

Seminar „DB-Aspekte des E-Commerce: Schwerpunkt Anwendungen“

Das Semantic Web – Ontologien, RDF, DAML+OIL

Boris Stumm

16. Januar 2002

Zusammenfassung

In Zukunft wird es immer wichtiger, die gewaltige Anzahl an Dokumenten und Daten, die im Internet vorhanden, sind in einer Weise zu beschreiben, die es nicht nur Menschen, sondern auch Maschinen ermöglicht, sie zu verarbeiten und zu *verstehen*. Agenten werden dann sinnvoll einsetzbar, und auch Suchmaschinen werden bessere Ergebnisse liefern. Da das Internet und die Anzahl der in ihm enthaltenen Daten mit ungeheurer Geschwindigkeit wachsen, wird es immer unüberschaubarer und ohne die Unterstützung von Maschinen immer weniger effizient nutzbar.

Eine Grundlage für das Semantic Web ist ein einheitliches Beschreibungsmodell und eine standardisierte Sprache, um maschinelle Verarbeitung und Kommunikation überhaupt erst zu ermöglichen. Weiterhin muß man berücksichtigen, daß nicht jede Information im Internet zuverlässig ist und durchaus auch widersprüchliche Informationen existieren.

Hier wird ein kurzer Abriß gegeben über die allgemeine Problematik und Notwendigkeit eines Semantic Web, dann folgt eine Einführung in RDF, RDF Schema und DAML+OIL, welche die Basis zur Beschreibung von Metadaten und der Erstellung von Ontologien liefern. Schließlich wird das Thema Zuverlässigkeit von Informationen diskutiert, und danach werden einige Anwendungsbereiche vorgestellt.

Inhaltsverzeichnis

1	Einführung	1
1.1	Die Semantik im Internet	1
1.2	Wie können Computer Informationen verstehen?	1
1.3	Anwendungen	2
2	Grundlagen des Semantic Web	2
2.1	Uniform Resource Identifier (URI)	2
2.2	Daten und Metadaten	3
2.3	Schemas und Ontologien	3
3	RDF	4
3.1	Datenmodell	4
3.2	Bezug zu OO-Technologie und ER-Diagrammen	6
3.3	Serialisierung (N3, XML)	6
3.3.1	Notation 3 (N3)	7
3.3.2	RDF/XML	7
3.4	Weitere Merkmale	8
4	RDF-Schema	9
4.1	Klassen und Eigenschaften	9
4.2	Erweiterungsmechanismen und Evolution	10
5	DAML+OIL	10
5.1	Spracheigenschaften	10
5.1.1	Allgemeines	10
5.1.2	Typsystem	11
5.1.3	Definition von Klassen	11
5.1.4	Definition von Eigenschaften und Restriktionen	13
6	Weitere Aspekte	13
6.1	Dublin Core Metadata Element Set	13
6.2	Inferenz im Semantic Web	13

7	Das Internet als Wissensbasis – Open-World-Modell	14
7.1	Einführung	14
7.2	Das Web of Trust	14
8	Anwendungen	15
8.1	Composite Capabilities/Preference Profiles (CC/PP)	15
8.2	Suchmaschinen	16
8.3	Agenten	16
8.4	Mozilla	16
8.5	Das Open Directory Project	17
9	Ausblick und Kritik	17

1 Einführung

In diesem Abschnitt wird eine kurze Einführung in das Semantic Web gegeben, und im nächsten Abschnitt werden einige notwendige Grundlagen erläutert.

Danach folgt eine Beschreibung von RDF und der zugehörigen Erweiterungen. Im Anschluß daran wird noch auf andere Sprachen zur Datenbeschreibung und auch zur Beschreibung von Inferenzregeln eingegangen.

In den letzten Abschnitten werden dann noch ein grundlegendes Problem des Internet, nämlich die Frage nach der Zuverlässigkeit von Informationen, behandelt sowie einige Anwendungen vorgestellt. Letztendlich wird versucht, das Semantic Web jetzt und in der Zukunft bedeutungsmäßig zu positionieren.

1.1 Die Semantik im Internet

Das Internet war und ist primär ein Kommunikations- und Informationsmittel für Menschen. Auch Computer kommunizieren über das Internet, aber sie können es nicht als Informationsmedium nutzen, da seine Inhalte für Maschinen „unverständlich“ und somit unbrauchbar sind.

Im Semantic Web jedoch soll es auch Maschinen ermöglicht werden, Informationen zu suchen und zu verwerten, und so komplexere Aufgaben wahrzunehmen, die heute nur von Menschen gelöst werden können.

1.2 Wie können Computer Informationen verstehen?

Um das Semantic Web zu verwirklichen, ist es nötig, daß Maschinen selbständig Daten im Internet verwerten und nutzbringende Informationen herausfiltern können. Daß dies nicht allzu einfach ist, kann man am Umfang und der Qualität der Ergebnismengen sehen, die heutzutage Suchmaschinen liefern.

Es muß also ein Weg gefunden werden, den Maschinen die vorhandenen Informationen im Internet „verständlich“ zu machen. Prinzipiell kann man das auf zwei verschiedene Arten tun:

1. Die Maschinen können dazu gebracht werden, daß sie natürliche Sprache verstehen. Die KI-Forschung arbeitet seit langem an diesem Problem, ohne eine allgemeine und zufriedenstellende Lösung gefunden zu haben. Wann und ob überhaupt jemals eine Lösung gefunden wird, ist nicht vorauszusehen.
2. Eine andere Möglichkeit ist es, Daten so aufzubereiten, daß sie von Maschinen „verstanden“ werden. Das bedeutet einerseits einen Mehraufwand für die Informationsaufbereitung und -bereitstellung, vermeidet allerdings die im ersten Punkt beschriebenen Probleme weitgehend, da die Computer nun sozusagen in ihrer Sprache kommunizieren können.

