



Multilingual Linked Open Data Patterns

Jose Emilio Labra Gayo

University of Oviedo, Spain

Joint work with:

Dimitris Kontokostas

Sören Auer

Universität Leipzig, Germany

More info: <http://www.weso.es/MLODPatterns>

MLOD Patterns

From *best practices* (Dublin) to *patterns* (Rome)

We propose a catalog of 20 MLOD patterns

Pattern = generic solution to a problem in a context

Common vocabulary

Patterns can be related to each other

Some patterns can contradict other patterns

There are already Linked data patterns

We focus on **multilingual** linked data patterns

Each pattern contains

Name

Description

Context

Example

Discussion

See also

Based on DBPedia I18n experience

More info: <http://www.weso.es/MLODPatterns>

MLOD Patterns

Patterns are classified by activity:

Naming:

URI design, URIs, IRIs, etc.

Dereference:

How is the content that we return affected by multilingualism

Labeling:

Handling multilingual labels

Longer descriptions:

Longer textual descriptions

Linking

Links between concepts in different languages?

Reuse

Vocabularies and multilingualism

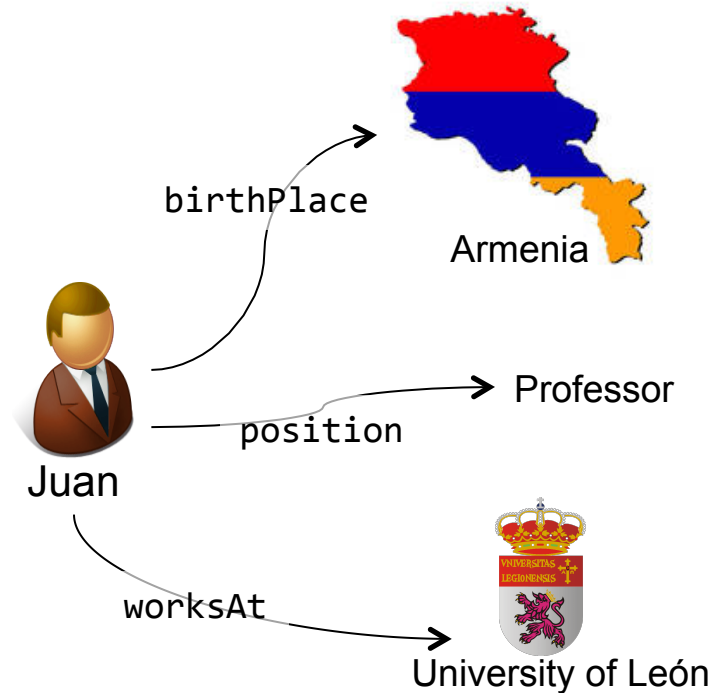
More info: <http://www.weso.es/MLODPatterns>

General overview

Goal	Pattern	Description
Naming	Descriptive URIs	Use descriptive URIs with ASCII characters, % encoding extended characters
	Opaque URIs	Use non human-readable URIs
	Full IRIs	Use IRIs with unicode characters
	Internationalized local names	Use Unicode characters only for local names
	Language in URIs	Include language information in the URI
Dereference	Return Language independent data	Return the same triples independently of the language
	Language content negotiation	Return different triples depending on user agent preferences
Labeling	Label everything	Define labels for all the resources
	Multilingual labels	Add language tags to labels
	Labels without language tag	Add labels without language tags in a default language
Longer descriptions	Divide longer descriptions	Replace long descriptions by more resources with labels
	Add lexical information	Add lexical information to long descriptions
	Structured literals	Use HTML/XML literals for longer descriptions
Linking	Identity links	Use owl:sameAs and similar predicates
	Soft links	Use predicates with soft semantics
	Linguistic metadata	Add linguistic metadata about the dataset terms
Reuse	Monolingual vocabularies	Attach labels to vocabularies in a single language
	Multilingual vocabularies	Prefer multilingual vocabularies
	Localize existing vocabularies	Translate labels of existing vocabularies
	Create new localized vocabularies	Create custom vocabularies and link to existing ones

Motivating example

Juan is an armenian professor at the University of León.



