Overview of RDF Data Model

Jose Emilio Labra Gayo
WESO Research group
University of Oviedo, Spain

Eric Prud'hommeaux
World Wide Web Consortium
MIT, Cambridge, MA, USA

Harold Solbrig
Mayo Clinic, USA

Iovka Boneva
LINKS, INRIA & CNRS
University of Lille, France

Short history of RDF

Around 1997 - PICS, Dublin core, Meta Content Framework

1997 1st Working draft https://www.w3.org/TR/WD-rdf-syntax-971002

RDF/XML

1999 1st W3c Rec https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

First applications RSS, EARL

2004 - RDF Revised https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

Emergence of SPARQL, Turtle, Linked Data

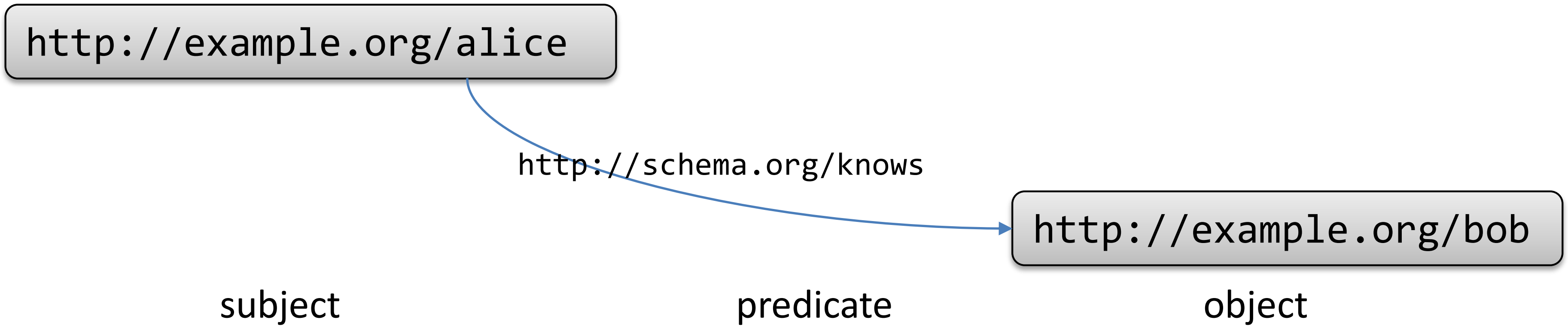
2014 - RDF 1.1 https://www.w3.org/TR/rdf11-concepts/

RDF Data Model

RDF Graph = set of triples

A triple = (subject, predicate, object)

Example:



