Multi-agent and Semantic Web Systems:
Linked Open Data

Fiona McNeill
School of Informatics
14th February 2013
N3 Triples

@prefix vCard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix info: <http://somewhere/peopleInfo#> .
@prefix s: <http://somewhere/> .

s:RebeccaSmith
             vCard:Given   "Rebecca"
       ] .

s:SarahJones
  vCard:N [ vCard:FN  "Sarah Jones" ;

s:JohnBurns
  info:age 25 .
Jena VCard 2: Triples

N3 Triples

@prefix vCard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix info: <http://somewhere/peopleInfo#> .
@prefix s: <http://somewhere/> .

s:RebeccaSmith
   info:age 23 ;
   vCard:FN "Becky Smith" .

s:MattGreen
   vCard:FN "Matt Green" ;
   vCard:N [ vCard:Family "Green" ;
             vCard:Given "Matthew"] .

s:SarahJones

s:JohnBurns
   vCard:FN "John Burns" ;
   vCard:N [ vCard:Family "Burns" ;
             vCard:Given "John" ] .
Jena VCard 2: Graph

s:RebeccaSmith

info:age

<23>

vCard:FN

<Becky Smith>

s:JohnBurns

vCard:FN

_<bn007>

vCard:N

<John Burns>

vCard:Family

<Burns>

vCard:Given

<John>

s:SarahJones

vCard:N

_<bn004>

vCard:Family

<Jones>

s:MattGreen

vCard:N

_<bn001>

vCard:Family

<Green>

vCard:Given

<Matthew>
Jena VCard Merged: Graph

Graph showing relationships between different individuals and their details.
• Note problem with trying to merge blank nodes.

• rdfcat is one way of merging:
  rdfcat file1 file2 > mergedfile

• Visualization:
  • IsaViz (www.w3.org/2001/11/IsaViz/) — also does merging
  • Protegé (uses Graphviz)
  • RDFS Explorer http://xml.mfd-consult.dk/ws/2003/01/rdfs/
**N3 Triples**

```
PREFIX info: <http://somewhere/peoplenfo#>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>

SELECT ?name ?age
WHERE
{
}
```
Query

This query only returns people for whom we have age information.

What if we want to return people and also ages just when it is available?
### Query

<table>
<thead>
<tr>
<th>name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Burns</td>
<td>25</td>
</tr>
<tr>
<td>Becky Smith</td>
<td>23</td>
</tr>
</tbody>
</table>

- This query only returns people for whom we have age information.
- What if we want to return people and also ages just when it is available?
- Use the **OPTIONAL** keyword.
Query

N3 Triples

PREFIX info: <http://somewhere/peopleInfo#>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>

SELECT ?name ?age
WHERE
{
    OPTIONAL { ?person info:age ?age }
}

Results

<table>
<thead>
<tr>
<th>name</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;John Burns&quot;</td>
<td>25</td>
</tr>
<tr>
<td>&quot;Matt Green&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;Becky Smith&quot;</td>
<td>23</td>
</tr>
<tr>
<td>&quot;Sarah Jones&quot;</td>
<td></td>
</tr>
</tbody>
</table>

- RDF is semi-structured data
- [OPTIONAL](http://www.w3.org/TR/2009/REC-sparql11-query-20090520/grammar/dml-query-hdl.html#optional) gives SPARQL the ability not to fail a query when specific data does not exist.
Linked Data Principles

• Use URIs as names for things.

• Use HTTP URIs, so that people can look up those names.

• When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).

• Include links to other URIs, so that they can discover more things.
Why HTTP URIs?

• Globally unique names can be created in a decentralised fashion by domain name owners; no central naming authority is required.

• Not just a name, but a means of accessing information describing the identified entity.
URIs

- These URIs point to web documents - or in the terminology of WebArch, information resources.
  - by definition, all its essential characteristics can be conveyed in a message

- Web clients request a representation of a resource

- One and the same resource might have different representations; e.g., text in English, Greek, Chinese, etc.
Content Negotiation

• HTTP clients send HTTP headers with each request to indicate what kinds of documents they prefer
• Servers inspect headers and select an appropriate response.
• Client can say prefers language X over Y.
• Or prefers RDF over HTML

Header of GET request
GET /people/staff/Ewan_Klein.html HTTP/1.1
Host: www.inf.ed.ac.uk
Accept: text/html, application/xhtml+xml
Accept-Language: en, gr, cn

Server Response
HTTP/1.1 200 OK
Content-Type: text/html
Content-Language: gr
URIs for Things

• We need mechanisms to ensure that when URIs are dereferenced,
  • real-world objects are not confused with documents that describe them, and
  • humans as well as machines can retrieve appropriate representations.

• Two strategies for deferencing URIs for real world objects:
  • 303 URIs
  • hash URIs
Solution 1: 303 (See other) URIs

- Server should not return a 200 OK for a real-world object URI — it doesn’t have a representation of the resource.

- Instead (cf. httpRange-14 resolution), server should send 303 See Other plus the URI of a web document that describes the object; this is also called a 303 redirect.

