An Introduction to Linked Open Data

Felix.Ostrowski@googlemail.com (@literarymachine)
Adrian Pohl@hbz-nrw.de (@acka47)

SWIB 2013 Pre-Conference Workshop
Monday, November 25th 2013
Hamburg
Schedule

- Organize in teams
- Introduction: Data – Graphs – Triples
- Groupwork
- URIs and Namespaces
- Groupwork
- Open Data Principles
- Groupwork
- Identification vs. Description
- Groupwork
- Triple Stores & SPARQL
- Groupwork
- RDF Schema
- Groupwork
- Summary, Questions & Discussion
Linked Open Data

➡️ It's about data …
➡️ … more precisely: about open data …
➡️ … even more precisely: about linked open data!
Data, how we know it

(To be honest, we might actually be the only ones knowing such data. And there aren't too many things that one can describe in this way.)
Along came the Internet

http://www.w3.org/DesignIssues/Abstractions.html
Data, how others know it

(Of course, "others" does not mean "everybody". But at least you can describe many things this way. Maybe even everything.)
The World Wide Web

http://www.w3.org/DesignIssues/Abstractions.html
Data, how the web likes it

(No wonder, it actually looks like a web. Or, if you will, a directed labelled graph.)
The Giant Global Graph

http://www.w3.org/DesignIssues/Abstractions.html
Your turn!
Draw a graph of your social network.

(For now, stick with the people on your table)
A simple social graph

Adrian knows Felix

"Adrian" knows "Felix"

"Pohl" knows "Ostrowski"
Obviously a computer will have trouble interpreting such a diagram. The **graph data model** is an **abstract** one, but we can concrete it for the computer.
Graphs, (almost) how computers like them

(This notation is called **Turtle** and it is one of several writing styles for a data model called **RDF**. RDF stands for "**Resource Description Framework**"; this is the de-facto standard for publishing Linked Data. A big advantage of the Turtle notation: humans can actually read it!)
Basic element: the triple

Weaving the Web is written by Tim Berners-Lee.

(Weaving the Web) <is written by> <Tim Berners-Lee>.

(A triple is the smallest possible graph. It's components are called subject, predicate and object.)
Your turn!
Open the etherpad for your group. In this etherpad, express the graph you have drawn in RDF.
Simple social graph in RDF

```rdf
<Adrian> <first name> "Adrian" .
<Adrian> <last name> "Pohl" .
<Adrian> <knows> <Felix> .
<Felix> <first name> "Felix" .
<Felix> <last name> "Ostrowski" .
<Felix> <knows> <Adrian> .
```
What does ... 

... <Tim Berners-Lee>,
... <London> and
... <England> stand for, and what does 

<has first name>,
<is located in> and
<has population> mean?
We need **unambiguous reference!**

Authority files are a good start, but again we'll be the only ones understanding those. On the web, people use **URIs!**

(URI stands for **Uniform Resource Identifier**)
http://de.wikipedia.org/wiki/Uniform_Resource_Identifier
ftp://ftp.is.co.za/rfc/rfc3986.txt
file:///home/fo/doc/swib13/slides.odp
urn:isbn:978-1608454303
Graphs, how computers really like them

(A pleasant side-effect when using HTTP-URIs – which is what Linked Data is based upon, is that they can be dereferenced. When following such a link, one should get a description of the resource. More on that later.)
Graphs, (sort of) readable for humans and machines

@prefix dc: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix gnd: <http://d-nb.info/gnd/> .

gnd:121649091 foaf:givenName "Tim" .
gnd:121649091 foaf:familyName "Berners-Lee" .
gnd:121649091 foaf:birthday "06/08/1955" .

(You can abbreviate URIs using prefixes. This also makes it easier to identify the vocabularies you use.)
But isn't some data we had missing!? 

(There may not be a URI for everything you want to refer to, neither for entities nor for vocabularies.)
Don't repeat others, link!

- Reuse properties from existing vocabularies
- Link to things by simple URI reference
- Think Data-Library (as in Software-Library)
(When something you want to describe does not have a URI yet, you can use IDs that are relative to the describing document. Since two documents can't be at the same place at the same time, these IDs only have to be unique within that document. "<>" stands for the document itself. You can check here if you are creating valid turtle.)
Your turn!
Reformulate your RDF using the FOAF vocabulary. Also, use DC Terms to assert that you are the authors of the describing document. You can also add further metadata about the document if you want.
Simple social graph using FOAF

@prefix :     <#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc:   <http://purl.org/dc/terms/> .

