Data validation workshop
Rail Data Forum

Veronika Heimsbakk

Jose Emilio Labra Gayo

About us

Veronika Heimsbakk

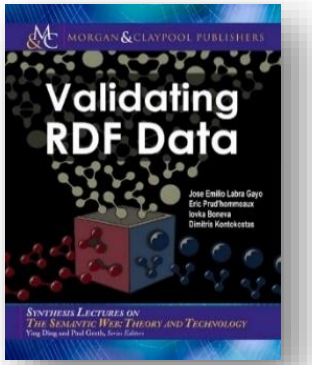
https://veronahe.wordpress.com/

2025

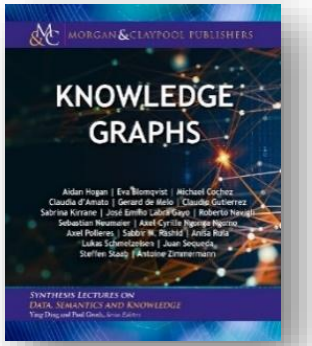
sh@veronahe.no

Jose Emilio Labra Gayo

http://labra.weso.es



2017 HTML version:
http://book.validatingrdf.com



2021, HTML version
https://kgbook.org/

Contents

Motivation: why validating RDF data?

Languages for validating RDF: ShEx and SHACL

Introduction to SHACL

SHACL hands-on section

SHACL applications and use cases

Discussion and wrap up

RDF overview

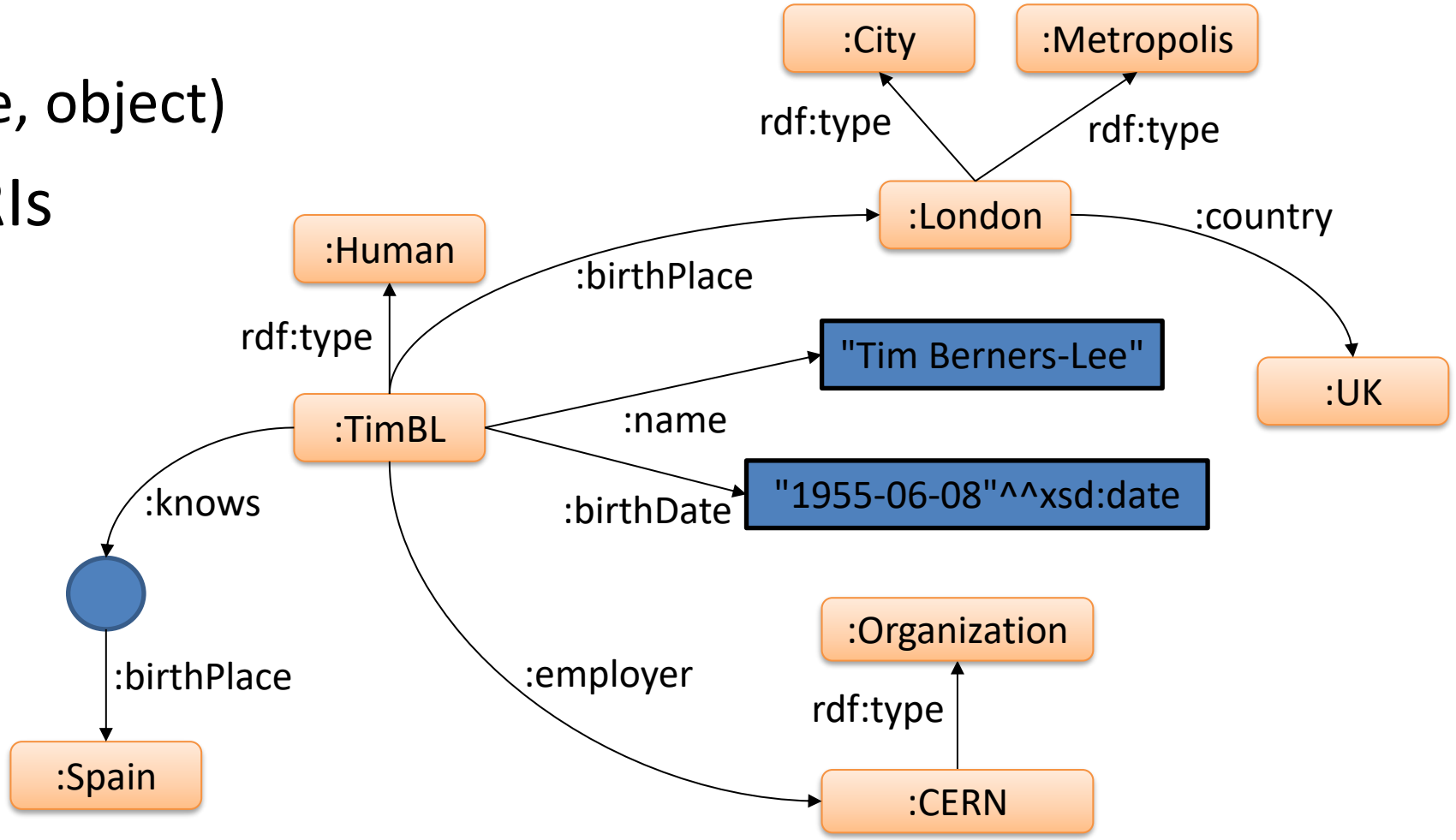
Based on triples

(subject, predicaje, object)

Most nodes are URIs

Interoperability

Simple & flexible



RDF ecosystem

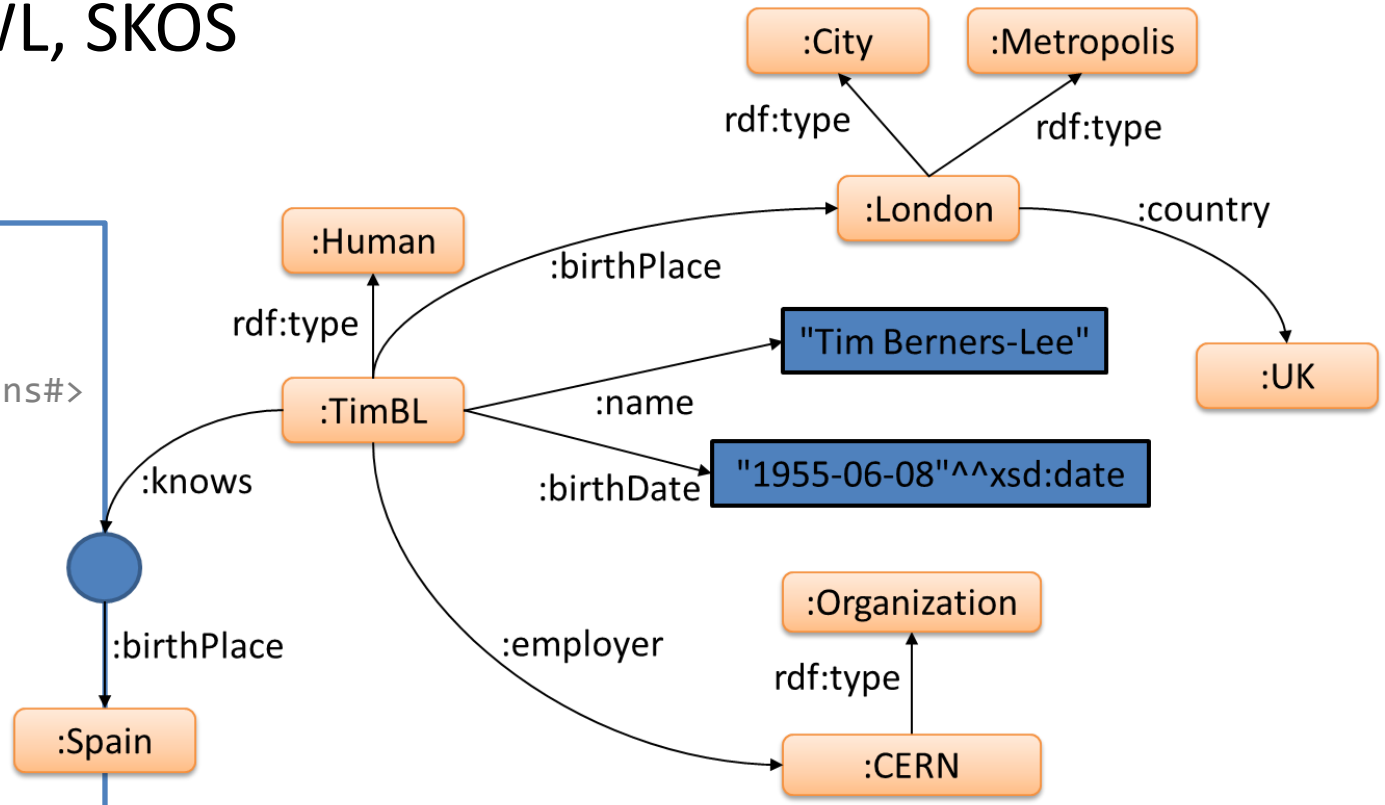
One data model, several syntaxes like Turtle, JSON-LD, …

