Bedeutungsvolle Klammern

XML und das Problem der Semantik

Eckhardt Schön
Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Institut für Praktische Informatik und Medieninformatik
Fachgebiet Telematik

Mit Hilfe von XML ist es möglich, Dokumente nach ihrem Inhalt zu strukturieren. Dazu können beliebig benannte Tags benutzt werden. In Hinblick auf das erstrebte Semantic Web wird in dem Artikel diskutiert, welche Möglichkeiten XML zu Kodierung von Semantik bietet.

---

1. Einleitung

Die eXtensible Markup Language (XML) entwickelt sich immer mehr zur Universalsprache für die Speicherung von Dokumenten und für den Datenaustausch. XML-Dokumente sind einerseits viel besser strukturierbar als reine ASCII-Texte und andererseits nicht an eine bestimmte Firma oder Software gebunden, weshalb sie universell nutzbar sind. Die Strukturen in Texten und Daten sind in der Regel kein Selbstzweck, sondern dienen der Abgrenzung von inhaltlich zusammengehörigen Teilen gegenüber anderen. Die Tags in XML-Dokumenten eigenen sich dazu in besonderer Weise, da sie sehr flexibel eingesetzt werden können. Man spricht deshalb auch oft von "semantische Tags" oder von "semantische Markup". Daran ist prinzipiell richtig, dass es möglich ist, Texte und Daten mit XML-Tags zu versehen, die eine eigene Bedeutung besitzen. Diese können dem Dokument semantische Zusatzinformationen verleihen. Es gibt dabei allerdings keinen Automatismus.
Dieser Artikel hat zum Ziel, das Problem der Semantik in XML-Dokumenten genauer zu beleuchten. Es wird zunächst der semantische Gehalt von reinem XML betrachtet. Dabei wird deutlich, dass auch XML sehr schnell an Grenzen stößt, wenn es um die maschinenverständliche Kodierung von Semantik geht. Deshalb werden mit dem Resource Description Framework (RDF) und den Topic Maps Technologien vorgestellt, die diesbezüglich weit größere Potenzen besitzen. Beide haben die Unterstützung des Semantic Web zum Ziel. Es handelt sich dabei um ein Konzept des World Wide Web Consortiums (W3C) zur Weiterentwicklung des WWW. Durch die Integration von semantischen Informationen sollen die Ressourcen des Internets besser vernetzt, die Suche und Navigation weit zielgerichteter möglich und ganz neue Dienste realisierbar werden. Vieles davon ist heute zwar noch Vision, aber XML und die darauf basierenden Technologien ebnen den Weg dahin.

2. Das Problem der Semantik

In diesem Artikel geht es um die Strukturierung und den Austausch von Nachrichten. Nachrichten sind mehr als Daten, denn es geht nicht nur um die syntaktisch korrekte Anordnung von Zeichen, sondern die Inhalte von Nachrichten haben eine Beziehung zur

realen Welt. Sie bilden einen kleinen Ausschnitt davon ab. Zu Informationen werden Nachrichten dadurch, dass Sie eine Bedeutung für den Nachrichtenempfänger (Person oder auch Maschine) haben. Oft wird das stillschweigend vorausgesetzt, was eine Gleichsetzung der Begriffe Nachricht und Information zur Folge hat. Auch in diesem Artikel wird die Unterscheidung nicht immer konsequent getroffen.
Nähern wir uns nun dem Begriff der Semantik zunächst von einem allgemeinen Verständnis her. Wenn Nachrichten die reale Welt beschreiben, dann geschieht das auf einer Metaebene. Das kann die natürliche Sprache sein, eine Programmiersprache, ein Datenmodell oder eine Graphik. Man spricht allgemein von einer Repräsentation der realen Welt. Die Semantik als Wissenschaft ist die Theorie vom Verhältnis zwischen der Repräsentation und dem was repräsentiert wird. Im Bereich der Informatik haben sich mehrere Disziplinen entwickelt, die sich mit semantischen Problemen beschäftigen, die Computerlinguistik, die Wissenspräsentation oder auch der Bereich der Expertensysteme.
Man muss aber gar nicht unbedingt Spezialdisziplinen betrachten. Die Frage des Verhältnisses von Repräsentation und realer Welt stellt sich bereits überall dort, wo Daten und Prozesse modelliert werden.

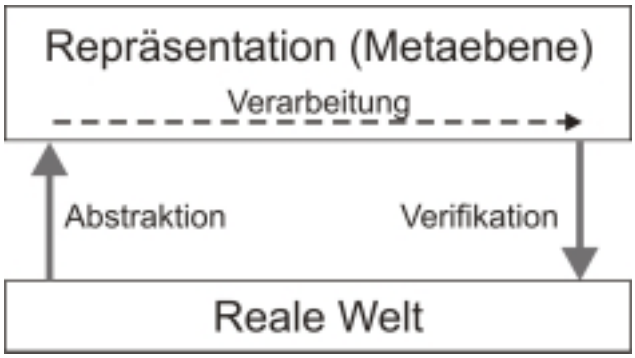

Abb. 1: Repräsentation und reale Welt

Um eine Repräsentation der realen Welt zu finden, muss man abstrahieren, muss sich auf Wesentliches konzentrieren und unwichtige Aspekte vernachlässigen. Wenn man das geeignet gemacht hat, kann man auf der Metaebene Schlussfolgerungen ziehen, die man durch Verifikation bestätigen kann (Abb. 1). Für die Verarbeitung auf der Metaebene braucht man formalisierte Konstrukte, z.B. Sprachen und Datenstrukturen. Diese Notwendigkeit der Formalisierung besteht insbesondere, wenn die Datenverarbeitung mit Hilfe eines Computers durchgeführt werden soll. Je stärker die Nachrichten strukturiert und mit (semantischen) Zusatzinformationen versehen sind, desto besser lassen sie sich von einem Computer verarbeiten.

wenig strukturierte Daten	**stark strukturierte Daten**
z.B. freier Text	z.B. Datenbank
starke semantische Variabilität	geringe semantische Variabilität
schwer zu formalisieren	gut untersucht, z.B. Entity-Relationship-Diagramme

Betrachten wir als einfaches Beispiel den Satz:

Paris ist die Hauptstadt von Frankreich.

