RDF AND SPARQL

Part I: Basics of RDF

Sebastian Rudolph
ICCL Summer School

Dresden, August 2013
Agenda

- RDF – Motivation
- URIs
- RDF data model
- Syntax for RDF: Turtle and XML
- Datatypes
- Multi-Valued Relationships
- Blank Nodes
- Lists
- RDF in Practice
Agenda

- RDF – Motivation
- URIs
- RDF data model
- Syntax for RDF: Turtle and XML
- Datatypes
- Multi-Valued Relationships
- Blank Nodes
- Lists
- RDF in Practice
The Semantic Web approach aims at presenting information on the Web in a (semi)structured machine-processable way.

Why not use XML?
Disadvantages of XML

- tag names ambiguous
- tree structure not optimal for
  - intuitive description of the data
  - information integration
- Example: how to encode in a tree the sentence:
  “The book ‘Semantic Web – Grundlagen’ was published by Springer-Verlag.”
Modeling Problems in XML

“The book ‘Semantic Web – Grundlagen’ was published by Springer-Verlag.”
Modeling Problems in XML

“The book ‘Semantic Web – Grundlagen’ was published by Springer-Verlag.”

<Published>
  <Publisher>Springer-Verlag</Publisher>
  <Book>Semantic Web -- Grundlagen</Book>
</Published>
Modeling Problems in XML

“The book ‘Semantic Web – Grundlagen’ was published by Springer-Verlag.”

<Published>
  <Publisher>Springer-Verlag</Publisher>
  <Book>Semantic Web -- Grundlagen</Book>
</Published>

<Publisher Name="Springer-Verlag">
  <Published Book="Semantic Web -- Grundlagen"/>
</Publisher>
Modeling Problems in XML

“The book ‘Semantic Web – Grundlagen’ was published by Springer-Verlag.”

<Published>
  <Publisher>Springer-Verlag</Publisher>
  <Book>Semantic Web -- Grundlagen</Book>
</Published>

<Publisher Name="Springer-Verlag">
  <Published Book="Semantic Web -- Grundlagen"/>
</Publisher>

<Book Name="Semantic Web -- Grundlagen">
  <Publisher Publisher="Springer-Verlag"/>
</Book>
Agenda

- RDF – Motivation
- URIs
- RDF data model
- Syntax for RDF: Turtle and XML
- Datatypes
- Multi-Valued Relationships
- Blank Nodes
- Lists
- RDF in Practice
URIs – Idea

- **URI** = Uniform Resource Identifier
- serve for denoting resources in a world-wide unique way
- a resource can be any object that has (in the context of a given application) a clear identity (e.g. books, cities, persons, publishers, relations between those, abstract concepts etc.)
- in certain domains, something similar already exists: ISBN number for books
URIs – Syntax

• extension of the notion of URLs; not every URI relates to a Web document but mostly a Web document is referred to by using its URL as URI

• starts with the so-called URI schema, which is separated by a colon (:) from the subsequent part (e.g.: http, ftp, mailto)

• URIs often hierarchically structured
Agenda

- RDF – Motivation
- URIs
- RDF data model
- Syntax for RDF: Turtle and XML
- Datatypes
- Multi-Valued Relationships
- Blank Nodes
- Lists
- RDF in Practice
RDF: Graphs instead of Trees

Solution: Representation as (directed) Graphs

http://example.org/SemanticWeb
ex:publishedBy
http://springer.com/Publisher
General Remarks about RDF

• “Resource Description Framework”
• W3C Recommendation (http://www.w3.org/RDF)
• currently being revised
• RDF is a data model
  – originally: assign metadata to Web resources, later more general usage
  – encodes structured information
  – universal, machine-readable exchange format
Constituents of RDF Graphs

- URIs
  - for uniquely referencing resources
  - (already discussed at XML lecture)
- literals
  - describe data values that do not have an independent existence
- blank nodes
  - allow for stating the existence of some individual (and describing its properties) without giving it a name
Literals

- for representing data values
- noted as strings
- interpreted by an associated datatype
- literals without datatype are treated like strings
Graph as a Set of Triples

