Semantic Web

Gerd Gröner, Ansgar Scherp, and Steffen Staab

{groener,scherp,staab}@uni-koblenz.de
Institute for Web Science and Technologies, WeST
University of Koblenz-Landau

1 Einleitung

Im World Wide Web werden unstrukturierte Informationen und informelles Wissen in Form von Hypertext dargestellt, um die Verbreitung von Informationen und Wissen zu erleichtern. Ziel des *Semantic Webs* ist es, strukturierte Informationen und formales Wissen verteilt im Web bereitzustellen, um daraus mittels automatisierten Schlussfolgerungen Antworten ableiten zu können. Auf diese Weise lassen sich Wissensbestandteile aus verschiedenen Quellen intelligent miteinander integrieren und komplexe Fragen beantworten. Anfragen, die sich auf diese Weise beantworten lassen, sind zum Beispiel "Welche Arten von Musik werden in britischen Radiostationen gespielt?" oder "Welche Radiostation spielt Lieder von schwedischen Künstlern?" Heutige Suchmaschinen können solche Anfragen nur unzureichend beantworten, obwohl alle benötigten Informationen bereits im Web verfügbar sind. Insbesondere liegt eine große Menge an Daten bereits in maschinenverarbeitbaren Formaten im Semantic Web vor. Allerdings können die Inhalte nur sehr eingeschränkt von heutigen Suchmaschinen analysiert werden. Im Wesentlichen indizieren Suchmaschinen die Hypertexte und Dokumente im Web, die in natürlicher Sprache vorliegen, einzeln und sind nicht in der Lage, dort vorhandene Inhalte intelligent miteinander zu kombinieren. Die Stärke des Semantic Webs liegt nun darin, Daten und deren Bedeutung aus verschiedenen Quellen miteinander zu kombinieren.

Schauen wir uns dazu an, wie man mit Hilfe des Semantic Webs die oben genannten Fragen beantworten kann: Die BBC veröffentlicht die Abspiellisten ihrer Radiostationen online in Formaten des Semantic Webs. Die Musikgruppe "ABBA" hat einen eindeutigen Bezeichner in Form einer URI (`http://www.bbc.co.uk/music/artists/d87e52c5-bb8d-4da8-b941-9f4928627dc8#artist`). Diese URI kann verwendet werden, um die Musikgruppe mit Informationen aus dem Musikportal MusicBrainz[1] zu verknüpfen. MusicBrainz kennt die Mitglieder der Band, wie beispielsweise Benny Andersson, sowie das Genre und die Lieder. Zudem ist MusicBrainz mit Wikipedia verknüpft, um beispielsweise Informationen zu Künstlern, wie Biographien auf DBpedia [3], nutzen zu können. Informationen über britische Radiostationen können in Form von Listen auf Web-Seiten wie beispielsweise ListenLive[2] gefunden werden, welche ebenfalls in

[1] `http://musicbrainz.org/`
[2] `http://www.listenlive.eu/uk.html`

eine Semantic Web Repräsentation, eine sogenannte Ontologie, überführt werden könnten—nämlich als eine Beschreibung von Objekten und ihren Beziehungen. MusicBrainz ist mittels der Playcount-Ontologie[3] mit der BBC verbunden. Derzeit werden von der BBC mindestens neun verschiedene Ontologien mit unterschiedlichem Grad an Formalität und verschiedenen Beziehungen zueinander genutzt um ihre Daten zu beschreiben (vgl. auch [47]). Die Bedeutung von *Beziehungen* zwischen Daten wird ebenfalls mittels Ontologien des Semantic Webs beschrieben, z.B. bietet Dublin Core[4] ein Metadaten-Schema zur Beschreibung allgemeiner Eigenschaften von Informationsobjekten. Dadurch dass diese Daten im Semantic Web verfügbar sind, können weitere Fragen gestellt werden, z.B. wie oft Musik eines bestimmten Genre von britischen Radiostationen gespielt wird oder welche Radiostationen Lieder von schwedischen Künstlern spielen. Um Fragen in diesem und in anderen Szenarien zu beantworten, werden generische Software-Komponenten, Sprachen und Protokolle benötigt, die nahtlos miteinander interagieren können.

Das Szenario zeigt, dass die Daten im Semantic Web aus verschiedenen Quellen stammen können und miteinander vernetzt sind. Abgesehen von technischen Gesichtspunkten ist das Semantic Web auch als ein gesellschaftspolitisches Phänomen zu verstehen. Ähnlich dem World Wide Web veröffentlichen verschiedene Personen und Organisationen im Semantic Web Daten und arbeiten zusammen, um diese miteinander zu verknüpfen und zu verbessern. Neben dem oben genannten Beispiel ist das Semantic Web auch für eine Vielzahl anderer Anwendungsgebiete einsetzbar, wie beispielsweise für persönliches Informationsmanagement [22], für die interaktive Exploration sozialer Medien [52] oder für mobile Informationssysteme [11].

In diesem Kapitel wird der prinzipielle Aufbau des Semantic Webs vorgestellt. Analog zu einschlägiger Grundlagenliteratur im Bereich Semantic Web [1,36,38] befasst sich dieses Kapitel mit den verschiedenen grundlegenden Semantic Web Technologien und deren Anwendung. Sehr viele dieser Technologien stammen aus der Künstlichen Intelligenz oder haben zumindest einen starken Bezug dazu. Ontologien dienen zur formalen Repräsentation von Wissen. Die Web Ontology Language (OWL) [29] basiert auf Beschreibungslogik und entsprechend werden Schlussfolgerungsdienste von wissensbasierten Systemen für OWL-Ontologien angewendet.

Im folgenden Abschnitt stellen wir zunächst die allgemeine Architektur des Semantic Webs vor. Anschließend zeigen wir in Abschnitt 3, wie verteilte Daten mit Hilfe von Technologien des Semantic Webs verwaltet, das heißt verknüpft und angefragt werden können. Das im Beispiel verwendete Netzwerk von Ontologien wird in Abschnitt 4 näher analysiert und allgemeine Strategien zur verteilten Wissensrepräsentation und -integration im Semantic Web werden dargestellt. In Abschnitt 5 zeigen wir, wie Schlussfolgerungen aus semantischen Daten gezogen werden können. Um die Daten in einer sich über das Web erstreckenden Wissensbasis verwalten zu können, sind eine Reihe von Herausforderungen zu lösen.

[3] `http://dbtune.org/bbc/playcount/`
[4] `http://dublincore.org/documents/dc-rdf/`

Diese sind zum einen die Identifizierung von Ressourcen und deren Verknüpfung (Abschnitt 6), Herkunft und Vertrauenswürdigkeit der Daten (Abschnitt 7) sowie generische Benutzungsschnittstellen und Semantic Web Anwendungen (Abschnitt 8). Der Artikel schließt mit einer Übersicht der aktuellen, praktischen Nutzung von semantischen Technologien und einer Zusammenfassung.

2 Semantic Web Architektur

Das Szenario in Abschnitt 1 beschreibt *was* das Semantic Web als Infrastruktur realisiert, aber nicht *wie* dies erreicht wird. In der Tat wurden die beschriebenen Fähigkeiten *im Kleinen* bereits von einigen wissensbasierten Systemen, die aus der Tradition der Künstlichen Intelligenz stammen, umgesetzt. Für eine Umsetzung *im Großen* mangelte es diesen Systemen aber an Flexibilität, Robustheit und Skalierbarkeit. Teilweise lag dies an der Komplexität der verwendeten Algorithmen. So waren beispielsweise Wissensbasen in Beschreibungslogik, die als Grundlage von Webontologien dienen, in den 1990-er Jahren bezüglich ihrer Größe so eingeschränkt, dass sie höchstens einige hundert Konzepte handhaben konnten (vgl. [37]). Zwischenzeitlich wurden enorme Verbesserungen erreicht. Stark angestiegene Rechenleistung und optimierte Algorithmen ermöglichen eine praktische Handhabung von großen Ontologien wie SNOMED[5] mit hunderttausenden von Axiomen. Allerdings gibt es einige grundlegende Unterschiede zwischen klassischen wissensbasierten Systemen und dem Semantic Web.

Die Verwaltung von Daten in traditionellen wissensbasierten Systemen weist Schwachstellen im Bezug auf die Verarbeitung großer Mengen an Daten und Datenquellen auf, unter anderem wegen (1) unterschiedlicher zugrundeliegender Formalismen, (2) verteilten Standorten, (3) verschiedenen Befugnissen, (4) unterschiedlicher Datenqualität und (5) einer hohen Änderungshäufigkeit der verwendeten Daten. Um mit diesen Problemen umgehen zu können, wendet das Semantic Web einige grundlegende Prinzipien an:

1. *Explizite und einfache Datendarstellung:* Eine allgemeine Datenrepräsentation abstrahiert von den zugrundeliegenden Formaten und erfasst nur das Wesentliche.
2. *Verteilte Systeme:* Ein verteiltes System arbeitet auf einer großen Menge an Datenquellen, ohne dass eine zentrale Steuerung regelt, welche Informationen wohin und wem gehören.
3. *Querverweise:* Die Vorteile eines Netzwerks von Daten bei der Beantwortung von Anfragen begründen sich nicht alleine aus der reinen Mengen von Daten sondern aus ihrer Verknüpfung, die es erlaubt, Daten und Datendefinitionen aus anderen Quellen wieder zu verwenden.
4. *Lose Koppelung mit gemeinsamen Sprachkonstrukten:* Das World Wide Web und ebenso das Semantic Web sind Mega-Systeme, also Systeme, die aus vielen Teilsystemen bestehen, die ihrerseits groß und komplex sind. In einem solchen Mega-System müssen einzelne Bestandteile lose gekoppelt sein, um

[5] `http://www.snomed.org/`

größtmögliche Flexibilität zu erreichen. Die Kommunikation zwischen den Bestandteilen erfolgt auf Grundlage von standardisierten Sprachen, wobei diese individuell an spezifische Systeme angepasst werden können.

5. *Einfaches Veröffentlichen und einfacher Konsum:* Gerade in einem Mega-System muss die Teilnahme, also das Veröffentlichen und der Konsum von Daten, möglichst einfach sein.

Diese Prinzipien werden durch einen Mix an Protokollen, Sprachdefinitionen und Softwarekomponenten erzielt. Einige dieser Bestandteile sind bereits durch das W3C standardisiert, das sowohl Syntax als auch formale Semantik der Sprachen und Protokolle festgelegt hat. Weitere Bestandteile sind noch nicht standardisiert, aber sie sind bereits im sogenannten *Semantic Web Layer Cake* von Tim Berners-Lee vorgesehen (vgl. `http://www.w3.org/2007/03/layerCake.png`). Wir stellen eine Variante des Semantic Web Layer Cakes vor, wobei wir zwischen standardisierten Sprachen und derzeitigen Entwicklungen unterscheiden. Eine graphische Darstellung des Semantic Web Layer Cakes ist in Abbildung 1 dargestellt. Nachfolgend werden die entsprechenden Bausteine kurz vorgestellt.

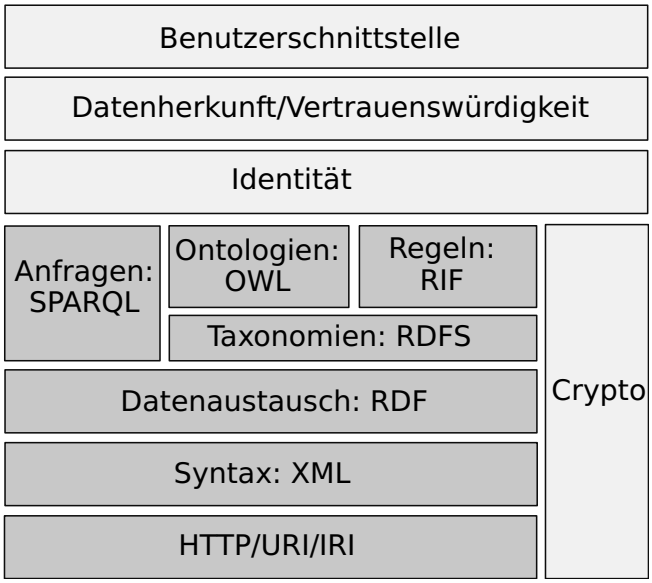

Abbildung 1. Darstellung der Bestandteile als Semantic Web Layer Cake. W3C-Sprachstandards sind in dunkelgrau dargestellt. Derzeitige Entwicklungen sind in hellgrau abgebildet.

