

Linked Data:
What are they and how to create them.
An introduction

Carlo Meghini

Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche, Pisa

*Accesso aperto al patrimonio culturale digitale e linked open data:
strategie, progetti e nuove opportunità*

Roma, 4 Marzo, 2015

Outline

- ▶ The web
- ▶ The semantic web
- ▶ RDF
- ▶ Vocabularies and ontologies
- ▶ Linked Data
- ▶ How to publish Linked Data

The World Wide Web

The web consists of two main ingredients:

- ▶ a **distributed knowledge base**, where knowledge is expressed informally (text) or pictorially (images, videos, graphics) and is embedded in hypertexts (HTML documents) that provide *links* between the parts of the knowledge base
- ▶ a **mechanism** to access knowledge by GETting the parts that constitute it

Distribution is a key factor for the success of the web:

- ▶ technically, it allows scalability and growth from the bottom up
- ▶ conceptually, each source contributes the knowledge on which it is competent
- ▶ economically, it allows sustainability by sharing the effort

Conceptually, the web is based on a few, simple notions:

- ▶ *resource*: everything that has an identity
 - ▶ In particular, a web resource is a structure accessible on the web
- ▶ *URI*: a string of characters that univocally identifies a resource
- ▶ *state*: the way a resource is at a certain time
- ▶ *representation*: data that encode the state of a resource
 - ▶ a state can be represented by many different representations.

In practice:

URI

```
http://weather.example.com/oaxaca
```

Identifies

Resource

Oaxaca Weather Report

Represents

Representation

Metadata:

Content-type:
application/xhtml+xml

Data:

```
<!DOCTYPE html PUBLIC "...  
    "http://www.w3.org/...  
<html xmlns="http://www...  
<head>  
<title>5 Day Forecaste for  
Oaxaca</title>  
...  
</html>
```

The semantic web

The semantic web is a parallel web, that differs from the original web in the kind of knowledge that is stored and served to the users.

The knowledge found on the semantic web is *formal* knowledge, that is knowledge expressed in a formal language having:

- ▶ a machine-readable notation
- ▶ a formal syntax that is strongly coupled with the web architecture
- ▶ a formal semantics that provides an access mechanism.

The semantic web started as a *vision* by the inventor of the web:

Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. Scientific American Magazine, 2001.

The vision is becoming true via Linked Data.

Linked Data

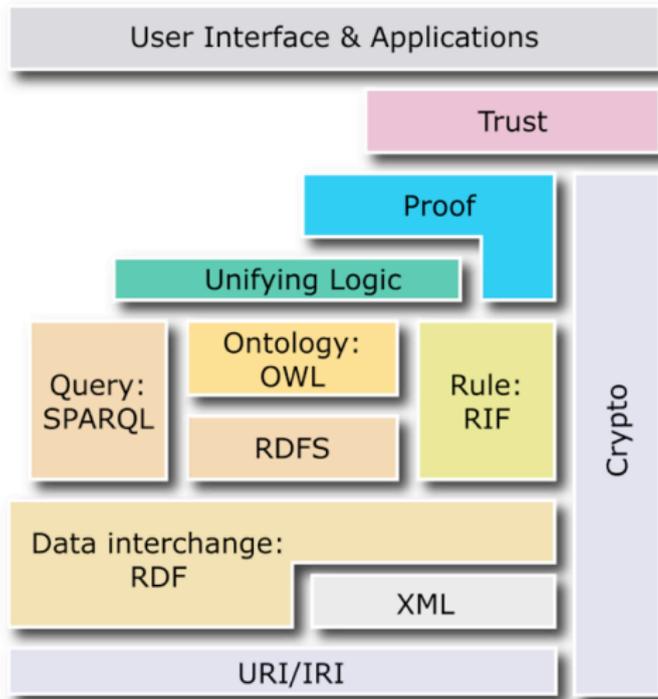
Linked Data are data that follow 4 recommendations:

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.