Vocabularies: RDF Schema, OWL, SKOS

Turtle

```
prefix :       <http://example.org/>
prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
prefix xsd:    <http://www.w3.org/2001/XMLSchema#>
prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

:timbl  rdf:type     :Human ;
        :birthPlace  :london ;
        :name        "Tim Berners-Lee" ;
        :birthDate   "1955-06-08"^^xsd:date ;
        :employer    :CERN ;
        :knows       _:1    .
:london rdf:type     :City, :Metropolis ;
        :country     :UK .
:CERN   rdf:type     :Organization .
_:1     :birthPlace  :Spain .
```



Try it:
https://rdfshape.weso.es/link/17313171788

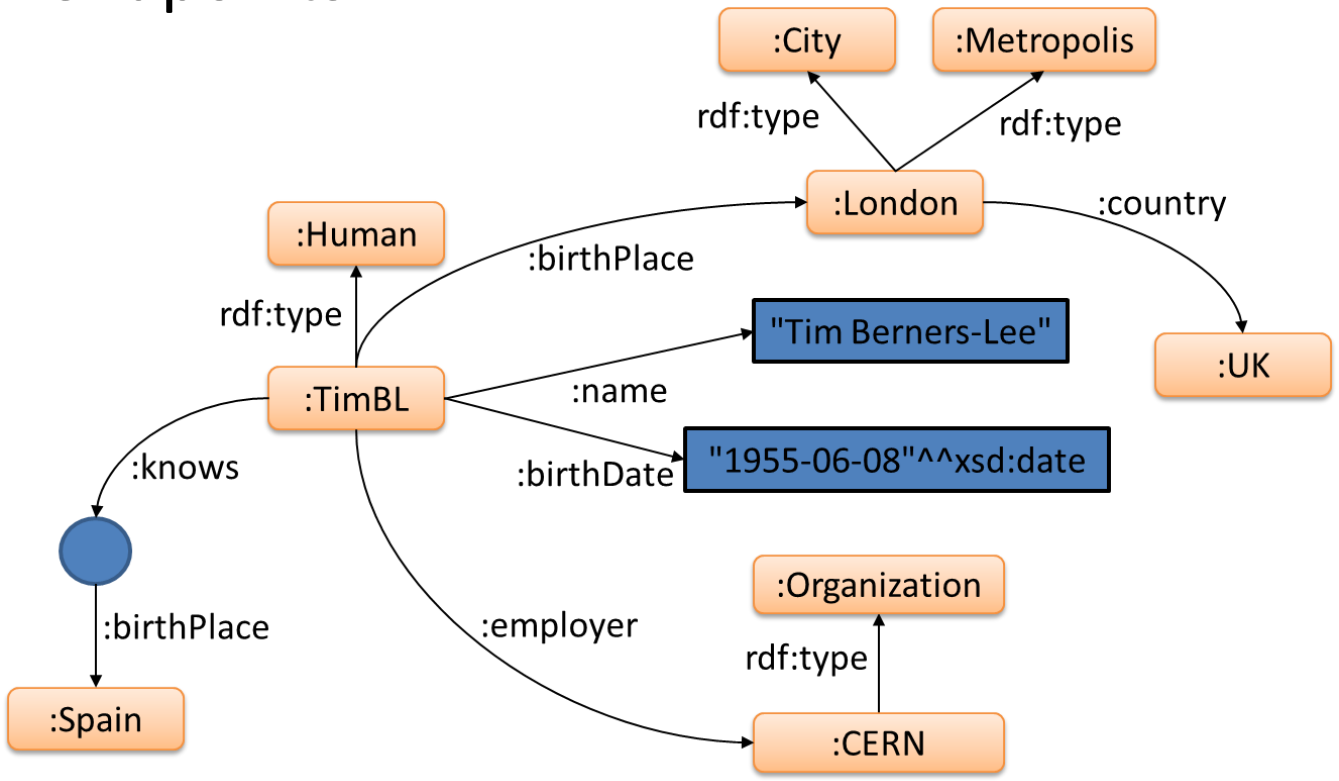
RDF ecosystem: SPARQL

SPARQL = RDF query language and protocol

Enables the creation of SPARQL endpoints

```
SELECT ?person ?date ?country WHERE {
 ?person :birthDate  ?date .
 ?person :birthPlace ?p .
 ?p      :country    ?country
}
```

?person	?date	?country
:timbl	1955-06-08	:UK



Try it: https://rdfshape.weso.es/link/17313175698

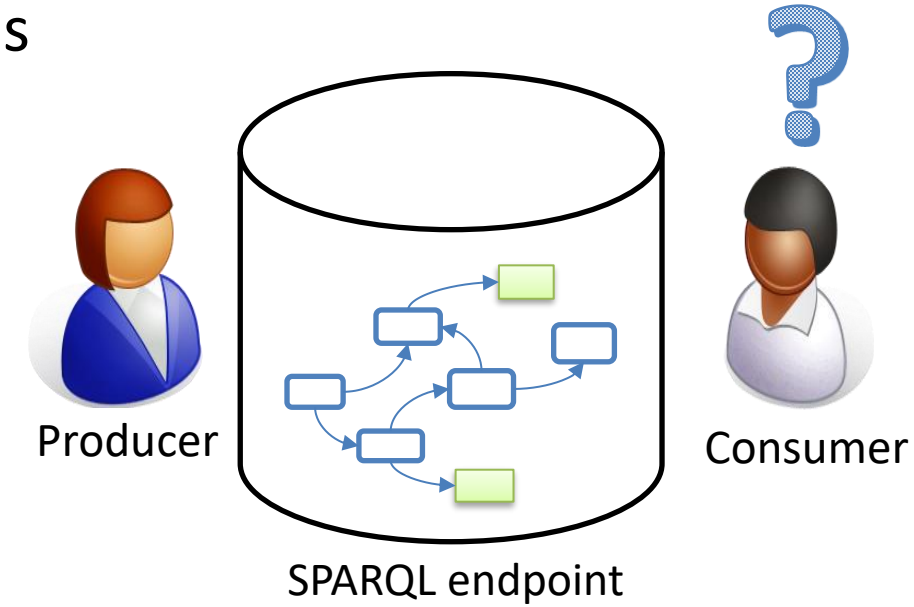
WESO

RDF, the good parts...

- RDF as an integration language
  - RDF as a *lingua franca* for semantic web and linked data
  - Basis for knowledge representation
- RDF flexibility
  - Data can be adapted to multiple environments
  - Reusable data by default
  - **AAA** principle: **A**nyone can say **A**nything about **A**ny topic
- RDF tools
  - RDF data stores & SPARQL
  - Several serializations: Turtle, JSON-LD, RDF/XML...
  - Can be embedded in HTML (Microdata/RDFa)

RDF, the other parts

- Consuming & producing RDF
  - Describing and validating RDF content
  - SPARQL endpoints are not well documented
    - Typical documentation = set of SPARQL queries
    - Difficult to know where to start doing queries
  - Too much flexibility?
    - AAA principle allows to add any thing
    - Most SPARQL endpoints contain errors



Running example: some errors

Well formed RDF can contain some mistakes

```
:timbl a            :Human              ;
       :name        "Tim Berners-Lee"   ;
       :knows       :bob                .

:alice a            :Human ;
       :name        234    .

:bob   a            :Human           ;
       :name        "Bob", "Robert"  ;
      :knows        :CERN            .

:carol a            :Human  ;
       :birthPlace  :london .

:dave a             :Human  ;
     :name          "David" .

:CERN a             :Organization ;
      :name         "CERN"        .
```

In other technologies...