Dieser Satz ist zwar für einen Menschen verständlich, weil er weiß, dass Paris ein Stadt, Frankreich ein Land und was eine Hauptstadt ist. Für einen Computer handelt es sich jedoch

um eine unstrukturierte Aneinanderreihung von Wörtern. Um die Aussage des Satzes zu erfassen und maschinell weiterverarbeiten zu können, muss eine semantische Analyse mit Hilfe von Methoden der Computerlinguistik vorgenommen werden. Der Satz müsste in seine Bestandteile (semantische Entitäten) zerlegt und die Relationen zwischen diesen Entitäten erkannt werden, um die Aussage schließlich auf einer höheren Metaebene verarbeiten und mit anderen Statements verknüpfen zu können. Die Abbildung 2 zeigt eine mögliche graphische Veranschaulichung der semantischen Entitäten (nach [HELB2001]). Die Darstellung zeigt Knoten und Kanten, die semantischen Gehalt tragen. Man kann daraus semantische Netze bilden, die auch komplexe Sachverhalte abbilden können. Der Aufwand ist schon für einfache Sätze beträchtlich. Von einem Verstehen normaler Texte sind Computer heute noch weit entfernt.

Abb. 2: Semantische Entitäten

Ein anderer Weg zur Kodierung der obigen Aussage besteht darin, sie in einer (Datenbank-) Tabelle abzulegen.

Land	Hauptstadt
Frankreich	Paris

Vorausgesetzt, dass in der Zeile die zu dem Land gehörige Hauptstadt steht, hat man die gewünschte Aussage in recht einfacher und maschinenverständlicher Weise kodiert.
Allerdings muss man dem Computer auch beibringen, was er unter einem Land und unter einer Hauptstadt zu verstehen hat. Das ist der eigentliche semantische Gehalt dieser Tabelle. Daten derartig zu strukturieren, ist deshalb nur sinnvoll, wenn viele Datensätze mit gleichem semantischen Gehalt vorliegen.
Vielfach ist es jedoch so, dass man eine Menge von verschiedenartigen Objekten und Beziehungen der Realität hat, die auf einer Metaebene durch Entities und Relationen repräsentiert werden sollen, um sie von Computern speichern, verarbeiten und als Nachrichten austauschen lassen zu können. Hier bietet XML Ansätze zu einer neuen Art der semantischen Kodierung zu gelangen.

3. XML und "semantisches" Markup

Die eXtensible Markup Language (XML) eignet sich für eine "semistrukturierte" Datenmodellierung. Diese ermöglicht es, Nachrichten zu strukturieren, ohne sich an starre Vorgaben halten zu müssen. XML kann also Datenstrukturen realisieren, die sich irgendwo zwischen einem freien Text und einer Datenbankstruktur befinden, wobei diese beiden Grenzfälle ebenfalls modelliert werden können.
Für die Modellierung werden Markups (auch Tags genannt) benutzt. Da diese bei XML frei gewählte Namen tragen und (fast) beliebig verschachtelt werden können, lassen sich Dokumente und Daten gut strukturieren. Die Tags können dazu benutzt werden, um Dokumententeile semantisch zu charakterisieren. Damit lässt sich eine flexible Repräsentation der Realität erstellen. Man spricht deshalb oft von "semantischem" Markup.
Das oben angeführte Beispiel könnte in XML folgendermaßen aussehen:

```
<Aussage>
   <Stadt>Paris</Stadt>
   ist die Hauptstadt von
   <Land>Frankreich</Land>
   .
</Aussage>
```

Paris und Frankreich lassen sich zwar gut charakterisieren, indem sie durch die Tags `<Stadt>` und `<Land>` ausgezeichnet werden, aber schon die Relation zwischen beiden ist nicht so einfach durch XML-Tags semantisch zu beschreiben. Das Problem liegt aber noch tiefer. Den semantischen Gehalt der verwendeten Tags kann zwar ein (deutsch sprechender) Mensch erkennen, für einen Computer sind `<Stadt>` und `<Land>` bisher jedoch lediglich Zeichenketten.

Ein weiteres Problem ist, dass innerhalb der `<Aussage>`-Tags sowohl strukturierte Nachrichten wie `<Stadt>`Paris`</Stadt>` als auch unstrukturierter Text auftreten. Man bezeichnet dies als "mixed content".

Die Semantik eines solchen Textes hat (etwas vereinfacht) zwei Aspekte:

lexikalische Semantik	kompositionelle Semantik
Definition und Bedeutung der Tags	komplexe Bedeutungen in Form von syntaktischen Konstrukten

Möglichkeiten zur Definition von Tags und Festlegung von syntaktischen Konstrukten bieten Document Type Definitions (DTD) und XML Schema [XMLS2001]. Letztere sind dabei, die DTDs als Mittel zur Festlegung von Strukturen in XML-Dokumenten abzulösen.

