

Medical Informatics

Lecture 9: RDF Data Model

Dr Areti Manataki



Nanjing Medical University

In the previous lecture

- Semantic web data: connected, machine-understandable data
- Ontologies
 - “a formal, explicit specification of a shared conceptualisation”
 - a way of encoding domain knowledge
 - components: classes, attributes, relations, etc.
- Frames
 - class taxonomy, slot with values
 - ISA vs IO

In this lecture

- RDF data model
- Using URIs to uniquely identify resources
- Merging RDF data
- RDF syntax & vocabularies

Semantic Web data

- Connected, machine-readable data
- Also known as linked data or knowledge graph data
- Proposed by Sir Tim Berners-Lee, the father of the World Wide Web
 - Further info at https://www.ted.com/talks/tim_berniers_lee_the_next_web
- The RDF (Resource Description Framework) data model is at the heart of the semantic web.
- “Resource” in RDF refers to entities or things in the world, e.g. Peter, the Mona Lisa painting, heart disease, etc.

RDF triples

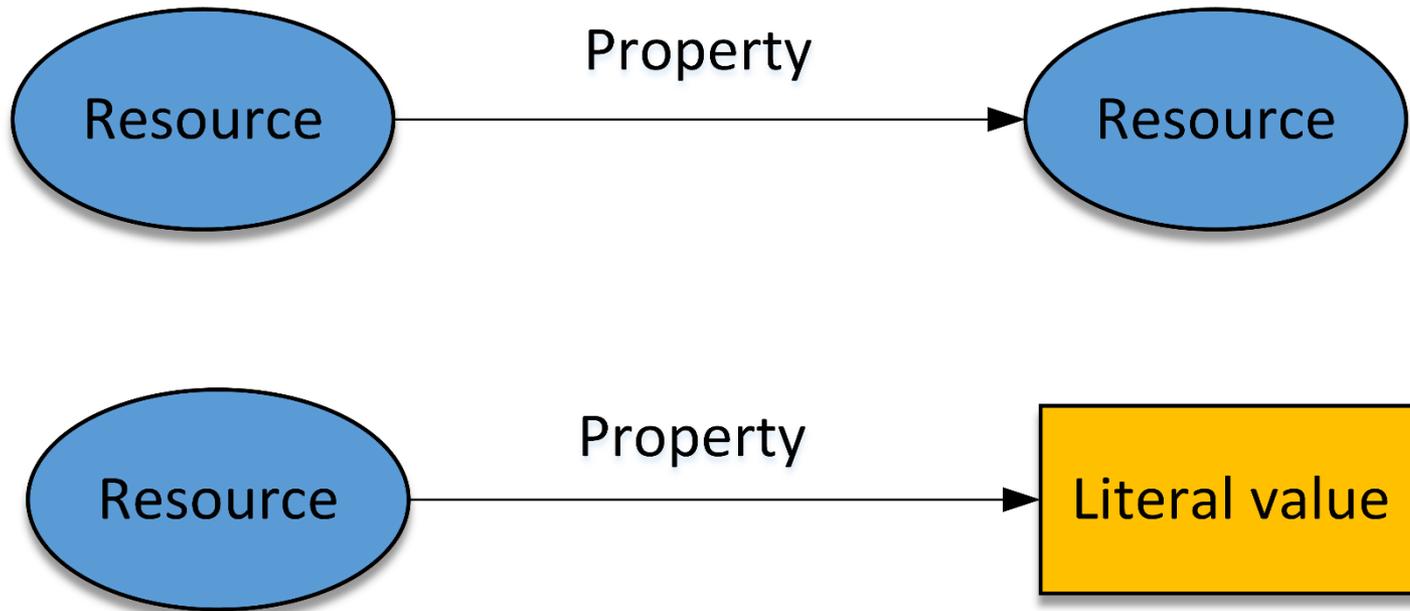
- RDF statements identify:
 - a **resource** being described
 - a specific **property**
 - **value** of the property

