COMP318

Ontologies and Semantic Web

RDF - Part 2

Dr Valentina Tamma

V.Tamma@liverpool.ac.uk

Where were we

- RDF motivation
- RDF triples

RDF Building Blocks

Statements

Statements are subject-predicate-object triples.

They assert the properties of a resource the resource, a property, and a value

Objects can be resources or literals (atomic values - strings)

Resources are similar to entities in ER models

- –"something" we want to describe
  - E.g. *authors*, *books*, *publishers*, *places*, *people*, *hotels*

Every resource has a URI

- –a URL (Web address) or
- –some other kind of unique identifier: URNs

Advantages of using URIs:

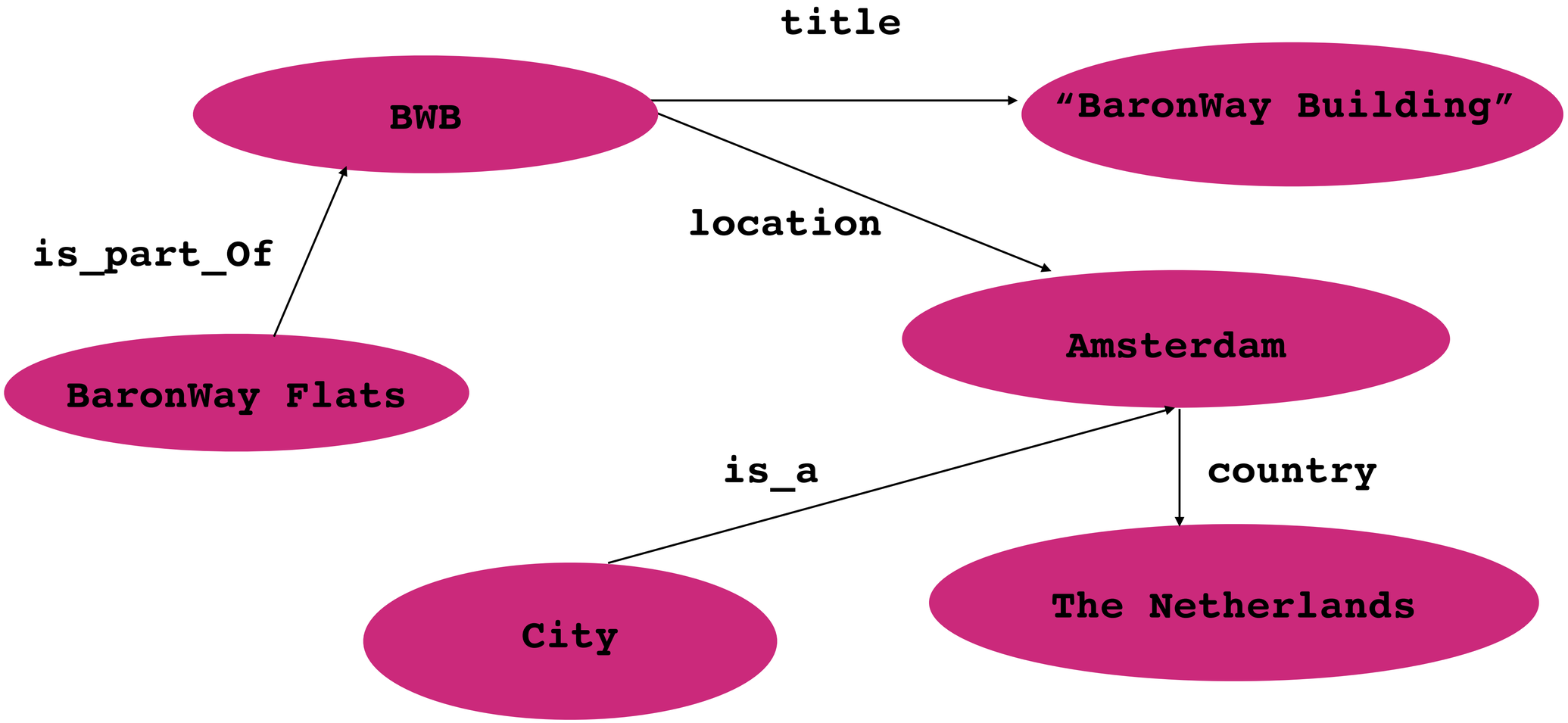
- –a global, worldwide, unique naming scheme
- –reduces the homonym problem in distributed data