XML Schema legen fest, welche Tag-Name in einem XML-Dokument erlaubt sind, mit welchen Attributen sie versehen sein können, in welcher Reihenfolge sie auftreten und wie sie verschachtelt werden dürfen. Mit ihrer Hilfe werden Datentypen festgelegt und können neue definiert werden. In einer XML-Schema-Definition werden allerdings keine Festlegungen zur Semantik getroffen. Beim Vergleich mit der natürlichen Sprache entspricht ein Schema dem Wörterbuch einschließlich einer formaler Grammatik. Es ist aber kein Lexikon, welches die Bedeutung der Worte erklärt.

Der semantische Gehalt eines XML-Dokumentes wird von einem menschlichen Autor oder von einer Applikation festgelegt. Er kann sehr unterschiedlich komplex sein. In der Regel ist es das Ziel, dass eine andere Applikation die XML-Nachricht automatisch verarbeiten kann. Sie muss dazu die Tags "verstehen". Dieses Verstehen bedeutet, dass sie die Tags und ihre Attribute kennt und weiß, wie sie damit umzugehen hat. Die semantischen Überlegungen werden dabei vom Entwickler der Applikation in Form von Datentypen und Programmlogik eingebracht.

Die syntaktischen Festlegungen eines Schemas sind trotzdem für den Nachrichtenaustausch auch in Hinblick auf die Semantik wichtig. Durch die Festlegung der erlaubten Tags und syntaktischen Konstrukte reduziert sich der Aufwand für die Erstellung von Applikationen. Man hat die Möglichkeit sicherzustellen, dass die Nachrichten nicht nur korrekt verarbeitet, sondern dass sie auch richtig interpretiert werden können.

Je genauer ein Schema die Strukturen festlegt, desto leichter lassen sich die darauf fußenden Nachrichten von einem Programm verarbeiten. Unser Beispiel könnte auch in folgender Weise mittels XML kodiert werden:

```
<Aussage>
   <Land>Frankreich</Land>
   <Hauptstadt>Paris</Hauptstadt>
```

```
  </Aussage>
```
Dieses XML-Fragment lehnt sich an die obige Tabelle an. Besser und offen für Erweiterungen ist allerdings folgende Struktur:
```
  <Land>
     <Name>Frankreich</Name>
     <Hauptstadt>Paris</Hauptstadt>
  </Land>
```
Hier wurde die semantische Hierarchie, dass die Hauptstadt logisch dem Land untergeordnet ist, auch auf eine syntaktische Enthaltensein-Relation abgebildet. Derartige XML-Dateien, die eng strukturiert sind und keinen freien Text auf der gleichen Hierarchiestufe mit Tags enthalten, werden oft als XML-Daten bezeichnet.
Die folgende Tabelle zeigt die wichtigsten Unterschiede zwischen XML-Daten und -Dokumenten. Es muss allerdings betont werden, dass diese Trennung rein aus der Praxis erwachsen ist. Die Spezifikation [XML2000] kennt diesen Unterschied nicht, und alle Werkzeuge sollten beide Spielarten gleichermaßen behandeln können. Trotzdem hat sich diese Unterscheidung in Hinblick auf die Anwendungsentwicklung bewährt und ist auch bei der Diskussion von semantischen Problemen nützlich.

XML-Daten	**XML-Dokumente**
stark strukturiert	variablere Strukturen
kein "mixed content" erlaubt	"mixed content" möglich
Reihenfolge von Elementen auf einer Hierarchieebene meist unwichtig	Reihenfolge von Elementen auf einer Hierarchieebene wichtig
genutzt als Datenbank-Ersatz, für den Nachrichtenaustausch zwischen Applikationen und für formal-logische Konstrukte	Anwendung u.a. bei der Dokumentenverarbeitung (Redaktionssysteme) und für das WWW (XHTML)
Semantik der Tags genau festgelegt	Semantik oft nur schwer zugänglich

XML-Daten sind sehr gut geeignet, um Nachrichten zwischen Applikationen auszutauschen. Man legt dann nicht nur die Syntax mittels XML Schema genau fest, sondern definiert auch den semantischen Gehalt von Tags und komplexeren Konstrukten sehr genau. Folgendes Beispiel ist ein kleiner Ausschnitt aus der Katalog-Datei eines Reiseanbieters im XML-Format:
```
  <Reise id="0815">
     <Typ>Städtereise</Typ>
     <Reiseziel>Paris</Reiseziel>
     <Abreise>
        <Ort>Erfurt</Ort>
        <Datum>2002-05-04</Datum>
     </Abreise>
     <Aufenthaltsdauer Einheit="d">6</Aufenthaltsdauer>
     <Hotel>Grand Opera</Hotel>
     <Preis Waehrung="Euro">699</Preis>
```

```
    . . .
  </Reise>
```
Bietet ein Veranstalter seine Reiseinformationen in dieser Form an, kann auch ein unabhängiges Reisebüro diese Informationen leicht in sein eigenes Angebot integrieren. Die Bedeutung (Semantik) der Tags ist klar bzw. es existiert eine genaue Beschreibung des Anbieters. Diese Beschreibung ist aber sicher nicht maschinenlesbar, womit wir wieder an Grenzen stoßen.
Ein zweiter Anbieter verwendet vermutlich andere Tag-Namen und Datenstrukturen, oder auch die gleichen Tag-Namen mit anderer Bedeutung. Will man XML-Daten aus mehreren Quellen gemeinsam verarbeiten, lässt sich zwar mit Hilfe der XML Namespaces [XML2000] die Eindeutigkeit der Tags erreichen, über semantische Unterschiede oder Ähnlichkeiten ist damit jedoch keine Aussage getroffen.
Um XML-Daten in Applikationen zu nutzen, muss man die XML-Datei parsen. Damit kann man die Informationen auslesen und sie verarbeiten oder weitergeben. Die verarbeitenden Komponente kann ein spezielles Programm sein, dass genau auf die Nachrichtenstruktur abgestimmt ist und deshalb die XML-Daten semantisch korrekt verarbeiten kann. Oft wünscht man sich jedoch eine flexiblere Lösung. Für eine Vorverarbeitung oder Änderung der Datenstruktur bietet sich die Transformation mit Hilfe von eXtensible Style Language (XSLT) an.

