Information Retrieval - Semantic Technologies

Resource Description Framework

Albert Weichselbraun

RDF

- RDF ist ein Datenmodell; Grundlegende Struktur: Graph
- Darstellung von Aussagen: Subject - Prädikat - Objekt
- Durch Kombination von Einzelaussagen sind komplexe Konstrukte möglich.
- Serialisierungen:
  - Graphendarstellung
  - Turtle
  - XML/RDF Kurz-/Langformat

RDF - Konzepte - Tripel

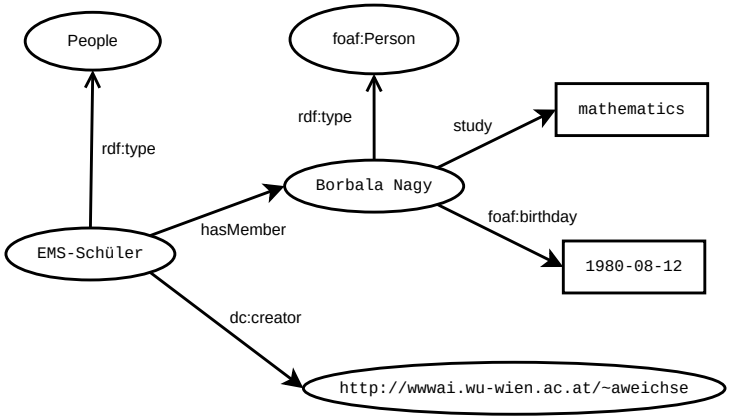
- atomare Einheit: Statement (Aussage)
- jedes Statement ist aus einem Tripel aufgebaut:

```
Resource  Eigenschaft  Wert.
(subj)    (pred)       (obj).

Borbala_Nagy foaf:birthday "1980-08-12".
Borbala_Nagy ems:study     "mathematics".
```

- Ressourcen werden immer mit URLs (Namespaces) benannt.
- Literals: konkrete Werte (!=Ressourcen); ohne *Datentyp* immer als Strings interpretiert
- Subjekt und Prädikat sind *immer* Ressourcen.

RDF - Konzepte - Graph



RDF - Populäres XML-Vokabular

- Dublin Core (http://purl.org/dc/elements/1.1/#) - umfasst 15 Elemente zur Beschreibung von Webressourcen
  - `dc:creator` - der Ersteller einer Ressource
  - `dc:subject` - Thema (Schlagwörter)
  - ...
- Friend of a Friend (http://xmnls.com/foaf/0.1) - Vokabeln zur Modellierung von sozialen Netzen
  - `foaf:Person` - eine Person
  - `foaf:birthday` - Geburtstag einer Person
  - ...

Serialisierung - Turtle

- Einfaches und leicht lesbares Serialisierungsformat
- Syntax:
  - `@prefix {identifier}: <{url}>.`
  - `{subj} {pred} {obj}.`
    `{subj} {pred} {obj}[, ... ,{obj}].`
    `{subj} {pred}{obj}[; ... ;{pred} {obj}].`
  - Ressourcen: `ex:name, <http://name.org/res>`
  - Literals: `"value"`
  - Vorteile:
    - einfache Darstellung
    - Abbildung in Datenbanken

Serialisierung - Turtle

```
1  @prefix aw:   <http:/bsp.at/x#>.
2  @prefix dc:   <http://purl.org/dc/elements/1.1/#>.
3  @prefix foaf: <http://xmlns.com/foaf/1.1/>.
4  @prefix rdf:  <http://.../22-rdf-syntax-ns#>.

6  aw:Student rdf:type       aw:People;
7             dc:publisher   <http://wu.at/~ana>;
8             aw:members        aw:Bori.
9  aw:Bori    rdf:type       aw:Person;
10            foaf:birthday  "1980-08-12".
```


Serialisierung - XML/RDF (lang)

