder

Der DBpedia Relationship Finder (Abb. 8) ist ein Werkzeug, um Verbindungen zwischen Objekten in Semantic Web Ontologien aufzudecken. Das bedeutet, dass zwischen zwei gegebenen Objekten, die einen Nutzer interessieren, mehrere mögliche Pfade über verschiedene in der betrachteten Wissensbasis vorhandene Objekteigenschaften präsentiert werden. Momentan wird der Relationship Finder speziell für DBpedia eingesetzt, aber kann mit leichten Änderungen auch für andere RDF-basierte Wissensbasen eingesetzt werden. Im Bereich Social Semantic Web könnten dies zum Beispiel die Analyse von Verbindungen zwischen Personen, Ereignissen und Plätzen sein. Für viele große Wissensbasen wären andere Darstellungsformen wie RDF-Graphen zu unübersichtlich.

Auf die Funktionsweise des DBpedia Relationship Finder soll hier nur kurz eingegangen werden: In einem ersten Vorverarbeitungsschritt zerlegt er den vorgegebenen RDF-Graphen in Komponenten, d. h. nicht zusammenhängende Knotenmengen, und speichert einige Zusatzinformationen zu den Objekten in den einzelnen Komponenten. Mit diesen Zusatzinformationen ist der Relationship Finder schnell in der Lage eine Verbindung zwischen zwei Objekten zu ermitteln. Um dann – wie in vielen Fällen gewünscht – auch die kürzesten Verbindungen zwischen Objekten zu errechnen, werden entsprechende Abfragen an den zugrunde liegenden Triple Store generiert.

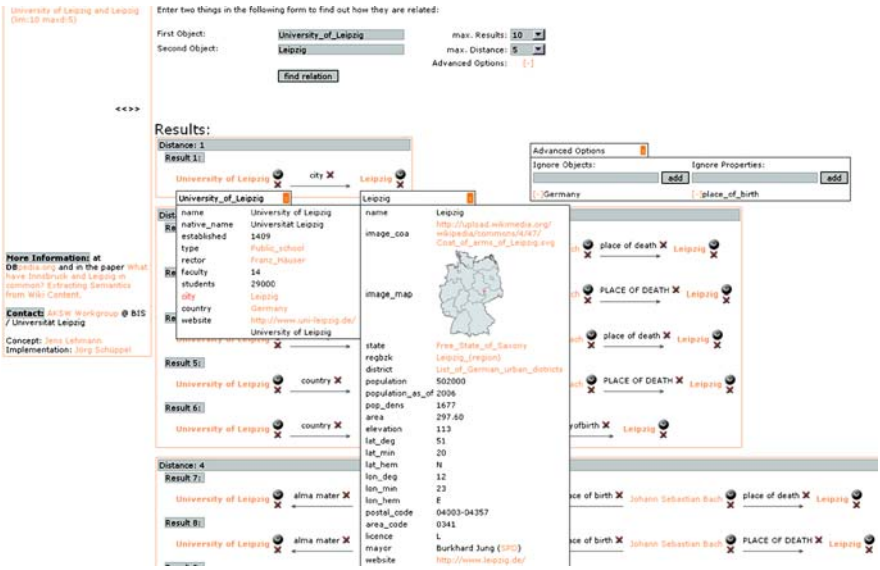

Abb. 8. DBpedia Relationship Finder im Einsatz

Jedes Objekt wird als Link zu der entsprechenden Wikipedia-Seite dargestellt. Durch Klicken eines Icons erhält man in DBpedia enthaltene Zusatzinformationen zu jedem Objekt. Diese Informationen werden, falls notwendig, als Grafiken, Listen, Links usw. dargestellt.

Semantische Mashups

Dieser Abschnitt gibt einen Überblick über verschiedene Mashups, die vernetzte Daten nutzen. Es werden sowohl generische Mashups wie Browser und Suchmaschinen für vernetzte Daten vorgestellt als auch anwendungsspezifische Portale wie z. B. der DOAP Store.

Generische Browser für vernetzte Daten

Generische Browser für vernetzte Daten ermöglichen die integrierte Darstellung von Daten aus verschiedenen Datenquellen und die Navigation zwischen Datenquellen anhand von RDF-Verweisen. Browser für vernetzte Daten unterscheiden sich von allgemeinen RDF-Browsern darin, dass sie nicht davon ausgehen, dass die zu visualisierenden RDF-Daten bereits lokal in einem Repositorium vorliegen, sondern dass sie Daten, je nach Navigationspfad des Nutzers, dynamisch aus dem Web nachladen.

