
2/9 Semantic Web

**Autoren: Dipl.-Inf. Fabian Wleklinski,
Dipl.-Math. Karsten Tolle**

2/9.1 Einleitung

Die viel beschworene Informationsgesellschaft ist da: Internetzugänge für wenige Cents, Milliarden von Webseiten, mehrere neue Webseiten pro Sekunde: gemeinsam mit der Informationsgesellschaft kam die Informationsüberflutung – genannt *Data Smog*. Internet-Suchmaschinen wie Google oder AltaVista werden der Flut an Daten nicht mehr Herr [1].

Data Smog

Das *Semantic Web* (*Semantisches Web*), kurz SW, wird als nächster Evolutionsschritt des Internet propagiert [2, 3]. Dabei werden zusätzlich zu den für Menschen gedachten Bildern, Musikdateien und Texten auch so genannte *Metadaten* in einer standardisierten Form veröffentlicht. Diese Metadaten können von Maschinen gelesen und „verstanden“ werden. Durch das Verständnis der Bedeutung, der Zugriffsfähigkeit oder der Qualität der Daten können bessere und neue Dienste im Internet angeboten werden. Dies wird vielleicht in naher Zukunft unser alltägliches Leben und unser Arbeitsumfeld ähnlich stark beeinflussen, wie dies durch das Internet geschehen ist.

Metadaten

Internet-Suchmaschinen sind nur eine Anwendung für das Semantic Web, aber eine wichtige. Sie unterliegen heute zahlreichen Problemen: Synonyme, Homonyme, Rechtschreibfehler und verschiedene Wortformen verhindern eine exakte Suche, und den Sinnzusammenhang versuchen Internet-Suchmaschinen erst gar nicht zu „verstehen“. Ein „Jaguar“ ist für eine Internet-Suchmaschine eine Katze, ein Flugzeug, ein Auto und ein Fahrrad zugleich. Das Semantic Web wird genau das ermöglichen: Maschinen werden Webseiten „verstehen“.

**Probleme heutiger
Suchmaschinen**

Einleitung

Die Vision des Semantic Web wird u.a. in [4, 5, 6] durch Anwendungsszenarien mit Leben gefüllt. Die Szenarien reichen von erweiterten Suchmöglichkeiten bis hin zu komplexen Aktionsketten, z.B. selbstständige Reiseplanung und Buchung oder automatische Terminabsprache durch Abgleich elektronischer Terminkalender.

Aufbau des SW

Das Semantic Web ist keine einzelne Technik, sondern basiert auf einer Menge verschiedener Einzeltechnologien wie XML¹, HTTP², URIs³, RDF⁴, RDF Schema und OWL, die man in Gesamtheit „Semantic Web“ nennt. Oft spricht man auch von verschiedenen „Schichten“ des Semantic Web, da diese Techniken aufeinander aufbauen: jede Technik nutzt die Dienste anderer, zugrunde liegender Techniken. Mit dem klassischen Schichtenmodell der Softwareentwicklung haben diese Schichten allerdings nichts zu tun. Das Schichtenmodell wird in Abbildung 2/9.1-1 grafisch dargestellt.

Neben den Standards RDF und RDF Schema hat das World Wide Web Consortium (W3C)⁵ für die Ontologie-Ebene die *Web Ontology Language* (OWL) als Standard verabschiedet. Eine *Ontologie* ist ein formal definiertes System von Konzepten und Beziehungen mit zusätzlichen Inferenz- und Integritätsregeln. OWL baut dabei auf RDF Schema auf und ergänzt es, sodass logische Axiome ausgedrückt werden können. Als Beispiel sei hier die Transitivität von Eigenschaften oder Mengenbeziehungen zwischen Klassen (disjunkt, überlappend) genannt. In der [OWL]-Spezifikation sind drei aufeinander aufbauende Sprachen mit wachsender Ausdrucksfähigkeit definiert: OWL lite, OWL DL⁶ und OWL full.

¹ Extensible Markup Language

² Hypertext Transfer Protocol

³ Uniform Resource Identifier

⁴ Resource Description Framework

⁵ <http://www.w3.org>

⁶ DL steht für *Description Logic*

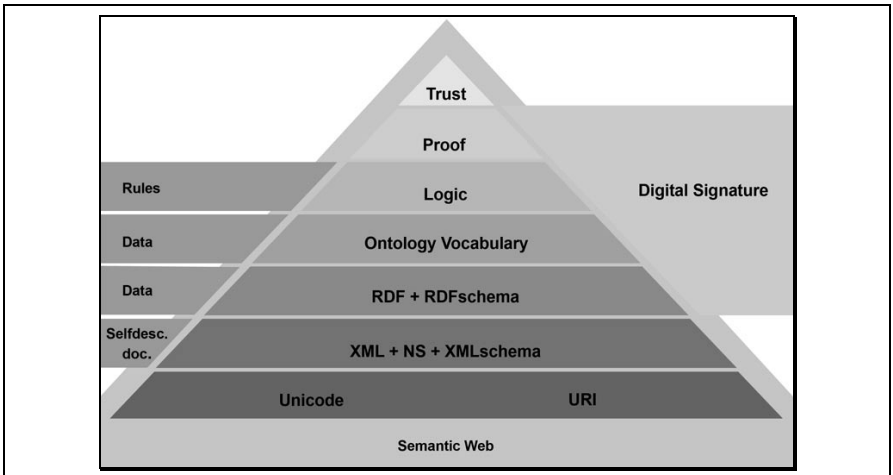


Abbildung 2/9.1-1: Schichtenmodell der Semantic-Web-Standards

Es ist zu erwarten, dass das Semantic Web in Zukunft auch für die weiteren Schichten Standards anbieten wird, z.B. zur Hinterlegung von Vertrauenswürdigkeit („Trust“, siehe dazu auch [Schichten]).

OWL

Eine einfache Beschreibung des Semantic Web ist, dass es sich um einen ersten Versuch handelt, für Maschinen das zu erreichen, was das World Wide Web (kurz Web, WWW) für die Menschen erreicht hat. Das Web wird von einem Hyperlink-Buch zu einer Hyperlink-Datenbank erweitert.

Oftmals ist die Rede davon „dass es das Semantic Web ermöglichen wird, dieses oder jenes zu tun“. Diese Aussagen klingen ebenso visionär wie noch vor wenigen Jahrzehnten die Idee, irgendwann einmal „via Datennetz“ Bankgeschäfte tätigen zu können. Derlei Aussagen wirken irreführend, weil sie suggerieren, dass das Semantic Web zunächst existieren muss, bevor es nützlich werden kann (in einer fernen Zukunft). Das unterschlägt die Tatsache, dass sich das Semantic Web Stück für Stück und vom Anwender weitestgehend unbemerkt „innerhalb“ des Internet etabliert und nicht etwa „neben dem Internet“ oder „statt des Internet“.

Einleitung

Dieselbe Aussage klingt weniger visionär, wenn man sie so formuliert: „die Techniken des Semantic Web ermöglichen es, dieses oder jenes zu tun“. „Semantic Web“ bezeichnet nicht etwa den Nachfolger des Web, sondern dessen schleichende Erweiterung um neue Techniken wie RSS⁷, RDF und OWL –zusammengefasst unter dem Begriff „Semantic Web“.

Probleme des SW

Eine gewisse Skepsis gegenüber dem Semantic Web ist sicherlich gerechtfertigt. So vergleicht R. V. Guha laut [5] die oft propagierte Möglichkeit des Ableitens (Inferenz) neuer Fakten aus Semantic-Web-Daten mit dem *Traveling-Salesman-Problem*⁸. Hierdurch wird klar, dass zwar vieles möglich ist, aber auch die Kosten hierfür beträchtlich sein können.

Hohe Komplexität

Ein weiteres Problem stellt die Komplexität des Semantic Web dar, die mittlerweile erreicht wurde. Zwar sind die Grundgedanken des Semantic Web einfach zu verstehen, jedoch stößt man allein beim Resource Description Framework (RDF) auf mittlerweile sechs W3C-Vorschläge. Eine breite Nutzung im nicht professionellen Bereich ist daher in naher Zukunft nicht zu erwarten. Man beachte, dass RDF mit seinen sechs Standards dabei nur das Fundament (siehe Abbildung 2/9.1-1) des Semantic Web darstellt, und höhere Ebenen auf RDF aufbauen, die die Komplexität des Semantic Web weiter erhöhen.

Dieses Dokument möchte die Einsatzmöglichkeiten der Techniken des Semantic Web vermitteln, wie RDF, RDF Schema und OWL. Auf die formalen Definitionen der erwähnten Standards wird im Rahmen dieses Dokuments nicht eingegangen, diese können bei Bedarf in den frei zugänglichen Quellen nachgelesen werden. An den entsprechenden Stellen auf wird diese Quellen hingewiesen. Vorkenntnisse in den erwähnten Techniken sind nicht nötig, helfen aber beim Verständnis.

⁷ *RDF Site Summary* oder auch *Really Simple Syndication*

⁸ Auch „Problem des Handlungsreisenden“ genannt, siehe online http://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden.

2/9.1.1 RDF = Semantic Web?

Nein. RDF unterstützt das Semantic Web, wie HTML das Web unterstützt. HTML allein ist aber nicht das Web, und RDF ist nicht das Semantic Web.

Zitat ([10]): The Resource Description Framework (RDF) is a language designed to support the Semantic Web, in much the same way that HTML is the language that helped initiate the original Web. RDF is a framework for supporting resource description, or metadata (data about data), for the Web. RDF provides common structures that can be used for interoperable XML data exchange.

RDF ist eine universell einsetzbare Sprache für verschiedene Einsatzzwecke. RDF baut auf anderen Techniken wie URIs und XML auf, aber dient seinerseits als Basistechnik für andere, wiederum darauf aufbauende Techniken wie RSS, OWL und RDF Schema. RDF ist in Kombination mit der OWL ein grundlegendes Format zur Repräsentation von Taxonomien und Ontologien (d.h. formalen Vokabularen).

**Universelle
Sprache**

RDF ist seinen Kinderschuhen entwachsen und wird mittlerweile in vielen weitläufig genutzten Applikationen eingesetzt, z.B. als Übertragungsformat für Content-Syndikation via RSS, als Metadaten-Speicherformat *Extensible Metadata Platform* (XMP) in vielen Adobe-Produkten (Adobe Acrobat), für Kommentierungssysteme (Annotea) und Soziale Netzwerke (FOAF: Friend Of A Friend).

