THE UNIVERSITY of EDINBURGH
informatics

Semantic Web Systems

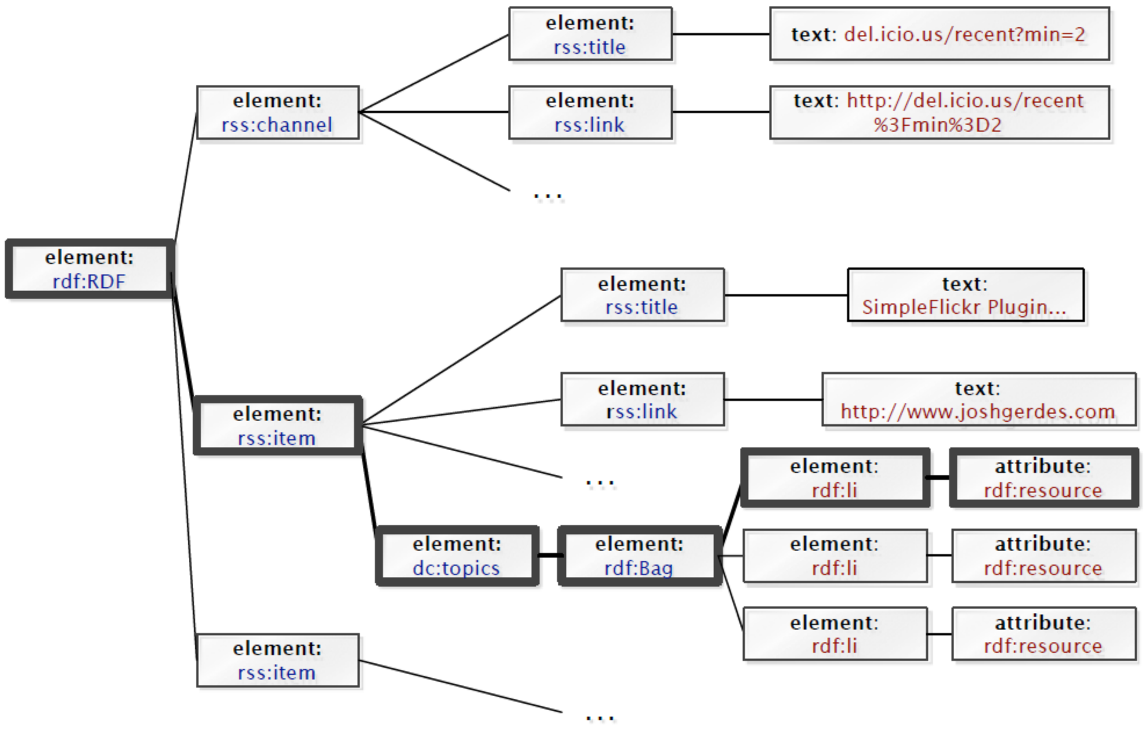
Linked Open Data

Jacques Fleuriot

School of Informatics

In the previous lecture

- Querying with XML
  - Basic idea: search along paths in an XML tree
  - e.g. path expression: `/rdf:RDF/rss:item/dc:topics/rdf:Bag/rdf:li`