Im Semantic Web wird der zweite Ansatz verfolgt, und in den folgenden Abschnitten werden die nötigen allgemeinen Grundlagen und Voraussetzungen sowie eine gemeinsame, standardisierte Metadatensprache und eine Sprache zur Erstellung von Ontologien vorgestellt.

1.3 Anwendungen

Es sind viele Anwendungen für das semantische Web bzw. Anwendungen, die das semantische Web nutzen, denkbar:

- Suchmaschinen können mittels aufbereiteter Daten bessere Ergebnisse erzielen.
- Inhalte von Websites können in einer Inhaltsübersicht beschrieben werden.
- Es werden die Grundlagen für „Intelligente Agenten“, die miteinander kommunizieren und ihnen gestellte Aufgaben selbständig bewältigen können, möglich.

Für die meisten Anwendungen ist es wichtig, daß die Informationen, die gelesen und verarbeitet werden, zuverlässig sind. Darauf wird später noch eingegangen.

2 Grundlagen des Semantic Web

In diesem Abschnitt werden die wichtigen Grundlagen und Voraussetzungen für das Semantic Web beschrieben.

2.1 Uniform Resource Identifier (URI)

Um über beliebige Objekte zu sprechen, müssen diese eindeutig identifizierbar sein. Dies wird mit den Uniform Resource Identifiers („Einheitliche Ressourcen-Bezeichner“) erreicht.

Eine URI sagt im allgemeinen jedoch weder etwas darüber aus, wie das referenzierte Objekt erreicht werden kann, noch bietet sie Informationen über das Objekt selbst. Das ist verständlich, da ein von einer URI identifiziertes Objekt nicht unbedingt über das Internet erreichbar sein muß.

Eine bekannte Form der URI, die URL (Unified Resource *Locator*), enthält auch Information, um die Ressource zu lokalisieren. Im Gegensatz dazu steht der Unified Resource Name (URN).

Die genaue Syntax einer URI wird in RFC 2396 [1] beschrieben.

Prinzipiell kann jedes denkbare Objekt eine URI haben und sogar mehrere, da es jedermann möglich ist, eine URI für ein Objekt zu kreieren. Daraus ergibt sich

das Problem, daß man im allgemeinen von zwei verschiedenen URIs nicht sagen kann, ob sie dasselbe Objekt referenzieren. Das ist ein nicht lösbares Problem im dezentral organisierten Internet, da jedermann für beliebige Ressourcen URIs definieren kann. Dazu im Abschnitt 7 mehr.

2.2 Daten und Metadaten

Metadaten sind „Daten über Daten“, d.h., sie beschreiben andere Daten. Kataloge sind beispielsweise eine Form von Metadaten, da sie Beschreibungen von Dingen enthalten, nicht die Darstellung der Dinge selbst.

Viele Anwendungen benötigen Metadaten, um mit Daten arbeiten zu können, wie zum Beispiel Schemainformationen von Datenbanken, um SQL-Anfragen stellen zu können.

Im Semantic Web haben Metadaten eine große Bedeutung, da Anwendungen hier zumeist auf Metadaten arbeiten und die „echten“ Daten weiterhin von Menschen verarbeitet werden. Bücher sind ein Beispiel: Der Bibliothekscomputer arbeitet nur mit den Metadaten wie Autor, Verlag, Zusammenfassung, Titel usw., wohingegen nur Personen das Buch tatsächlich lesen werden.

Die Unterscheidung von „normalen“ Daten einerseits und Metadaten andererseits ist fließend, je nach Anwendung mag sie variieren, auch Metadaten höherer Ordnung (Daten über Daten über Daten...) kommen vor.

Für das Semantic Web muß eine standardisierte Sprache zur Beschreibung von Metadaten existieren, um den Austausch von Dokumenten zu vereinfachen und das Verstehen von Information überhaupt erst zu ermöglichen.

2.3 Schemas und Ontologien

Auch eine Metadatensprache bringt alleine noch keine Semantik ins Internet. Dazu müssen Schemas und Ontologien definiert werden.

Durch die Verwendung von Standard-Schemas zur Beschreibung von Netzwerkreisourcen (siehe auch Abschnitt 6.1) oder Produkten eines Onlineshops kann ein Agent genau feststellen, mit welchen Ressourcen er zu tun hat und ob diese für seine Aufgabe nützlich sind. Beispielsweise hat jeder Onlineshop ein Schema für seinen Produktkatalog. Unterscheiden sich allerdings alle Onlineshop-Schemas voneinander, so wird es unmöglich, Anwendungen zu schreiben, die mit *allen* Onlineshops umgehen können. Wünschenswert ist die Verwendung von Standard-Schemas in solchen Umgebungen. Allerdings ist das nicht immer möglich, und dann braucht man eine Möglichkeit, verschiedene Schemas aufeinander abzubilden.

Einer der wichtigsten Aspekte des Semantic Web ist die Spezifizierung von Ontologien. In [2] wird eine Ontologie folgendermaßen definiert: „*Eine Ontologie*

stellt eine formale Beschreibung der Gegenstände und Beziehungen dar, die für eine Person oder Gruppe von Personen begriffsbildend sind“.

Durch solche formalen Beschreibungen wird es für Maschinen möglich, mit den in der Ontologie spezifizierten Objekten zu arbeiten, weiterhin wird die Interaktion zwischen verschiedenen Anwendungen ermöglicht, da sie sozusagen „die gleiche Sprache sprechen“.