2/9.1.2 Ein paar Worte zu RDF & RDF Schema

Eine zentrale Rolle für das Semantic Web und für diese Arbeit spielt das *Resource Description Framework* (RDF).

RDF-Modell

Einleitung

Es wurde unter Leitung des W3C entwickelt und legt fest, wie die Metadaten erstellt und für den Datenaustausch in XML codiert werden müssen. Das zugrunde liegende *RDF-Modell* basiert auf einem gerichteten und beschrifteten Graphen. Die Knoten des Graphen werden *Ressourcen* und die Kanten *Eigenschaften* genannt.

RDF-Aussagen

Zwei miteinander durch eine Eigenschaft verbundene Ressourcen werden als *RDF-Aussage* bezeichnet. Sie können als einfacher Satz mit Subjekt, Prädikat und Objekt verstanden werden. Durch das Zusammenfügen vieler kleiner Sätze können so komplizierte Sachverhalte beschrieben werden.

RDF Schema

Neben dem RDF-Modell existiert die Beschreibungssprache *RDF Schema*, die ihrerseits wiederum auf dem RDF-Modell aufbaut. Durch RDF Schema ist es möglich, eigene Eigenschaften, Klassen oder weitere RDF-Konstrukte zu erzeugen. Es wird dadurch jedem ermöglicht, ein eigenes Vokabular zu definieren, zu nutzen und zu veröffentlichen.

RDF und RDF Schema sind kein Dateiformat, und auch nicht als solches definiert. Bei der Definition von RDF und RDF Schema wird stattdessen ein mathematisches Modell verwendet, um die Semantik der zulässigen Operationen darauf mit Formeln und logischen Operatoren definieren zu können.

XML-Serialisierung

Um die Daten austauschen zu können, wurde eine verlustfreie *XML-Serialisierung* (oder XML-Codierung) der Aussagen entwickelt, genannt *RDF/XML*. Diese wird in [7] festgelegt. Sie erlaubt Spielraum für z.B. verkürzte Schreibweisen, wodurch ein RDF-Graph zu unterschiedlichen XML-Serialisierungen führen kann. Bei deren Rückübersetzung wird jedoch jeweils der Ausgangsgraph erzeugt. Wichtig ist die Unterscheidung, dass es sich bei RDF nicht direkt um eine XML-Anwendung handelt, sondern dass das RDF-Modell lediglich von und nach XML serialisiert werden kann.

Notation3 (N3)

Als weitere, weniger aufgeblähte Serialisierung hat sich *Notation3* oder kurz *N3* etabliert. Für eine geringe Anzahl von RDF-Aussagen ist N3 auch für den Menschen sehr gut lesbar und wird oft bei kleineren Beispielen verwendet.

Es gibt hierfür jedoch auch umfangreiche Abkürzungsvarianten, und N3 wird von diversen Programmen als Eingabe unterstützt.

Das erste *Arbeitspapier (Working Draft)* zur *RDF Model and Syntax Specification* wurde 1997 veröffentlicht. Anfang 1999 wurde diese Spezifikation als W3C-Vorschlag⁹ verabschiedet. Ergänzend gab es nur die *RDF Schema Specification 1.0*, die jedoch noch nicht den Status eines W3C-Vorschlags (*W3C Recommendation*) erreicht hatte und später auf die Ebene eines Arbeitspapiers zurückgestuft wurde. Im Laufe der Zeit wurden vier weitere Spezifikationen in die RDF-Standard-Familie aufgenommen. In Abbildung 2/9.1-2 ist die zeitliche Entwicklung der RDF-Standards visualisiert.

Historische Entwicklung von RDF

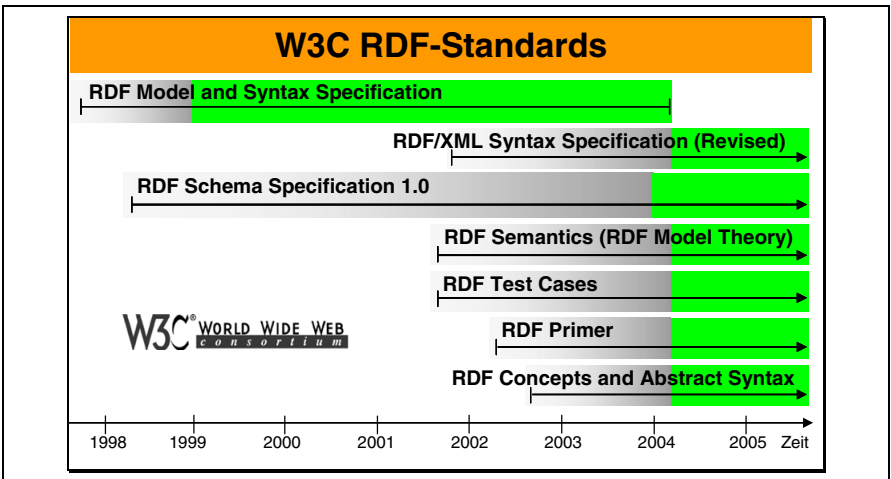


Abbildung 2/9.1-2: Übersicht über die zeitliche Entwicklung der RDF-Standards. Der Farbwechsel ins Dunklere soll dabei eine höhere Reife bzw. Gültigkeit anzeigen.

⁹ Ein W3C-Vorschlag ist mit einem Standard gleichzusetzen.

Einleitung

2/9.1.3 RDF oder XML?

Gemeinsamkeiten RDF ↔ XML

RDF eignet sich als Technologie zur Speicherung, Verarbeitung und Übertragung von Daten. Daher kann RDF oft alternativ zu XML oder zu dessen Vorläufern GML/SGML¹⁰ eingesetzt werden. Bezogen auf einen konkreten Anwendungszweck gibt es erhebliche Gemeinsamkeiten zwischen RDF und XML, weshalb eine Entscheidung für oder gegen RDF oft schwer ist:

- Sowohl RDF als auch XML sind kostenlose („freie“) und gut anpassbare Technologien.
- Sowohl RDF als auch XML verfügen über ein Dokumentenformat zur Übertragung, Speicherung und Archivierung. Im Fall von XML ist das die standardisierte, HTML-ähnliche XML-Syntax:

```
<title>Semantic Web</title>.
```

Zusätzlich gibt es für XML weitere, nicht standardisierte Dokumentenformate, die z.B. eine „Platz sparendere“ Binärschreibweise ermöglichen. RDF kennt gleich zwei Dokumentenformate, genannt „Serialisierungen“:

- N3 (einfaches Textformat ähnlich zu CSV¹¹)
- XML.
- Sowohl RDF als auch XML verfügen über ein Datenmodell zur Verarbeitung oder Manipulation von Daten, [RDF Model] bzw. [XML Infoset]. Dieses ist vom Dokumentenformat losgelöst definiert. Vereinfacht ausgedrückt bedeutet dies, dass ein RDF- oder XML-Dokument zwar in verschiedenen Formaten gespeichert werden kann, diese aber trotzdem demselben Dokument entsprechen.

¹⁰ (Standard) Generalized Markup Language

¹¹ Comma Separated Values

- Sowohl RDF als auch XML können durch ergänzende Technologien erweitert werden wie [XML Schema]/[DTD] bzw. [RDF Schema] und [OWL] für die Validierung von Dokumenten.
- Sowohl RDF als auch XML können auf zahlreiche Werkzeuge zur Bearbeitung, Speicherung und Visualisierung verweisen (einige Werkzeuge für RDF werden im folgenden Kapitel vorgestellt).
- Sowohl RDF als auch XML unterstützen *Namensräume* (bzw. englisch *Namespaces*) für die Zusammenführung verschiedener Daten.
- RDF-Dokumente sind entweder bereits XML-Dokumente oder können via XML-Serialisierung in XML-Dokumente umgewandelt werden. Umgekehrt können XML-Dokumente in gültige RDF-Dokumente überführt werden. Beide Umwandlungen sind über standardisierte Techniken wie [XQuery] oder [XSLT] leicht realisierbar. Die Tatsache, dass ein anderes IT-System XML- oder RDF-Daten verlangt, ist also kein zwingendes Argument für XML oder RDF. [RSS] veranschaulicht diese Dualität sehr gut, da es von einigen Betrachtern als RDF-Variante und von anderen als XML-Variante verstanden wird.

Anders als im Fall von [XML Schema], wo ein Schema eine bestimmte Art von Dokument vollständig und abschließend beschreibt (auch Grammatik genannt), werden mit RDF Schema so genannte Vokabulare beschrieben. Der entscheidende Vorteil der Vokabulare besteht darin, dass sie miteinander kombiniert werden können –auch innerhalb ein und desselben Dokuments.

Mit XML Schema ist dies nicht möglich. Dadurch unterstützt RDF Schema ausdrücklich die Vermischung von Daten verschiedener und voneinander unabhängiger Anwendungen.

Für einen Vergleich von RDF Schema mit XML Schema und Typsystemen der objektorientierten Programmierung sei auf [1] verwiesen.

Einleitung

Die folgende Stichpunktliste versucht, einige Argumente aufzuzählen, die bei der Entscheidung für oder gegen RDF eine Rolle spielen können. Die Argumente sind dabei ansatzweise nach ihrer technischen Relevanz sortiert, beginnend mit dem schwerwiegendsten Argument.

Existiert ein Branchenstandard?

Branchenstandards

Vor allem im Bereich des E-Commerce (vorwiegend B2B¹²) sind XML-Standards verfügbar, die Einigungen einzelner Branchenteilnehmer auf ein einheitliches Vokabular darstellen (auch XML-Branchenstandards genannt). Als eines von vielen Beispielen sei hier FundsXML als Standard des Bundesverbands Investment und Asset Management e.V. (BVI)¹³ genannt. Bleibt man in seiner Tätigkeit auf die entsprechenden Teilnehmer begrenzt, ist sicherlich die Einigung auf einen Standard eine Lösung. Jedoch stellt dies eine Insellösung dar. Eine Transformation in einen verwandten Standard muss jeweils aufwendig erstellt werden. Ist ein Branchenstandard verfügbar, gleich ob via XML oder RDF, ist die Nutzung des existierenden Standards in der Regel die beste Wahl.

Liegen die zu modellierenden Daten in Aussageform vor?

RDF modelliert Daten in so genannten „Aussagen“ bzw. „Tripeln“, vergleichbar zu natürlichsprachlichen Aussagen bestehend aus Subjekt, Prädikat und Objekt. Letztlich lassen sich alle Daten durch Umwandlung in diese Form bringen bzw. „zwingen“, und letztlich lassen sich Aussagen auch ohne RDF und z.B. via XML repräsentieren. Daten, die von vornherein in Aussagenform oder Tripelform vorliegen, sind aber besonders geeignet für die Modellierung durch RDF.