More info: <http://www.weso.es/MLODPatterns>

Naming

Selecting a URI scheme for Armenia

Descriptive URIs

Human-readable
Good tool support

<http://example.org/Armenia>

Only ASCII, %-encode non-ASCII characters

<http://example.org/Universidad%20de%Le%C3%B3n>

More info: <http://www.weso.es/MLODPatterns>

Naming

Selecting a URI scheme for Armenia

Descriptive URIs

<http://example.org/Armenia>

Opaque URIs

<http://example.org/I23AX45>

Independence between concept and natural language representation

Non Human-readable
Difficult to handle by developers

More info: <http://www.weso.es/MLODPatterns>

Naming

Selecting a URI scheme for Armenia

Descriptive URIs

<http://example.org/Armenia>

Opaque URIs

<http://example.org/I23AX45>

Full IRIs

<http://օրինակ.օրգ#Հայաստան>

More natural for non-Latin based languages

Subject to visual spoofing attacks
Not so good tool support

More info: <http://www.weso.es/MLODPatterns>

Naming

Selecting a URI scheme for Armenia

Descriptive URIs

<http://example.org/Armenia>

Opaque URIs

<http://example.org/I23AX45>

Full IRIs

<http://օրինակ.օրգ#Հայաստան>

Internationalized local names

<http://example.org#Հայաստան>

Avoids domain name spoofing
More human-friendly identifiers

Visual spoofing attacks can still possible

More info: <http://www.weso.es/MLODPatterns>

Naming

Selecting a URI scheme for Armenia

Descriptive URIs

<http://example.org/Armenia>

Opaque URIs

<http://example.org/I23AX45>

Full IRIs

<http://օրինակ.օրգ#Հայաստան>

Internationalized local names

<http://example.org#Հայաստան>

Include language in URIs

<http://hy.example.org#Հայաստան>

<http://en.example.org#Armenia>

Independent development of datasets
by language

Adding language info to the URI can become unwieldy
Example: languages & sublanguages

hy-Latin-IT-arevela

Where should we put the language tag?

More info: <http://>



Dereference

Language based content negotiation?

No language content negotiation

<http://example.org/Armenia>



Always returns the same data

Easy to develop
Consistency of data

Clients have to filter triples in other languages
Computation & network overhead

More info: <http://www.weso.es/MLODPatterns>

Dereference

Language based content negotiation?

No language content negotiation

<http://example.org/Armenia>



Always returns the same data

Language content negotiation

<http://example.org/Armenia>

Returns different data depending on
Accept-language

Accept-language:en



```
:Armenia rdfs:label "Armenia"@en .
```

Improves clients performance
Less network overhead

Accept-language:hy



```
:Armenia rdfs:label "Հայաստան"@hy .
```

Difficult to implement
Semantic equivalence between data

More info: <http://www.weso.es/MLODPatterns>

Labeling

Label everything

```
:Armenia rdfs:label "Armenia" .  
:juan rdfs:label "Juan López" .  
:position rdfs:label "Job position" .
```

User agents can show labels instead of URIs
Labels can be used for searching

Not always feasible, which labels?
Labels are for humans
Avoid machine-oriented notations

More info: <http://www.weso.es/MLODPatterns>

Labeling

Label everything

```
:Armenia rdfs:label "Armenia" .  
:juan rdfs:label "Juan López" .  
:position rdfs:label "Job position" .
```

Multilingual Labels

```
:juan :position "Professor"@en .  
:juan :position "Catedrático"@es .
```

Multilingual labels are part of RDF Model

SPARQL can be more difficult

```
SELECT * WHERE {  
  ?x :position "Professor" .  
}
```

More info: <http://www.weso.es/MLODPatterns>

Labeling

Label everything

```
:Armenia rdfs:label "Armenia" .  
:juan rdfs:label "Juan López" .  
:position rdfs:label "Job position" .
```

Multilingual Labels

```
:juan :position "Professor"@en .  
:juan :position "Catedrático"@es .
```

Labels without language tag

```
:juan :position "Professor"@en .  
:juan :position "Catedrático"@es .  
:juan :position "Professor" .
```

SPARQLing is easier

Choosing a default language is controversial

More info: <http://www.weso.es/MLODPatterns>

Longer descriptions

Divide long descriptions

```
:juan :jobtitle  
      "Professor at the University of León"@en .
```



```
:juan :position :professor .  
:juan :workPlace :uniLeón .  
  
:professor rdfs:label "Professor"@en .  
:uniLeón rdfs:label "University of León"@en .
```



Fine-grained data is more amenable to semantic web apps
Apps can generate more readable information

