# Vokabelheft fürs Web

von Dr. Christoph Lange, Universität Bremen

Wer KDE 4 installiert und Nepomuk einschaltet, bekommt einen Eindruck, was das semantische Web ist. Im Internet der Zukunft sollen Computer die Bedeutung aller im Web veröffentlichten Informationen verstehen und diese für den Anwender intelligent aufbereiten. Die Voraussetzung ist ein einheitliches Vokabular für das Beschreiben der Begriffe.

Das derzeit auf die mehr oder weniger reine Sammlung und ungewichtete Anzeige von Informationen ausgerichtete World Wide Web soll mittelfristig zu einem semantischen Web weiterentwickelt werden. Das Ziel ist, daß Computer die Bedeutung aller im Web veröffentlichten Informationen verstehen und sie für den Anwender intelligent aufbereiten können. Die Voraussetzung sind intelligente Dienste, Agenten und Geräte, die maschinenverständliches Wissen aus dem Web abgreifen und daraus auf vertrauenswürdige Weise logische Schlußfolgerungen ziehen (»Wo ist die bestbewertete Klinik im näheren Umkreis, die die richtige Behandlung für meine Mutter anbietet?«). Dies haben die Pioniere des semantischen Web in ihrem inzwischen legendären Artikel im Scientific American von 2001 beschrieben [1]. Diese Vision ist nach wie vor gültig, doch der Weg dorthin hat sich als langwieriger als angenommen herausgestellt.

Die Forschung und Entwicklung in den letzten Jahre konzentrierte sich darauf, die schon vorhandenen Daten in das semantische Web zu integrieren und es seinen Diensten und Agenten zu ermöglichen, Schlüsse aus ihnen zu ziehen. Aus öffentlichen Webpräsenzen wird unter dem Stichwort »Linked Open Data« [2] zum Beispiel das Wissen aus der Enzyklopädie Wikipedia, aus sozialen Netzen, aus staatlich erhobenen Statistiken und vielen weiteren Quellen für das semantische Web aufbereitet. Im Privaten geschieht dasselbe inzwischen dank der Nepomuk-Technologie auf jedem KDE-Desktop [3]. Die dortige semantische Suchmaschine beantwortet Fragen wie beispielsweise »Welche in den letzten Tagen geänderten Dateien, die das Wort ›Oma‹ enthalten, habe ich mit dem Tag ›familie‹ versehen und mit mindestens vier Sternchen bewertet?«. Der Index dieser Suchmaschine wie auch die im semantischen Web veröffentlichten, wollen allerdings gefüttert werden. Die Grundlage bilden Daten im RDF-Modell.

## Das RDF-Netzdatenmodell

RDF (Resource Description Framework) ist ein Datenmodell zur Beschreibung von Ressourcen und ihren Eigenschaften. Eine Ressource kann – tatsächlich – alles Mögliche sein: ein konkretes, im Web veröffentlichtes Dokument, eine Person, aber auch ein immaterielles Konzept wie »das Datenmodell RDF«. Zu den Eigenschaften (Properties) einer Ressource können Datenwerte gehören (auch Literale genannt), wie zum Beispiel das Datum, an dem ein Dokument veröffentlicht wurde, Verbindungen zu anderen Ressourcen, etwa von einem Dokument zu seinem Autor oder seinem Thema. In diesem Sinne sind Ressourcen vergleichbar mit Objekten in objektorientierten Programmiersprachen; auch dort spricht man ja von Eigenschaften eines Objekts.

Als Anwender kann man sich diese Informationen gut als Graph vorstellen, Bild 1 zeigt ein Beispiel eines RDF-Graphen. Ressourcen sind durch Ellipsen dargestellt, Literale durch Rechtecke und die Verbindungen durch beschriftete Pfeile (der Informatiker spricht von Kanten). Das heißt, daß ein RDF-Graph nicht nur besagt, daß etwa zwei Ressourcen in irgendeiner Verbindung miteinander stehen, sondern auch welcher Art die Verbindung zwischen ihnen ist. In Bild 1 ist beispielsweise die Rede von Horst, seiner E-Mail-Adresse und den Beiträgen, deren Autor er ist.