¹² Business-to-Business

¹³ <http://www.bvi.de>

Ein eng verwandter Vorteil von RDF gegenüber XML sind die so genannten „vergegenständlichten Aussagen“ („Reified Statements“). Sie ermöglichen, vereinfacht gesagt, dass jede Aussage eines RDF-Dokuments selbst wieder zu einem Objekt wird, über das Aussagen gemacht werden können. XML unterstützt dieses Konzept nicht.

Reified Statements

Müssen ganze Dokumente oder einzelne Aussagen validiert werden?

Validierung ist die Prüfung von Daten gegen ein vorgegebenes Regelwerk. Sowohl RDF als auch XML unterstützen Validierung, in der Regel durch [RDF Schema] bzw. [XML Schema] oder [DTD]. Allerdings unterscheiden diese Ansätze sich erheblich: RDF Schema definiert Gültigkeitsregeln für einzelne Aussagen, wohingegen XML Schema Gültigkeitsregeln für vollständige Dokumente definiert. Verbildlicht regelt [XML Schema] das Erscheinungsbild eines gesamten XML-Dokuments, wie die Dokumentvorlage eines Verlagshauses das Erscheinungsbild einer Veröffentlichung regelt. RDF Schema hingegen regelt die Gültigkeit einzelner Aussagen, wie die Regeln einer menschlichen Sprache die Gültigkeit gesprochener oder geschriebener Aussagen regeln – ohne dabei die Struktur ganzer Dokumente vorzuschreiben. Müssen die zu modellierenden Daten vor allem Aussage für Aussage und nicht jeweils für ein ganzes Dokument validiert werden, kann dies ein Argument für den Einsatz von RDF sein.

RDF Schema vs. XML Schema

Existiert die Bereitschaft, (teilweise kommerzielle) Werkzeuge einzuführen und zu verwenden?

Nicht zu unterschätzen ist in der Praxis die Verfügbarkeit von Tools oder ergänzenden Standards. Sowohl für XML als auch für RDF gibt es inzwischen zahlreiche Tools. Pauschal darf man sicherlich feststellen, dass es für XML mehr und vor allem ausgereifere Werkzeuge als für RDF gibt. Trotzdem muss in dem konkreten Szenario abgewogen werden, welche Tools für XML oder RDF zur Verfügung stehen und evtl. bereits lizenziert bzw. erworben worden sind.

Nutzung (kommerzieller) Werkzeuge

Einleitung

So gibt es z.B. zahlreiche relationale Datenbankserver, die XML-Daten auf die eine oder andere Weise direkt verarbeiten können –für RDF gibt es diese Unterstützung (noch?) nicht. Werden voraussichtlich ohnehin keine Werkzeuge erworben und eingeführt, ist dieser Aspekt zu vernachlässigen.

Werden standardisierte Parser benötigt?

DOM, SAX

Für XML existieren Parserstandards wie [DOM] und [SAX], die teilweise plattformunabhängig sind. Vor allem DOM¹⁴ ist inzwischen auf zahlreichen Plattformen verfügbar, unter Java genauso wie unter .NET oder Javascript/DHTML¹⁵. Für RDF existieren derartige etablierte Parserstandards nicht. Existiert in einem konkreten Anwendungsfall bereits Entwickler-Know-how für einen dieser Parserstandards, kann dies ein Argument für die Nutzung von XML sein. Dasselbe gilt, wenn es darum geht, den Parser austauschbar zu gestalten. Durch die Nutzung von XML und DOM oder SAX¹⁶ kann ein Parser (fast) problemlos gegen einen anderen Parser ausgetauscht werden. Für RDF gibt es diese Möglichkeit nicht.

Dreht es sich um Metadaten über Internetressourcen?

Metadaten über Webressourcen

RDF ist besonders geeignet, um Metadaten über Internet-Ressourcen zu modellieren, da die Verwendung von URIs oder URLs¹⁷ ein elementarer Bestandteil ist. Selbstverständlich können auch in XML-Dokumenten URIs/URLs verwendet werden, dies muss aber vom Autor individuell ausdefiniert werden. RDF trägt dieses Konzept in sich und ist daher besonders geeignet für derlei Szenarien, siehe „Metadaten für Annotationen und Anreicherungen“.

¹⁴ Document Object Model

¹⁵ Dynamic HTML

¹⁶ Simple API for XML

¹⁷ Uniform Resource Locator

Spielt Datenvisualisierung eine tragende Rolle?

RDF-Daten lassen sich durch den RDF-Graphen einfach, standardisiert und intuitiv visualisieren. Für XML-Dokumente gibt es kein standardisiertes Verfahren für eine Visualisierung. Selbstverständlich lassen sich auch XML-Daten visualisieren. Trotzdem sind Daten, die in erster Linie visualisiert werden müssen und evtl. darüber hinaus noch in Aussageform vorliegen (siehe oben), ein starkes Argument für den Einsatz von RDF.

RDF-Graphen

Haben die zu modellierenden Daten einen verteilten Charakter?

RDF besitzt per se eine dezentrale, verteilte Struktur. Dies äußert sich z.B. darin, dass mittels [RDF Schema] nicht verhindert werden kann, dass einem RDF-Dokument Aussagen hinzugefügt werden – das Dokument ist und bleibt ein gültiges RDF-Dokument. Die dezentrale Struktur von RDF äußert sich auch darin, dass Aussagen in einem Dokument dieselbe Aussagekraft besitzen wie Aussagen in jedem anderen Dokument. RDF ist also per se verteilt. [XML Schema] hingegen legt XML-Dokumenten enge Regeln auf. XML an sich ermöglicht (nahezu) beliebige Dokumente, konkrete XML-Dokumente hingegen sind aber nicht zwingend erweiterbar. Dies ist anders mit RDF.

RDF ist dezentral

Wird Export in ein Textformat benötigt?

RDF-Daten lassen sich durch die Serialisierung in N3 in eine besonders einfache, lineare Struktur exportieren, die der von CSV ähnelt. Genau wie CSV ist N3 nicht mit Metadaten ausgestattet. Mit XML-Dokumenten ist das nicht in jedem Fall möglich, und es gibt insbesondere kein definiertes Verfahren dafür.

RDF nach N3/CSV

Einleitung

2/9.2 Anwendungsarten

Anwendungen des Semantic Web lassen sich einer oder mehreren der folgenden Anwendungsarten zuordnen. Diese Aufzählung wurde inspiriert von [2] und um eigene Ideen erweitert. Sie erhebt keinen Anspruch auf Vollständigkeit, vermittelt aber einen guten Überblick über die Bandbreite der Semantic-Web-Anwendungen.

2/9.2.1 Integration über Daten/ „Datenintegration“ (Data Integration)

Integration über Daten nehmen zwei oder mehr Anwendungen vor, wenn Daten gegenseitig ausgetauscht oder gemeinsam genutzt werden. Dabei kann es sich z.B. um eine Abfrage über mehrere Datenquellen handeln wie bei einem Data Warehouse. Von einer „Integration über Daten“ spricht man aber nur dann, wenn der Datenaustausch erheblichen Mehrwert liefert und den Daten das Hauptaugenmerk gilt. Verschiedene Anwendungen benutzen oft verschiedene Datenmodelle, die für eine Integration der Daten aufeinander abgebildet werden müssen.

Integration über Daten und Data Warehouses

Integration über Daten kann durch RDF ermöglicht werden, das dann als „Ontology Middleware“ die (semi-)automatische Integration erleichtert. Der Datenaustausch kann über [XML] oder [N3] erfolgen. Die Datenspeicherung kann entweder ebenfalls in XML oder N3 in einfachen Textdateien erfolgen oder in einer relationalen Datenbank, XML-Datenbank oder speziellen RDF-Datenbank. RDF-Daten lassen sich aufgrund der einfachen Struktur sehr leicht in einer relationalen Datenbank ablegen. Allerdings ist der einfachste Weg, nämlich eine Tabelle mit drei Spalten, in der Regel nicht sinnvoll. Sinnvoller ist ein speziell auf den Anwendungszweck zugeschnittenes Datenbankschema oder eine spezielle RDF-Datenbank.

Anwendungsarten

Parsen von RDF

RDF-Daten lassen sich leicht einlesen. N3-serialisierte RDF-Daten besitzen ein Textformat und können zeilenweise eingelesen und zerlegt werden. XML-serialisierte RDF-Daten lassen sich mit einem XML-Parser einlesen (z.B. [DOM] oder [SAX]). Beide Vorgehensweisen sind einfach realisierbar, ermöglichen aber keine Validierung gegen ein RDF-Schema. Für Validierung muss ein RDF-Parser verwendet werden. Der Einsatz eines speziellen RDF-Parsers ist auch deswegen sinnvoll, weil dieselben RDF-Daten sich auf verschiedene Weisen serialisieren lassen. Werden XML-serialisierte RDF-Daten ohne RDF-Parser eingelesen, muss beim Einlesen damit gerechnet werden, dass es verschiedene Arten und Weisen („Schreibweisen“) für dieselben RDF-Daten gibt.

**2/9.2.1.1 Datenabhängige Agenten
(Data-dependent Agents)**

Autonome Agenten

Datenabhängige Agenten sind Anwendungen, die Informationen sammeln, anhand eines Regelwerks „Entscheidungen“ treffen und diese einem Anwender oder einer anderen Software präsentieren. Technisch betrachtet besteht Ähnlichkeit zur Integration über Daten. Aus Anwendersicht stehen aber nicht die Daten im Vordergrund, sondern die „autonome“ Agenten-Software, die Entscheidungen trifft und dem Anwender dadurch echten Mehrwert bietet.

**2/9.2.1.2 Wissensmanagement
(Knowledge Management)**

Kollaboration

Wissensmanagement hat das automatische Finden von Lösungen für konkrete Problemstellungen zum Ziel, und ist ein gut erforschtes Forschungsgebiet und ein „umkämpftes“ Marktsegment zugleich. Anwendungen sind z.B. Unternehmensanwendungen zur verteilten Zusammenarbeit („Kollaboration“ bzw. „Collaboration“) oder automatische Wissensableitung in Internetcommunities.

Techniken des Semantic Web können eingesetzt werden, um Inhalte von Dokumenten, E-Mails und Notizen zu klassifizieren und gegen ein gegebenes Problem zu vergleichen.

