

Combining semantic precision with biologist-friendly systems through RDF and Metarel

Ward Blondé, July 2012

RDF triples

- **RDF** consists of triples **subject-predicate-object** (**s p o**). 'Predicate' is synonym of 'relation' here...
- RDF is **not** a decidable **logic**, however, it does provide an intuitive semantics by interpreting predicates as **verbs** or verbal expressions (subject-verb-object or **SVO-triples**):

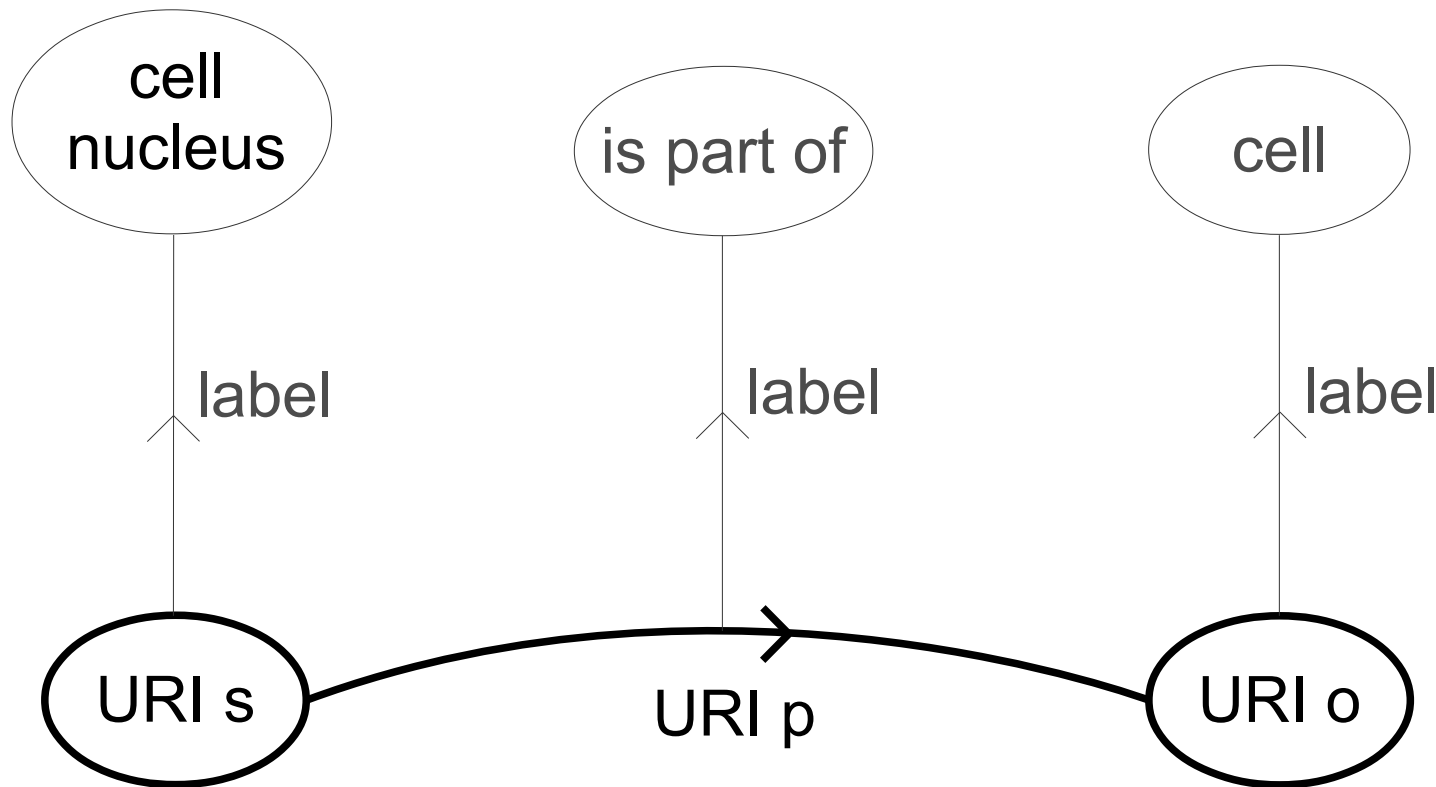
The triple (ISA123 isa biomolecule)
should mean something like:
ISA123 'is a' biomolecule...

