Semantic Web Technologies
Resource Description Framework (RDF)
Semantic Web – Architecture

here be dragons...

Semantic Web Technologies (This lecture)

Technical Foundations

Berners-Lee (2009): Semantic Web and Linked Data
Overview

• A brief history of RDF
• Encodings of RDF
• Semantics and principles of RDF
• Embedding RDF in HTML – RDFa, Microdata, Microformats
• RDF Tools
• Examples for RDF in the wild
History: Metadata on the Web

- Goal: more effective rating and ranking of web contents, e.g., by search engines
  - Who has created this page?
  - When has it been changed the last time?
  - What is its topic?
  - Which is the content's license?
  - How does it relate to other pages?
Metadata on the Web: Dublin Core

- Developed in 1995 at a workshop in Dublin, Ohio
- 15 predefined tags
- A widely accepted standard (ISO 15836:2009)

- May be embedded into HTML:

```html
<html>
<head profile="http://dublincore.org/documents/2008/08/04/dc-html/">
    <title>Semantic Web</title>
    <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
    <meta name="DC.publisher" content="University of Mannheim" />
    <meta name="DC.subject" content="Semantic Web" />
    <meta name="DC.creator" content="Heiko Paulheim" />
    <meta name="DC.relation" content="http://www.w3.org/2001/sw" />
    ...
</head>
<body>
    ...
</body>
</html>
```
Metadata on the Web: Dublin Core

- Identifier
- Format
- Type
- Language
- Date
- Title
- Subject
- Coverage
- Description
- Creator
- Publisher
- Contributor
- Rights
- Source
- Relation
What is RDF?

• „Resource Description Framework“
• A W3C standard since 2004
• Description of arbitrary things

• View 1: Sentences in the form <subject, predicate, object>
  „Heiko works for University of Mannheim.“

• View 2: Directed graphs with labeled edges

Heiko \(\xrightarrow{\text{works for}}\) Uni Mannheim
Basic Building Blocks of RDF

- Resources
  - denote things
  - are identified by a URI
  - can have one or multiple types

- Literals
  - are values like strings or integers
  - can only be objects, not subjects or predicates
    (graph view: they can only have ingoing edges)
  - can have a datatype or a language tag (but not both)

- Properties (Predicates)
  - Link resources to other resources and to literals
Types

- All resources (not literals) can have a type
- Types can be arbitrarily defined
- The predefined predicate \texttt{rdf:type} defines the type of a resource

- Semantic Web is a lecture

\[\text{http://www.dws.informatik.uni-mannheim.de/sw} \quad \text{rdf:type} \quad \text{http://www.dws.informatik.uni-mannheim.de/Lecture}\]
Resource vs. Literal

• A literal is an atomic value
  – can only be object
  – i.e., a literal terminates always a graph

  ![Diagram](http://www.dws.informatik.uni-mannheim.de/sw)

  dc:creator → "Heiko Paulheim"

• A resource can be a subject itself

  ![Diagram](http://www.dws.informatik.uni-mannheim.de/sw)

  dc:creator → http://www.dws.informatik.uni-mannheim.de/heiko)
Datatypes for Literals

• (Almost) all XML Schema datatypes may be used
• Exception:
  – XML specific types
  – The underspecified type "duration"
  – sequence types
XML Specific Data Types vs. RDF

• XML: Datatypes for ID and IDREF
  
  ```xml
  <author id="0815">
    <name>Thomas Glavinic</name>
  </author>

  <book>
    <title>Die Arbeit der Nacht</title>
    <author idref="0815"/>
  </book>
  ```

• RDF: everything is identified by a URI
  
  ```xml
  <author rdf:about="http://mylibrary.org/author/Thomas_Glavinic">
    <name>Thomas Glavinic</name>
  </author>

    <title>Die Arbeit der Nacht</title>
  </book>
  ```
Language Tags for Literals

- Literals may be defined in different natural languages
  - "München"@de
  - "Munich"@en
- Those can be marked
- Note: the Semantic Web is multilingual!

- Language codes according to ISO 963
  - ISO 963-1 (1963): two-digit codes, 136 languages
  - if both are defined, ISO 963-1 has to be used!

Language Tags for Literals
Datatypes in RDF

• Examples:
  :Muenchen :hasName "München"@de .
  :Muenchen :hasName "Munich"@en .
  :Muenchen :hasPopulation "1356594"^^xsd:integer .
  :Muenchen :hasFoundingYear "1158-01-01"^^xsd:date .