2/9.2.1.3 Semantische Indizes und Portale (Semantic Indexing and Portals)

Semantische Indizes und Semantische Portale sind eine viel versprechende Zukunftsperspektive für heutige Internet-Suchmaschinen, -Verzeichnisse und -Portale wie [Yahoo], [DMOZ] und [Google]. Internet-Suchmaschinen sammeln heutzutage (nahezu) wahllos Informationen, ohne sie zu kategorisieren, wohingegen Internetverzeichnisse zwar kategorisieren, aber auf redaktionelle Arbeit angewiesen sind. Das Semantic Web bietet diesen Diensten die Möglichkeit, durch maschinenlesbare, eindeutige URIs („Uniform Resource Identifier“) Informationen an natürlichsprachigen Hürden vorbei automatisch zu klassifizieren, zu bewerten und selbstständig ins Portal zu integrieren. Diese „besseren“ Daten ermöglichen es Portalen zukünftig, Anwendern präzisere Suchmöglichkeiten an die Hand zu geben. [Google] merkt sich heute (via Cookie), welche Landessprache ein Anwender in der Vergangenheit bevorzugt hat –in Zukunft vielleicht, ob ein nach „Chips“ suchender Anwender „Kartoffelchips“ oder „Computerchips“ sucht (*Ranking*).

Suchmaschinen und Verzeichnisse

2/9.2.1.4 Verwaltung persönlicher Daten (Personal Information Management)

Verwaltung persönlicher Daten ist auch unter dem ursprünglichen Begriff „Personal Information Management“ oder kurz „PIM“ bekannt und dreht sich um die Verwaltung persönlicher Informationen wie Anschriften und Aufgaben. Der Begriff „PIM“ bezeichnet sowohl Softwarelösungen wie Microsoft Outlook und Lotus Notes als auch Hardware wie Palm Computer oder Pocket PCs.

Personal Information Management

Anwendungsarten

Auch viele Anwendungen wie der Webbrowser [FireFox]/[Mozilla] fallen in diesen Bereich, da sie ebenfalls persönliche Daten verwalten: Lesezeichen/Bookmarks/Favoriten, eine Historie und Präferenzen. Im Gegensatz zum Wissensmanagement wird PIM von Anwendern zur Verwaltung eigener Daten eingesetzt, wohingegen Wissensmanagementsoftware Wissen einer Gruppe oder eines ganzen Unternehmens verwaltet. Einige PIM-Systeme setzen bereits RDF ein, da sein dezentraler und semistrukturierter Charakter sich besonders gut dafür eignet, unterschiedlich strukturierte Informationen wie Kontakte, Aufgaben und Termine einheitlich zu verwalten. PIM-Systeme dieser Art sind keine Semantic-Web-Anwendungen, da die Semantic-Web-Techniken tief im Inneren verborgen sind und allenfalls für Softwareentwickler zum Vorschein kommen. Zudem sind die Daten der PIM-Systeme privater Natur und sollen nicht der Allgemeinheit zur Verfügung gestellt werden. Dennoch nutzen PIM-Techniken das Semantic Web wie z.B. RDF und tragen daher auch zu dessen Verbreitung und Akzeptanz bei.

2/9.2.1.5 Metadaten für Annotationen und Anreicherungen (Metadata for Annotating and Enriching)

„Jeder kann alles über alles sagen“

Metadaten für Annotationen und Anreicherungen sind Kommentare, Meinungen und Bewertungen, die von der Leserschaft an eine Webseite, eine Grafik oder ein Dokument „geheftet“ worden sind. Diese Metadaten ergänzen oder diskutieren den Inhalt des Dokuments und können von den Lesern des Dokuments gesehen werden. Anwendungen dieser Art entsprechen einer grundlegenden Idee des Semantic Web: *Jeder kann alles über alles sagen*. Von einer anderen Idee des Semantic Web machen sie hingegen keinen Gebrauch: Diese Annotationen werden von Menschen für Menschen angeheftet, beinhalten für Maschinen nur geringen semantischen Informationsgehalt und werden (so gut wie) nicht von Maschinen ausgewertet.

RDF in Kombination mit HTML ermöglicht es, in Webseiten zusätzliche Informationen zu hinterlegen. Diese zusätzlichen Informationen sind nicht für Menschen lesbar, da sie zwar genau wie die visuellen Informationen in der Seite enthalten sind, aber im Gegensatz zu diesen nicht im so genannten *Body* (Textkörper), sondern im *Head*. Diese zusätzlichen Informationen können entweder doppelt sowohl in HTML als auch in RDF auf der Website untergebracht sein oder auch ausschließlich in RDF – in diesem Fall kann eine „Maschine“ der Seite mehr oder zumindest andere Informationen entnehmen als ein Mensch, der auf die HTML-Darstellung angewiesen ist. Im Umkehrzug kann aber eine Maschine die HTML-Informationen nicht (bzw. nur geringfügig) verarbeiten, die für den Menschen veröffentlicht sind.

HTML + RDF

2/9.2.1.6 Metadaten für Beschreibung, Suche und Auswahl (Metadata for Description, Discovery and Selection)

Metadaten für Beschreibung, Suche und Auswahl helfen Anwendern, Informationen zu finden. Sie helfen dabei, anhand eines Suchbegriffs passende Produkte oder Dienste zu finden, die Ergebnisliste zu filtern oder ein gefundenes Ergebnis aussagekräftig zu beschreiben. Diese Metadaten sind z.B. redaktionell vorgenommene Kategoriezuweisungen oder Verknüpfungen. Kurzum: sie ermöglichen eine bessere Suche als die zurzeit übliche Volltextsuche. Im Unterschied zu den Metadaten für Annotationen werden diese Metadaten von Maschinen genutzt. Viele Anwendungen lassen sich nicht zweifelsfrei einer der beiden Kategorien zuordnen, und auch die Abgrenzung zu den Semantischen Indizes/Semantischen Portalen ist nicht immer eindeutig.

Metadaten zur Suche

Für die Suche nach RDF-Daten gibt es spezielle Abfragesprachen wie RDQL (RDF Data Query Language), RQL (RDF Query Language), SeRQL (Sesame RDF Query Language), SPARQL (Simple Protocol and RDF Query Language) und eRQL (easy RDF Query Language, siehe oben).

Anwendungsarten

Ein De-facto-Standard mit ähnlicher Bedeutung wie SQL (Structured Query Language) für relationale Datenbanken hat sich für RDF bislang nicht etabliert.

2/9.2.1.7 Metadaten für Medien und Inhalte (Metadata for Media and Content)

Metadaten für Medien und Content

Metadaten für Medien und Inhalte helfen bei der Verwaltung von Medienobjekten wie Bildern, Videos und Animationssequenzen. Man spricht von *Media Management* oder *Asset Management*, oft verbunden mit *Content Management*. Im Gegensatz zu den Metadaten für Annotationen und Anreicherungen werden diese Metadaten nicht von Anwendern angeheftet, sondern von Redakteuren oder automatisch (z.B. Datum der letzten Änderung). Im Gegensatz zu den Metadaten für Beschreibung, Suche und Auswahl wiederum sind diese Metadaten oftmals nicht für den Anwender sichtbar. Sie werden vielmehr während der Erzeugung, Verwaltung oder Archivierung der Medien oder Inhalte angezeigt und/oder aktualisiert, teilweise automatisch und teilweise per Hand. Wie bei der Verwaltung persönlicher Daten bleiben die Metadaten auch hierbei innerhalb der Anwendung versteckt. Diese Metadaten demonstrieren daher nicht sonderlich gut das Wesen des Semantic Web, tragen aber dennoch maßgeblich zu seiner Weiterentwicklung und Akzeptanz bei.

Dublin Core (DC), hCalendar, Adobe XMP

Derartige Informationen haben unbemerkt Einzug in Desktop-Anwendungen und das Web gehalten und nutzen z.B. Vokabularien wie [Dublin Core] oder [hCalendar]. Beispielsweise hat der Softwarehersteller Adobe RDF als Speicherformat für PDF-Dokument-Metadaten entdeckt (Autor, Titel, Änderungsdatum, .). und dafür eine eigene Schnittstelle [XMP] entwickelt, die z.B. in der Software Adobe Acrobat eingesetzt wird.

2/9.2.1.8 Wissensgenerierung (Knowledge Formation)

Wissensgenerierung aus bereits vorhandenem Wissen ist konsequenterweise der nächste Schritt nach der reinen Verwaltung von Wissen, wie sie z.B. bei den Metadaten-Anwendungen eingesetzt wird. Maschinen können gegenwärtig nicht in menschlichem Sinne „denken“, aber sie können Schlussfolgerungen ziehen.

In einem Netzwerk sozialer Kontakte wie z.B. [openBC] treten Menschen über die Internetplattform zu anderen Menschen in Kontakt, die sie kennen. Aus diesen Angaben in der Form „Person A kennt Person B“ und „Person B kennst Person C“ schlussfolgert das System, dass Person A die Person C „um eine Ecke“ oder mit gewisser Wahrscheinlich auch unmittelbar kennt, und schlägt Person A vor, Kontakt zu Person C aufzunehmen.

2/9.2.1.9 Management von Katalogen & Thesauri (Catalogue & Thesaurus Management)

Management von Katalogen und Thesauri wird in digitalen Bibliotheken benötigt, in Webverzeichnissen und auf B2B-Marktplätzen; im Grunde genommen immer dann, wenn eine gemeinsame Sprache benötigt wird (*Vokabular*). In einer Bibliothek suchen Anwender Veröffentlichungen, in einem Webverzeichnis Hyperlinks und auf B2B-Marktplätzen nach Dienstleistungen einer bestimmten Art. Bislang werden zu diesem Zweck oftmals simple Kategorielisten verwendet, teilweise linear und teilweise hierarchisch (Baumstruktur). Diese Kategorisierungen stoßen Tag für Tag an ihre Grenzen, wenn sich die Realität weiterentwickelt und neue Kategorien hinzukommen, alte Kategorien verschwinden, sich Bedeutungen verändern und Synonyme hinzukommen.

**Vokabulare für
B2B-Marktplätze**

Anwendungsarten

Jeder, der einmal unter „Frisör“ nach der Nummer seines Haarschneiders gesucht hat, um nach stundenlanger Suche festzustellen, dass dieser unter „Coiffeur“ eingetragen ist, kennt diese Problematik. Das Semantic Web bietet eine natürliche Möglichkeit, diese starren Kategorielisten durch flexible und mitwachsende Vokabulare zu ersetzen.