Im Semantic Web wird es nicht *eine* Ontologie geben, sondern sehr viele, die jeweils einen kleinen Anwendungsbereich abdecken. Es werden sich auch viele Ontologien ähnlich sein, da sie von unterschiedlichen Personen für gleiche Anwendungsgebiete erstellt wurden. Man braucht auch hier eine Möglichkeit, um Ontologien beziehungsweise die von ihnen spezifizierten Begriffe aufeinander abzubilden.

Möglicherweise widersprechen sich zwei Ontologien in ihren Definitionen. Das muß kein Fehler in der Spezifikation sein, sondern kann auf Unterschieden in der Weltanschauung der jeweiligen Autoren beruhen (siehe [13]). Dieses Problem wird in Abschnitt 7 näher beleuchtet.

3 RDF

Das Resource Description Framework ist ein Modell zur Repräsentation von Metadaten und wird in einer W3C Recommendation [3] spezifiziert. Laut Spezifikation soll RDF eine anwendungs- und domänenunabhängige Beschreibungsmöglichkeit für Ressourcen bieten, ohne im voraus eine Semantik festzulegen.

3.1 Datenmodell

Das Basis-Datenmodell von RDF ist sehr einfach aufgebaut, man kann folgende Objekte unterscheiden:

- Ressourcen („*Resources*“): Alle in RDF beschriebenen Dinge werden als Ressourcen bezeichnet. Eine Ressource wird durch ihre URI benannt und kann alles sein, eine einzelne Webseite, eine Sammlung von Webseiten oder auch eine nicht im Internet vorhandene Sache wie beispielsweise ein Buch.
- Eigenschaften („*Properties*“): Eine Eigenschaft beschreibt Charakteristika, Relationen und Attribute von Ressourcen. Auch Eigenschaften sind Ressourcen und können somit wiederum beschrieben werden.
- Aussagen („*Statements, Assertions*“): Eine Aussage besteht aus drei Teilen: Subjekt, Prädikat und Objekt. Das Subjekt ist eine Ressource, über die die Aussage gemacht wird, das Prädikat ist eine bestimmte Eigenschaft und das Objekt der Wert dieser Eigenschaft. Das Objekt einer Aussage kann eine Ressource sein oder ein Literal.

In der RDF-Syntax-Spezifikation werden Aussagen als gerichtete, benannte Graphen dargestellt, jeder ovale Knoten stellt eine Ressource dar, jeder rechteckige ein Literal, und die gerichteten Kanten entsprechen Eigenschaften und weisen vom Subjekt zum Objekt. In diesem Abschnitt werden Eigenschaften der Lesbarkeit halber mit einem einfachen Namen bezeichnet.



Abbildung 1: Zu lesen als: „The resource `http://www.w3.org/Home/Lassila` has Ora Lassila as its creator.“

In Abbildung 1 wird die Eigenschaft „Creator“ aus dem Dublin Core Element Set [6] als Prädikat benutzt (siehe auch Abschnitt 6.1). „Ora Lassila“ ist hier ein Literal, und keine Ressource. Literale haben jedoch keine weiteren Eigenschaften, da sie nicht als Subjekt in einer RDF-Aussage erlaubt sind.

Um weitere Aussagen über den Erzeuger der Ressource „`http://www.w3.org/Home/Lassila`“ zu machen, ist es nötig, diesen auch als Ressource zu modellieren:

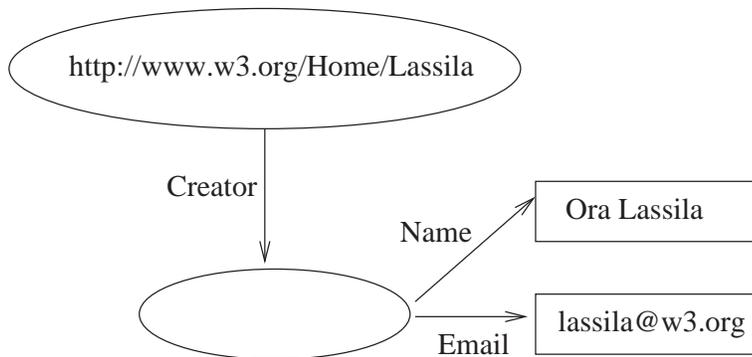


Abbildung 2: Zu lesen als: „The individual whose name is Ora Lassila, email <lassila@w3.org>, is the creator of `http://www.w3.org/Home/Lassila`.“

Hier (Abb. 2) ist das Objekt eine anonyme Ressource, dargestellt als ein leeres Oval. Wenn das Objekt durch eine URI identifizierbar ist, beispielsweise `http://www.w3.org/staffId/85740`, kann man das Objekt mit dieser URI benennen (Abb. 3).

Wie man solche Graphen serialisieren kann, wird in 3.3 erklärt.

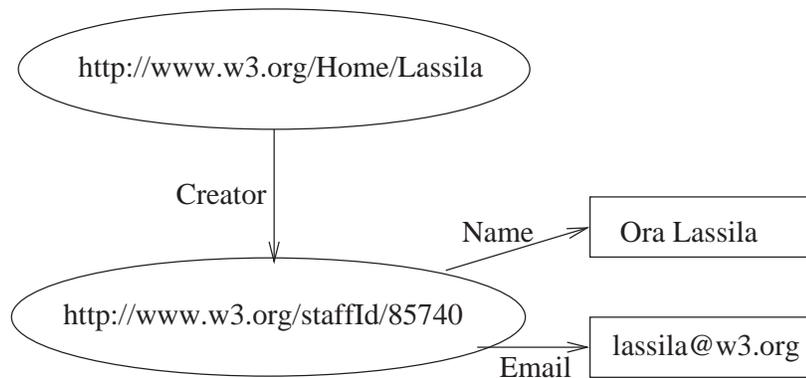


Abbildung 3: Zu lesen als: „The individual referred to by employee id 85740 is named Ora Lassila and has the email address lassila@w3.org. The resource <http://www.w3.org/Home/Lassila> was created by this individual.“

3.2 Bezug zu OO-Technologie und ER-Diagrammen

Das RDF-Modell baut auf bewährten Prinzipien der Datenrepräsentation auf. Im objektorientierten Design entspricht eine Ressource einem Objekt und seine Eigenschaften den Instanzvariablen. Weiterhin repräsentieren Eigenschaften auch Beziehungen zwischen Ressourcen. So können RDF Schemas auch als Entity-Relationship-Diagramme aufgefaßt werden.