- there are several different ways to represent graphs
- we use list of (node-edge-node) triples
RDF Triple

Constituents of an RDF triple

- terms inspired by linguistics but not always an exact match
- permitted occurrences of constituents:
  - subject: URI or blank node
  - predicate: URI (also called property)
  - object: URI or blank node or literal
- node and edge labels are unique, thus the original graph can be reconstructed from the list of triples
Agenda

- RDF – Motivation
- URIs
- RDF data model
- Syntax for RDF: Turtle and XML
- Datatypes
- Multi-Valued Relationships
- Blank Nodes
- Lists
- RDF in Practice
Simple Syntax for RDF

- direct enumeration of triples:
  - N3: “Notation 3” – comprehensive formalism
  - N-Triples: fraction of N3
  - Turtle: extension of N-Triples (by abbreviations)

- Turtle syntax:
  - URIs in angular brackets
  - literals in quotes
  - tripels terminated by full stop
  - spaces and line breaks outside such delimiters are ignored
Turtle Syntax: Abbreviations

Example

  <http://springer.com/Publisher> .
  "Semantic Web -- Grundlagen" .
<http://springer.com/Verlag> <http://ex.org/Name>  
  "Springer Verlag" .

In Turtle we can define prefix abbreviations:

@prefix ex:  <http://ex.org/> .
@prefix springer:  <http://springer.com/> .
ex:SemanticWeb ex:publishedBy springer:Publisher .
ex:SemanticWeb ex:Title "Semantic Web -- Grundlagen" .
springer:Publisher ex:Name "Springer Verlag" .
Turtle Syntax: Abbreviations

Multiple triples with the same subject can be grouped:

```turtle
@prefix ex: <http://ex.org/> .
@prefix springer: <http://springer.com/> .

ex:SemanticWeb ex:publishedBy springer:Publisher ;
    ex:Title "Semantic Web -- Grundlagen" .

springer:Publisher ex:Name "Springer Verlag" .
```
Turtle Syntax: Abbreviations

Multiple triples with the same subject can be grouped:

```turtle
@prefix ex: <http://ex.org/> .
@prefix springer: <http://springer.com/> .

ex:SemanticWeb ex:publishedBy springer:Publisher ;
    ex:Title "Semantic Web -- Grundlagen" .
springer:Publisher ex:Name "Springer Verlag" .
```

Likewise triples with coinciding subject and predicate:

```turtle
@prefix ex: <http://ex.org/> .

ex:SemanticWeb ex:Author ex:Hitzler, ex:Krötzsch,
    ex:Rudolph, ex:Sure ;
    ex:Titel "Semantic Web -- Grundlagen" .
```
XML Syntax of RDF

- Turtle intuitively understandable, machine-processable
- yet, better tool support and available libraries for XML
- thus: XML syntax more wide-spread
XML Syntax of RDF

- like in XML, name spaces are used in order to disambiguate tag names
- RDF-specific tags have a predefined name space, by convention abbreviated with ‘rdf’

```xml
<?xml version="1.0" encoding="utf-8"?>
<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/SemanticWeb">
    <ex:publishedBy>
      <rdf:Description rdf:about="http://springer.com/Publisher"/>
    </ex:publishedBy>
  </rdf:Description>
</rdf:RDF>
```
XML Syntax of RDF

- the rdf:Description element encodes the subject (the URI of which is stated as the value of the associated rdf:about attribute)
- every element directly nested into an rdf:Description element denotes a predicate (the URI of which is the element name),
- predicate elements in turn contain the triple’s object as rdf:Description element

```
<rdf:Description rdf:about="http://example.org/SemanticWeb">
  <ex:publishedBy>
    <rdf:Description rdf:about="http://springer.com/Publisher"/>
  </ex:publishedBy>
</rdf:Description>
```
XML Syntax of RDF

```xml
<?xml version="1.0" encoding="utf-8"?>
<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/SemanticWeb">
        <ex:publishedBy>
            <rdf:Description rdf:about="http://springer.com/Publisher"/>
        </ex:publishedBy>
    </rdf:Description>
</rdf:RDF>
```