2/9.2.1.10 „Syndikation“ (Syndication)

RSS, Syndikation

Syndikation bezeichnet die Weitergabe von (lizenzierten) Inhalten an Empfänger, die diese Inhalte innerhalb definierter Nutzungsbedingungen weiterverwenden.

Oft wird dieser Begriff im Zusammenhang mit [RSS] verwendet (*Really Simple Syndication*). RSS ist eine Technik, die es interessierten Besuchern einer Website erlaubt, mittels eines so genannten *RSS Reader* über Änderungen an der Website, aktuelle Nachrichten, neue Forenbeiträge und andere Informationen auf dem Laufenden gehalten zu werden, ohne die Website besuchen zu müssen. Bei den betroffenen Websites kann es sich um Internetportale, -communities, Nachrichtenseiten oder [Blogs] handeln, kurzum um jede Webseite, die veränderliche Inhalte aufweist.

Syndikation hat wenig zu tun mit der Annotierung von Inhalten oder der Verwaltung von Metadaten, sondern dreht sich ausschließlich um die Verteilung und Zusammenstellung von Inhalten, im Fall von RSS oftmals durch RDF transportiert (je nach RSS-Version). Der RSS-Reader des Anwenders, auch als „News Aggregator“ bezeichnet, wertet die empfangenen Nachrichten nicht aus, aber stellt sie zusammen (*aggregiert* sie). Beispielsweise kann ein RSS Reader Nachrichten abonnisierter Websites in einer gemeinsamen, chronologischen Auflistung anzeigen – insofern handelt es sich tatsächlich um eine maschinelle Auswertung der Nachrichten, wie sie mit dem sichtbaren Web allein nicht möglich ist.

2/9.3 Semantic-Web-Programme

Für den Umgang mit und den Aufbau des Semantic Web gibt es verschiedene Funktionen, die durch Programme unterstützt werden müssen. Zu den wichtigsten zählen sicherlich:

Funktionen

- Erzeugung, Bearbeitung und Visualisierung
- Speicherung (Persistenz)
- Anfragen
- Inferenz

Für diese und weitere Funktionen steht mittlerweile eine Vielzahl an Programmen zur Verfügung. Unter dem Portal semanticweb.org werden insgesamt 267 Programme aufgelistet¹. Dort kann über eine Anfragemaske nach bestimmten Programmen gesucht werden. Auch zu empfehlen sind die Seite von Dave Beckett² und die RDF-Seite des W3C³. Dave Becketts Seite ist sehr umfangreich und ordnet Programme zeitlich entsprechend ihrer Ankündigung. Die W3C-Seite bietet nur eine kleine Übersicht und ordnet die Programme nach der verwendeten Programmiersprache. Viele der Programme stammen aus Forschungsprojekten oder wissenschaftlichen Arbeiten. Oft ist dabei ein weit reichender Support nicht gewährleistet. Dafür sind diese Programme allgemein frei verfügbar und werden zusammen mit dem Quellcode (Open Source) angeboten. Einige der kommerziellen Produkte entstammen historisch solchen Forschungsprojekten und wissenschaftlichen Arbeiten.

Im Folgenden gehen wir auf einige der SW-Programme aufgeteilt nach den genannten Funktionalitäten näher ein.

¹ Online unter: <http://www.semanticweb.org/software> – Stand vom 29. März 2006

² Online unter: <http://planetrdf.com/guide/> – Dave Beckett's Resource Description Framework (RDF) Resource Guide

³ Online unter: <http://www.w3.org/RDF/> – die RDF-Seite des World Wide Web Consortiums

Semantic-Web-Programme

2/9.3.1 Erzeugung, Bearbeitung und Visualisierung

Ebenen

Dies sind wohl die fundamentalsten Funktionalitäten, die unterstützt werden müssen. Dabei kann in zwei Ebenen unterschieden werden:

- Datenebene – in der die eigentlichen Aussagen zu finden sind, wie: „Der Ball ist rund!“
- Schema- bzw. Ontologieebene – in der das Vokabular beschrieben wird, das in der Datenebene zur Beschreibung verwendet werden kann. So kann dort der Ball als ein Spielzeug definiert sein und die semantische Definition von „rund“ hinterlegt werden.

Der deutlich komplexere Teil dabei ist die Ontologieebene, wobei nicht jedes Programm alle Ausdrucksmöglichkeiten unterstützt, die OWL anbietet. Auch ist zu beachten, dass die Ontologien ebenfalls durch RDF-Dateien beschrieben werden. Programme zur Erzeugung, Bearbeitung oder Visualisierung von Ontologien unterstützen daher in der Regel auch die Datenebene.

Protégé

Protégé von der Stanford University ist Open Source und frei verfügbar. Das Produkt ist sehr verbreitet und recht ausgereift. Es gibt zahlreiche Plug-ins, um Protégé an gegebene Bedürfnisse anzupassen. Neben der Unterstützung von RDF Schema und OWL werden auch Ontologien unterstützt, die kompatibel zum OKBC-Protokoll (*Open Knowledge Base Connectivity*) sind. Insbesondere bei sehr großen Ontologien ist Protégé zu empfehlen. Der Fokus des Programms liegt auf dem Erstellen und Bearbeiten von Ontologien.

IsaViz

IsaViz wird vom W3C gehostet. Es ist eine Java-Open-Source-Entwicklung und frei verfügbar. Die erste Veröffentlichung von IsaViz war 2002. Die zurzeit stabile Version 2.1 stammt vom Oktober 2004 und kann als ausgereift bezeichnet werden. Die Darstellung von größeren Datenmengen und das Navigieren in ihnen sind problemlos möglich. Der Fokus des Programms liegt auf der Visualisierung, es ist für kleinere Änderungen oder kleine Projekte auch zum Erstellen und Bearbeiten ausreichend. Es wird an der Nachfolgeversion 3.0 gearbeitet, die als Alpha-Version zur Verfügung steht.

SemanticWorks

Relativ neues kommerzielles Programm von Altova (bekannt durch XMLSpy). Eine 30-Tage-Testversion kann kostenlos heruntergeladen werden. SemanticWorks ist ein grafischer Editor für RDF, RDF Schema und OWL. Die grafische Benutzeroberfläche macht einen sehr guten Eindruck. Inwieweit auch sehr große Ontologien mit SemanticWorks bearbeitbar sind, haben wir nicht getestet.

SemanticWorks
(Altova XMLSpy)

InferEd

InferEd befindet sich in der Beta-Phase und kann zurzeit kostenlos bezogen werden. Künftige Versionen sind für den nicht kommerziellen Einsatz und die Lehre weiterhin kostenlos geplant. Das Programm wird von Intellidimension entwickelt und kann auf deren Produkt RDF Gateway aufbauen (siehe unter „Speicherung“). Als Besonderheit können abgeleitete Aussagen angezeigt werden, wobei die Aussagen, die zu der neuen abgeleiteten Aussage geführt haben, mit angezeigt werden können. Die Ableitung wird dadurch nachvollziehbar bzw. bewiesen. Die Visualisierung steht bei InferEd nicht im Vordergrund und beschränkt sich auf Navigationshilfen innerhalb der Ontologien. Dabei werden RDF Schema und OWL unterstützt.

Semantic-Web-Programme

SMORE

Der *Semantic Markup, Ontology and RDF Editor*, kurz SMORE, wurde an der University of Maryland, College Park, unter Leitung von Professor James Hendler entwickelt. Ziel des Editors ist die gleichzeitige Bearbeitung und Erstellung von Webseiten zusammen mit entsprechenden Annotationen in RDF, RDF Schema oder OWL. Dies soll das lästige Springen zwischen entsprechenden Programmen vermeiden. Dabei zählen auch weniger erfahrene Benutzer zur Zielgruppe. Was die Erstellung von Ontologien angeht, werden nicht alle Feinheiten von OWL unterstützt. Das Programm ist sehr einfach auszuprobieren und komplett als GUI⁴-Anwendung in Java implementiert. Der Start des Programms erfolgt über einen einfachen Java-Aufruf. SMORE ist frei verfügbar unter der GNU General Public License (GNU GPL).

OntoStudio

F-Logic

OntoStudio stammt von der deutschen Firma Ontoprise GmbH. Die Firma entstand als Spin-off der Universität Karlsruhe. Die Vorgängerversion von OntoStudio ist unter dem Namen OntoEdit bekannt. OntoStudio ist modular aufgebaut, in Java entwickelt und basiert auf Eclipse. Neben dem Erstellen und Bearbeiten von Ontologien ist besonders die angebotene Visualisierung gut gelungen. Unterstützt werden RDF Schema, OWL und *F-Logic*⁵. Das Programm kann um zusätzliche Plug-ins erweitert werden. Neben OntoStudio bietet Ontoprise weitere Produkte wie den OntoBroker an, der weiter unten behandelt wird.

⁴ Graphical User Interface

⁵ F-Logic ist eine Ontologiesprache, die Frames (daher das F) mit Prädikatenlogik verbindet.

Produkt	Quellcode	Bemerkung	Lizenz/ Kosten	Betriebssystem
Protégé	Java	gut für große Ontologien	Open Source / free	unabhängig
IsaViz	Java	Fokus liegt auf der Visualisierung	GPL-kompatibel	unabhängig
Semantic-Works	—	von Altova (XMLSpy)	30-Tage-Testversion	Windows NT, XP, 2000, 03
InferEd	—	bietet die Herleitung abgeleiteter Aussagen	zurzeit Beta-Phase (kostenlos)	Windows 2000, XP, NT
SMORE	Java	für Einsteiger, jedoch mit begrenzter Funktionalität	GPL	unabhängig
OntoStudio	—	sehr gute Visualisierung	als Hauptspeicherversion kostenlos erhältlich	unabhängig

2/9.3.2 Speicherung

Das Semantic Web stellt eine Erweiterung des Internets dar, wobei die Daten als RDF/XML-Dateien oder als Zusätze zu den vorhandenen Daten abgelegt werden. Für Anwendungen, die Ergebnisse in kurzer Zeit liefern sollen, z.B. bei Informationsportalen, wäre ein direkter Zugriff auf diese verteilten Daten zu langwierig. Auch ist das RDF/XML-Format für den Datenaustausch konzipiert, nicht für den schnellen Zugriff. Ferner ist für manche Anwendungen eine permanente Zugriffsmöglichkeit wichtig. Dies kann in einem verteilten System nicht immer gewährleistet werden. Insgesamt bleibt festzuhalten, dass die Möglichkeit zur Speicherung der RDF-Daten in Datenbanken zwingend notwendig ist.