3.3 Serialisierung (N3, XML)

Das RDF-Modell liefert eine abstrakte Vorlage zum Definieren und Benutzen von Metadaten. Zur praktischen Nutzung, wie der Erstellung von Metadaten und deren Austausch, ist jedoch eine konkrete Syntax erforderlich.

Da Ressourcen immer durch ihre URI identifiziert werden, und URIs oft etwas länglich sind, bieten die Serialisierungssyntaxen Abkürzungsmöglichkeiten, um Präfixe von URIs zu verkürzen. Im folgenden werden die Abkürzungen `rdf`, `rdfs`, `dc` und `daml` für die Spezifikationen von RDF, RDF Schema, Dublin Core Element Set und DAML+OIL verwendet; `dc:creator` entspricht dann also beispielsweise der Creator-Ressource, die im Dublin Core Element Set spezifiziert wird.

Nachfolgend werden zwei Serialisierungssyntaxen für RDF-Graphen vorgestellt, in den Beispielen werden die im Abschnitt 3.1 gezeigten Graphen serialisiert. Die Notationen sind im wesentlichen äquivalent, und ein Parser sollte beide unterstützen.

Die „Creator“-Eigenschaft wird hier nun vollständig als Ressource dargestellt, die im Dublin Core Element Set (siehe [6] und Abschnitt 6.1) definiert ist.

3.3.1 Notation 3 (N3)

Hier werden Aussagen als einfache Tripel geschrieben, die mit einem Punkt abgeschlossen werden. Man kann auch Abkürzungen definieren, ähnlich den Namensräumen bei XML. Verschiedene Ausprägungen einer Eigenschaft werden durch Kommas getrennt und verschiedene Eigenschaften eines Subjekts durch Semikolons. Die N3-Syntax ist etwas einfacher und intuitiver als RDF/XML aufgebaut.

Beispiel:

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <#>

# Beispiel aus Abb. 1
<http://www.w3.org/Home/Lassila> dc:creator "Ora Lassila" .

# Beispiel aus Abb. 2
<http://www.w3.org/Home/Lassila> dc:creator
    [ :name "Ora Lassila" ] ;
    [ :email "lassila@w3.org" ] .

# Beispiel aus Abb. 3
<http://www.w3.org/Home/Lassila> dc:creator
    <http://www.w3.org/staffId/85740> .
<http://www.w3.org/staffId/85740> :name "Ora Lassila";
    :email "lassila@w3.org".
```

3.3.2 RDF/XML

In der RDF-Spezifikation des W3C [9] wird RDF/XML eingeführt, eine auf XML basierende Syntax zur Beschreibung von RDF-Dokumenten. Es gibt zur Zeit (Dezember 2001) einen Working Draft zur RDF/XML-Spezifikation [4]. Für die Beispiele hier wird RDF/XML genutzt, da es die verbreitetere und „offizielle“ Syntax für RDF ist.

Beispiel:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns="http://wwwdbis...uni-kl.de/...ausarbeitung3.pdf#">

<!-- Beispiel aus Abb. 1 --!>
```

```

    <rdf:Description about="http://www.w3.org/Home/Lassila">
      <dc:Creator>Ora Lassila</dc:Creator>
    </rdf:Description>

<!-- Beispiel aus Abb. 2 --!>
    <rdf:Description about="http://www.w3.org/Home/Lassila">
      <dc:Creator>
        <rdf:Description>
          <Name>Ora Lassila</Name>
          <Email>lassila@w3.org</Email>
        </rdf:Description>
      </dc:Creator>
    </rdf:Description>

<!-- Beispiel aus Abb. 3 --!>
    <rdf:Description about="http://www.w3.org/Home/Lassila">
      <dc:Creator rdf:resource="http://www.w3.org/staffId/85740"/>
    </rdf:Description>
    <rdf:Description about="http://www.w3.org/staffId/85740">
      <Name>Ora Lassila</Name>
      <Email>lassila@w3.org</Email>
    </rdf:Description>

</rdf:RDF>

```

Der „about“-Tag des Description-Elements enthält die URI einer Ressource, die von ihm eingeschlossenen Elemente sind die Eigenschaften dieser Ressource. Es werden somit, ähnlich N3, mehrere Tripel, die dasselbe Subjekt haben, zusammengefasst. Das Beispiel zur Abb. 3 sieht etwas anders aus, hier wird eine Abkürzungstechnik benutzt, die das Objekt der Aussage nicht als Subelement der Eigenschaft, sondern als Attribut der Eigenschaft angibt.

3.4 Weitere Merkmale

Mit der RDF/XML-Notation kann man RDF-Tripel derart in HTML-Dokumente einbetten, das sie von Browsern nicht angezeigt werden, und somit die Beschreibung einer Webseite direkt in die Webseite integrieren. Solche HTML-Dokumente entsprechen dann zwar nicht mehr der HTML-Spezifikation, aber alle gängigen Browser behandeln solche HTML-Dokumente korrekt in der Form, daß das RDF nicht angezeigt wird.

