RDF and SPARQL

… with acknowledgements to the W3C’s RDF primer, to SPARQL By Example, and to Wolfgang Kienreich for slides from year 2012
Semantic Web Stack revisited
RDF – Resource Description Framework

RDF is a language for making statements about resources

What resources?
RDF was created to make statements about resources in the WWW. Resources then are web pages or web documents.

By generalising what is a “web resource”, RDF can be used to make statements about things that are identified by a “web resource”, such as real persons, real enterprises, etc.

Resources are identified by URIs
Example from http://www.w3.org/TR/rdf-primer/#intro
Kinds of resources in RDF

Individuals (entities)
Kinds of things (classes, entity types)
Properties (relationships)
Data Values

Example from http://www.w3.org/TR/rdf-primer/#intro
RDF as a labelled directed graph

Nodes and edges are labelled by URIs

Statements are triples
• start node – edge – end node
• In RDF-speak: subject – predicate – object

Subject:  http://www.example.org/index.html
Predicate: http://purl.org/dc/elements/1.1/creator
Object:  http://www.example.org/staffid/85740

Picture taken from:  http://www.w3.org/TR/rdf-primer/#rdfmodel
QName shorthand for RDF

RDF statement:

QName for URI: Prefix:localName

Often-used prefixes are:
prefix rdf:, namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#
prefix rdfs:, namespace URI: http://www.w3.org/2000/01/rdf-schema#
prefix dc:, namespace URI: http://purl.org/dc/elements/1.1/
prefix owl:, namespace URI: http://www.w3.org/2002/07/owl#
prefix ex:, namespace URI: http://www.example.org/ (or http://www.example.com/
prefix xsd:, namespace URI: http://www.w3.org/2001/XMLSchema#

QName statement:
Example: Transform ER model into RDF

Franz works at the Institute of Software Technology, which belongs to the Dptmt of Computer Science, and Franz is also dean of the Dptmt of Computer Science.
Step 1

Make resources out of entities and entity types

Franz: [http://kti.tugraz.at/examples/Franz](http://kti.tugraz.at/examples/Franz)
IST: [http://kti.tugraz.at/examples/IST](http://kti.tugraz.at/examples/IST)
Dptmt. of CS: [http://kti.tugraz.at/examples/CS](http://kti.tugraz.at/examples/CS)
Works-At: [http://kti.tugraz.at/examples/works-at](http://kti.tugraz.at/examples/works-at)
Belongs-to: [http://kti.tugraz.at/examples/belongs-to](http://kti.tugraz.at/examples/belongs-to)
Is-dean-of: [http://kti.tugraz.at/examples/is-dean-of](http://kti.tugraz.at/examples/is-dean-of)

Could I have used the URLs of the websites of Franz, IST, Dtpmt of CS as resource identifiers?

YES
Step 2

Translate binary relationships between entities into RDF triples

Prefix: ex:, namespace URI http://kti.tugraz.at/examples/

ex:Franz ex:works-at ex:IST .
ex:IST ex:belongs-to ex:CS .
Beyond triples…

Triples represent binary relationships.
For n-ary relationships, nodes must be created that „collect“ the n relationships:

Picture taken from: http://www.w3.org/TR/rdf-primer/#structuredproperties
Triples for example from previous slide

exaddressid:85740 exterms:street "1501 Grant Avenue" .
exaddressid:85740 exterms:city "Bedford" .
exaddressid:85740 exterms:state "Massachusetts" .
... lie blank nodes

Picture taken from: http://www.w3.org/TR/rdf-primer/#structuredproperties
Triples for example from previous slide

exstaff:85740 exterms:address _:johnaddress .
_:johnaddress exterms:street "1501 Grant Avenue" .
_:johnaddress exterms:city "Bedford" .
_:johnaddress exterms:state "Massachusetts" .
_:johnaddress exterms:postalCode "01730" .
RDF/XML Syntax

RDF‘s conceptual model is a graph. The language itself does not depend on a particular syntax. BUT: RDF/XML is the normative syntax for writing RDF.

```xml
<?xml version="1.0"?>
<rdf:RDF
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:ex="http://www.example.org/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="http://www.example.org/index.html">
    <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
  </rdf:Description>
</rdf:RDF>
```
RDF in relation to ER

- RDF can express statements about entities
  - Provides only „rdf:type“ to state something like entity types

- RDF can express relationships and attributes
Exercise 1

<table>
<thead>
<tr>
<th>Jahr</th>
<th>Monat</th>
<th>Herkunftsland</th>
<th>Naechtigung</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>1</td>
<td>Arabische Länder in Asien</td>
<td>74</td>
</tr>
<tr>
<td>2012</td>
<td>1</td>
<td>Australien</td>
<td>188</td>
</tr>
<tr>
<td>2012</td>
<td>1</td>
<td>Baden-Württemberg</td>
<td>1158</td>
</tr>
<tr>
<td>2012</td>
<td>1</td>
<td>Bayern</td>
<td>2699</td>
</tr>
<tr>
<td>2012</td>
<td>1</td>
<td>Belgien</td>
<td>180</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

„Triplify“ the highlighted row using the ER model below.
Exercise 2

1. Express in valid RDF/XML (use the [W3C RDF Validator](http://validator.w3.org) to check your RDF/XML):

„Viktoria works at the Knowledge Technologies Institute of Graz Univ. of Technology, and her full name is Viktoria Pammer-Schindler.“