Semantic-Web-Programme

Dabei spielt das verwendete Datenmodell, also wie die Daten in einer Datensenke abgebildet werden, eine entscheidende Rolle. Es hat Auswirkungen auf sämtliche Anwendungen, die auf der Datensenke aufbauen.

Jena

Generische Repräsentation

Jena ist frei verfügbar, Open Source und in Java entwickelt. Es stellt ein Framework zur Speicherung, Verwaltung und Abfrage dar. Die Abfrage erfolgt mittels RDQL und neuerdings SPARQL. Auf beide wird im folgenden Abschnitt zu Anfragesprachen näher eingegangen. Die erste Version des von Hewlett-Packard (HP) entwickelten Produkts wurde im Jahr 2000 veröffentlicht. Zur persistenten Speicherung baut Jena auf einem relationalen Datenbankmodell auf. Dieses Datenmodell basiert auf Quadrupeln, die (etwas vereinfacht) flach in einer Tabelle abgespeichert werden. Man spricht dabei von der *generischen Repräsentation* (auch *monolithischer Ansatz* genannt). Zusätzlich zu den Tripeln der RDF-Aussagen kann dabei ein Modell-Knoten abgespeichert werden. Die genaue semantische Definition dieses Modell-Knotens wird nicht festgelegt. Jena unterstützt zurzeit die relationalen Datenbankmanagementsysteme (Relational Database Management System, RDBMS) MySQL, Oracle und PostgreSQL. Neben der persistenten Speicherung kann Jena auch als In-Memory-Version betrieben werden.

RDFSuite

Schemaspezifische Repräsentation

RDFSuite wurde am ICS-FORTH in Griechenland entwickelt. Es ist frei verfügbar und Open Source. Der enthaltene *Validating RDF Parser* (VRP) und der *RDF Schema Specific Database Loader* (RSSDB) sind in Java entwickelt. Als Anfragesprache wird RQL unterstützt. Der entsprechende RQL-Interpreter ist in C implementiert. Im Datenbankmodell werden die objektorientierten Eigenschaften von RDF Schema abgebildet, d.h. die verschiedenen RDF-Schema-Konstrukte wie Klassen, Eigenschaften, Container etc. werden gesondert abgebildet. Das gleiche gilt für die Klassen- und Eigenschaftshierarchien.

Man spricht dabei auch von der *schemaspezifischen Repräsentation*. Damit dies entsprechend gespeichert werden kann, wird das objektrelationale DBMS PostgreSQL verwendet und dessen objektorientierten Eigenschaften ausgenutzt. Andere Datenbanken werden daher nicht unterstützt.

RDF-S3

Das RDF-Source related Storage System (RDF-S3) ist frei verfügbar, Open Source und in Java entwickelt. Das zugrunde liegende Datenbankmodell ist eine Synthese der generischen und der schemaspezifischen Repräsentation. Trotzdem ist das Datenmodell als rein relationales Datenmodell aufgebaut, sodass es mit allen RDBMS über JDBC (Java Database Connectivity) laufen sollte. Die Lauffähigkeit wurde explizit mit MySQL und DB2 von IBM getestet. Empfohlen wird die Nutzung von DB2, da hierfür einige Optimierungen integriert sind. Aussagen werden in RDF-S3 jeweils zusammen mit ihrer Quellinformation gespeichert. Das Datenmodell basiert also auf Quadrupeln, wobei der zusätzliche Knoten zu den RDF-Aussagen semantisch als Quellinformation vordefiniert ist. Dies ermöglicht es, in RDF-S3 gespeicherte Quellen wieder zu löschen oder zu aktualisieren. RDF-S3 ist sehr einfach über grafische Benutzeroberflächen zu bedienen. Unterstützt wird die Anfragesprache eRQL. Vom Reifegrad ist RDF-S3 noch nicht so weit wie die anderen vorgestellten Produkte. Insbesondere an zusätzlichen Schnittstellen und der Performance muss noch gearbeitet werden.

Sesame

Sesame von ADUNA wurde früher von Administrator (Niederlande⁶) als Teil eines EU-Projekts entwickelt. Es stellt eine Open-Source- und webbasierte Java-Architektur dar und kann über HTTP bzw. SOAP⁷ angesprochen werden. Das verwendete Datenbankmodell entspricht der generischen Repräsentation, jedoch werden Klassen- und Eigenschafts-

⁶ <http://aduna.biz/index.html>

⁷ Simple Object Access Protocol

Semantic-Web-Programme

hierarchien gesondert gespeichert. Als Datenbank wird PostgreSQL verwendet. Als Anfragesprachen unterstützt Sesame RDQL und RQL. Für Sesame gibt es mittlerweile eine SPARQL Java API.

RDF Gateway

Deduktive Datenbank

RDF Gateway ist ein Produkt von Intellidimension⁸. Es werden verschiedene Editionen angeboten, wobei die Nutzung der nicht kommerziellen Personal Edition kostenfrei ist. RDF Gateway stellt einen Applikations-, Datenbank- und Webserver dar. Es ist als Service für Microsoft Windows (NT, XP und 2000) implementiert. Intern baut RDF Gateway auf einer mitgelieferten *deduktiven Datenbank* auf. In deduktiven Datenbanken werden Fakten, Integritätsbedingungen und Ableitungsregeln gespeichert (im Allgemeinen basierend auf einem RDBMS). Bei Anfragen werden nicht nur die reinen Fakten, sondern auch die über die Bedingungen und Regeln ableitbaren Aussagen berücksichtigt (siehe auch später unter Inferenz). In andere Anwendungen kann über Microsoft ADO oder JDBC integriert werden. Als Anfragesprache unterstützt RDF Gateway die eigene Anfragesprache RDFQL.

Redland

Redland ist frei verfügbar, Open Source und in C entwickelt. Es wird von Dave Beckett von der Universität in Bristol gepflegt. Die Speicherung erfolgt In-Memory oder in einem RDBMS: Sleepycat/Berkeley DB, MySQL, AKT Triplestore oder SQLite. Durch das *Redland Binding Package* kann das Produkt durch verschiedene Programmiersprachen angesprochen werden, wie C#, Java, Perl, PHP und Python. Als Anfragesprachen werden RDQL und SPARQL unterstützt.

⁸ <http://www.intellidimension.com>

Kowari

Kowari wurde von Tucana Technologies Inc. entwickelt. Es ist frei verfügbar, Open Source und in Java geschrieben. Die erste Veröffentlichung war im Herbst 2003. Kowari baut nicht auf einer externen Datenbank auf, sondern enthält eine eigenentwickelte Datenbank zum Speichern der Aussagen zusammen mit einem Meta-Knoten. Es können also Quadrupel und nicht nur Tripel gespeichert werden. Zur Bearbeitung von Anfragen baut Kowari auf der interactive Tucana Query Language (iTQL) auf. Kowari verfügt zusätzlich über mehrere Schnittstellen, wodurch auch eingeschränkt RDQL bzw. SPARQL unterstützt werden kann.

Eigenentwickelte Datenbank

Produkt	Quellcode	Datenbanken	Anfragesprachen	Betriebssystemen
Jena	Java	MySQL, Oracle, PostgreSQL	SPARQL, RDQL	unabhängig
RDFSuite	Java/C	PostgreSQL – obj.-relational	RQL	Solaris/Linux
RDF-S3	Java	DB2, MySQL	eRQL	unabhängig
Sesame	Java	PostgreSQL	RDQL, RQL	unabhängig
RDF Gateway	—	enthalten – deduktiv	RDFQL	Windows 2000 / XP / NT
Redland	C	Berkeley DB, SQLite, AKT Trip- lestore, MySQL	SPARQL, RDQL	Linux, FreeBSD, Solaris, OSF/1, MacOS X
Kowari	Java	enthalten (Eigenentwicklung)	iTQL, (RDQL, SPARQL)	unabhängig

2/9.3.3 Anfragen

Da es vom W3C lange keinen Vorschlag für eine Anfragesprache gab, haben sich über die Zeit sehr viele Ansätze mit unterschiedlichen Vor- und Nachteilen entwickelt. Anfang 2004 wurde die *W3C RDF Data Access Working Group (DAWG)* gegründet. Im Oktober 2004 wurde ein erstes Arbeitspapier (Working Draft) für die RDF-Anfragesprache SPARQL veröffentlicht. SPARQL ist zwar noch nicht als Standard verabschiedet, langfristig wird es aber sicherlich dazu kommen. Die vorhandenen Anfragesprachen werden dadurch jedoch nicht obsolet. Es ist vielmehr davon auszugehen, dass vorhandene Werkzeuge zusätzlich zu den eigenen und den unterstützten Anfragesprachen eine Schnittstelle für SPARQL anbieten werden. Als Vorgänger von SPARQL kann die Anfragesprache RDQL bezeichnet werden, die wir als erstes betrachten wollen.

RDQL

Die *RDF Data Query Language (RDQL)* basiert auf einer früheren Anfragesprache mit dem Namen SquishQL. Als Bestandteil von Jena hat RDQL einen sehr hohen Verbreitungsgrad erreicht und wird daher von vielen anderen Systemen unterstützt. RDQL ist wohl die am weitesten verbreitete Anfragesprache für RDF.

RDQL arbeitet auf dem Graph-Modell von RDF. Die Syntax für RDQL-Anfragen ist an SQL angelehnt. Eine einfache Anfrage hat die Form: *select RÜCKGABE where BEDINGUNG*. Dabei werden in der Bedingungsklausel RDF-Teilgraphen formuliert, die Variablen enthalten. Variablen werden durch vorangestellte Fragezeichen gekennzeichnet. In der Rückgabeklausel wird definiert, welche der verwendeten Variablen zurückgegeben werden sollen. Die Anfrage nach dem Vornamen von Picasso könnte folgendermaßen lauten, wobei die beiden verwendeten Eigenschaften bekannt sein müssten:

```
select ?name
where (?x, <http://www.culture.net/lname>,
"Picasso") , (?x, <http://www.culture.net/fname>, ?name)
```

In dieser RDQL-Anfrage werden als Bedingung zwei so genannten *Graph-Patterns* durch Kommata verbunden. Die Graph-Patterns stellen Beispielaussagen dar mit den Variablen als Platzhalter. In den gespeicherten Graphen werden passende Fragmente gesucht und die in den Fragmenten vorhandenen Werte für die Ausgabevariablen dem Ergebnis hinzugefügt. Zusätzlich können Filter über den Operator *AND* angehängt werden. Hierüber lassen sich die Variablen durch arithmetische Vergleiche, Text-Vergleiche oder andere Bedingungen einschränken. Möchte man nach Vornamen von *Picasso* suchen, die nicht *Pablo* lauten, würde man an die obige Anfrage Folgendes anhängen:

Graph-Patterns

```
AND ?name NE "Pablo"
```

Dabei wird die Notation der Programmiersprache *Perl* verwendet. Das *NE* steht für „*not equal*“.

