An introduction to the semantic web technologies

And their use within the **@Web** platform

Leandro Lovisolo

INRA SupAgro and INRIA GraphiK
Montpellier, France

September 23, 2015

Outline of the presentation

- What's an ontology?
- RDF
- RDFS
- OWL
- SKOS
- SPARQL
- The n-ary relationship pattern used in **@Web**
- Examples of tables in scientific documents annotated using n-ary relationships in **@Web**

What's an ontology?

What's an ontology?

It's a formal description of a domain of interest based on:

What's an ontology?

It's a formal description of a domain of interest based on:

- a set of *individuals* (also called entities or objects),

What's an ontology?

It's a formal description of a domain of interest based on:

- a set of *individuals* (also called entities or objects),
- a set of *classes* of individuals, and

What's an ontology?

It's a formal description of a domain of interest based on:

- a set of *individuals* (also called entities or objects),
- a set of *classes* of individuals, and
- a set of *relationships* (sometimes called properties) between these individuals;

What's an ontology?

It's a formal description of a domain of interest based on:

- a set of *individuals* (also called entities or objects),
- a set of *classes* of individuals, and
- a set of *relationships* (sometimes called properties) between these individuals;

and a set of logical constraints to specify, among other things:

What's an ontology?

It's a formal description of a domain of interest based on:

- a set of *individuals* (also called entities or objects),
- a set of *classes* of individuals, and
- a set of *relationships* (sometimes called properties) between these individuals;

and a set of logical constraints to specify, among other things:

- class membership,

What's an ontology?

It's a formal description of a domain of interest based on:

- a set of *individuals* (also called entities or objects),
- a set of *classes* of individuals, and
- a set of *relationships* (sometimes called properties) between these individuals;

and a set of logical constraints to specify, among other things:

- class membership,
- subclass/subproperty relationships,

What's an ontology?

It's a formal description of a domain of interest based on:

- a set of *individuals* (also called entities or objects),
- a set of *classes* of individuals, and
- a set of *relationships* (sometimes called properties) between these individuals;

and a set of logical constraints to specify, among other things:

- class membership,
- subclass/subproperty relationships,
- domain/range restrictions on properties,

What's an ontology?

It's a formal description of a domain of interest based on:

- a set of *individuals* (also called entities or objects),
- a set of *classes* of individuals, and
- a set of *relationships* (sometimes called properties) between these individuals;

and a set of logical constraints to specify, among other things:

- class membership,
- subclass/subproperty relationships,
- domain/range restrictions on properties,
- cardinality constraints,

What's an ontology?

It's a formal description of a domain of interest based on:

- a set of *individuals* (also called entities or objects),
- a set of *classes* of individuals, and
- a set of *relationships* (sometimes called properties) between these individuals;

and a set of logical constraints to specify, among other things:

- class membership,
- subclass/subproperty relationships,
- domain/range restrictions on properties,
- cardinality constraints,
- class union/intersection/disjointness constraints,

What's an ontology?

It's a formal description of a domain of interest based on:

- a set of *individuals* (also called entities or objects),
- a set of *classes* of individuals, and
- a set of *relationships* (sometimes called properties) between these individuals;

and a set of logical constraints to specify, among other things:

- class membership,
- subclass/subproperty relationships,
- domain/range restrictions on properties,
- cardinality constraints,
- class union/intersection/disjointness constraints,
- etc.

Web resources, URI, namespaces

Web resources, URI, namespaces

A *resource* is anything that can be referred to: a web page, a person, a city, a university course, etc.

Web resources, URI, namespaces

A *resource* is anything that can be referred to: a web page, a person, a city, a university course, etc.

Resources are identified by *URIs*, for example:

- `http://example.com/MyOntology`,
- `http://example.com/MyOntology#Leandro`,
- `http://example.com/MyOntology#Pizza`,
- etc.

Web resources, URI, namespaces

A *resource* is anything that can be referred to: a web page, a person, a city, a university course, etc.

Resources are identified by *URIs*, for example:

- `http://example.com/MyOntology`,
- `http://example.com/MyOntology#Leandro`,
- `http://example.com/MyOntology#Pizza`,
- etc.

To avoid carrying long URIs, *namespaces* are used.

Web resources, URI, namespaces

A *resource* is anything that can be referred to: a web page, a person, a city, a university course, etc.

Resources are identified by *URIs*, for example:

- `http://example.com/MyOntology`,
- `http://example.com/MyOntology#Leandro`,
- `http://example.com/MyOntology#Pizza`,
- etc.

To avoid carrying long URIs, *namespaces* are used. Thus,

Web resources, URI, namespaces

A *resource* is anything that can be referred to: a web page, a person, a city, a university course, etc.