subject predicate object
Robert Burns wrote A Red Red Rose

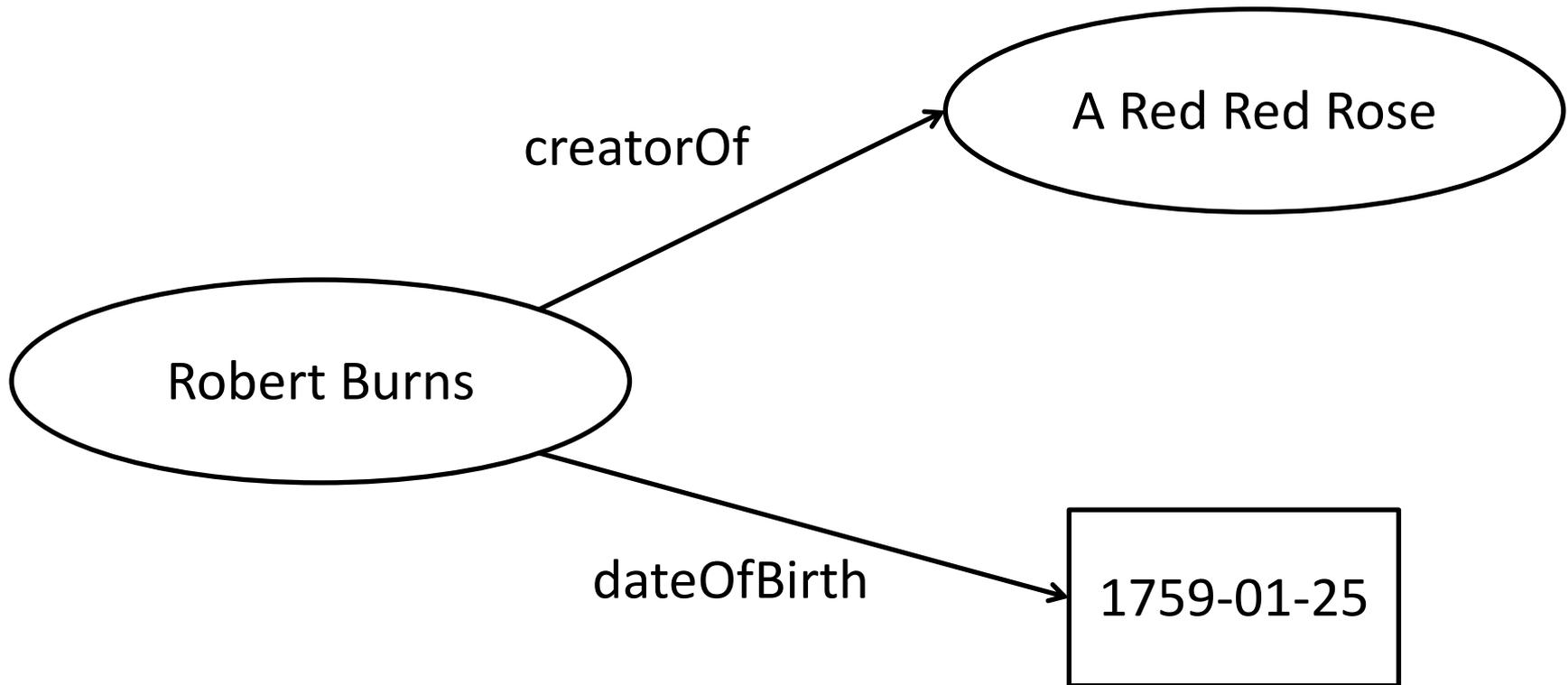
subject predicate object
Robert Burns has birth date 1759-01-25

- We typically refer to **RDF triples**, consisting of a subject, a predicate and an object.
- Subjects can only be resources, while objects can be literals (e.g. strings) or resources.
- Further info at <https://www.w3.org/TR/rdf11-primer/>

RDF triple visualisation



RDF triple visualisation



Unique Identifiers: URIs

- We often need to deal with ambiguous and synonymous names.
 - e.g. which Robert Burns are we referring to here?
Robert Burns wrote A Red Red Rose
- We use **URIs** (Uniform Resource Identifiers) to uniquely identify concepts and entities.

http://dbpedia.org/resource/Robert_Burns

<http://www.WorkingOntologist.org/Examples/Chapter3/Shakespeare#Shakespeare>

- A URI provides a global identification for a resource that is common across the Web.

How to use URIs (1)

- 1st approach (recommended): use existing URIs
 - DBPedia (<http://dbpedia.org>) is a very good source of URIs.
 - Every resource that is the subject of a page in Wikipedia has a corresponding URI in DBpedia.
 - URI for Johann Strauss I:
http://dbpedia.org/resource/Johann_Strauss_I
 - MusicBrainz (<http://musicbrainz.org>) is a user-maintained 'metadatabase' for music.
 - Each artist receives an ArtistID
 - URI for Johann Strauss I:
<https://musicbrainz.org/artist/725fb443-0a26-42f8-b4b7-5257b3a61eb5>

How to use URIs (2)

- 2nd approach: create your own URIs
 - Keep out of namespaces you do not control, e.g. <http://www.imdb.com/>
 - If you don't own a domain name, you can make one up, e.g. <http://example.com/>
<http://www.example.com/id/EwanMcGregor>
 - Keep it simple

QNames

- Instead of writing out the entire URI, we can use XML Qualified Names (QNames).
- QName form: *namespace:Local_name*
 - ex: EwanMcGregor
- In order to create a QName for a particular resource, we:
 1. associate a **prefix** with a URI

