Shapes applications and tools

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

Contents

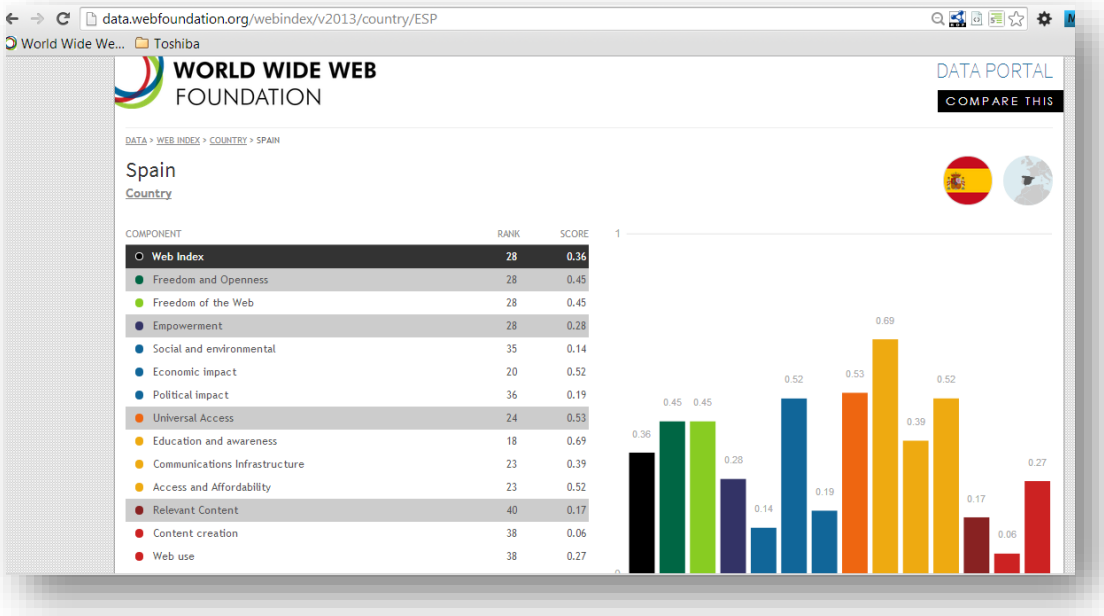
Data portals

In 2013, at WESO, we were hired to develop some data portals

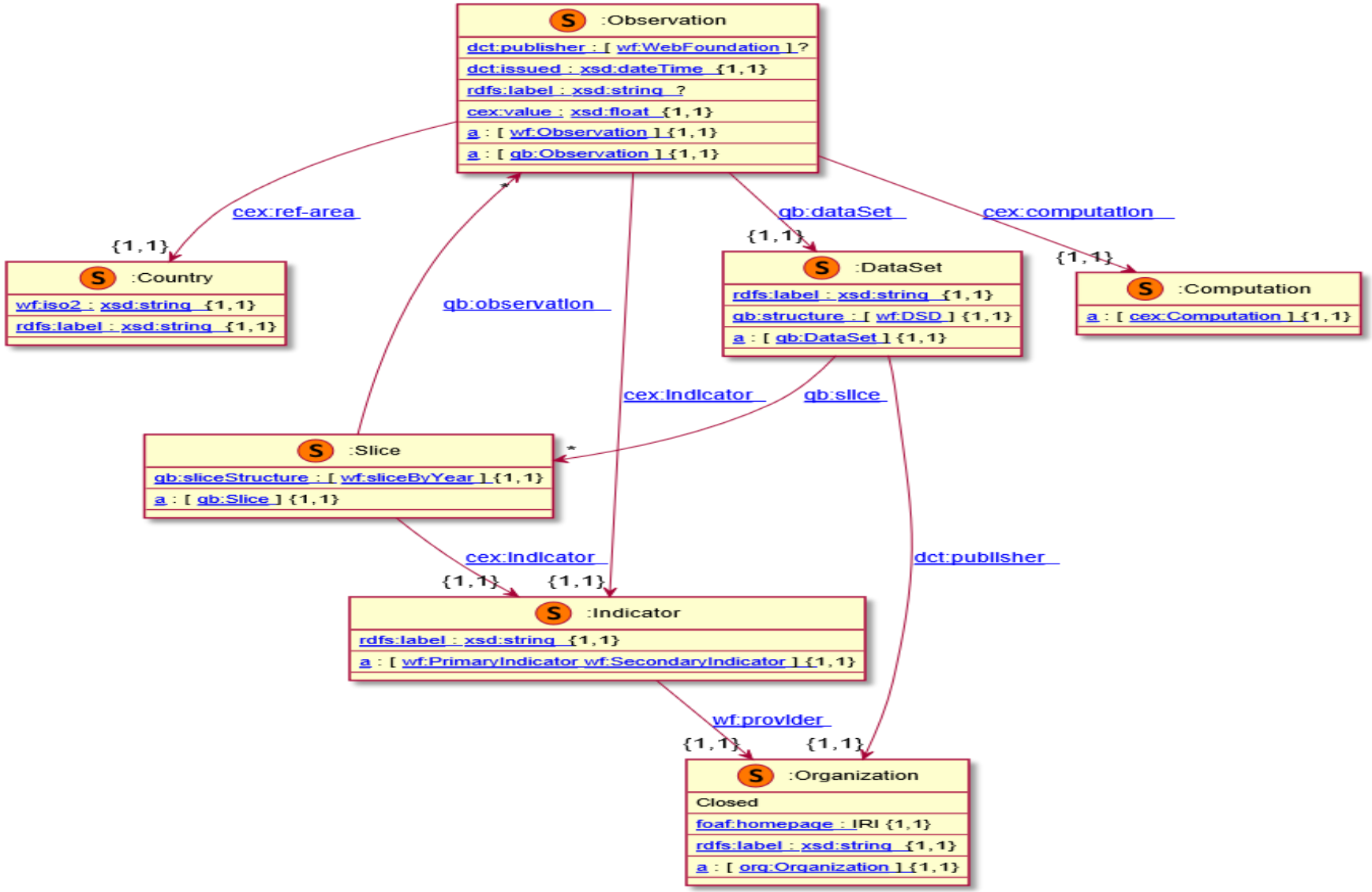
Examples: WebIndex (Web Foundation)