Properties

Properties are special types of resources

- –they describe semantic relations between resources
  - E.g. *written by, age, smaller than, etc*
- –they are also identified by a URI

An RDF graph


title
BWB
“BaronWay Building”
location
is_part_Of
Amsterdam
BaronWay Flats
is_a
country
City
The Netherlands

Node and Edge labels in RDF graphs

- Node and edge labels can be:
  - URI
  - Literal (string)
  - Bnode (anonymous label)

- However:
  - Only URIs and Bnodes can be the **subject** of a triple
  - Only URIs can be the predicate of a triple
  - Only URIs, Bnodes and literals can be the **object** of a triple

Same Statements in Turtle syntax

```
@prefix swp:<http://www.swpExample.org/ontology/flats.ttl#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix dbpedia: <http://dbpedia.org/resource>
@prefix dbpedia_owl: <http://dbpedia.org/ontology>
@prefix dc: <http://purl.org/dc/terms>
{
   swp:BaronWayBuilding dc:title “BaronWay Building” .
   swp:BaronWayFlat swp:isPartOf swp:BaronWayBuilding .
   swp:BaronWayBuilding dbpedia_owl:location dbpedia:Amsterdam .
   dbpedia:Amsterdam dbpedia_owl:country dbpedia:TheNetherlands .
   dbpedia:Amsterdam rdfs:subClassOf  dbpedia_owl:Country .    rdf:type dbpedia_owl:City
 }
```