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <rdf:RDF xmlns="http://bsp.at/x#"
3           ... >
4  <rdf:Description rdf:about="http://bsp.at/x#Student">
5    <rdf:type rdf:resource="http://bsp.at/x#People" />
6    <dc:publisher rdf:resource="http://wu.at/~ana" />
7    <members>
8      <rdf:Description rdf:about="http://bsp.at/x#Bori">
9        <rdf:type
10           rdf:resource="http://bsp.at/x#Person" />
11       <foaf:birthday>1980-08-12</foaf:birthday>
12     </rdf:Description>
13   </members>
14 </rdf:Description></rdf:RDF>
```

Serialisierung - XML/RDF

Aufbau: **Wichtige Merkmale:**

- gültige XML Dokumente
- Wenn nicht aus dem Kontext klar ist, dass es sich um XML/RDF Daten handelt → `rdf:RDF`-Tag als Root-Element
- Ressourcen werden beschrieben durch:
  - Literals (Angabe des Wertes) oder durch
  - andere Ressourcen, auf welche mittels `rdf:resource` verlinkt wird, oder welche direkt vor Ort definiert werden

Serialisierung - XML/RDF

- Sämtliche Elemente (Ressourcen, Klassen, Eigenschaften) müssen *eindeutig* identifizierbar sein.
  - Verwendung von Namenräumen
  - Klassen/Eigenschaften erhalten eine eindeutige URL durch Kombination mit `xmlns`
- Ressourcen werden mittels `rdf:about` referenziert.
- keine Namespace Prefixe in Attributwerten (volle URL notwendig; vergleiche: `rdf:about`)

Serialisierung - XML/RDF (kurz)

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <rdf:RDF xml:base="http://bsp.at/y.xml#"
3           xmlns="http://bsp.at/x#"
4           ...>
5  <!-- Typ und Ressourcenbezeichnung werden
6       Zusammengefasst -->
7  <People rdf:ID="Student">
8    <dc:publisher rdf:resource="http://wu.at/~ana" />
9    <members>
10      <Person rdf:ID="Bori">
11         <foaf:birthday>1980-08-12</foaf:birthday>
12      </Person>
13   </members>
14 </People></rdf:RDF>
```


Serialisierung - XML/RDF (kurz)

Aufbau: **Wichtige Merkmale:**

- `rdf:Description` und `rdf:type` werden durch Verwendung des Typen als Elementname ersetzt.
- Ressourcen werden mittels `rdf:ID` eindeutig durch XML-Namen identifiziert.
- eindeutige Identifikation von Ressourcen mittels *rdf:ID*
  → enthält nur den lokalen Teil des URL
  → gesamter URL durch Kombination mit `xml:base`
- Konvention: Klassennamen → groß, Eigenschaften → klein

Serialisierung - XML/RDF

```
<Class rdf:ID="Resource" ...>

    <!-- Beschreibung durch Literal -->
    <property>value</property>

    <!-- Referenz auf andere  Ressource -->
    <property rdf:resource="url" />

    <!-- Definition der Ressource vor Ort -->
    <property>
      <Class rdf:ID="Resource2">...</Class>
    <property>
    ...
</Class>
```

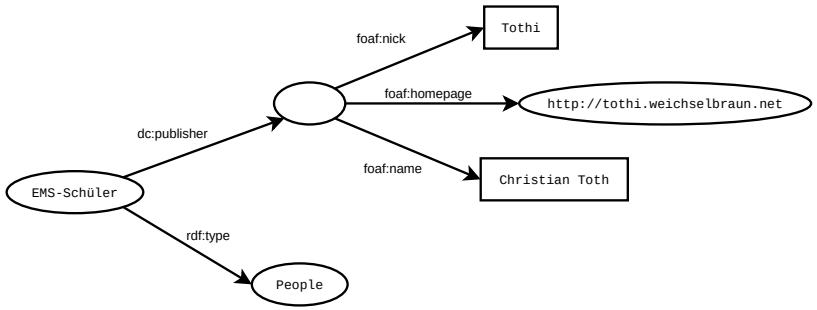

XML/RDF - Design-Pattern

```
<Resource-1>
   <eigenschaft-1>
        <Resource-2>
             ...
                 <eigenschaft-n>
                    Wert n
                 </eigenschaft-n>
             ...
        </Resource-2>
   </eigenschaft-1>
</Resource-1>
```