HTTP/URI/IRI. Da das Semantic Web dezentral organisiert ist, benötigt man Mechanismen, um nicht nur auf selbst eingeführte Entitäten zu verweisen, sondern auch auf Entitäten die von Dritten veröffentlicht wurden. Entitäten (auch

Ressourcen genannt) werden im Internet durch sogenannte Uniform Resource Identifiers (URIs) [5] identifiziert. Wie im Web halten sich URIs auch im Semantic Web an das Domain Name System (DNS), das die globale Eindeutigkeit von Domainnamen und URIs garantiert, welche mit "http" beginnen. In unserem Beispiel, beschreibt die URI `http://www.bbc.co.uk/music/artists/2f031686-3f01-4f33-a4fc-fb3944532efa#artist` Benny Andersson von ABBA. Ein Anwender kann eine URI, die beispielsweise auf ABBA verweist, dereferenzieren, indem ein sogenannter Look-up mittels HTTP ausgeführt wird, um eine ausführliche Beschreibung der URI zu erhalten. Internationalized Resource Identifiers (IRIs) [15] ergänzen URIs um internationale Zeichensätze aus Unicode/ISO10646. HTTP-Anfragen und URI/IRI-Referenzen werden in Abschnitt 3 detaillierter behandelt.

XML. Nachdem nun Ressourcen eindeutig referenzierbar und dereferenzierbar sind, wird eine Syntax benötigt, um Beschreibungen von Ressourcen im Web auszutauschen. Die Extensible Markup Language (XML) wird zur Strukturierung von Dokumenten verwendet und ermöglicht die Spezifikation und Serialisierung strukturierter Daten.

RDF. Neben der Referenzierbarkeit von Ressourcen und einer einheitlichen Syntax für den Austausch von Daten, benötigt man ein Datenmodell, das es erlaubt, Ressourcen sowohl im Einzelnen als auch in ihrer Gesamtheit und ihrer Verknüpfung zu beschreiben. Eine integrierte Darstellung von Daten aus mehreren Quellen wird durch ein auf gerichteten Graphen basierendes Datenmodell erreicht [46]. Die entsprechende W3C-Standardsprache ist RDF (Resource Description Framework). RDF-Graphen können auf verschiedene Arten serialisiert werden. Am häufigsten ist die XML-basierte Serialisierung. Ein RDF-Graph besteht aus einer Menge von RDF-Tripeln, wobei ein Tripel aus Subjekt, Prädikat (Eigenschaft) und Objekt besteht. Auf RDF wird, in Verbindung mit der Verteilung von Daten, in Abschnitt 3 weiter eingegangen.

SPARQL. Nachdem RDF die Integration von Daten verschiedener Quellen ermöglicht, ist der nächste Schritt die Anfrage von RDF-Graphen. SPARQL[6] (ein rekursives Akronym für SPARQL Protocol and RDF Query Language) ist eine deklarative Anfragesprache für RDF-Graphen. SPARQL 1.1[7] ist die aktuelle Version von SPARQL. SPARQL-Anfragen werden ausführlicher in Abschnitt 3 vorgestellt.

RDFS. RDF-Graphen können die Bedeutung von Daten nur teilweise beschreiben. Sehr oft werden Konstrukte zur Modellierung von hierarchischen Beziehungen zwischen Klassen und Eigenschaften benötigt. Derartige Beziehungen werden typischerweise in Taxonomien und Ontologien beschrieben. RDFS (RDF Schema) ist eine Ontologie-Beschreibungssprache, die unter anderem Hierarchien zwischen Klassen und Eigenschaften beschreiben kann [12].

[6] `http://www.w3.org/TR/rdf-sparql-query/`
[7] `http://www.w3.org/TR/sparql11-query/`

OWL. Daten von verschiedenen Quellen sind sehr heterogen. RDFS ist nicht ausdrucksstark genug, um Daten aus verschiedenen Quellen zusammenzuführen und darüber Konsistenzkriterien zu definieren, wie beispielsweise die Disjunktheit von Klassen. OWL (Web Ontology Language) ist eine Ontologie-Beschreibungssprache, die im Vergleich zu RDFS ausdrucksstärkere Sprachkonstrukte bereit hält. Beispielsweise ermöglicht OWL die Spezifikation von Äquivalenzen zwischen Klassen und Kardinalitätseinschränkungen von Eigenschaften [29].

Die Bedeutung von OWL-Konstrukten kann durch zwei alternative Semantiken definiert werden. Die RDF-basierte Semantik und die direkte modelltheoretische Semantik, auf der Semantik von Beschreibungslogik ($\mathcal{SROIQ}$) aufbaut und Schlussfolgerungen in Form von deduktiver Inferenz ermöglicht. OWL 2 als die aktuelle Version von OWL beinhaltet mehrere Untersprachen, sogenannte Profile, die je nach Anwendungskontext zwischen Ausdrucksmächtigkeit und effizienter Schlussfolgerung abwägen. OWL wird ausführlich in Abschnitt 4 behandelt und Inferenz in OWL wird in Abschnitt 5 beispielhaft dargestellt.

RIF. Regeln sind weitere Formalismen zur Repräsentation von Wissen im Semantic Web. RIF (Rule Interchange Format)[8] beinhaltet eine Menge von Regelsprachen, die in logische Regelsprachen und ereignisbasierte Regelsprachen unterteilt werden können. RIF Core Dialect [8] beschreibt eine Teilmenge von verbreiteten Regelsprachen aus beiden Mengen.

Crypto. Weitere Aspekte im Semantic Web sind Verschlüsselung und Authentifizierung, um sicher zu stellen, dass Datenübertragungen nicht abgehört, gelesen oder modifiziert werden können. Crypto-Module, wie beispielsweise SSL (Secure Socket Layer), verifizieren digitale Zertifikate und ermöglichen Datenschutz und Authentifizierung.

Identifizierung und Verknüpfung. Inhalte, die aus vielen Datenquellen aggregiert sind, können viele verschiedene Identitäten beinhalten, die jedoch das selbe reale Objekt repräsentieren. Integration und Verknüpfungsmechanismen ermöglichen Bezüge zwischen Daten aus verschieden Quellen herzustellen. Eine weitere Betrachtung dieser Themen erfolgt in Abschnitt 6.

Herkunft und Vertrauenswürdigkeit. Daten im Semantic Web können mit zusätzlicher Information über ihre Vertrauenswürdigkeit und Herkunft erweitert werden. Abschnitt 7 beschreibt diese Aspekt ausführlich.

Benutzeroberfläche. Eine Benutzeroberfläche ermöglicht Anwendern die Interaktion mit Daten im Semantic Web. Aus funktioneller Sicht sind einige Benutzeroberflächen generisch und arbeiten auf der Graphstruktur der Daten, wobei andere auf bestimmte Aufgaben, Anwendungen oder Ontologien zugeschnitten sind. Neue Paradigmen untersuchen aktuell das Spektrum an möglichen Benutzeroberflächen zwischen Allgemeingültigkeit und speziellen Anforderungen von

[8] `http://www.w3.org/2005/rules/wiki/RIF_Working_Group`

Endanwendern. Benutzeroberflächen werden im Zusammenhang mit verschiedenen Beispielanwendungen in Abschnitt 8 behandelt.

3 Verteilte semantische Daten im Web

Das zentrale Ziel des Semantic Webs ist das Teilen von Wissen und Informationen und das Zusammenwirken und die Kooperation von menschlichen und maschinellen Akteuren. Jeder kann eine Ontologie erstellen und sie mit anderen Datenquellen so verknüpfen, dass daraus ein Mehrwert für diesen Akteur entsteht und dabei aber als "Abfallprodukt" andere Akteure diese neuen Verknüpfungen ebenfalls verwenden können. Auf diese Weise entsteht eine Daten- und Wissensbasis, die es erlaubt aus der enormen Menge an Informationen und ihrer Verknüpfungen neue Zusammenhänge abzufragen und zu entdecken. Um diese Art der kollektiver Wissensgenerierung zu unterstützen sind effiziente Zugriffe auf verteilte Daten, klar definierte Veröffentlichungsprinzipien und geeignete Anfragemöglichkeiten notwendig.

3.1 Verknüpfte Daten

Die *Linked Data* Prinzipien[9] beschreiben relevante Methoden zur Darstellung, Veröffentlichung und Verwendung von Daten. Sie können wie folgt zusammengefasst werden:

1. URIs werden als Namen für Entitäten verwendet.
2. Das Protokoll HTTP GET wird verwendet, um Beschreibungen zu einer URI abzurufen.
3. Datenprovider sollen auf HTTP GET Abfragen von URIs relevante Informationen mit Hilfe von Standardsprachen (z.B. in RDF) zurückgeben.
4. Verknüpfungen (Links) zu anderen URIs sollen verwendet werden, um die Entdeckung und Verwendung von weiteren Informationen zu erleichtern.

Die Veröffentlichung von Daten anhand der Linked Data Prinzipien ermöglicht einfachen Remote-Zugriff auf Daten via HTTP. Dies erlaubt die Erkundung von Ressourcen und die Navigation durch Ressourcen im Web. URIs (1) werden mittels HTTP-Anfragen dereferenziert (2), um zusätzliche Informationen über eine bestimmte Ressource zu erhalten, insbesondere können diese Informationen mittels standardisierter Syntax (3) ebenfalls Verknüpfungen zu anderen Ressourcen enthalten (4). Abbildung 2 stellt ein Beispiel für Linked Data zu ABBA dar. Das Beispiel stammt von MusicBrainz. Es beschreibt verschiedene Prädikate, die Entitäten mit der URI von ABBA verknüpfen, wie beispielsweise `foaf:member` und `rdf:type`. In der Abbildung ist ABBA, oder genauer die URI von ABBA, das Subjekt, Property bezieht sich auf Prädikate und die Werte (Value) stellen Objekte der RDF-Tripel dar. Das Prädikat `owl:sameAs` wird im weiteren Verlauf noch ausführlicher betrachtet. Die Präfixe `foaf`, `rdf` und `owl`

[9] `http://www.w3.org/DesignIssues/LinkedData.html`

beziehen sich auf Vokabulare der FOAF-Ontologie[10] , bzw. der Sprachspezifikationen von RDF und OWL.

ABBA

Resource URI: **http://dbtune.org/musicbrainz/resource/artist/d87e52c5-bb8d-4da8-b941-9f4928627dc8**

Property	Value
vocab:alias	Abba
vocab:alias	Björn + Benny + Anna + Frieda
bio:event	<http://dbtune.org/musicbrainz/resource/artist/d87e52c5-bb8d-4da8-b941-9f4928627dc8/birth>
bio:event	<http://dbtune.org/musicbrainz/resource/artist/d87e52c5-bb8d-4da8-b941-9f4928627dc8/death>
foaf:homepage	<http://www.abbasite.com>
foaf:member	<http://dbtune.org/musicbrainz/resource/artist/042c35d3-0756-4804-b2c2-be57a683efa2>
foaf:member	<http://dbtune.org/musicbrainz/resource/artist/2f031686-3f01-4f33-a4fc-fb3944532efa>
foaf:member	<http://dbtune.org/musicbrainz/resource/artist/aebbb417-0d18-4fec-a2e2-ce9663d1fa7e>
foaf:member	<http://dbtune.org/musicbrainz/resource/artist/ffb77292-9712-4d03-94aa-bdb1d4771d38>
mo:musicbrainz	<http://musicbrainz.org/artist/d87e52c5-bb8d-4da8-b941-9f4928627dc8>
foaf:name	ABBA
owl:sameAs	<http://dbpedia.org/resource/ABBA>
owl:sameAs	<http://sv.wikipedia.org/wiki/Abba>
owl:sameAs	<http://www.bbc.co.uk/music/artists/d87e52c5-bb8d-4da8-b941-9f4928627dc8#artist>
rdf:type	mo:MusicArtist

Abbildung 2. Linked Data Beispiel für ABBA.