:adrian foaf:givenName "Adrian" .
:adrian foaf:familyName "Pohl" .
:felix foaf:givenName "Felix" .
:felix foaf:familyName "Ostrowski" .

<> dc:creator <Felix> .
<> dc:creator <Adrian> .
<> dc:created "25.11.2013" .
Break
Open Data
Your turn!
Agree on a Creative Commons License within your group and link your document to that license.

(The predicate `<http://creativecommons.org/ns#license>` is well suited for this link, but searching the Web will reveal alternatives.)
Open licencing
Linked Data in Action
Ceci n'est pas la Tour Eiffel.
Identification and description of a resource ought to be distinguished! But in the Linked-Data-Paradigm, both are linked.
The description of a resource can be made available in various formats. Which format will be delivered can be decided by Content-Negotiation.
Your turn!
In your description, link yourself to people from other groups that you know. This doesn't have to be reciprocal.

Also, link (approximately) to the place you live or work. Use DBpedia for this.
Break
Scattered machine-readable descriptions are useful, but we can do better than that! RDF is a distributed data model that makes it easy to combine several descriptions. Furthermore, special databases exist that allow to query RDF data.
@prefix foaf: <xmlns.com/foaf/0.1/> .
@prefix ex1: <http://ex1.org/> .
@prefix ex2: <http://ex2.org/> .

ex1:adrian foaf:givenName "Adrian" .
ex1:adrian foaf:knows ex2:felix .

@prefix foaf: <xmlns.com/foaf/0.1/> .
@prefix there: <http://ex1.org/> .
@prefix here: <http://ex2.org/> .

here:felix foaf:givenName "Felix" .
here:felix foaf:knows there:adrian .

<http://ex1.org/adrian> <xmlns.com/foaf/0.1/givenName> "Adrian" .
<http://ex2.org/felix> <xmlns.com/foaf/0.1/givenName> "Felix" .
Felix Ostrowski

Graph thinker

Profiles

Facebook  Twitter  Slideshare  Github  LinkedIn  Delicious

Activity

- Is there something missing in the link? (30. November 2011)
- Dead Simple: RDF and SPARQL using PHP (17. Januar 2011)

About me

INTERESTS

- Semantisches Web
- Informationsarchitektur
- Virtuelle Bibliothek
- Linked Open Data

Contacts

WORK

Address

- Reichenberger Straße 1, Berlin, 10999, Deutschland
- Phone: +49 30 12092041

Network

- Knud Möller
- Adrian Pohl
- Pascal Christoph

Interests

SEMANTISCHES WEB

-

-
Triple Stores

http://www.example.org/data/alice

http://de.dbpedia.org/page/Berlin

http://www.example.org/data/carol

http://de.dbpedia.org/page/Kölن
**SPARQL** facilitates queries on the data in a triple store. The foundations for this are simply graph **patterns**. These look almost like triples, the difference being that the contain **variables**.
@prefix ex: <http://example.org/people#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

ex:alice foaf:name "Alice" .

PREFIX ex: <http://example.org/people#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT * WHERE {
    ex:alice foaf:name ?name .
}

name
"Alice"
@prefix ex: <http://example.org/people#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

ex:alice foaf:name "Alice" ;
    foaf:knows ex:bob .

ex:bob foaf:name "Bob" ;
    foaf:knows ex:carol .

ex:carol foaf:name "Carol" ;
    foaf:knows ex:alice .

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name1 ?name2 WHERE {
    ?person1 foaf:knows ?person2 .
    ?person1 foaf:name ?name1 .
}
@prefix ex: <http://example.org/people#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dbpedia: <http://de.dbpedia.org/resource/> .

ex:alice foaf:name "Alice" ;
    foaf:knows ex:bob ;
    foaf:based_near dbpedia:Berlin .

ex:bob foaf:name "Bob" ;
    foaf:knows ex:carol ;
    foaf:based_near dbpedia:Dresden .

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

SELECT ?name ?ortname WHERE {
    ?person1 foaf:knows ?person2 .
    ?person2 foaf:name ?name .
}

<table>
<thead>
<tr>
<th>name</th>
<th>ortname</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Bob&quot;</td>
<td>&quot;Dresden&quot;@de</td>
</tr>
</tbody>
</table>
Your turn!
Use SPARQL to analyse your connections. For example you might want to determine who you know directly or indirectly or who comes from the same city as you.
Let's put some **Semantic** in the **Web**

The **classes** and **properties** being used can be using **description languages for vocabularies**. The relatively simple RDF Schema (**RDFS**) is wide spread, but more complex issues can be expressed in the Web Ontology Language (**OWL**).
# RDF Schema
foaf:knows rdfs:type rdfs:Property ;
    rdfs:range foaf:Person ;
    rdfs:domain foaf:Person .
foaf:Person rdfs:type rdfs:Class .