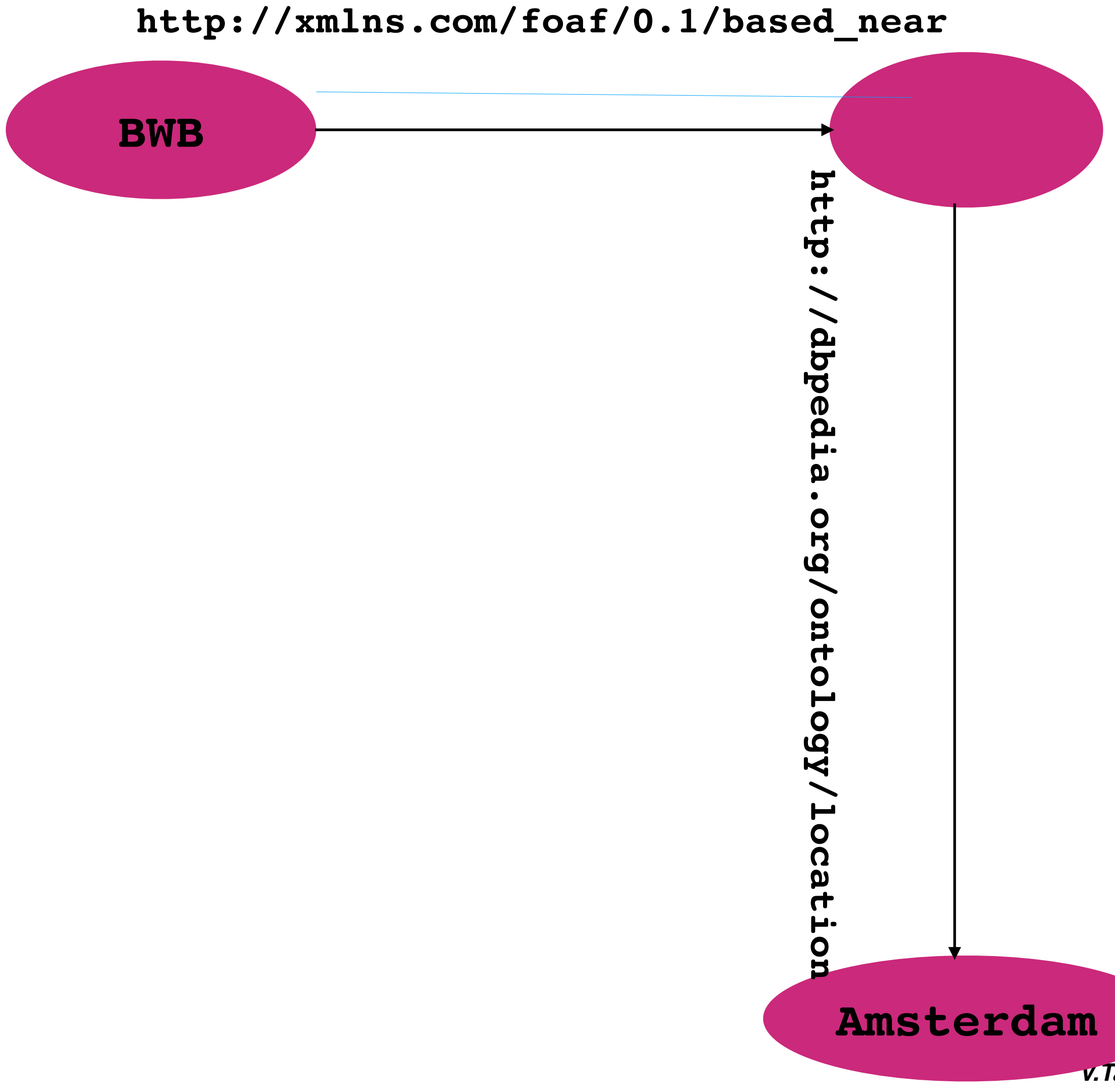
Complex values

- Values of properties do not need to be simple strings
- The value of a property can also be a graph node (corresponding to a resource)
  - arbitrarily complex tree and graph structures are possible
  - Values can be syntactically embedded (i.e., lexically in-line) or referenced (linked)

Blank Nodes

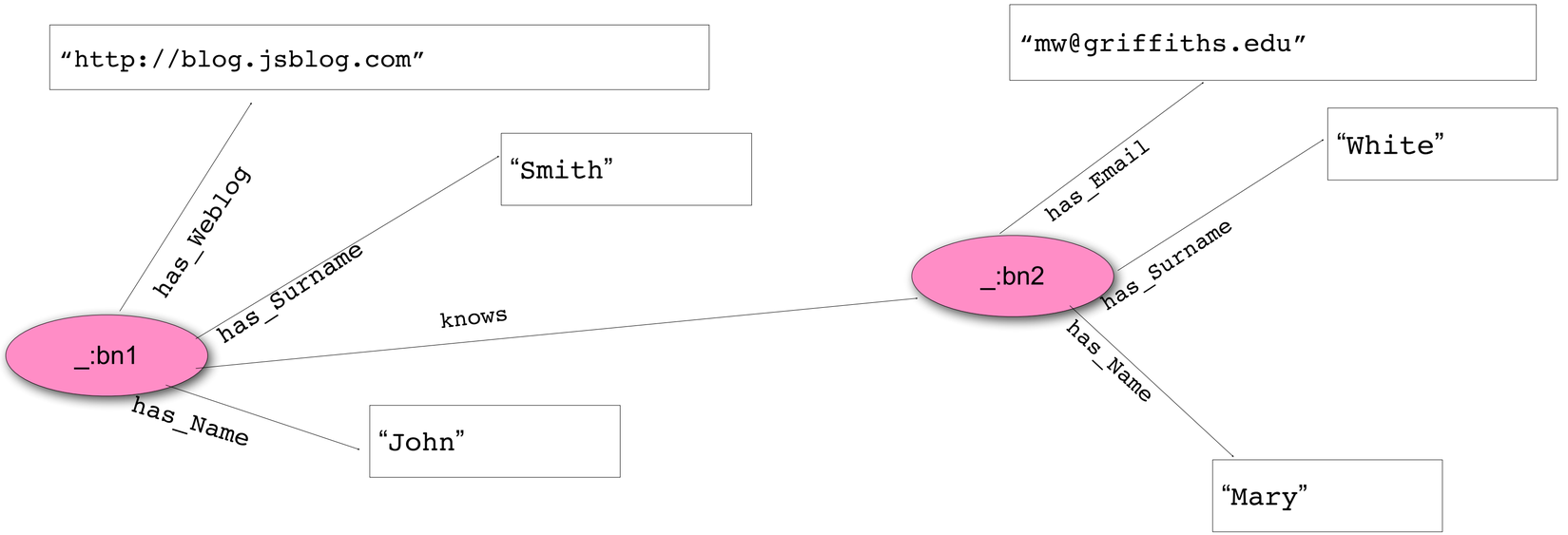
- Blank nodes (bnode) denote an RDF node with “anonymous label”:
  - the node is not associated with a URI
- Bnodes can be used both as subjects and objects
  - _:p1 is the blank node (bnode)

```
ex: BWB   foaf:based_near     _:p1
_:p1      dbpedia:location    Amsterdam
```