Am Ende einer RDQL-Anfrage kann eine *USING*-Klausel angefügt werden. In ihr können Präfixe für Namensräume definiert werden, um URIs innerhalb der Anfrage abzukürzen und so komplexe Anfragen kompakter darstellen zu können. Weiterführende Vereinfachungen oder Kurzschreibweisen sind nicht vorgesehen. Das Ergebnis einer RDQL-Anfrage ist eine Menge von Werten für die Rückgabevariablen, die den gegebenen Bedingungen entsprechen.

Mithilfe der optionalen *FROM*-Klausel zwischen *RÜCKGABE*- und *USING*-Klausel kann durch Angabe der URI gezielt auf ein einzelnes Modell eingeschränkt werden. Dies ist nur möglich, wenn z.B. wie beim Jena-System die Aussagen in einzelnen Modellen abgespeichert werden. Im gewissen Umfang bietet daher auch RDQL die Möglichkeit, auf Quadrupeln zu arbeiten. Die Bedeutung eines Modells ist in RDQL nicht festgelegt.

Semantic-Web-Programme

Dies ist zwar sehr flexibel, verhindert jedoch, dass Werkzeuge es eindeutig interpretieren und dass entsprechende Funktionen mitgeliefert werden können.

SPARQL

RDQL-Nachfolger

SPARQL ist stark an RDQL angelehnt und gilt mittlerweile als Nachfolger von RDQL. Wiederholen wir die Anfrage nach dem Vornamen von Picasso, sehen wir nur sehr geringe Unterschiede in der Syntax:

```
select ?name
where {?x <http://www.culture.net/lname> "Picasso" .
      ?x <http://www.culture.net/fname> ?name }
```

RDF Datasets

SPARQL bietet verglichen mit RDQL einige Erweiterungen. Als Beispiel wurde der Umgang mit *RDF Datasets* eingefügt. Ein RDF Dataset ist eine Menge von RDF-Graphen, die über eine URI identifiziert werden können (*Named Graphs*).

Hintergrund-Graph

Ferner existiert ein *Hintergrund-Graph (Background Graph)*, der z.B. Metadaten über die Graphen aufnehmen kann. Das heißt, SPARQL geht über die reine Anfrage auf Tripel oder Aussagenebene hinaus und ermöglicht es, die Quadrupelstruktur auszunutzen, die viele Speichersysteme anbieten.

RQL

Typisierte Anfragesprache

Die *RDF Query Language (RQL)* ist eine typisierte Anfragesprache, die auf einem funktionalen Ansatz aufbaut (ähnlich OQL⁹). Das formale Datenmodell und Typsystem, auf dem RQL aufbaut, unterscheidet dabei nach einfachen Daten, Schemadaten und Metaschemadaten.

⁹ Object Query Language

RQL bietet eine Reihe von Basisanfragen und Funktionen, wobei der Fokus auf der Abfrage von so genannten *Schemaanfragen* liegt. Unter *Schemaanfragen* wird die gezielte Suche nach RDF-Klassen, Mitgliedern von RDF-Klassen, Ableitungen von RDF-Klassen usw. verstanden.

Schemaanfragen

Beispielhaft seien hier nur die Funktionen *subClassOf* und *subPropertyOf* zum Anzeigen von Klassen- und Eigenschaftshierarchien genannt. Die Anfrage *subClassOf(Artist)* würde alle Unterklassen der RDF-Klasse mit dem Namen *Artist* zurückgeben. Als großer Unterschied gegenüber RDQL werden bei RQL-Anfragen generell auch die transitiven Beziehungen berücksichtigt, d.h. bei der Verwendung von RDF-Klassen werden auch die davon abgeleiteten Klassen berücksichtigt. Dies kann jedoch durch das \wedge -Zeichen unterbunden werden. So gibt die Anfrage *subClassOf \wedge (Artist)* nur die direkten Unterklassen der Klasse *Artist* wieder.

Gleiches gilt für Eigenschaften. Werden alle Verbindungen der Eigenschaft *creates* gesucht, kann dies in RQL über die Kurzschreibweise *creates* erfolgen, also durch einfache Angabe des Eigenschaftsnamens. Dabei werden neben den direkten Verbindungen auch die Verbindungen zurückgegeben, die durch Untereigenschaften von *creates* gebildet werden. Die direkten Verbindungen können durch \wedge *creates* ermittelt werden.

Des Weiteren nutzt auch RQL die Anfragestruktur *select ... from ..where ..* wie von SQL bekannt und auch von RDQL verwendet. Dabei werden in der *FROM*-Klausel *Pfadausdrücke* wie z.B. *{X}title{Y}* eingetragen. Pfadausdrücke beschreiben Beziehungen zwischen Ressourcen und binden die Beschreibungen an Variablen. Ein Beispiel wäre die Anfrage *select X from {X}title{Y}*, die alle Ressourcen zurückgibt, die als Subjekt mit der Eigenschaft *title* verwendet werden. Einfache Pfadausdrücke können durch einen Punkt verbunden werden. Die Anfrage nach Ressourcen vom Typ *Museum* und ihren Änderungsdaten, wenn sie nach 2005 geändert wurden, lautet:

```
select X, Y from Museum{X}.last_modified{Y}
where Y >= 2005-01-01
```

Semantic-Web-Programme

Als Alternative zur Verbindung von Pfadausdrücken können diese auch einzeln aufgeführt und in der *WHERE*-Bedingung verknüpft werden. Die obige Anfrage ist damit äquivalent zu:

```
select X, Y from Museum{X}, {Z}last_modified{Y}
where Y >= 2005-01-01 and X = Z
```

Die RQL-Anfragen können über Mengenoperatoren (Vereinigung, Durchschnitt, Differenz) verknüpft werden. Auch die Bildung geschachtelter Anfragen ist möglich. Dadurch wird RQL zu einer sehr mächtigen Anfragesprache. In seiner Ausdrucksfähigkeit geht es damit über RDQL hinaus. Das Ausnutzen existierender Quadrupelstrukturen wie bei SPARQL wird jedoch nicht unterstützt. Durch die Basisanfragen, Funktionen und einige Kurzschreibweisen bleibt RQL dabei einfacher zu bedienen als RDQL.

eRQL

„Google-artige“ RDF-Abfragen

Die Anfragesprache *easy RDF Query Language* (eRQL) ist eine Forschungsarbeit der Universität Frankfurt/Main. Sie ist sicherlich noch nicht reif für einen professionellen Einsatz, auch ist sie von ihrer Mächtigkeit sehr eingeschränkt. Im Gegensatz zu anderen Anfragesprachen lehnt sie sich jedoch nicht an SQL an und ist durch eine *relativ* einfache Syntax auch von unerfahrenen Nutzern bedienbar. In der aktuellen Version werden auch Zusatzinformationen wie Quellinformationen mit ausgenutzt. Ein eRQL-Interpreter ist in RDF-S3 enthalten und kann durch eine eigene Benutzeroberfläche einfach bedient werden.

RDFQL

RDF Query Language (RDFQL) wird von Intellidimension unterstützt. Es handelt sich dabei um eine serverseitige Scriptsprache, die auf ECMAScript (Java-Script) basiert. RDFQL geht damit über eine reine Anfragesprache hinaus. Die eigentlichen Anfragen innerhalb der Scriptsprache basieren auf einer SQL-ähnlichen Syntax.

Als weitere Besonderheit können innerhalb des Skripts Ableitungsregeln definiert werden, die von der RDF Gateway unterliegenden deduktiven Datenbank mitberücksichtigt werden. Dies macht RDFQL zu einem sehr mächtigen Werkzeug.

iTQL

Die *interactive Tucana Query Language* (iTQL) ist eine deklarative Anfragesprache, die auf SQL und SquishQL aufbaut (ähnlich wie RDQL). Die Anfrage nach dem Vornamen von Picasso würde in iTQL folgendermaßen lauten:

```
select $name
from <rmi://141.2.11.100/server1#sampledata>
where $x <http://www.culture.net/lname> "Picasso"
and $x <http://www.culture.net/fname> $name;
```

Zusätzlich zu den reinen Anfragen unterstützt iTQL jedoch auch die Manipulation der Daten. So können mit iTQL RDF-Modelle erzeugt und RDF-Aussagen hinzugefügt oder gelöscht werden. Auch werden einige Data Mining Operationen durch iTQL unterstützt.

Anfragesprache	Bemerkung/Besonderheit
RDQL	sehr hoher Verbreitungsgrad
SPARQL	Nachfolger von RDQL mit Erweiterungen zur Unterstützung von Quadrupeln, eventuell zukünftiger W3C-Standard
RQL	mächtiger als RDQL mit Fokus auf Schemaabfragen
eRQL	sehr einfach zu lernen, basiert nicht auf SQL, jedoch noch nicht marktreif
RDFQL	Scriptsprache mit der Möglichkeit, Regeln zum Ableiten neuer Aussagen zu integrieren
iTQL	bietet zusätzliche Funktionen wie Datenmanipulation und Data-Mining-Operationen

Semantic-Web-Programme

2/9.3.4 Inferenz

Inferenz

Die Möglichkeit der Ableitung neuer Aussagen aus Vorhandenen (*Inferenz*) macht das Semantic Web gerade so interessant. Bei einigen Produkten, die wir oben vorgestellt haben, wird dies bereits ausgenutzt. So werden bei RQL-Anfragen Klassenhierarchien automatisch mitberücksichtigt. Deutlich weiter geht die Firma Intellidimension mit ihren Produkten IferEd und RDF Gateway sowie mit ihrer Anfragesprache RDFQL. Das generelle Vorgehen ist, zu den existierenden Daten Regeln aufzustellen, wie neue Aussagen aus vorhandenen Aussagen abgeleitet werden dürfen. Ein Beispiel ist die Transitivität von Klassenhierarchien, die bereits in RDF Schema als Regel enthalten ist. Ist als Beispiel die Klasse der Säugetiere als Unterklasse der Tiere und die Klasse der Affen als Unterklasse der Säugetiere definiert, folgt daraus, dass die Klasse der Affen auch eine Unterklasse der Tiere ist. In diesem Abschnitt soll auf weitere Produkte und Möglichkeiten zur Inferenz eingegangen werden.