# Explicit triples
ex:bob foaf:knows ex:alice .

# Implicit triple, that follow from the schema
ex:bob rdf:type foaf:Person .
ex:alice rdf:type foaf:Person .
# RDF Schema as a "bridge" across vocabularies

```
ex:colleague rdfs:subPropertyOf foaf:knows ;
rdfs:domain ex:Employee ;
rdfs:range ex:Employee .
ex:Employee rdf:type rdfs:Class ;
rdfs:subClassOf foaf:Person .
```

# Explicit triples
```
ex:bob ex:colleague ex:alice .
```

# Implicit triple, that follow from the schema
```
ex:bob foaf:knows ex:alice .
ex:bob rdf:type foaf:Person .
ex:alice rdf:type foaf:Person .
ex:bob rdf:type foaf:Employee .
ex:alice rdf:type foaf:Employee .
```
Your turn!
Create an RDF Schema so that from these assertions

```ttl
@prefix team: <http://example.org/soccer/vocab#> .
@prefix ex: <http://example.org/soccer/resource#> .

ex:team1 team:player ex:bob .
ex:team2 team:player ex:alice .
ex:game1 team:home ex:team1 .
ex:game1 team:away ex:team2 .
```

the following triples can be inferred.

```ttl
@prefix team: <http://example.org/soccer/vocab#> .
@prefix ex: <http://example.org/soccer/resource#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

ex:team1 rdf:type foaf:Group .
ex:team2 rdf:type foaf:Group .
ex:team1 foaf:member ex:bob .
ex:team2 foaf:member ex:alice .
ex:bob rdf:type foaf:Person .
ex:alice rdf:type foaf:Person .
ex:game1 rdf:type team:Game .
ex:game2 rdf:type team:Game .
```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix team: <http://example.org/soccer/vocab#> .

team:player  rdf:type rdfs:Property ;
             rdfs:subPropertyOf foaf:member ;
             rdfs:domain foaf:Person ;
             rdfs:range foaf:Group .

team:home    rdf:type rdfs:Property ;
             rdfs:domain team:Game .

team:away    rdf:type rdfs:Property ;
             rdfs:domain team:Game .

team:Game    rdf:type rdfs:Class .
The expressiveness and the possibilities of inference of RDFS and OWL are not always needed.

For controlled vocabularies, the Simple Knowledge Organization System (SKOS) is a simpler alternative that is also based on RDF.

The Dewey Decimal Classification and the Library of Congress Subject Headings have already found their way into the Linked-Data-world.
Elements of Linked (Open) Data

Using: Mashups
Mashups combine multiple datasets to create a new service, visualisation or information.

Using: Search
Linked data search engines allow search across the web of data. Conventional search may present information derived from linked data.

Using: Productivity
Linked data facilitates data integration for business intelligence or research.

Storing and publishing
Linked data can be published in simple flat files on a web server, in databases with a translation layer, or in specialised ‘triple stores’ built to store and share linked data. Publishing platforms understand requests for linked data & return it formatted as RDF.

Querying: SPARQL
SPARQL Protocol and RDF Query Language provides a way to run structured queries over linked data datasets. SPARQL servers expose linked open data to be queried.

Representing: Vocabularies
Vocabularies provide lists (and definitions) of common terms that can be used to describe the things and relationships in a dataset.

Representing: Ontologies
Ontologies are vocabularies that record the logical relationships between their terms and support reasoning.

Interchanging: RDF
Resource Descriptor Framework (RDF) is a model for representing data as ‘triples’. RDF can be serialised into a range of different file formats, including RDF-XML and text-based Turtle or N3 syntax.

Identifying: URLs
Using HTTP Uniform Resource Locators (URLs) means that (a) data can be looked up across the internet; (b) decisions about ‘namespaces’ for data are managed through the Domain Name System (DNS).

Transporting: HTTP (The World Wide Web)
Data is hosted on servers that can talk Hypertext Transfer Protocol (HTTP) to each other and to browsers in order to exchange data across the Internet.


Elements of the Linked Open Data Stack (revision 3) - 5th May 2011. CC BY-SA-NC
Thank you!
License

These slides are published under a Creative Commons license:
http://creativecommons.org/licenses/by/3.0/de/