Ingredients:

- ▶ language: URIs, RDF, SPARQL
- ▶ mechanics: HTTP look up

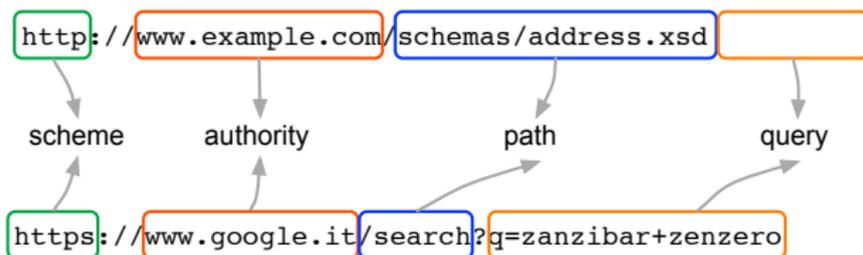
Semantic web architecture



URIs

A URI is a sequence of four main parts:

```
scheme ':' ('://' authority)? path? ('?' query)?
```



Qnames

URIs in XML can be written in an more readable form as *qualified names*.

A qualified name is associated with a **namespace**, a container of names identified by a URI.

There are two kinds of qnames:

- ▶ prefixed names, such as `myns:ricetta`, where:
 - ▶ `myns` is the prefix, and is bound to a namespace
 - ▶ `ricetta` is the local name
- ▶ unprefixed names, such as `ricetta`, where:
 - ▶ the prefix is the *default namespace*
 - ▶ `ricetta` is the name

I can write the URI: `http://www.example.com/people/gelsomina` as

- ▶ `ex:gelsomina` where `ex:` is the prefix bound to the namespace `http://www.example.com/people/`
- ▶ `gelsomina` where the default namespace is `http://www.example.com/people/`

RDF as a knowledge representation language

The *Resource Description Framework* (RDF) is a modern version of *semantic nets*, allowing to express very simple statements about the individuals and the relations in the domain of discourse.

Individuals and relations: our *conceptualization* of the world.

Statements: how we *express* our conceptualization.

RDF as a knowledge representation language

The *Resource Description Framework* (RDF) is a modern version of *semantic nets*, allowing to express very simple statements about the individuals and the relations in the domain of discourse.

Individuals and relations: our *conceptualization* of the world.

Statements: how we *express* our conceptualization.



RDF as a knowledge representation language

The *Resource Description Framework* (RDF) is a modern version of *semantic nets*, allowing to express very simple statements about the individuals and the relations in the domain of discourse.

Individuals and relations: our *conceptualization* of the world.

Statements: how we *express* our conceptualization.



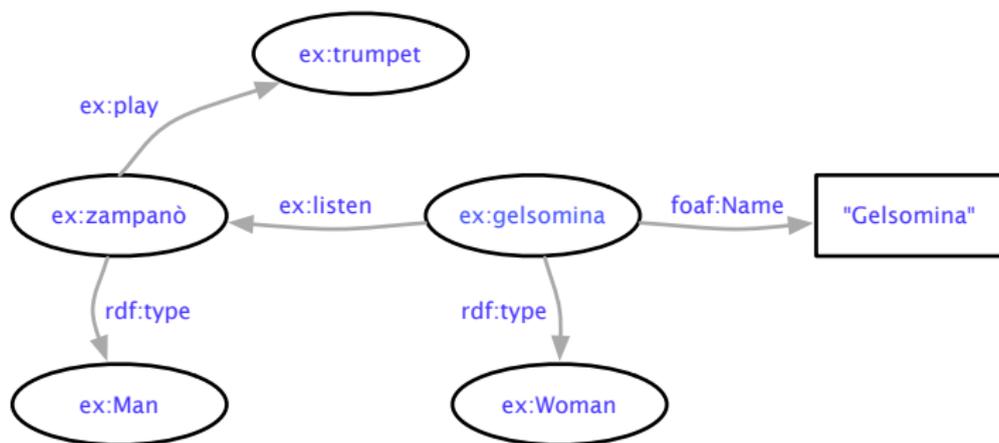
An RDF representation:

```
ex:zampanò rdf:type ex:Man .  
ex:gelsomina rdf:type ex:Woman .  
ex:zampanò ex:play ex:trumpet .  
ex:gelsomina ex:listen ex:zampanò .  
ex:gelsomina foaf:name "Gelsomina" .
```