Prefix	Namespace URI
edstaff	http://www.ed.ac.uk/staffid#
dc	http://purl.org/dc/terms/
dbpedia	http://dbpedia.org/resource/

QNames

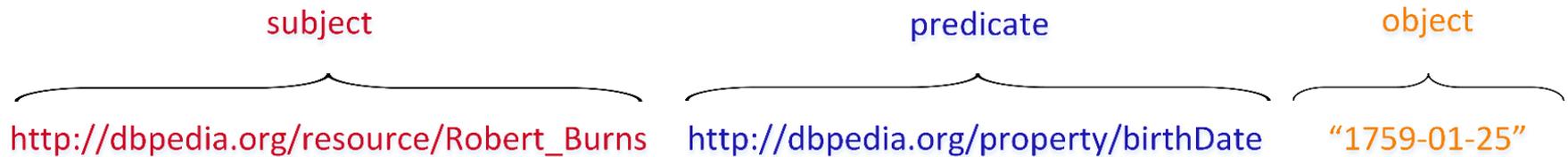
- Instead of writing out the entire URI, we can use XML Qualified Names (QNames).
- QName form: *namespace:Local_name*
 - ex:EwanMcGregor
- In order to create a QName for a particular resource, we:
 1. associate a **prefix** with a URI
 2. follow it with a colon (:) and a **local name**
 - edstaff:amanatak
 - dc:creator
 - dbpedia:Johann_Strauss_I

QNames

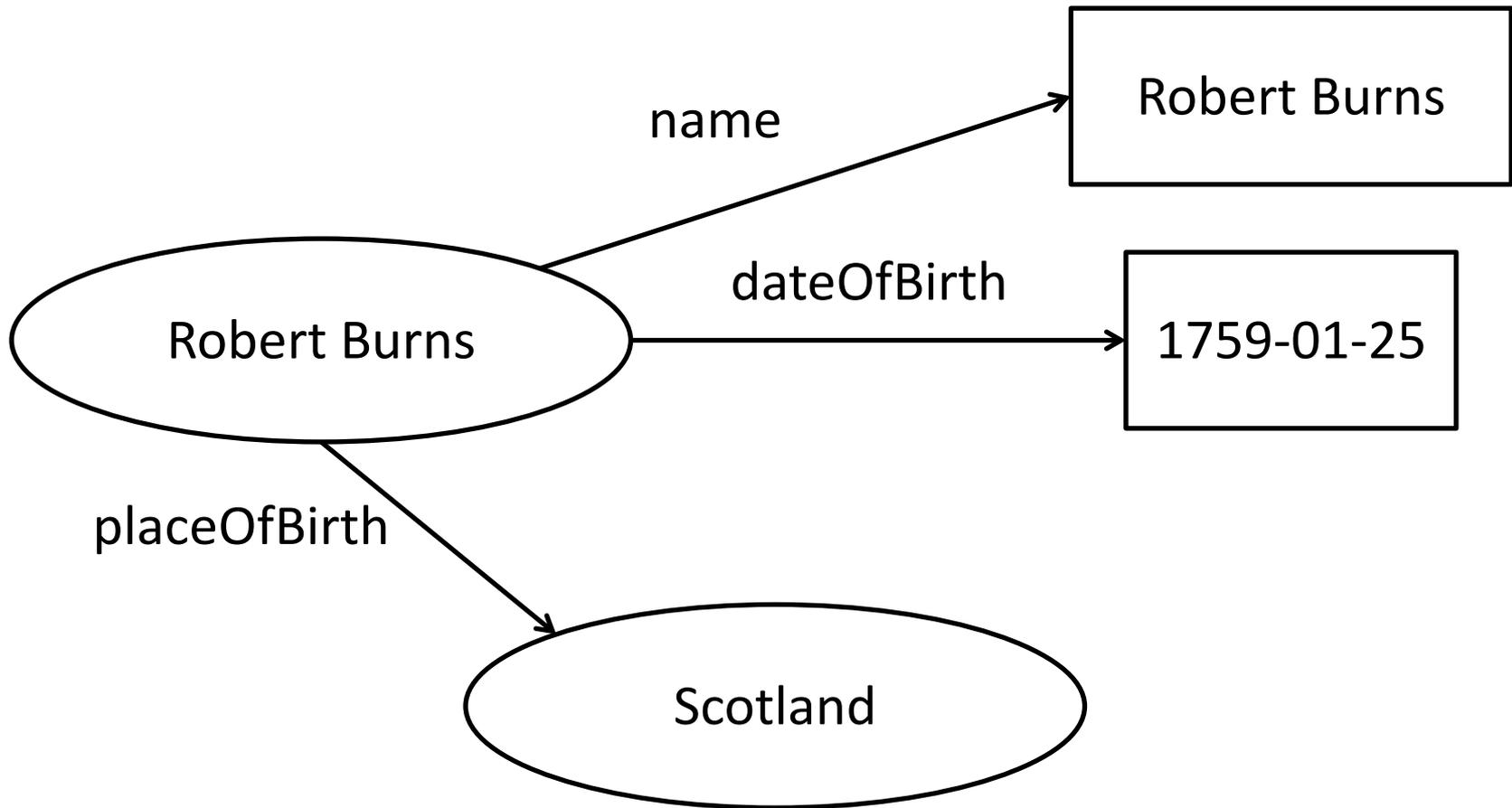
- But remember: QNames are not URIs!
 - Always include a declaration of the namespace correspondence when using a QName
- How do we convert QNames back to full URIs?
 1. replace the prefix plus colon by the Namespace URI
 2. append the local name