3.2 Anfragen mit SPARQL

Die Verknüpfung von Daten nach den Linked Data Prinzipien ermöglicht Anfragen über sehr große (verknüpfte) Datenmengen. Allerdings sind in vielen Anwendungen einfache Anfragen, wie beispielsweise die Ausgabe aller Prädikate der Ressource ABBA (vgl. Abbildung 2), nicht ausreichend, sondern komplexe Anfragen mit mehreren Anfragebedingungen werden benötigt. SPARQL ist eine Anfragesprache für RDF. Ähnlich zu SQL beschreibt die `WHERE`-Klausel unter welchen Bedingungen Daten selektiert werden.

Neben der eigentlichen Anfragesprache definiert SPARQL auch Zugriffsprotokolle und Datenformate. Ein Repositorium, das SPARQL unterstützt, muss diese Protokolle und Datenformate einhalten. Einige der meistgenutzten Repositorien sind Sesame [13], Jena [65], Virtuoso[11] und OWLIM[12]. Repositorien übernehmen neben der Speicherung von Daten auch die Bereitstellung von Anfragemöglichkeiten in Form von *SPARQL-Endpunkten*. Datensätze, die einen

[10] http://xmlns.com/foaf/spec/
[11] `http://virtuoso.openlinksw.com`
[12] `http://www.ontotext.com/owlim`

SPARQL-Endpunkt haben sind in der Regel mittels REST-Protokoll [19] zu erreichen.

Neben der Anfrage von explizit vorhandenen Fakten sieht SPARQL auch die Unterstützung durch *Entailment-Regimes* vor. Diese erweitern die Abfrage von explizit vorhandenen Fakten um Fakten, die anhand von RDFS- und OWL-Konstrukten geschlussfolgert wurden (vgl. Abschnitt 5). Je nach Funktionsumfang des jeweiligen Repositoriums werden verschiedene (oder auch gar keine) Entailment-Regimes unterstützt. Die entsprechenden Schlussfolgerungsdienste werden durch die Verwendung von geeigneten Reasoners bereitgestellt.

SPARQL 1.1 liegen verschiedene Semantiken zugrunde und entsprechend werden alternative Entailment-Regimes realisiert. Ein Entailment-Regime ist das RDF- und das RDFS-Entailment für SPARQL-Anfragen. Ein RDF-Graph G_1 folgt aus einem anderen RDF-Graph G_2 wenn jede RDF-Interpretation die G_2 erfüllt auch G_1 erfüllt. Analog gilt dies für RDFS-Interpretationen, d.h. zusätzlich werden RDFS-Konstrukte wie *rdfs:subClassOf* interpretiert. Das D-Entailment basiert auf dem RDF-Entailment und berücksichtigt noch die Interpretation von Datentypen. Ein weiteres Entailment-Regime folgt der RDFS-basierten OWL 2 Semantik. Es erweitert das D-Entailment, d.h. die Interpretation von RDF-Graphen entspricht der Interpretation von dem RDF-Entailment. Zusätzlich wird das D-Entailment aber noch um die Interpretation von OWL 2-Konstrukten erweitert. Eine andere Semantik ist durch die OWL 2 Direct Semantic gegeben, die unter anderem auch die OWL 2 Profile OWL 2 DL, EL und QL abdeckt. Dieser Semantik liegt eine Abbildung von OWL 2 Konstrukten nach RDF-Tripel zugrunde.

Nachfolgend wird nochmals das Beispiel von MusicBrainz betrachtet, in dem Informationen über ABBA gesucht werden. Wir sind nun an den Interpreten von ABBA interessiert, die auch Mitglieder anderer Bands sind. Folgen wir dem Linked Data Prinzip, so würde das bedeuten, dass zunächst nach der URI angefragt wird, die für ABBA steht, dann würde zu den einzelnen Bandmitgliedern navigiert werden, und anschließend den Links zu allen Bands der Mitglieder gefolgt werden.

SPARQL geht über das prozedurale Verfolgen von Links hinaus und basiert auf dem Abgleich von Graphmustern (Graph Pattern Matching). Graphmuster aus der SPARQL-Anfrage werden mit vorhandenen RDF-Tripeln in den angefragten RDF-Graphen verglichen und eventuell nach weiteren Kriterien gefiltert. Das Graphmuster für Bands, deren Mitglieder ebenfalls ABBA-Mitglied sind, ist in Abbildung 3 dargestellt, und die entsprechende SPARQL-Anfrage ist in Abbildung 4 beschrieben.

Die eigentlichen Vergleichsbedingungen der Graphmuster mit RDF-Tripeln werden in der `WHERE`-Klausel beschrieben. Das Graphmuster besteht aus einzelnen Tripelmustern. An jeder Stelle eines Tripelmusters (Subjekt, Prädikat, Objekt) kann entweder eine URI oder eine Variable stehen, in der Objektposition auch ein Literal (eine Art String). Wenn die Nichtvariablen eines Tripelmusters mit vorhandenen Tripeln übereinstimmen, dann können die Variablen an die entsprechenden Werte dieser Tripel gebunden werden. Wenn mehrere Tri-

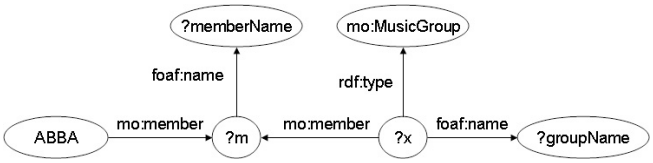

Abbildung 3. Graphische Darstellung einer Anfrage für Musikgruppen (dargestellt durch die Variable *?groupName*), deren Mitglieder ebenfalls Mitglieder bei ABBA sind. Die Variable *?m* bezieht sich auf die Mitglieder von ABBA. Der Knoten mit Beschriftung "ABBA" stellt die URI für ABBA dar. Das Präfix *mo* bezieht sich auf die Musikontologie, *foaf* auf die FOAF-Ontologie und *rdf* auf das Vokabular der RDF-Spezifikation.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX bbc: <http://www.bbc.co.uk/music/>
SELECT ?memberName ?groupName
WHERE  { bbc:artists/d87e52c5-bb8d-4da8-b941-9f4928627dc8#artist mo:member ?m .
         ?x mo:member ?m .
         ?x rdf:type mo:MusicGroup .
         ?m foaf:name ?memberName .
         ?x foaf:name ?groupName }
FILTER (?groupName <>  "ABBA")
```

Abbildung 4. SPARQL Anfrage für Musikgruppen, deren Mitglieder auch Mitglieder bei ABBA sind. Im ersten Tripelmuster des WHERE-Teils ist die URI von ABBA das Subjekt.

pelmuster ein Graphmuster bilden, dann muss es Kombinationsmöglichkeiten dieser Variablenbindungen geben, die in der Belegung der gemeinsamen Variablen übereinstimmen, damit das Gesamtmuster passen kann. Der `WHERE`-Klausel kann ein `FILTER`-Ausdruck folgen, der die Ergebnisse nach dem angegebenen Kriterium (hier muss der Name der Gruppe ungleich zum Namen "ABBA" sein) reduziert. Das Schlüsselwort `PREFIX` ermöglicht die Einführung von Bezeichnern für URIs (vgl. Abbildung 4).

3.3 Anfragen auf verknüpfte und verteilte Daten

Anfragen auf einzelne Datenquellen sind die Grundbausteine verteilter Anfragen, allerdings bedarf es noch Erweiterungen um mehrere Datenquellen anzufragen. Ein möglicher Ansatz sind Anfragen von mehreren benannten RDF-Graphen (engl. named graphs). Ein benannter Graph in RDF ist eine Menge von RDF-Tripel. Eine Datenquelle kann aus mehreren benannten Graphen bestehen. Einige konkrete Implementierungen realisieren Anfragen über mehrere RDF-Graphen. Eine Architektur, sowie Indexstrukturen und Algorithmen zur Ausführung von verteilten Anfragen wurden in [60] vorgestellt. Dieser Ansatz wurde in Form von sogenannten Networked Graphs verfeinert und erweitert [53]. Networked Graphs ermöglichen neben der Anfrage auf verteilten Quellen und

der Generierung von Sichten auf verteilte Graphen auch die Verbindung dieser Graphen in Form von rekursiven Sichten, die als `CONSTRUCT`-Anfragen definiert werden.

Eine Beispielanfrage auf DBpedia und MusicBrainz ist in Abbildung 5 dargestellt. Es werden alle Künstler von ABBA (dargestellt durch das Subjekt `bbc:artists/d87e52c5-bb8d-4da8-b941-9f4928627dc8#artist)` mit Namen und Biographie gesucht. Die Ergebnisse sind verknüpfte Informationen über Künstler die aus zwei Datenquellen stammen: DBpedia und MusicBrainz. Die Variable für Mitglieder (`?member`) ist das Verknüpfungselement beider Graphen, wie in Abbildung 6 dargestellt. Es ist durchaus möglich, dass URIs der selben Interpreten in DBpedia und MusicBrainz verschieden sind. In diesem Fall kann mittels der Beziehung `owl:sameAs` die Äquivalenz beider URIs definiert werden. Aus Sicht des Anwenders wird eine Anfrage, wie in Abbildung 5 dargestellt, über SPARQL-Endpunkte ausgeführt. Über SPARQL-Endpunkte werden Daten zweier benannter RDF-Graphen extrahiert. Networked Graphs verbergen die Komplexität von Anfragen auf verteilten und entfernten Repositorien Sie kombinieren die Ergebnisse der verteilten Anfragen zu einem Gesamtergebnis.