SVO-triples

- **SPARQL** queries RDF and has therefore the potential to return **SVO-triples** like:
 - Ghent *is located in* Belgium
 - Biological cell *contains* water molecule
 - John *is a* human
 - John *loves* Mary
 - Stefan *is located in* train ABC
- SPARQL in combination with the SVO-intuition can be used for **browsing and visualizing RDF** and creating **integrated queries** over several triples

SVO-triples

- An **SVO-triple** can be represented in RDF through unique identifiers for the subject, predicate and object, and through **labels**:

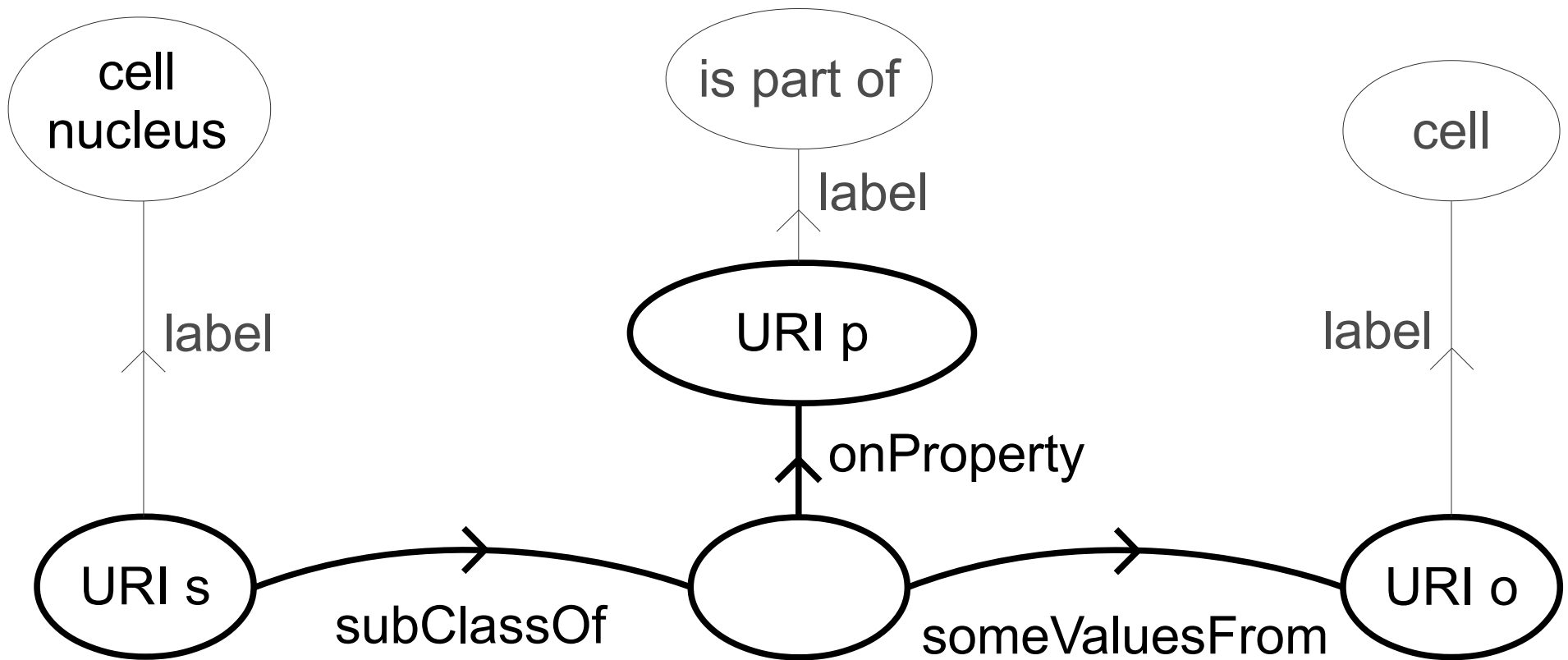


Expressive Knowledge Management

- **Knowledge Management** is concerned with more **expressive** statements like:
 - **Every** biological cell contains **some** water molecule **at** each moment.
 - Stefan is located in train ABC **between** 9 am and 10 am **on** 12/07/2012
 - Protein ABC4 has the **disposition** to interact with protein CDK2

