Let's dive into EntitySchemas

Jose Emilio Labra Gayo
WESO, University of Oviedo

Seyed Amir Hosseini Beghaeiraveri
University of Edinburgh, LFCS

Andra Waagmeester
Micelio

Eric Prud'hommeaux
Janeiro Digital

Slides for Data Modelling Days event
Wikidata - 30/Nov/2023

Agenda

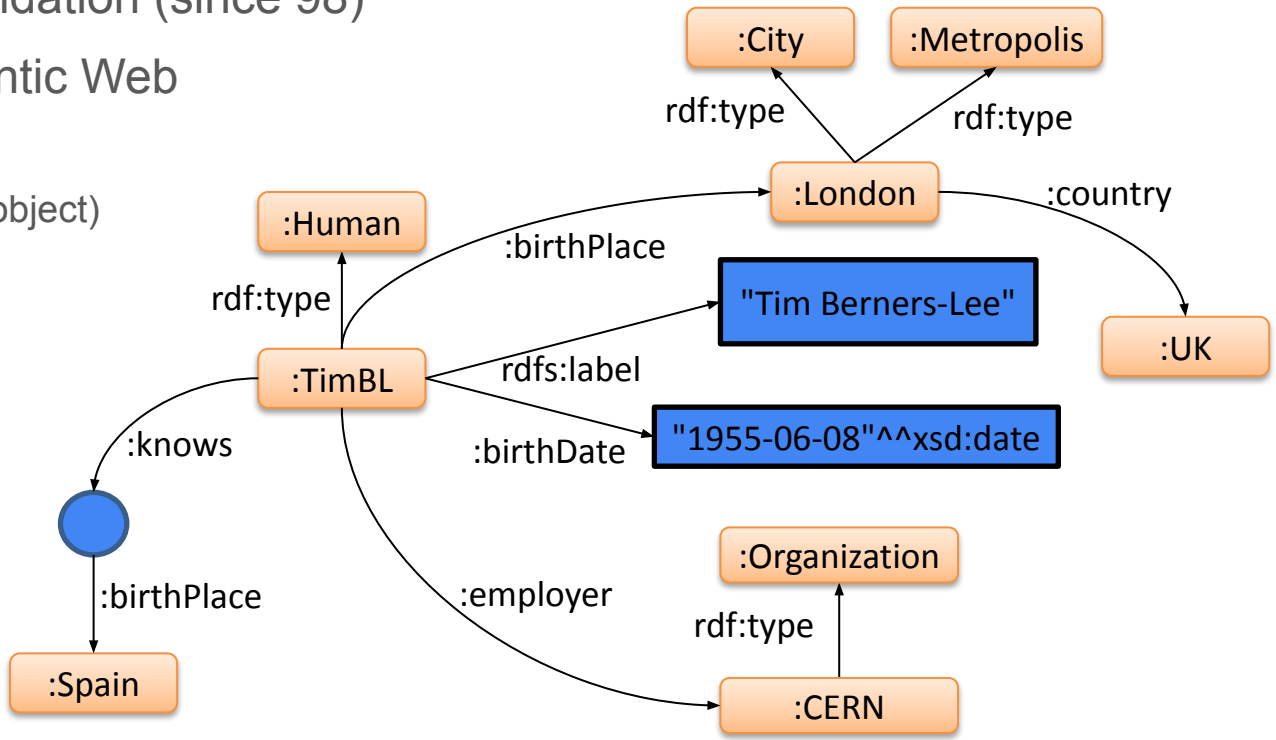
Preliminaries: RDF, ShEx, Entity schemas, …
Tools for entity schemas
Exercise creating an entity schema
Some applications of entity schemas
Discussion

RDF graphs

RDF = W3C recommendation (since 98)

Lingua franca of Semantic Web

- Based on triples
  - (subject, predicaje, object)
- Most nodes are URIs
  - Interoperability



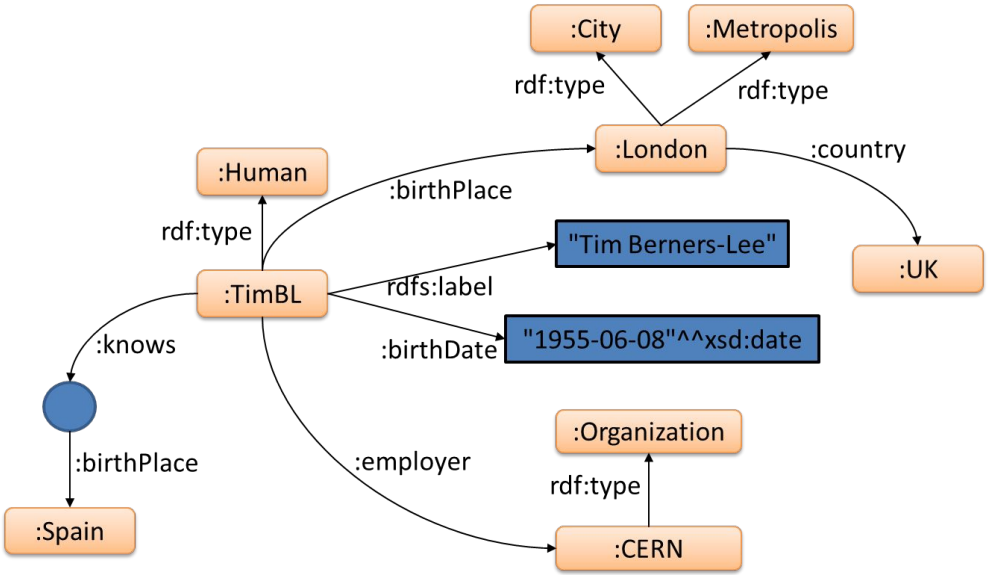
RDF ecosystem

One data model, several syntaxes: Turtle, N-Triples, JSON-LD

Vocabularies: RDF Schema, OWL, SKOS, etc.