Bild 1 zeigt das Szenario einer semantischen Suche: Das Web-Startup-Unternehmen WoIstX sucht ab sofort einen Software-Entwickler, um den Index seiner semantischen Suchmaschine zu optimieren. Horst, ein Entwickler mit Erfahrung in RDF-basierten Anwendungen, ist seit dem 1. Oktober 2011 auf Stellensuche; diese Informationen hat er als RDF auf seiner Homepage hinterlegt, indem er diese Informationen nicht nur als Text auf seine Homepage geschrieben (»Ich kenne mich hervorragend mit RDF aus und bin seit Oktober offen für neue Herausforderungen«), sondern auch in einer für (Such-)Maschinen verständlichen Weise bereitgestellt hat.

Nun könnte man meinen, daß Google doch auch eine Maschine ist und sich manchmal sogar scheinbar intelligent verhält. Im Prinzip stimmt das auch, aber Google wertet aus, wie häufig Wörter auf Seiten vorkommen (unabhängig davon, was die Wörter tatsächlich bedeuten) und wie stark Seiten miteinander verlinkt sind (unabhängig davon, was ein Link bedeutet). Google kann also nicht sicher wissen, daß mit RDF das Resource Description Framework gemeint ist (und nicht die Richard Dawkins Foundation for Reason and Science), daß »Oktober« sich auf das Jahr 2011 bezieht, und welche Aussage sich hinter »offen für neue Herausforderungen sein« verbirgt. All das kann mit RDF explizit gemacht werden, indem man die Kanten eines Graphen nicht irgendwie beschriftet, und, wenn von allgemeinen Begriffen die Rede ist, nicht auf irgendwelche Ressourcen verweist, sondern mit einem standardisierten Vokabular arbeitet.