RDF has a simple graphical notation

A set of RDF triples can be visualized as a directed, labelled graph.

```
ex:zampanò rdf:type ex:Man .  
ex:gelsomina rdf:type ex:Woman .  
ex:zampanò ex:play ex:trumpet .  
ex:gelsomina ex:listen ex:zampanò .  
ex:gelsomina foaf:name "Gelsomina" .
```



RDF has an official XML notation

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF xmlns:ex="http://www.example.org/"
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/spec/>

  <rdf:Description rdf:about="http://www.example.org/gelsomina">
    <foaf:Name>"Gelsomina"</foaf:Name>
    <rdf:type>
      <rdf:Description rdf:about="http://www.example.org/Woman">
        </rdf:Description>
      </rdf:type>
    <ex:listen>
      <rdf:Description rdf:about="http://www.example.org/zampanò">
        <rdf:type>
          <rdf:Description rdf:about="http://www.example.org/Man">
            </rdf:Description>
          </rdf:type>
        <ex:play>
          <rdf:Description rdf:about="http://www.example.org/trumpet">
            </rdf:Description>
          </ex:play>
        </rdf:Description>
      </ex:listen>
    </rdf:Description>
  </rdf:RDF>
```

A resource
named "Gelsomina"
is
a Woman

and listens to
a resource who
is
a Man

and plays
trumpet

RDF has a query language: SPARQL

SPARQL is designed to navigate directed, labelled graphs.

Who is being listened by someone, and **what** are they?

```
SELECT distinct ?ind ?cl
FROM <http://carlo.eu/mygraph>
WHERE
?x ex:listen ?ind .
?ind rdf:type ?cl .
```

RDF has a query language: SPARQL

SPARQL is designed to navigate directed, labelled graphs.

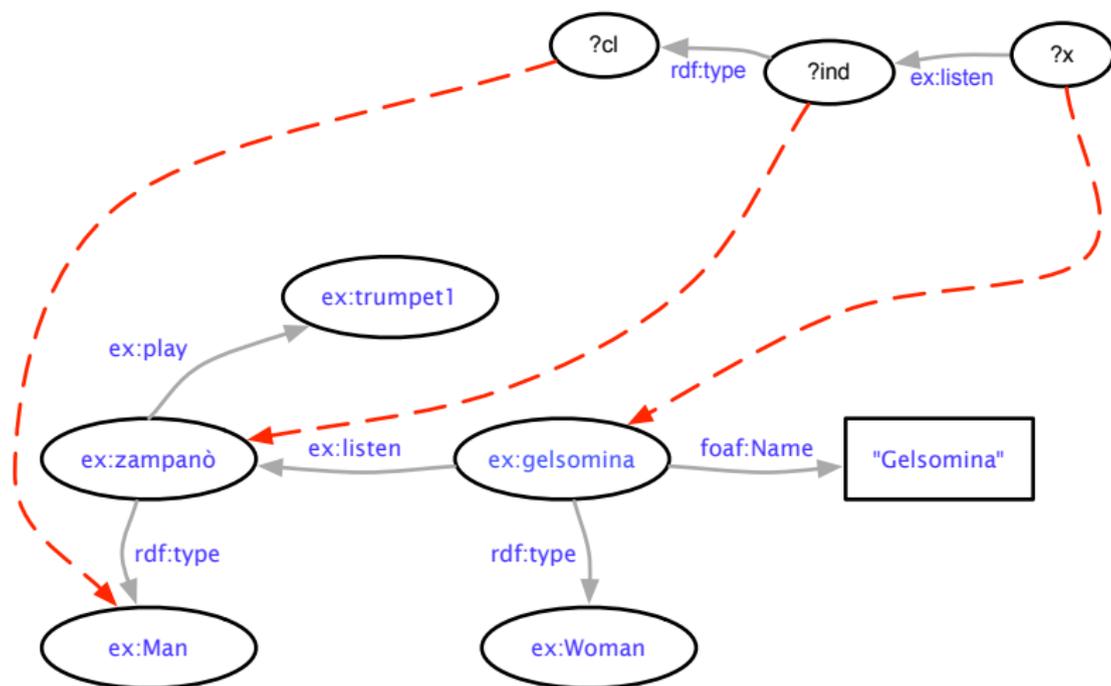
Who is being listened by someone, and **what** are they?