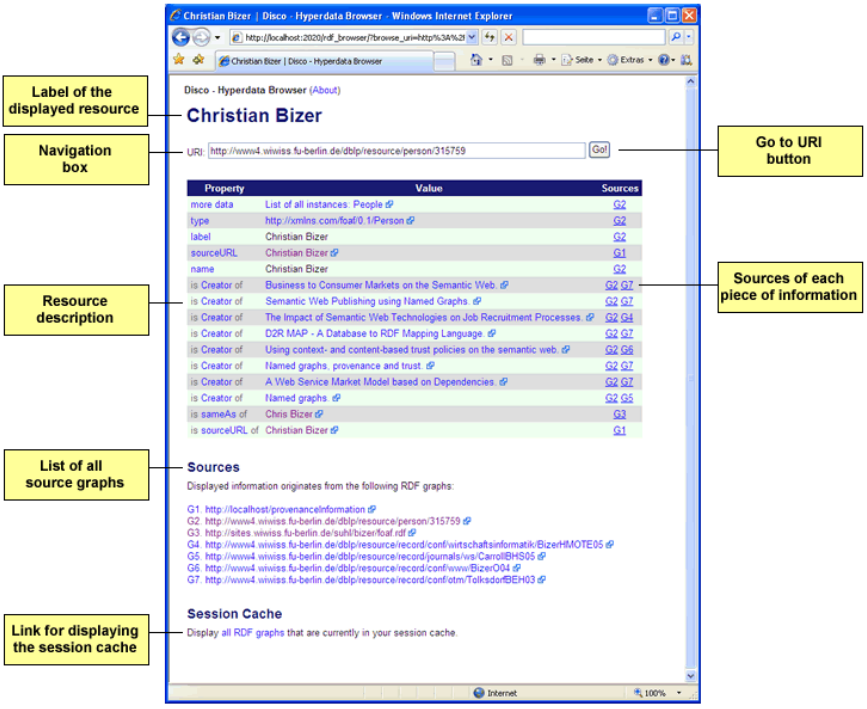

Abb. 9. Benutzerschnittstelle des DISCO Browsers

Beispiele generischer Browser für vernetzte Daten:

- *Tabulator* [29] war der erste verfügbare Browser für vernetzte Daten. Der Browser wurde von der Arbeitsgruppe um Tim Berners-Lee am Massachusetts Institute of Technology entwickelt. Tabulator visualisiert vernetzte Daten in Form eines Baums, in dem jeder Knoten einer Ressource entspricht. Durch Ausklappen einzelner Ressourcen navigiert der Benutzer zwischen Datenquellen. Zusätzlich zur Baumansicht bietet Tabulator auch die Möglichkeit, Abfragen gegen geladene Daten zu stellen und geladene Daten auf einer Landkarte zu visualisieren.
- Der *OpenLink RDF Browser*[7] ermöglicht es, vernetzte Daten mittels unterschiedlicher AJAX-Komponenten in Tabellenform, als Graph, als Zeitreihe sowie als Fotoalbum oder auf einer Landkarte darzustellen. Der Browser unterstützt die RDF-Stylesheet-Sprache Fresnel [9].

[7] http://demo.openlinksw.com/DAV/JS/rdfbrowser/index.html

- Der *Zitgist-Browser*[8] bietet unterschiedliche Vorlagen zur benutzerfreundlichen Visualisierung bekannter Typen von Daten wie Personen, Musikern oder Musikalben.
- Der *DISCO Browser*[9] wurde an der Freien Universität Berlin entwickelt. Ziel war es, einen Browser mit einer minimalistischen Benutzerschnittstelle zu entwerfen, welche die Herkunft von Daten aus unterschiedlichen Quellen klar hervorhebt. Abbildung 9 zeigt die Benutzerschnittstelle des DISCO Browsers. Unterhalb der Navigationsbox werden alle Informationen, die der Browser in unterschiedlichen Datenquellen zu einer Ressource gefunden hat, gemeinsam angezeigt. Am Ende jeder Zeile werden die Datenquellen aufgeführt, aus denen die jeweilige Information stammt.

Suchmaschinen und Verzeichnisdienste für vernetzte Daten

Suchmaschinen für vernetzte Daten verwenden Crawler, die Verknüpfungen zwischen Datensätzen folgen, um Daten aus verschiedenen Web-Datenquellen zu einer lokalen Datenbasis zusammenzufassen. Die Suchmaschinen indizieren diese Datenbasis und ermöglichen es Anfragen gegen die indizierten Inhalte zu stellen.

Beispiele derartiger Suchmaschinen:

- *Swoogle*[10] ist einer der ersten Verteter semantischer Suchmaschinen. Swoogle sucht stichwortbasiert und nutzt damit die Möglichkeiten semantischer Auszeichung nur sehr begrenzt. Technisch basiert Swoogle auf der Text-Suchmaschine Lucene des Apache-Projekts.
- Die *Semantic Web Search Engine (SWSE)*[11] geht einen Schritt weiter, indem zusätzlich zu einem Volltextindex über den Inhalten von Wissensbasen der jeweilige Inhaltstyp indiziert wird. Eine SWSE-Suche kann also auf in den Ergebnisdokumenten gefundene Typen (wie z. B. Personen, Orte etc.) eingeschränkt werden. Diese Typen müssen dabei nicht im Voraus festgelegt werden, sondern werden aus entsprechenden |rdf:type|-Properties (zu Objekten aus den RDF-Dokumenten) gewonnen. Swoogle indiziert derzeit etwa 2,3 Millionen RDF Dokumente.
- Die *Sindice*[12] Suchmaschine indiziert derzeit etwa 11 Millionen RDF-Dokumente. Die Suchmaschine ermöglicht es semantischen Mashups,

[8] http://browser.zitgist.com/
[9] http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/disco/
[10] http://swoogle.umbc.edu
[11] http://swse.deri.org
[12] http://www.sindice.com/

alle bekannten Dokumente, in denen eine spezielle URI verwendet wird, zu finden.

