RDF: Fortgeschrittene Themen

Markus Stocker

7. Mai 2018

Rekapitulation

- Was ist ein RDF Tripel?
- Wozu benötigt man eine Serialisierung?
- Welche RDF Syntax liest sich am besten?
- Welche Gründe sprechen für RDF/XML?

Übersicht

- Das Prädikat `rdf:type`
- Datentypen
- Angabe zu natürlicher Sprache (*language tag*)
- Listen in RDF
- Reifizierung (*reification*)

Das Prädikat `rdf:type`

- Das `rdf:type` Prädikat ist Teil des RDF Vokabular
- Es wird benutzt um einem URI einen Typ zuzuordnen
- Die URI referenzierte Ressource gehört dem entsprechenden Typ
- In Turtle auch mit 'a' kürzbar
- Solche Typisierung im Semantischen Web von zentraler Bedeutung
- Mehr dazu in RDF Schema

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

ex:Earth rdf:type ex:Planet .

ex:Mars a ex:Planet .
```

Datentypen

- Literale werden grundsätzlich als Zeichenfolge interpretiert
- Dies ist in praktischen Anwendungen unzureichend
- Man benötig weit mehr an Datentypen, z.B. Nummern oder Zeiten
- Datentyp hat Auswirkungen auf die Interpretation eines Wertes
- Klassisches Beispiel: “02”, “2”, “20“
- Sortierung als Nummern oder Zeichenfolgen ist unterschiedlich

Datentypen

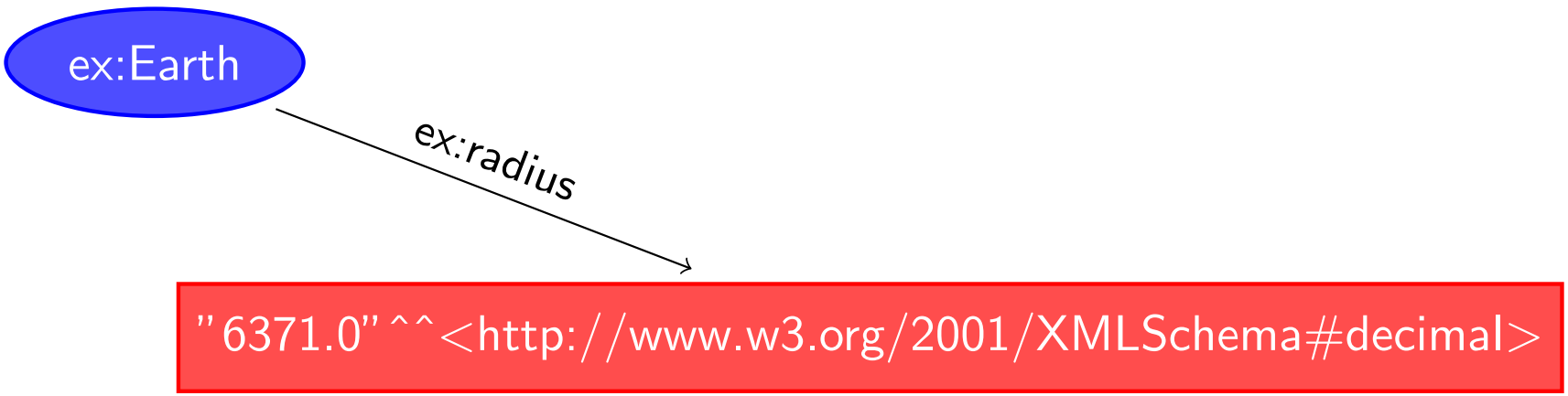
- RDF Literale können Datentyp explizit angeben
- Solche Literale werden *typisierte* Literale genannt
- Datentypen sind mittels URI identifiziert
- Beispiele aus XML Schema
  - `http://www.w3.org/2001/XMLSchema#string`
  - `http://www.w3.org/2001/XMLSchema#date`
  - `xsd:int`
- XML Schema Datetypen sind weit verbreitet (für elementare Typen)
- Datentypen können allerdings beliebig erweitert werden
- Beispiel: `http://www.opengis.net/ont/geosparql#wktLiteral`
- Bedeutet nicht, dass diese von einer Software auch unterstützt werden
- Selbst XML Schema Datentypen nicht zwingend unterstützt

Datentypen

- Syntaktisch ungleiche Literale können semantisch gleich sein
- Datentypen ermöglichen solch differenzierte Handhabung
- Beispiel
  - Die Literale `3.14`, `+03.14`, `3.140` sind syntaktisch ungleich
  - Als untypisierte Literale werden diese ungleich behandelt
  - Als typisierte Literale (`xsd:decimal`) sind sie semantich gleich
  - Somit ist `decimal("3.14") == decimal("3.140")` wahr

Datentypen

- Typisierte Literale müssen entsprechend Serialisiert werden
- In graphischer Darstellung wird meist "..."^^<...> verwendet
- Notation auch von Turtle und N-Triples Syntaxen verwendet



Datentypen: Turtle und N-Triples