```
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX bbc: <http://www.bbc.co.uk/music/>
CONSTRUCT { ?member mo:wikipedia ?biography . ?member foaf:name ?name}
FROM NAMED :MusicBrainz   FROM NAMED :DBpedia
WHERE {
    GRAPH :MusicBrainz {
        bbc:artists/d87e52c5-bb8d-4da8-b941-9f4928627dc8#artist mo:member ?member .
        ?member foaf:name ?name }
    GRAPH :DBpedia {
        ?member mo:wikipedia ?biography }
}
```

Abbildung 5. SPARQL-Anfrage für ABBA-Mitglieder.

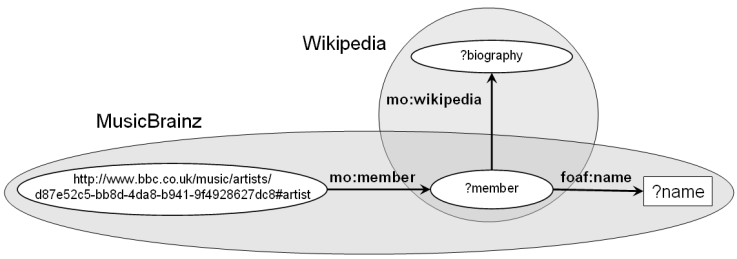

Abbildung 6. Verknüpfte Information über ABBA aus zwei Datenquellen.

Anfragen auf verteilte RDF-Daten sind sehr ähnlich zu Anfragen auf schemalosen und verteilten Datenbanken oder auf Peer-to-peer-Datenbanken (vgl. [31]). Es gibt noch weitere Anfrageparadigmen, welche direkt die Vorteile von Linked Data und den Linked Data Prinzipien nutzen. Das Anfrageprinzip von Hartig et al. [34] kombiniert die grundlegende Eigenschaft der Dereferenzierbarkeit von URIs in Linked Data mit Daten-Crawling. Es umgeht das Problem mangelnder Indizes, indem während der Anfrageausführung den RDF-Links gefolgt wird, die für die Anfrage relevant sein könnten, um weitere Informationen zu entdecken. Der Suchraum wird erweitert anhand der Tripelmuster in der Anfrage zu den nächst verbundenen Datenquellen, die zu einem Anfrageergebnis beitragen könnten. Umgesetzt wird dies mittels eines Indexmechanismus, der Datenbeschreibungen benutzt, die auf Informationen über Instanzen und Schemata beruhen [31]. Solche Indexstrukturen ermöglichen die Auswahl von relevanten Datenquellen für Anfragen, die sich auf mehrere, anfangs eventuell noch unbekannte Datenquellen beziehen. Dieser Ansatz ermöglicht auch Anfragen an Linked Data Quellen, die keinen SPARQL-Endpunkt anbieten, allerdings auf Kosten der Vollständigkeit der Anfrageergebnisse. Detaillierte Aussagen zur Vollständigkeit dieses Anfrageparadigmas werden in [33,35] getroffen.

4 Wissensrepräsentation und -integration

Eine Ontologie wird verstanden als formale, maschinenverarbeitbare Repräsentation von relevanten Begriffen und Relationen einer Domäne [45,43]. Ontologien stellen damit eine gemeinsame Sicht dar [45], das heißt die formale Begriffsbildung von Ontologien drückt eine übereinstimmende Meinung verschiedener Ersteller aus.

4.1 Analyse des einführenden Beispiels

Um das Beispielszenario aus Abschnitt 1 zur Verknüpfung von MusicBrainz und dem BBC-Programm zu modellieren, werden mehrere Ontologien verwendet und zu einem Netzwerk im Web verbunden. Der Ansatz eine Ontologie nicht monolithisch zu definieren, sondern als einen Baustein in einem miteinander verknüpften Netz von Ontologien zu betrachten, ist eine wesentliche Neuorientierung im Semantic Web im Vergleich zu den klassischen KI-Ansätzen. Ein Ausschnitt für das im Szenario verwendete Netzwerk von Ontologien ist in Abbildung 7 dargestellt. Es wird eine an UML angelehnte grafische Notation zur Darstellung der Ontologien verwendet, da diese weit verbreitet und leicht verständlich ist. Die Rechtecke wie beispielsweise *ProgrammeItem* oder *MusicArtist* stellen die relevanten Konzepte der Musikdomäne dar. Diese sind aus verschiedenen Ontologien entnommen, die über die gestrichelten Kästen angedeutet sind. Beziehungen zwischen Konzepten sind über einen beschrifteten Pfeil dargestellt. Zur Illustration von Vererbungsbeziehungen zwischen Konzepten wird wie in UML ein Dreieckpfeils verwendet.

In MusicBrainz werden Künstler durch das Konzept `MusicArtist` aus der Music Ontology[13] repräsentiert. Diese werden über die Relation `object` mit dem Konzept `Playcount` der Playcount Ontology verknüpft. Das Konzept `Playcount` wird verwendet, um die Anzahl der gespielten Musikstücke eines Künstlers darzustellen. Die Playcount Ontology ist über das Konzept `Brand` (auf Deutsch: Handelsmarke) mit der Programmontologie der BBC[14] verbunden. Des Weiteren ist das Konzept `Brand` über seine Oberklasse `Programme` mit dem Konzept `Service` verbunden, welches beschreibt wo eine Handelsmarke bzw. Künstler gespielt wird. Ein `Service` kann beispielsweise ein auf eine bestimmte Region zugeschnittene Version des BBC-Programms sein. Ein `Programme` wird von einem `Broadcaster` übertragen, welches vom Konzept `Organization` der FOAF-Ontologie spezialisiert wurde. Zudem wird das Ereignis einer Übertragung, das heißt eines `Broadcast` in der BBC-Ontologie als Spezialisierung des Konzeptes `Event` der Event Ontology[15] modelliert. Ein `Broadcast` steht über die Relation `broadcast_of` in Bezug auf ein bestimmte Version eines übertragenen Elements wie beispielsweise eine bestimmte Version eines gekürzten Radioprogramms. Das Konzept `Version` hat die Relationen `time` und assoziert damit ein `Interval` mit dem übertragenen Element für temporale Annotationen wie beispielsweise Untertitel und abgespielte Tracks. Das `Interval`-Konzept stammt von der Timeline Ontology[16].

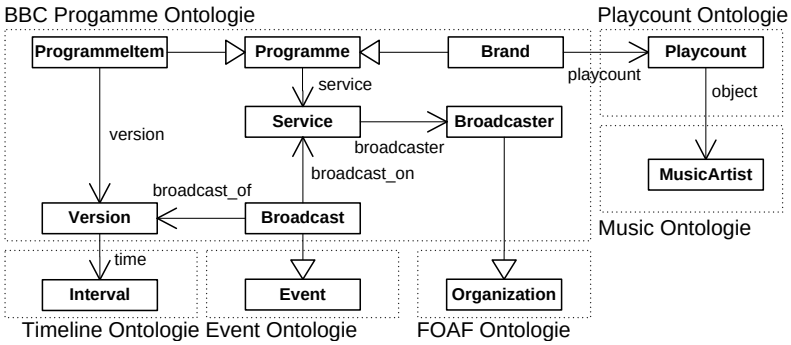

Abbildung 7. Ausschnitt der BBC-Ontology mit Verknüpfungen zu anderen Ontologien (Notation angelehnt an UML).

Die Ontologien in dem Netzwerk, wie es in Abbildung 7 dargestellt ist, sind hinsichtlich ihrer Größe und formalen Beschreibung sehr homogen. Dies muss jedoch nicht so sein. So können die Ontologien auch sehr verschieden groß sein oder können sehr formal definiert sein oder auch nicht. Insgesamt lassen sich

[13] `http://musicontology.com`
[14] `http://www.bbc.co.uk/ontologies`
[15] `http://motools.sourceforge.net/event`
[16] `http://motools.sourceforge.net/timeline`

Ontologien auf Grund unterschiedlicher nicht-funktionaler Eigenschaften in drei verschiedene Arten unterscheiden [56]. Diese verschiedenen Arten von Ontologien werden im folgenden Abschnitt eingeführt, bevor wir das oben genannte Beispiel noch einmal und zusammen mit diesem Hintergrundwissen betrachten.

4.2 Verschiedene Arten von Ontologien

Ein Netzwerk von Ontologien, wie das in Abbildung 7 dargestellte Beispiel, kann aus einer Vielzahl von Ontologien bestehen, die von unterschiedlichen Akteuren und Communities erstellt wurden. Ontologien können das Ergebnis einer Transformation oder einer Reengineering-Tätigkeit eines Altsystems sein, wie beispielsweise einer relationalen Datenbank oder existierender Taxonomie wie zum Beispiel der Dewey Decimal Classification[17] oder Dublin Core. Andere Ontologien werden von Grund auf neu erstellt. Dabei werden existierende Methoden und Werkzeuge zum Ontologie-Engineering angewendet und eine geeignete Repräsentationssprache für die Ontologie wird gewählt (siehe Abschnitt 5). Ontologien können sehr einfach sein, wie die genannte FOAF-Ontologie oder Event-Ontologie, oder sehr komplex und umfangreich, da sie von Domänenexperten entwickelt wurden, wie die medizinische Ontologie SNOMED. Ontologie-Engineering beschäftigt sich also mit Methoden zur Erstellung von Ontologien [28] und hat seinen Ursprung im Software-Engineering in der Erstellung von Domänenmodellen und im Datenbankentwurf in der Erstellung von konzeptuellen Modellen. Eine gute Übersicht zum Thema Ontologie-Engineering ist in verschiedenen Referenzbüchern zu finden [58,28]. Ontologien unterscheiden sich stark in ihrer Struktur, Größe, angewendeten Entwicklungsmethode und betrachteten Anwendungsbereich. Komplexe Ontologien werden zudem hinsichtlich ihres Zwecks und ihrer Granularität unterschieden:

Domänenontologien wie SNOMED stellen die Repräsentationen von Wissen dar, das spezifisch ist für eine bestimmte Domäne [17,43]. Domänenontologien werden als externe Quellen von Hintergrundwissen verwendet [17]. Sie können auf Basisontologien [44] oder Kernontologien [54] aufbauen, die der Domänenontologie Strukturierungen vorgeben und damit die Interoperabilität zwischen verschiedenen Domänenontologien verbessern.

Kernontologien stellen eine präzise Definition strukturierten Wissens in einem bestimmten Bereich dar, der sich über mehrere Anwendungsdomänen hin erstreckt [56,43]. Beispiele für Kernontologien sind die Kernontologie für Software-Komponenten und Web-Services [43], für Ereignisse und Ereignisbeziehungen [55], für persönliches Informationsmanagement [22] oder für Multimedia Metadaten [49]. Kernontologien sollten dabei auf Basisontologien aufsetzen, um von deren Formalisierung und starker Axiomatisierung zu profitieren [56]. Dazu werden in Kernontologien neue Konzepte und Relationen für die betrachtete Anwendungsdomäne hinzugefügt und von den Basisontologien spezialisiert.

[17] `http://dewey.info/`

Basisontologien haben einen sehr breiten Anwendungsbereich und können in den verschiedensten Modellierungsszenarien wiederverwendet werden [10]. Sie dienen daher zu Referenzzwecken [43] und haben zum Ziel die allgemeinsten und generischen Konzepte und Relationen zu modellieren mit denen fast beliebige Aspekte unserer Welt beschrieben werden können [10,43], wie beispielsweise Objekte und Ereignisse. Ein Beispiel ist die Basisontologie Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [10]. Basisontologien haben eine reichhaltige Axiomatisierung, die zum Entwicklungszeitpunkt von Ontologien wichtig ist. Sie helfen dem Ontologie-Entwickler eine formale und in sich konsistente Konzeptualisierung des betrachteten Ausschnitts der Welt, die zu modellieren und auf Konsistenz zu überprüfen ist. Für die spätere Anwendung von Basisontologien in einer konkreten Anwendung, das heißt während der Laufzeit einer Anwendung, kann die reichhaltige Axiomatisierung oft entfernt und durch eine leichtgewichtigere Version der Basisontologie ersetzt werden.

Im Gegensatz dazu werden Domänenontologien spezifisch dafür gebaut, um zur Laufzeit automatische Schlussfolgerungen ziehen zu können. Daher ist beim Entwurf und der Entwicklung von Ontologien immer die Vollständigkeit und Komplexität auf der einen Seite mit der Effizienz auf der anderen Seite abzuwägen. Um strukturiertes Wissen wie das in Abbildung 7 dargestellte Szenario abzubilden werden vernetzte Ontologien benötigt, die in einem Netzwerk über das Internet aufgespannt werden. Dazu müssen die verwendeten Ontologien zueinander passen und abgeglichen werden.