In the previous lecture

- Querying with SPARQL
  - matching triple patterns in the RDF graph

Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name1  ?name2
WHERE {
   ?person1   foaf:knows   ?person2 .
   ?person1   foaf:name   ?name1 .
   ?person2   foaf:name   ?name2 .
  }
```

In this lecture

- Merging graphs that contain blank nodes
- OPTIONAL in SPARQL querying
- Linked Data principles

Merging graphs that contain blank nodes

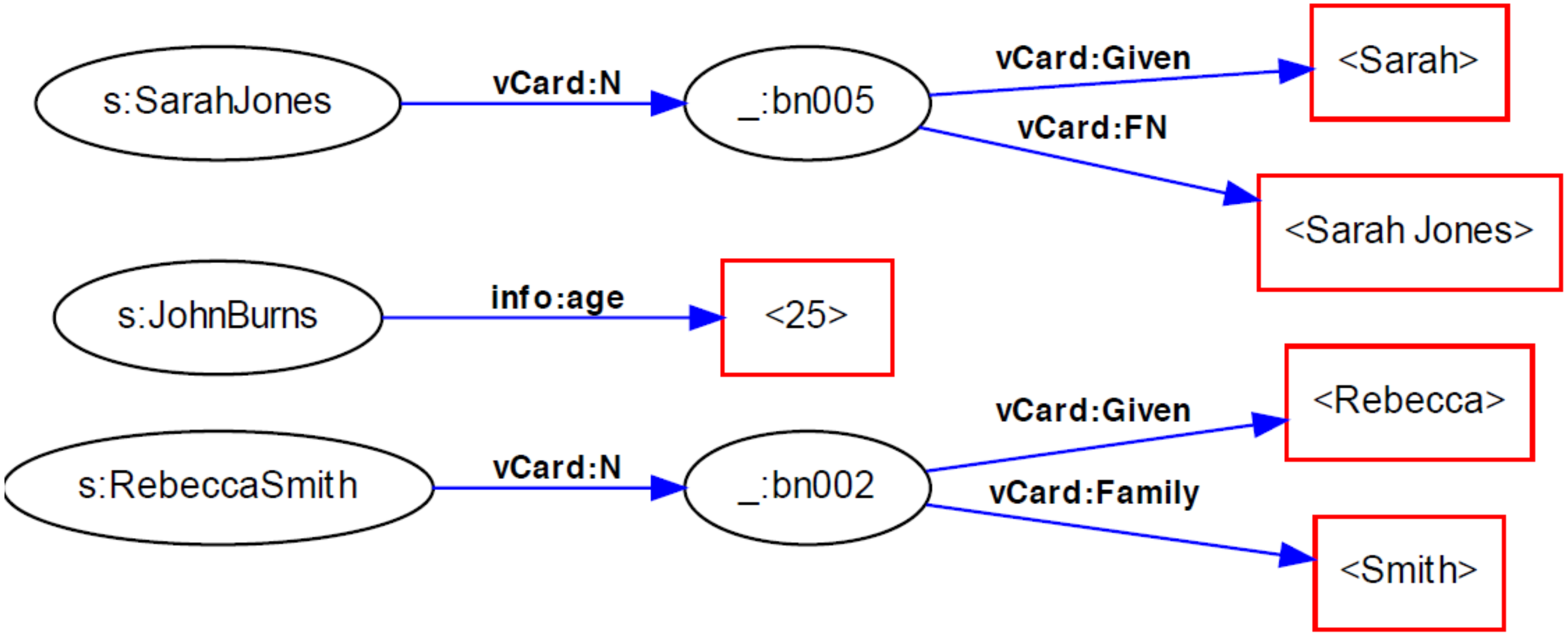
Jena vCard 1: Triples

N3 Triples

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

s:RebeccaSmith
      vCard:N [ vCard:Family "Smith" ;
                vCard:Given "Rebecca" ] .
s:SarahJones
      vCard:N [ vCard:FN "Sarah Jones" ;
                vCard:Given "Sarah" ] .
s:JohnBurns
      info:age 25 .
```

