



XML-Technologien

Tutorium 6

Thema:

Semantic Web
Linked Data
RDF
SPARQL

von:

Kain Kordian Gontarska
kainkordian@gmail.com

Semantic Web

- Das Internet soll nicht nur aus "dummen" Inhalten bestehen. (Ansammlung von unstrukturierten Dokumenten)
- Informationen sollen stattdessen strukturiert und verständlich für Maschinen sein.
- Weiter sollen nicht nur einfache Metainformationen vergeben werden, sondern einzelne Ressourcen miteinander in Beziehungen gebracht werden (Referenzierung).
- Linked Data als Grundbaustein des Semantic Webs.

Linked Data

- Per URI identifizierbare Daten auf weitere Daten verweisen lassen (per URI).
- Miteinander verknüpfte Daten bilden einen Graphen (Giant Global Graph – GGG)
- WWW – Netz aus Webseiten ↔ GGG – Netz von Daten
- Tim Berners-Lee four principles of linked data:
 1. Use URIs as names for things.
 2. Use HTTP URIs so that these things can be looked up.
 3. Provide useful information about what a name identifies when it's looked up, using open standards such as RDF, SPARQL, etc.
 4. Refer to other things using their HTTP URI-based names when publishing data on the Web.
- Umsetzung insbesondere durch RDF.

Linked Data

- „New“ principles (Tim Berners-Lee, TED 2009) for linked data:
 1. All kinds of conceptual things, they have names now that start with HTTP.
 2. If I take one of these HTTP names and I look it up [...] I will get back some data in a standard format which is kind of useful data that somebody might like to know about that thing, about that event.
 3. When I get back that information it's not just got somebody's height and weight and when they were born, it's got relationships. And when it has relationships, whenever it expresses a relationship then the other thing that it's related to is given one of those names that starts with HTTP.

RDF – Resource Description Framework

- Jede Aussage im RDF-Modell ist ein Tripel und besteht aus 3 Einheiten:

[Subjekt] [Prädikat] [Objekt]

- Hierbei wird eine Ressource (Subjekt) in eine Beziehung (Prädikat) zu einer anderen Ressource oder einem Literal (Objekt) gesetzt.
- Vereint man diese Tripel, so erhält man einen gerichteten Graphen (RDF-Graph), an den man SPARQL-Anfragen stellen kann.

RDF – Resource Description Framework

- Prädikate erfordern ein gewisses vereinbartes Vokabular um Beziehungen eindeutig zu benennen.
- schema.org - Eine von den 3 großen Suchmaschinen entwickelte und vereinbarte Ontologie zur Strukturierung von Daten. Ein Großteil der enthaltenen Klassen wurde aus früheren Spezifikationen übernommen (Microformats, FOAF, ...)
- Ein wichtiges Prädikat ist `rdf:type`, damit kann einem Subjekt ein vereinbarter Typ gegeben werden (um vereinbarte Eigenschaften zu verwenden)

Beispiel: schema.org/Person - folgt

RDFa

- Ein kleines Beispiel für die RDFa-Syntax (in HTML einbindbar) :

```
<div>
  <p>Kaufen Sie den Staubsauger XF704 jetzt im
  Sonderangebot!
  
</p>
</div>
```

RDFa

```
<div vocab="http://schema.org/"
  resource="#xf704" typeof="Product">
  <p>Kaufen Sie den
    <span property="name">
      Staubsauger XF704
    </span>
    jetzt im Sonderangebot!
    
  </p>
</div>
```

```
[#xf704] [rdf:type] [schema:Product]
```

```
[#xf704] [schema:name] ["Staubsauger XF704"]
```

```
[#xf704] [schema:image] [<http://example.com/images/acmeXF704.jpg>]
```

RDF

Die Tripel können auch direkt aufgeschrieben werden und einen Triple-Store bilden.

Dafür wird entweder die ursprünglich RDF-Syntax genommen (wie in VL-Folien) oder der neuste vom W3C akzeptierte Standard (N3/Turtle) welcher in seiner Syntax SPARQL ähnelt.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
@prefix schema: <http://schema.org> .
```

```
@prefix myns: <http://example.com/staubsauger#> .
```

```
myns:xf704 rdf:type schema:Product .
```

```
myns:xf704 schema:name "Staubsauger XF704" .
```

```
myns:xf704 schema:image <http://example.com/images/acmeXF704.jpg> .
```

SPARQL

- SPARQL - SPARQL Protocol And RDF Query Language
- Ist eine Anfragesprache an einen RDF Graphen - Prinzip ähnlich einer relationalen Datenbank.
- Das Rückgabeformat kann definiert werden: XML, JSON, CSV, RDF, HTML
- Anfrage besteht aus:
 1. (Optional) Präfix Deklaration - Zur Abkürzung von URIs - **PREFIX**
 2. Ergebnisklausel - Welche Informationen will man aus der Anfrage erhalten? - **SELECT**
 3. (Optional) Datensatz Definition - welcher RDF-Graph soll verwendet werden? - **FROM**
 4. Das Anfragemuster - Filtert die Informationen die man haben möchte. - **WHERE**
 5. (Optional) Anfragemodifizierer - Verarbeitet die Anfrage (Sortieren, Reduzieren, ...) - **LIMIT, OFFSET, ORDER BY**

SPARQL - Beispiel

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX : <http://dbpedia.org/resource/>

SELECT ?name ?birth ?description
WHERE {
    ?person dbo:birthPlace :Berlin ;
        <http://purl.org/dc/terms/subject>
            <http://dbpedia.org/resource/Category:German_musicians> ;
        dbo:birthDate ?birth ;
        foaf:name ?name ;
        rdfs:comment ?description .
    FILTER (LANG(?description) = 'en') .
}
ORDER BY ?name
```

Probieren wir es auf <http://dbpedia.org/sparql> aus!