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 ;
        rdfs:label   "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 .
```



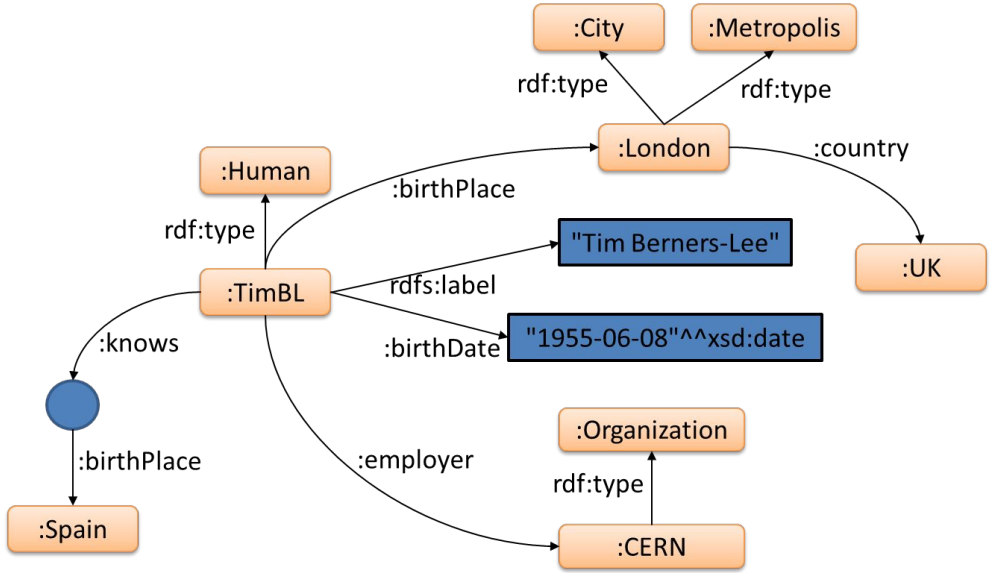
RDF ecosystem: SPARQL

SPARQL is an RDF query language and protocol

It 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



Wikibase graphs

Popularized by Wikidata

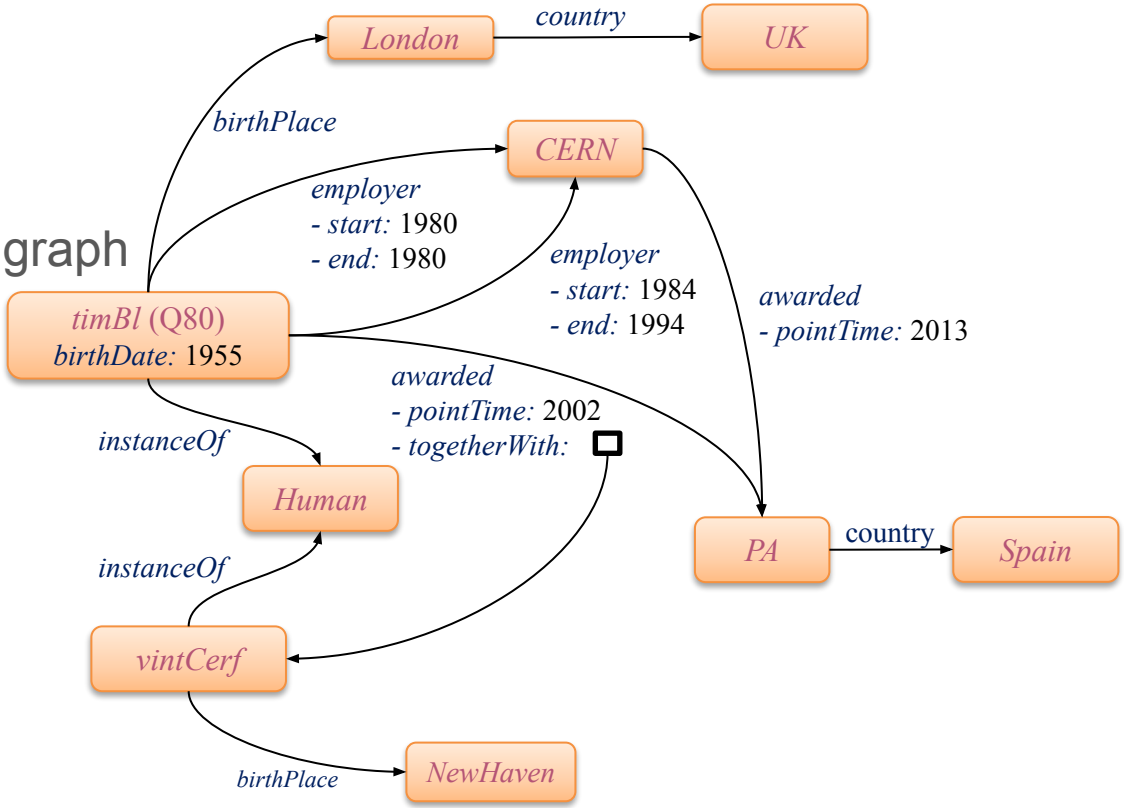
Wikibase = software supporting Wikidata

The values can be nodes in the graph

Example:

Tim Berners Lee

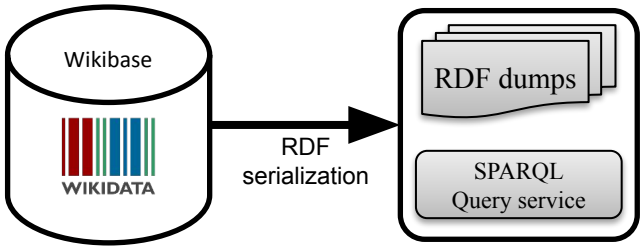
http://www.wikidata.org/entity/Q80



Wikibase graphs and RDF

Wikibase graphs generate RDF serializations for each item

SPARQL endpoint and Query service available




```
select ?name ?date?country where {
 wd:Q80 wdt:P1559 ?name .
 wd:Q80 wdt:P569  ?date .
 wd:Q80 wdt:P19   ?place .
 ?place wdt:P17   ?country
}
```

?name	?date	?country
Tim Berners-lee	1955-06-08	:UK

Try it:
https://w.wiki/5yGu

Wikibase RDF representation

RDF

```
wd:Q80       wdt:P166  wd:Q3320352  ;
             p:P166    s:PQ80-494FA .

s:PQ80-494FA ps:P166   wd:Q3320352 ;
             pq:P585   2002        ;
             pq:P1706  wd:Q62843, wd:Q92935, wd:Q92743.
```

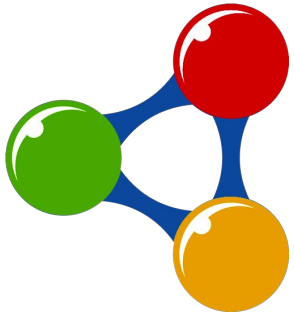
Could be represented as