Expressive statements in RDF

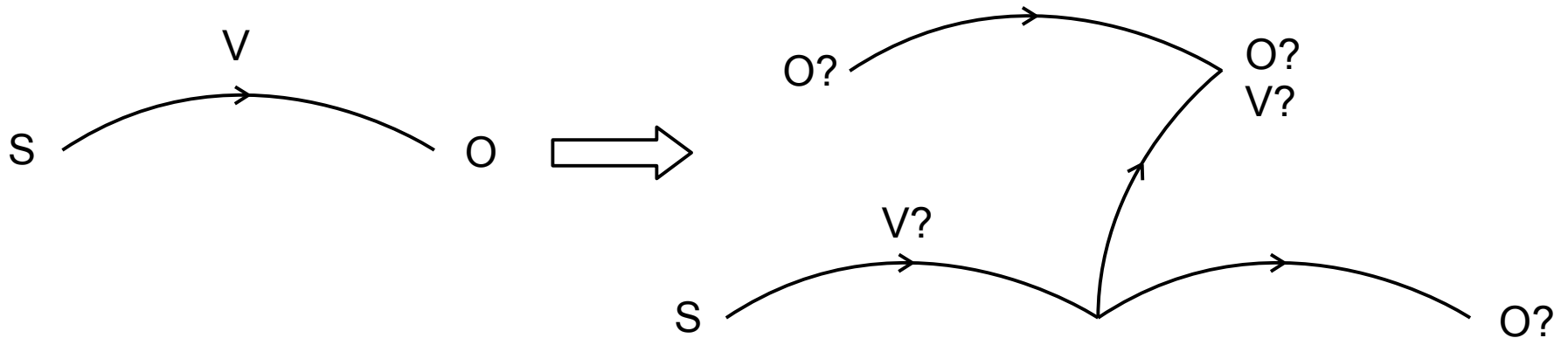
- Engineering advanced statements in RDF can be done by **breaking SVO-triples** in several triples:



“Every cell nucleus is part of some cell”

Problem statement

- **Browsing** via the SVO-intuition is **not longer possible** for this kind of expressive statements in RDF.
- Blank nodes are only allowed in subjects and objects for the current RDF standard. This forces the break-up of SVO-triples.