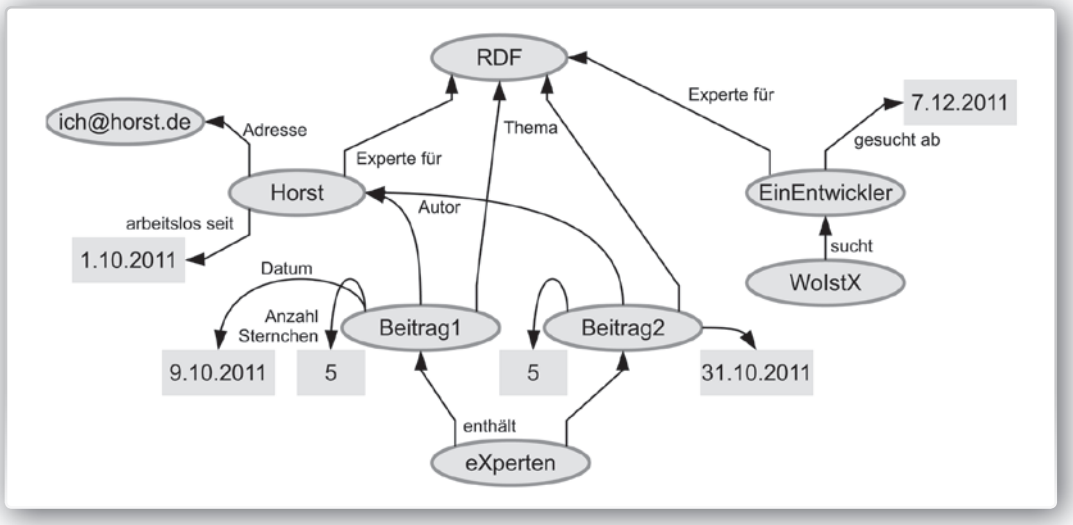



Bild 1: Ein RDF-Graph

## Verständnis durch Vokabulare

Die Begriffe in solchen RDF-Vokabularen sind weltweit eindeutig durch URIs (Uniform Resource Identifiers) bezeichnet; die häufigste Art solcher URIs sind URLs (L = Locator), die zusätzlich zur Identifikation noch die Information enthalten, unter welcher Adresse im Netz die Ressource zu finden ist – hier also, wo die Beschreibung eines Vokabular-Begriffs zu finden ist. Ein weithin anerkannter Begriff, der die Beziehung zwischen einer Person und ihrer E-Mail-Adresse ausdrückt, ist beispielsweise *http://xmlns.com/foaf/0.1/mbox*, oft abgekürzt als foaf:mbox. Wird diese Adresse im Browser geöffnet, findet man die verständliche Beschreibung des Vokabulars, zu dem dieser »mbox«-Begriff gehört (einfach auf der Seite nach »mbox« suchen). Für Maschinen sind diese Vokabulare wiederum als RDF beschrieben, und tatsächlich erhält man eine RDF-Beschreibung des Begriffs »mbox«, wenn dem Server per HTTP-Accept-Header gesagt wird, daß er die Antwort nicht als HTML, sondern als RDF/XML geben soll. (RDF/XML ist das am weitesten verbreitete Format zum Austausch von RDF-Graphen.)

```
$ curl -L \
      -H 'Accept: application/rdf+xml' \
       http://xmlns.com/foaf/0.1/mbox
```

Oder alternativ:

```
$ wget -O - \
 --header='Accept: application/rdf+xml'\
 http://xmlns.com/foaf/0.1/mbox
```

Hier kommt wie zuvor im Browser als Ausgabe das ganze Vokabular, nicht nur der eine Begriff; das heißt, man muß die Ausgabe wiederum durchsuchen. Ohne jetzt auf die technischen Hintergründe einzugehen, wie RDF-Vokabulare codiert werden, sei soviel festgehalten: Man wird feststellen, daß die kurze Beschreibung des Begriffs foaf:mbox nicht erklärt, was es eigentlich bedeutet, eine E-Mail-Adresse zu besitzen. Das muß es auch nicht, denn bei RDF-Vokabularen geht es lediglich darum, sich auf einheitliche Begriffe für Sachverhalte zu einigen und einige für Anwendungen relevante Aspekte zu beschreiben. foaf:mbox ist zum Beispiel als »invers funktionale Eigenschaft« deklariert, das heißt auf Deutsch: Eine E-Mail-Adresse hat genau einen Besitzer. Semantische Suchmaschinen wie beispielsweise *http://sig.ma* stellen auf Basis dieses Hintergrundwissens fest, ob mehrere Namen dieselbe Person bezeichnen. Auf unterschiedlichen Webseiten könnte beispielsweise die Rede von »Horst Hacker«, »Horst W. Hacker« oder dem »Gewinner des Semantic-Web-Programmierwettbewerbs 2011« die Rede sein und dabei auf *ich@horst.de* verwiesen werden. Da zu dieser E-Mail-Adresse aber nur eine Person gehören kann, müssen all dies unterschiedliche Bezeichnungen für dieselbe Person sein. Weiter gefolgert heißt das: Wenn Horst auf seiner Homepage ein Foto von sich zeigt, kann die Suchmaschine dieses Foto auch als Ergebnis anzeigen, wenn jemand nach dem »Gewinner des Semantic-Web-Programmierwettbewerbs 2011« sucht.

## Ablauf einer semantischen Suche

Weiter im Szenario: Der Personalchef der Firma WoIstX ist dank der oben genannten Beschreibung bei seiner Suche nach geeigneten Kandidaten auf Horst aufmerksam geworden. Aber was

garantiert ihm, daß Horst nicht nur ein selbsternannter, sondern auch ein von anderen anerkannter RDF-Experte ist? Horst nimmt regelmäßig an Diskussionen zu RDF im Webforum eXperten teil; seine Hilfestellungen erhalten stets Spitzenbewertungen von anderen Teilnehmern, und die Forensoftware stellt diese Informationen als RDF bereit [4]. Das genügt dem Personalchef nun, um Horst per E-Mail zu kontaktieren.