Jena vCard 1: Graph



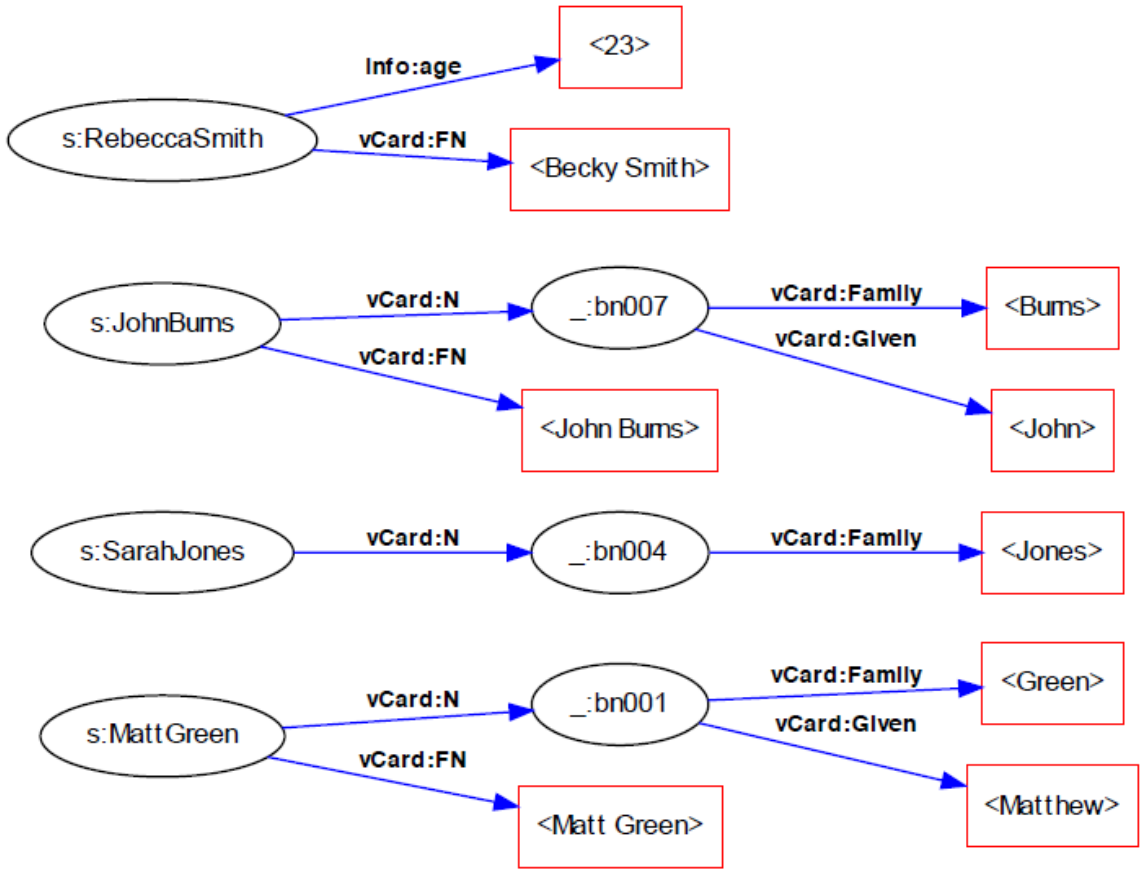
Jena vCard 2: Triples

N3 Triples

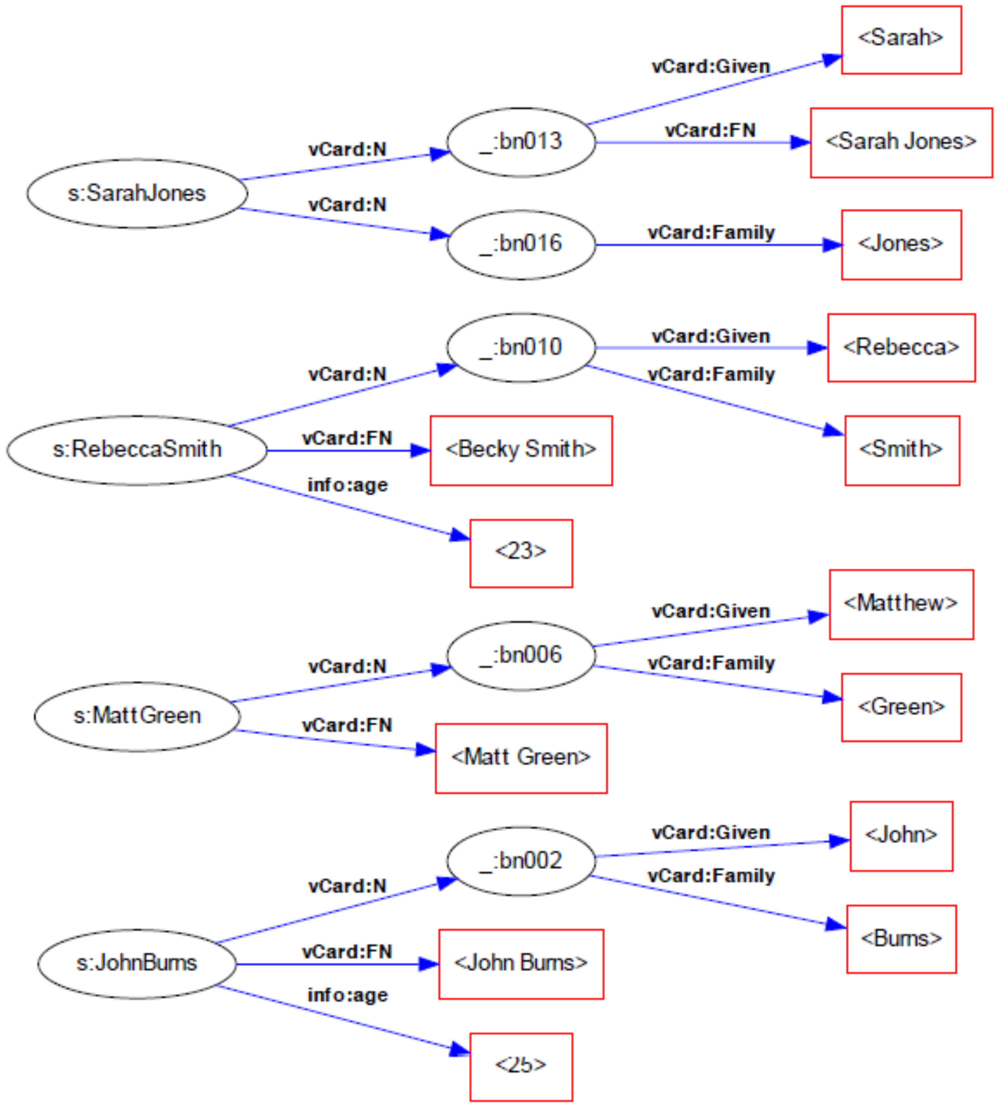
```
@prefix vCard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix info: <http://somewhere/peopleInfo#> .
@prefix s: <http://somewhere/> .

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
        vCard:N [ vCard:Family "Jones" ] .
s:JohnBurns
        vCard:FN "John Burns" ;
        vCard:N [ vCard:Family "Burns" ;
                  vCard:Given "John" ] .
```