```
@prefix ex: <http://example.org#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Earth ex:radius "6371.0"^^xsd:decimal ;
  ex:label "Earth"^^<http://www.w3.org/2001/XMLSchema#string> .
```

```
<http://example.org#Earth>
  <http://example.org#radius>
    "6371.0"^^<http://www.w3.org/2001/XMLSchema#decimal> .
```

Datentypen: RDF/XML

```
<!DOCTYPE rdf:RDF[
  <!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#'>
  <!ENTITY ex 'http://example.org#'>
]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:ex="http://example.org#">

  <rdf:Description rdf:about="&ex;Earth">
    <ex:radius rdf:datatype="&xsd;decimal">6371.0</ex:radius>
  </rdf:Description>

</rdf:RDF>
```

Der RDF Datentyp *XMLLiteral*

- `rdf:XMLLiteral` ist ein in RDF eingebauter Datentyp
- Einbindung von XML als Werte in RDF Literale

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:ex="http://example.org#">

  <rdf:Description rdf:about="http://example.org#Earth">
    <ex:radius rdf:parseType="Literal">
      <length unit="km">6371.0</length>
    </ex:radius>
  </rdf:Description>

</rdf:RDF>
```

Angabe zu Natürlicher Sprache (*language tag*)

- Untypisierte Literale können eine Sprachangabe haben
- Wie der Datentyp ist auch dieser *tag* Teil des Literals
- Sprich es wird kein weiteres Tripel dafür benötigt
- Die Sprachangabe ist für typisierte Literale nicht erlaubt
- Typisierte Literale gelten als Sprachunabhängig

Angabe zu Natürlicher Sprache (*language tag*)

```
@prefix ex: <http://example.org#> .

ex:Earth ex:label "Earth"@en, "Erde"@de, "Terra"@it .
```

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:ex="http://example.org#">

  <rdf:Description rdf:about="http://example.org#Earth">
    <ex:label xml:lang="en">Earth</ex:label>
    <ex:label xml:lang="de">Erde</ex:label>
    <ex:label xml:lang="it">Terra</ex:label>
  </rdf:Description>

</rdf:RDF>
```

Quiz: Wieviele Tripel?