- *Falcons*[13] indiziert derzeit circa 2 Millionen RDF-Dokumente. Neben der eigentlichen Suchfunktionalität bietet Falcons auch einen Daten-Browser, mittels dessen sich die Suchergebnisse analysieren lassen.

Ein Beispiel eines Verzeichnisdienstes für Linked Data bzw. Verlinkte Daten ist PingtheSemanticWeb.com (PTSW). Ping-The-Semantic-Web ist ein Web Service, der Aufschluss darüber gibt, welche RDF-Dokumente kürzlich im Web erstellt oder aktualisiert wurden. Autoren und Editoren von solchen Dokumenten benachrichtigen PTSW darüber, indem sie die URL des erstellten oder geänderten Dokuments übermitteln. PTSW ist also eine Art Basiskomponente einer semantischen Suchmaschine, da sie von Crawlern und anderen Software-Agenten genutzt werden kann um herauszufinden, wo zuletzt aktualisierte RDF-Dokumente gefunden werden können.

Portale auf Basis vernetzter Daten

Portale sind Einstiegspunkte im Web zu bestimmten Themen. Portale greifen dazu auf mehrere Inhaltsquellen zu und aggregieren und präsentieren diese in einer anwendungsdomänenspezifischen Weise.

Ein Portal auf der Basis vernetzter Daten ist zum Beispiel Revyu.[14] Revyu ermöglicht es, Dinge beliebiger Art zu bewerten und mit (persönlichen) Kommentaren zu versehen. Revyu nutzt nicht nur vernetzte Daten zum Annotieren von Bewertungen, sondern stellt vernetzte Daten für alle Dinge in der Revyu-Wissensbasis für Dritte bereit. Generell überwiegt bei Revyu der Anteil anwendergenerierter Daten im Gegensatz zur Nutzung bestehender Quellen vernetzter Daten.

Ein weiteres Beispiel eines semantischen Portals ist DOAP-Store,[15] der Informationen über Forschungs- sowie Open Source Software-Entwicklungsprojekte bereitstellt. Die Funktionsweise von DOAP-Store unterscheidet sich stark von Revyu. DOAP-Store nutzt keinerlei direkt anwendergenerierten Inhalte, sondern sucht RDF-Dokumente im Web, die Informationen enthalten, die mittels des DOAP-Vokabulars[16] ausgedrückt sind. Um entsprechende Dokumente zu finden nutzt DOAP-Store den Service Ping-The-Semantic-Web Verzeichnisdienst.

---

[13] http://iws.seu.edu.cn/services/falcons/
[14] http://revyu.com
[15] http://doapstore.org
[16] Description Of A Project: http://usefulinc.com/doap

Reasoning im Social Semantic Web

Mittels in Form von Ontologien repräsentierten Wissens lässt sich die Funktionalität von Semantischen Mashups weiter verbessern. Dabei wird auf Erkenntnisse innerhalb der Wissensrepräsentation zurückgegriffen, die sich über Jahrzehnte entwickelt haben. Neben der reinen Modellierung von Wissen, das heißt dem Festlegen einer Terminologie und den Beziehungen zwischen diesen Termen, wird dabei auch das Ziehen von Schlussfolgerungen (englisch: Reasoning) aus dem vorhandenen Wissen ermöglicht. Am häufigsten wird dabei sogenanntes deduktives Reasoning betrachtet. Es bedeutet, dass aus explizit gespeicherten Fakten weiteres implizit vorhandenes Wissen ermittelt werden kann. Wir betrachten hier zusätzlich das induktive Reasoning (eine Teildisziplin des maschinellen Lernens), bei dem aus dem vorhandenen Wissen allgemeinere Behauptungen aufgestellt werden. Im Gegensatz zu anderen Bereichen der Wissensrepräsentation sind Wissensmodelle im Social Web oft weniger formalisierbar, sehr groß, unvollständig und oft sogar widersprüchlich. Wir werden beschreiben, wie auf diese spezifischen Anforderungen eingegangen werden kann.

Überblick über Reasoning

Das Modellieren von Wissen in Form von Ontologien ist zentraler Bestandteil des Semantic Web. Ausgehend von frühen Formen der Wissensrepräsentation, wie Frames und Semantischen Netzen, haben sich seit Ende der 1980er Jahre Beschreibungslogiken entwickelt. Aus einer Reihe verschiedener Gründe wurden Beschreibungslogiken als Basis der Ontologiesprache OWL gewählt, die das formale Rückgrat des Semantic Web bildet. Dank dieser logischen Basis haben OWL-Ontologien eine klare Semantik, das heißt neben der rein syntaktischen Repräsentation einer Ontologie kann man ihr auch eine Bedeutung zuweisen. Eine wohldefinierte Semantik bietet die Möglichkeit, Schlüsse aus dem gespeicherten Wissen zu ziehen. Mit dem Ziehen solcher Schlüsse speziell im Kontext eines Social Semantic Web befasst sich dieses Kapitel.