Insgesamt lautet die Suchanfrage von WoIstX so: »Finde alle Personen, die 1) sich selbst als RDF-Experten bezeichnen, die 2) zum Thema RDF mit fünf von fünf Sternchen bewertete Beiträge veröffentlicht haben, und die 3) zu dem Zeitpunkt, da wir sie einstellen können, auf Arbeitssuche sind. Gib von diesen Personen die E-Mail-Adressen aus.« Die Wiedergabe als Text soll hier genügen, in Programmen werden solche Abfragen üblicherweise in der der Datenbank-Abfragesprache SQL ähnlichen Sprache SPARQL formuliert.

Zu beachten ist, daß diese Abfrage dank der Standardisierung der Vokabulare nicht nur in diesem konkreten Fall der Firma WoIstX hilft, Horst anhand seiner Homepage und seiner Beiträge im »eXperten«-Forum als RDF-Experten zu identifizieren, sondern daß der Wettbewerber Kuugel mit genau derselben Abfrage auch den Web-Arbeitsmarkt abgrasen kann und möglicherweise im »Netz-Intelligenz«-Forum den Experten Heinz findet.

## RDF-Graphen als Bits und Bytes

Ein Mensch kann sich einen RDF-Graphen gut bildlich vorstellen. Hat man die obige Kommandozeile ausprobiert, wird man gesehen haben, daß Maschinen Folgen von Bits und Bytes bevorzugen, das heißt eine serialisierte Form eines Graphen. Die RDF/XML-Serialisierung ist aus historischen Gründen weit verbreitet, aber für Menschen und sogar für Maschinen eher schwer zu lesen, weshalb die Serialisierung von RDF hier intuitiver erklärt sei. Ein Graph wird serialisiert, indem jede Kante als Satz der Form »Subjekt Prädikat Objekt« in einer einfachen Sprache aufgefaßt

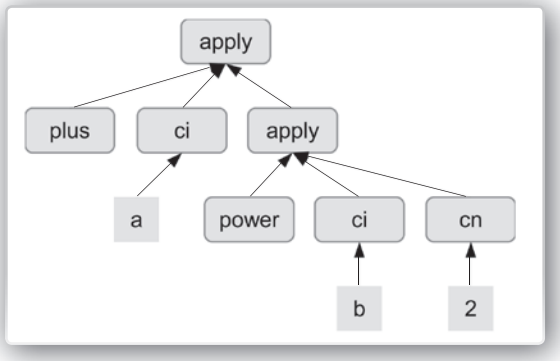



Bild 2: XML-Baum von $a+b^2$ in Content MathML

wird, wegen seiner drei Komponenten auch »Tripel« genannt – zum Beispiel

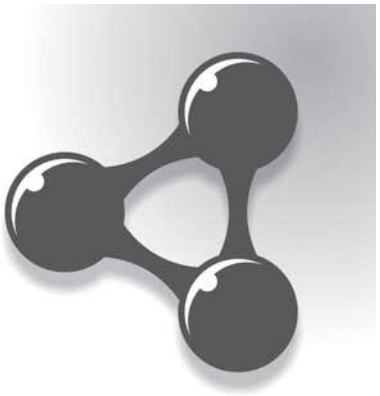
```
<http://horst.de/#ich>
<http://xmlns.com/foaf/0.1/mbox>
<mailto:ich@horst.de> .
```

… oder ausgesprochen: »Horst hat die E-Mail-Adresse ich@horst.de.« Man sieht, daß die in Bild 1 gezeigten Ressourcen eigentlich auch alle durch vollständige URIs identifiziert sind. Der ganze RDF-Graph aus Bild 1 wird zu einer Folge solcher Sätze serialisiert. XML (eXtended Markup Language) ist wie RDF ein Datenmodell für strukturierte Daten. XML erlaubt es – ähnlich wie RDF – Vokabulare zur Strukturierung von Daten zu definieren und dann konkrete Sachverhalte mit diesen Vokabularen zu beschreiben. So weit klingt beides sehr ähnlich, was weiter oben auch daran erkannt werden konnte, daß RDF-Graphen in dem XML-basierten Format RDF/XML serialisiert werden können.