4.3 Verteiltes Netzwerk von Ontologien im Web

Ein Netzwerk von Ontologien muss hinsichtlich der ihr auferlegten funktionalen Anforderungen flexibel sein. Dies liegt daran, dass Systeme über die Zeit hin verändert, erweitert, kombiniert oder integriert werden. Zudem müssen die vernetzten Ontologien zu einem gemeinsamen Verständnis der modellierten Domäne führen. Dieses gemeinsame Verständnis kann durch ein ausreichendes Maß an Formalisierung und Axiomatisierung sowie durch Verwendung von Ontologiemustern erzielt werden. Ontologiemuster erlauben, Teile aus der Originalontologie auszuwählen und entweder alle oder nur bestimmte Teile der Ontologie im Netzwerk wiederzuverwenden. Um ein Netzwerk von Ontologien zu schaffen, können also beispielsweise bereits existierende Ontologien im Web zusammengeführt werden. Auf der anderen Seite kann der Ontologieentwickler auch die Modularisierung von Ontologien mit Hilfe von Ontologiemustern vorantreiben beziehungsweise explizit vorsehen. Einen Ansatz um ein Netzwerk von Ontologien zu entwerfen stellen die Kernontologien dar. Sie erlauben strukturiertes Wissen in komplexen Domänen zu erfassen und auszutauschen. Wohldefinierte Kernontologien erfüllen die im vorangegangenen Abschnitt genannten Eigenschaften und ermöglichen eine einfache Integration und ein reibungsloses Zusammenspiel der Ontologien (siehe auch [56]). Der Ansatz der vernetzten Ontologien führt zu einer flachen Struktur, wie in Abbildung 7 dargestellt, bei der alle verwendeten Ontologien auf derselben Ebene verweilen. Solche Strukturen lassen sich bis zu einem gewissen Grad an Komplexität beherrschen.

Der Ansatz der vernetzten Kernontologien wird am Beispiel von Ontologieschichten beginnend bei Basis- über Kern- zu Domänenontologien veranschaulicht. Wie in Abbildung 8 dargestellt, ist DOLCE als Basisontologie auf der unteren Schicht, die Multimedia Metadata Ontology (M3O) [50] als Kernontologie für Multimedia-Metadaten und eine Erweiterung der M3O für die Musikdomäne.

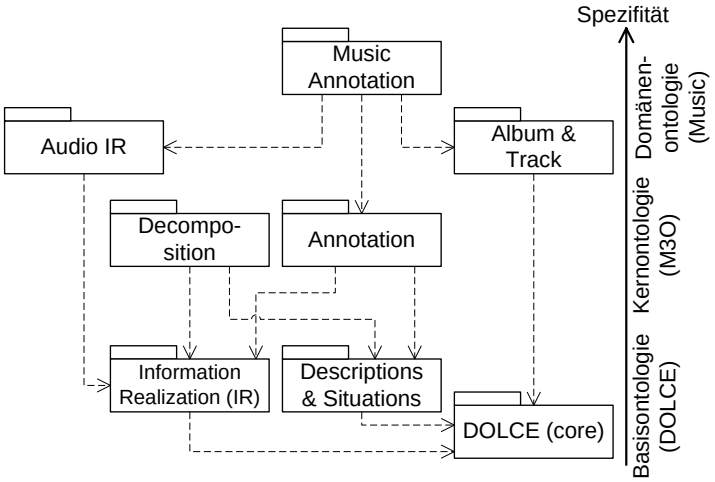

Abbildung 8. Ontologieschichten mit der Kombination von DOLCE, M3O, domänenspezifische Erweiterungen der M3O zur Annotation von Audio-Daten und Musik und eine Domänenontologie für Alben und Tracks.

Kernontologien sind typischerweise groß und decken mit ihrer Wissensmodellierung einen Bereich ab, der vielleicht größer ist als es die spezifische Anwendungsdomäne erfordert [23]. Konkrete Informationssysteme werden typischerweise nur einen Teil der Kernontologien nutzen. Um eine Modularisierung von Kernontologien zu erreichen, sollten Sie mit Hilfe von Ontologiemustern entworfen sein. Ein Ontologiemuster stellt ähnlich wie ein Entwurfsmuster in der Software-Technik eine generische Lösung für ein wiederkehrendes Modellierungsproblem dar. Durch eine präzise Abstimmung der Konzepte in der Kernontologie mit den angebotenen Konzepten der Basisontologie stellen sie eine solide Basis für zukünftige Erweiterungen dar. Neue Muster können hinzugefügt werden und existierende Muster können durch Spezialisierung der Konzepte und Rollen erweitert werden. Abbildung 8 zeigt verschiedene Muster der M3O und DOLCE Ontologien. Im Idealfall werden die Ontologiemuster der Kernontologien in den Domänenontologien wiederverwendet [23] wie in Abbildung 8 dargestellt. Da jedoch nicht davon ausgegangen werden kann, dass alle Domänenontologien mit einer Basis- oder Kernontologie abgestimmt sind, muss auch die Option berücksichtigt werden, dass Domänenontologien unabhängig davon entwickelt und gepflegt werden. In diesem Fall kann Domänenwissen in Kernonto-

logien durch die Anwendung des Ontologiemusters Descriptions and Situations (DnS) der Basisontologie DOLCE wiederverwendet werden. Das Ontologiemuster DnS ist eine ontologische Formalisierung von Kontext [43] durch die Definition verschiedener Sichten mittels Rollen. Diese Rollen können sich auf Domänenontologien beziehen und erlauben eine klare Trennung des strukturierten Wissens der Kernontologie und domänenspezifischen Wissens. Wir erwarten, dass in Zukunft weitere Ontologie-Entwurfsmuster und Kernontologien entstehen. Beispielsweise könnte das Musikszenario in Abschnitt 1 von einer Kernontologie für das Medienmanagement profitieren.

Zur Modellierung eines Netzwerkes von Ontologien, wie das oben beschriebene Beispiel, wird oftmals die Web Ontology Language (OWL) und deren Möglichkeit zur Axiomatisierung mittels Beschreibungslogik [4] verwendet. Neben der Verwendung zur Modellierung einer verteilten Wissensrepräsentation und -integration wird OWL wie in Abschnitt 5 beschrieben insbesondere auch verwendet, um Schlussfolgerungen mittels Inferenz aus diesem Wissen abzuleiten.

5 Inferenz im Web

In Abschnitt 2 wurden verschiedene formale Sprachen zur Wissensrepräsentation im Semantic Web vorgestellt. RDF ermöglicht die Beschreibung einfacher Fakten (Aussagen mit Subjekt, Prädikat und Objekt, sogenannte RDF-Tripel), z. B. “Anni-Frid Lyngstad” “ist Mitglied von” “ABBA”. Entitäten werden mit benannten Beziehungen verknüpft. Die Menge von solchen Verknüpfungen bildet einen gerichteten Graphen. RDFS ermöglicht die Definition von Typen von Entitäten (Klassen), Beziehungen zwischen Klassen und eine Sub- und Superklassenhierarchie zwischen Typen. OWL ist noch ausdrucksstärker als RDF und RDFS. OWL erlaubt beispielsweise die Definition von disjunkten Klassen (Begriffen) oder die Beschreibung von Klassen in Form von Schnitt, Vereinigung und Komplement anderer Klassen (vgl. Abschnitt 4).

Basierend auf diesen formalen Sprachen und deren Semantik können durch deduktive Inferenz weitere (implizite) Fakten aus der Wissensbasis abgeleitet werden. Im Folgenden wird beispielhaft die Herleitung von impliziten Fakten aus einer Menge von explizit gegebenen Fakten mittels des RDFS-Konstrukts `rdfs:subClassOf` und des OWL-Konstrukts `owl:sameAs` dargestellt. `rdfs:subClassOf` beschreibt hierarchische Beziehungen zwischen Klassen und mit `owl:sameAs` können zwei Ressourcen als identisch definiert werden.

Als erstes Beispiel betrachten wir die Klasse `foaf:Person`, welche in der FOAF-Ontologie definiert ist, und die Klassen `mo:Musician` und `mo:Group`, die in der Musikontologie definiert sind. In der Musikontologie gibt es zusätzlich ein Axiom, welches `mo:Musician` als Subklasse von `foaf:Person` mittels `rdfs:subClassOf` definiert. Aufgrund dieses Axioms kann durch deduktive Inferenz hergeleitet werden, dass Instanzen von `mo:Musician` auch Instanzen von `foaf:Person` sind. Falls es nun eine solche Hierarchie von Klassen gibt und zusätzlich noch eine Aussage das Anni-Frid Lyngstad vom Typ `mo:Musician` ist,

so kann mittels Inferenz hergeleitet werden, dass Anni-Frid Lyngstad auch vom Typ `foaf:Person` ist. Dies bedeutet, dass alle Anfragen die nach Entitäten vom Typ `foaf:Person` fragen auch Anni-Frid Lyngstad im Anfrageergebnis enthalten, auch wenn diese Instanz nicht explizit als Instanz von `foaf:Person` definiert ist. Abbildung 9 stellt diese Fakten und die entsprechende Klassenhierarchie in RDFS als gerichteten Graph dar.

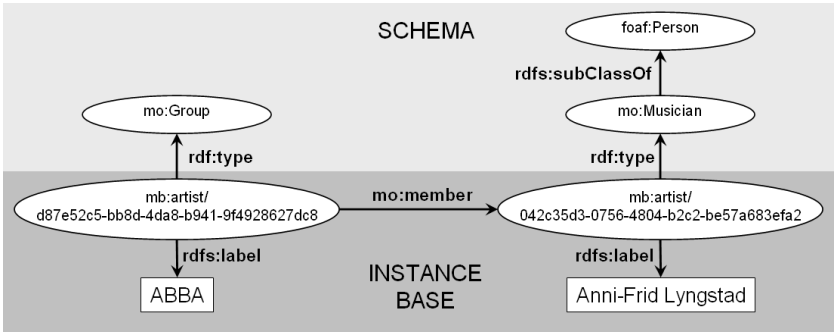

Abbildung 9. Visualisierung der RDF-Beispieldaten über ABBA und Anni-Frid Lyngstad.

Im zweiten Beispiel werden mittels des OWL-Konstrukts `owl:sameAs` zwei Ressourcen als identisch definiert, z.B. `http://www.bbc.co.uk/music/artists/d87e52c5-bb8d-4da8-b941-9f4928627dc8#artist` und `http://dbpedia.org/resource/ABBA`. Durch Schlussfolgerung können Informationen über ABBA aus verschiedenen Quellen verbunden werden (vgl. Abschnitt 3.3). Da Ontologien im Web unabhängig voneinander erstellt werden und URIs lokalen Namenskonventionen unterliegen, ist es durchaus möglich, dass ein reales Objekt durch verschiedene URIs (in verschiedenen Ontologien) referenziert wird.

OWL bietet noch eine Vielzahl weiterer Konstrukte zur Beschreibung von Klassen, Beziehungen und konkreten Fakten. OWL ermöglicht beispielsweise die Definition von transitiven Beziehungen und inversen Beziehungen (z.B. "ist-Mitglied" ist invers zu "hat-Mitglieder"). Für OWL-Ontologien gibt es Schlussfolgerungsdienste, die unter anderem Konsistenzprüfung einer Ontologie oder die Überprüfung der Erfüllbarkeit von Klassen ermöglichen. Ein Klasse ist erfüllbar, wenn es Instanzen dieser Klasse geben kann.

6 Identität und Verknüpfung von Objekten und Begriffen

Im Semantic Web kann nicht die Annahme getroffen werden, dass zwei URIs auf zwei verschiedene Entitäten verweisen. Eine URI hat von sich aus, beziehungsweise in sich, keine Identität [30]. Vielmehr wird die Identität beziehungsweise

Interpretation einer URI durch den Kontext, in dem sie im Semantic Web verwendet wird, deutlich. Zu bestimmen, ob zwei URIs auf dieselbe Entität verweisen oder nicht, ist keine einfache Aufgabe und wurde in der Vergangenheit intensiv im Data Mining und im Sprachverstehen untersucht. Um zu erkennen, ob sich die Autorennamen auf Forschungsbeiträgen auf dieselbe Person beziehen oder nicht, ist es oftmals nicht ausreichend den Namen, den Veranstaltungsort, Titel und Koautoren aufzulösen und zu betrachten [41].

Der Vorgang zur Bestimmung der Identität einer Ressource wird oftmals als Entitätenauflösung (englisch entity resolution) [41], Koreferenzauflösung [63], Objektidentifikation [48] und Normalisierung [63,64] bezeichnet. Die korrekte Bestimmung der Identität von Entitäten im Internet wird zunehmend wichtiger, da immer mehr Datensätze im Internet erscheinen und dies eine signifikante Hürde für sehr große Semantic Web Anwendungen darstellt [25].

Um dem gerecht zu werden, existieren eine Reihe von Diensten, die Entitäten erkennen und ihre Identität bestimmen können: Thomson Reuters bietet mit OpenCalais[18] einen Dienst an, mit dem Text in natürlicher Sprache mittels Erkennung von Entitäten mit anderen Ressourcen verknüpft werden kann. Ziel des OKKAM-Projekt[19] ist die Entwicklung von skalierbaren Systemen zur Erkennung von Entitäten im Internet wie beispielsweise Personen, Orte, Organisationen und Ereignisse und diese mit anderen Entitäten im Internet zu verknüpfen. Der Dienst sameAs[20] zielt auf die Erkennung von doppelten Ressourcen auf dem Semantic Web unter Verwendung der Rolle `owl:sameAs` ab. Damit können Koreferenzen zwischen verschiedenen Datensätzen aufgelöst werden, zum Beispiel wird für die Anfrage mit der URI `http://dbpedia.org/resource/ABBA` eine Liste von 21 Ressourcen zurückgegeben, die ebenfalls auf die Musikgruppe ABBA verweisen. Eine davon ist die BBC mit der Ressource `http://www.bbc.co.uk/music/artists/d87e52c5-bb8d-4da8-b941-9f4928627dc8#artist`.