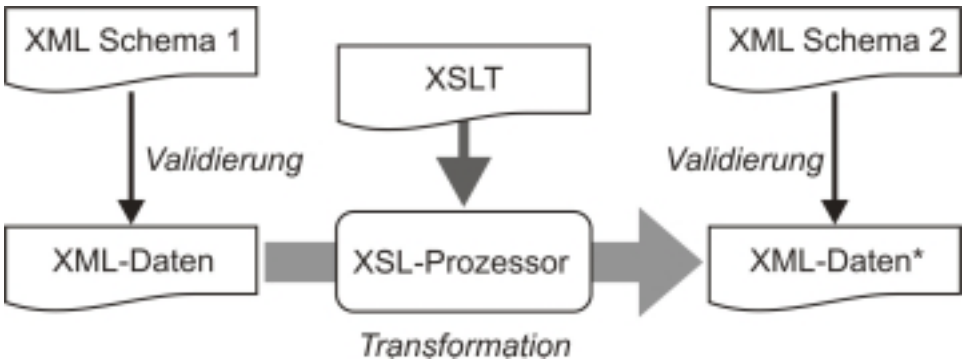

Abb. 3: Arbeitsweise eines XSLT-Prozessors

Mit Hilfe einer XSLT-Datei, kann man aus einer XML-Datenstruktur eine andere erzeugen. Auf diese Weise könnte man zum Beispiel die Katalogstruktur von Drittanbietern auf die eigene Datenstruktur abbilden. Durch Validierung des Transformationsergebnisses gegenüber dem eigenen Schema ("XML Schema 2" in Abb. 3) kann man dessen syntaktische Richtigkeit überprüfen.
Was geschieht aber mit der Semantik der Tags bei einer Transformation? Sie wird in der Regel abnehmen. Ganz deutlich wird das bei der verbreiteten Erzeugung von HTML aus XML mit Hilfe von XSLT.

```
  <Aussage>
     <Stadt>Paris</Stadt>
     ist die Hauptstadt von
     <Land>Frankreich</Land>
     .
  </Aussage>
```

Dieses XML-Fragment lässt sich zum Beispiel mit Hilfe von folgendem Style Sheet (Ausschnitt) in HTML transformieren. "xsl:" kennzeichnet dabei den Namensraum der XSL-Transformationen.

```
  <xsl:template match="Aussage">
    <p>
      <xsl:apply-templates/>
```

```
    </p>
  </xsl:template>
  <xsl:template match="Stadt">
    <em>
      <xsl:value-of select="."/>
    </em>
  </xsl:template>
  <xsl:template match="Land">
    <em>
      <xsl:value-of select="."/>
    </em>
  </xsl:template>
```

Das Ergebnis ist folgender HTML-Abschnitt:

```
  <p><em>Paris</em> ist die Hauptstadt von
  <em>Frankreich</em>.</p>
```

In diesem Satz sind die Worte "Paris" und "Frankreich" zwar hervorgehoben, die semantischen Informationen darüber sind allerdings verloren gegangen.
Diese Reduktion an semantischem Gehalt kann gewollt sein, z.B. für reine Präsentationszwecke, verhindert allerdings eine automatische Weiterverarbeitung von Daten bzw. Dokumenten.
Kommen wir noch einmal auf das Beispiel unseres Reisebüros zurück. Will man dort die XML-Angebotsdaten verschiedener Reiseanbieter in den eigenen Datenbestand integrieren, benötigt man vermutlich für jede Angebotsdatei ein extra Style Sheet, um die Informationen in die eigene XML-Datenstruktur zu integrieren. Man muss dabei sehr genau auf die Semantik der Tags und Konstrukte achten. `<Typ>` kann bei einzelnen Anbietern etwas ganz Verschiedenes bedeuten oder sich auch nur um Nuancen unterscheiden. Man muss die XSL-Transformationen deshalb mit sehr kritischem Blick entwickeln. Eine automatische Auswertung der Semantik ist auf dieser Ebene noch nicht möglich. Dazu sind erweiterte Ansätze nötig, wie sie im folgenden Abschnitt erläutert werden.

4. Resource Description Framework (RDF)

Das Resource Description Framework (RDF) wurde vom World Wide Web Consortium (W3C) geschaffen, um Metadaten so zu kodieren, dass sie nicht nur maschinenlesbar sondern maschinenverständlich abgelegt und ausgetauscht werden können [RDF1999]. Das RDF greift dabei auf Erkenntnisse aus dem Bereich der Wissenspräsentation zurück.
Eine Ressource ist im Sinne des RDF alles, was durch einen Uniform Resource Identifier (URI) beschrieben werden kann: eine Web-Seite, ein Buch, eine Person, ein Begriff oder auch eine Eigenschaft. Das grundlegende Modell ist denkbar einfach. Es besteht aus einem Tripel, das i.allg. zwei Knoten (Ressourcen) durch einen gerichteten Graphen verbindet, welcher selbst auch wieder eine Ressource ist (Abb. 4).