Diagram:
```
ex:SemanticWeb -- ex:publishedBy --> springer:Publisher
```
XML Syntax of RDF

<?xml version="1.0" encoding="utf-8"?>
<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/SemanticWeb">
        <ex:publishedBy>
            <rdf:Description rdf:about="http://springer.com/Publisher"/>
        </ex:publishedBy>
    </rdf:Description>
</rdf:RDF>
<?xml version="1.0" encoding="utf-8"?>
<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/SemanticWeb">
    <ex:publishedBy>
      <rdf:Description rdf:about="http://springer.com/Publisher"/>
    </ex:publishedBy>
  </rdf:Description>
</rdf:RDF>
XML Syntax of RDF

```xml
<?xml version="1.0" encoding="utf-8"?>
<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/SemanticWeb">
    <ex:publishedBy>
      <rdf:Description rdf:about="http://springer.com/Publisher"/>
    </ex:publishedBy>
  </rdf:Description>
</rdf:RDF>
```
**XML Syntax of RDF**

- untyped literals can be included as free text into the predicate element
- condensed forms admissible:
  - one subject containing several property elements
  - one object description serves as subject for another triple

```xml
<rdf:Description rdf:about="http://example.org/SemanticWeb">
  <ex:Title>Semantic Web -- Grundlagen</ex:Title>
  <ex:publishedBy>
    <rdf:Description rdf:about="http://springer.com/Publisher">
      <ex:Name>Springer Verlag</ex:Name>
    </rdf:Description>
  </ex:publishedBy>
</rdf:Description>
```
XML Syntax of RDF

- alternative (but semantically equivalent) representation of literals as XML attributes
- property URIs are then used as attribute names
- object URIs can be given as value of the rdf:resource attribute inside a property tag

```xml
<rdf:Description rdf:about="http://example.org/SemanticWeb"
    ex:Title="Semantic Web -- Grundlagen">
    <ex:publishedBy rdf:resource="http://springer.com/Publisher"/>
</rdf:Description>
<rdf:Description rdf:about="http://springer.com/Verlag"
    ex:Name="Springer Verlag"/>
```
RDF/XML Syntax: Complications

- name spaces are needed (not just for abbreviation reasons), because colons inside XML elements and attributes are always interpreted as name space delimiters

- problem: in XML, no name spaces in attribute values allowed (would be interpreted as URI schema), thus we cannot write:
  \[
  \text{rdf:about="ex:SemanticWeb"}
  \]

- “work around” via XML entities:
  Declaration:
  \[
  \text{<!ENTITY ex 'http://example.org/'}
  \]
  Usage:
  \[
  \text{rdf:resource="&ex;SemanticWeb"}
  \]
RDF/XML Syntax: Base URIs

- usage of base URIs:

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

   <rdf:Description rdf:about="SemanticWeb">
      <ex:publishedBy
         rdf:resource="http://springer.com/Publisher"/>
   </rdf:Description>

</rdf:RDF>

- relative URIs (i.e. those that are to be preceded by the given base URI) are recognized by the absence of a schema part
Agenda

• RDF – Motivation
• URIs
• RDF data model
• Syntax for RDF: Turtle and XML
• Datatypes
• Multi-Valued Relationships
• Blank Nodes
• Lists
• RDF in Practice
Datatypes

Example: `xsd:decimal`

For `xsd:decimal` holds "3.14" = "+03.14"
But not for `xsd:string`!
Datatypes in RDF

- by now: untyped literals, treated like strings (e.g.: "02" < "100" < "11" < "2")
- typing allows for a better (more semantic = meaning-adequate) handling of values
- datatypes are themselves denoted by URIs and can essentially be freely chosen
- common: usage of xsd datatypes
- syntax: "datavalue"^^datatype_URI
Datatypes in RDF – Example

Graph:

```
http://springer.com/Publisher

http://example.org/Name

"Springer Verlag"^^http://www.w3.org/2001/XMLSchema#string

"1842-05-10"^^http://www.w3.org/2001/XMLSchema#date
```