Digression: blank nodes

- Social networks APIs do not issue URIs for the members of their community, even if they have lots to say about them.
  - a blank node is used to represent a member and and the facts about the member are linked to the blank node



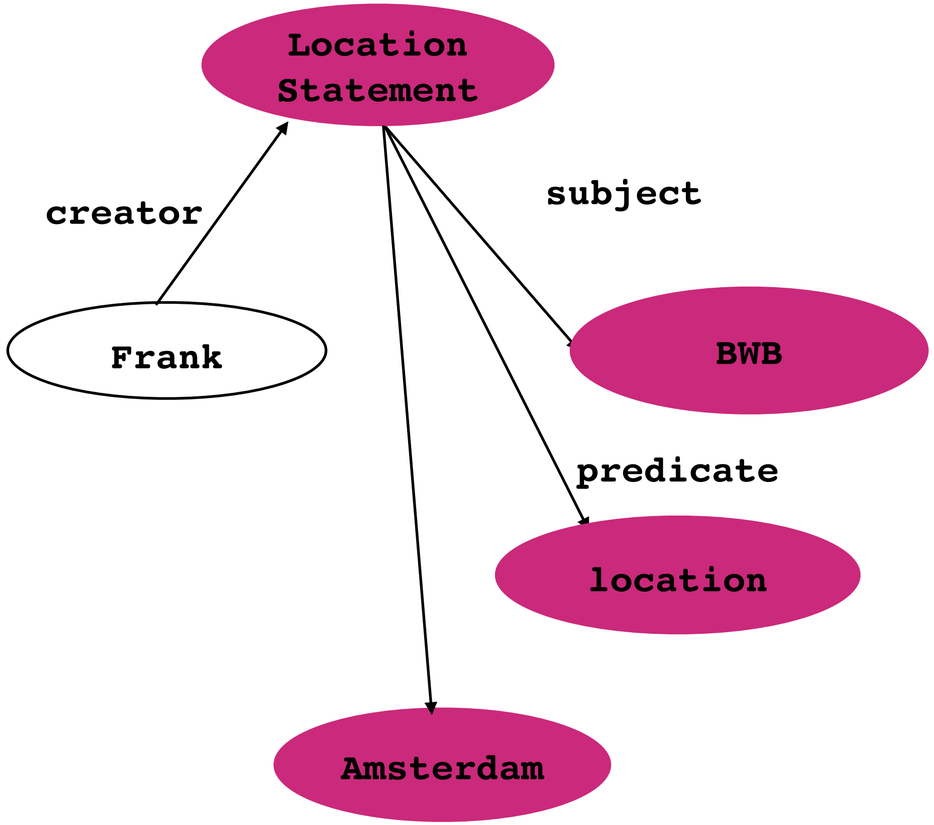
Pointing to statements

- Sometimes it is useful to point to particular statements or to part of a graph
  - especially when assigning a level of trust to a statement
  - or when identifying the source of a statement