Weiterhin ist das Problem des Schema-Matching [64] sehr verwandt mit dem der Auflösung von Entitäten, Koreferenzauflösung und Normalisierung. Ziel von Schema-Matching ist die, selbst für kleine Schemata nicht-triviale Frage, wie Daten integriert werden können [64]. Im Semantic Web bedeutet Schema-Matching der Abgleich von verschiedenen Ontologien beziehungsweise die in den diesen Ontologien definierten Konzepte. Verschiedene (halb-)automatische Verfahren oder Verfahren des maschinellen Lernen zum Abgleich von Ontologien wurden in der Vergangenheit entwickelt [18,16,7]. Kernontologien wie in Abbildung 4.2 stellen generische Modellierungsframeworks zur Integration und Abgleich mit anderer Ontologien dar. Zudem können Kernontologien auch Linked Open Data integrieren, das typischerweise keine oder nur sehr wenige Schema-Informationen beinhaltet. Die YAGO-Ontology [61] wurde generiert aus der Verschmelzung von Wikipedia und Wordnet unter Verwendung von regelbasierten und heuristischen Methoden. Eine manuelle Evaluation konnte eine Genauigkeit von 95% nachweisen. Ein manueller Abgleich von verschiedenen Datenquellen wird auch im

[18] `http://www.opencalais.com/`
[19] `http://www.okkam.org/`
[20] `http://sameas.org/`

Linked Open Data Projekt der Deutschen Nationalbibliothek verfolgt[21]. Beispielsweise wurde die Datenbank mit den Autoren aller in Deutschland publizierten Dokumente händisch mit der DBpedia und anderen Datenquellen verknüpft. Eine besondere Herausforderung war dabei die Identität der Autoren wie oben beschrieben zu identifizieren. Zum Beispiel hat der frühere Bundeskanzler Helmut Kohl einen Schiedsrichter als Namensvetter, dessen Arbeiten nicht mit dem DBpedia-Eintrag des Kanzlers verknüpft werden sollten. Beziehungen zwischen Schlüsselwörtern zur Beschreibung von Publikationen werden mit dem SKOS-Vokabular beschrieben. Beispielsweise werden Schlüsselwörter über die Relation `skos:related` zueinander in Beziehung gesetzt. Hyponyme und Hypernyme werden durch die Relationen `skos:narrower` und `skos:broader` ausgedrückt. Schließlich sei die Ontology Alignment Evaluation Initiative[22] erwähnt, die zum Ziel hat, einen etablierten Konsensus zur Evaluation von Methoden des Ontologie-Abgleichs zu erreichen.

7 Herkunft und Vertrauenswürdigkeit von Daten

Vertrauenswürdigkeit von Web-Seiten und Daten im Web kann anhand verschiedener Indikatoren erkannt werden, z.B. durch Zertifikate, anhand der Platzierung von Ergebnissen von Suchmaschinen, und über Links (Forward- und Backwardlinks) zu anderen Seiten. Allerdings gibt es im Semantic Web für Benutzer nur wenig Möglichkeiten, die Vertrauenswürdigkeit von einzelnen Daten zu bewerten.

Vertrauenswürdigkeit von Daten im Web kann von der Vertrauenswürdigkeit anderer Benutzer ("Wer sagt das?"), der zeitlichen Gültigkeit von Fakten ("Wann wurde ein Fakt beschrieben?") oder in Bezug auf Unsicherheit von Angaben ("Zu welchem Grad ist die Aussage wahr?") abgeleitet werden. Artz et Gil [2] fassen Vertrauenswürdigkeit wie folgt zusammen: "Vertrauenswürdigkeit von Daten ist kein neues Forschungsgebiet der Informatik, sondern sie existiert bereits in verschiedenen Bereichen der Informatik, z. B. in Form von Sicherheit und Zugriffskontrolle in Netzwerken, Zuverlässigkeit in verteilten Systemen, in Agentensystemen, und bei Richtlinien und Regeln zur Entscheidungsfindung unter Unsicherheit. Vertrauenswürdigkeit wird in jedem dieser Bereiche unterschiedlich behandelt."

Obwohl Vertrauenswürdigkeit in diesen Bereichen schon lange betrachtet wird, bringt die Bereitstellung und Veröffentlichung von Daten durch viele Benutzer an verschiedenen Quellen im Semantic Web neue und einzigartige Herausforderungen mich sich. Des Weiteren spielt Vertrauenswürdigkeit auch für Schlussfolgerungsdienste im Semantic Web eine Rolle, da bei der Herleitung von Daten Angaben bezüglich der Vertrauenswürdigkeit zu beachten sind und Daten nach ihrer Vertrauenswürdigkeit zu bewerten sind. Wichtige Aspekte für Vertrauenswürdigkeit von Daten sind unter anderem: (i) die Herkunft von Daten, (ii) das Vertrauen, das anhand vorheriger Interaktionen bereits gewonnen

[21] `http://www.d-nb.de/`
[22] `http://oaei.ontologymatching.org/`

wurde, (iii) Bewertungen, die durch Richtlinien eines Systems zugewiesen wurden, und (iv) Zugriffskontrollen und teilweise auch Sicherheit und Wichtigkeit von Informationen.

Diese Aspekte werden in verschiedenen Systemen realisiert. Datenherkunft und Vertrauenswürdigkeit von Daten im Semantic Web wurde für RDF-Daten in [14,20] und für OWL und Regeln in [14] behandelt. Vertrauen in sozialen und semantischen Netzwerken wird in [27,26] betrachtet, und im Zusammenhang von Richtlinien in Semantic Web Anwendungen in [9,57] erläutert. Andere Arbeiten befassen sich mit Zugriffskontrolle über verteilten Daten im Semantic Web [24]. Schließlich gibt es noch Ansätze zur Berechnung von Vertrauenswerten in [59], und zur Beurteilung der Relevanz von Datenquellen in [21] und von Teilgraphen in [42].

8 Semantic Web Anwendungen und Benutzerschnittstellen

Mit der zunehmenden Verbreitung und Verwendung von semantischen und verknüpften Daten im Web sind gleichzeitig neben den Anforderungen an Semantic Web Anwendungen auch deren Anwendungsmöglichkeiten gestiegen. Die Anforderungen sind anhand der Daten im Web gegeben. Anwendungen, die Daten aus relationalen Datenbanken oder XML-Dokumenten verwenden, können von einem festgelegten Schema ausgehen. Dies kann allerdings bei Daten im Web nicht vorausgesetzt werden. Oft sind weder die Datenquellen noch die Art und Menge der Daten einer Quelle vollständig bekannt.

Die Dynamik von semantischen Daten im Web muss von Anwendungen entsprechend berücksichtigt werden, sowohl bei der Anfrage und Aggregation von Daten, als auch bei der Visualisierung von Daten. Die eigentliche Herausforderung von Semantic Web Anwendungen liegt darin, eine bestmögliche Flexibilität der Anwendung zu garantieren, um die Dynamik von Datenquellen, Daten, und Schemas bei der Eingabe, Verarbeitung und Ausgabe zu berücksichtigen.

Nachfolgend werden vier Beispiele von Semantic Web Anwendungen bzw. Anwendungsbereichen vorgestellt. Sie verdeutlichen wie Flexibilität und Qualität der Suche, Integration, Aggregation und Darstellung von Daten aus dem Web realisiert werden kann. Sie zeigen zugleich das Potential von Semantic Web Anwendungen. Zunächst werden einheitliche Vokabulare und Schemas am Beispiel von *schema.org* vorgestellt. Sie dienen als Grundlage für eine semantische Suche, um Suchmaschinen Informationen über die Bedeutung von Inhalten von Web-Dokumenten zu geben. Die Suche und Integration von Daten aus verschiedenen Quellen wird von *Sig.ma* unterstützt. *Sig.ma* ist ein Semantic Web Browser. Andere Anwendungen realisieren semantische Suche durch weitere Repräsentationsformalismen (z.B. *Knowledge Graph*). Anschließend wird die *Facebook Graph-API*, eine Programmierschnittstelle zum Facebook-Graph, kurz vorgestellt. Schließlich wird *SemaPlorer* zur Visualisierung von heterogenen und verteilten Daten betrachtet. *SemaPlorer* verwendet verschiedene Facetten zur

Suche, wobei eine Facette im Wesentlichen ein Kriterium zur Aufteilung einer Datenmenge ist [51].

8.1 Vokabulare und Schemas

In HTML-Dokumenten kann die Struktur und der Aufbau von Seiten mit Tags beschrieben werden, nicht aber die Bedeutung der Informationen. Vokabulare, Schemas und Mikrodaten können als Mark-up in HTML-Dokumenten verwendet werden, um Angaben über Seiteninhalte und deren Bedeutung so zu beschreiben, dass Suchmaschinen diese Information verarbeiten können. Schema.org[23] ist eine Sammlung von Vokabularen und Schemas um HTML-Seiten mit zusätzlicher Information anzureichern.

Das Vokabular von *Schema.org* beinhaltet eine Menge von Entitäten und deren Eigenschaften. Eine universelle Entität "Thing" ist die allgemeinste Entität, die eine Art Oberbegriff aller Entitäten ist. Weitere geläufige Entitäten sind *Organization*, *Person*, *Event* und *Place*. Eigenschaften werden zur genaueren Beschreibung von Entitäten verwendet. Z.B. hat eine Person (Entität *Person*) Eigenschaften wie Name, Adresse und Geburtsdatum.

Neben Vokabularen wird in Schema.org auch die Anwendung von HTML-Mikrodaten festgelegt, mit dem Ziel, Daten in HTML-Dokumenten in einer möglichst eindeutigen Form darzustellen, so dass Suchmaschinen diese richtig interpretieren können. Ein Beispiel hierfür sind Formate für eindeutige Datum- und Zeitangaben, die auch Intervalle zur Angabe der Dauer von Ereignissen beschreiben können.

Unterstützt wird Schema.org unter anderem von den Suchmaschinen Bing, Google und Yandex. Es gibt Erweiterungen und Bibliotheken für verschiedene Programmiersprachen, u.a. für PHP, JavaScript, Ruby und Phyton, um Webseiten und Webanwendungen mit Vokabularen und Mikrodaten von Schema.org zu erstellen. Ebenso gibt es Abbildungen von Vokabularen und Mikrodaten aus Schema.org zu RDFS.

8.2 Semantic Web Browser und Semantische Suche

Ein Web Browser ermöglicht die Darstellung von Web-Seiten. Ein Semantic Web Browser geht noch einen Schritt weiter, indem zusätzlich noch die zugrundeliegende Informationen einzelner Seiten, die z.B. in Form von RDF-Metadaten vorliegen, dem Anwender visualisiert werden können. Somit sind gewöhnliche Nutzer in der Lage, Semantic Web Daten für ihre Informationssuche zu verwenden und auszunutzen.

Sig.ma [62] ist eine Anwendung zum (Durch-)Suchen von Semantic Web Daten, die aus mehreren verteilten Datenquellen stammen können. Sig.ma stellt eine API zur automatischen Integration von mehreren Datenquellen im Web zur Verfügung. Die angefragten Datenquellen beschreiben Informationen in RDF. Eine Suche in Sig.ma wird durch eine textuelle Anfrage vom Anwender gestartet.