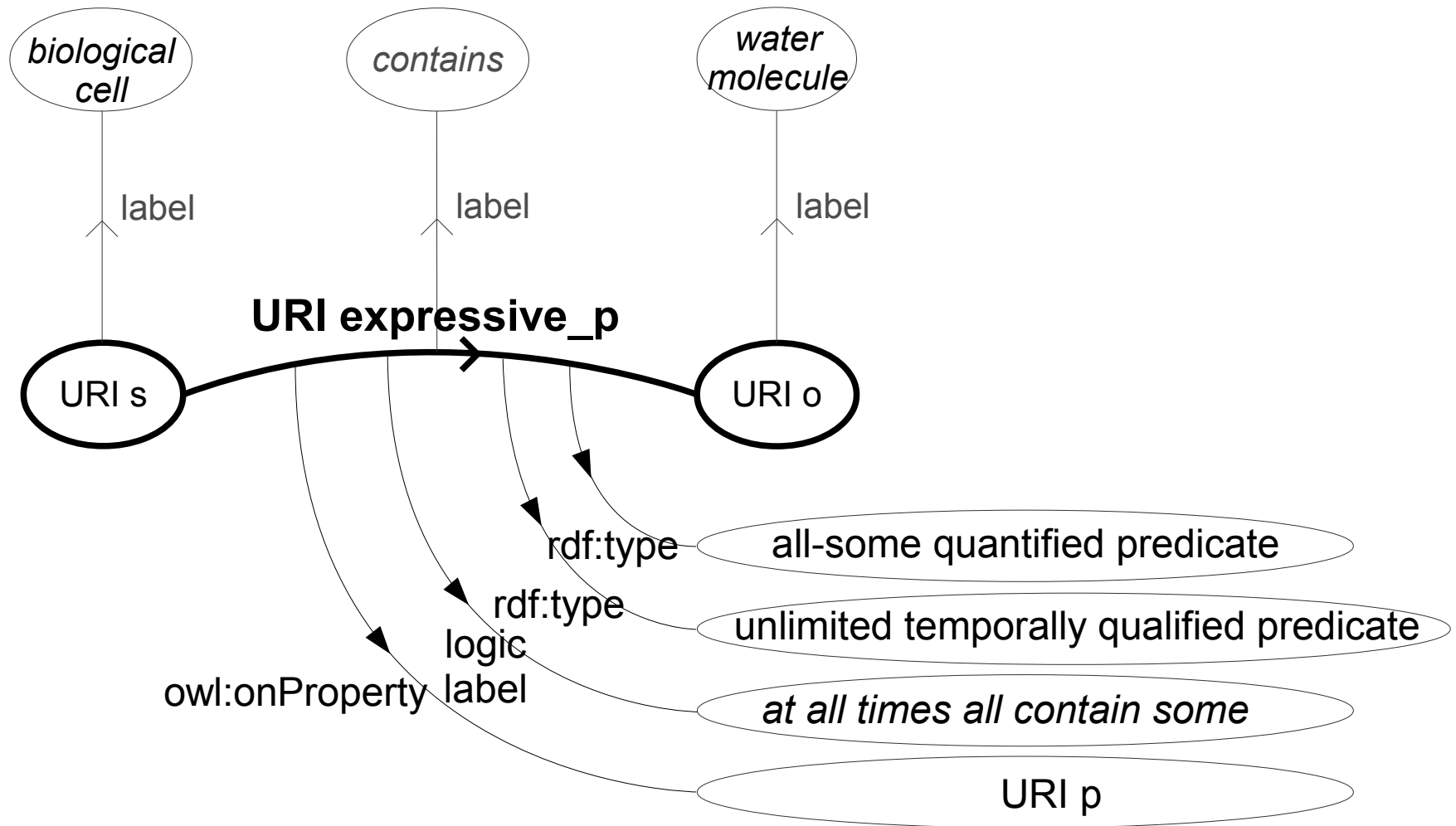
No generic way of finding a verb and an object.

Problem statement

- **Different ways** of breaking up SVO-triples are used:
 - Logical **quantification** (like for all, there exists)
 - **Reification** (for e.g. specifying when a triple is true, how certain is the triple)
 - **Lists** (for e.g. grouping triples as a set of sufficient logical conditions)
 - **N-ary** relations (as a binary relation between subject and object, with some external entities of importance)

Solution

- **Solution:** encode extra information in the **predicates**:



Solution

- **Solution:** encode extra information in **OBO-tags**:

[Term]

id: URI_s

name: biological cell

def: "A biological is a cell that is biological." [WB]

is_a: URI_cell ! cell

relationship: URI_p URI_o

has_quantification: **at-all-times-all-some**

URIs of expressive predicates remain **outside the OBO Format**, but can be maintained in a meta-ontology about predicates, like **Metarel**. This also provides a basis for translation to **OWL**.

Advantages

- A **library of basic SPARQL** queries, that exploit the **SVO-intuition**, can be engineered **for all statements**:
 - To which objects is this subject related?
 - Show ten SVO-triples.
 - Is there a relation between this subject and that object, and if so, which?
- More advanced SPARQL queries can take into account the **extra information** from the **classification** of the **predicates**

Advantages

- **Modules** can be extracted via **SPARQL/Update**:
 - Retain only **SVO**-triples for a browser-application
 - Retain only relations that hold **at all times**
 - Retain only relations compatible with **OWL EL**.
 - ...