• Note: there are no default datatypes (not even “string”!)
• These are three different literals:
  – "München"
  – "München"@de
  – "München"^^xsd:string .
Triple Notation

- Triples consist of a subject, predicate, and object
- An RDF document is an *unordered* set of triples

- **Simple triple:**
  
  `<http://www.dws.informatik.uni-mannheim.de/teaching/semantic-web>`  
  `<http://purl.org/dc/elements/1.1/relation>`  
  `<http://www.w3.org/2001/sw/>` .

- **Literal with language tag:**
  
  `<http://www.dws.informatik.uni-mannheim.de/teaching/semantic-web>`  
  `<http://purl.org/dc/elements/1.1/subject>`  
  "Semantic Web"@en .

- **Type literal:**
  
  `<http://www.dws.informatik.uni-mannheim.de/teaching/semantic-web>`  
  `<http://www.uni-mannheim.de/mhb/creditpoints>`  
  "6"^^<http://www.w3.org/2001/XMLSchema#integer> .
Turtle Notation

• A simplified triple notation

• Central definition of namespaces (and a default namespace):

```turtle
@prefix dc: <http://purl.org/dc/elements/1.1/>
@prefix : <http://www.dws.informatik.uni-mannheim.de/teaching/>
```

• Triples sharing the same subject or subject+predicate:

```turtle
:semantic-web dc:subject "Semantisches Web"@de ,
    "Semantic Web"@en ;
    dc:creator "Heiko Paulheim".
```

• Shorthand notation for `rdf:type`:

```turtle
:semantic-web a :lecture .
```
Notation RDF/XML

- A W3C standard since 2004
- Encodes RDF in XML
- Suitable for machine processing (plenty of XML tools!)

- Defining resources:
  
  `<rdf:Description rdf:about="http://dws.informatik.uni-mannheim.de/teaching/sw">
    <dc:creator>Heiko Paulheim</dc:creator>
  </rdf:Description>`

- Defining typed resources:
  
  `<rdf:Description rdf:about="http://dws.informatik.uni-mannheim.de/teaching/sw">
    <rdf:type rdf:resource="http://www.uni-mannheim.de/mhb/Lecture"/>
  </rdf:Description>`

- Alternative representation:
  
  `<mhb:Lecture rdf:about="http://dws.informatik.uni-mannheim.de/teaching/sw"
  xmlns:mhb="http://www.uni-mannheim.de/mhb/" />`
Notation RDF/XML

- **Relations between resources by nesting**

  ```xml
  <mhb:Lecture rdf:about="http://dws.informatik.uni-mannheim.de/teaching/sw">
    <mhb:givenBy>
      <mhb:Lecturer rdf:about="http://dws.informatik.uni-mannheim.de/heiko"/>
    </mhb:givenBy>
  </mhb:Lecture>
  ```

- **Relations between resources by explicit links**

  ```xml
  <mhb:Lecturer rdf:about="http://dws.informatik.uni-mannheim.de/heiko"/>

  <mhb:Lecture rdf:about="http://dws.informatik.uni-mannheim.de/teaching/sw">
    <mhb:givenBy rdf:resource="http://dws.informatik.uni-mannheim.de/heiko"/>
  </mhb:Lecture>
  ```
Notation RDF/XML

- An RDF graph may contain cycles
- XML may not → explicit links are necessary

```xml
<mhb:University rdf:about="http://www.uni-mannheim.de">
  <mhb:hasEmployee>
    <mhb:UniversityMember rdf:about="http://www.heikopaulheim.com/">
      <mhb:worksFor rdf:resource="http://www.uni-mannheim.de"/>
    </mhb:UniversityMember>
  </mhb:hasEmployee>
</mhb:University>
```
• JSON is popular in script programming
• JSON-LD: Standard for serializing RDF in JSON

```
{
  "@id": "http://www.heikopaulheim.com/",
  "http://dws.informatik.uni-mannheim.de/name": "Heiko Paulheim",
  "http://dws.informatik.uni-mannheim.de/teaches":
  {
    "http://http://www.w3.org/1999/02/22-rdf-syntax-ns#type":
      "http://dws.informatik.uni-mannheim.de/Lecture",
    "http://dws.informatik.uni-mannheim.de/title":
      "Semantic Web"
  }
}
```
Blank Nodes

• Information that is not or cannot be specified
  – "Dieter Fensel has written a book about the Semantic Web"
Blank Nodes

• Information that is not or cannot be specified
  – "Dieter Fensel has written something about the Semantic Web."