- Client then dereferences this new URI and gets a description of the resource.

**DBPedia URIs for Real-world Objects**

<table>
<thead>
<tr>
<th>URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://dbpedia.org/resource/Bo_Diddley">http://dbpedia.org/resource/Bo_Diddley</a></td>
<td>[resource]</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/data/Bo_Diddley.rdf">http://dbpedia.org/data/Bo_Diddley.rdf</a></td>
<td>[RDF description]</td>
</tr>
<tr>
<td><a href="http://dbpedia.org/page/Bo_Diddley">http://dbpedia.org/page/Bo_Diddley</a></td>
<td>[HTML description]</td>
</tr>
</tbody>
</table>
Solution 1: 303 (See other) URIs

```
http://dbpedia.org/resource/Bo_Diddley
```

303 redirect with content negotiation

```
http://dbpedia.org/data/Bo_Diddley.rdf
http://dbpedia.org/page/Bo_Diddley
```

- **thing**
  - **RDF**
    - `application/rdf+xml`
      - `http://dbpedia.org/data/Bo_Diddley.rdf`
  - **HTML**
    - `text/html`
      - `http://dbpedia.org/page/Bo_Diddley`
Solution 1: Hash URIs

- Use ‘hash URIs’ for non-document resources, i.e., add a fragment, indicated by #

- Following HTTP protocol, clients must strip off the fragments before sending request to server.

- So the URI with the fragment cannot be retrieved directly and cannot therefore identify a Web document.

- So hash URI can identify real-world objects without creating ambiguity.
Hash v 303

- Hash URIs reduce number of HTTP requests; cf. http://www.w3.org/TR/cooluris/#choosing for arguments in favour.

- But all resources that share same hash URI dereference to same description document; can mean lots of redundant data is transmitted.

- 303 redirects can be configured separately for each resource.

- Some data publishers (e.g., http://kasabi.com/doc/api/linked-data don’t support hash URIs).
5- ★ Data

LINKED DATA
On the web, open license
Machine-readable data
Non-proprietary format
RDF standards
Linked RDF

Is your data 5 ★?
Is Your Data 5-★ ?

★ Data available on the web (in whatever format), but with an open licence

★★ Available as machine-readable structured data (e.g. Excel instead of image scan of a table)

★★★ as ★★ plus: Use non-proprietary data format (e.g. CSV instead of Excel)

★★★★ All the above plus: Use open standards from W3C (e.g. HTTP URIs) to identify things, so that people can point at your stuff

★★★★★ All the above, plus: Link your data to other people’s data to provide context
RDF for Linked Data

RDF is standardly used for Linked Data. Advantages include:

- Easy to insert RDF links between data from different sources.
- Information from different sources can be combined by graph merging.
- Information using different schemas can be expressed in a single graph, i.e., by mixing different vocabularies.
- Data can be tightly or loosely structured.

Features of RDF that are avoided:

- Reification (hard to query with SPARQL)
- Collections and containers (ditto). Use multiple triples with same predicate instead.
- Blank nodes: makes merging less effective.
Kinds of Links

**Relationship Links** point at related things in other data sources. LD counterpart to outgoing hyperlinks in a web document. E.g., foaf:based_near dbpedia:Edinburgh

**Identity Links** point at URI aliases used by other data sources to identify the same real-world object or abstract concept.

**Vocabulary Links** point from data to the definitions of the vocabulary terms that are used to represent the data.
Identity Links

- Many different URIs used to refer to same real-world object.
- Standard mechanism for saying that two URI aliases refer to same object: http://www.w3.org/2002/07/owl#sameAs.
- Motivations for this approach:
  - Different aliases can be dereferenced to different description of same resource (AAA principle).
  - Can support provenance for LD consumers: trace back to who published the URI.
  - Having only one, canonical, URI for each object would require centralised naming authority, and act as barrier to spread of web of data.

Potential problems:
- Identity may be context dependent
- Facts vs. opinions
Summary

Stuff we looked at today:

- Adding RDFS schema information
- Graph merging
- Graph visualisation
- SPARQL OPTIONAL
- URIs for informational vs. real-world objects
- LOD principles for publishing data
• Structured data is made available on web (i.e., open) in many formats: CSV, Excel, HTML Microdata (e.g., http://schema.org/), web APIs, PDF tables (shudder), ...

• Advantages of Linked Data:
  • A unifying data model (RDF)
  • A standardised data access mechanism (HTTP).
  • Hyperlink-based data discovery: links connect all Linked Data into a single global data space and enable Linked Data applications to discover new data sources at run-time.
  • Self-descriptive data: vocabulary definitions are recoverable like other data, and vocabulary terms can be linked to one another.
Linked Data Summary

• Linked data adopts perspective of data integration.
• Not interested in reasoning aspect of Semantic Web.

http://blog.paulwalk.net/2009/11/11/linked-open-semantic/:
• Data can be open, while not being linked.
• Data can be linked, while not being open.
• Data which is both open and linked is increasingly viable.
• The Semantic Web can only function with data which is both open and linked.