

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*.

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:

- ▶ `<:Dupond :Leads :InfoDept>`
- ▶ `<:Dupond :TeachesIn :UE111>`

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:

- ▶ `<:Dupond :Leads :InfoDept>`
- ▶ `<:Dupond :TeachesIn :UE111>`
- ▶ `<:Dupond :TeachesTo :Pierre>`
- ▶ `<:Pierre :EnrolledIn :InfoDept>`
- ▶ `<:Pierre :RegisteredTo :UE111>`

RDF

Triplets

Facts are stated as *RDF triplets*.

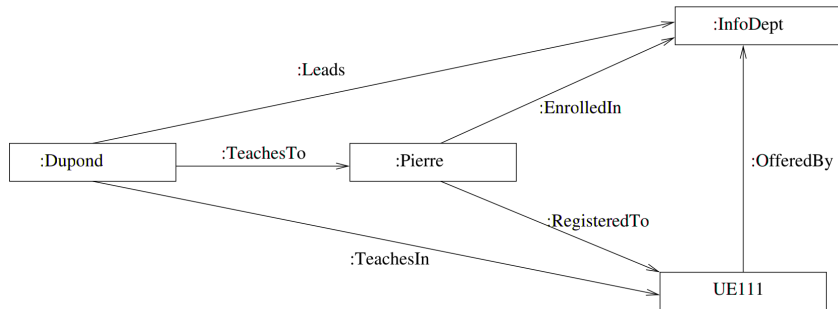
A triplet is made of a *subject*, an *object* and a *predicate*.

Some examples:

- ▶ `<:Dupond :Leads :InfoDept>`
- ▶ `<:Dupond :TeachesIn :UE111>`
- ▶ `<:Dupond :TeachesTo :Pierre>`
- ▶ `<:Pierre :EnrolledIn :InfoDept>`
- ▶ `<:Pierre :RegisteredTo :UE111>`
- ▶ `<:UE111 :OfferedBy :InfoDept>`

RDF

Graph representation



```
<:Dupond :Leads :InfoDept>  
<:Dupond :TeachesIn :UE111>  
<:Dupond :TeachesTo :Pierre>  
<:Pierre :EnrolledIn :InfoDept>  
<:Pierre :RegisteredTo :UE111>  
<:UE111 :OfferedBy :InfoDept>
```

RDF

Syntax

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**),

RDF

Syntax

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.

Thanks!