Resources are identified by *URIs*, for example:

- `http://example.com/MyOntology`,
- `http://example.com/MyOntology#Leandro`,
- `http://example.com/MyOntology#Pizza`,
- etc.

To avoid carrying long URIs, *namespaces* are used. Thus,

- `http://example.com/MyOntology`

Web resources, URI, namespaces

A *resource* is anything that can be referred to: a web page, a person, a city, a university course, etc.

Resources are identified by *URIs*, for example:

- `http://example.com/MyOntology`,
- `http://example.com/MyOntology#Leandro`,
- `http://example.com/MyOntology#Pizza`,
- etc.

To avoid carrying long URIs, *namespaces* are used. Thus,

- `http://example.com/MyOntology` becomes

Web resources, URI, namespaces

A *resource* is anything that can be referred to: a web page, a person, a city, a university course, etc.

Resources are identified by *URIs*, for example:

- `http://example.com/MyOntology`,
- `http://example.com/MyOntology#Leandro`,
- `http://example.com/MyOntology#Pizza`,
- etc.

To avoid carrying long URIs, *namespaces* are used. Thus,

- `http://example.com/MyOntology` becomes
- `example:MyOntology`

Web resources, URI, namespaces

A *resource* is anything that can be referred to: a web page, a person, a city, a university course, etc.

Resources are identified by *URIs*, for example:

- `http://example.com/MyOntology`,
- `http://example.com/MyOntology#Leandro`,
- `http://example.com/MyOntology#Pizza`,
- etc.

To avoid carrying long URIs, *namespaces* are used. Thus,

- `http://example.com/MyOntology` becomes
- `example:MyOntology` abbreviated as

Web resources, URI, namespaces

A *resource* is anything that can be referred to: a web page, a person, a city, a university course, etc.

Resources are identified by *URIs*, for example:

- `http://example.com/MyOntology`,
- `http://example.com/MyOntology#Leandro`,
- `http://example.com/MyOntology#Pizza`,
- etc.

To avoid carrying long URIs, *namespaces* are used. Thus,

- `http://example.com/MyOntology` becomes
- `example:MyOntology` abbreviated as
- `:MyOntology`

Web resources, URI, namespaces

A *resource* is anything that can be referred to: a web page, a person, a city, a university course, etc.

Resources are identified by *URIs*, for example:

- `http://example.com/MyOntology`,
- `http://example.com/MyOntology#Leandro`,
- `http://example.com/MyOntology#Pizza`,
- etc.

To avoid carrying long URIs, *namespaces* are used. Thus,

- `http://example.com/MyOntology` becomes
- `example:MyOntology` abbreviated as
- `:MyOntology`

if `example` is the default namespace.

RDF

A simple language for describing *annotations* about Web resources identified by URIs, from now on referred to as **facts**.

RDF
Triplets

Facts are stated as *RDF triplets*.

Facts are stated as *RDF triplets*.

A triplet is made of a *subject*, an *object* and a *predicate*.

RDF

Triplets

Facts are stated as *RDF triplets*.

A triplet is made of a *subject*, an *object* and a *predicate*.

Some examples:

RDF
Triplets

Facts are stated as *RDF triplets*.

A triplet is made of a *subject*, an *object* and a *predicate*.

Some examples:

- ⟨`:Dupond :Leads :InfoDept`⟩

RDF

Triplets

Facts are stated as *RDF triplets*.

A triplet is made of a *subject*, an *object* and a *predicate*.

Some examples:

- $\langle$`:Dupond :Leads :InfoDept`$\rangle$
- $\langle$`:Dupond :TeachesIn :UE111`$\rangle$

RDF
Triplets

Facts are stated as *RDF triplets*.

A triplet is made of a *subject*, an *object* and a *predicate*.

Some examples:

- ⟨`:Dupond :Leads :InfoDept`⟩
- ⟨`:Dupond :TeachesIn :UE111`⟩
- ⟨`:Dupond :TeachesTo :Pierre`⟩

RDF

Triplets

Facts are stated as *RDF triplets*.

A triplet is made of a *subject*, an *object* and a *predicate*.

Some examples:

- ⟨`:Dupond :Leads :InfoDept`⟩
- ⟨`:Dupond :TeachesIn :UE111`⟩
- ⟨`:Dupond :TeachesTo :Pierre`⟩
- ⟨`:Pierre :EnrolledIn :InfoDept`⟩

RDF

Triplets

Facts are stated as *RDF triplets*.

A triplet is made of a *subject*, an *object* and a *predicate*.