SPARQL core example

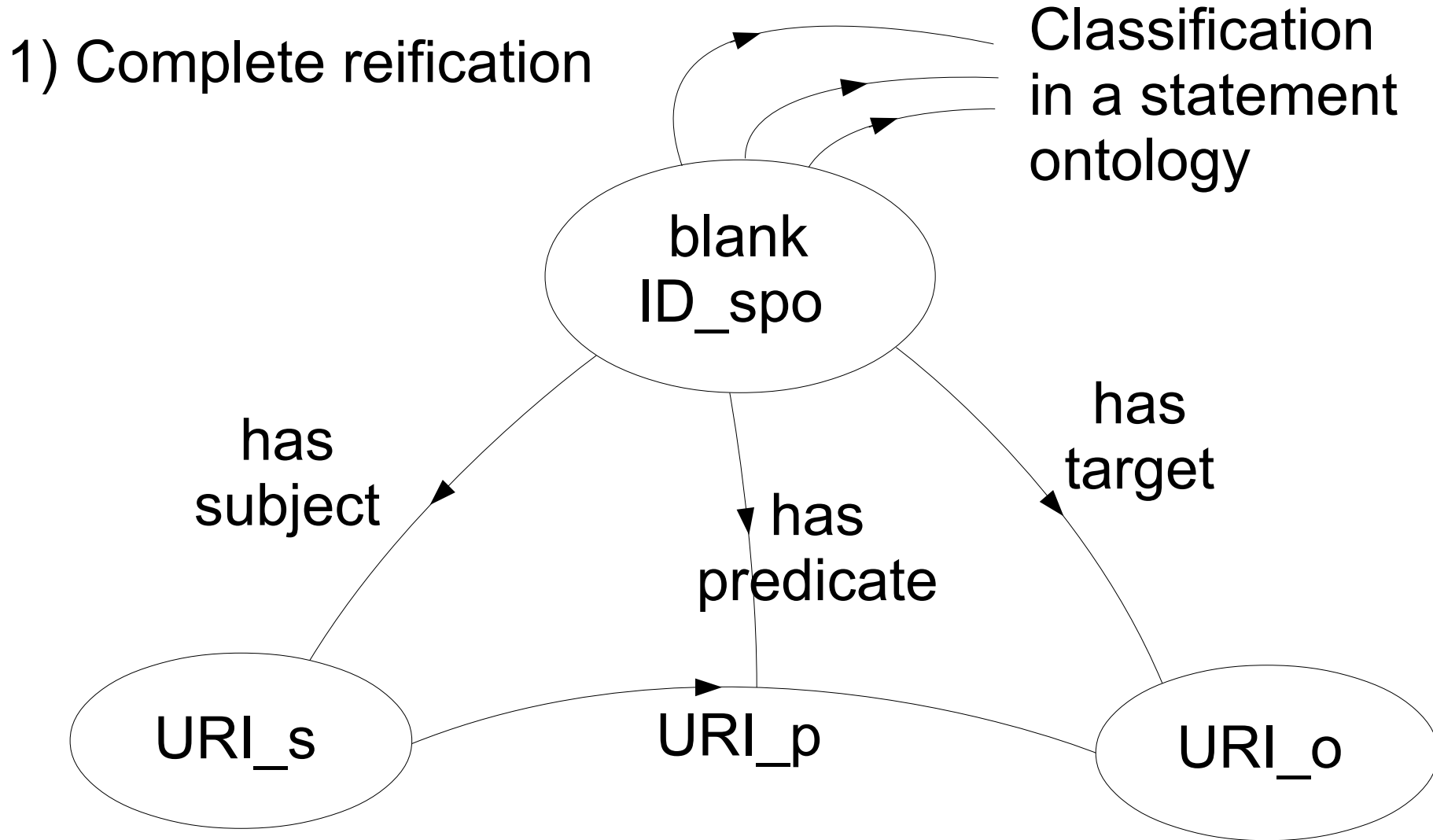
Describe a URI_a via the SVO-intuition:

```
SELECT *  
WHERE {  
  {?s ?p URI_a.  
   ?s rdfs:label ?subject.  
   ?p rdfs:label ?predicate.}  
  UNION  
  {URI_a ?p ?o.  
   ?p rdfs:label ?predicate.  
   ?o rdfs:label ?object.}  
}
```

Describing a URI is the core of the **Linked Data** idea. It can facilitate browsing and automated **visualization**.

This single query fits for all statements.