Turtle:
```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://springer.com/Publisher>
  <http://example.org/Name> "Springer Verlag"^^xsd:string ;
  <http://example.org/foundation\_date> "1842-05-10"^^xsd:date .
```

XML:
```
<rdf:Description rdf:about="http://springer.com/Publisher">
  <ex:Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Springer Verlag</ex:Name>
  <ex:foundation\_date rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1842-05-10</ex:foundation\_date>
</rdf:Description>
```
XML Schema Datatypes

- string
- normalizedString
- token
  - language
    - NCName
    - NMTOKEN
  - Name
  - NMTOKEN
- ID
  - IDREF
  - ENTITY
  - IDREFS
  - ENTITIES

- ur types
- built-in primitive types
- built-in derived types
- complex types

- derived by restriction
- derived by list
- derived by extension or restriction
XML Schema Datatypes

- decimal
- integer
  - nonPositiveInteger
  - long
  - nonNegativeInteger
  - negativeInteger
  - int
  - unsignedLong
  - positiveInteger
  - short
  - unsignedInt
  - byte
  - unsignedShort
  - unsignedByte

- ur types
- built-in primitive types
- built-in derived types
- complex types

- derived by restriction
- derived by list
- derived by extension or restriction
XML Schema – Canonical Values

- there may be several lexical forms for one value
- one of these is picked as the value’s canonical form
- useful to detect equivalence between different notations of the same values
- the following lexical forms of the datatype `decimal` represent the same value: 100.5, +100.5, 0100.5, 100.50, 100.500, 100.5000, the canonical variant is: 100.5
The Predefined Datatype

- `rdf:XMLLiteral` is the only datatype that is pre-defined within the RDF standard
- denotes arbitrary balanced XML snippets
- in RDF/XML special syntax for unambiguous representation:

```xml
<rdf:Description rdf:about="http://example.org/SemanticWeb">
  <ex:Titel rdf:parseType="Literal">
    <b>Semantic Web</b><br />
    Grundlagen
  </ex:Titel>
</rdf:Description>
```
Language Information and Datatypes

- Language information can only be provided for untyped literals

Example:

**XML:**

```xml
<rdf:Description rdf:about="http://springer.com/Publisher">
  <ex:Name xml:lang="de">Springer Verlag</ex:Name>
  <ex:Name xml:lang="en">Springer Science+Business Media</ex:Name>
</rdf:Description>
```

**Turtle:**

```
<http://springer.com/Publisher> <http://example.org/Name>
```
Language Information and Datatypes

According to the spec, the following literals are all different from each other:

@prefix xsd: <http://www.w3.org/2001/XMLSchema#>
<http://springer.com/Verlag> <http://example.org/Name>
"Springer Verlag", "Springer Verlag"@de,
"Springer Verlag"^^xsd:string .

In practice they are, however, often implemented as equal.
Agenda

- RDF – Motivation
- URIs
- RDF data model
- Syntax for RDF: Turtle and XML
- Datatypes
- Multi-Valued Relationships
- Blank Nodes
- Lists
- RDF in Practice
Multi-Valued Relationships

- Cooking with RDF:
  “For the preparation of Chutney, you need 1 lb green mango, a teaspoon Cayenne pepper, . . .”

- first modeling attempt:
  ```rdfs
  @prefix ex: <http://example.org/> .
  ex:Chutney ex:hasIngredient "1lb green mango",
      "1 tsp. Cayenne pepper",
      ...
  ```

- Not satisfactory: ingredients plus amounts encoded as one string. Search for recipes containing green mango not possible (or difficult).
Multi-Valued Relationships

- Cooking with RDF:
  “For the preparation of Chutney, you need 1 lb green mango, a teaspoon Cayenne pepper, ...”