Der wesentliche Unterschied ist allerdings, daß XML nicht in Graphen denkt, sondern in Dokumenten. Nicht umsonst werden mehr und mehr Dokumente zwischen Anwendungsprogrammen mit Hilfe von XML-Vokabularen ausgetauscht. Beispiele für solche XML-Vokabulare sind zum Beispiel das Open-Document-Format von LibreOffice und anderen Office-Programmen, das Vektorgrafikformat SVG (Scalable Vector Graphics), und schließlich das Webseitenformat HTML, dessen Variante XHTML ein XML-Vokabular ist. Ein XML-Dokument hat eine Baumstruktur; so besteht ein Office-Dokument zum Beispiel aus Absätzen, die Unterabsätze enthalten, wobei ein Absatz aus seiner Überschrift und dem darauf folgenden Text besteht. Ein Beispiel in einem fiktiven Format zeigt die hierarchische Schachtelung:

```
<dokument>
  <titel>Doktorarbeit: ...</titel>
  <autor>A. Ka. Demiker</autor>
  <absatz>
   <überschrift>Einleitung</überschrift>
    <text>In dieser Arbeit geht es
          darum, ...</text>
    <absatz>
     <überschrift>Vorbemerkungen
     </überschrift>
     <text>Wie schon Sokrates sagte, ...
     </text>
    </absatz>
    <absatz>
      <überschrift>Danksagungen
      </überschrift>
      <text>Beim Verfassen dieser
        Abhandlung haben mich
        unterstützt: ...</text>
    </absatz>
  </absatz>
  <absatz>
    <überschrift>Stand der Forschung
    </überschrift>
    <text>...</text>
  </absatz>
  ...
</dokument>
```

Dabei sind »dokument«, »titel« und so weiter sogenannte Elemente, und sie enthalten wiederum Elemente oder Textknoten. Dieses Dokument hat eine schwache Semantik; alles, was eine Maschine verstehen kann, ist, daß es aus geschachtelten Absätzen besteht.



Es gibt aber auch XML-Vokabulare mit stärkerer Semantik, beispielsweise die Sprache MathML zum Austausch mathematischer Formeln. In seiner Teilsprache Content MathML wird eine Formel nicht nach dem beschrieben, wie sie auf dem Bildschirm oder Papier aussieht, sondern was sie bedeutet, zum Beispiel die Formel $a+b^2$:

```
<apply>
  <plus/>
  <ci>a</ci>
  <apply>
    <power/>
    <ci>b</cn>
    <cn>2<cn>
  </apply>
</apply>
```

Das heißt: Wende den Operator »plus« an auf die Variable a und auf das Ergebnis der Anwendung (Application) des Operators »power« (Potenz) auf die Variable b und die Zahl 2. Bild 2 verdeutlicht die Baumstruktur mit Elementen als gerundeten Rechtecken und Textknoten als Rechtecken.

Nachdem nun auch XML-Dokumente eine Semantik haben können: Warum wird im semantischen Web nicht gleich mit XML- statt RDF-Vokabularen gearbeitet? Oder umgekehrt: Wenn man sich, warum auch immer, im semantischen Web für das RDF-Datenmodell entschieden hat, warum werden dann immer noch neue XML-Vokabulare entwickelt, statt daß alle neuen Daten gleich in RDF repräsentiert werden?