Jena vCard 2: Graph



Jena vCard Merged: Graph



Jena vCard: Merged

- 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)

OPTIONAL in SPARQL Querying

Query

Query

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

SELECT ?name ?age
WHERE
{
  ?person   vcard:FN   ?name .
  ?person   info:age   ?age .
}
```

Query

Results

```
| name            | age |
=====================
| "John Burns"    |  25 |
| "Becky Smith"   |  23 |
-------------------------------
```

- 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

Query

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

SELECT ?name ?age
WHERE
{
  ?person   vcard:FN   ?name .
   OPTIONAL  {  ?person   info:age   ?age . }
}
```

Query

Results

```
| name              | age |
=====================
| "John Burns"      |  25 |
| "Matt Green"      |     |
| "Becky Smith"     |  23 |
| "Sarah Jones"     |     |
--------------------------------
```

- OPTIONAL gives SPARQL the ability to not fail a query when specific data does not exist.

Linked Data Principles

Linked Data Principles

1. Use URIs as names for things.
2. Use HTTP URIs, so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
4. 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

Homepage of School of Informatics
http://www.inf.ed.ac.uk/

Homepage of Jacques Fleuriot
http://www.inf.ed.ac.uk/people/staff/Jacques_Fleuriot.html

- These URIs point to web documents - or in the terminology of WebArch (https://www.w3.org/TR/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
- Client can say prefers language X over Y.
- Or prefers RDF over HTML
- Servers inspect headers and select an appropriate response.

Header of GET request

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

Server's Response (excerpt)

HTTP/1.1 200 OK
Content -Type: text/html
Content-Language: en

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 dereferencing 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