```
:Q80 :P166  :Q3320352 {|
       :P585  2002;
       :P1796 :Q62843, :Q92935, :Q92743
  |}
```

Syntax inspired by RDF-Star annotated triples syntax

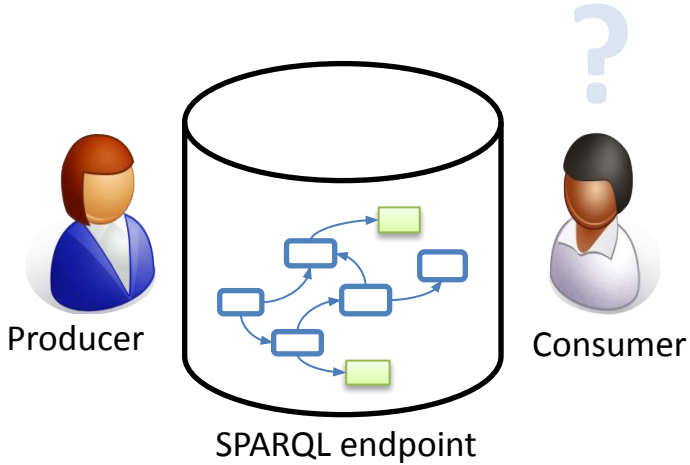
RDF, the good parts...

- 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
- 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



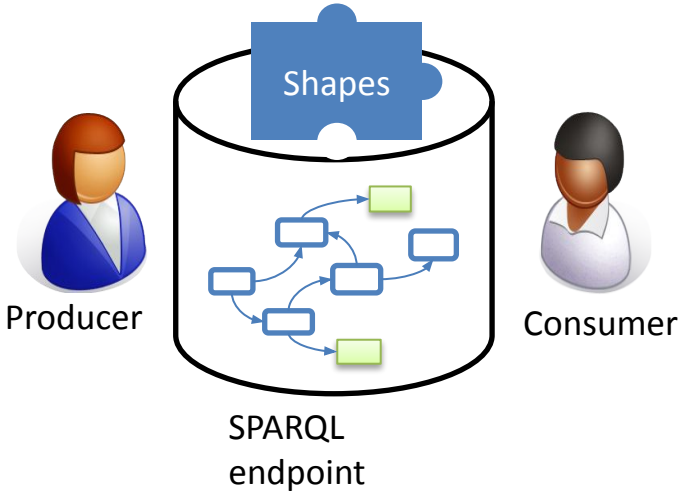
Why describe & validate RDF?

For producers

- Understand the contents they are going to produce
- Ensure they produce the expected structure
- Advertise and document the structure
- Generate interfaces

For consumers

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



Similar technologies

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

Fill that gap

Schemas for RDF?

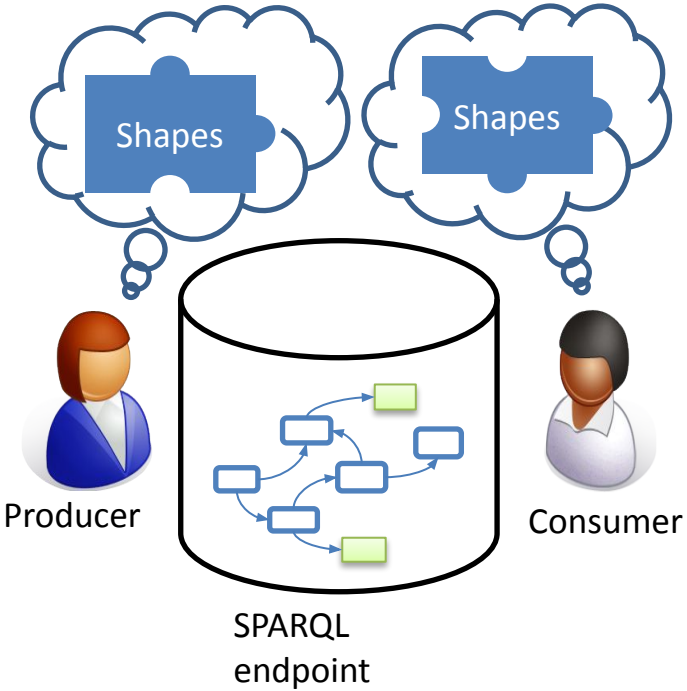
RDF flexibility doesn't want to impose a schema, but...

In practice, there are **implicit schemas**

- Assumed by producers and consumers

Shapes can make schemas explicit

- Handle malformed/incomplete data
- Avoid defensive programming



Shapes for consensus building

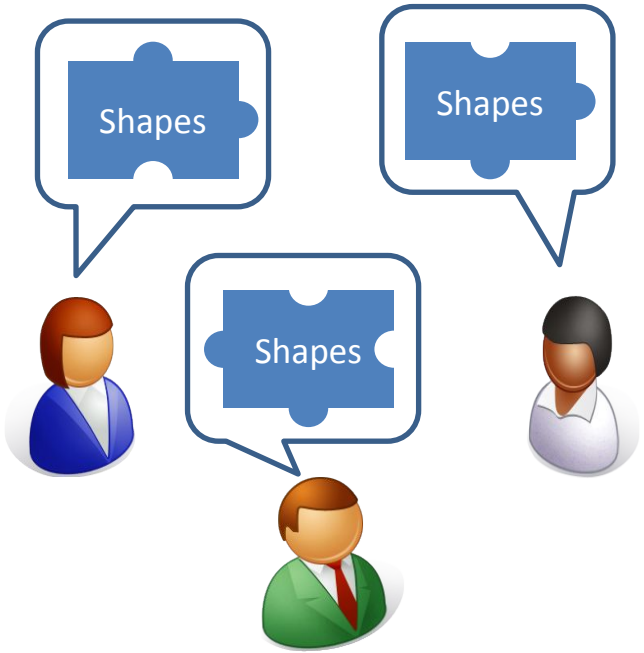
Initial motivation: clinical data models (FHIR)

Distributed, extensible content models

- Distributed by location and authority
- Extensible content models

Shared schemas

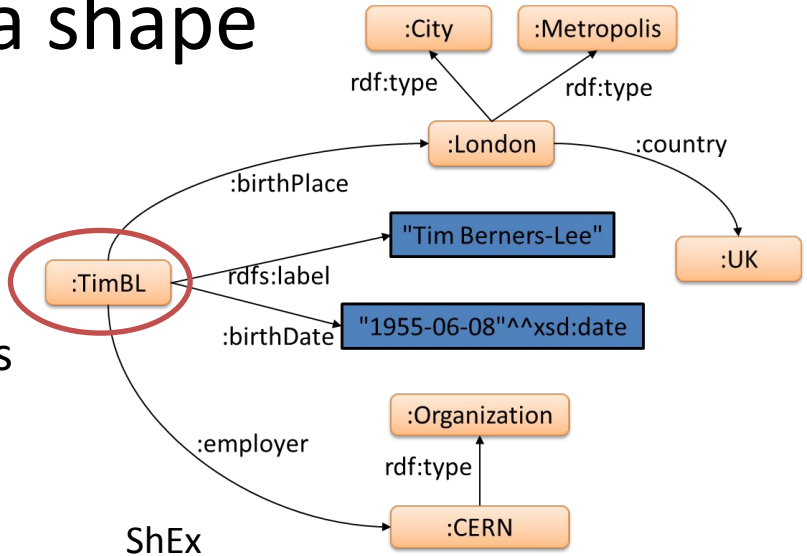
- Understandable by domain experts
- ...and machine processable



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 rdfs:label   "Tim Berners-Lee" ;
       :birthPlace  :london ;
       :birthDate   "1955-06-08"^^xsd:date ;
       :employer    :CERN .
```

ShEx

```
<Researcher> {
 rdfs:label   xsd:string         ;
 :birthPlace  @<Place>          ? ;
 :birthDate   xsd:date          ? ;
 :employer    @<Organization>   * ;
}
```

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

ShEx evolution

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

2022 ShEx + extends

2023 IEEE ShEx (current work in progress)

RDF description - validation - constraints

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

ShEx

Validation
Check that something conforms

SHACL

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

Short intro to ShEx

ShEx (Shape Expressions Language)

Concise and human-readable

Syntax similar to SPARQL, Turtle

Semantics inspired by regular expressions & RelaxNG

2 syntaxes: Compact and RDF/JSON-LD

Official info: http://shex.io

Semantics: http://shex.io/shex-semantics/, primer: http://shex.io/shex-primer

ShEx implementations and playgrounds

Implementations:

- shex.js: Javascript
- Jena-ShEx: Java
- SHaclEX: Scala (Jena/RDF4j)
- PyShEx: Python
- shex-java: Java
- Ruby-ShEx: Ruby
- shex-ex: Elixir
- NEW shex-rs: Rust

Online demos & playgrounds

- ShEx-simple
- RDFShape
- ShEx-Java
- ShExValidata
- Wikishape

Simple example

Prefix declarations are similar to Turtle/SPARQL

```
prefix schema: <http://schema.org/>
prefix xsd:    <http://www.w3.org/2001/XMLSchema#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>

<Researcher> {
 rdfs:label    xsd:string      ;
 schema:knows  @<Researcher> * ;
}
```

Nodes conforming to `<Researcher>` must:

- Have exactly one `rdfs:label` with a value of type `xsd:string`
- Have zero or more `schema:knows` whose values conform to `<Researcher>`

RDF Validation using ShEx

Schema

```
<Researcher> {
 rdfs:label    xsd:string       ;
 schema:knows @<Researcher> * ;
}
```

Shape map

```
:alice@<Researcher>,✔
:bob  @<Researcher>,✔
:carol@<Researcher>,□
:dave @<Researcher>,□
:emily@<Researcher>,□
:frank@<Researcher>,✔
```