2. Which triples are expressed by the following RDF/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:ex="http://example.org/stuff/1.0/">
  <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
                  dc:title="RDF/XML Syntax Specification (Revised)">
    <ex:editor>
      <rdf:Description ex:fullName="Dave Beckett"/>
    </ex:editor>
  </rdf:Description>
</rdf:RDF>
```
SPARQL – Query Language for RDF

SPARQL Endpoints accept queries and return results via HTTP

Result formats

- XML
- JSON
- CSV
- RDF
SPARQL 1.0 and 1.1

**SPARQL 1.0**
- SPARQL 1.0 Query Language
- SPARQL 1.0 Protocol (for sending queries over HTTP)

**SPARQL 1.1**
- 1.1 Versions of SPARQL Query ([W3C Document](https://www.w3.org/TR/rdf-sparql-11/)) and Protocol
- SPARQL 1.1 Update (for inserting, deleting, modifying RDF data)
- SPARQL 1.1 Service Descriptions (how to describe capabilities of SPARQL Endpoints)
- SPARQL 1.1 Entailments (how to combine reasoning with SPARQL)
SPARQL Query Structure

A SPARQL query comprises, in order:

- **Prefix declarations** for abbreviating URIs (PREFIX)
- **Dataset definition**, stating what RDF graph(s) are being queried (FROM)
- A **result clause**, identifying what information to return from the query (SELECT)
- The **query pattern**, specifying what to query for in the underlying dataset (WHERE)
- **Query modifiers**, slicing, ordering, and otherwise rearranging query results (ORDER BY, LIMIT, …)
SPARQL Query Structure

# prefixes
PREFIX kti: <http://kti.com/res/>
...

# results clause
SELECT ...

# datasets
FROM ...

# query pattern
WHERE

# query modifiers
ORDER BY ...
Example: Exploring DBPedia

http://dbpedia-live.openlinksw.com/sparql

# prefixes already defined
# see "namespace prefixes" at endpoint
# website

# datasets already defined
# implicit from dbpedia.org

# get me some triples
SELECT * WHERE { ?s ?p ?o } LIMIT 20

SELECT * is an abbreviation that selects all of the variables in a query. “SELECT ?s ?p ?o” is equivalent to “SELECT *” in this example query.

Variables are prefixed by “?” or “$“.

The result set is limited to 20 results.
Pattern Matching

# List rdfs:classes
SELECT DISTINCT ?o WHERE { ?s a ?o } LIMIT 20

# What about the unbound ?s
SELECT DISTINCT ?o WHERE { [] a ?o } LIMIT 20

# Select along multiple triples
SELECT DISTINCT * WHERE { [] a ?o .
                             ?o rdfs:label ?label }
                        LIMIT 20

SELECT returns all triples in the chosen dataset that match the triple pattern.
Triple patterns are like triples, except they contain variables.

„a“ is shorthand for „rdf:type“ – try this in dbpedia SPARQL Endpoint!
Dataset Selection

In generic SPARQL Endpoints the dataset is not predefined.

```
PREFIX ... 
SELECT * 
FROM <http://www.w3.org/2002/07/owl.rdf> 
WHERE { 
    ... 
}
```
Modify Solution Sequence

ORDER BY put the solutions in order
DISTINCT: ensure solutions in the sequence are unique
LIMIT: restrict the number of solutions

```
select distinct * 
where {
  [] a ?Concept .
  ?Concept rdfs:label ?label
}
ORDER BY ASC(?label)
LIMIT 20
```
Other queries

**ASK** returns true if the query has a solution; else false

```
ASK FROM <http://dig.csail.mit.edu/2008/webdav/timbl/foaf.rdf>
{?x a <http://xmlns.com/foaf/0.1/Person>}
```

**DESCRIBE** returns an RDF graph with data about all resources that match

```
DESCRIBE <http://dbpedia.org/resource/Leonardo_da_Vinci>
```

```
DESCRIBE ?x WHERE {?x rdf:type <http://dbpedia.org/ontology/Artist> } LIMIT 20
```

**CONSTRUCT** returns an RDF graph (triples)

Try at [SPARQLer](#)

Try at [DBPEDIA Virtuoso Endpoint](#)
Exercise 3

1. Find me all the people in Tim Berners-Lee's FOAF file (http://dig.csail.mit.edu/2008/webdav/timbl/foaf.rdf) that have names and email addresses. Return each person's URI, name, and email address. Hint: Use Redland Rasqal Demo to test your query.

2. Find out everything about http://dbpedia.org/resource/Marie_Curie in DBPEDIA. Hint: Use the DBPEDIA Snorql Explorer to test your queries.

3. Find out all statements that describe http://dbpedia.org/resource/Marie_Curie (as subject).
RDF and SPARQL Tools

**RDF Gravity** to visualise RDF graphs

**W3C RDF Validator**

**RDF Validator and Converter**

Generic SPARQL Endpoint:
- [Redland Rasqal Demo](#)
- [SPARQLer](#)
- [Yasgui](#)

DBPEDIA SPARQL Endpoint:
- [SNORQL Explorer](#)
- [Virtuoso Endpoint](#)