Um Aussagen über Aussagen zu machen (und nicht über Ressourcen), muß ein Modell der Aussage erstellt werden, das dann wie eine Ressource behandelt wird. Um solche Modelle zu erstellen, definiert RDF die folgenden Eigenschaften: `rdf:subject`, `rdf:predicate`, `rdf:object`, `rdf:type`. Das Modell der

Aussage hat diese Eigenschaften jeweils einmal, der Wert der Typ-Eigenschaft ist `rdf:Statement`, die Werte der anderen Eigenschaften entsprechen den Teilen der ursprünglichen Aussage.

RDF ist eine sehr beschränkte Sprache. Erweiterungen wie RDF Schema und DAML+OIL erweitern RDF und bieten Möglichkeiten, Schemas und Ontologien zu erstellen.

4 RDF-Schema

RDF-Schema [5] ist kein Schema im eigentlichen Sinne, sondern erweitert RDF um Konstrukte zur Kreierung von Schemas, beispielsweise ein einfaches Spezialisierungskonzept. Dabei werden ein Typsystem und Eigenschaften wie `rdfs:Class` und `rdfs:subClassOf` definiert.

RDF-Schema ist eine Candidate Recommendation des W3C.

Im Gegensatz zu DTDs (siehe [11]), die eine Syntax spezifizieren, gibt RDF-Schema Informationen über die Bedeutung von Aussagen, also die Semantik.

4.1 Klassen und Eigenschaften

RDF-Schema definiert eine Reihe von Eigenschaften und Klassen, die zur Erstellung von Schemas benutzt werden können. Einige wichtige RDF-Schema-Elemente sind nachfolgend aufgeführt:

- `rdfs:Resource`: Alle in RDF beschriebenen Dinge sind Instanzen dieser Klasse, haben also implizit eine `rdf:type`-Eigenschaft mit dem Wert `rdfs:Resource`.
- `rdfs:Class`: Das Konzept ist ähnlich dem der Klassen in objektorientierten Sprachen. Jede Klasse, die definiert wird, muß also eine `rdf:type`-Eigenschaft mit Wert `rdfs:Class` haben.
- `rdfs:Property`: Dieses Element hat eine Funktion analog zu `rdfs:Class`, aber für Eigenschaften.
- `rdfs:subClassOf` und `rdfs:subPropertyOf`: Diese Eigenschaft entspricht dem Vererbungsmechanismus in den objektorientierten Sprachen.
- `rdfs:range` und `rdfs:domain`: Durch diese beiden Eigenschaften wird es möglich, den Wertebereich (`rdfs:range`) und den Definitionsbereich (`rdfs:domain`) einer Eigenschaft festzulegen. Eine Eigenschaft kann jeweils höchstens eine `rdfs:range`- und `rdfs:domain`-Eigenschaft haben.

4.2 Erweiterungsmechanismen und Evolution

Schemas sind statisch und ändern sich nicht, denn eine Schemadefinition wird im Internet von sehr vielen verschiedenen Personen benutzt, und Änderungen in einem Schema würden Änderungen in einer großen Anzahl anderer Dokumente nach sich ziehen sowie Anpassungen der Anwendungen.

Wenn doch Änderungen erforderlich werden, muß ein neues Schema erstellt werden, welches das alte erweitert. Bei den Teilen, die ohne Änderung übernommen werden, muß deutlich werden, daß sich die Bedeutung nicht geändert hat, das geschieht mit `rdfs:subClassOf`:

```
<rdf:Description rdf:ID="XY"
    rdfs:subClassOf="http://example.org/oldschema#XY"/>
```

Mit den Problemen, die diese Vorgehensweise mit sich bringt, befaßt sich Abschnitt 5.

5 DAML+OIL

In Abschnitt 2.3 wurde als einer der wichtigsten Voraussetzungen für das Semantic Web eine Möglichkeit zur Kreierung von Ontologien genannt. RDF und auch RDF Schema bieten eine solche Möglichkeit nicht bzw. nur ansatzweise.

Um Ontologien zu erstellen braucht man Konstrukte um Definitions- und Wertebereiche sowie Kardinalitäten genau zu spezifizieren. Weiterhin muß man Klassen genau voneinander abgrenzen können, und braucht logische Operatoren, um Axiome zu spezifizieren.

Ein wichtiges Sprachkonstrukt einer Ontologiesprache für das Internet ist die Erklärung der Äquivalenz zweier verschiedener Ressourcen (siehe Abschnitt 7).

DAML+OIL [7] ist eine solche Ontologiesprache, die auf RDF und RDF Schema aufbaut. DAML-ONT (Darpa Agent Markup Language - Ontology) und OIL (Ontology Inference Layer [8]) fließen in DAML+OIL zusammen. Die Sprache enthält viele der in Frame-basierten Sprachen gefundenen Konstrukte.

5.1 Spracheigenschaften

Im folgenden werden die Möglichkeiten von DAML+OIL aufgeführt.

5.1.1 Allgemeines

Mit den `daml:equivalentTo` und `daml:inverseOf`Eigenschaften können verschiedene Ontologien aufeinander abgebildet werden.

5.1.2 Typsystem

DAML+OIL unterscheidet zwischen Objekten und Datentypen. Objekte sind Instanzen von Klassen im Sinne von DAML+OIL oder RDF Schema, und die Datentypen bestehen aus den XML-Schema-Datentypen. DAML+OIL ist allerdings hauptsächlich auf die Kreierung von Klassen zur Beschreibung der Objektdomäne ausgerichtet.