Abb. 4: RDF-Tripel

Man findet für ein solches Tripel auch die Bezeichnungen Subjekt (Ressource_1), Prädikat (Eigenschaft_1) und Objekt (Ressource_2) in Analogie zu natürlichsprachigen Aussagen. Das Objekt ist der Eigenschaftswert und muss nicht unbedingt selbst eine Ressource sein. Er kann auch ein Literal sein und wird dann in einem Rechteck dargestellt.

Da eine Ressource mit beliebig vielen anderen verknüpft werden kann, können sehr komplexe Netze aus Metadaten entstehen. Das schon oft benutzte Beispiel ließe sich in folgender Form darstellen:

Abb. 5: RDF-Beispiel

In diesem Beispiel ist "Land_01" eine Ressource, die ein konkretes Land meint, ohne selbst Informationen darüber zu besitzen. Alle Nachrichten über diesen Knoten (der auch unbenannt sein kann) liefern seine Eigenschaften. Im vorliegenden Fall gibt es nur die Namenseigenschaft, die durch folgendes Tripel ausgedrückt werden kann: {Name, Land_01, "Frankreich"}. Für den Knoten "Stadt_01" gilt Ähnliches. Er besitzt allerdings noch eine zweite Eigenschaft, nämlich Hauptstadt zu sein.
Vielleicht ist bei manchem Leser inzwischen die Frage aufgetreten, was die Ausführungen dieses Abschnittes mit der Semantik von XML-Dokumenten zu tun haben. Erstens lassen sich XML-Dateien (wie alle Ressourcen) mit Hilfe des Resource Description Framework beschreiben, und zweitens gibt es eine XML-Syntax für RDF.
Wichtiger ist allerdings zunächst die Frage, wie nun wirklich die maschinenverständliche Beschreibung von Ressourcen realisiert werden kann. Bisher wurden Bezeichner wie "Land_01" oder "ist Hauptstadt von" benutzt, die sicher kein Computer versteht. Sie waren auch nur Platzhalter für die Uniform Resource Identifier (URI) der Ressourcen. Man benötigt also die URIs aller beteiligten Ressourcen. Das ist im Fall einer Webseite einfach, man verwendet den Uniform Resource Locator (URL). Bei anderen Ressourcen muss man solche weltweit eindeutigen Identifier u.U. selbst definieren. Das W3C gibt dazu den Weg vor. Während die Modell-und-Syntax-Spezifikation [RDF1999] die grundlegende Arbeitsweise von RDF festlegt, beschreibt die RDF-Schema-Spezifikation [RDFS1999] wie neue Begriffe und Zusammenhänge definiert werden können. Leider kann auf diese Möglichkeiten nur im Rahmen unseres Beispiels eingegangen werden. Einen guten Einstieg in die Grundlagen bietet das Tutorial von P.-A. CHAMPIN [CHAM2001].