CWM**Datengesteuert**

Die *Closed World Machine* (CWM) wurde von Tim Berners-Lee und Dan Connolly vom W3C entwickelt. CWM ist in Python geschrieben. Neben der Speicherung wird auch die Ableitung von Aussagen angeboten. Dabei wird ein *datengesteuerter* Algorithmus (mit *Vorwärtsverkettung*) genutzt, d.h. es wird von den vorhandenen Daten ausgegangen und versucht, aus diesen neue Aussagen abzuleiten.

Euler Proof Mechanism**Euler Proof**

Hier handelt es sich um einen so genannten Beweiser (Reasoner). Die Inferenz-Maschine arbeitet mit einem *zielgerichteten* Algorithmus (mit *Rückwärtsverkettung*). Dabei wird erst eine Hypothese aufgestellt, die anhand der vorhandenen Daten und Regeln abzuleiten versucht wird.

GraphPath

GraphPath ist eine Sprache zur Analyse von Daten, die als Graphen strukturiert sind, insbesondere RDF. Es existiert eine Implementierung in Python, die über eine Schnittstelle z.B. mit Redland zusammenarbeitet. Die Implementierung befindet sich im Beta-Status und steht unter GNU Public Licence.

OntoBroker

OntoBroker ist ein kommerzielles Produkt der Ontoprise GmbH. Es beruht auf einer deduktiven Datenbank, in der die Aussagen als F-Logic-Fakten abgelegt werden. OntoBroker verfügt über die Möglichkeit, an verschiedene Datenquellen anzubinden, und unterstützt diverse Programmierschnittstellen. Laut Ontoprise-Angaben existieren bereits Konnektoren zu vielen gängigen Systemen.

Tucana's Knowledge Discovery Suite

Dieses kommerzielle Produkt von Tucana arbeitet auf gerichteten Graphen. Dabei wird intern RDF verwendet. Ob die Suite auf das von Tucana entwickelte Speicherprogramm Kowari aufbaut, wird in der Dokumentation nicht erwähnt, liegt aber nahe. Wie OntoBroker kann das Programm zur Integration firmenweiter Informationen, Stichwort *Enterprise Information Integration* (EII), eingesetzt werden.

Enterprise Information Integration

Semantic-Web-Programme

2/9.4 Ausblick

Trotz aller Vorzüge und Innovationen des Semantic Web gibt es auch kritische Anmerkungen und Bedarf für weitere Techniken.

Wenn im Semantic Web Maschinen das Filtern von Informationen für den Menschen übernehmen und beginnen, selbstständig Informationen miteinander zu verknüpfen, dann erhält die Glaubwürdigkeit dieser Informationen eine besondere Brisanz. Ist heutzutage eine angezeigte Internetseite für den Menschen (eventuell) noch als „Fälschung“ oder „Suchmaschinen-Spamming“ zu erkennen, ist dies im Semantic Web nicht mehr ohne weiteres der Fall. Einer einzelnen Aussage lässt sich kein Wahrheitsgehalt ansehen. Aus den Erfahrungen dieser Suchmaschinen und aus ihrem Kampf gegen Suchmaschinen-Spamming ist davon auszugehen, dass auch im Semantic Web der Zukunft falsche und irreführende Informationen absichtlich veröffentlicht werden. Auf dieses Problem wird ausführlich in [9] eingegangen.

Suchmaschinen-Spamming

Trotz vieler Vorzüge von RDF fällt es insbesondere bei kommerziellen Projekten schwer, Unternehmen für RDF zu gewinnen. Oftmals scheitert es an einer übermächtigen XML-Lobby und der Angst vor dem „Neuen“. Auch spielt das *Bootstrapping-Problem* eine Rolle, die fehlende kritische Masse an RDF-Daten, sodass es schwer fällt, den wirklichen Nutzen des Semantic Web erfahren zu können. Da dadurch keine zusätzlichen RDF-Daten erstellt werden, entsteht ein Zyklus, der nur schwer zu durchbrechen ist.

Bootstrapping-Problem

Durch die hohe Varianz in den RDF-Spezifikationen der vergangenen Jahre wurden weitere Probleme geschaffen. Da die Standards seit nun über zwei Jahren nicht verändert wurden, steigt die Hoffnung, dass hier etwas Stabilität erreicht wurde. Jedoch gibt es immer noch Probleme und offene Punkte. Einige können vielleicht durch klarere Strukturen in höheren Ebenen behoben werden –wie Ontologie- oder Logik-Ebene. Dadurch steigt aber die Gesamtkomplexität weiter an und so auch die Hemmschwelle, in das Semantic Web einzusteigen.

Varianz der RDF-Spezifikationen

Ausblick

Die Datenmenge im Internet wächst stetig, und das Semantic Web wird kommen. Ob dies durch heutige Techniken wie RDF und OWL geschieht, oder ob andere Standards sich durchsetzen, muss die Zukunft zeigen. Vergleicht man RDF auch hier wieder mit SGML bzw. XML, muss evtl. auch RDF zunächst an Komplexität verlieren und auf seine Zeit warten.

Rolle der Unternehmen

Die breite Nutzung von RDF im privaten Sektor ist vermutlich für die nächsten fünf bis zehn Jahren unrealistisch. Die Entwicklung sollte von Unternehmen vorangetrieben werden, z.B. durch die Bildung wissensbasierter Systeme, vorwiegend infolge der besseren Wartbarkeit und der erhöhten Flexibilität im Vergleich zu XML. Nachdem RDF sich in den Intranets der Unternehmen etabliert hat, wird es für diese nahe liegen, unkritische Bereiche frei zugänglich zu machen, wodurch schließlich das Bootstrapping-Problem gelöst werden kann.

Bundesförderung

Die Lukrativität des Semantic Web zeigt die steigende Anzahl von Firmen wie Ontoprise, Intellidimension und insbesondere Altova, aber auch die Höhe der Fördergelder, die in diesen Bereich fließen. Das Bundesministerium für Bildung und Forschung (BMBF) fördert das Semantic Web bis 2006 mit 13,7 Millionen Euro.

2/9.5 Quellen

Referenzen

- [1] Fabian Wleklinski: Suche im Semantic Web –Erweiterung des VRP um eine intuitive und RQL-basierte Anfrageschnittstelle. Diplomarbeit an der Universität Frankfurt am Main, November 2003.
- [2] Tim Berners-Lee, Eric Miller: The Semantic Web lifts off. ERCIM News Nr. 51, online unter: http://www.ercim.org/publication/Ercim_News/enw51/berners-lee.html, Oktober 2002.
- [3] Dieter Fensel, James Hendler, Henry Lieberman, Wolfgang Wahlster (Hrsg.): Spinning the Semantic Web. MIT Press, ISBN 0-262-06232-1, 2003.
- [4] Tim Berners-Lee, James Hendler, Ora Lassila: The Semantic Web. Scientific American, May 2001.
- [5] Mark Frauenfelder: A Smarter Web. Technology Review, online unter: http://www.technologyreview.com/Info-Tech/wtr_12640,258,p1.html, November 2001.
- [6] Cai Ziegler: Deus ex Machina –Das Web soll lernen, sich und uns zu verstehen. c't Heft 6, 2002.
- [7] Dave Beckett: RDF/XML Syntax Specification (Revised). W3C Recommendation, online unter: <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>, 10 Februar 2004.
- [8] SWAD-Europe Deliverable 12.1.1: Semantic Web applications –analysis and selection: http://www.w3.org/2001/sw/Europe/reports/chosen_demos_rationale_report/hp-applications-selection.html
- [9] Dissertation von Karsten Tolle: „Semantisches Web und Kontext –Speicherung von und Anfragen auf RDF-Daten unter Berücksichtigung des Kontextes“
- [10] Wikipedia Enzyklopädie: RDF, online unter: <http://de.wikipedia.org/wiki/RDF>

Quellen**Internetressourcen**

[N3] An readable language for data on the Web:
<http://www.w3.org/DesignIssues/Notation3.html>

[DOM] Document Object Model (DOM):
<http://www.w3.org/DOM/>

[SAX] official website for SAX: <http://www.saxproject.org/>

[XSLT] XSL Transformations (XSLT):
<http://www.w3.org/TR/xslt>

[XML Infoset] RDF Primer –XML Information Set:
http://de.wikipedia.org/wiki/XML_Information_Set

[RDF Model] The RDF Model: <http://www.w3.org/TR/rdf-primer/#rdfmodel>

[XQuery] XQuery 1.0: An XML Query Language:
<http://www.w3.org/TR/xquery/>

[CSV] Wikipedia –CSV-Datei:
<http://de.wikipedia.org/wiki/CSV-Datei>

[XML Schema] XML Schema:
<http://www.w3.org/XML/Schema>

[DTD] Wikipedia –Dokumenttypdefinition:
<http://de.wikipedia.org/wiki/DTD>

[DAML] The DARPA Agent Markup Language Homepage:
<http://www.daml.org/>

[OWL] Web Ontology working group:
<http://www.w3.org/2001/sw/WebOnt/>

[XML] Extensible Markup Language (XML):
<http://www.w3.org/XML/>

[RDF] Resource Description Framework (RDF):
<http://www.w3.org/RDF/>

[RDFS] RDF Vocabulary Description Language 1.0: RDF Schema: <http://www.w3.org/TR/rdf-schema/>

[Yahoo] Yahoo!: <http://www.yahoo.com>

[DMOZ] Open Directory Project: <http://www.dmoz.org>

[Google] Google: <http://www.google.com>

[Schichten] Semantic Web Architecture:
http://www.w3.org/2001/sw/Europe/reports/chosen_demos_rationale_report/hp-applications-selection.html#SEMWEB_LAYERS

[Mozilla] Semantic Web technologies in Mozilla:
<http://www.mozilla.org/rdf/doc/>

[FireFox] Firefox: <http://www.mozilla.com/firefox/>

[openBC] Open Business Club: <https://www.openbc.com>

[RSS] Really Simple Syndication:
<http://de.wikipedia.org/wiki/RSS>

[Blogs] Weblog: <http://de.wikipedia.org/wiki/Blog>

[XMP] Adobe's Extensible Metadata Platform:
<http://www.adobe.com/products/xmp/>

[Dublin Core] The Dublin Core Metadata Initiative:
<http://dublincore.org>

[hCalendar] <http://microformats.org/wiki/hcalendar>

Quellen