Die häufigste Form des Reasoning ist dabei das deduktive Reasoning, das heißt aus den explizit gespeicherten Fakten wird implizites Wissen geschlossen. Ein einfaches Beispiel ist folgendes:

Nehmen wir an, unsere Ontologie enthält das Wissen, dass Anna eine Mutter ist und jede Mutter auch eine Frau ist:

```
Anna        rdf:type            Mutter
Mutter      rdfs:subClassOf     Frau
```

Daraus lässt sich schlussfolgern, dass Anna eine Frau ist.

Durch die vielfältigen Sprachkonstrukte, die OWL besitzt, kann das Ziehen von Schlüssen ein komplexer Prozess sein. Es haben sich unterschiedliche Algorithmen und Programme entwickelt, von denen ein Ansatz in Abschn. 3 vorgestellt wird.

Neben dem deduktiven Reasoning möchten wir hier auch auf induktives Reasoning eingehen. Induktives Reasoning ist der Lernprozess, bei dem aus vorhandenem Faktenwissen allgemeinere Behauptungen aufgestellt werden. Dies sei wieder an einem kurzen Beispiel illustriert:

Nehmen wir wieder an, es sei eine Ontologie mit folgendem Wissen gegeben:

```
Anna        rdf:type                Frau
Anna        hasChild                Franz
Beate       rdf:type                Frau
```

Beim induktiven Reasoning wird eine bisher nicht existierende Klasse aus dem Faktenwissen gelernt. Man wählt dazu positive und negative Beispiele für eine Klasse aus. Nehmen wir an, ein bisher nicht definiertes Konzept Mutter soll gelernt werden und Anna wird als positives, sowie Beate als negatives Beispiel ausgewählt. Dann kann ein Lernprogramm die Klassendefinition $\text{Frau} \sqcap \exists \text{hasChild}$ („Frau mit Kind“ in üblicher Beschreibungslogiksyntax ausgedruckt) aufstellen.

Im Gegensatz zum deduktiven Reasoning werden beim induktiven Reasoning keine sich aus der Ontologie ergebenden Erkenntnisse gefunden, sondern Behauptungen aufgestellt, die zu dem vorhandenen Wissen passen. Insbesondere kann es mehrere oder keine möglichen Lösungen für ein bestimmtes Lernproblem geben. Wie das induktive Reasoning funktioniert, wird in Kapitel 4 beschrieben. Zuerst soll auf einige Spezifika von Reasoning im Social Semantic Web eingegangen werden.

Social Semantic Web-Anforderungen

Wissensmodelle im Social Web sind oft weniger formalisierbar, sehr groß, unvollständig und oft sogar widersprüchlich. Daher müssen gängige Verfahren des Reasoning für das Social Semantic Web adaptiert werden. Beispiele für häufig auftretende Reasoningprobleme sind:

Symmetrie

Eines der meist verwendeten Vokabulare im Social Semantic Web ist das Friend-of-a-Friend Vokabular. Es erlaubt z. B. Aussagen der Form:

```
Klaus       foaf:knows              Petra
```

Erweiterungen zum FOAF-Vokabular (z. B. [23]) schlagen vor diese Beziehung weiter zu spezialisieren wie z. B. mit Eigenschaften |friendOf|, |spouseOf| oder |siblingOf|. Diese Beziehungen sollten dabei in einer Social Semantic Web-Applikation als symmetrische Beziehung interpretiert werden und folglich soll aus der Aussage:

```
Petra        spouseOf                    Klaus
```

die Aussage

```
Klaus        spouseOf                    Petra
```

schlussgefolgert werden können. Auch DBpedia enthält eine Reihe solcher symmetrischer RDF-Properties.

Klassenhierarchie

Wissensbasen im Social Semantic Web enthalten oft eine Form von Kategorien oder Klassenhierarchie. DBpedia z. B. enthält über 200.000 Kategorien. Kategorien sind z. B. „Städte in Europa“, „Städte in Deutschland“ und „Städte in Sachsen“, die erste und zweite sowie die zweite und dritte sind durch eine Sub-Kategorienbeziehung miteinander verknüpft, nicht jedoch „Städte in Europa“ und „Städte in Sachsen“. Eine Social Semantic Web-Applikation soll nun in der Lage sein, dieses implizite Wissen der transitiven Sub-Klassenbeziehung zu nutzen.