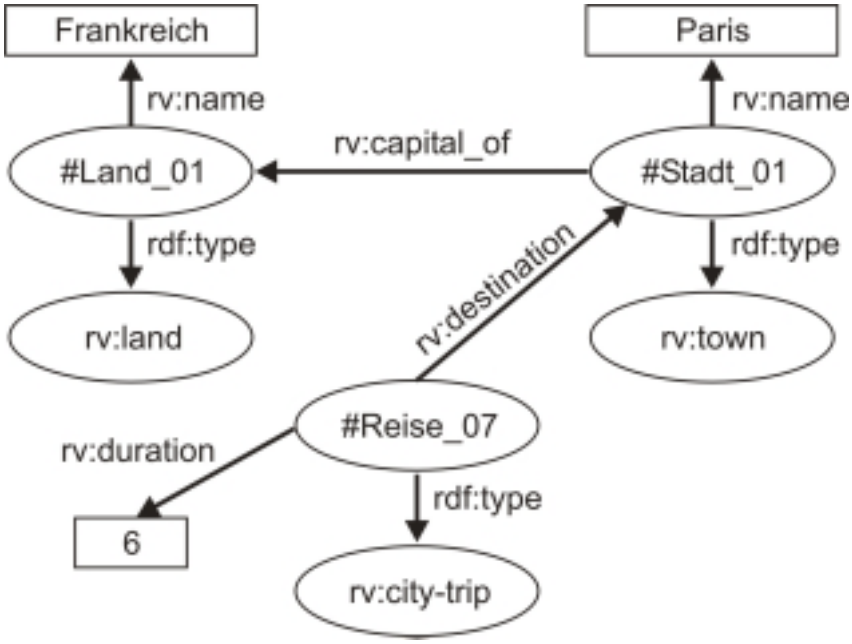

Abb. 6: Erweitertes RDF-Beispiel

Die Abbildung 6 zeigt den RDF-Graphen in etwas erweiterte Form. Im oberen Teil wurden die bisherigen Bezeichner für die Eigenschaften durch URIs ersetzt. Nur auf den ersten Blick scheint die Änderung von "Name" zu "rv:name" nicht viel zu bedeuten. In Wirklichkeit bezeichnet "rv:" den Namensraum eines fiktiven Zusammenschlusses von Reiseveranstaltern. Diese Organisation könnte unter ihrer URL Knoten und Eigenschaften aus der Domäne der Reiseinformationen definieren. Dazu werden Methoden benutzt, wie sie in RDF-Schema spezifiziert werden. Für unser Beispiel sollten dort Knoten definiert sein, die semantisch ein Land (rv:land), eine Stadt (rv:town) und eine Städtereise (rv:city-trip) beschreiben. Außerdem müssen dort mindestens die Eigenschaften rv:name, rv:capital_of, rv:destination und rv:duration definiert sein. Es lassen sich in einem RDF-Graphen die Ressourcen aus unterschiedlichen Namensräumen gemeinsam verwenden, wie die Benutzung von rdf:type zeigt. Die Knoten #Land_01, #Stadt_01 und #Reise_07 müssen nicht extern definiert werden, da sie nur temporär bestehen und vollständige durch ihre Eigenschaften bestimmt sind.
Bisher wurden die semantischen Zusammenhänge zwischen den Ressourcen lediglich graphisch beschrieben, wie schon erwähnt, gibt es jedoch auch eine Serialisierung der RDF-Graphen in Form von XML. Für obiges Beispiel hat die RDF-Datei folgende Gestalt.

```
<?xml version="1.0"?>
 <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:rv="http://www.reise_veranstalter.de/rdf-schema#">
  <rdf:Description about="#Land_01">
    <rdf:type resource="rv:land"/>
    <rv:name>Frankreich</rv:name>
  </rdf:Description>
  <rdf:Description about="#Stadt_01">
    <rdf:type resource="rv:town"/>
    <rv:name>Paris</rv:name>
    <rv:capital_of resource="#Land_01"/>
  </rdf:Description>
  <rdf:Description about="#Reise_07">
    <rdf:type resource="rv:city-trip"/>
    <rv:duration>6</rv:duration>
    <rv:destination resource="#Stadt_01"/>
  </rdf:Description></rdf:RDF>
```

Nach der XML-Deklaration folgt das Wurzelelement <RDF>. Mit seiner Hilfe werden drei Namensräume vereinbart, zwei Standard-Namensräume des W3C und der des fiktiven Reiseveranstalters. Damit können nun die weiteren Tags qualifiziert werden. Struktur und Syntax des RDF-Kodes sind relativ einfach. Es besteht in diesem Fall aus drei Beschreibungen (Description) von Knoten durch ihre Eigenschaften. Der Quelltext dürfte weitgehend selbsterklärend sein, wenn man die Abbildung 6 zu Hilfe nimmt. Genauer kann auf die RDF-Syntax, die durchaus komplexere Konstrukte kennt, in diesem Artikel nicht eingegangen werden (siehe [RDF1999]). Ein geeigneter RDF-Viewer kann aus obigem Quelltext genau den Graphen aus Abbildung 6 erzeugen, oder zumindest einen topologisch äquivalenten.
Wenden wir uns nun wieder direkt dem Problem der Semantik zu. Es ist hoffentlich einsichtig, dass mit Hilfe des RDF semantische Netzwerke, wie sie in Abschnitt 2 beschrieben wurden, aufgespannt werden können. Der große Vorteil ist nun, dass es sich bei den Knoten und Kanten um weltweit eindeutige und adressierbare Ressourcen handelt. Für diese Ressourcen kann mit Hilfe der RDF-Schema eine maschinenverständliche Beschreibung erzeugt werden.
In unserem kleinen Beispiel könnte ein Software-Agent selbstständig alle Informationen über eine bestimmte Reise (#Reise_07) ermitteln. Er würde nicht nur feststellen, dass es sich um eine 6-tägige Städtereise nach Paris handelt, sondern auch dass das Reiseland Frankreich ist. Im konkreten Fall ist das zwar trivial, soll aber die Möglichkeiten andeuten. Bei einem ausgebauten semantischen Netz auf der Basis von RDF, könnte der Agent weitere Informationen zum Reiseland (z.B. Passbestimmungen und Gesundheitstipps) oder zur Stadt Paris (Veranstaltungen, Verkehrsinformationen usw.) automatisch zusammentragen. Neben diesen direkten semantischen Verbindungen ist es prinzipiell auch möglich, auf einer noch höheren Ebene ontologische Regeln zu formulieren, die auch implizite Verknüpfungen erkennen lassen. Unter Ontologie wird hier ein maschinenlesbarer Satz von Definitionen verstanden, die ein System von (meist abstrakten) Klassen und Subklassen bilden. Wichtig sind die vielfältigen Beziehungen untereinander. Die Beschreibung solcher struktureller Ontologien ist allerdings nicht Gegenstand des RDF. Man ist unter Verwendung der RDF-Schema-Sprache jedoch prinzipiell in der Lage, eigene ontologische Regeln zu modellieren. Es ist zwar denkbar, aber nicht sinnvoll, dass jeder mit Hilfe von RDF-Schema seine eigenen semantischen Entitäten entwirft. Für wichtige Anwendungsdomänen sollten international praktikable Ressourcenbeschreibungen erstellt werden, um den Austausch von Metadaten zu unterstützen. In einigen Bereichen gibt es dazu bereits Ansätze. So standardisiert die Dublin Core Metadata Initiative [DC2002] die Beschreibung von Dokumenten in digitalen Bibliotheken unter anderem mit Hilfe des RDF.
Die RDF-Informationen können Bestandteil eines Dokumentes (vor allem bei XML- und HTML-Dokumenten) sein, oder getrennt vom eigentlichen Dokument mit eindeutigem Bezug zu ihm verwaltet werden. XML-Dateien eignen sich besonders gut zur Beschreibung durch das RDF, da sie nicht nur als Ganzes betrachtet werden können, sondern mit Hilfe von XPointer [XPOI2001] auch Dokumententeile gezielt adressiert und mit Metadaten versehen werden können.