Technology	Schema
Relational Databases	DDL
XML	DTD, XML Schema, RelaxNG, Schematron
JSON	JSON Schema
RDF	?

Fill that gap

RDF Schema for validating ?

RDF Schema can define classes, properties

It adds a simple inference layer

But not useful to validate

```
:Human rdfs:subClassOf :HomoSapiens .
:knows rdfs:domain     :Human       ;
       rdfs:range      :Human       .
```

```
:timbl a       :Human              ;
       :name   "Tim Berners-Lee"   ;
       :knows  :bob                .

:bob a         :Human              ;
     :name     "Bob", "Robert"     ;
     :knows    :CERN               .

:CERN a :Organization .
```

It infers that CERN is a homo sapiens

Try it: https://rdfshape.weso.es/link/17496263237

Note:
Maybe, the name “RDF Schema” was not a good idea

OWL for validating?

OWL is a language to define ontologies

Ontologies are great to describe domains

- Open World Assumption: If something is not declared, it can be true
- OWL is not very helpful to validate

```
:Human a owl:Class ; rdfs:subClassOf :HomoSapiens .
:HomoSapiens a owl:Class .

:name a owl:DatatypeProperty, owl:FunctionalProperty ;
    rdfs:domain :Human;
    rdfs:range  xsd:string .

:knows a owl:ObjectProperty,
       rdfs:domain :Person ;
       rdfs:range  :Person .
```

```
:timbl a :Human                          ;
       :name  "Tim Berners-Lee" ;
       :knows :bob                       .

:bob a :Human                        ;
     :name "Bob", "Robert" ;
     :knows :CERN                    .

:CERN a :Organization .
```

It infers that CERN is a homo sapiens

Try it: https://rdfshape.weso.es/link/17496276991

SPARQL for validating?

It works on the existing RDF data

- Closed World Assumption

Can be used to validate

Pros: It is very expressive

Cons: It is ***too*** expressive

- Queries difficult to write and debug
- Too low level

```
SELECT ?person WHERE {
  ?person a :Human ;
  {
    SELECT ?person ?name WHERE {
    ?person :name ?name .
    FILTER (!isLiteral(?name) ||
              datatype(?name) != xsd:string)
    }
  } UNION {
    SELECT ?person (COUNT(?name) AS ?nameCount)
    WHERE {
        ?person a :Human .
        OPTIONAL { ?person :name ?name }
    }
    GROUP BY ?person
    HAVING (COUNT(?name) != 1)
  }
}
```

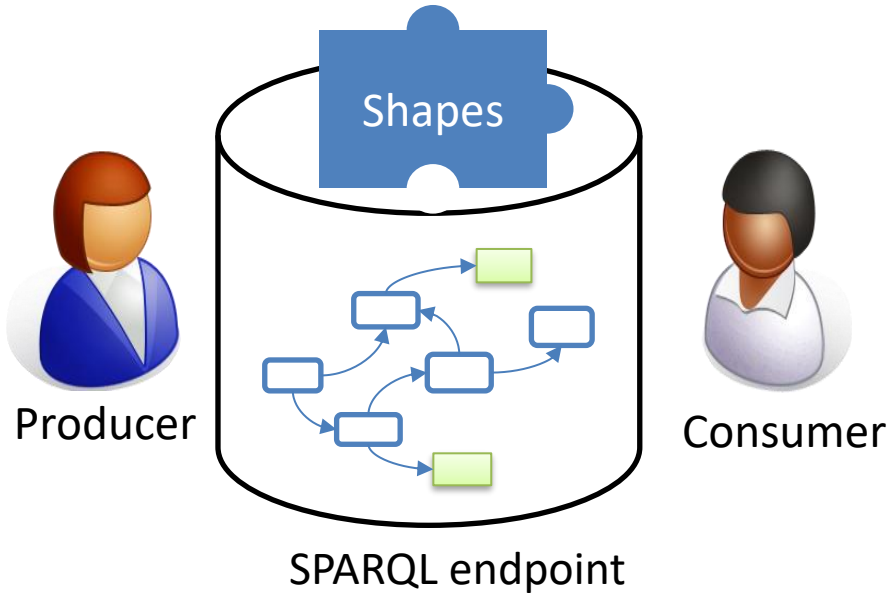
Shapes for validating

For producers

- Validate contents they are producing
- Ensure they produce the expected structure
- Advertise and document the structure
- Generate interfaces

For consumers

- Understand contents
- Verify structure before processing it
- Query generation & optimization



Contents

Motivation: why validating RDF data?