Dieter Fensel \( \text{dc:creator} \) "Semantic Web" \( \text{dc:subject} \)
Blank Nodes in Turtle

- **Variant 1: explicitly named with an underscore**
  ```turtle
  :Dieter_Fensel dc:creator _:x .
  _:x a :Book ;
      dc:subject "Semantic Web" .
  ```

- **Variant 2: unnamed with square brackets**
  ```turtle
  :Dieter_Fensel dc:creator
  [ a :Book;
      dc:subject "Semantic Web" ].
  ```

- **Notes:**
  - both are equivalent
  - changing blank node names does not change the semantics!
Application of Blank Nodes: n-ary Predicates

• RDF predicates always connect a subject and an object
  – i.e., in the sense of predicate logic, they are binary predicates
    
    \[
    :\text{Heiko} :\text{works\_for} :\text{UniMannheim} .
    \quad \Leftrightarrow \quad \text{works\_for}(\text{Heiko}, \text{UniMannheim}) .
    \]

• Sometimes, n-ary predicates are needed
  – \text{has\_ingredient}(\text{Recipe, Sugar, 100g})
Application of Blank Nodes: n-ary Predicates

:recipe :hasIngredient [ :ingredient :Sugar; :amount "100g" ] .
Application of Blank Nodes: n-ary Predicates


How does this differ from the version on the previous slide?
Reification

- Latin res ("Thing"), facere ("make")
  - an Explication
  - making a statement, an opinion etc. the subject of a statement
- In RDF: Statements about statements

- "Peter says that Rome is the capital of Spain."
  Implementation:
  - RDF Statements are considered resources themselves
  - Can be subject or object of other statements
- Reification can have multiple levels
  - “Peter says that Wikipedia states that Rome is the capital of Spain.”
Peter says Rome is capital of Spain
Implementing Reification as Standard RDF

Peter says: Rome is capital of Spain.
Encoding Reification in Turtle

• Variant 1: Named Statement (with URI)

```turtle
:triple1 rdf:type rdf:Statement ;
    rdf:subject :Rome ;
    rdf:predicate :isCapitalOf ;
    rdf:object :Spain .
:Peter :says :triple1 .
```

• Variant 2: Unnamed Statement (Blank Node)

```turtle
:Peter :says [a rdf:Statement ;
    rdf:subject :Rome ;
    rdf:predicate :isCapitalOf ;
    rdf:object :Spain .]
```

Semantic Principles of RDF

• On the Web, "anybody can say anything about anything"
  – This is called the AAA principle (Allemang & Hendler)

• This principle also holds for the Semantic Web
Semantic Principles of RDF

• One thing can have multiple names

: Munich : capitalOf : Bavaria .

• On the semantic web, there is not just one name for each thing
  – this is called the *Non-unique name assumption*

• This means: Just that two things have different names does not mean that they are different!
Let us consider the following example:

```plaintext
:Peter :fatherOf :Julia ,
     :Mary .
```

How many children does Peter have?

Intuitively, we assume that Julia and Mary are two different persons.
However, this is not trivial for a machine.
  (and the assumption may even be wrong)
Semantic Principles of RDF

• We (probably) do not know all the contents of the Semantic Web
• Therefore, there may be more information on a resource than what we have

• This principle is called "Open World Assumption"
RDF: Intuition and Actual Semantics

• Let us consider this example again:

:Peter :fatherOf :Julia ,
    :Mary .

• How many children does Peter have?

• Intuition says: two children
• However, he could also have three or more (oder also just one, as we have learned just a minute ago)
RDF: Intuition and Actual Semantics

- Both
  - Non-unique Name Assumption and
  - Open World Assumption

will re-occur quite a few times in this lecture

- Hint: consider those two
  whenever something seems weird
  when interpreting RDF data
RDF and HTML