5. Topic Maps

Im Gegensatz zur ausführlichen Darstellung im letzten Abschnitt soll auf die Topic Maps hier nur kurz eingegangen werden.
Topic Maps sind ebenfalls aus dem Konzept der semantischen Netze entwickelt worden. Das Ziel dieser Entwicklung war es, die Navigation, Indexierung und das Retrieval in umfangreichen Datenbeständen zu unterstützen. Das Topic Maps wurde unter Leitung der

International Organization for Standardization (ISO) entwickelt. Der Standard ISO/IEC 13250 beschreibt das Modell und die Austauschsyntax. Ein vorrangiges Ziel von Topic Maps ist die Unterstützung bei der Visualisierung von semantischen Zusammenhängen. Dazu werden die Informationsressourcen (Topics) über Assoziationen (Beziehungen zwischen den Topics) verknüpft. Es gibt Methoden zur Filterung und zum Clustering vom Informationen aufgrund ihres semantischen Gehaltes.
Das Herangehen ist trotz einiger Unterschiede nicht unähnlich dem des RDF. Die Entwicklung beider Technologien startet zwar aus verschiedenen Richtungen und wurde von zwei unterschiedlichen Organisationen (ISO und W3C) vorangetrieben, heute laufen beide Konzepte bei dem Ziel der Unterstützung des Semantic Web jedoch aufeinander zu. Inzwischen konnte G. MOORE zeigen, dass beide Modelle auf der grundlegenden Ebene interoperabel sind und sich gegenseitig modellieren können [MOOR2001]. Das schafft die wissenschaftliche Basis für die Weiterentwicklung und wünschenswerte Integration beider Konzepte, zumal MOORE einer der Herausgeber der XTM-Spezifikation ist.
Denn auch Topic Maps besitzen eine XML-Darstellung, XML Topic Maps (XTM) [XTM2001], die ebenfalls der Serialisierung semantische Netze dient, sich allerdings in der Syntax vom RDF unterscheidet.
Da eines der Ziele von Topic Maps die Visualisierung von Zusammenhängen zwischen Topics ist, bietet sich hierfür die Benutzung der XML-basierten Scalable Vector Graphics (SVG) an. Dazu gibt es eine Reihe von Arbeiten. Gerade in diesem Bereich der Präsentation zeigen sich die unterschiedlichen Intensionen von Topic Maps und RDF. Während das RDF mehr im Hintergrund wirkt, um Applikationen semantische Informationen zur Verfügung zu stellen, sollen Topic Maps auch eine Schnittstelle zum menschlichen Benutzer des Netzes bieten. Sie sollen ihm eine semantisch basierte Navigation ermöglichen.

6. Semantic Web

Das Semantic Web ist heute noch eine Vision, die jedoch sehr engagiert vom W3C-Direktor TIM BERNERS-LEE verfolgt wird [BERN2001]. Es geht darum, das heutige World Wide Web so zu erweitern, dass es seine Inhalte nicht nur menschlichen Benutzern präsentieren kann, sondern auch von Maschinen und Anwendungen automatisiert benutzt werden kann. Da die Menge der Nachrichten immer schneller wächst, ist sie für einen menschlichen Nutzer nicht mehr überschaubar. Die heutigen Hilfsmittel wie Suchmaschinen und Kataloge sind bei der Zusammenstellung und Aufbereitung der Informationen bereits überfordert, so dass neue Wege gegangen werden müssen, um den Informationsinfarkt zu verhindern.
Die Idee des Semantic Web ist nun, auf einer Metaebenen die vorhandenen Informationsquellen in einem semantischen Netz zu verknüpfen. Es soll damit eine verteilte Wissenspräsentation entstehen, da das Web schließlich ebenfalls dezentral organisiert ist.