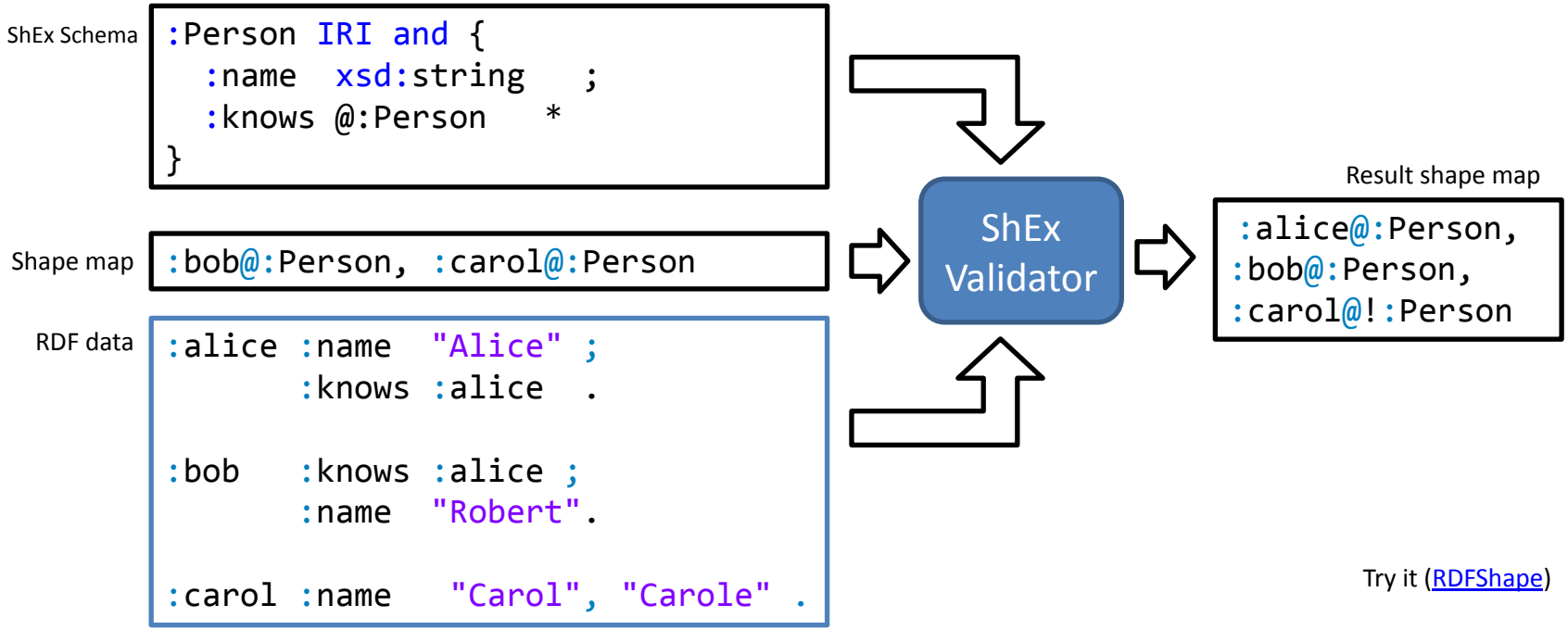
Data

```
:alice rdfs:label   "Alice" ;
       schema:knows :alice  .

:bob   schema:knows :alice ;
       rdfs:label   "Robert".

:carol rdfs:label   "Carol", "Carole" ;
       schema:knows "bob"   .

:dave  rdfs:label   234 ;
       schema:knows :bob .

:emily schema:name  "Emily" .

:frank rdfs:label   "Frank" ;
       schema:email <mailto:frank@example.org> ;
       schema:knows :alice, :bob .
```

Try it (RDFShape): https://rdfshape.weso.es/link/16685282129

Validation process

Input: RDF data, ShEx schema, Shape map
Output: Result shape map



Try it (RDFShape)

Node constraints

Describe the shape of a node

```
:Book {
 :name          xsd:string                      ;
 :datePublished xsd:date                        ;
 :numberOfPages MinInclusive 1                  ;
 :author        @:Person                        ;
 :genre         [ :Action :Comedy :NonFiction ] ;
 :isbn          /isbn:[0-9X]{10}/               ;
 :publisher     IRI                             ;
 :audio         .                               ;
 :maintainer    @:Person OR @:Organization
}

:Person {}
:Organization {}
```

```
:item23
 :name          "Weaving the Web"                ;
 :datePublished "2012-03-05"^^xsd:date           ;
 :numberOfPages 272                              ;
 :author        :timbl                           ;
 :genre         :NonFiction                      ;
 :isbn          "isbn:006251587X"                ;
 :publisher     <http://www.harpercollins.com/>  ;
 :audio         <http://audio.com/item23>        ;
 :maintainer    :alice                           .
```

Try it: (RDFShape)

Cardinalities

Inspired by regular expressions: +, ?, *, {m,n}

By default {1,1}

```
:Book {
 :name           xsd:string         ;
 :numberOfPages  xsd:integer  ?     ;
 :author         @:Person     +     ;
 :publisher      IRI          ?     ;
 :maintainer     @:Person     {1,3} ;
 :related        @:Book       *
}
:Person {}
:Organization {}
```

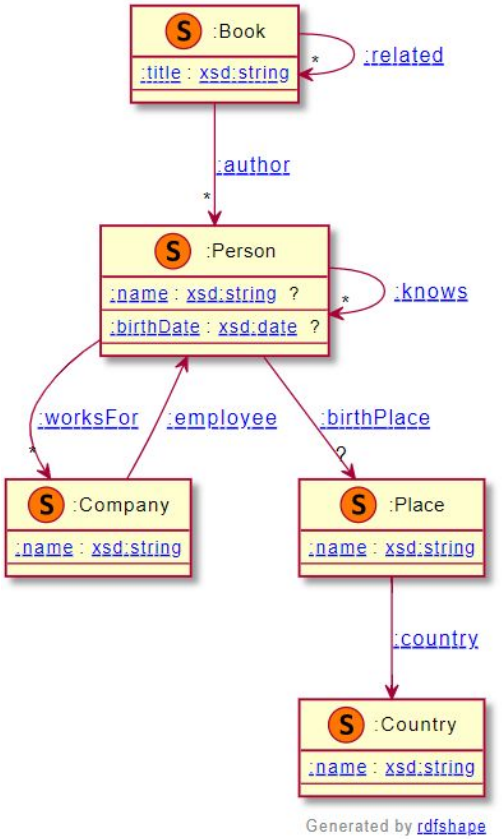
```
:item23
  :name           "Weaving the Web"       ;
  :numberOfPages  272                     ;
  :author         :timbl, :markFischetti  ;
  :maintainer     :alice,:bob             .
```

Try it: RDFShape

Recursive schemas

Well defined supports for cyclic (recursive) schemas

```
:Book {
  :title   xsd:string   ;
  :author  @:Person    * ;
  :related @:Book      * ;
}
:Person {
  :name       xsd:string ? ;
  :birthDate  xsd:date   ? ;
  :birthPlace @:Place    ? ;
  :knows      @:Person   * ;
  :worksFor   @:Company  * ;
}
:Place {
  :name       xsd:string ;
  :country    @:Country  ;
}
:Country {
  :name       xsd:string
}
:Company {
  :name       xsd:string
  :employee   @:Person
}
```



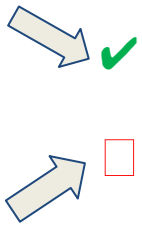
Try it: [RDFShape]

Open/Closed content models

- RDF semantics mostly presume open content models
- Shape expressions are open by default
  - Enable extensibility
- But…some use cases require closed content models
  - E.g. warn about dropped expressions

```
:Person {
 :name  xsd:string   ;
 :knows .          *
}
```

```
:Person CLOSED {
 :name  xsd:string   ;
 :knows .          *
}
```



```
:frank :name "Frank" ;
       :knows :alice, :bob ;
       :email <mailto:frank@e.com> .
```

Try it: RDFShape

Open/Closed properties

Property values are closed by default (closed properties)

```
:Book {
 :code /isbn:[0-9X]{10}/ ;
}
```

✔
```
:item23  :code "isbn:006251587X" .
```

✘
```
:item23  :code 23 .
```

Properties can be repeated

```
:Book {
  :code  /isbn:[0-9X]{10}/ ;
  :code  /isbn:[0-9]{13}/
}
```