[23] `http://schema.org`

Dabei kann nach Entitäten wie Personen, Orte oder Produkten gesucht werden. Ergebnisse einer Anfrage werden in aggregierter Form (Profil einer Entität) dargestellt, d.h. Eigenschaften der gesuchten Entität, z.B. einer Person, werden aus verschiedenen Datenquellen zusammengefasst dargestellt. Beispielsweise können bei einer Personensuche Informationen wie E-mail-Adresse, Anschrift oder aktueller Arbeitgeber angezeigt werden. Neben den eigentlichen Informationen werden auch Links zu den zugrundeliegenden Datenquellen angezeigt, um Anwendern eine Navigation zur Verfeinerung ihrer Suche zu ermöglichen. Sig.ma unterstützt auch strukturierte Anfragen, in denen zu einer Entität bestimmte Merkmale angefragt werden können, z.B. Kontaktdaten einer bestimmten Person.

Anfragen an Datenquellen erfolgen parallel. Die Ergebnisse aus den einzelnen Datenquellen in Form von RDF-Graphen werden zusammengefasst, indem Eigenschaften von Links in RDF-Daten, wie beispielsweise *owl:sameAs* oder inversfunktionale Prädikate, verwendet werden. Bei der Suche in Datenquellen werden Techniken wie Indexe, logische Inferenz und Heuristiken zur Datenaggregation verwendet.

Watson[24] ist ein Programm von IBM um Fragen, die in natürlicher Sprache gestellt sind, zu beantworten. Watson verwendet eine Vielzahl von Algorithmen, Techniken zur Verarbeitung von natürlichen Sprachen, Methoden aus dem Information Retrieval und Machine Learning, aber auch Wissensrepräsentation und Inferenz.

Google bietet mit Google Knowledge Graph[25] eine semantische Suchfunktion. Die englischsprachige Version der Suchmaschine ist um den *Knowledge Graph* erweitert. Ein Knowledge Graph ist ein Graph, in dem Entitäten (als Knoten) miteinander verknüpft sind, wobei diese Verknüpfungen Beziehungen zwischen den Entitäten darstellen. Tritt nun ein Suchbegriff in einer Anfrage auf, so wird nach der entsprechenden Entität im Knowledge Graph gesucht. Ausgehend von dieser Entität (Knoten) kann dann mittels der Verknüpfungen zu weiteren Entitäten navigiert werden.

8.3 Zugriff auf soziale Netzwerke

Ein soziales Netzwerk ist im wesentlichen ein Graph, in dem Verbindungen von Benutzern zu anderen Benutzern (z.B. in Form einer Freundschaftsbeziehung) oder zu Ereignissen und Gruppen existieren. Die Graph-API von Facebook beschreibt eine Programmierschnittstelle zum Facebook-Graph (genannt *Open Graph*). Innerhalb des Graphen werden Personen, Ereignisse, Seiten und Photos als Objekte dargestellt, wobei jedes Objekt einen eindeutigen Bezeichner hat, z.B. ist `https://graph.facebook.com/abba` der Bezeichner der Facebook-Seite von ABBA. Für die möglichen Beziehungsarten eines Objekts gibt es ebenfalls eindeutige Bezeichner, die das Navigieren von einem Objekt zu allen verbundenen Objekten bezüglich einer bestimmten Beziehung ermöglichen.

[24] `http://www-03.ibm.com/innovation/us/watson/index.html`
[25] `http://www.google.com/insidesearch/features/search/knowledge.html`

Die Graph-API ermöglicht neben dem Navigieren im Facebook-Graph und dem Lesen von Objekten, einschließlich deren Eigenschaften und Beziehungen zu anderen Objekten, auch das Erstellen von neuen Objekten im Facebook-Graph und das Bereitstellen von Applikationen. Die API unterstützt ebenfalls Anfragen von Metadaten eines Objekts, wie beispielsweise *wann* und *von wem* ein Objekt erstellt wurde.

8.4 Visualisierung semantisch heterogener und verteilter Daten

Neben der Bereitstellung und Suche im Web ist die Entwicklung flexibler Benutzungsschnittstellen zur interaktiven Visualisierung von verteilten, semantischen Daten ein weiterer wichtiger Aspekt. Ein Beispiel einer solchen Anwendung ist die interaktive Anwendung SemaPlorer [52] zur facettierten Suche und Navigation. Facetten sind insbesondere bei der Visualisierung von komplexen und großen Datenmengen hilfreich, z.B. SMILE Timeline Widget [39] für zeitliche Information und Google Maps[26] für räumliche Informationen. SemaPlorer ermöglicht ein interaktives Durchsuchen und Visualisieren von sehr großen und heterogenen semantischen Daten in Echtzeit entlang verschiedener Facetten. Die von SemaPlorer benutzten Datenquellen sind DBpedia, GeoNames[27], WordNet[28] und persönliche FOAF-Dateien. Zusätzlich wird ein Live-Wrapper zur Flickr-API[29] verwendet. SemaPlorer hat vier Facetten zur Suche und Visualisierung anhand von Orten, Personen, Tags und Zeit. Ein Screenshot von SemaPlorer ist in Abbildung 10 dargestellt. Textuelle Anfragen können auf der linken Seite eingegeben werden. Das entsprechende Anfrageergebnis kann über eine geeignete Facette betrachtet werden. Der Inhalt der aktuellen Facette, wie beispielsweise Orte und ortsbezogenen Informationen, ist auf der rechten Seite zu sehen. In der Mitte sind Visualisierungen und Bilder eines bestimmten Ortes zu sehen.

Bezüglich der Systemarchitektur ist SemaPlorer eine sehr flexible und generisch implementierte Anwendung. Allerdings sind die verschiedenen Facetten und die Daten, die in den jeweiligen Facetten bearbeitet und visualisiert werden können, fest in die Applikation eingebunden. Ähnlich zu vielen anderen Semantic Web Anwendungen sind die verwendeten Datenquellen direkt mit den einzelnen Bestandteilen der Applikation verbunden.

Ein weiteres Beispiel für eine Anwendung mit flexiblem Benutzungsinterface ist Paggr[30]. Paggr verwendet strukturierte, selbstbeschreibende Daten auf dem Web um ad-hoc semantische Mashups zu erzeugen und in einem personalisierten Dashboard anzuzeigen. Schließlich bietet der RDF-Browser Lena [21] ein flexible Darstellung von RDF-Daten mit Hilfe des Fresnel Display Vocabulary[31].

[26] `http://maps.google.com`
[27] `http://geonames.org`
[28] `http://wordnet.princeton.edu`
[29] `http://flickr.com`
[30] `http://www.paggr.com/`
[31] `http://www.w3.org/2005/04/fresnel-info/`

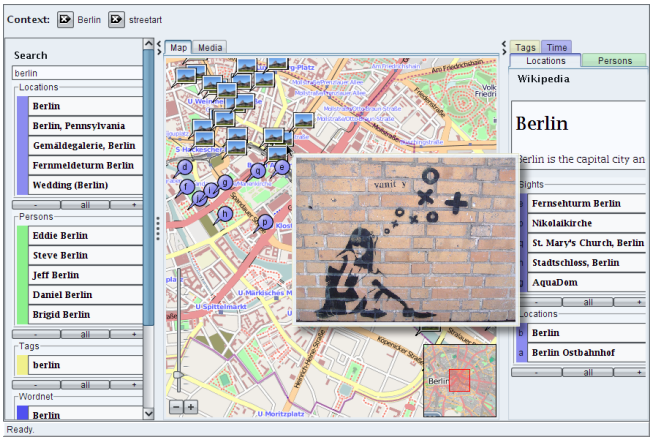

Abbildung 10. Facettierte Suche und Navigation von Semantischen Daten in SemaPlorer [52].

9 Zusammenfassung und Ausblick

Das Semantic Web besteht aus einer Vielzahl von Techniken, die stark von der Langzeitforschung der Künstlichen Intelligenz und deren Ergebnissen beeinflusst wurden. Methoden aus der Künstliche Intelligenz zur Modellierung, Darstellung und Inferenz werden um verteiltes Wissen und verteilte Wissensrepräsentation erweitert. Ausgehend von dem Web in seiner aktueller Form beinhaltet das Semantic Web weitere Standards und Sprachen um die Semantik von Dokumenten und Daten in maschinenverarbeitbarer Form darzustellen. Das volle Potential des Semantic Webs wurde allerdings noch nicht vollständig ausgenutzt, da vor allem einige wichtige Komponenten der Semantic Web Architektur noch erforscht werden, z.B. die Datenherkunft und Vertrauenswürdigkeit.

Allerdings gewinnt das Semantic Web stetig an Bedeutung. In den letzten Jahren hat die Veröffentlichung von Linked Data enorm zugenommen, insbesondere in den Bereichen bibliographischer Informationsverwaltung, Bioinformatik und E-Government. DBpedia ist dabei die Kerndatenquelle, um die verschiedenen Bereiche gruppiert sind (vgl. [6]). Dies wird beispielsweise an dem enormen Wachstum der Linked Open Data Cloud[32] seit 2007 deutlich. Dieses schnelle Wachstum hat schon Einfluss auf die Industrie. Mit Schema.org werden Schemata zur ausführlicheren Beschreibung von Daten auf Webseiten definiert, um Informationen über die zugrundeliegende Datenstrukturen und die Bedeutung der Daten zu geben. Suchmaschinen können diese zusätzliche Information nutzen, um die Inhalte von Webseiten besser analysieren zu können. Schema.org wird von den Suchmaschinen Bing, Google und Yandex unterstützt.

[32] Das Wachstum der Linked Open Data Cloud wird dokumentiert unter: `http://linkeddata.org/`.

Studien auf ausgewählten Quellen haben gezeigt, dass Webseiten unter den Top-10 Ergebnissen eine um bis zu 15 % höhere Klickrate haben[33]. Andere Unternehmen wie BestBuy.com berichten sogar von bis zu 30 % höheren Zugriffsraten, seit der Erweiterung ihrer Webseiten mit semantischen Daten (vgl. Abschnitt 8) in 2009. BestBuy.com benutzt das GoodRelations Vokabular[34] um Online-Angebote zu beschreiben. Ebenso hat Google begonnen semantische Daten von Online-Handelsportalen, die das GoodRelations Vokabular benutzen, bei der Suche zu berücksichtigen[35].

Ein weiterer Erfolg ist die Veröffentlichung von Regierungsdaten. Zum Beispiel stellt die US-Regierung mit Data.gov[36] Regierungsdaten öffentlich bereit, und US Census[37] veröffentlicht statistische Daten über die USA. In Großbritannien ist data.gov.uk[38] ein wesentlicher Teil eines Programms zu mehr Transparenz von Daten im öffentlichen Sektor. Auch in Deutschland werden zunehmend offene Daten frei zur Verfügung gestellt. Eine Übersicht über offene Daten in Deutschland ist u.a. im Katalog für offene Daten unter `http://de.ckan.net` dargestellt.

Schließlich kann ein starkes Wachstum von semantischen Daten der Biomedizin im Web festgestellt werden. Im Rahmen von Bio2RDF[39] wurden viele Datenbanken der Bioinformatik miteinander verknüpft. Die Transinsight GmbH bietet die wissensbasierte Suchmaschine GoPubMed[40] an, um Forschungsartikel der Biomedizin zu finden. Ontologien werden zur Suche verwendet.

Zusammenfassend kann beobachtet werden, dass semantische Daten im Web einen echten Einfluss auf kommerzielle Anbieter von Produkten und Dienstleistungen und auch auf Regierungen und öffentliche Verwaltungen haben. Dies verspricht eine erfolgreiche Zukunft des Semantic Webs.

Danksagung

Teile dieses Kapitels basieren auf den Veröffentlichungen von Janik et al. [40] und Harth et al. [32].

Literatur

1. D. Allemang and J. Hendler. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufmann, 2011.
2. D. Artz and Y. Gil. A Survey of Trust in Computer Science and the Semantic Web. *J. Web Sem.*, 5(2):58–71, 2007.

[33] `http://developer.yahoo.net/blog/archives/2008/07/`
[34] `http://www.heppnetz.de/projects/goodrelations/`
[35] `http://www.ebusiness-unibw.org/wiki/GoodRelationsInGoogle#GoodRelations_in_Google_Rich_Snippets`
[36] `http://www.data.gov/`
[37] `http://www.rdfabout.com/demo/census/`