Der Grund ist, daß RDF und XML jeweils ihre Vor- und Nachteile haben, so daß es sich je nach Anwendung anbietet, eines der beiden Modelle zu wählen. Das einführende Beispiel zeigte, daß RDF dazu geeignet ist, übers ganze Web vernetztes Wissen zu repräsentieren. Ähnlich wie das immer beliebtere NoSQL-Prinzip bei Datenbanken eignet sich RDF für unvollständige oder erweiterbare Datensammlungen, die sich nicht an ein Vokabular mit einer festen Zahl von Begriffen halten müssen. RDF in seiner Einfachheit macht es einem aber sehr schwer, Reihenfolge und 1:n-Beziehungen zu modellieren. In den zu XML gezeigten Beispielen ist zu sehen, daß in einem Dokument mehrere Abschnitte aufeinanderfolgen können. Mathematische Operatoren nehmen oft mehrere Argumente – nicht nur a+b ist möglich, auch a+b+c –, und oft ist die Reihenfolge wichtig – $2^b$ ist nicht dasselbe wie $b^2$. Da Reihenfolge und 1:n-Beziehungen in XML gut ausgedrückt werden können, haben sich XML-Vokabulare für den Austausch von Office-Dokumenten und, zum Teil, mathematischen Formeln durchgesetzt. Ein wesentlicher Nachteil von XML ist wiederum, daß es zwar durchaus Möglichkeiten gibt, Teile von XML-Dokumenten oder in XML-Dokumenten beschriebene Dinge mit URIs weltweit eindeutig zu identifizieren und »beschriftete« Verbindungen zwischen ihnen auszudrücken, daß sich aber kein solches Verfahren im großen Umfang durchgesetzt hat. Zweitens sind XML-Vokabulare genau wie RDF-Vokabulare weltweit eindeutig durch URIs bezeichnet. Wenn man sie im Browser öffnet, findet man auch meistens eine menschenverständliche Spezifikation des Vokabulars – aber praktisch nie eine maschinenverständliche. Aus diesen Gründen ist XML in der Praxis weit weniger geeignet als RDF, um Wissen übers Web hinweg auf maschinenverständliche Art zu vernetzen.

## RDF und XML im Vergleich: Netz oder Dokument?

Es ist allerdings oft nötig, denselben Sachverhalt in RDF und in XML zu modellieren. Eine RDF-basierte Suchmaschine könnte es interessieren, welche Absatzüberschriften in einem XML-Office-Dokument vorkommen, wer das Dokument geschrieben hat und was dieselbe Person im Web sonst noch veröffentlicht hat. Umgekehrt müssen RDF-Graphen für menschliche Anwender visualisiert werden, und als Format dafür bietet sich das XML-basierte SVG an.

Die Zukunft liegt also in einer friedlichen Koexistenz von RDF und XML, beide können sogar parallel im selben Dokument vorkommen. RDFa [5] wurde entwickelt, um RDF-Graphen in XML-Dokumente einbetten zu können, insbesondere in HTML-Seiten. Der Entwickler Horst aus dem ersten Beispiel muß also die maschinenverständlichen Informationen über sich gar nicht separat als RDF/XML-Datei auf seiner Homepage bereitstellen, sondern kann sie direkt in den HTML-Quelltext seiner Homepage einbetten, zum Beispiel so:

```
<div about="http://horst.de/#ich">
  Ich bin ein erfahrener
  Software-Entwickler. Sie können mich
  jederzeit per
  <a rel="http://xmlns.com/foaf/0.1/mbox
   href="mailto:ich@horst.de">E-Mail</a>
  erreichen.
</div>
```

Dieser erste Teil unserer Semantic-Web-Serie hat das semantische Web kurz vorgestellt, die unterschiedlichen Datenmodelle RDF und XML eingeführt und klargestellt, warum beide gebraucht werden. Der nächste Teil behandelt die Übersetzung von XML nach RDF, um etwa semantische Suchmaschinen mit Wissen aus Dokumenten zu füttern. Dazu werden konkrete Übersetzungen entwickelt und es wird gezeigt, wie semantische Suchmaschinen auf den resultierenden RDF-Graphen Suchanfragen beantworten können, die mit reiner XML-Technik wesentlich schwieriger zu beantworten wären.

**Literatur**


[1] Tim Berners-Lee, James Hendler, Ora Lassila: The Semantic Web – A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American 284, 2001.

[2] Zur Einführung siehe »Linked Open Data« in Wikipedia: *http://de.wikipedia.org/wiki/Linked_Open_Data*

[3] Nepomuk – The Semantic Desktop in KDE: *http://nepomuk.kde.org*

[4] Siehe dazu das RDF-Vokabular SIOC (Semantically-Interlinked Online Communities): *http://sioc-project.org*

[5] Ben Adida, Ivan Herman, Manu Sporny: RDFa 1.1 Primer – Rich Structured Data Markup for Web Documents. W3C Working Draft, 2011: *http://www.w3.org/TR/rdfa-primer/*