✔
```
:item23 :code "isbn:006251587X" ,
        :code "isbn:9780062515872" .
```

EXTRA declares properties as open

```
:Book EXTRA :code {
 :code /isbn:[0-9X]{10}/ ;
}
```

✔
```
:item23 :code "isbn:006251587X" ,
        :code 23 .
```

Try it: RDFShape

Triple expressions

"Unordered" regular expressions: Regular bag expressions

```
:Person {
  (:name  xsd:string |
   :firstName xsd:string + ;
   :lastName xsd:string
  ) ;
}
```

```
name |
firstName + ; lastName
```

```
:alice :name      "Alice Cooper" .

:bob   :firstName "Robert"       ;
       :lastName  "Smith"        .

:carol :firstName "Carol"        ;
       :lastName  "King"         ;
       :firstName "Carole"       .
```

```
:dave  :firstName "Dave"         ;
       :name      "Dave Navarro" .
```

Try it: RDFShape

Logical operators

Shape Expressions can be combined with AND, OR, NOT

Some restrictions on the use of NOT combined with recursion

```
:Book {
 :name   xsd:string ;
 :author @:Person OR @:Organization ;
}

:AudioBook @:Book AND {
 :name           MaxLength 20 ;
 :readBy         @:Person     ;
} AND NOT {
 :numberOfPages . +
}

:Person {}
:Organization {}
```

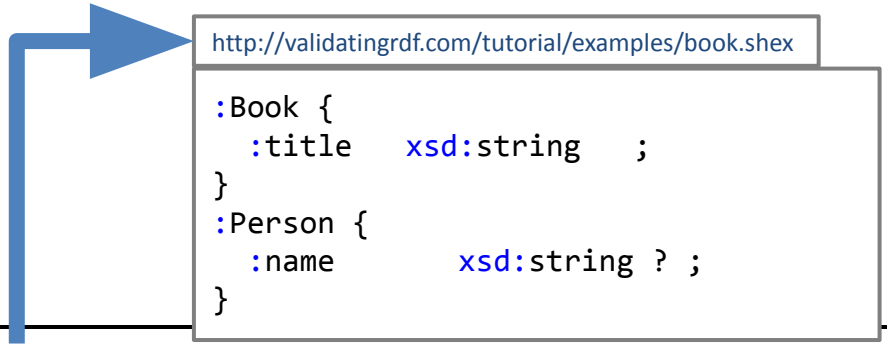
```
:item24 :name          "Weaving the Web" ;
        :author        :timbl            ;
        :readBy        :timbl            .
```

```
:item23 :name          "Weaving the Web" ;
        :author        :timbl            ;
        :numberOfPages 272               ;
        :readBy        :timbl            .
```

Try it: RDFShape

Importing schemas

`import` statement can be used to import schemas



Try it: RDFShape

Inheritance model for ShEx

`extends` allows to reuse existing shapes adding new content

Handles closed properties and shapes

```
:Book {
 :name    xsd:string ;
 :author  @:Person   ;
 :code    /isbn:[0-9]{13}/   ;
 :code    /isbn:[0-9X]{10}/
}

:LibraryBook extends @:Book {
 :code    /internal:[0-9]*/ ;
}
```

```
:item23 :name        "Weaving the Web"     ;
        :author      :timbl                ;
        :code        "isbn:006251587X"     ;
        :code        "isbn:9780062515872" ;
        :code        "internal:234"        .
```

Other features

- Multiple inheritance
- Abstract shapes

Try it: RDFShape

Machine processable annotations

They look like comments but are machine processable

```
:Book {
 :name    xsd:string // rdfs:label    "Name"@en
                     // rdfs:label    "Nombre"@es
                     // rdfs:comment  "Name of person"  ;
 :author  @:Person   // rdfs:label    "author"@en
                     // rdfs:label    "autor"@es
                     // rdfs:comment  "Book author"  ;
}
```

Other ShEx features

Machine processable annotations

Value set ranges

Language tagged values

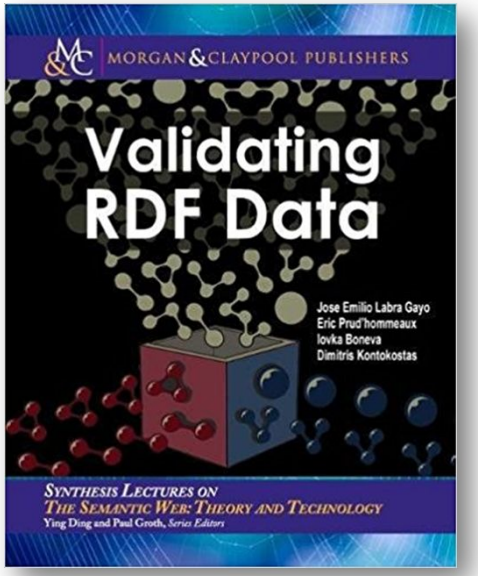
Semantic actions

Named expressions

Nested shapes

Shape maps

. . .

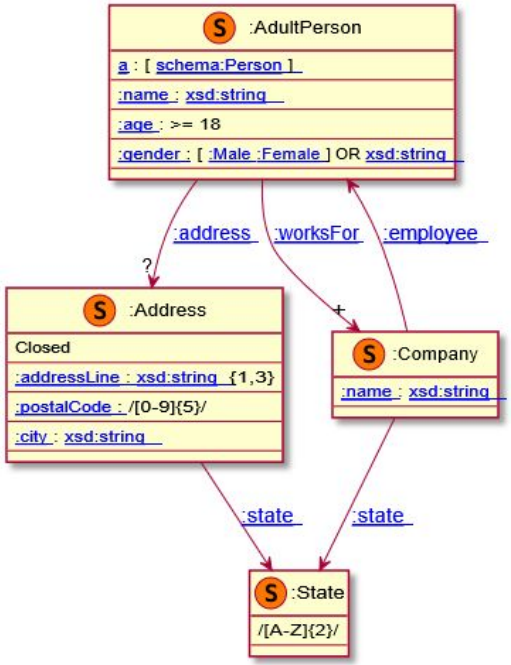


Jose E. Labra Gayo, Eric Prud'hommeaux, Iovka Boneva, Dimitris Kontokostas, *Validating RDF Data*, Synthesis Lectures on the Semantic Web, Vol. 7, No. 1, 1-328, DOI: 10.2200/S00786ED1V01Y201707WBE016, Morgan & Claypool (2018)
Online version: http://book.validatingrdf.com/

Example with more ShEx features

```
:AdultPerson EXTRA rdf:type {
 rdf:type    [ schema:Person ]              ;
 :name       xsd:string                     ;
 :age        MinInclusive 18                ;
 :gender     [:Male :Female] OR xsd:string  ;
 :address    @:Address ?                    ;
 :worksFor   @:Company +
}
:Address CLOSED {
 :addressLine xsd:string {1,3}
 :postalCode  /[0-9]{5}/
 :state       @:State
 :city        xsd:string
}
:Company {
 :name      xsd:string
 :state     @:State
 :employee  @:AdultPerson *
}
:State    /[A-Z]{2}/
```