Klassifikation

Das Inferenzproblem der Klassifikation tritt auf, wenn zu einer gegebenen Klasse (oder Kategorie) alle durch diese Klasse umfassten Instanzen ermittelt werden sollen. Nicht immer sind diese Klassen-Instanzen-Beziehungen explizit (mittels der RDF-Property |rdf:type|) gegeben. So sind Artikel über Städte in Sachsen zwar der Kategorie „Städte in Sachsen“ zugeordnet, nicht jedoch den Kategorien „Städte in Deutschland“ oder „Städte in Europa“. Eine Social Semantic Web-Applikation, wie der im vorangegangenen Abschnitt vorgestellte Graph Pattern Builder, sollte solche impliziten Informationen jedoch berücksichtigen.

Wie der Anwendungsfall DBpedia zeigt, ist eine der wichtigsten Anforderungen von Social Semantic Web-Anwendungen an Reasoningalgorithmen Skalierbarkeit. Im folgenden Abschnitt stellen wir eine Strategie vor, wie skalierbares Reasoning auf der Basis relationaler Datenbanken, wie sie für die Implementierung von Semantischen Mashups und Social Semantic Web Anwendungen eingesetzt werden, realisiert werden kann.

Skalierbares Reasoning auf Basis relationaler Datenbanken

In Semantic Web-Anwendungen werden meist entweder relationale Datenbanken oder spezialisierte Triple-Stores zur persistenten Speicherung eingesetzt. Vertreter spezialisierter Triple-Stores sind Sesame [18] oder Redland [6], auf relationaler Datenbankbasis arbeitet z. B. RAP [25]. Reasoner arbeiten dagegen meist im Hauptspeicher und mit Datenstrukturen, die mit relationalen Datenbanken inkompatibel sind. Die Serialisierung, Übertragung und Verarbeitung von Wissensbasen zwischen Datenbank oder Triple-Store und Reasoner ist sehr zeitaufwändig, daher für Semantic Web Anwendungen mit großen Wissensbasen oft nicht praktikabel.

Eine Alternative sind regelbasierte Inferenzsysteme, die direkt auf dem Tripel-Datenmodell arbeiten. In der Arbeit von Royer und Quantz [27] werden Beschreibungslogiken analysiert und ein System (für bestimmte Reasoningaufgaben) vollständiger Inferenzregeln aufgestellt. Diese Inferenzregeln lassen sich direkt in Anfragen auf dem Triple-Datenmodell übersetzen, deren Anfrageergebnisse können als inferierte Aussagen direkt wieder zur Wissensbasis hinzugefügt werden.

Wir illustrieren das mit SPARQL-Anfragen für die drei erwähnten Reasoningaufgaben Symmetrie, Klassenhierarchie und Klassifikation:

```
CONSTRUCT {?i2?p?i1} WHERE {
  ?p <rdf:type> <owl:SymmetricProperty>.
  ?i1?p?i2
  }

CONSTRUCT {?c1 <rdfs:subClassOf>?c3} WHERE {
  ?c1 <rdfs:subClassOf>?c2.
  ?c2 <rdfs:subClassOf>? c3
  }

CONSTRUCT {?i <rdfs:type>?c2} WHERE {
  ?i <rdf:type>?c1.
  ?c1 <rdfs:subClassOf>?c2
  }
```

Diese drei SPARQL-Anfragen liefern exakt die für Inferenzaufgaben notwendigen Triples. Die Ergebnisse der Anfragen können also zur Wissensbasis hinzugefügt werden und die impliziten Informationen sind damit für weitere Anfragen verfügbar. Die dargestellten Beispiele stellen allerdings nur einen kleinen Teil der Inferenzregeln dar, die aus der OWL-Semantik abgeleitet werden können. Für eine detailliertere und umfassendere Darstellung verweisen wir auf [27] und [1]. Einige Inferenzregeln lassen sich bislang auch nicht mittels SPARQL-Anfragen ausdrücken, da

SPARQL z. B. Funktionen zum Zählen und Aggregieren von Ergebnissen fehlen. Mit SQL-Anfragen, die auf einem Datenbankschema zur Speicherung von Triples ausgeführt werden, sind entsprechende Inferenzen jedoch möglich. Da die inferierten Aussagen einer Inferenzregel (oder SPARQL/SQL-Anfrage) die Ergebnisse weiterer Inferenzregeln beeinflussen, ist eine mehrfache Ausführung der Abfragen bis zum Erreichen eines Fixpunktes notwendig.

Die in diesem Abschnitt beispielhaft umrissene Vorgehensweise zur Implementierung skalierbaren Reasonings auf Basis relationaler Datenbanken hat einige entscheidende Vorteile: Die Algorithmen arbeiten direkt mit der nativen Datenhaltung der Wissensbasen in Semantischen Web-Anwendungen. Das Serialisieren, Übertragen, De-Serialisieren etc., das bei Verwendung verschiedener Repräsentationsformen, wie z. B. Datenbanken und tableaubasierten Reasoner auftritt, entfällt. Es kann sehr spezifisch festgelegt werden, welche Inferenzregeln berücksichtigt werden sollen und welche nicht, und es können dadurch wesentliche Geschwindigkeitsverbesserungen für spezifische Reasoningaufgaben erreicht werden.