RDF triples with the use of URIs

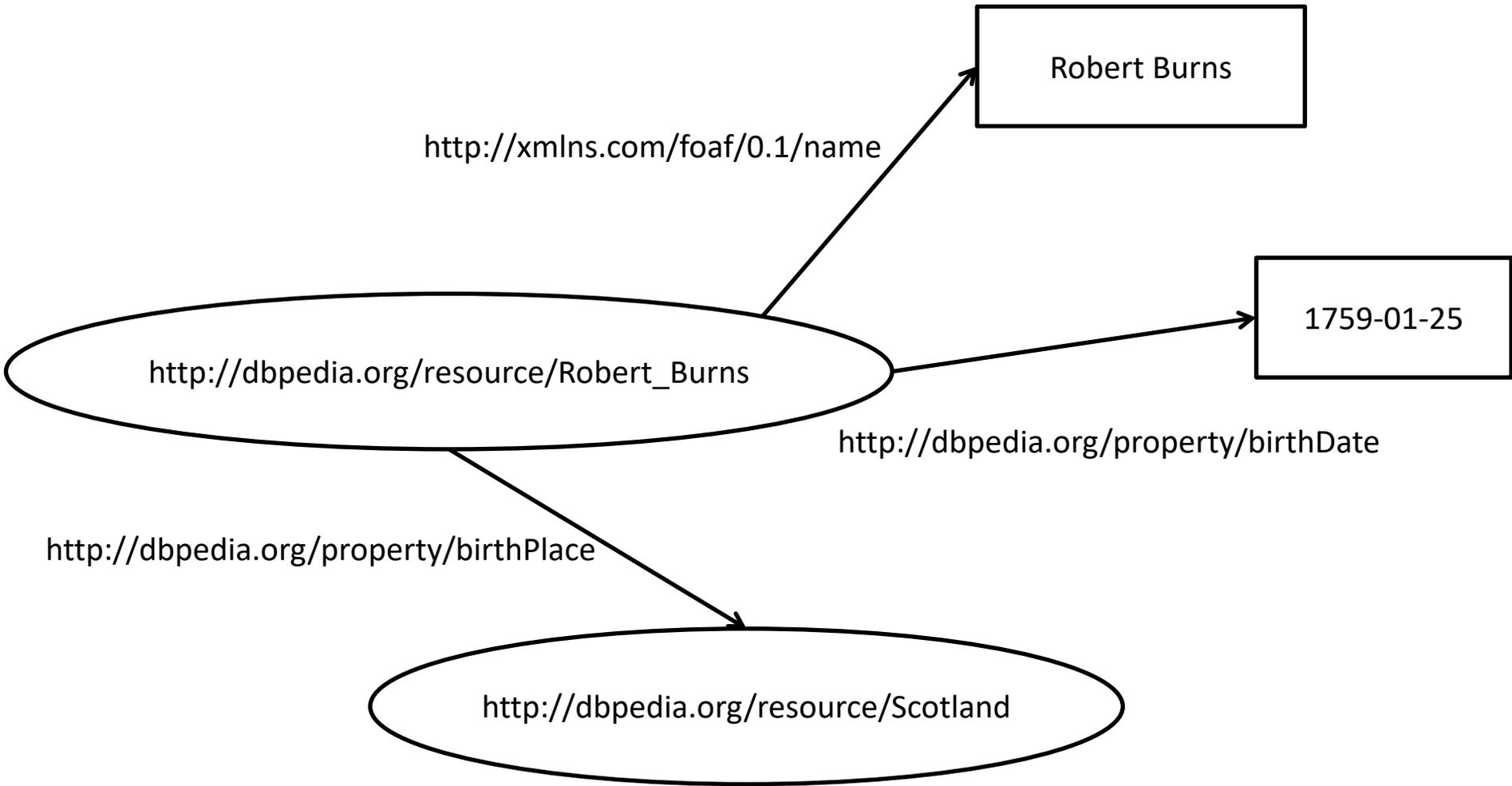
- It is best practice to use URIs to uniquely identify resources.
- So “Robert Burns has birth date 1759-01-25” is captured like this:



RDF example

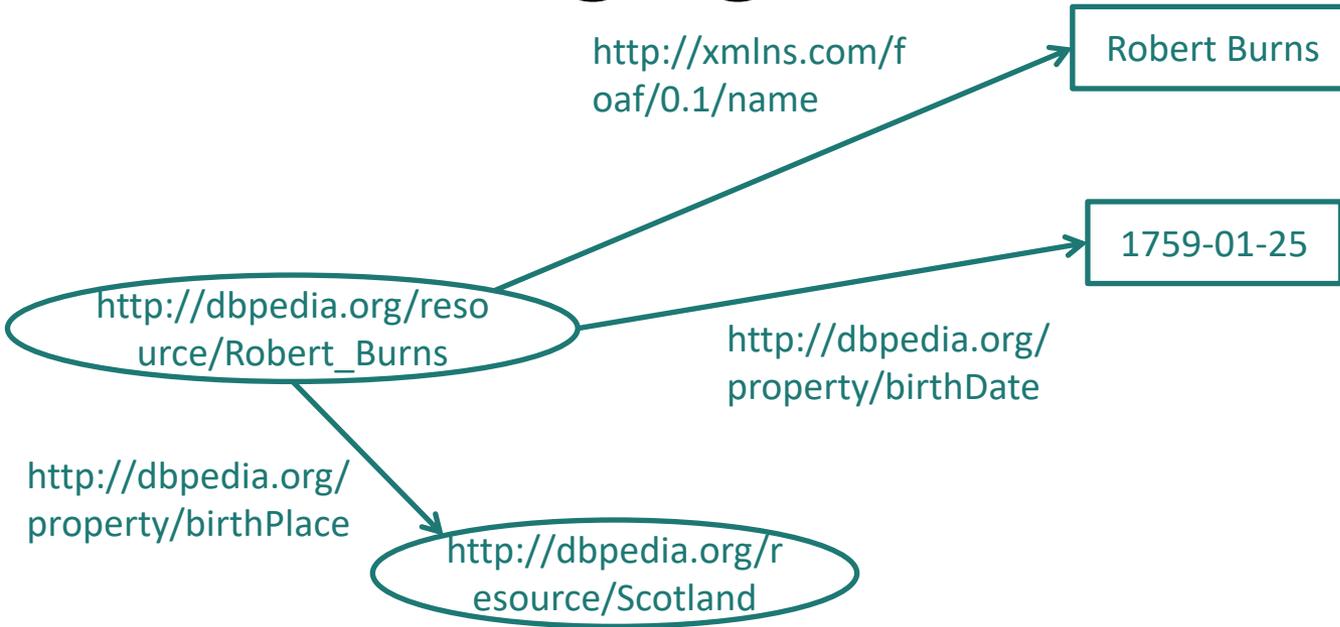


RDF example with URIs

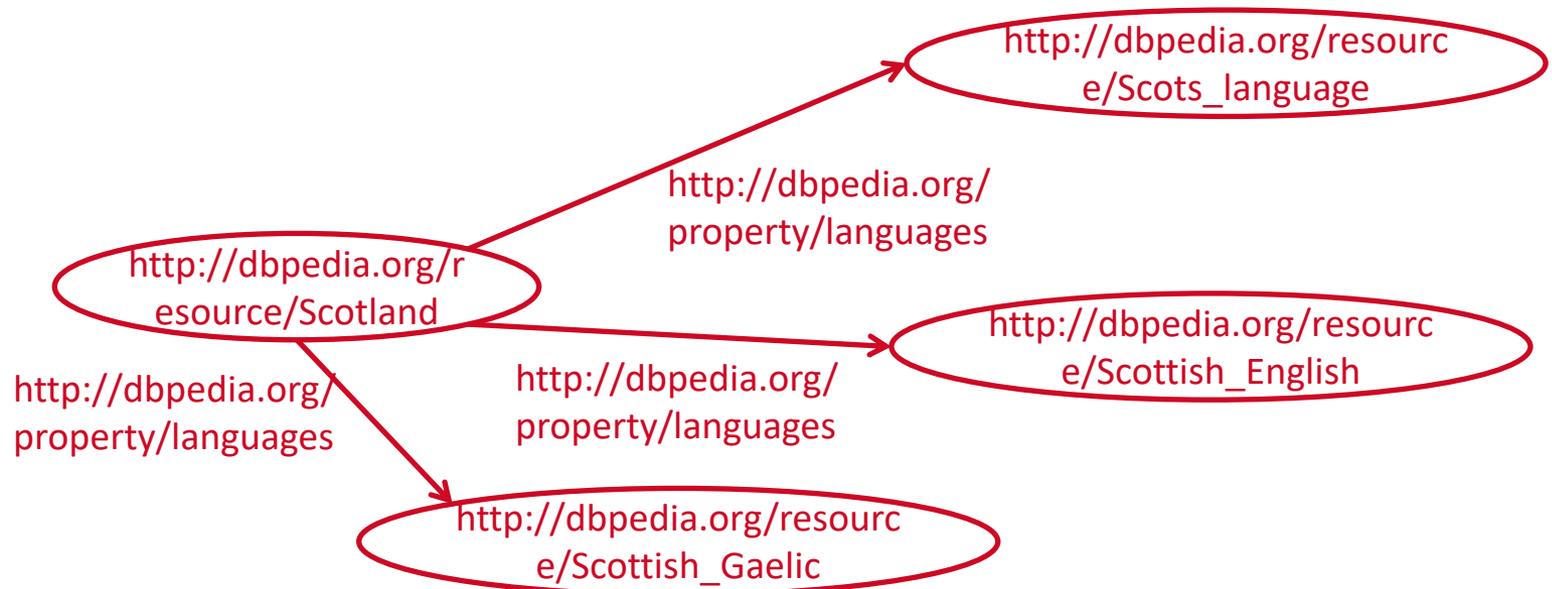


Merging RDF data

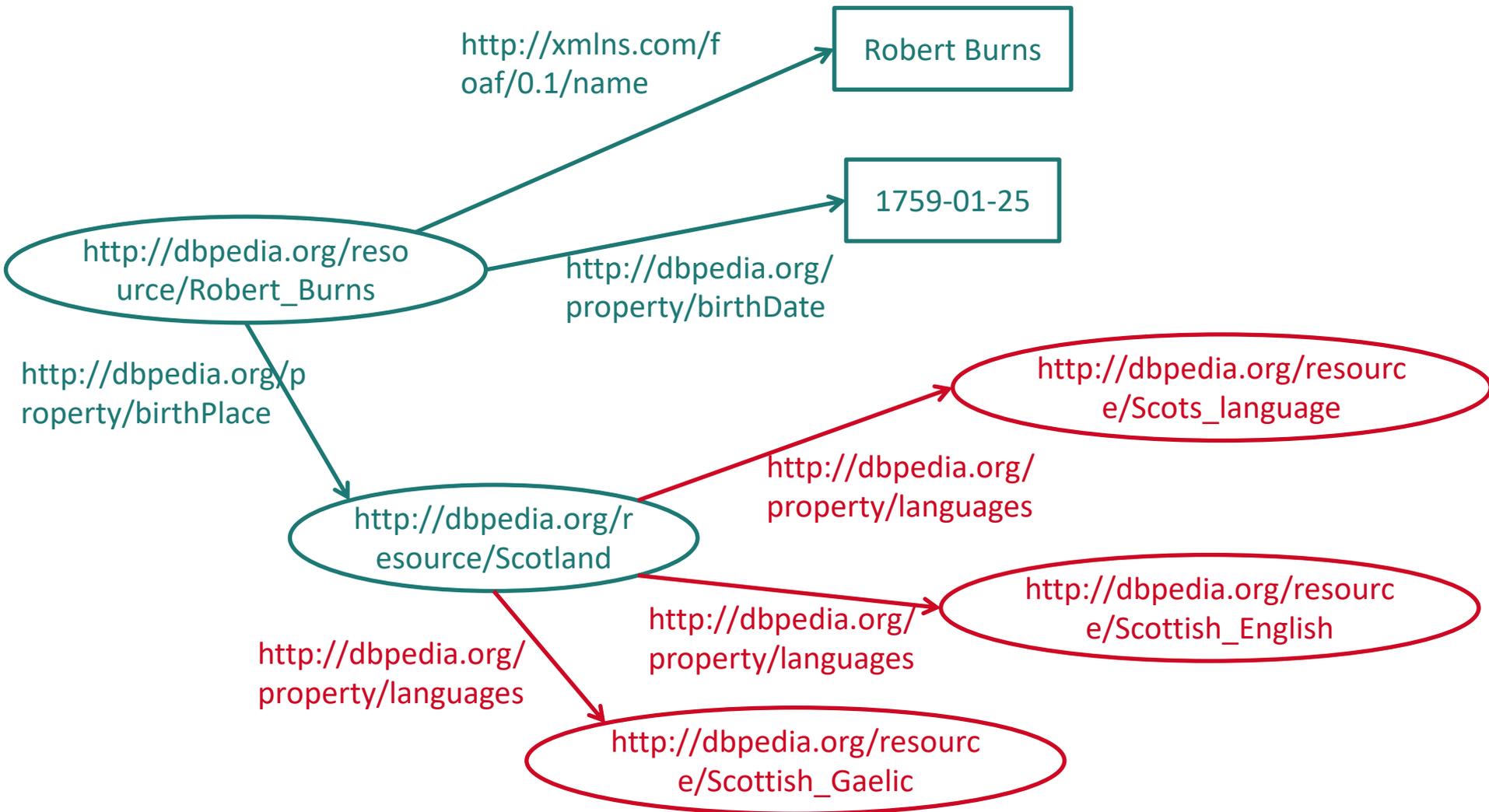
Merging RDF data is easy



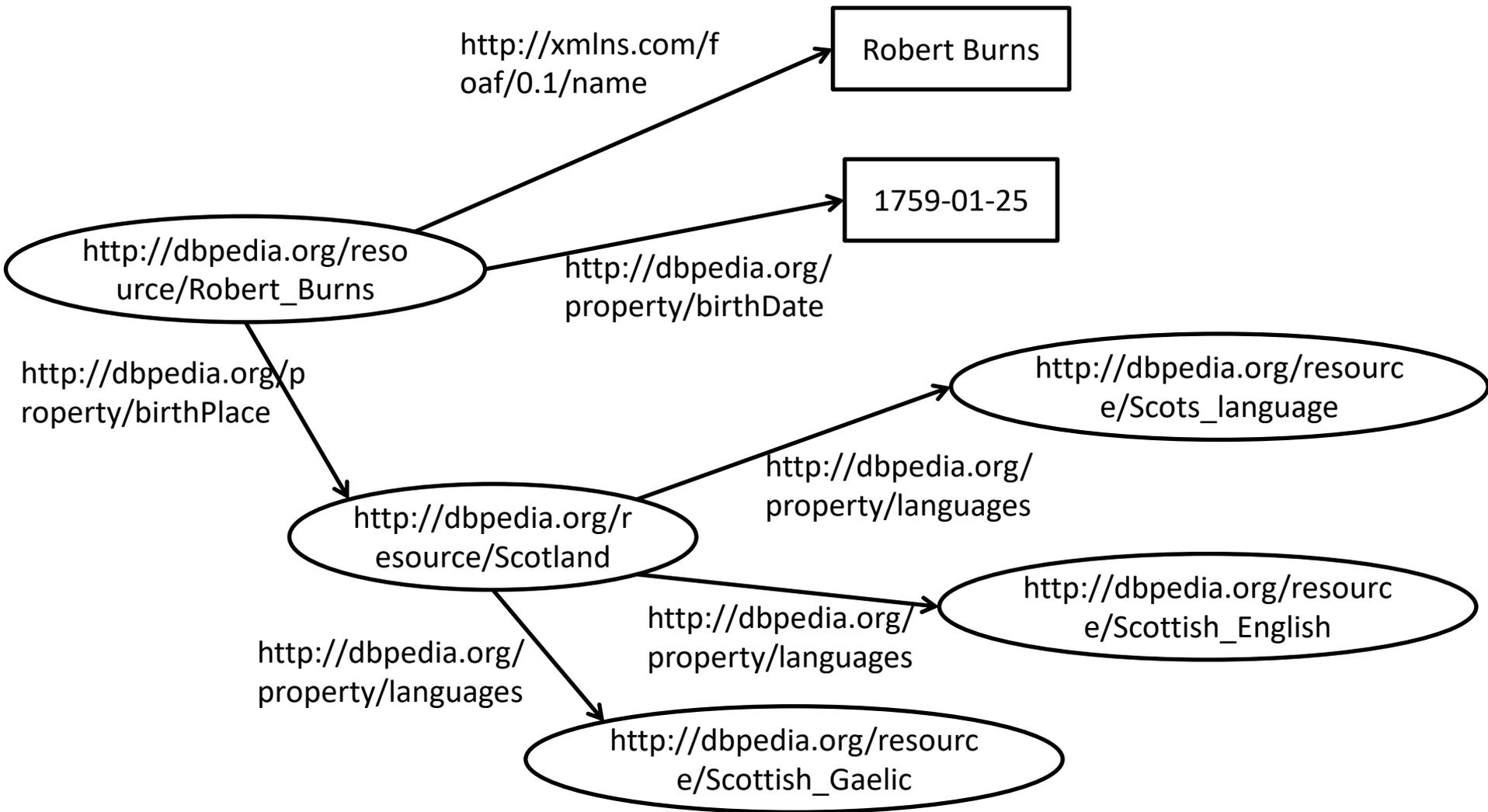
Merging RDF data is easy



Merging RDF data is easy



Merging RDF data is easy



RDF syntax and vocabularies

Processing RDF Statements

- RDF is designed to make **machine-processable** statements.
- Two things required:
 1. a machine-processable syntax for expressing RDF statements (aka serialization format): **Turtle**, XML and others
 2. a machine-processable system for unambiguously identifying subjects, predicates and objects: **URIs**