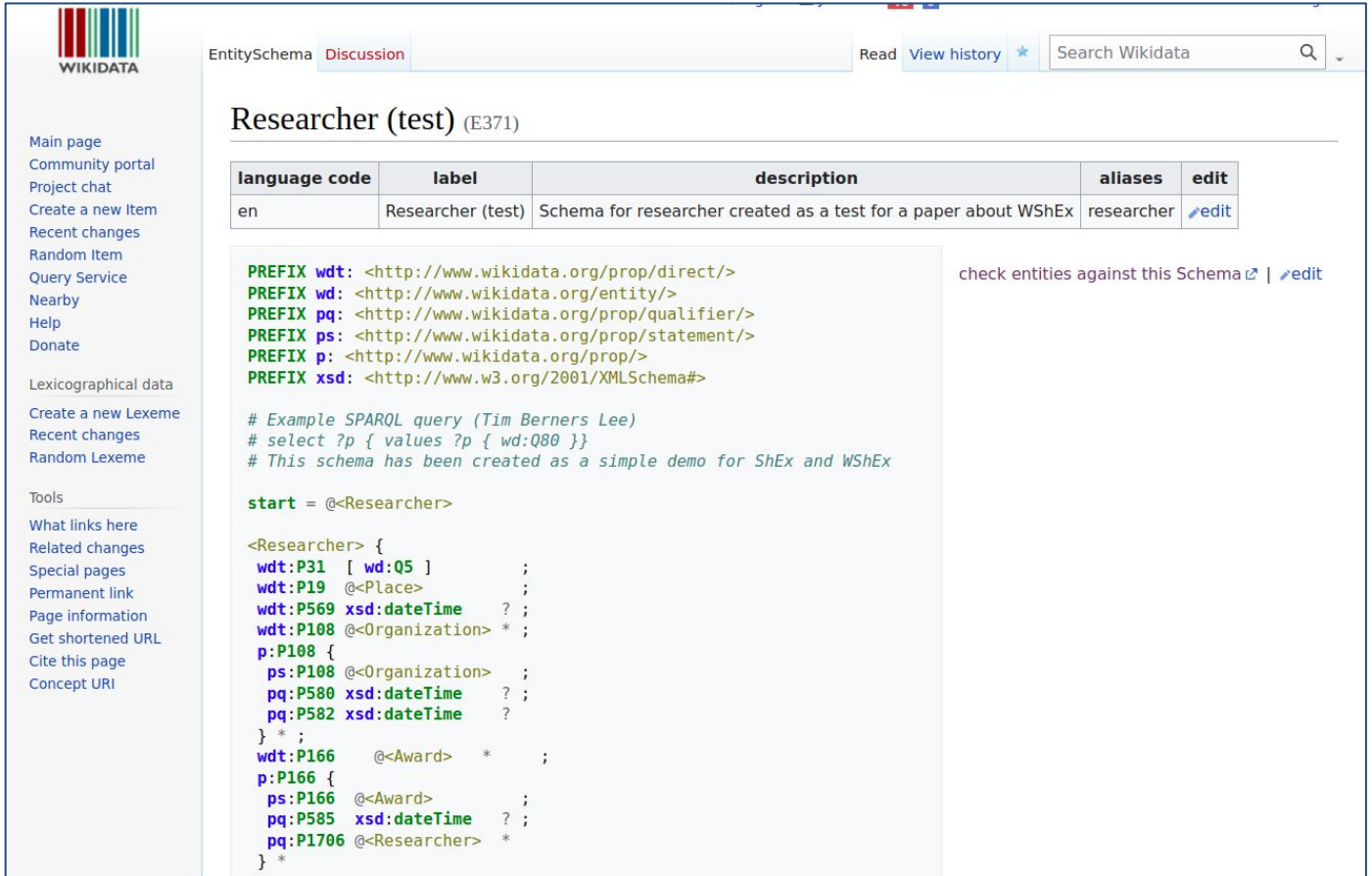
```
:alice rdf:type :Student, schema:Person ;
 :name       "Alice" ;
 :age        20 ;
 :gender     :Male ;
 :address    [
  :addressLine  "Bancroft Way" ;
  :city         "Berkeley" ;
  :postalCode   "55123" ;
  :state        "CA"
 ] ;
 :worksFor   [
  :name        "Company" ;
  :state       "CA"      ;
  :employee    :alice
] .
```



Try it: https://rdfshape.weso.es/link/16964080860

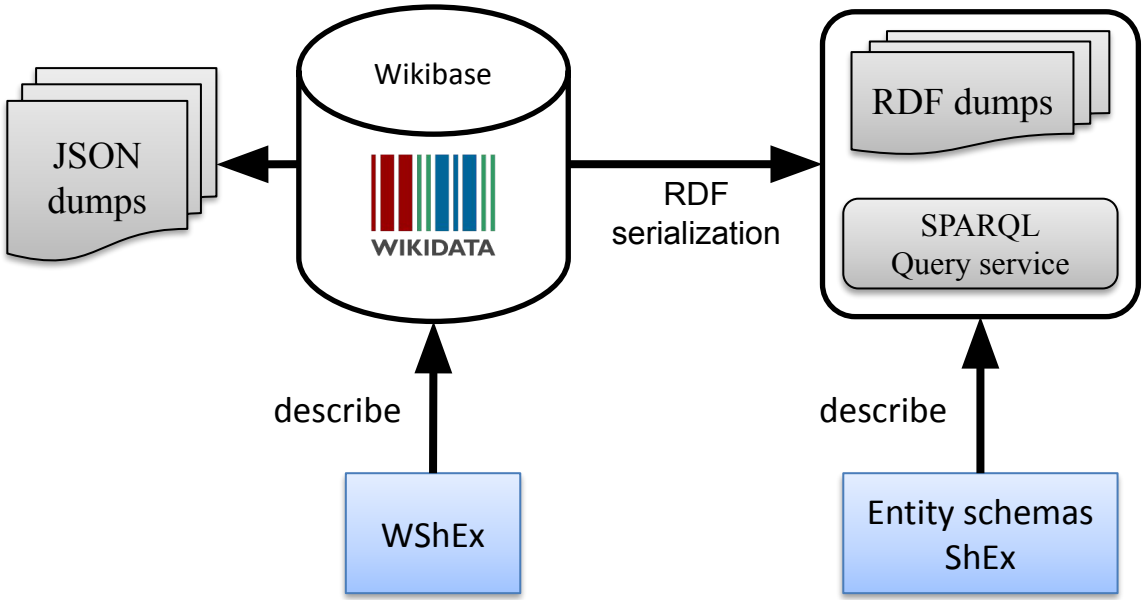
ShEx and Wikidata

Entity schemas namespace



[Example E371]

WShEx = ShEx variant for Wikibase



More info: https://www.weso.es/WShEx/

ShEx vs WShEx

Entity-schema - ShEx

```
PREFIX pq: <.../prop/qualifier/>
PREFIX ps: <.../prop/statement/>
PREFIX p: <.../prop/>
PREFIX wdt: <.../prop/direct/>
PREFIX wd: <.../entity/>
PREFIX xsd: <...XMLSchema#>

<Researcher> {
 wdt:P31   [ wd:Q5 ]          ;
 wdt:P166  @<Award>          * ;
 p:P166 { ps:P166  @<Award>          ;
          pq:P585  xsd:dateTime    ? ;
          pq:P1706 @<Researcher>   *
 } *
}
<Award> { wdt:P17 @<Country> ? }
<Country> {}
```

WShEx

```
PREFIX : <.../entity/>

<Researcher> {
 :P31   [ :Q5 ]     ;
 :P166  @<Award> {| :P585  Time           ? ,
                    :P1706 @<Researcher> ?
                 |} *
}
<Award>   { :P17 @<Country> }
<Country> {}
```

More info: https://www.weso.es/WShEx/

Agenda

Preliminaries: RDF, ShEx, Entity schemas,...