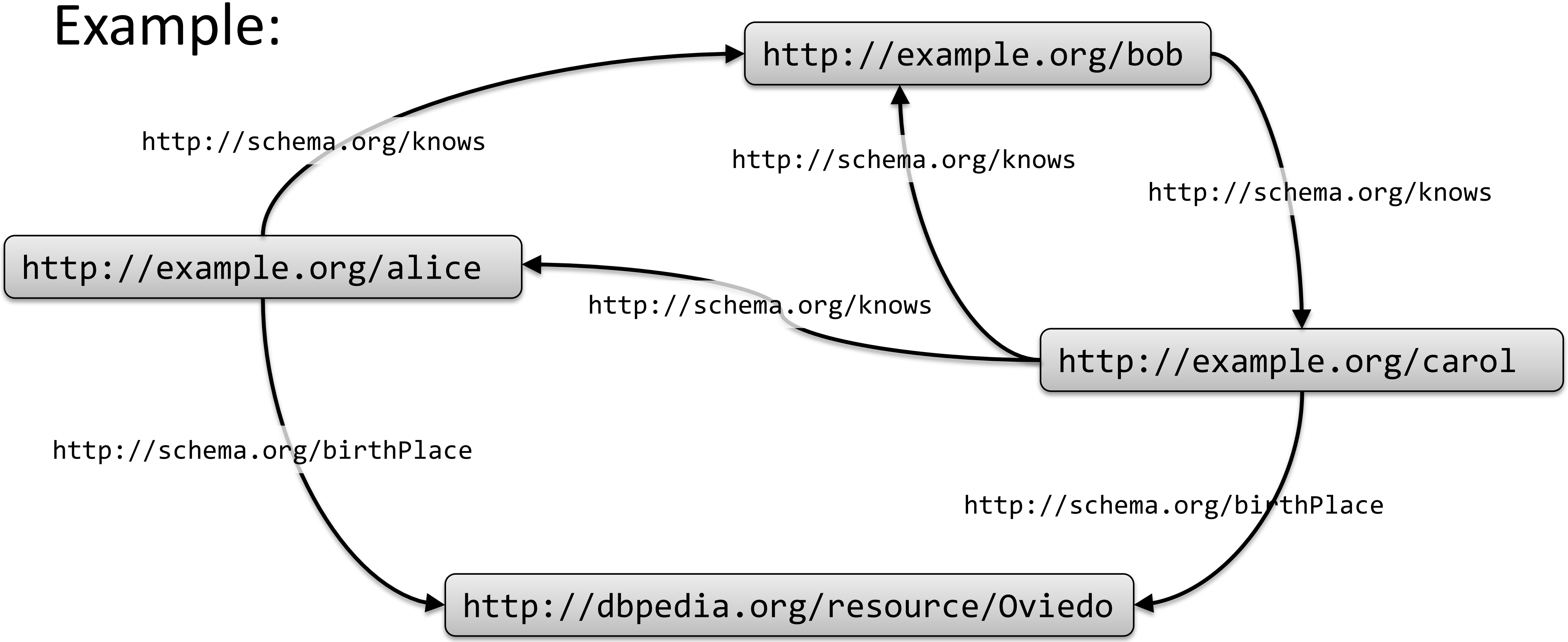
N-Triples representation

```
<http://example.org/alice> <http://xmlns.com/foaf/0.1/knows> <http://example.org/bob> .
```

RDF Graph

RDF Graph = set of triples

Example:



RDF Graph

N-triples representation

```
<http://example.org/alice> <http://schema.org/knows>      <http://example.org/bob> .
<http://example.org/bob>   <http://schema.org/knows>      <http://example.org/carol> .
<http://example.org/carol> <http://schema.org/knows>      <http://example.org/alice> .
<http://example.org/carol> <http://schema.org/knows>      <http://example.org/bob> .
<http://example.org/alice> <http://schema.org/birthPlace> <http://dbpedia.org/resource/Oviedo> .
<http://example.org/carol> <http://schema.org/birthPlace> <http://dbpedia.org/resource/Oviedo> .
```

Turtle Syntax

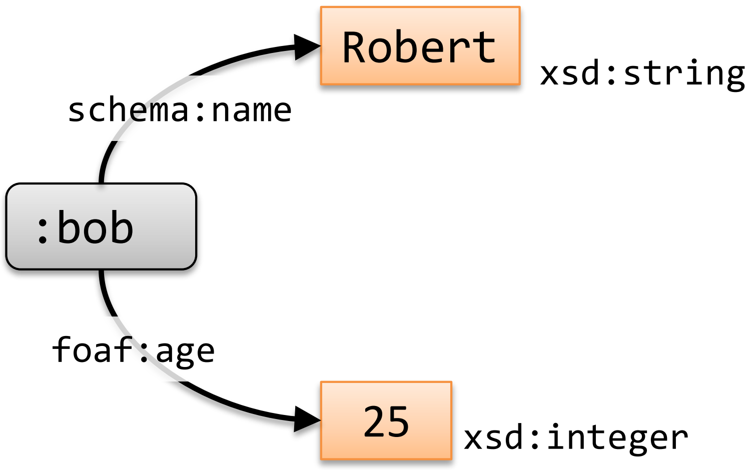
Some simplifications

- prefix declarations
- ; when triples share the subject
- , when triples share subject and object

```
prefix :       <http://example.org/>
prefix schema: <http://schema.org/>
prefix dbo:    <http://dbpedia.org/ontology/>
prefix dbr:    <http://dbpedia.org/resource/>

:alice  schema:birthPlace  dbr:Oviedo ;
        schema:knows      :bob .

:bob    schema:knows      :carol .

:carol  schema:birthPlace  dbr:Oviedo ;
        schema:knows      :alice ,
                          :bob .
```