Languages for validating RDF: ShEx and SHACL

Introduction to SHACL

SHACL hands-on section

SHACL applications and use cases

Discussion and wrap up

ShEx & SHACL

2013 RDF Validation Workshop

- Conclusions of the workshop:
  - *There is a need of a higher level, concise language for RDF Validation*
- ShEx initially proposed (v 1.0)

2014 W3c Data Shapes WG chartered

2017 SHACL accepted as W3C recommendation

2017 ShEx 2.0 released as W3C Community group draft

2019 ShEx adopted by Wikidata

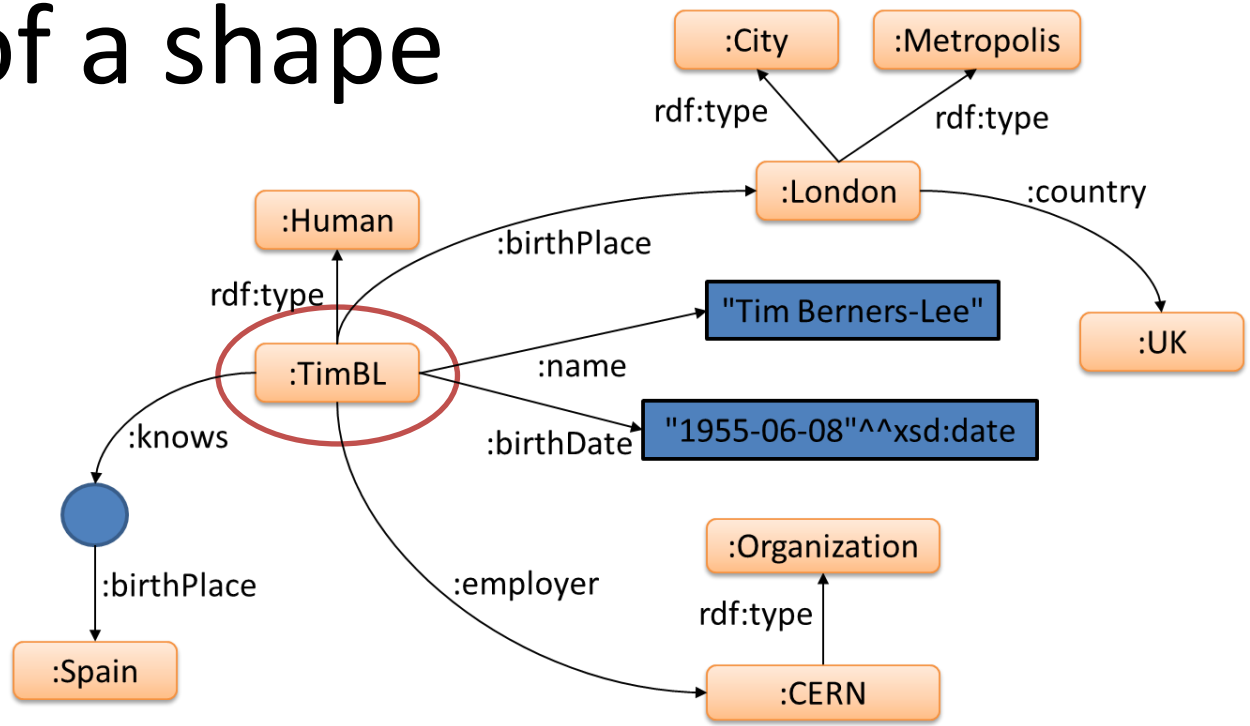
2025 IEEE ShEx (*work in progress*)

2025 Data Shapes WG chartered again for SHACL 1.2 (*work in progress*)

Example of a shape

A shape describes

- The form of a node (node constraint)
- Incoming/outgoing arcs from a node
- Possible values associated with those arcs