[38] `http://data.gov.uk`
[39] `http://bio2rdf.org/`
[40] `http://www.gopubmed.org/`

3. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In *Semantic Web Conference and Asian Semantic Web Conference*, pages 722–735, November 2008.
4. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
5. T. Berners-Lee. Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web. RFC 1630, Internet Engineering Task Force, June 1994.
6. C. Bizer. The Emerging Web of Linked Data. *IEEE Intelligent Systems*, 24(5):87–92, 2009.
7. E. Blomqvist. Ontocase-automatic ontology enrichment based on ontology design patterns. In *International Semantic Web Conference*, pages 65–80, 2009.
8. H. Boley, G. Hallmark, M. Kifer, A. Paschke, A. Polleres, and D. Reynolds. RIF Core Dialect. W3C candidate recommendation, W3C, October 2009. http://www.w3.org/TR/rif-core/.
9. P. A. Bonatti and D. Olmedilla. Rule-Based Policy Representation and Reasoning for the Semantic Web. In *Reasoning Web Summer School*, volume 4636 of *Lecture Notes in Computer Science*, pages 240–268. Springer, 2007.
10. S. Borgo and C. Masolo. *Handbook on Ontologies*, chapter Foundational choices in DOLCE. Springer, 2nd edition, 2009.
11. M. Braun, A. Scherp, and S. Staab. Collaborative semantic points of interests. In *The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010, Proceedings, Part II*, volume 6089 of *Lecture Notes in Computer Science*, pages 365–369. Springer, 2010.
12. D. Brickley and R. Guha. RDF vocabulary description language 1.0: RDF schema. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/rdf-schema/.
13. J. Broekstra, A. Kampman, and F. V. Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *International Semantic Web Conference*, pages 54–68. Springer, 2002.
14. R. Q. Dividino, S. Schenk, S. Sizov, and S. Staab. Provenance, Trust, Explanations - and all that other Meta Knowledge. *KI*, 23(2):24–30, 2009.
15. M. Duerst and M. Suignard. Internationalized resource identifiers (IRIs). RFC 3987, Internet Engineering Task Force, Jan. 2005.
16. M. Ehrig. *Ontology Alignment: Bridging the Semantic Gap*, volume 4 of *Semantic Web and Beyond*. Springer, 2007.
17. J. Euzenat and P. Shvaiko. *Ontology matching*, chapter Classifications of ontology matching techniques. Springer, 2007.
18. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007.
19. R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, USA, 2000.
20. G. Flouris, I. Fundulaki, P. Pediaditis, Y. Theoharis, and V. Christophides. Coloring RDF Triples to Capture Provenance. In *International Semantic Web Conference*, volume 5823 of *LNCS*, pages 196–212. Springer, 2009.
21. T. Franz, A. Schultz, S. Sizov, and S. Staab. TripleRank: Ranking Semantic Web Data by Tensor Decomposition. In *International Semantic Web Conference*, volume 5823 of *LNCS*, pages 213–228. Springer, 2009.
22. T. Franz, S. Staab, and R. Arndt. The X-COSIM integration framework for a seamless semantic desktop. In *Knowledge capture*, pages 143–150. ACM, 2007.

23. A. Gangemi and V. Presutti. *Handbook on Ontologies*, chapter Ontology Design Patterns. Springer, 2nd edition, 2009.
24. R. Gavriloaie, W. Nejdl, D. Olmedilla, K. E. Seamons, and M. Winslett. No Registration Needed: How to Use Declarative Policies and Negotiation to Access Sensitive Resources on the Semantic Web. In *European Semantic Web Symposium*, volume 3053 of *LNCS*, pages 342–356. Springer, 2004.
25. H. Glaser, A. Jaffri, and I. Millard. Managing co-reference on the semantic web. In *WWW2009 Workshop: Linked Data on the Web*, 2009.
26. J. Golbeck and J. A. Hendler. Inferring binary trust relationships in Web-based social networks. *ACM Trans. Internet Techn.*, 6(4):497–529, 2006.
27. J. Golbeck and A. Mannes. Using Trust and Provenance for Content Filtering on the Semantic Web. In *Models of Trust for the Web*, CEUR Workshop Proceedings. CEUR-WS.org, 2006.
28. A. Gómez-Pérez, M. F. López, and O. Corcho. *Ontological engineering*. Springer, 2004.
29. W. O. W. Group. OWL 2 Web Ontology Language Document Overview . W3C recommendation, W3C, October 2009. http://www.w3.org/TR/owl2-overview/.
30. H. Halpin and V. Presutti. An ontology of resources: Solving the identity crisis. In *European Semantic Web Conference*, pages 521–534, 2009.
31. A. Harth, K. Hose, M. Karnstedt, A. Polleres, K.-U. Sattler, and J. Umbrich. Data Summaries for On-Demand Queries over Linked Data. In *World Wide Web*. ACM, 2010.
32. A. Harth, M. Janik, and S. Staab. Semantic Web Architecture. *Handbook of Semantic Web Techologies*, 1:43–76, 2011.
33. O. Hartig. SPARQL for a Web of Linked Data: Semantics and Computability. In *9th Extended Semantic Web Conference (ESWC)*, volume 7295 of *LNCS*, pages 8–23. Springer, 2012.
34. O. Hartig, C. Bizer, and J. C. Freytag. Executing SPARQL Queries over the Web of Linked Data. In *International Semantic Web Conference*, pages 293–309, 2009.
35. O. Hartig and J.-C. Freytag. Foundations of Traversal Based Query Execution over Linked Data. In *23rd ACM Conference on Hypertext and Social Media, HT*, pages 43–52, 2012.
36. J. Hebeler, M. Fisher, R. Blace, and A. Perez-Lopez. *Semantic Web Programming*. Wiley, 2011.
37. J. Heinsohn, D. Kudenko, B. Nebel, and H.-J. Profitlich. An Empirical Analysis of Terminological Representation Systems. *Artif. Intell.*, 68(2):367–397, 1994.
38. P. Hitzler, M. Krötzsch, S. Rudolph, and Y. Sure. *Semantic Web: Grundlagen*. Springer-Verlag, 2008.
39. D. F. Huynh, D. R. Karger, and R. C. Miller. Exhibit: Lightweight Structured Data Publishing. In *World Wide Web*, pages 737–746. ACM, 2007.
40. M. Janik, A. Scherp, and S. Staab. The Semantic Web: Collective Intelligence on the Web. *Informatik Spektrum*, 34(5):469–483, 2011.
41. P. Kanani, A. McCallum, and C. Pal. Improving author coreference by resource-bounded information gathering from the web. In *Conference on Artifical intelligence*, pages 429–434, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
42. G. Kasneci, S. Elbassuoni, and G. Weikum. MING: Mining Informative Entity Relationship Subgraphs. In *Information and Knowledge Management*, pages 1653–1656. ACM, 2009.
43. D. Oberle. *Semantic Management of Middleware*. Springer, 2006.

44. D. Oberle, A. Ankolekar, P. Hitzler, P. Cimiano, M. Sintek, M. Kiesel, B. Mougouie, S. Baumann, S. Vembu, M. Romanelli, P. Buitelaar, R. Engel, D. Sonntag, N. Reithinger, B. Loos, H.-P. Zorn, V. Micelli, R. Porzel, C. Schmidt, M. Weiten, F. Burkhardt, and J. Zhou. Dolce ergo sumo: On foundational and domain models in the smartweb integrated ontology (swinto). *Web Semant.*, 5(3):156–174, Sept. 2007.
45. D. Oberle, N. Guarino, and S. Staab. What is an ontology? In S. Staab and R. Studer, editors, *Handbook on Ontologies*. Springer, 2nd edition, 2009.
46. Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. Object Exchange Across Heterogeneous Information Sources. In *Data Engineering*, pages 251–260, Washington, DC, USA, 1995. IEEE Computer Society.
47. Y. Raimond, C. Sutton, and M. B. Sandler. Interlinking Music-Related Data on the Web. *IEEE MultiMedia*, 16(2):52–63, 2009.
48. S. Rendle and L. Schmidt-Thieme. Object identification with constraints. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, pages 1026–1031. IEEE Computer Society, 2006.
49. C. Saathoff and A. Scherp. Unlocking the semantics of multimedia presentations in the web with the multimedia metadata ontology. In M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 831–840. ACM, 2010.
50. C. Saathoff and A. Scherp. Unlocking the semantics of multimedia presentations in the web with the multimedia metadata ontology. In *World Wide Web*, 2010.
51. G. M. Sacco and Y. Tzitzikas, editors. *Dynamic Taxonomies and Faceted Search : Theory, Practice, and Experience*. Springer, Berlin, 2009.
52. S. Schenk, C. Saathoff, S. Staab, and A. Scherp. SemaPlorer—interactive semantic exploration of data and media based on a federated cloud infrastructure. *Journal of Web Semantics*, 2009.
53. S. Schenk and S. Staab. Networked graphs: a declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the web. In *World Wide Web*, pages 585–594. ACM, Apr. 21-25, 2008.
54. A. Scherp, D. Eißing, and C. Saathoff. A method for integrating multimedia metadata standards and metadata formats with the multimedia metadata ontology. *Int. Journal on Semantic Computing*, 2012.
55. A. Scherp, T. Franz, C. Saathoff, and S. Staab. A core ontology on events for representing occurrences in the real world. *Multimedia Tools Appl.*, 58(2):293–331, 2012.
56. A. Scherp, C. Saathoff, T. Franz, and S. Staab. Designing core ontologies. *Applied Ontology*, 6(3):177–221, 2011.
57. F. Schwagereit, A. Scherp, and S. Staab. Representing Distributed Groups with dgFOAF. In *Extended Semantic Web Conference*, LNCS. Springer, 2010.
58. S. Staab and R. Studer, editors. *Handbook on Ontologies*. Springer, 2009.
59. G. Stoilos, G. B. Stamou, J. Z. Pan, V. Tzouvaras, and I. Horrocks. Reasoning with Very Expressive Fuzzy Description Logics. *J. Artif. Intell. Res.*, 30:273–320, 2007.
60. H. Stuckenschmidt, R. Vdovjak, J. Broekstra, and G.-J. Houben. Towards distributed processing of RDF path queries. *Int. J. Web Eng. Technol.*, 2(2/3):207–230, 2005.

61. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706, New York, NY, USA, 2007. ACM.
62. G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, and S. Decker. Sig.ma: live views on the Web of Data. In *Semantic Web Challenge 2009 at the 8th International Semantic Web Conference (ISWC2009)*, 2009.
63. M. L. Wick, A. Culotta, K. Rohanimanesh, and A. McCallum. An entity based model for coreference resolution. In *SIAM International Conference on Data Mining*, pages 365–376, 2009.
64. M. L. Wick, K. Rohanimanesh, K. Schultz, and A. McCallum. A unified approach for schema matching, coreference and canonicalization. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 722–730, New York, NY, USA, 2008. ACM.
65. K. Wilkinson, C. Sayers, H. A. Kuno, and D. Reynolds. Efficient RDF storage and retrieval in Jena2. In I. F. Cruz, V. Kashyap, S. Decker, and R. Eckstein, editors, *SWDB*, pages 131–150, 2003.