One of the first applications of ShEx

- Measure WWW's contribution to development and human rights by country
- Developed by the Web Foundation
- Content of data portal = statistical observations
- We employed RDF Data Cube vocabulary (qb:Observation)



Simplified WebIndex data model



Lessons learnt from ShEx usage at WebIndex

1. Documentation of linked data portal

   Human-readable, machine processable

   http://weso.github.io/wiDoc

2. Team communication

   Communicate the developers which shapes they had to generate

3. Validation

   For example: check if a value of type `qb:Observation` had shape <Observation>

4. Reuse

   Another data portal was later developed for http://landportal.org base on observations

   Easy to reuse and adapt the data model

   Same types (`qb:Observation`) but different structure

http://weso.github.io/landportalDoc/data/

Wikidata and wikibase

In May, 2019, Wikidata announced ShEx adoption

New namespace for schemas

Example:

https://www.wikidata.org/wiki/EntitySchema:E2

Wikibase also contains entity schemas

Online demo: wikishape

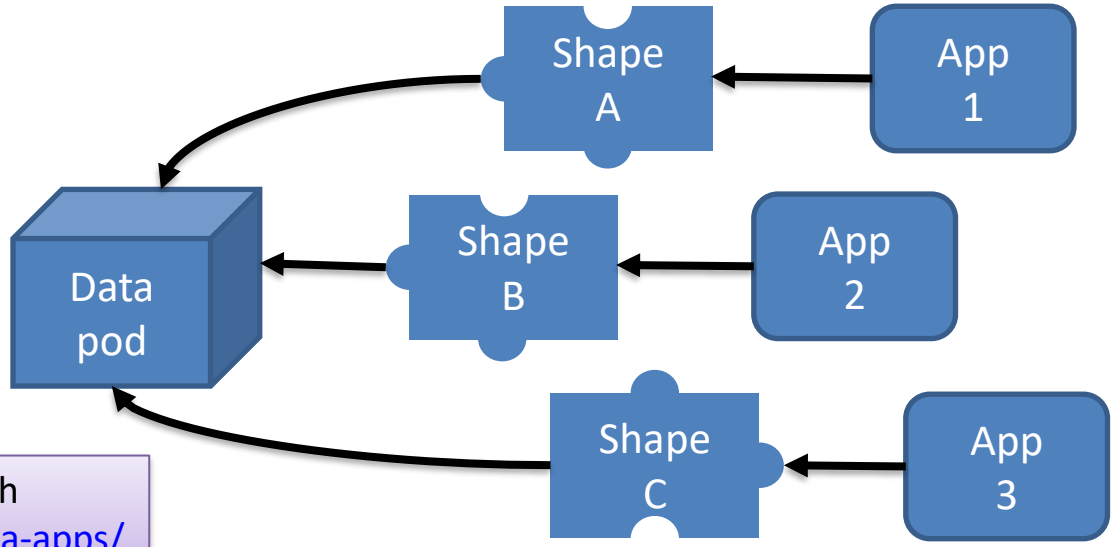
Solid project

SOLID (SOcial Linked Data): Promoted by Tim Berners-Lee

Goal: Re-decentralize the Web

- Separate data from apps
- Give users more control about their data
- Internally using linked data & RDF

Shapes needed for interoperability



"...I just can't stop thinking about shapes.", Ruben Verborgh
https://ruben.verborgh.org/blog/2019/06/17/shaping-linked-data-apps/

Other use cases

HL7 FHIR.