5.1.3 Definition von Klassen

Klassen können in DAML+OIL auf verschiedene Art und Weise deklariert werden:

- Mit `rdfs:subClassOf` wird eine normale Spezialisierungsbeziehung wie auch in RDF Schema deklariert.
- Durch die Enumeration aller Elemente kann eine Klasse mit einer endlichen Anzahl von Instanzen definiert werden.
- Die Verwendung einer Boolean-Kombination von Klassenausdrücken¹, die die nachfolgenden Elemente enthalten kann, wird ebenfalls eine Klasse deklariert:
 - `daml:intersectionOf`: Die Klasse enthält alle Elemente, die in jeder der zugehörigen Klassen enthalten sind (analog der logischen Konjunktion)
 - `daml:unionOf`: Die Klasse setzt sich aus allen Elementen zusammen, die mindestens Instanz einer der enthaltenen Klassen sind (analog der logischen Disjunktion).
 - `daml:complementOf`: Die Klasse setzt sich aus allen Elementen zusammen, die in der spezifizierten Klasse nicht enthalten sind (analog der logischen Negation).

Es gibt zwei vordefinierte Klassen, `daml:Thing` und `daml:Nothing`. Jedes Objekt ist ein Element von `daml:Thing`, und kein Objekt ist ein Element von `daml:Nothing`.

Im Beispiel (Abb. 5.1.3) wird ausgedrückt, das die Klassen „Mann“ und „Frau“ disjunkt sind, und daß ein Mensch entweder ein Mann oder eine Frau sein muß.

¹Ein Klassenausdruck ist eine anonyme oder benannte Ressource, die vom Typ `rdfs:Class` ist.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns="http://wwwdbis.informatik.uni-kl.de/.../ausarbeitung3.pdf#">

  <daml:Ontology rdf:about=""/>

  <daml:Class rdf:ID="Mensch">
    <daml:unionOf rdf:parseType="daml:collection"
      <daml:Class rdf:about="#Mann/">
      <daml:Class rdf:about="#Frau/">
    </daml:unionOf>
  </daml:Class>

  <daml:Class rdf:ID="Mann">
    <rdfs:subClassOf rdf:resource="#Mensch"/>
  </daml:Class>

  <daml:Class rdf:ID="Frau">
    <rdfs:subClassOf rdf:resource="#Mensch"/>
    <daml:disjointWith rdf:resource="#Mann"/>
  </daml:Class>

</rdf:RDF>

```

Abbildung 4: DAML+OIL-Beispielontologie, notiert in der abgekürzten RDF/XML-Syntax.

5.1.4 Definition von Eigenschaften und Restriktionen

In DAML+OIL gibt es die Möglichkeit, Eigenschaften detaillierter zu beschreiben durch Angabe verschiedener Bedingungen.

- Es können Kardinalitätsrestriktionen spezifiziert werden.
- Es gibt Typrestriktionen ähnlich `rdfs:domain` und `rdfs:range`, jedoch mit etwas anderer Bedeutung.
- Es können Transitivitätsbeziehungen von Eigenschaften definiert werden.
- `daml:inverseOf` gibt an, daß eine Eigenschaft das Gegenteil einer anderen ist:

```
<daml:ObjectProperty rdf:ID="ehfrauVon" daml:inverseOf="#ehemannVon"/>
```

6 Weitere Aspekte

6.1 Dublin Core Metadata Element Set

Der Dublin Core Metadata Standard [6] definiert fünfzehn Elemente zur Beschreibung der Eigenschaften von Ressourcen, wie „Creator“, „Title“, „Language“, „Format“ und andere.

6.2 Inferenz im Semantic Web

DAML+OIL bietet die Möglichkeit, Ontologien zu erstellen und Objekte einer Ontologie in Beziehung zu setzen mit Objekten anderer Ontologien. Beispiele dafür sind `daml:equivalentTo` und `daml:subclassOf`. Dadurch wird es für Anwendungen möglich, Schlußfolgerungen zu ziehen und unbekannte Ressourcen in Beziehung zu setzen mit bereits bekannten, was dann wiederum die Verarbeitung solcher vorher unbekannter Ressourcen ermöglicht.

Alle mit `daml:equivalentTo` zu einem bekannten Objekt in Beziehung gesetzten Ressourcen können durch dieses ausgetauscht werden und alle Tripel, deren Prädikat als invers (`daml:inverseOf`) zu einem anderen Prädikat deklariert ist, können durch Ersetzung des Prädikats und Tausch von Objekt und Subjekt „verständlich“ gemacht werden.

Andererseits kann es notwendig werden, daß eine Aussage bewiesen wird, das heißt, von den Axiomen einer Ontologie muß auf die Korrektheit einer Aussage geschlossen werden.

Für die meisten Semantic-Web-Anwendungen werden Schlußfolgerungen und Beweise sich nicht über mehr als zwei oder drei Stufen erstrecken, da das Anwendungsfeld eng eingegrenzt ist und es eine Reihe von Standard-Vorgehensweisen gibt.

Prinzipiell sind auch automatische Beweiser vorstellbar, die ähnlich den aus der künstlichen Intelligenz bekannten Beweis-Systemen [17] vorgehen und das Internet als ihre Wissensbasis nutzen.

7 Das Internet als Wissensbasis – Open-World-Modell

7.1 Einführung

Das Internet unterscheidet sich von den gewöhnlichen Wissensbasen in der KI sehr stark; es ist dezentral organisiert, und somit gibt es eigentlich keine Kontrolle über den Inhalt.

In einem Closed-World-Modell wird angenommen, daß die Wissensbasis alles Wissen besitzt. Nicht in ihr enthaltene Aussagen besitzen den Wahrheitswert „falsch“, und alle enthaltenen Aussagen sind verläßlich.

Dagegen muß man das Internet als offen („Open-World“) ansehen, und das hat zwei Dinge zur Folge:

Erstens ist das Internet *nicht* vollständig; es gibt Informationen, die nicht im Internet vorhanden sind, und selbst wenn sie vorhanden sind, kann man sie nicht unbedingt erreichen. Somit können nicht vorhandene Aussagen nicht einfach als „falsch“ angenommen werden, sondern müssen als „unbekannt“ vermerkt werden.

Zweitens kann jeder Daten ins Internet stellen, und es gibt keine zentrale Kontrollstelle, die für die Konsistenz des Internet sorgt. Es ist möglich, daß vorhandene Informationen falsch sind oder im Konflikt zueinander stehen.

Wie mit nicht vorhandenen Informationen umgegangen wird, ist ein anwendungsspezifisches Problem. Die Feststellung der Zuverlässigkeit von Informationen und die Lösung von Konflikten sind jedoch Probleme, für die ein allgemeiner Weg gefunden werden muß.

7.2 Das Web of Trust

Um die oben genannten Probleme anzugehen, muß die Quelle einer Information² feststellbar sein sowie ihre Zuverlässigkeit.

In [14] werden verschiedene Modelle zur Schaffung von Vertrauen im Internet vorgestellt, darunter auch das Web of Trust (Netz des Vertrauens). Man kann hier anderen sein Vertrauen „aussprechen“, in dem man Zertifikate erstellt.

²Eine Information könnte hier ein beliebiges Dokument sein oder beispielsweise ein einziges RDF-Tripel.

Wenn man nun jemandem vertraut, vertraut man auch allen Personen, denen er sein Vertrauen ausgesprochen hat. Somit entsteht ein „Netz“ des Vertrauens. Denkbar ist es, verschiedene Vertrauensstufen einzuführen, die die Länge der „Vertrauenskette“ widerspiegeln und bei einer längeren Kette den Informationen nicht mehr so stark zu vertrauen.

Bei Anwendung des „Web of Trust“-Modells auf das Semantic Web fällt ein weiterer Punkt auf: Die gleiche Informationsquelle könnte in verschiedenen Kontexten unterschiedlich zuverlässig sein. Dies könnte ein Agent feststellen durch Vergleich der (RDF)-Beschreibung der Quellen oder einer Wertung von Dritten, die möglicherweise Ontologien für Zuständigkeitsbereiche definiert haben.

Durch ein solches Modell des Vertrauens kann jedem Dokument bzw. jeder Information ein bestimmter „Vertrauenswert“ zugeordnet werden, und Anwendungen mögen entscheiden, welche Informationen noch akzeptabel sind und welche nicht.

8 Anwendungen

Die eigentliche Zielrichtung beim Semantic Web besteht darin, eine höhere Informationsqualität zu gewinnen. Es soll möglich sein, bisher von Menschen getätigte Arbeit zu automatisieren und von Maschinen erledigen zu lassen, wie das heute beispielsweise schon Roboter im Fertigungsbereich tun. Die Qualität ist jedoch selbstverständlich eine andere. Es handelt sich nicht mehr nur um rein repetitive und relativ einfache Tätigkeiten, sondern um teilweise komplexe und sich immerwährend ändernde Aufgaben.

Nachfolgend sind einige mögliche Anwendungsbereiche aufgeführt, die vom Semantic Web profitieren können oder erst durch das Semantic Web möglich werden. Weiterhin werden noch zwei Beispiele für schon existierende Anwendungen gegeben, die RDF nutzen.

8.1 Composite Capabilities/Preference Profiles (CC/PP)

Eine Arbeitsgruppe des W3C arbeitet an einem Standard, der es erlaubt, Benutzerpräferenzen und Eigenschaften bzw. Einschränkungen von benutzter Hardware und Software zu beschreiben, so daß Anbieter ihre Dokumente entsprechend anpassen können.

So können ein Dokument beispielsweise in verschiedenen Sprachen und Formaten, oder Bilder in unterschiedlichen Auflösungen vorliegen, und je nach Profil wird dann das passendste ausgewählt. CC/PP wird entwickelt vor allem auch im Hinblick auf drahtlose Netzwerke, die oft nur geringe Bandbreiten haben und in denen es nötig ist, das Übertragungsvolumen klein zu halten.

Näheres zu CC/PP ist unter [10] zu finden.

8.2 Suchmaschinen