```
@prefix ex: <http://example.org#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Earth ex:label "Earth", "Earth"@en, "Earth"^^xsd:string .
```

Listen in RDF

- Daten werden oft in listenartige Strukturen organisiert
- RDF stellt dafür verschiedene Konstrukte zur Verfügung
- Für offene (*containers*) und geschlossene (*collections*) Listen
- Es handelt sich hier nur um kürzungen für RDF Graphen
- Also *"syntactic sugar"* für eine ansonsten (etwas) längere Form

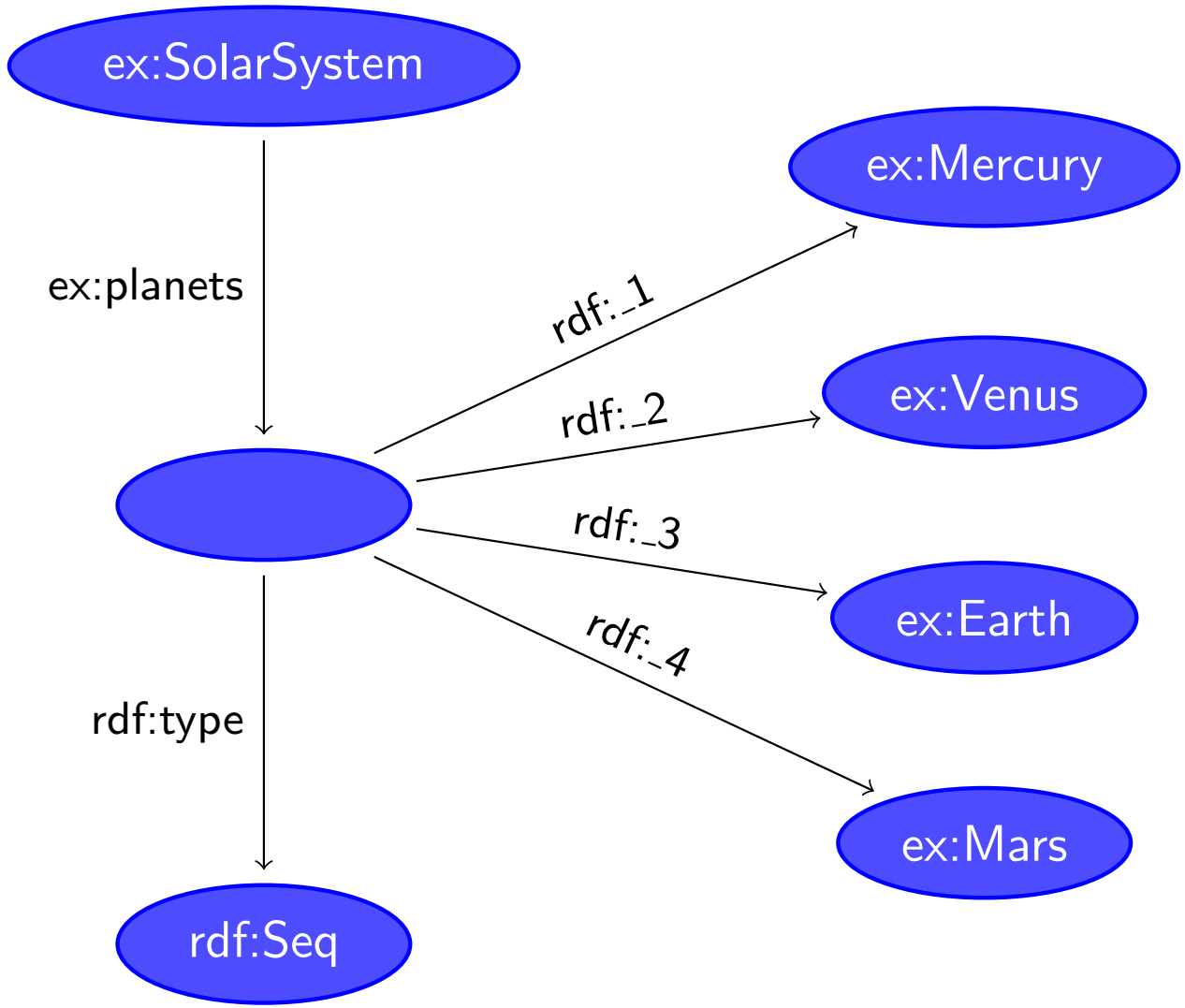
Offene Listen: *Containers*

- Es gibt drei Arten von *container*
  - *rdf:Seq*: Geordnete Liste
  - *rdf:Bag*: Ungeordnete Liste
  - *rdf:Alt*: Liste an alternativen
- Diese Konstrukte haben nur informelle Semantik (Bedeutung)
- Eine Anwendung kann die zusätzliche Information wahrnehmen
- Die Anwendung muss das aber nicht

Offene Listen: Beispiel, RDF/XML, Spezielle Syntax