Tools for entity schemas

Exercise creating an entity schema

Some applications of entity schemas

Discussion

Tools for entity schemas

Wikidata

ShEx-simple tool

YaSHE

Wikishape

Command line tools

Wikidata

+ Edit entity schemas
+ Validate entity schemas
- No autocompletion
- Validator is a fork of ShEx simple, may have different behaviour

ShEx-simple tool

Available at: https://shex-simple.toolforge.org

Or at rawgit

+ I would call it the reference implementation of ShEx
- Not very well announced
- UI may be a bit old-fashion

YaSHE

It is deployed here: https://www.weso.es/YASHE/

+ Nice editor for entity schemas
+ Auto-completion
+ Online tool (javascript) that can be embedded in other systems
- Doesn't automatically store the schemas in Wikidata

Note: Jetbrains seems to have another nice plugin:
https://plugins.jetbrains.com/plugin/13838-rdf-and-sparql
(I didn't try it yet)

Wikishape

Online tool for wikidata, available here: https://wikishape.weso.es/

+ Supports entity schemas
+ Online tool (doesn’t need to be installed)
- Integrated validator is work in progress
- Passively maintained

Command line tools

ShEx validators: shex.js, PyShEx,

wb

+ Wikidata and wikibase tool
+ Available at: https://www.weso.es/wb/
- Implemented in Scala
- Not actively maintained: I am planning to rewrite it in Rust

ShEx-rs

- https://github.com/weso/shex-rs
- New implementation of ShEx in Rust
    - Work-in-progress
    - Not finished yet

Agenda

Preliminaries: RDF, ShEx, Entity schemas, …
Tools for entity schemas

Exercise creating an entity schema
Some applications of entity schemas
Discussion

Create an entity schema

Link: https://www.wikidata.org/wiki/Special:NewEntitySchema

See existing ones: https://www.wikidata.org/wiki/EntitySchema:E187

Tips to create entity schemas

Understand the difference between the

- Item: Q80
- Item's data model (JSON)
- Items RDF serialization (Turtle)

Select some prototypical data

Start from examples and generalize

Some tips:

- Don't over constrain
- Add example SPARQL queries of intended items

Possibility: Infer from some existing data? (sheXer)

Other possibility: Reuse and import other schemas

Example: https://www.wikidata.org/wiki/EntitySchema:E69

Agenda

Preliminaries: RDF, ShEx, entity schemas, …
Tools for entity schemas
Exercise creating an entity schema
Some applications of entity schemas
Discussion

Applications of entity schemas

- Directory of entity schemas
  - https://www.wikidata.org/wiki/Wikidata:Database_reports/EntitySchema_directory
  - Towards an ecosystem of entity schemas
- Validating data and improving the quality of wikidata
- Other applications
  - Subsetting
  - Form generation
  - …

Using entity schemas for Wikidata subsetting

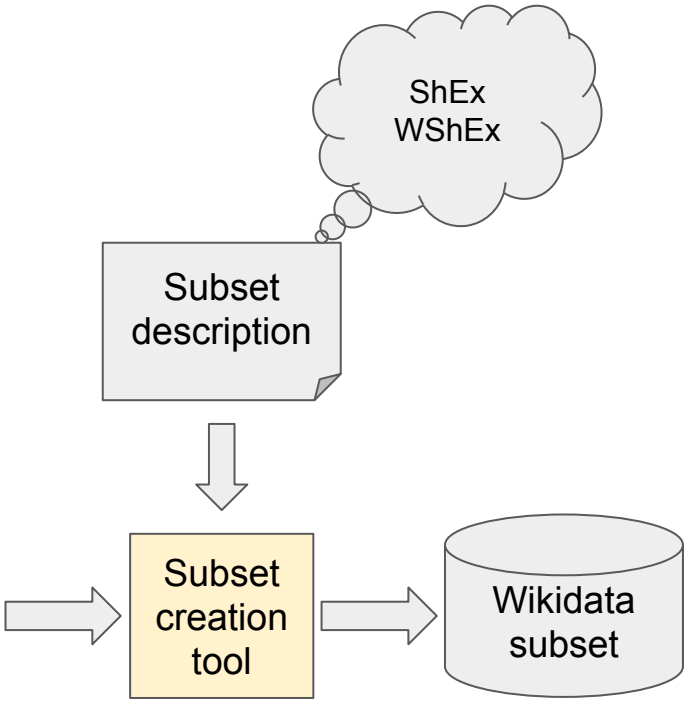
Entity schemas can be the input of Wikidata subsetting tools

They describe the content of the subset

More info:

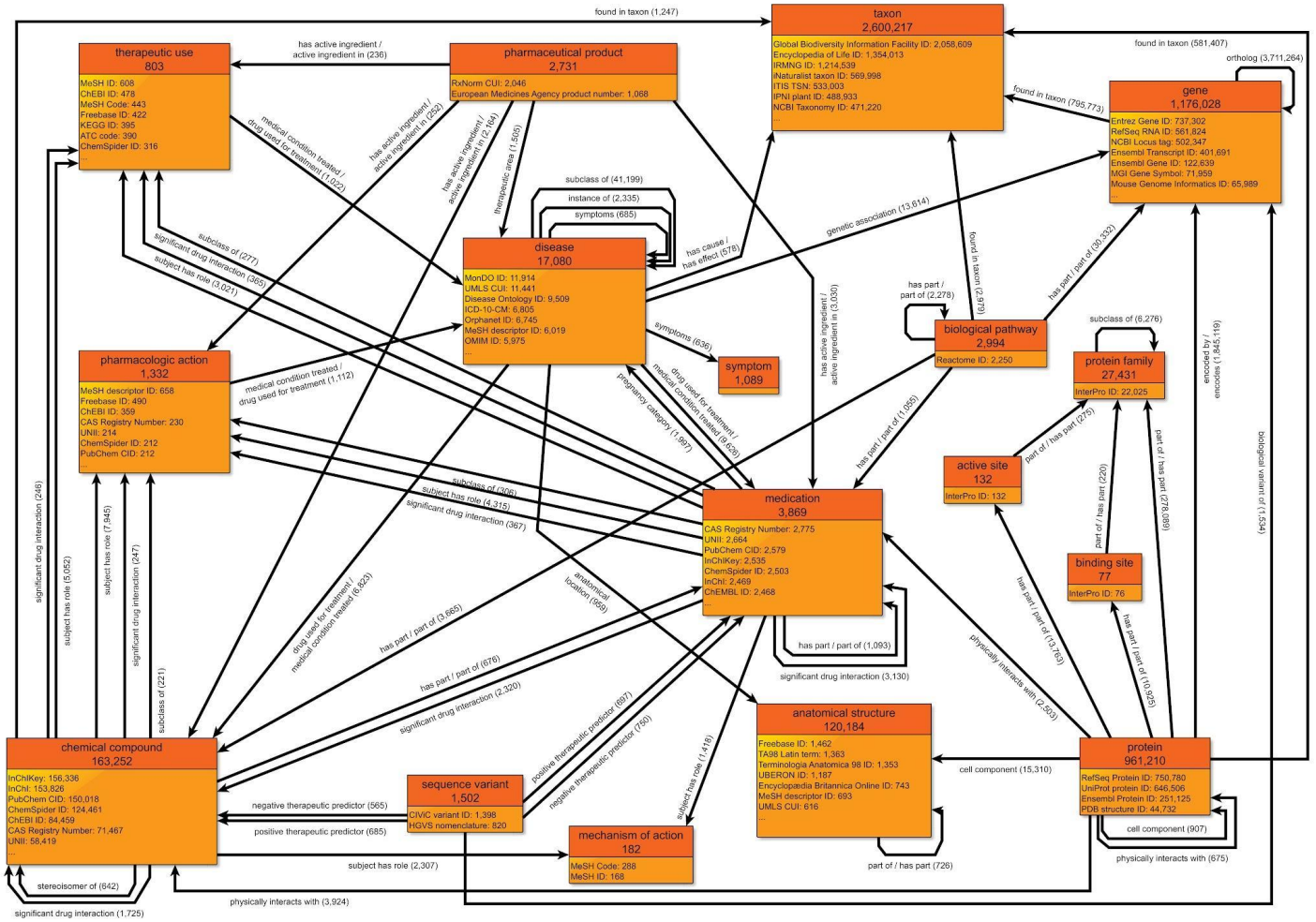
Wikidata subsetting: approaches, tools and evaluation, S. Hosseini, J. Labra, A. Waagmeester, A. Ammar, C. González, D. Slenter, S. Ui-Hasan, E. Willighagen, F. McNeill, A. Gray, accepted at Semantic Web Journal