- The Semantic Web uses RDF
- The “classic” Web uses HTML

- Does that mean that each information has to be encoded twice?
  - once for humans, once for machines

```html
<html>
  ...
  <b>Dr. Mark Smith</b>
  <i>Physician</i>
  Main St. 14
  Smalltown
  Mon-Fri 9-11 am
  Wed 3-6 pm
  ...
</html>
```

```rdfs
:p a :Physician .
:p :hasDegree "Dr." .
:p :hasName "Mark Smith" .
:p :hasAddress :a .
:a :street "Main Street" .
:a :number "14"^^xsd:int .
:a :city "Smalltown" .
:p :hasOpeningHours [ a rdf:Bag ;
  [ :day :Monday;
    :from "9"^^xsd:int;
    :to "11"^^xsd:int;
  ]
]
...
```
Using RDF and HTML Together – Variant 1

• Explicit reference to a RDF version
  – an agent stumbling on the HTML page can download the RDF data file

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:owl="http://www.w3.org/2002/07/owl#"
         xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
  <rdf:Description rdf:about="http://www.ke.tu-darmstadt.de/lehre/semantic-web">
    <dc:publisher>TU Darmstadt</dc:publisher>
    <dc:subject>Semantic Web</dc:subject>
    <dc:creator>Heiko Paulheim</dc:creator>
    <dc:relation rdf:resource="http://www.w3.org/2001/sw/"/>
  </rdf:Description>
</rdf:RDF>
```

```html
<html>
<head>
  <link rel="meta" type="application/rdf+xml" title="DC" href="dc.rdf"/>
</head>
<body>
  ...
</body>
</html>
```
Using RDF and HTML Together – Variant 2

• Content Negotiation
Content Negotiation in Detail

HTTP/1.1 303 See Other
Location: http://www.mannheim.de/index.html

GET / HTTP/1.1
Host: www.mannheim.de
Accept: text/html

Server

Address for HTML
HTTP Get

Browser
Content Negotiation in Detail

HTTP/1.1 303 See Other
Location: http://www.mannheim.de/data.rdf

GET / HTTP/1.1
Host: www.mannheim.de
Accept: application/rdf+xml
Content Negotiation: MIME Types

- MIME: Multipurpose Internet Mail Extensions
- Original purpose: classifying e-mail attachments
  - Text, PDF, ..
- First version: 1996
- Administred by IANA

- Important MIME types for the Semantic Web
  - application/rdf+xml
  - text/turtle
  - text/n3
  - application/sparql-query
  - application/sparql-results+xml
Using RDF and HTML Together

• Link to RDF Document
  – Can be done with a simple HTML editor
  – No special server configuration needed

• Content Negotiation
  – Requires particular server setup
  – One URI can be used for different representations

• Both cases require
  – two different representations
  – “double bookkeeping”

→ Potential source of inconsistencies!
RDF in Attributes (RDFa)

- Idea of RDFa
  - Why not encode HTML and RDF in *one* document
  - The essential information only has to be encoded once

- RDFa combines XHTML mit RDF

```html
<html>
  ...
  <body about="http://www.marcsmith.com/MarcSmith">
    <b><span property="doc:name">Dr. Mark Smith</span></b>
    <i><span property="doc:profession">Physician</span></i>
    <span rel="doc:address" href="http://www.marcsmith.com/Address">
      <span property="doc:street">Main Street</span>
      <span property="doc:number">14</span>
      <span property="doc:city">Smalltown</span>
    </span>
  ...
</body>
</html>
```
RDFa Language Constructs

- `about="http://foo.bar/aSubject"`
  - Defines the subject of a page or section

- `property = "http://foo.bar/aProperty"
  - Defines a relation
  - Contents of the tag are interpreted as a literal

- `rel = "http://foo.bar/aRelation"
  - Defines a relation to another resource

- `href = "http://foo.bar/aResource"
  - Defines a relation's object
  - can be the subject of a resource again

- `typeof = "http://foo.bar/aType"
  - defines a resource's type`
RDF in Attributes (RDFa)

```html
<html>
  ...
  <body about="http://www.marcsmith.com/MarcSmith">
    <b><span property="doc:name">Dr. Mark Smith</span></b>
    <i><span property="doc:profession">Physician</span></i>
    <span rel="doc:address" href="http://www.marcsmith.com/Address">
      <span property="doc:street">Main Street</span>
      <span property="doc:number">14</span>
      <span property="doc:city">Smalltown</span>
    </span>
  </body>
</html>
```
RDF in Attributes (RDFa)

<html>
  ...
  <body about="http://www.marcsmith.com/MarcSmith">
    <b><span property="doc:name">Dr. Mark Smith</span></b>
    <i><span property="doc:profession">Physician</span></i>
    <span rel="doc:address" href="http://www.marcsmith.com/Address">
      <span property="doc:street">Main Street</span>
      <span property="doc:number">14</span>
      <span property="doc:city">Smalltown</span>
    </span>
  </body>
  ...
</html>
RDF in Attributes (RDFa)