Induktives Reasoning im Social Semantic Web

Das in der Einleitung des Reasoning-Abschnitts kurz eingeführte induktive Reasoning soll hier kurz dargestellt werden. Das sogenannte Lernproblem (genauer gesagt, eine mögliche Variante des Lernproblems) besteht darin, bei gegebenem Hintergrundwissen und positiven und negativen Beispielen eine Konzeptdefinition zu finden, so dass alle positiven Beispiele aus dieser Definition (und dem bereits vorhandenen Hintergrundwissen) folgen, die negativen Beispiele jedoch nicht. Das Problem lässt sich auch für andere Wissensrepräsentationssprachen außer OWL betrachten und wird hauptsächlich im Bereich der induktiven Logikprogrammierung [24] erforscht. In der Literatur wurden verschiedene Ansätze zur Lösung des Lernproblems vorgestellt [4, 15, 16, 17].

Eines der existierenden Lernsysteme ist DL-Learner,[17] welches als Open Source verfügbar ist.[18] Eine schematische Darstellung der groben Funktionsweise des DL-Learner und anderer Ansätze findet sich in Abb. 10. Die Funktionsweise beruht darauf, dass ein intelligenter Algorithmus mögliche Konzeptdefinitionen vorschlägt. Diese werden durch einen Reasoner getestet und das dadurch erhaltene Feedback fließt wieder in den Kernalgorithmus ein. Konkrete Beschreibungen der verwendeten Algorithmen im DL-Learner finden sich in [20, 21, 22]. Induktives Reasoning beruht

[17] http://dl-learner.org
[18] http://sf.net/projects/dl-learner

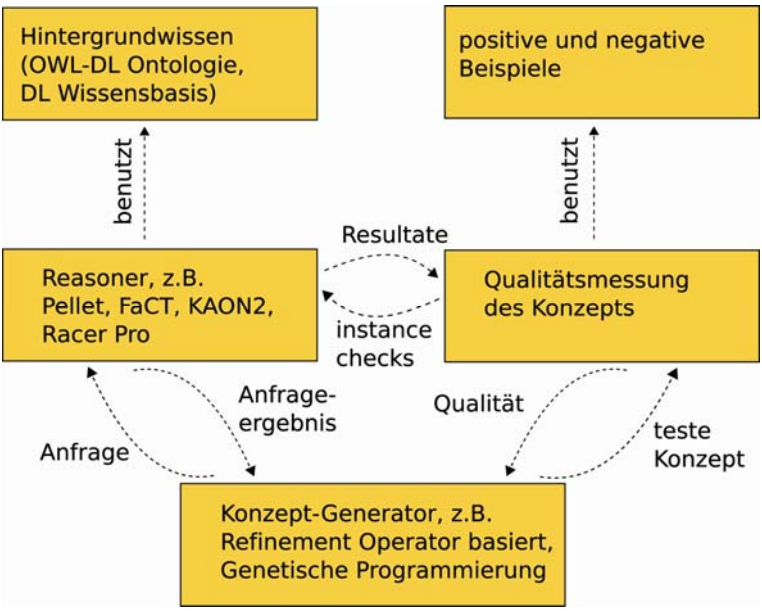

Abb. 10. Lernansatz „generate and test“

in diesem Fall also auf einer potentiell großen Anzahl an deduktiven Reasoninganfragen.

Im Social Semantic Web hat man häufig sehr große verteilte Wissensbasen, so dass sich die Frage stellt, ob solche Lernsysteme geeignet sind um in diesem Kontext angewandt werden zu können. Oft sind diese Wissensbasen, wie im Beispiel DBpedia, als vernetzteDaten vorhanden, über die man mit Hilfe eines SPARQL-Endpunkts Anfragen stellen kann. Aus diesem Grund unterstützt das DL-Learner-Tool direkt SPARQL-Endpunkte als Hintergrundwissen. Um eine Skalierbarkeit der Ansätze zu gewähren, wählt der DL-Learner dabei durch mehrere SPARQL-Anfragen einen Teil des im SPARQL-Endpunkt vorhandenen Wissens aus und schickt dieses an den Reasoner. Somit wird beim Lernen nicht das komplette Wissen berücksichtigt, sondern nur ein für das konkrete Problem möglichst relevanter Teil ausgewählt.

Als Beispiel für ein solches Problem kehren wir wieder zu DBpedia als einer der größten Wissensbasen zurück. Nehmen wir an, jemand stellt eine Anfrage mit den positiven Beispielen „Pythagoras“, „Philolaus“ und „Archytas” und den negativen Beispielen „Socrates“, „Plato“, „Zeno of Elea“ und „Democritus“ mit DBpedia als Hintergrundwissen. Das kann zum Beispiel für eine Internetrecherche relevant sein, bei der jemand Wissen über bestimmte Personen (die positiven Beispiele) und zu ihnen