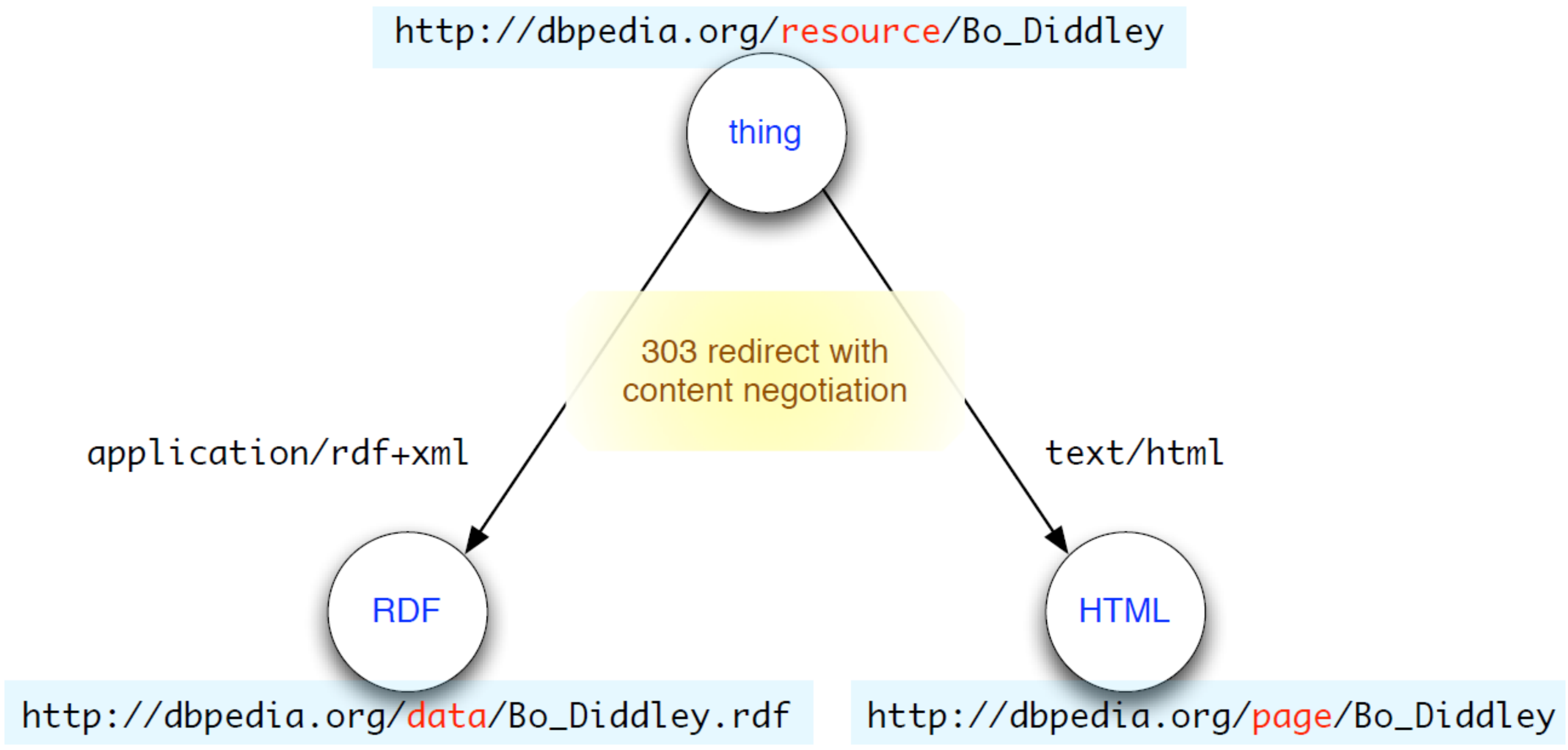
http://dbpedia.org/resource/Bo_Diddley [resource]
http://dbpedia.org/data/Bo_Diddley.rdf [RDF description]
http://dbpedia.org/page/Bo_Diddley [HTML description]

HTTP Response Header (Excerpt) when accessing URI: http://dbpedia.org/resource/Bo_Diddley using a standard browser:

Example:

Status: HTTP/1.1 303 See Other
Date: Sun, 07 Feb 2016 11:05:17 GMT
Content-Type: text/html; charset=UTF-8
Server: Virtuoso/07.20.3215 (Linux) i686-generic-linux-glibc212-64 VDB
Location: http://dbpedia.org/page/Bo_Diddley

Solution 1: 303 (See other) URIs



Solution 2: Hash URIs

Hash URI

http://homepages.inf.ed.ac.uk/jdf/foaf.rdf#jdf

- 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 vs 303

- 303 redirects:
  - can be configured separately for each resource.
  - but two HTTP requests required to retrieve a single description of a real-world object.
- 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.
- In practice we tend to use:
  - 303 URIs for resource descriptions that are part of very large datasets
  - hash URIs to identify terms within RDF vocabularies

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, whereby a statement is modelled as a resource referenced by another statement (hard to query with SPARQL)
  - Collections e.g. lists and containers e.g. bags (also hard to query with SPARQL). 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. Linked Data (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

5-★ Data


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

Reflecting on Linked 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

Reflecting on Linked Data

- 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.

Summary

- Merging graphs that contain blank nodes
  - can be problematic
- SPARQL OPTIONAL
  - so that a query doesn't fail when specific data does not exist
- Linked Data principles
  - Naming things with URIs
  - Making URIs dereferenceable
  - Providing useful RDF information
  - Including links to other things

Reading

- *Linked Data: Evolving the Web into a Global Data Space* Chapter 2
- Non-compulsory additional reading:
  - Ngomo, A. C. N., Auer, S., Lehmann, J., & Zaveri, A. (2014). *Introduction to Linked Data and Its Lifecycle on the Web*. In Reasoning Web. Reasoning on the Web in the Big Data Era (pp. 1-99). Springer.

Practical Task (Optional)

- Task description (step by step how-to)

  http://www.inf.ed.ac.uk/teaching/courses/sws/linkeddata.html

- Why do it?
  - It brings together SWS topics, practical experience, “learning by doing”.