More complexity of the model
Not always possible

More info: <http://www.weso.es/MLODPatterns>

Longer descriptions

Provide lexical information

More value to dataset
Automatic manipulation
Can improve message generation

Complexity overhead
Feasibility

```
:uniLeón a lemon:LexicalEntry ;
  lemon:decomposition (
    [ lemon:element :University ]
    [ lemon:element :Of ]
    [ lemon:element :León ]
  );
rdfs:label "University of León"@en .

:University a lemon:LexicalEntry ;
  lexinfo:partOfSpeech lexinfo:commonNoun ;
  rdfs:label "University"@en ;
  rdfs:label "Universidad"@es .

:Of a lemon:LexicalEntry ;
  lexinfo:partOfSpeech lexinfo:preposition ;
  rdfs:label "of"@en ;
  rdfs:label "de"@es .

:León a lemon:LexicalEntry ;
  lexinfo:partOfSpeech lexinfo:properNoun ;
  rdfs:label "León"
```

More info: <http://www.weso.es/MLOD>

Example provided by J. McCrae

Longer descriptions

Structured literals

```
:uniLeón :desc
  "<p>University of
    <span translate="no">León</span>,
    Spain.
  </p>"^^rdf:XMLLiteral .
```

Leverage existing I18N techniques
Bidi, Ruby, Localization notes, ...

Interaction between 2 abstraction models
RDF vs XML/HTML
Large portions of structured literals can hinder LD

More info: <http://www.weso.es/MLODPatterns>

Linking

Inter-language identity links

```
<http://hy.example.org#Հայաստան>  
  owl:sameAs  
  <http://en.example.org#Armenia> .
```

owl:sameAs is a well known property
Already supported by linked data applications

Too strong semantics of owl:sameAs
Concepts may not be the same
Contradictions

More info: <http://www.weso.es/MLODPatterns>

Linking

Inter-language identity links

```
<http://hy.example.org#Հայաստան>  
  owl:sameAs  
  <http://en.example.org#Armenia> .
```

Soft Inter-language links

```
<http://hy.example.org#Հայաստան>  
  rdfs:seeAlso  
  <http://en.example.org#Armenia> .
```

Several predicates

`rdfs:seeAlso`

`skos:related`

`dbo:wikiPageLanguageLink`

No standard property

No support for inference

More info: <http://www.weso.es/MLODPatterns>

Linking

Inter-language identity links

```
<http://hy.example.org#Հայաստան>  
  owl:sameAs  
  <http://en.example.org#Armenia> .
```

Soft Inter-language links

```
<http://hy.example.org#Հայաստան>  
  rdfs:seeAlso  
  <http://en.example.org#Armenia> .
```

Link linguistic meta-data

```
:Catedrático  
  lexvo:means wordnet:Professor ;  
  lexvo:language  
  <http://lexvo.org/id/iso639-3/spa> .
```

Links between multilingual labels
Can declare language of a dataset

No standard practice
Semantic equivalence between concepts

More info: <http://www.weso.es/MLODPatterns>

Reuse

Monolingual vocabularies

FOAF, Dublin Core, OWL, RDF Schema, ... are only in English

Easier to control vocabulary evolution
Avoid bad translations, ambiguities

Monolingual vocabularies in multilingual applications require a translation layer

More info: <http://www.weso.es/MLODPatterns>

Reuse

Monolingual vocabularies

FOAF, Dublin Core, OWL, RDF Schema, ... are only in English

Multilingual vocabularies

```
:position a owl:DatatypeProperty ;  
  rdfs:domain :UniversityStaff ;  
  rdfs:label "Position"@en ;  
  rdfs:label "Puesto"@es .  
  
:UniversityStaff a owl:Class ;  
  rdfs:label "University staff"@en ;  
  rdfs:label "Trabajador universitario"@es .
```

Elegant solution in multilingual contexts
More control over translations

Some common vocabularies are monolingual
Maintenance is more difficult

More info: <http://www.weso.es/MLODPatterns>

Reuse

Localize existing vocabularies

```
dc:contributor rdfs:label "Colaborador"@es .
```

Transparently select label in preferred language
Principle AAA
Anyone can say Anything about Any topic

Polluting well known vocabularies = controversial

More info: <http://www.weso.es/MLODPatterns>

Reuse

Localize existing vocabularies

```
dc:contributor rdfs:label "Colaborador"@es .
```

Create new localized vocabularies

```
dc:contributor  
  owl:equivalentProperty :colaborador .  
:colaborador  
  rdfs:label "Colaborador"@es .
```

Freedom to tailor vocabulary to specific needs

More difficult to humans/agents to recognize new properties/classes

More info: <http://www.weso.es/MLODPatterns>

Future work

The catalog is not closed

Other issues & patterns

Microdata & RDFa

Other I18n topics

Handle big datasets

Localization workflows

Feedback from the community

Best practices, Patterns, Anti-patterns?



More info: <http://www.weso.es/MLODPatterns>