semantisch verbunden Personen gewinnen möchte, aber andere Personen, die nicht Teile seiner Recherche sind (die negativen Beispiele) ausschließt. Das DL-Learner-SPARQL-Modul setzt dann zuerst Anfragen an den DBpedia-SPARQL-Endpunkt ab um relevante Informationen zu erhalten. Das gewonnene Wissen in OWL-Form wird dem Reasoner mitgeteilt und der Kernalgorithmus gestartet. Dieser ermittelt in diesem Fall $\text{mathematician} \sqcap (\text{physicist} \sqcup \text{vegetarian})$ (Mathematiker, die zusätzlich entweder Vegetarier oder Physiker sind) als eine mögliche Lösung. Alle positiven und keines der negativen Beispiele folgen aus dieser Definition. Mit diesen Techniken könnten in Zukunft auch für das Social Semantic Web typische große Wissensbasen, die über einen SPARQL-Endpoint oder als vernetzte Daten (siehe Abschn. 2) publiziert wurden, analysiert werden oder neue Klassen in diesen Wissensbasen gelernt werden, insbesondere um das Anfragen großer heterogener Datenbestände zu unterstützen.

Zusammenfassung und Ausblick

Dieses Kapitel stellte mit dem Web-Datenintegrationsrahmenwerk vernetzte Daten, dem DBpedia-Projekt zur Extraktion strukturierter Informationen aus Wikipedia und Ansätzen zur Lösung spezifischer Inferenzprobleme drei zentrale Elemente semantischer Mashups vor. Wir haben einen Überblick über erste Beispiele semantischer Mashups gegeben. Die vorgestellten Ansätze und Beispiele sind jedoch nur der Beginn einer Entwicklung, die einerseits weitere technologische Felder erfassen und andererseits die bestehenden Ansätze weiter vertiefen muss, um einen nachhaltigen Einfluss auf das Web zu entfalten.

Herausforderungen, denen wir gegenüberstehen, sind zum Beispiel:

- Semantische Mashups, die Daten aus einer Vielzahl an Quellen nutzen, sind mit verschiedenen Informationsqualitätsproblemen konfrontiert. Eine Analyse der DBpedia-Datenpakete z. B. zeigt, dass Informationen oft noch nicht auf eine Weise repräsentiert sind, die einfache Integration und Querying ermöglichen.
- Wenn es um die Integration personenbezogener Daten im Web geht, sind der Schutz der Privatsphäre sowie die klare Auszeichnung der Daten mit Lizenzinformationen, die bestimmen, wofür Daten verwendet werden dürfen, essentiell. Erste Ansätze in dieser Hinsicht sind die Abbildung von Privacy-Präferenzen z. B. mittels P3P oder das Lizenzieren von Semantic Webinhalten mittels Creative Commons.

- Projekte wie CyC oder SUMO haben relativ umfassende Upper-Level-Ontologien, Klassifikationssysteme und Informations-Taxonomien hervorgebracht. Im Rahmen von DBpedia wurden vor allem Instanz-Daten aus Wikipedia extrahiert. Wir sind zuversichtlich, dass eine stärkere Integration von Upper-Level-Ontologien und DBpedia ein enormes Potential zur Erleichterung von Informationsintegration im Web birgt.
- Die erwähnte Integration von Upper-Level-Ontologien und Instanzdaten kann andererseits dazu beitragen, Inkonsistenzen und Lücken in Informationsquellen und Wissensbasen des sozialen Webs (wie z. B. Wikipedia) aufzudecken und zu schließen.
- Das semantisch reichste DBpedia-Datenpaket resultiert aus der Infobox-Extraktion. Dies wurde bislang nur für die englische Wikipedia-Version erstellt. Die Erstellung von Infobox-Extraktionen für weitere Sprachversionen birgt das Potential, die DBpedia-Wissensbasis wesentlich zu vergrößern, stellt uns aber andererseits vor die Herausforderungen diese verschiedenen Sprachversionen sinnvoll zu integrieren.

Viele dieser Probleme können gelöst werden, indem bestehende Ansätze und Technologien sinnvoll kombiniert und erweitert werden. Wir sind überzeugt, dass eine solche iterative Weiterentwicklung und Konsolidierung des Konzeptes semantischer Mashups letztendlich einen entscheidenden Beitrag leisten wird, das Potential semantischer Repräsentationen für Suchfunktionen und Informationsaustausch im Netz zu realisieren.

Literatur

1. Sören Auer and Zachary Ives. Integrating ontologies and relational data. Technical Report MS-CIS-07-24, Computer and Information Sciences Department, School of Engineering and Applied Science, University of Pennsylvania, 3330 Walnut Street Philadelphia, PA 19104-6389, Oct 2007.
2. Sören Auer, Christian Bizer, Jens Lehmann, Georgi Kobilarov, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In Proceedings of the International Semantic Web Conference (ISWC 2007), 2007.
3. Sören Auer and Jens Lehmann. What have innsbruck and leipzig in common? extracting semantics from wiki content. In Enrico Franconi, Michael Kifer, and Wolfgang May, editors, ESWC, volume 4519 of Lecture Notes in Computer Science, pages 503–517. Springer, 2007.
4. Liviu Badea and Shan-Hwei Nienhuys-Cheng. A refinement operator for description logics. In J. Cussens and A. Frisch, editors, Proceedings of the 10th International Conference on Inductive Logic Programming, volume 1866 of Lecture Notes in Artificial Intelligence, pages 40–59. Springer-Verlag, 2000.
5. D. Beckett. Turtle – Terse RDF Triple Language. http://www.ilrt.bris.ac.uk/discovery/2004/01/turtle/, 2004.