Literals

- Objects can also be literals
  - Literals contain a lexical form and a datatype
  - Typical datatypes: XML Schema primitive datatypes
    - If not specified, a literal has type `xsd:string`



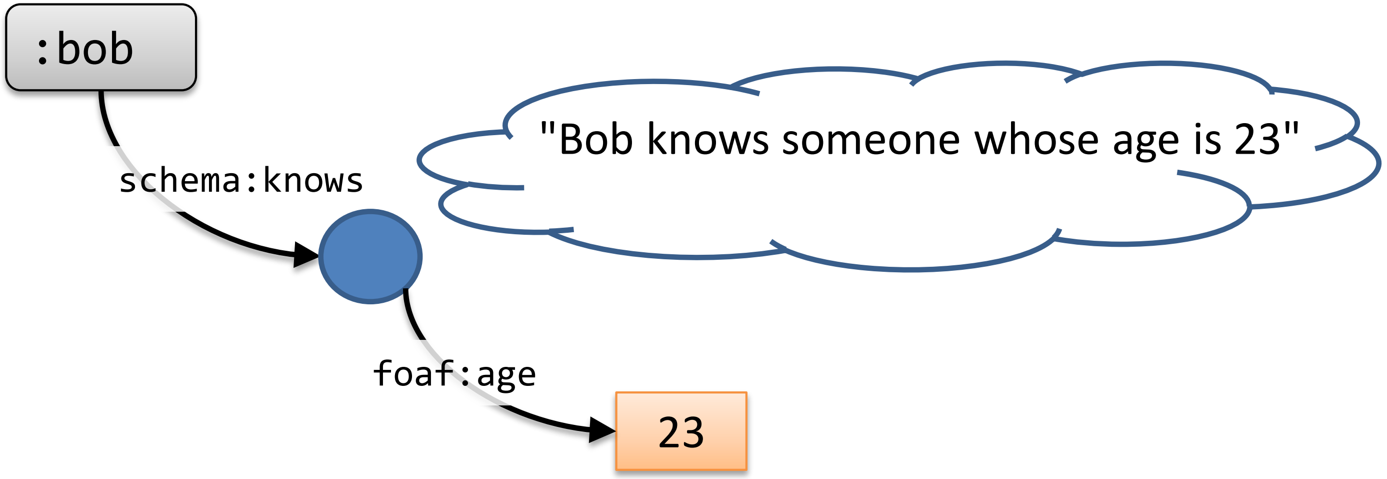
```
:bob schema:name "Robert" ;
     foaf:age   25 .
```

```
:bob schema:name "Robert"^^<xsd:string> ;
     foaf:age   25^^<xsd:integer>   .
```

Blank nodes

Subjects and objects can also be Blank nodes

Blank nodes can have local identifiers



```
:bob foaf:knows _:1 .
_:1  foaf:age 23  .
```

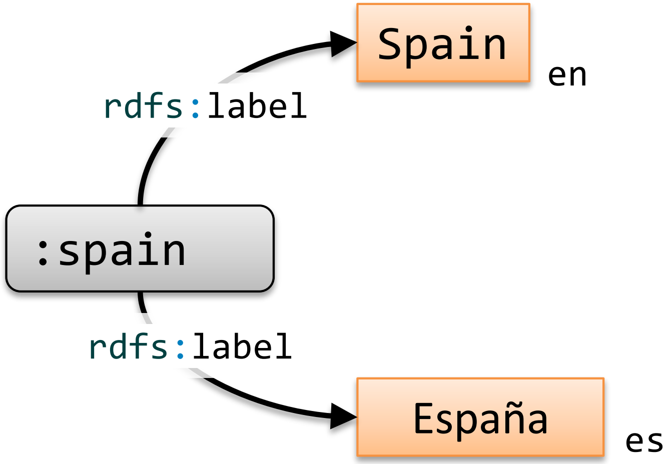
or

```
:bob foaf:knows [
           foaf:age 23
    ] .
```

Language tagged strings

String literals can be qualified by a language tag

- They have datatype rdfs:langString



```
:spain rdfs:label "Spain"@en ;
       rdfs:label "España"@es .
```