- second modeling attempt:
  ```
  @prefix ex: <http://example.org/> .
  ex:Chutney ex:hasIngredient ex:greenMango;
     ex:amount "1 lb";
  ex:hasIngredient ex:CayennePepper;
     ex:amount "1 tsp.";
  ...
  ```

- Even worse: no unique assignment of ingredient and amounts possible.
Multi-Valued Relationships

- Problem: we have a proper three-valued (aka: ternary) relationship (cf. databases)

<table>
<thead>
<tr>
<th>dish</th>
<th>ingredient</th>
<th>amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>chutney</td>
<td>geen mango</td>
<td>1 lb</td>
</tr>
<tr>
<td>chutney</td>
<td>Cayenne pepper</td>
<td>1 tsp.</td>
</tr>
</tbody>
</table>

- Direct representation in RDF not possible
- Solution: introduction of auxiliary nodes
Multi-Valued Relationships

auxiliary nodes in RDF:

- as graph

  - Turtle syntax (using rdf:value for the primary component)

```turtle
@prefix ex: <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
ex:chutneyIngredient1 rdf:value ex:greenMango;
ex:amount "1 lb" .
```

TU Dresden, August 2013

RDF and SPARQL

slide 51 of 68
Agenda

- RDF – Motivation
- URIs
- RDF data model
- Syntax for RDF: Turtle and XML
- Datatypes
- Multi-Valued Relationships
- Blank Nodes
- Lists
- RDF in Practice
Blank Nodes

auxiliary nodes in RDF:

- blank nodes (aka bnodes) can be used for resources that need not be named (e.g. auxiliary nodes)
- can be interpreted as existential statement
- syntax (as graph):

```
http://example.org/chutney
http://example.org/greenMango
http://example.org/hasIngredient
http://example.org/ingredient
http://example.org/amount
"1 lb"
```
Blank Nodes

RDF/XML-Syntax:

```xml
<rdf:Description rdf:about="http://example.org/chutney">
  <ex:hatZutat rdf:nodeID="id1" />
</rdf:Description>

<rdf:Description rdf:nodeID="id1">
  <ex:ingredient rdf:resource="http://example.org/greenMango" />
  <ex:amount>1 lb<ex:amount/>
</rdf:Description>

abbreviated:

```xml
<rdf:Description rdf:about="http://example.org/chutney">
  <ex:hasIngredient rdf:parseType="Resource">
    <ex:ingredient rdf:resource="http://example.org/greenMango" />
    <ex:amount>1 lb<ex:amount/>
  </ex:hasIngredient>
</rdf:Description>
```
Blank Nodes

Turtle syntax:

@prefix ex: <http://example.org/> .
ex:chutney ex:hasIngredient _:id1 .
_:id1 ex:ingredient ex:greenMango ;
ex:amount "1 lb" .

abbreviated:

@prefix ex: <http://example.org/> .
ex:chutney ex:hasIngredient [
ex:ingredient ex:greenMango ;
ex:amount "1 lb" ] .
Agenda

- RDF – Motivation
- URIs
- RDF data model
- Syntax for RDF: Turtle and XML
- Datatypes
- Multi-Valued Relationships
- Blank Nodes
- Lists
- RDF in Practice
Lists

- General data structures for enumerating arbitrary many resources (where order is relevant), e.g. authors of a book
- Distinction between
  - Open lists (containers)
    new entries can be added
  - Closed lists (collections)
    new entries cannot be added
- These structures are modeled using the already discussed means of representation, i.e. no additional expressivity!
Open Lists (Container)

graph:

abbreviated in RDF/XML:

```xml
<rdf:Description rdf:about="http://example.org/SemanticWeb">
  <ex:authors>
    <rdf:Seq>
      <rdf:li rdf:resource="http://example.org/Hitzler />
      <rdf:li rdf:resource="http://example.org/Krötzsch />
      <rdf:li rdf:resource="http://example.org/Rudolph />
      <rdf:li rdf:resource="http://example.org/Sure />
    </rdf:Seq>
  </ex:authors>
</rdf:Description>
```
Types of Open Lists

Via `rdf:type` the a list type is assigned to the root node of the list:

- `rdf:Seq`
  ordered list (sequence)

- `rdf:Bag`
  unordered set
  indicates that the encoded order is irrelevant

- `rdf:Alt`
  set of alternatives
  normally only one entry will be relevant
Closed Lists (Collections)

Graph:

http://example.org/Autoren

http://www.w3.org/1999/02/22-rdf-syntax-ns#rest

http://www.w3.org/1999/02/22-rdf-syntax-ns#rest

http://www.w3.org/1999/02/22-rdf-syntax-ns#rest

http://www.w3.org/1999/02/22-rdf-syntax-ns#first

http://example.org/SemanticWeb

http://example.org/Hitzler

http://example.org/Kröstzsch

http://example.org/Rudolph

http://example.org/Sure

http://www.w3.org/1999/02/22-rdf-syntax-ns#nil

underlying idea: recursive deconstruction of the list into head element and (possibly empty) rest list
Closed Lists (Collections)

RDF/XML-Syntax

```xml
<rdf:Description rdf:about="http://example.org/SemanticWeb">
  <ex:authors rdf:parseType="Collection">
    <rdf:Description rdf:about="http://example.org/Hitzler />
    <rdf:Description rdf:about="http://example.org/Krötzsch />
    <rdf:Description rdf:about="http://example.org/Rudolph />
    <rdf:Description rdf:about="http://example.org/Sure />
  </ex:authors>
</rdf:Description>
```

Turtle

```turtle
@prefix ex: <http://example.org/> .
ex:SemanticWeb ex:authors
  ( ex:Hitzler ex:Krötzsch ex:Rudolph ex:Sure ) .
```
Agenda

- RDF – Motivation
- URIs
- RDF data model
- Syntax for RDF: Turtle and XML
- Datatypes
- Multi-Valued Relationships
- Blank Nodes
- Lists
- RDF in Practice
Popularity of RDF

• today, a plethora of RDF tools exists
• there are libraries for virtually all programming languages
• freely available systems to work with large RDF data sets (so-called RDF Stores or Triple Stores)
• also commercial players (like Oracle) support RDF
• RDF is basis for other data formats: RSS 1.0, XMP (Adobe), SVG (vector graphics)
Assessment of RDF

- widely supported standard for data storage and interchange
- enables syntax-independent representation of distributed information via a graph-based data model
- pure RDF very oriented toward individuals
- few possibilities to encode schema knowledge
- → RDF Schema (next lecture)
RDFa – RDF-in-attributes

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
 "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
 xmlns:foaf="http://xmlns.com/foaf/0.1/"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 version="XHTML+RDFa 1.0" xml:lang="en">
<head>
<title>John’s Home Page</title>
<base href="http://example.org/john-d/" />
<meta property="dc:creator" content="Jonathan Doe" />
<link rel="foaf:primaryTopic"
 href="http://example.org/john-d/#me" />
</head>
<body about="http://example.org/john-d/#me">
  <h1>John’s Home Page</h1>
  <p>My name is <span property="foaf:nick">John D</span> and I like <a href="http://www.neubauten.org/" rel="foaf:interest" xml:lang="de">Einstürzende Neubauten</a>.</p>
</body>
RDFa – RDF Version

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" 
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
    <rdf:Description rdf:about="http://example.org/john-d/">
        <dc:creator xml:lang="en">Jonathan Doe</dc:creator>
        <foaf:primaryTopic>
            <rdf:Description rdf:about="http://example.org/john-d/#me">
                <foaf:nick xml:lang="en">John D</foaf:nick>
                <foaf:interest rdf:resource="http://www.neubauten.org/"/>
                <foaf:interest>
                    <rdf:Description rdf:about="urn:ISBN:0752820907">
                        <dc:creator xml:lang="en">Tim Berners-Lee</dc:creator>
                        <dc:title xml:lang="en">Weaving the Web</dc:title>
                    </rdf:Description>
                </foaf:interest>
            </rdf:Description>
        </foaf:primaryTopic>
    </rdf:Description>
</rdf:RDF>
Agenda

- RDF – Motivation
- URIs
- RDF data model
- Syntax for RDF: Turtle and XML
- Datatypes
- Multi-Valued Relationships
- Blank Nodes
- Lists
- RDF in Practice