```
SELECT distinct ?ind ?cl
FROM <http://carlo.eu/mygraph>
WHERE
?x ex:listen ?ind .
?ind rdf:type ?cl .
```

A query can be represented as a graph some nodes of which are **variables**:



Query answering is graph matching



Answer: `?ind=ex:zampanò` `?cl=ex:Man`

Vocabularies

The oval nodes in an RDF graph are URIs of individuals.

In well-principled descriptions, these URIs are drawn from *vocabularies*.

There are known vocabularies giving URIs for:

- ▶ place names (such as TGN)
 - ▶ Rome: <http://vocab.getty.edu/tgn/7000874>
- ▶ people names (such as VIAF)
 - ▶ Eugenio Montale: <http://viaf.org/viaf/73857542>
- ▶ concept names (such as Classification schemes, subject headings, and the like)
 - ▶ Informatica (Soggettario Nazionale): <http://purl.org/bnct/tid/1576>

The labels in an RDF graph are URIs of properties, which express linguistically relations between individuals.

Also properties can be drawn from vocabularies, such as:

- ▶ for describing bibliographic resources: Dublin Core
 - ▶ <http://purl.org/dc/terms/creator>
- ▶ for describing museum objects: CIDOC CRM
 - ▶ http://www.cidoc-crm.org/cidoc-crm/P1_is_identified_by
- ▶ for describing archival objects: EAD
 - ▶ *ehm* ... work in progress

Ontologies

If a vocabulary includes axioms then it is called an *ontology*.

- ▶ social ontologies: hasFather, hasFriend, ...
- ▶ space ontologies: point, region, containedIn, ...
- ▶ literary ontologies: text, citation, cites, ...

Axioms in ontologies capture the meaning of terms.

- ▶ the meaning that machines can understand

such as:

- ▶ social axioms: fatherOf is disjoint from motherOf
 - ▶ `DisjointObjectProperties(hasFather hasMother)`
- ▶ space axioms: containedIn is transitive
 - ▶ `TransitiveObjectProperty(containedIn)`
- ▶ literary axioms: a citation relates a fragment of text to a work
 - ▶ `cites rdfs:domain TextFragment .`
 - ▶ `cites rdfs:range Work .`

Why ontologies?

Two main reasons:

1. **Communication efficacy** By endowing our descriptions with the ontologies that define the terms that we use in them, we can make our descriptions *clear*.

- ▶ the richer the vocabulary, the clearer the descriptions

By using shared ontologies, we can make our description *interoperable*.

- ▶ the larger the community that shares the ontology, the greater the interoperability.