```
<html>
  ...
  <body about="http://www.marcsmith.com/MarcSmith">
    <b>Dr. Mark Smith</b>
    <i>Physician</i>
    <span rel="doc:address">
      <span>Main Street</span>
      <span>14</span>
      <span>Smalltown</span>
    </span>
  </body>
</html>
```

Relations without "href" become blank nodes!
Alternative to RDFa: Microdata

• Adding structured information to web pages
  – By marking up contents
  – Arbitrary vocabularies are possible
  – Introduced with HTML5

• Similar to RDFa

```html
<div itemscope
    itemtype="http://schema.org/PostalAddress">
  <span itemprop="name">Data and Web Science Group</span>
  <span itemprop="addressLocality">Mannheim</span>,
  <span itemprop="postalCode">68131</span>
  <span itemprop="addressCountry">Germany</span>
</div>
```
Alternative to RDFa: Microdata

- Markup can be extracted to RDF
  - See W3C Interest Group Note: Microdata to RDF [1]

```
<div itemscope
itemtype="http://schema.org/PostalAddress">
  <span itemprop="name">Data and Web Science Group</span>,
  <a href="http://schema.org/PostalAddress">_:1</a>,
  <a href="http://schema.org/name">_:1</a> "Data and Web Science Group",
  <a href="http://schema.org/addressLocality">_:1</a> "Mannheim",
  <a href="http://schema.org/postalCode">_:1</a> "68131",
  <a href="http://schema.org/adressCounty">_:1</a> "Germany".
```

Alternative to RDFa: Microdata

• Commonalities
  – Arbitrary classes/predicates are possible
  – Although Microdata is mainly used with schema.org

• Differences
  – Microdata is slightly less expressive
  – No URIs, only blank nodes
  – No cycles in the RDF graph
  – No reification
RDFa, MicroFormats, and Microdata

- MicroFormats: fixed vocabularies for persons, addresses, etc.

- WebDataCommons: Large-Scale Extraction of RDFa, MicroFormats, and Microdata from the Web
RDF Tools: Storage

• RDF is often stored in relational databases

• Different storage strategies
  – single triple table
  – one table per class
  – one table per property

• Strategies differ
  – by requirements of (disk) space
  – by query response time for different query types

• Examples: Virtuoso, Sesame, ...
RDF Tools: Visualization

• Mostly graph-based visualization tools
RDF Tools: Validation

- W3C RDF Validator:
  - W3C RDF Validator: http://www.w3.org/RDF/Validator/
  - Output of RDF/XML and graphs

- EasyRDF Validator:
  - http://www.easyrdf.org/ converter
  - Understands and converts a variety of notations
RDF Tools: Programming, Reasoning

• Programming Frameworks
  – for developing RDF-based applications
  – e.g., JENA, RDFReactor, ...

• Reasoners
  – can draw logical conclusions from RDF graphs
  – can answer queries on RDF graphs

• Both will be covered in separate lectures
Wrap Up

• RDF is a language for describing arbitrary things
  – interpretation: set of statements or directed graph
  – Notations: RDF/XML, Turtle

• Special language constructs
  – Blank nodes
  – Reification

• Semantics
  – Non-unique name assumption
  – Open world assumption

• Embedding in HTML is possible
• Large set of tools is available
A Critical Look in the Rear View Mirror

- Is RDF more powerful than XML?
- XML is a markup language for information
- In XML, arbitrary elements and attributes can be defined
- XML tag names are meaningless for a computer
- RDF is a markup language for information
- In RDF, arbitrary classes and predicates can be defined
- RDF class and predicate names are meaningless for a computer
So, why did we spend the last 90 minutes on RDF?
A Critical Look in the Rear View Mirror

How Standards Proliferate:

(See: A/C chargers, character encodings, instant messaging, etc)

SITUATION:
There are 14 competing standards.

14?! Ridiculous!
We need to develop one universal standard that covers everyone's use cases.

Yeah!

SITUATION:
There are 15 competing standards.

http://xkcd.com/927/
Semantic Web – Architecture

here be dragons...

Semantic Web Technologies
(This lecture)

Technical Foundations

Berners-Lee (2009): Semantic Web and Linked Data
Questions?