RDF Node

```
:timbl :name        "Tim Berners-Lee" ;
       :birthPlace  :london ;
       :birthDate   "1955-06-08"^^xsd:date ;
       :employer    :CERN .
       :knows       _:1 .
```

```
<Person> {
  :name        xsd:string          ;
  :birthPlace  @<Place>          ? ;
  :birthDate   xsd:date          ? ;
  :employer    @<Organization> * ;
  :knows       @<Person>         * ;
}
```

Try it: https://rdfshape.weso.es/link/16685137872

Several common features…

	ShEx	SHACL
Employ the word "shape"	✓	✓
Validate RDF graphs	✓	✓
Node constraints	✓	✓
Constraints on incoming/outgoing arcs	✓	✓
Defining cardinalities on properties	✓	✓
RDF syntax	✓	✓
Extension mechanism	✓	✓

ShEx and SHACL compared

ShEx

```
:Person {
 :name        xsd:string        ;
 :birthPlace @:Place          ? ;
 :birthDate   xsd:date        ? ;
 :employer   @:Organization * ;
 :knows      @:Person         *
}
```

SHACL

```
:Person a sh:NodeShape ;
   sh:targetClass :Human ;
   sh:property [ sh:path :name ;
    sh:minCount 1; sh:maxCount 1;
    sh:datatype xsd:string ;
  ] ;
  sh:property [ sh:path :birthPlace ;
    sh:node :Place
  ] ;
  sh:property [ sh:path :birthDate ;
   sh:maxCount 1;
   sh:datatype xsd:date;
  ] ;
  sh:property [ sh:path :employer ;
    sh:node :Organization
  ] ;
  sh:property [ sh:path :knows ;
    sh:node :Person
  ] .
```

Note:
Cyclic data models are implementation dependent in SHACL

But several differences…

Underlying philosophy

Syntactic differences

Notion of a shape

Syntactic differences

Default cardinalities

Shapes and Classes

Recursion

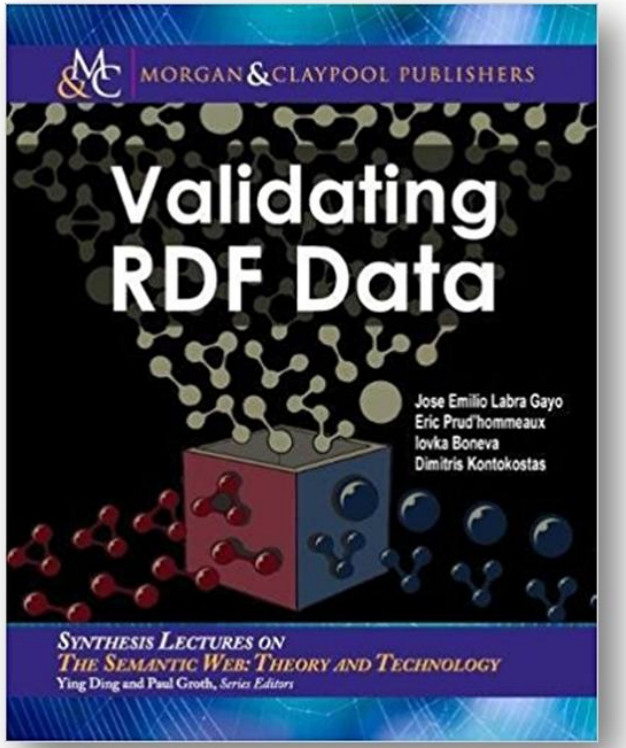
Repeated properties

Property pair constraints

Uniqueness

Extension mechanism

More info in Chapter 7 of:



http://book.validatingrdf.com/

Underlying philosophy
description - validation - constraints

Description
Tell what something is about
What data we want/expect

ShEx

Validation
Check that data conforms with expectations

SHACL

Constraint
Rule that something has to obey
What data we don't want/expect

Contents

Motivation: why validating RDF data?

Languages for validating RDF: ShEx and SHACL

Introduction to SHACL

SHACL hands-on section

SHACL applications and use cases

Discussion and wrap up