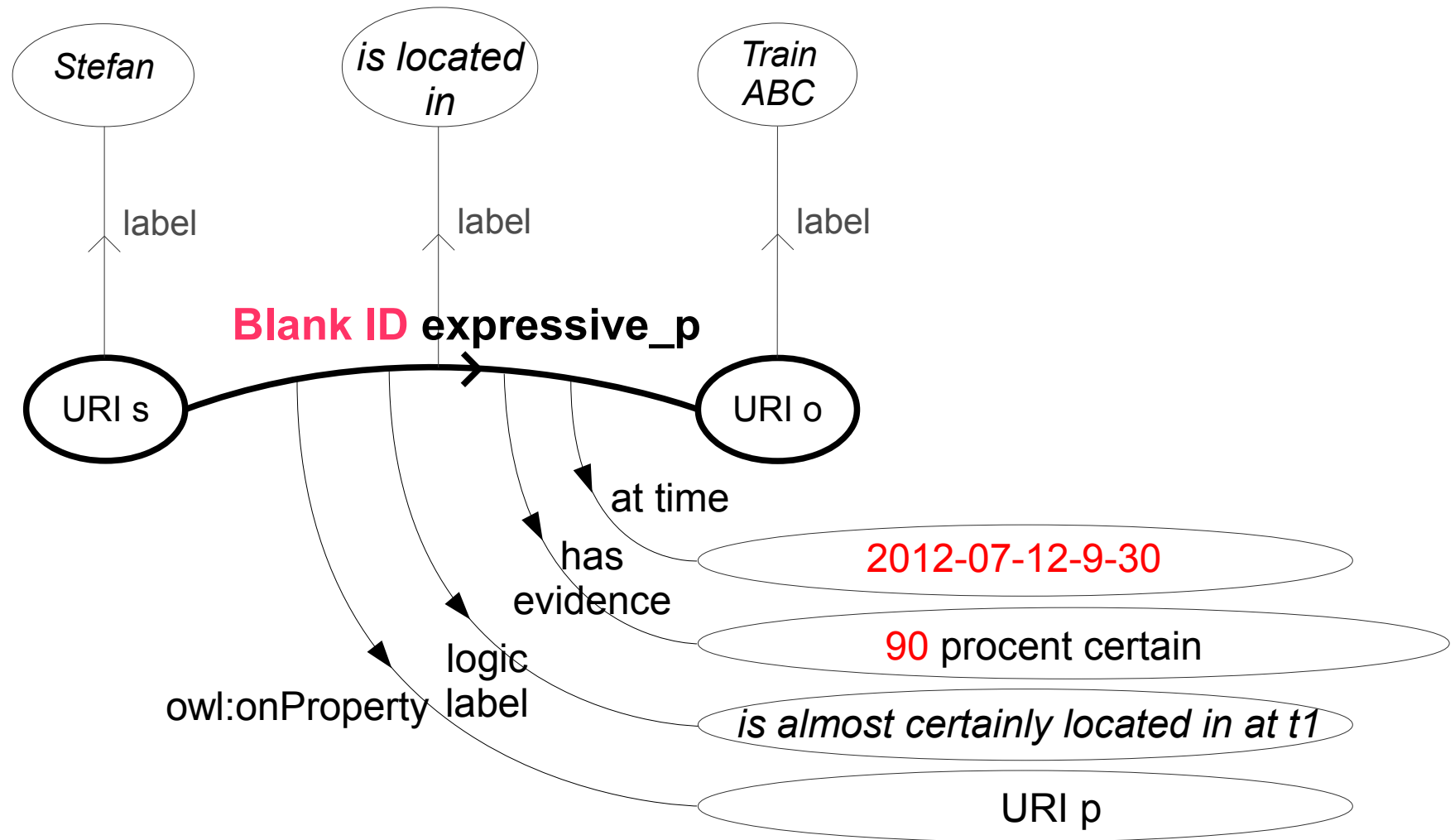
What the proposal is NOT



Very expressive, but a completely **different model**, requiring different queries and chain rules.

What the proposal is NOT

2) Blank predicates



Compatible model, but **not allowed** in standard RDF

Conclusions

- **Expressive** Knowledge Management can be combined with **intuitive** SPARQL browsing by avoiding the break-up of **SVO-triples**
- Extra information can be added on SVO-triples by the **distinction of many semantically precise predicates**, which can be classified in a special **predicate ontology** (like e.g. Metarel)