```
<rdf:Description rdf:about="http://example.org#SolarSystem">
  <ex:planets>
    <rdf:Seq>
      <rdf:li rdf:resource="http://example.org#Mercury"/>
      <rdf:li rdf:resource="http://example.org#Venus"/>
      <rdf:li rdf:resource="http://example.org#Earth"/>
      <rdf:li rdf:resource="http://example.org#Mars"/>
    </rdf:Seq>
  </ex:planets>
</rdf:Description>
```

Offene Listen: Beispiel, Visuell, Ungekürzte Form



Offene Listen: Beispiel, Turtle, Keine Spezielle Syntax

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

ex:SolarSystem
  ex:planets [
    a rdf:Seq ;
    rdf:_1 ex:Mercury ;
    rdf:_2 ex:Venus ;
    rdf:_3 ex:Earth ;
    rdf:_4 ex:Mars
  ] .
```

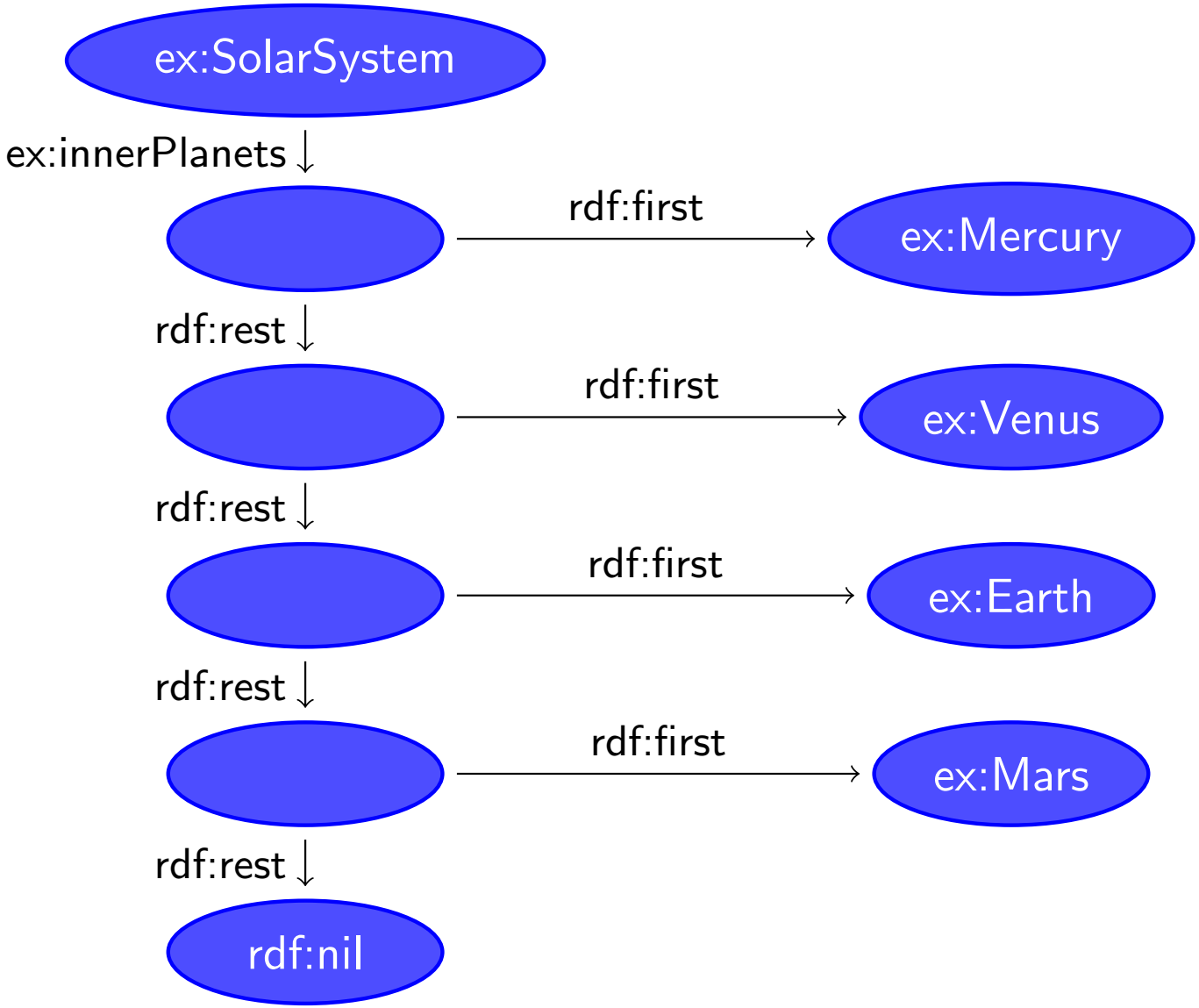
Geschlossene Listen: *Collections*

- Bei offenen Listen ist eine explizite Schliessung nicht möglich
- Man kann immer weitere Elemente hinzufügen (`rdf:_n`)
- Für geschlossene Listen stellt RDF *collections* zur Verfügung
- Wie bei offenen Listen, handelt es sich auch hier um Kurzformen
- Sprich kurzgefasstere RDF Serialisierungen

Geschlossene Listen: Beispiel, RDF/XML, Spezielle Syntax

```
<rdf:Description rdf:about="http://example.org#SolarSystem">
  <ex:innerPlanets rdf:parseType="Collection">
    <rdf:Description rdf:about="http://example.org#Mercury"/>
    <rdf:Description rdf:about="http://example.org#Venus"/>
    <rdf:Description rdf:about=="http://example.org#Earth"/>
    <rdf:Description rdf:about=="http://example.org#Mars"/>
  </ex:innerPlanets>
</rdf:Description>
```

Geschlossene Listen: Beispiel, Visuell, Ungekürzte Form



Geschlossene Listen: Beispiel, Turtle, Spezielle Syntax

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