2. **Computational utility** we can use special software agents, called *inference engines*, to ensure the quality of our descriptions:

- ▶ Are ontologies consistent?
- ▶ Are descriptions consistent?

and to extract the maximum knowledge from our descriptions

- ▶ *is Plato mortal?*

Ontologies in the Semantic Web

In the semantic web stack, there are two ontology languages:

1. *RDF Schema*, an RDF ontology allowing to express simple definitions of classes and properties
2. *Ontology Web Language*, a family of languages derived from Description Logics, allowing to express sophisticated definitions of classes and properties.

OWL 2 is a W3C Recommendation of October 2009.

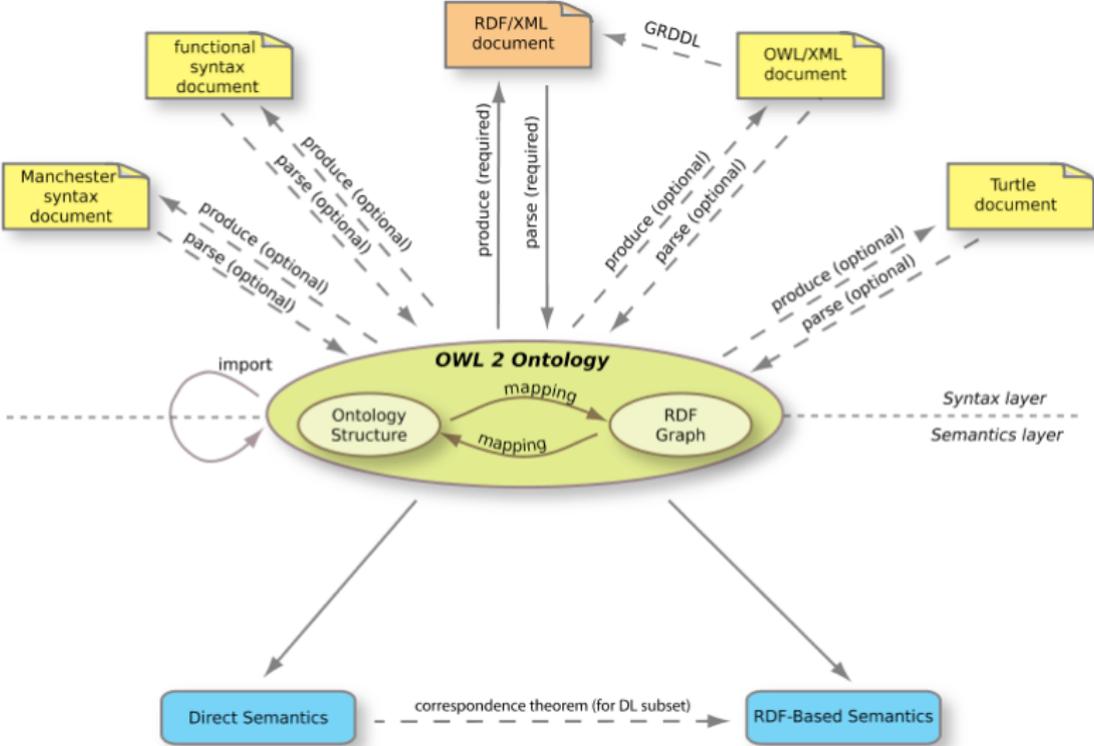
Predecessor: OWL 1, 2004.

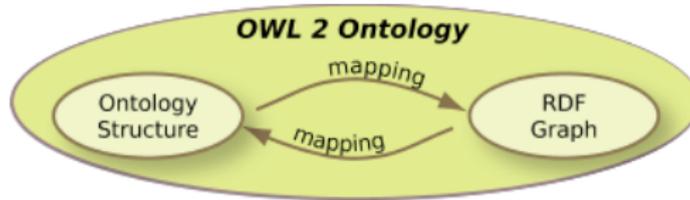
Back compatibility: all OWL 1 ontologies are also OWL 2 ontologies.

OWL 2 extends OWL 1 with:

- ▶ a richer language, fully compatible with OWL 1, but with more constructs adding expressivity
- ▶ three new profiles addressing applications, all efficiently implementable.

Expressing an OWL ontology in RDF

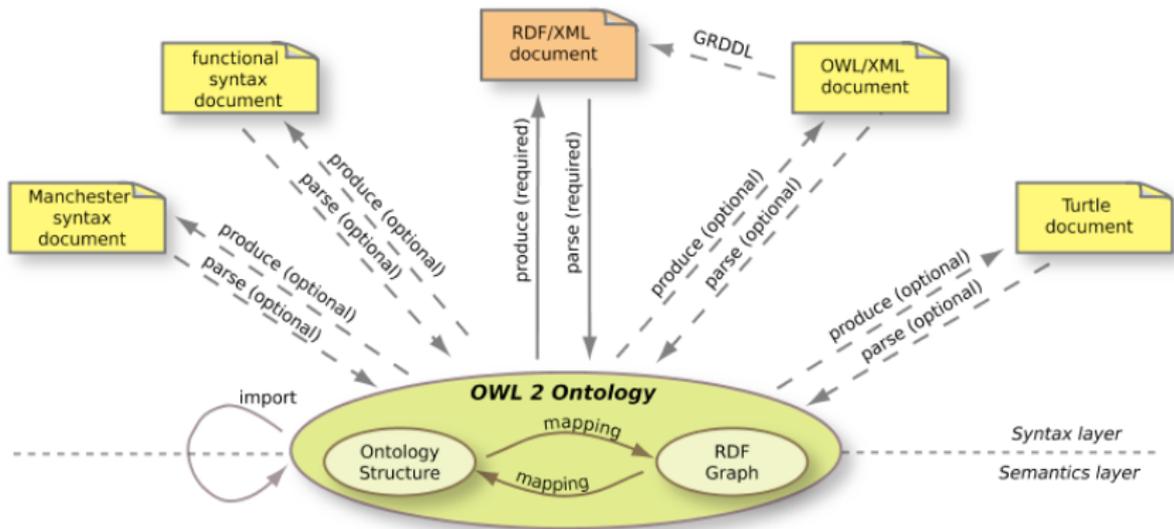




An OWL 2 ontology can be an RDF graph (of a very rich vocabulary) or an equivalent conceptual, specified in UML.

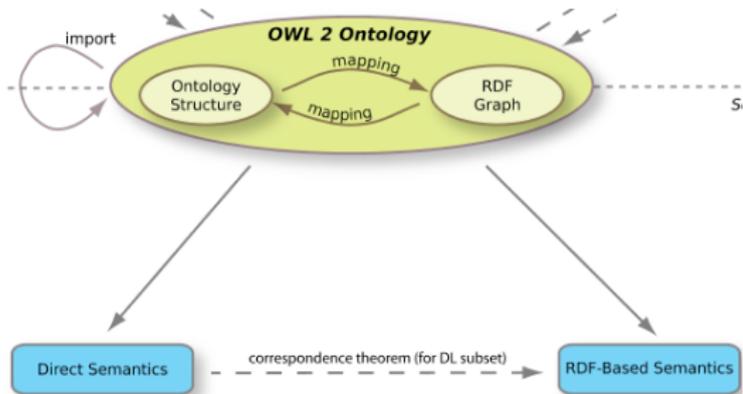
There is a translation between the two, so they are really a syntactic variant of one another.

But both are at an abstract level, they are not serialized, that is expressed in one of the accepted notations.



There are two families of notations:

- ▶ Notations based on RDF: the ontology is an RDF graph
 - ▶ RDF/XML, the only “official” notation
 - ▶ Turtle, easier to read, supported only by some tools
- ▶ Notations based on OWL:
 - ▶ Manchester, very easy to read and write
 - ▶ Functional, directly reflecting the UML conceptual structure
 - ▶ OWL/XML, which can be managed with any XML tool



As a consequence, there are two semantics:

- ▶ RDF-based: assigns meaning to ontologies specified as RDF graphs
- ▶ Direct: assigns meaning to ontologies specified in the functional syntax

The two semantics are mathematically equivalent for a subset of Description Logics.

Linked Data can convey an OWL ontology.

Linked Data vs. Open Data

Not all linked data is open and not all open data is linked!

(★) Available on the web (whatever format) but with an open license, to be Open Data