Some examples:

- $\langle$`:Dupond :Leads :InfoDept`$\rangle$
- $\langle$`:Dupond :TeachesIn :UE111`$\rangle$
- $\langle$`:Dupond :TeachesTo :Pierre`$\rangle$
- $\langle$`:Pierre :EnrolledIn :InfoDept`$\rangle$
- $\langle$`:Pierre :RegisteredTo :UE111`$\rangle$

RDF

Triplets

Facts are stated as *RDF triplets*.

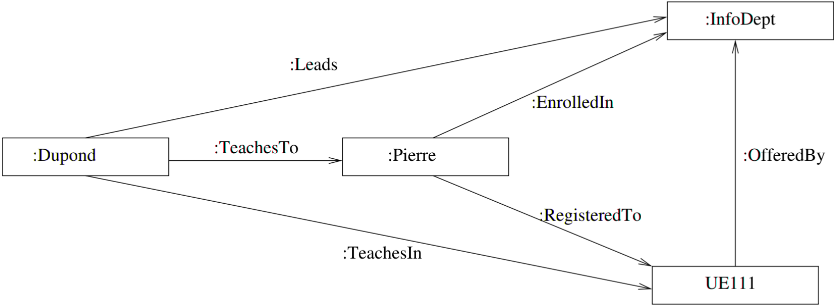
A triplet is made of a *subject*, an *object* and a *predicate*.

Some examples:

- $\langle$`:Dupond :Leads :InfoDept`$\rangle$
- $\langle$`:Dupond :TeachesIn :UE111`$\rangle$
- $\langle$`:Dupond :TeachesTo :Pierre`$\rangle$
- $\langle$`:Pierre :EnrolledIn :InfoDept`$\rangle$
- $\langle$`:Pierre :RegisteredTo :UE111`$\rangle$
- $\langle$`:UE111 :OfferedBy :InfoDept`$\rangle$

RDF

Graph representation



⟨:Dupond :Leads :InfoDept⟩
⟨:Dupond :TeachesIn :UE111⟩
⟨:Dupond :TeachesTo :Pierre⟩
⟨:Pierre :EnrolledIn :InfoDept⟩
⟨:Pierre :RegisteredTo :UE111⟩
⟨:UE110 :OfferedBy :InfoDept⟩

There are many different syntaxes for writing RDF triplets, including:

RDF

Syntax

There are many different syntaxes for writing RDF triplets, including:

- XML (as used in **@Web**),

There are many different syntaxes for writing RDF triplets, including:

- XML (as used in **@Web**),
- Turtle,
- N-Triples,
- N-Quads,
- etc.

RDF

Syntax

There are many different syntaxes for writing RDF triplets, including:

- XML (as used in **@Web**),
- Turtle,
- N-Triples,
- N-Quads,
- etc.

However, we're going to focus on the abstract ⟨`subject, predicate, object`⟩ syntax during this presentation.

Motivation

Problem statement

Motivation

Problem statement

- We're trying to answer questions that require consulting heterogeneous data sources.

Motivation

Problem statement

- We're trying to answer questions that require consulting heterogeneous data sources.
  - Literature with inconsistent, semi-structured data.

Motivation

Problem statement

- We're trying to answer questions that require consulting heterogeneous data sources.
  - Literature with inconsistent, semi-structured data.
  - No standard naming convention.

Motivation

Problem statement

- We're trying to answer questions that require consulting heterogeneous data sources.
  - Literature with inconsistent, semi-structured data.
  - No standard naming convention.
  - No information about the reliability of the data sources.

Motivation

Problem statement

- We're trying to answer questions that require consulting heterogeneous data sources.
  - Literature with inconsistent, semi-structured data.
  - No standard naming convention.
  - No information about the reliability of the data sources.
  - Each data source has its specific browsing/querying mechanism (no common interface.)

Motivation

Sample problem domain: **biorefinery**

- Ligno-cellulosic biomass pre-treatment before enzymatic hydrolysis is an essential step to obtain good yields.

Motivation

Sample problem domain: **biorefinery**

- Ligno-cellulosic biomass pre-treatment before enzymatic hydrolysis is an essential step to obtain good yields.
- Several pre-treatment principles available, but **no clear criteria on how to choose the best one** taking into account environmental sustainability for a given biomass and biorefinery product (e.g. glucose.)

Proposed solution

- Represent scientific knowledge with ontologies using recommended standardized tools and languages for such purposes (semantic web technologies, RDF(S), OWL, etc.)