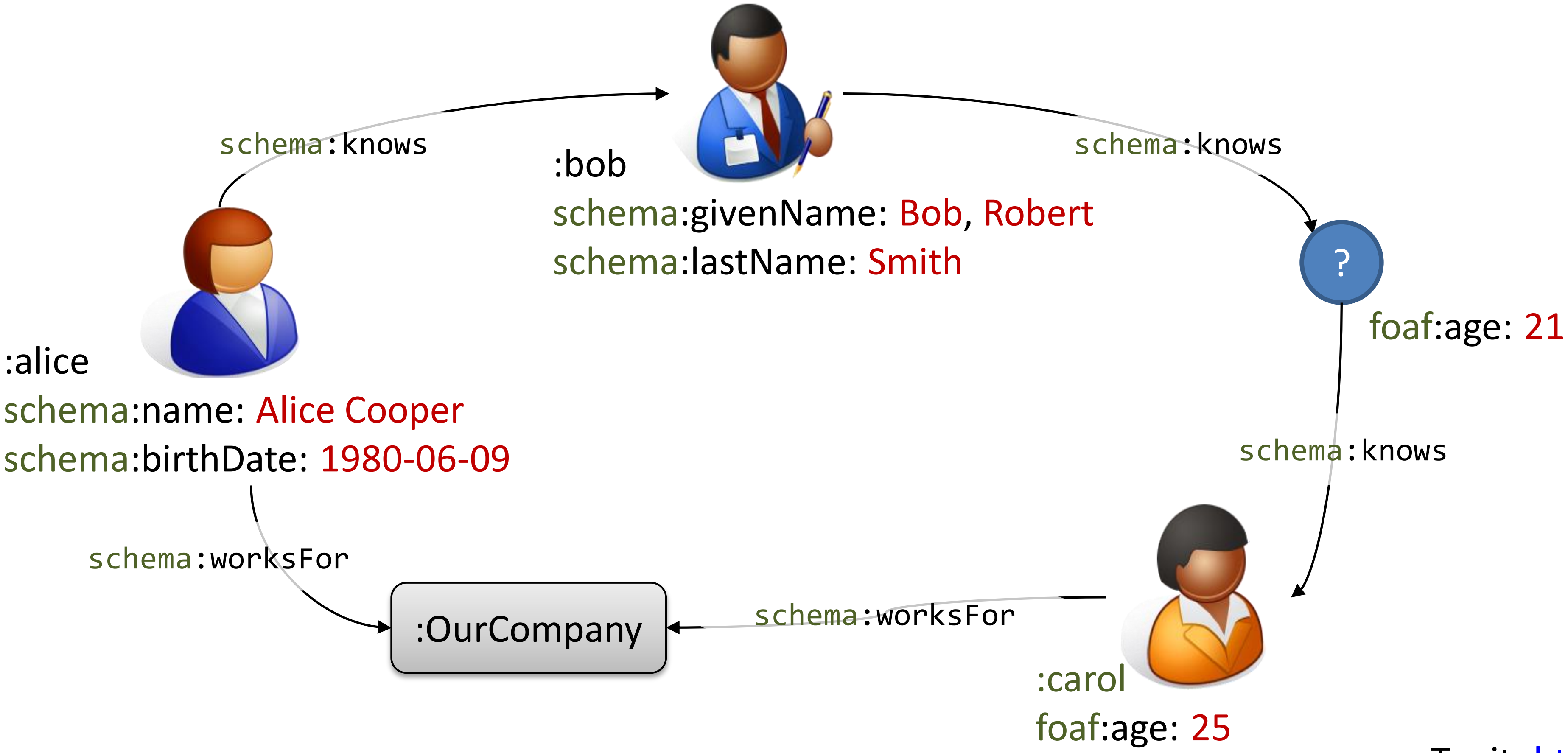
...and that's all?

Yes, the RDF Data model is simple

Simple
is
better

Exercise

Define the following information in RDF



Try it: http://goo.gl/Ve66q1

Continue with RDF Validation tutorial

http://www.slideshare.net/jelabra/rdf-validation-tutorial