RDF serialisation

- Various forms of syntax for expressing RDF: XML, N3, Turtle, N-Triples, N-Quads
- XML was the first serialisation standard for RDF, but it is not very intuitive.

```
<rdf:Description rdf:about="http://usher.ed.ac.uk/medinf/resource/aroast">  
  <dbp:name>Artisan Roast</dbp:name>  
</rdf:Description>  
<rdf:Description rdf:about="http://usher.ed.ac.uk/medinf/resource/aroast">  
  <dbo:location  
rdf:resource="http://usher.ed.ac.uk/medinf/resource/eastEdinburgh"/>  
</rdf:Description>  
</rdf:RDF>
```

- N3 (Notation 3) was developed as a simpler human-readable syntax. This has been superseded by Turtle.

RDF serialisation: Turtle

- Turtle stands for “Terse RDF Triple Language”
- It allows an RDF graph to be completely written in a compact and natural text form, with abbreviations for common usage patterns and datatypes.
- Usable within SPARQL RDF queries.
- Further information:
<https://www.w3.org/TR/turtle/>

RDF serialisation: Turtle

- RDF Triples with full URIs:

```
<http://usher.ed.ac.uk/medinf/resource/aroast>
  <http://dbpedia.org/property/name> "Artisan Roast" .
<http://usher.ed.ac.uk/medinf/resource/aroast>
  <https://dbpedia.org/property/locatedIn>
  <http://usher.ed.ac.uk/medinf/resource/eastEdinburgh> .
<http://usher.ed.ac.uk/medinf/resource/aroast>
  <http://usher.ed.ac.uk/medinf/vocab/stars> "5" .
```

- RDF Triples with prefixes:

```
@prefix dbp: <http://dbpedia.org/property/> .
@prefix usherres: <http://usher.ed.ac.uk/medinf/resource/> .
@prefix ushervoc: <http://usher.ed.ac.uk/medinf/vocab/> .
usherres:aroast dbp:name "Artisan Roast" .
usherres:aroast dbp:locatedIn usherres:eastEdinburgh .
usherres:aroast ushervoc:stars "5" .
```

RDF vocabularies

- An RDF vocabulary is a set of **URIs**, not words.
- A vocabulary typically consists of classes (e.g. Person, City, Anger) and properties (e.g. name, age, knows).
- An organisation can define its own vocabulary, using its own URI prefix.
- You can also create your own vocabulary terms
 - using a domain name that you control
 - or using `http://example.com/` or `http://usher.ed.ac.uk/medinf/resource/` and `http://usher.ed.ac.uk/medinf/vocab/`
- It is best practice to use **shared** vocabularies.
- You can search for existing vocabularies at <https://lov.okfn.org/dataset/lov/> and <http://vocab.cc/>

Example: FOAF vocabulary

- The FOAF vocabulary describes people and their relationships.
- Defined at <http://xmlns.com/foaf/0.1/>
- Commonly used FOAF classes include: Person, Organization, Project, Document, Image, etc.
- Commonly used FOAF properties include: name, knows, homepage, based_near, mbox, etc.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix : <http://usher.ed.ac.uk/medinf/resource/> .
:areti foaf:name "Areti Manataki" .
:areti foaf:based_near dbpedia:Edinburgh .
:areti foaf:knows <https://www.w3.org/People/Berners
Lee/card#i> .
```

Example: Dublin Core vocabulary

- Dublin Core describes digital and physical resources, e.g. videos and books.
 - Also known as Dublin Core Metadata Element Set
 - Metadata is data about data
- Defined at <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>
- It consists of 15 elements (properties): title, creator, type, format, publisher, date, language, etc.

```
@prefix dc: <http://purl.org/dc/terms/> .
```

```
@prefix dbr: <http://dbpedia.org/resource/> .
```

```
dbr:The_Wolf_of_Wall_Street    dc:title    "The Wolf of Wall Street" .
```

```
dbr:The_Wolf_of_Wall_Street    dc:date    "2013" .
```

```
dbr:The_Wolf_of_Wall_Street    dc:creator  dbr:Martin_Scorsese .
```

Conclusions

- RDF data model
 - Triples: subject predicate object
 - Visualised as a graph (knowledge graph)
- It is best practice to use URIs to uniquely identify resources.
- Merging RDF data is easy!
- RDF serialisation (e.g. Turtle) & RDF vocabularies (e.g. FOAF)
- In the next lecture we'll learn more about the Turtle syntax.

Acknowledgements

The content of these slides was originally created for the Medical Informatics course from The University of Edinburgh, which is licensed under a Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) license.

These lecture slides are also licensed under a CC BY-SA 4.0 license.