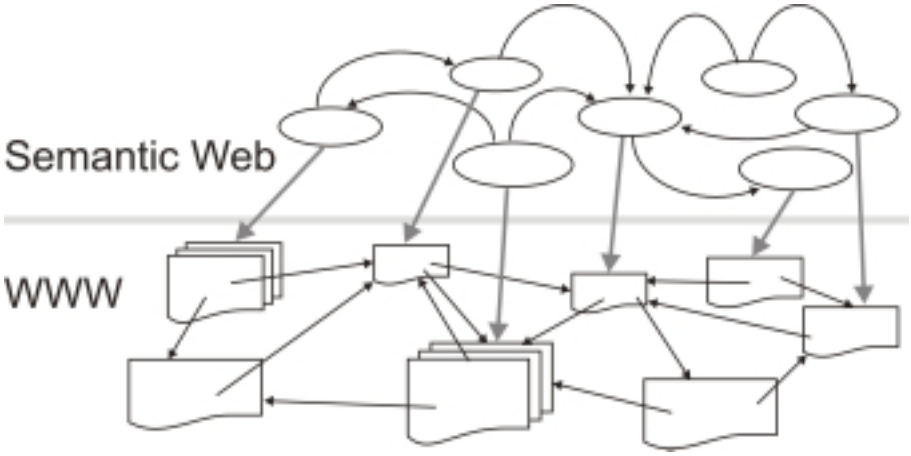

Abb. 7: Semantic Web

Die Abbildung 7 veranschaulicht dieses Konzept. Im unteren Teil ist das bisherige WWW mit seinen durch Hyperlinks verbundenen Ressourcen (meist HTML-Dateien) angedeutet. Auf der Ebene darüber befindet sich ein semantische Netz. Die Ellipsen repräsentieren die semantischen Knoten und die gerichteten Bögen dazwischen sind die Eigenschaften (RDF) oder Assoziationen (Topic Maps). Im Unterschied zu den Hyperlinks im unteren Teil, die willkürlich von den Webautoren gesetzt werden, basieren die Verbindungen im oberen Bildteil auf semantischen Analysen und Modellen. Deshalb ist auf dieser Ebene eine semantisch basierte Navigation durch menschliche Nutzer aber vor allem auch durch Software-Agenten möglich. Die Knoten sind zum Teil Repräsentanten von Dokumenten im WWW, die nach einer erfolgreichen semantischen Suche direkt angesprungen werden können. Das Retrieval und die Navigation verlagert sich auf die Ebene des Semantic Web, ohne dass die traditionellen Möglichkeiten auf der Basis der Hyperlinks im WWW verloren gingen.
Die Technologien zur Verwirklichung des Semantic Web werden unter Federführung des W3C derzeit entwickelt und sind zum großen Teil bereits verfügbar. Eine zentrale Rolle nimmt dabei das Resource Description Framework ein.
Ein kritischer Punkt des Konzeptes ist sicher die Akzeptanz. Es ist schließlich ein ungeheuerliches Unterfangen, die Riesenmenge an Informationen im WWW mit semantischen Metadaten zu versehen. Das wird sicher ein langwieriger Prozess. Aufgrund des verteilten Konzeptes ist es jedoch möglich, punktuell an verschiedenen Stellen zu beginnen. Man wird sicher zunächst die Domänen des Webs semantisch erschließen, wo der Bedarf an qualifizierten Metadaten am größten ist (z.B. wissenschaftliche Online-Zeitschriften) oder sich aufgrund des inhaltlichen Mehrwertes auch Gewinne in Euro und Cent abzeichnen (z.B. Reiseanbieter). Einzelnen Domänen können dann aufgrund der standardisierten Technologie problemlos zusammenwachsen.
Beim W3C hat man die begründete Hoffnung, dass das Semantic Web eine ähnliche Erfolgsgeschichte wird wie das World Wide Web, welches das vorher schwer bedienbare Internet zum Massenmedium machte. Die schon lange bekannte Hypertext-Technologie fiel endlich auf fruchtbaren Boden. Genauso könnte es mit den Erkenntnissen über semantische Netze sein. Ihr verbreiterter Einsatz würde zu einer neuen Revolution im weltweiten Netz führen.

7. Quellenverzeichnis

[BERN2001] Tim Berners-Lee, James Hendler, Ora Lassila: The Semantic Web, Scientific

lee.html

[CHAM2001] Pierre-Antoine Champin: RDF Tutorial, http://www710.univ-lyon1.fr/~champin/rdf-tutorial/rdf-tutorial.pdf

[DC2002] Dublin Core Metadata Initiative, http://dublincore.org

[GRAN2001] Benedicte Le Grand, Michel Soto, David Dodds: XML Topic Maps and Semantic Web Mining, XML Conference & Exposition 2001, http://www.idealliance.org/papers/xml2001/papers/html/04-04-05.html

[HAUS1989] R. Hausser: Grundlagen der Computerlinguistik. Mensch-Maschine-Kommunikation in natürlicher Sprache, Springer Verlag Berlin 1989

[HELB2001] H. Helbig: Die semantische Struktur natürlicher Sprache, Springer Verlag Berlin 2001

[LOBI2001] Henning Lobin: Netzwerkbasierte Modellierung der Semantik von XML-Strukturen, Proceedings der GLDV-Frühjahrstagung, Universität Gießen 2001 S.141-150, http://www.uni-giessen.de/fb09/ascl/gldv2001/proceedings/pdf/GLDV2001-lobin.pdf

[MOOR2001] Graham Moore: RDF and TopicMaps - An Exercise in Convergence, XML Europe 2001 Berlin 2001, http://www.topicmaps.com/topicmapsrdf.pdf

[RDF1999] Ola Lassila, Ralph R. Swick: Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[RDFS1999] Dan Brickley, R. V. Guha: Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation 27 March 2000, http://www.w3.org/TR/2000/CR-rdf-schema-20000327/

[SCHO1998] Eckhardt Schön: Das Resource Description Framework (RDF) - ein neuer Weg zur Verwaltung von Metadaten im Netz, 4. Workshop Multimediale Informations- und Kommunikationssysteme, Ilmenau 1998, Tagungsband 1998 S.14-21, http://www.prakinf.tu-ilmenau.de/~schoen/forschung/mik1998/worksh98.htm

[XML2000] Tim Bray, Jean Paoli, u.a.: Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation 6 October 2000, http://www.w3.org/TR/2000/REC-xml-20001006

[XMLS2001] XML Schema, W3C Recommendation, 2 May 2001, http://www.w3.org/XML/Schema

[XPOI2001] XML Pointer Language (XPointer) Version 1.0, W3C Candidate Recommendation 11 September 2001, http://www.w3.org/TR/xptr/

[XTM2001] Steve Pepper, Graham Moore: XML Topic Maps (XTM) 1.0, TopicMaps.Org Specification, http://www.topicmaps.org/xtm/index.html