6. David Beckett. The design and implementation of the redland RDF application framework. In Proceedings of the Tenth InternationalWorldWideWeb Conference (WWW2001), February 16 2001.
7. Tim Berners-Lee. Linked data, 2006. http://www.w3.org/DesignIssues/LinkedData.html.
8. Chris Bizer and Richard Cyganiak. D2r server – publishing relational databases on the semantic web, 2007. http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2r-server/.
9. Chris Bizer, Ryan Lee, and Emmanuel Pietriga. Fresnel – display vocabulary for RDF, 2004.
10. Christian Bizer, Richard Cyganiak, and Tobias Gauß. The RDF Book Mashup: From Web APIs to a Web of Data. In Proceedings of the 3rd Workshop on Scripting for the Semantic Web, 2007.
11. Christian Bizer, Richard Cyganiak, and Tom Heath. How to publish linked data on the web, 2007. http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/.
12. John Breslin. Sioc exporters, 2007. http://sioc-project.org/exporters.
13. Richard Cyganiak and Chris Bizer. Pubby – a linked data frontend for sparql endpoints, 2007. http://www4.wiwiss.fu-berlin.de/pubby/.
14. Orri Erling and Ivan Mikhailov. RDF support in the Virtuoso DBMS. volume P-113 of GI-Edition – Lecture Notes in Informatics (LNI), ISSN 1617-5468. Bonner Köllen Verlag, September 2007.
15. Nicola Fanizzi, Luigi Iannone, Ignazio Palmisano, and Giovanni Semeraro. Concept formation in expressive description logics. In Machine Learning: ECML 2004, 15th European Conference on Machine Learning, Pisa, Italy, September 20–24, 2004, Proceedings. Springer, 2004.
16. Luigi Iannone and Ignazio Palmisano. An algorithm based on counterfactuals for concept learning in the semantic web. In Proceedings of the 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pages 370–379, Bari, Italy, June 2005.
17. Luigi Iannone, Ignazio Palmisano, and Nicola Fanizzi. An algorithm based on counterfactuals for concept learning in the semantic web. Applied Intelligence, 26(2):139–159, 2007.
18. Arjohn Kampman, Frank Van Harmelen, and Jeen Broekstra. Sesame: An architecture for storing and querying RDF data and schema information, July 03 2001.
19. Graham Klyne and Jeremy J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax – W3C Recommendation, 2004. http://www.w3.org/TR/rdf-concepts/.
20. Jens Lehmann. Hybrid learning of ontology classes. In Proceedings of the 5th International Conference on Machine Learning and Data Mining, MLDM 2007. Springer, 2007.
21. Jens Lehmann and Pascal Hitzler. Foundations of refinement operators for description logics. In Proceedings of the 17th International Conference on Inductive Logic Programming (ILP). Springer, 2007.

22. Jens Lehmann and Pascal Hitzler. A refinement operator based learning algorithm for the alc description logic. In Proceedings of the 17th International Conference on Inductive Logic Programming (ILP). Springer, 2007.
23. Yutaka Matsuo, Masahiro Hamasaki, Junichiro Mori, Hideaki Takeda, and Koiti Hasida. Ontological consideration on human relationship vocabulary for foaf. In SWAD-Europe Final Workshop „Friend of a Friend, Social Networking and the Semantic Web“, held 1–2 September 2004 in Galway, Ireland, 2004.
24. Shan-Hwei Nienhuys-Cheng and Ronald de Wolf, editors. Foundations of Inductive Logic Programming. Lecture Notes in Computer Science. Springer, 1997.
25. Radoslaw Oldakowski, Christian Bizer, and Daniel Westphal. RAP: RDF API for PHP. In Sören Auer, Chris Bizer, and Libby Miller, editors, Proceedings of the Workshop Scripting for the Semantic Web, number 135 in CEUR-Workshop Proceedings, Heraklion, Greece, 05 2005.
26. o.V. Linking open data – w3c sweo community project, 2007. http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.
27. Veronique Royer and J. Joachim Quantz. Deriving inference rules for description logics: a rewriting approach into sequent calculi. Technical Report TUB-FB13-KIT-111, KIT Project Group Publications, December 1 1993. Tue, 07 Nov 1995 19:31:17 GMT.
28. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge. In 16th international World Wide Web conference (WWW 2007), New York, NY, USA, 2007. ACM Press.
29. Tim Berners-Lee et al. Tabulator: Exploring and analyzing linked data on the semanticweb. In Proceedings of the 3rd International Semantic Web User Interaction Workshop, 2006.
http://swui.semanticweb.org/swui06/papers/Berners-Lee/Berners-Lee.pdf.