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

(★★★) (2) in a non proprietary format (e.g., CSV instead of excel)

(★★★★) (3) plus using open standards from W3C (RDF and SPARQL) to identify things through dereferenceable HTTP URIs, to ensure effective access

(★★★★★) all the above plus establishing links between data of different sources

File format	Recommendations (on a scale of 0-5)
csv	★★★
xls	★
pdf	★
doc	★
xml	★★★★
rdf	★★★★★
shp	★★★
ods	★★
tiff	★
jpeg	★
json	★★★★
txt	★
html	★★

How to Publish Linked Data on the Web

A 15 steps recipe.

The steps form the basis for different workflows that can be used to publish Linked Data, depending on purpose, data and context.

Data of interest:

- ▶ knowledge organization systems (classification schemes, thesauri)
- ▶ authority files
- ▶ digital contents and their descriptions
- ▶ catalogues
- ▶ catalogue data including circulation data sets

All these datasets should have links within themselves and should establish outgoing links to many other web resources, in order to attract many incoming links.

Preparatory steps

- ▶ Motivation
- ▶ Management approval
- ▶ Sorting out the legal and financial issues
- ▶ Assessment of skills & data available
- ▶ Tools assessment and evaluation
- ▶ Dataset analysis

Ontological steps

- ▶ URI assignment
- ▶ Vocabulary Modeling

Implementation steps

- ▶ Generation of RDF Data
- ▶ Enriching the data

Meta-level steps

- ▶ Describing the data set

Publication and Curation cycle

- ▶ Evaluating the Dataset
- ▶ Publishing
- ▶ Incoming links
- ▶ Curation

URI assignment

Each resource in the dataset has to be identified by a unique URI, created according to the following guidelines:

- ▶ Use HTTP URIs so that they are dereferenceable.
- ▶ Ensure that the URIs are from a namespace that you control.
- ▶ Make sure your URIs do not carry implementation details which can change over time.
- ▶ It is advisable to use meaningful natural keys in URIs as unique identifiers of resources
 - ▶ for example, books can be identified by using the ISBN number instead of primary keys in the local database.

One resource in a dataset usually leads to the creation of at least three URIs:

- ▶ the one that represents the real world object
- ▶ the one that represents its HTML representation
- ▶ the one that represents its RDF/XML representation.

One way:

- ▶ http://dbpedia.org/resource/New_York_City
- ▶ http://dbpedia.org/page/New_York_City
- ▶ http://dbpedia.org/data/New_York_City

Another way:

- ▶ <http://mydomain.com/thing>
- ▶ <http://mydomain.com/thing.html>
- ▶ <http://mydomain.com/thing.rdf>

Vocabulary Modeling

Vocabulary modelling has to do with creating controlled terminology giving explicit meaning to the concepts in your dataset, and this is a key process in linked data.

If the original data is in some complex form such as relational tables, the semantics of the tables and the attributes has to be understood and encoded in RDFS or in OWL.

The emphasis is on the use of already existing vocabularies for the sake of communication efficacy (interoperability).

A vocabulary of choice should:

- ▶ be widely used to ensure widespread use of your dataset
- ▶ be actively maintained according to a clear governance process
- ▶ cover enough of your dataset to justify its terms
- ▶ be expressive enough to suit your particular requirements.

Creating your own vocabulary should be the very last option in vocabulary modelling.

Whenever you create a new vocabulary in the linked data world you have created a data island and have decreased the level of understanding in that domain.

Unless you are a very well known authority in that domain, it is likely that your vocabulary will not be used by someone else.

Creating a vocabulary is a very complex process which needs linguistic skills and domain expertise.

In the event that you create your own vocabularies, consider relating your new vocabulary to known vocabularies, making your concepts (classes and properties) sub-concepts of those in known vocabularies.

Enriching the data

This process involves adding to your dataset new triples that define relationships internally within the dataset (internal links) or relationships with outside resources (outgoing links).

- ▶ For internal links it must be ensured that every part of the dataset is reachable by a crawler when it is following links and therefore each file has to be connected to related files in the same dataset.
- ▶ For outgoing links, it is advisable to start by linking to such datasets as dbpedia, Geonames, Europeana, VIAF and others, which are already well established and stable in the linked data world.

This ensures that your dataset is easily discoverable since these are widely linked to by many other datasets.