Problem statement



GeneWiki project

Data model



GeneWiki subset

GeneWiki experiments

- ShEx schema (E258)

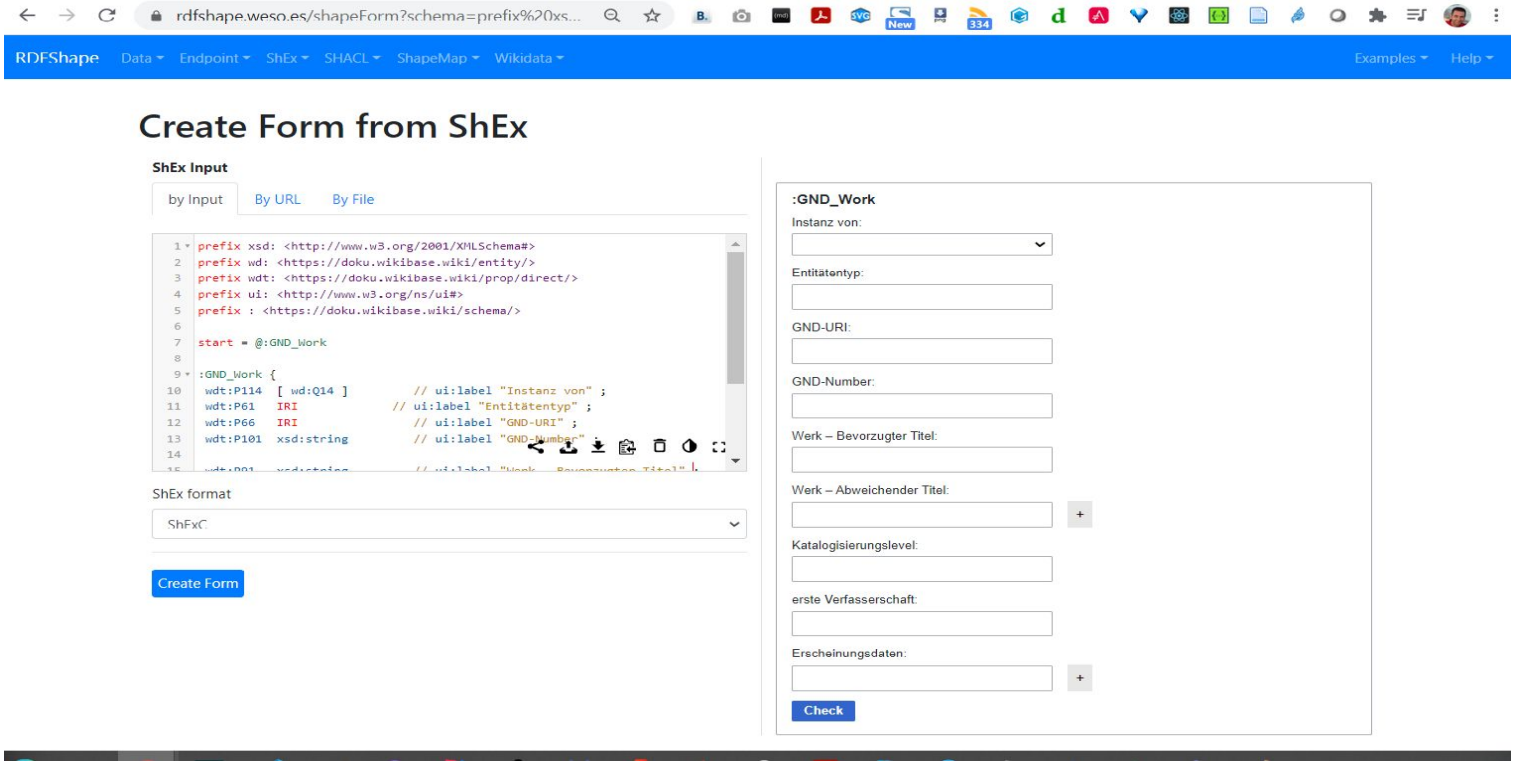
```
start= @:active_site OR
       @:anatomical_structure OR
       . . .
       @:gene OR
       . . .

:active_site EXTRA wdt:P31 {
  rdfs:label [ @en ] ;
  wdt:P31  [ wd:Q423026 ]      ;
  wdt:P361 @:protein_family * ;
  wdt:P527 @:protein_family * ;
}
. . .
:gene EXTRA wdt:P31 {
  rdfs:label [ @en ] ;
  wdt:P31   [ wd:Q7187 ];
  wdt:P684  @:gene *    ;    # ortholog (P684)
  wdt:P2293 @:disease * ;    # genetic association (P2293)
  wdt:P703  @:taxon *   ;    # found in taxon (P703)
  wdt:P1057 @:chromosome * ; # chromosome (P1057)
  wdt:P682  @:biological_process * ; # biological process (P682)
  wdt:P688  @:protein * ;    # encodes (P688)
}
. . .
```

Results about GeneWiki experiment

Class	2015	2016	2017	2018	2019	2020	2021	2022	Wikidata
active_site	0	0	132	132	132	132	132	132	132
anatomical_structure	4	62	470	483	614	732	738	812	746
binding_site	0	0	76	76	76	77	77	77	76
biological_pathway	0	0	425	2754	2929	3279	3429	3486	3554
biological_process	11	12	31263	31222	42058	43417	42061	41857	42449
cellular_component	1	1	4017	4081	4239	4298	4137	4139	4211
chemical_compound	19144	21128	156718	157018	157685	1050488	1201719	1245041	1249719
chromosome	0	0	149	152	432	9167	9224	9223	9224
disease	124	931	9578	9926	11439	13197	5395	5607	5698
gene	17	20	679372	677836	811574	1196193	1196334	1211506	Timed-Out
medication	46	2127	2459	2472	2699	3210	3336	3424	3450
molecular_function	0	0	9413	9801	11258	11226	10940	10898	11246
pharmaceutical_product	0	0	1067	1067	2725	2754	2759	2774	2784
protein_domain	2	3	9581	8847	9348	10770	11274	11709	11736
protein_family	0	212	20912	20632	22240	22170	23277	24204	24266
protein	118	166	450785	487781	579979	980520	985755	988099	Timed-Out
sequence_variant	0	0	1411	918	774	724	695	686	686
supersecondary_structure	0	0	687	687	688	688	694	696	696
symptom	16	235	273	283	328	366	319	335	343
taxon	1920049	2121404	2213907	2318731	2492613	2769303	2929068	3478871	3491430

Generating forms from ShEx (entity schemas)

Prototypes based on UI ontology: ShEx-forms (Eric), shapeForms (by WESO)



Agenda

Preliminaries: RDF, ShEx, entity schemas, …
Tools for entity schemas
Exercise creating an entity schema
Some applications of entity schemas

Discussion

Topics for discussion

- Improve Tools and User experience
    - Error messages
    - Validation/repairing/reporting experience
    - Integrate YaSHE in Wikidata?
- Entity schemas ecosystem
    - What is the role of an entity schema?
        - Descriptive vs prescriptive data models
        - Description vs validation vs Integrity constraints
- ShEx language features
- Generate schemas from existing data
    - sheXer

End of presentation