ex:SolarSystem
  ex:innerPlanets (
    ex:Mercury ex:Venus ex:Earth ex:Mars
  ) .
```

Reifizierung (*reification*)

- Aussagen über Aussagen
- Oder: Wenn die Aussage zum Gegenstand wird
- RDF: Wie sagt man etwas über ein Tripel aus?
- Dies wird mit Reifizierung ermöglicht
- Beispiel
  - Eratosthenes schätzte den Erdradius auf 7018 km
  - Teilaussage wie gehabt: `ex:Earth ex:radius "7018"`
  - Aussage: `ex:Eratosthenes ex:estimated ?`

Reifizierung: Beispiel (Turtle)

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

ex:Eratosthenes ex:estimated [
  rdf:type rdf:Statement ;
  rdf:subject ex:Earth ;
  rdf:predicate ex:radius ;
  rdf:object "7018"
] .
```

Reifizierung: Beispiel (RDF/XML)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:ex="http://example.org#">

  <rdf:Description rdf:about="http://example.org#Eratosthenes">
    <ex:estimated>
      <rdf:Statement>
        <rdf:subject rdf:resource="http://example.org#Earth"/>
        <rdf:predicate rdf:resource="http://example.org#radius"/>
        <rdf:object>7018</rdf:object>
      </rdf:Statement>
    </ex:estimated>
  </rdf:Description>

</rdf:RDF>
```

Reifizierung: Bemerkungen

- Ein reifiziertes Tripel ist keine Aussage über dessen Gültigkeit
- Das Tripel selbst folgt aus der Reifizierung nicht
- Es folgt nicht, dass der Erdradius 7018 km ist
- Was Sinn macht, denn dies war die Schätzung von Eratosthenes
- Die in der Tat etwa um etwa 10% falsch war

Zusammenfassung

- Das wichtige Prädikat `rdf:type`
- Der Zweck wird im Kapitel RDF Schema nochmals deutlicher
- Wie auch schon für XML, sind Datentypen auch in RDF wichtig
- Das *language tag* für untypisierte Literale
- Listen und Reifizierung in RDF