Describing the data set

Before publishing, there is the need to provide a description of the dataset, which includes:

- ▶ provenance metadata—the history of that dataset, how it was generated and the technical processes that have been undergone to establish the dataset
- ▶ license and waiver metadata—how that dataset maybe used by third parties.
- ▶ the topic of a data set
- ▶ URI of the dataset
- ▶ location of SPARQL end-point
- ▶ data dumps
- ▶ last-modified date of the dataset
- ▶ change frequency

These data may be described using the Vocabulary of Interlinked datasets (voID), which is an RDF vocabulary.

Publishing the data set

The most obvious way to publish Linked Data on the Web is to make the URIs that identify data items dereferenceable into RDF descriptions.

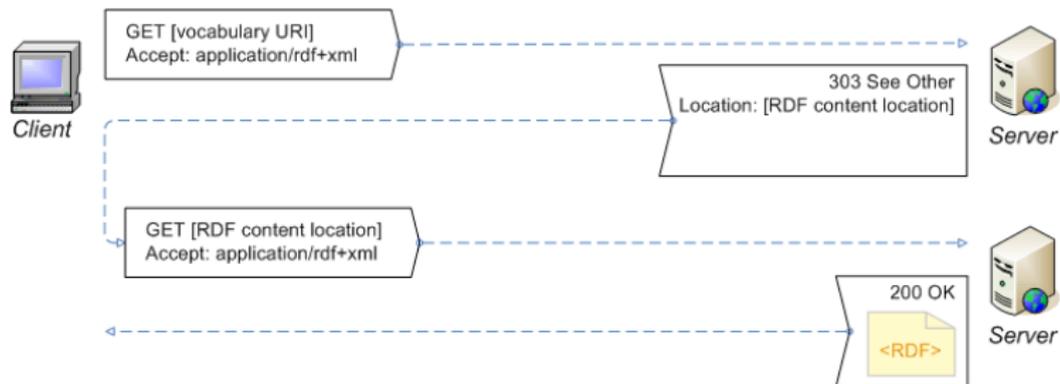
In addition, various Linked Open Data providers, including libraries, provide two alternative means of accessing the data:

- ▶ via SPARQL endpoints
- ▶ by providing RDF dumps of the complete data set

In general the system should provide access to both the RDF and HTML representations of the data.

This is usually done by configuring 303 redirects in triple stores in response to a client request to access either the HTML representation of an object or its RDF representation.

Architecture



Compare with the web mechanism for accessing informal knowledge

Incoming links

Incoming links originate from other datasets, linking into your dataset.

Third parties need to be convinced that your dataset is valuable to them so that they can link to it. However it is usually difficult for them to know your value unless you do some sort of marketing and promotion actions.

As a starting point a publisher can create triples that link to their own dataset and ask third parties like dbpedia to add those triples to their own dataset.

To continue attracting new links, there might be the need to employ marketing techniques so that the dataset is known by new users and be linked to.

Publishing Linked Data is an expensive process but it pays back in terms of making one's own data “linked” into the global database.

Sharing is the way to reduce costs:

- ▶ URIs, for increasing the linkedness
- ▶ vocabularies, for
 - ▶ increasing interoperability
 - ▶ cutting costs of vocabulary development and maintenance