- RDF allows you to make statements about other RDF statements
  - “Frank is the creator of the statement about the location of Baron Way Building”
  - Reification and named graphs

Reification

- Reification allows for making complex statements in RDF
  - By talking about statements themselves
  - “Frank is the creator of the statement about the location of Baron Way Building”



- Higher-order statements
  - allow us to express beliefs (and other modalities)
  - are important for trust models, digital signatures,etc.
  - also: metadata about metadata
  - are represented by modelling RDF in RDF itself

Reification

- Any RDF statement can be an object
- We must be able to refer to a statement using a unique identifier
  - allows users to point to a particular statement (and part of a graph)
  - but we cannot add an identifier directly to a triple, otherwise we would make it a quadruple
- RDF allows such reference through a reification mechanism which turns a statement into a resource
  - newer versions of RDF introduce named graphs where an identifier is assigned to a set of statements

Reification vocabulary

- **`rdf:subject`**, **`rdf:predicate`** and **`rdf:object`** allow us to access the parts of a statement

- The ID of the statement, **`Location Statement`**, can be used to refer to it, as can be done for any **`<rdf:description>`**
  - We write an **`<rdf:Description>`** if we don't want to talk about a statement further
  - We write an **`<rdf:Statement>`** if we wish to refer to a statement

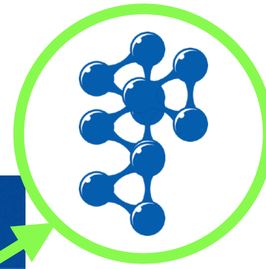
Named Graphs

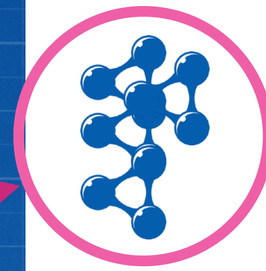
- Because of the overhead of reification, newer versions of RDF introduced the notion of named graphs:
  - an explicit identifier, a URI, is given to one or more of statements.
- This identifier can then be referred to in normal triples.
  - This is a more straightforward mechanism for identifying statements as well as graphs

Named graph example

```
@prefix swp:<http://www.swpExample.org/ontology/flats.ttl#>
@prefix dbpedia: <http://dbpedia.org/resource>
@prefix dbpedia_owl: <http://dbpedia.org/ontology>
@prefix dc: <http://purl.org/dc/terms>
{
   <http://www.swpExample.org/ontology/flats.ttl#>
   dc:creator <http://www.swpExample.org/frank>
}

<http://www.swpExample.org/ontology/apartments.ttl#>
{
   swp:BaronWayFlat swp:hasNumberOfBedrooms 3 .
   swp:BaronWayFlat swp:isPartOf swp:BaronWayBuilding .
   swp:BaronWayBuilding dbpedia_owl:location dbpedia:Amsterdam .
   dbpedia:Amsterdam dbpedia_owl:country dbpedia:TheNetherlands .
 }
```





http://www.swpExample.org/ontology/flats.ttl

n-ary predicates

- RDF only offers binary predicates, but sometimes we need to express predicates that have more than two arguments
  - therefore we need to use some auxiliary resources to split an *n-ary* predicate into *n* binary ones
  - SuttonHomes is the agent in a home sale between seller John and buyer Mary
    - *Agent(SuttonHomes, John, Mary)*

n-ary predicates

- SuttonHomes is the agent in a home sale between seller John and buyer Mary: *Agent(SuttonHomes, John, Mary)*
  - Auxiliary resource **swp:home_sale**

```
@prefix swp:<http://www.swpExample.org/ontology/flats.ttl#>

swp:home_sale swp:agent  swp:SuttonHomes .
swp:home_sale swp:buyer  swp:Mary .
swp:home_sale swp:seller swp:John .
```

COMP318

Ontologies and Semantic Web

End of RDF - Part 2

Dr Valentina Tamma

V.Tamma@liverpool.ac.uk