Die heutigen Suchmaschinen beruhen auf Wortvergleichen; bei einer Suchanfrage werden alle Dokumente zurückgeliefert, in denen die gesuchten Wörter vorkommen. Mit der sogenannten „Expertensuche“ kann man bei vielen Suchmaschinen auch die Suchwörter mit logischen Operatoren verknüpfen, nach zusammenhängenden Sätzen suchen oder sogar Wortentfernungen berücksichtigen.

Jedoch ist es dem Benutzer überlassen, die Suchanfrage so aufzubauen, daß die „richtigen“ Dokumente gefunden werden. Diese Aufgabe ist unter Umständen sehr schwierig. Oft ist man mehrerer Sprachen mächtig. Dann muß man, um optimale Ergebnisse zu erzielen, in jeder Sprache eine separate Anfrage stellen. Weiterhin muß man oft alle möglichen Synonyme durchprobieren und sich neue Suchwörter ausdenken, bis man irgendwann vielleicht eine brauchbare Ergebnismenge erhält.

Im Semantic Web von morgen ist aber durchaus folgendes denkbar: Man gibt einige Suchbegriffe ein, und klickt die Sprachen an, in denen Ergebnisse erscheinen dürfen. Wenn die Ergebnismenge zu groß ist, kann man die Anfrage weiter einschränken, allerdings nicht durch bloßes Hinzufügen von neuen Wörtern, sondern durch eine genauere Spezifizierung der ursprünglichen Wörter (z.B. „Koch“ soll ein Nachname und kein Beruf sein). Auch die Implementierung einer Relevanz-Feedback-Funktion wird vereinfacht, da die Suche nicht mehr rein lexikalisch, sondern semantisch eingeschränkt wird.

8.3 Agenten

Das Semantic Web ist Voraussetzung für den Einsatz von Agenten: Sie können sich selbständig im Internet bewegen und verschiedene Aufgaben erledigen. In [12] sind beispielsweise die Suche nach einem passenden Arzt oder das Finden von Terminen beschrieben.

Natürlich ist das Semantic Web nicht die alleinige Voraussetzung für effiziente Agenten, aber es vereinfacht ihre Entwicklung sehr stark, da das *Verstehen* von Informationen in den Hintergrund rückt und man sich auf das *Verarbeiten* der Information konzentrieren kann.

8.4 Mozilla

Der Open-Source-Webbrowser Mozilla [16] verwendet RDF in verschiedenen Komponenten wie dem *SmartBrowsing* und *Aurora*. Beim *SmartBrowsing* können Dritte, während der Benutzer auf einer Internetseite verweilt, weitere Informationen, wie beispielsweise die URLs thematisch verwandter Seiten, an den Benutzer senden. *Aurora* ist hingegen ein Projekt, um die verschiedenen Datenquellen wie Dateisystem, Internetseiten, Lesezeichen, Email-Ordner und so weiter zusammenzuführen und zu integrieren.

8.5 Das Open Directory Project

Das Open Directory Project [15] ist ein Katalog von Internetseiten, der hauptsächlich von freien Mitarbeitern verwaltet wird. Jeder Interessent kann sich anmelden und seinen Teil beitragen. Das ODP bietet sein komplettes Verzeichnis zum Download im RDF-Format an, und diese Daten werden mittlerweile von vielen anderen Sites verwendet.

9 Ausblick und Kritik

RDF wird mittlerweile immer öfter benutzt, und die Anzahl vorhandener Werkzeuge und Anwendungen dafür wächst ständig.

Nichtsdestotrotz ist das Semantic Web, so wie hier beschrieben, immer noch nicht mehr als eine Vision und noch nicht weit verbreitet. Zweifellos wird es sich jedoch weiterentwickeln, da die Menschheit schon seit ihrem Anbeginn versucht, sich die Arbeit zu erleichtern unter Zuhilfenahme von Werkzeugen. Doch um die Arbeit im Internet zu automatisieren, müssen die Maschinen das Internet zuerst „verstehen“.

Allerdings darf man das Semantic Web nicht als Allheilmittel sehen. Nicht alles wird möglich nur durch die Darstellung von Daten in Tripeln. Ein unvermeidbares Problem ist die in Abschnitt 7 angesprochene „Unendlichkeit“ des Internet. Weiterhin ist es keine triviale Aufgabe, Dokumente in RDF zu beschreiben, da dies mehr oder weniger auf das Herausfinden von Schlüsselwörtern reduzierbar ist, und das liefert selten alle gewünschten Resultate. Die Bearbeitung und Suche von Binärdaten wie Video, Audio und Bildern bleibt genauso unzufriedenstellend wie bisher.

Das Semantic Web und die hier vorgestellten Sprachen und Grundlagen werden jedoch in Zukunft sehr an Bedeutung gewinnen, denn das Internet wächst unaufhaltsam, und um die Möglichkeiten des Internet voll auszuschöpfen, braucht man Anwendungen, die sich selbständig im Internet bewegen können. Die Vision eines Agenten, dem man nur noch sagen muß, wo und wie lange man Urlaub machen will, und der dann selbständig den Flug bucht, ein Auto mietet und ein Hotelzimmer reserviert, und das alles beim jeweils günstigsten Anbieter, ist noch sehr weit von ihrer Verwirklichung entfernt. Doch das Semantic Web ist ein Grundstein dafür.

Literatur

- [1] RFC 2396: *Uniform Resource Identifiers (URI): Generic Syntax*,
<http://www.isi.edu/in-notes/rfc2396.txt>
- [2] *Ontologien*,
<http://www.pi.informatik.tu-darmstadt.de/milan/right/Ontology.html>
- [3] *Resource Description Framework (RDF) Model and Syntax Specification*,
<http://www.w3.org/TR/1999/PR-rdf-syntax-19990105>
- [4] *RDF/XML Syntax Specification (Revised)*,
<http://www.w3.org/TR/2001/WD-rdf-syntax-grammar-20011218/>
- [5] *Resource Description Framework (RDF) Schema Specification 1.0*,
<http://www.w3.org/TR/2000/CR-rdf-schema-20000327>
- [6] *Dublin Core Metadata Initiative*,
<http://dublincore.org>
- [7] *DAML+OIL (March 2001) Reference Description*,
<http://www.w3.org/TR/daml+oil-reference>
- [8] *The OIL Homepage*,
<http://www.ontoknowledge.org/oil/>
- [9] *World Wide Web Consortium*,
<http://www.w3.org>
- [10] *CC/PP Working Group*,
<http://www.w3.org/Mobile/CCPP>
- [11] *Extensible Markup Language (XML) 1.0*,
<http://www.w3.org/TR/2000/REC-xml-20001006>
- [12] *The Semantic Web*, Tim Berners-Lee, James Hendlers, Ora Lassila,
<http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>
- [13] *Towards the Semantic Web: knowledge representation in a dynamic, distributed environment*, Dissertation von Jeffrey Douglas Heflin
- [14] *Die Rolle von Verschlüsselungsverfahren, digitalen Signaturen, Zertifikation, etc. im E-Commerce*, Arno Hornberger
<http://www.dbis.informatik.uni-kl.de/courses/seminar/SS2001/ausarbeitung9.pdf>
- [15] *Open Directory Project*,
<http://dmoz.org>
- [16] *mozilla.org*,
<http://mozilla.org>
- [17] *Logisch schließende Systeme*, Boris Stumm
<http://www.wagr.informatik.uni-kl.de/bendeck/proseminar/ki/ws99/Cap10.zip>