Proposed solution

- Represent scientific knowledge with ontologies using recommended standardized tools and languages for such purposes (semantic web technologies, RDF(S), OWL, etc.)
- Develop an ontology and data management web application (e.g. the **@Web platform**) that makes it easy for scientists to introduce data from scientific publications into an ontology, execute queries against an ontology, etc.

Proposed solution

- Represent scientific knowledge with ontologies using recommended standardized tools and languages for such purposes (semantic web technologies, RDF(S), OWL, etc.)
- Develop an ontology and data management web application (e.g. the **@Web platform**) that makes it easy for scientists to introduce data from scientific publications into an ontology, execute queries against an ontology, etc.
- Create integrity constraints to automatically detect inconsistencies and errors in scientific publications and to automatically classify publications according to their topics.

Proposed solution

- Represent scientific knowledge with ontologies using recommended standardized tools and languages for such purposes (semantic web technologies, RDF(S), OWL, etc.)
- Develop an ontology and data management web application (e.g. the **@Web platform**) that makes it easy for scientists to introduce data from scientific publications into an ontology, execute queries against an ontology, etc.
- Create integrity constraints to automatically detect inconsistencies and errors in scientific publications and to automatically classify publications according to their topics.
  - *The focus of my internship!*

An example of a termino-ontological resource

Taken from the biorefinery application

Design goals for the core ontology

- **Simple** so as to make the annotator's task easier.

Design goals for the core ontology

- **Simple** so as to make the annotator's task easier.
- **Generic** enough so that the approach can be applied to different, unrelated domains.

Design goals for the core ontology

- **Simple** so as to make the annotator's task easier.
- **Generic** enough so that the approach can be applied to different, unrelated domains.
  - Proven in the domains of biorefinery and packaging selection.

A sample relation

Also from the biorefinery domain

The **@Web** platform

Exploring an ontology

The **@Web** platform

Browsing documents

The **@Web** platform

Querying an ontology: defining the search scope

The **@Web** platform

Querying an ontology: search parameters

The **@Web** platform

Querying an ontology: executing a query

The **@Web** platform

Querying an ontology: results

The annotator's task

- Given a scientific publication and a desired ontology, capture data from the publication using the appropriate concepts in the ontology.

The annotator's task

- Given a scientific publication and a desired ontology, capture data from the publication using the appropriate concepts in the ontology.
- Create and update concepts in the ontology as they're discovered during the annotation process (i.e. in an iterative fashion.)

The annotator's task

- Given a scientific publication and a desired ontology, capture data from the publication using the appropriate concepts in the ontology.
- Create and update concepts in the ontology as they're discovered during the annotation process (i.e. in an iterative fashion.)
- Write and edit **guidelines** associated to each concept explaining when and how a concept should be used.

An example of data captured from a scientific publication

A sample guideline

Some sample guidelines that can be easily translated into SPARQL constraints

Integrity constraints

- *"The output quantity of a step is equal to the sum of the quantity of water used and the quantity of biomass present in the step."*

Some sample guidelines that can be easily translated into SPARQL constraints

Integrity constraints

- *"The output quantity of a step is equal to the sum of the quantity of water used and the quantity of biomass present in the step."*
- *"The second milling step must give an "Output solid constituent size" smaller than 0,5-1 mm."*

Some sample guidelines that can be easily translated into SPARQL constraints

Classification constraints

- *"Topic Bioref-PM-PC-UFM-PS : included experiments are composed of a pre-milling step, followed by a physico-chemical treatment, then by an ultrafine milling step (ball milling, wet disk milling, etc.), a press and separation step (washing and filtration), and finally the enzymatic hydrolysis step. This topic requires a press and separation step because there are a lot of effluents in the physico-chemical step or because the milling is made with effluent. The second milling step must give an "Output solid constituent size" smaller than 0,5-1 mm. (en)"*

Examples of guidelines that **cannot** be easily translated into SPARQL constraints

- *"In all treatments, when the authors indicate "overnight", we considered a duration treatment between 10 and 15 hours"*

Examples of guidelines that **cannot** be easily translated into SPARQL constraints

- *"In all treatments, when the authors indicate "overnight", we considered a duration treatment between 10 and 15 hours"*
- *"Furthermore, we consider that the glucose rate equals to glucan rate divided by 0.9."*

Statistics

A promising approach

In the biorefinery ontology alone we have:

- 11 occurrences of the phrase *"equal to"*
- 5 occurrences of the phrase *"equals to"*
- 11 occurrences of the phrase *"sum of"*
- 3 occurrences of the phrase *"divided by"*
- 2 occurrences of the phrase *"multiplied by"*

spread across guidelines associated with 30 relation concepts.

At least 10 of them can be easily translated into SPARQL constraints.

Thanks!