Anonyme Ressourcen - Syntax

```
<People rdf:ID="EMS-Sch&uuml;ler"
  xmlns="http://.../2008/people/"
  xmlns:dc="http://purl.org/dc/elements/1.1/#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" >
   <dc:publisher>
      <rdf:Description>
         <foaf:name>Christian Toth</foaf:name>
         <foaf:nick>Tothi</foaf:nick>
         <foaf:homepage
           rdf:resource="http://tothi.net" />
      </rdf:Description>
   </dc:publisher>
</People>
```


Anonyme Ressourcen - Graph



Anonyme Ressourcen - Turtle

- Vergabe eines eindeutigen Namens vom Typ: `_:name`
- dieser wird für Referenzen auf die Ressource verwendet.

```
1  @prefix  foaf:  <http://xmlns.com/foaf/0.1/>.

3  <http://.../people/EMS-Sch&uuml;ler> dc:publisher
4      _:genid01.

6  _:genid01  foaf:name      "Christian Toth";
7             foaf:nick      "Tothi";
8             foaf:homepage
9               <http://tothi.weichselbraun.net>.
```


Datentypen und Sprachangaben

- Spezifizieren des Datentyps *oder* der Sprache von Literalen
- Die XML-Schema Spezifikation enthält vordefinierte Typen (`string`, `date`; siehe http://www.w3.org/2001/XMLSchema#)
- Definition eigener Typen möglich
- Angabe des Datentyps
  - XML/RDF: via `rdf:datatype`
  - Turtle/Graph : `"Value"^^{datatype}`
- Angabe der Sprache:
  - XML/RDF: via `xml:lang`
  - Turtle/Graph : `"Value"@{lang}`

Datentypen und Sprachangaben

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <rdf:RDF xmlns="http://bsp.at/x#"
3           ...>
4  <People rdf:ID="Staff"> <member>
5       <Person rdf:ID="Bori">
6          <desc xml:lang="de">Univ. Assistent</desc>
7          <desc xml:lang="en">Assistent Professor</desc>

9          <foaf:birthday
10            rdf:datatype="http://.../XMLSchema#date">
11               1980-08-12
12         </foaf:birthday>
13      </Person>
14 </member></People></rdf:RDF>
```


Datentypen und Sprachangaben

```
1  @prefix aw:   <http:/bsp.at/x#>.
2  @prefix rdf:  <http://.../22-rdf-syntax-ns#>.
3  @prefix foaf: <http://xmlns.com/foaf/1.1/>.
4  @prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.

6  aw:Staff rdf:type  aw:People;
7           aw:member aw:Bori.

9  aw:Bori  rdf:type  aw:Person;
10          aw:desc   "Univ.Assistent"@de,
11                    "Assistent Professor"@en;
12          foaf:birthday  "1980-08-12"^^xsd:date.
```

Reification

- Problem: wie kann man eine Aussage über eine Aussage machen.
- Graphendarstellung: man müsste auf eine Kante des Graphen verweisen
- Lösung: Refication - “Vergegenständlichung” des Statements

Reification

```
1  @prefix aw:   <http:/bsp.at/x#>.
2  @prefix foaf: <http://xmlns.com/foaf/1.1/>.
3  @prefix rdf:  <http://.../22-rdf-syntax-ns#>.

5  aw:Bori  foaf:birthday  "1980-08-12".

7  _:1      rdf:subject    aw:Bori;
8           rdf:predicate  foaf:birthday;
9           rdf:object     "1980-08-12";
10          rdf:type       rdf:Statement.

12 _:1      aw:claims      <http://x.org/n#Ana>.
```


Semantik von Aussagen

falsch (Ersteller der Ressource ist ein String?):

```
1  <http://vienna.iaeste.at/>
2      dc:creator "Petra Wagner" .
```

besser:

```
1  <http://vienna.iaeste.at/>
2     dc:creator _:dw .
3  _:dw foaf:name "Petra Wagner" .
```


XML/RDF - Vorteile

- XML/RDF Dokumente sind portabler
  $\rightarrow$ Klassen, Eigenschaften, Ressourcen
- fertige Struktur für das Design von XML-Dokumenten
- kombiniert die Vorteile von XML und RDF

XML/RDF - Nachteile

- Einschränkungen beim Design der XML-Dokumente
- zusätzlicher Aufwand: RDF-Vokabular/Konzepte