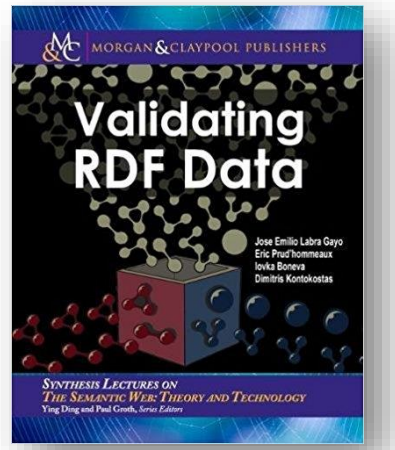
Example: https://www.hl7.org/fhir/observation.html

ELI validator

SHACL shapes obtained from Excel sheets:
https://webgate.ec.europa.eu/eli-validator/home

SHACL adoption supported by Top Quadrant

See: https://www.topquadrant.com/technology/shacl/



More info:
Chapter 6 of Validating RDF data: http://book.validatingrdf.com/bookHtml012.html

Tools: challenges and perspectives

Validating with shapes

Obtaining shapes

Other applications of shapes

Shapes ecosystems

Validating with shapes

Libraries and command line validators

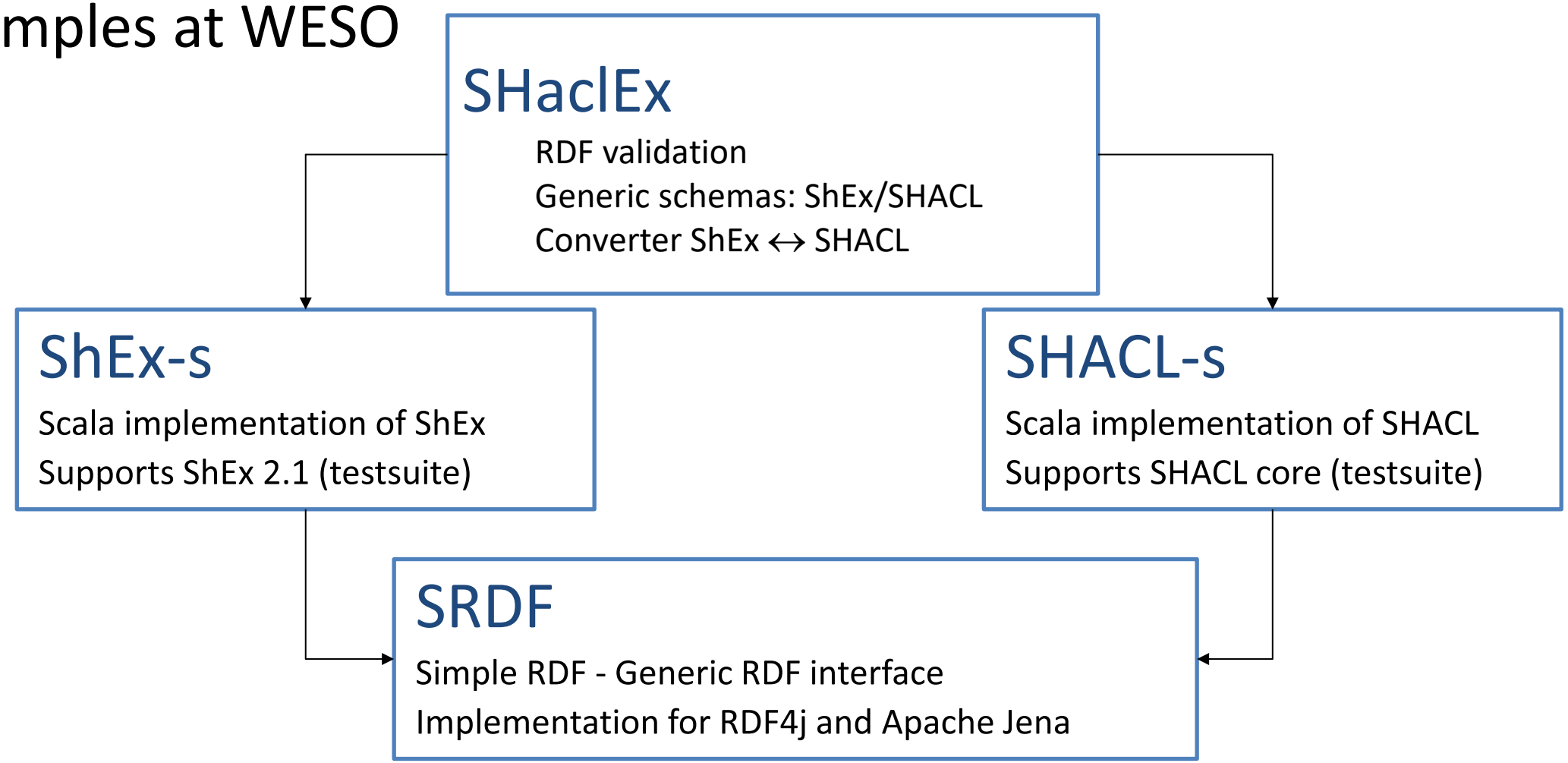
Online demos

Integrated in ontology editors

Continuous integration with Shapes

Libraries and command line validators

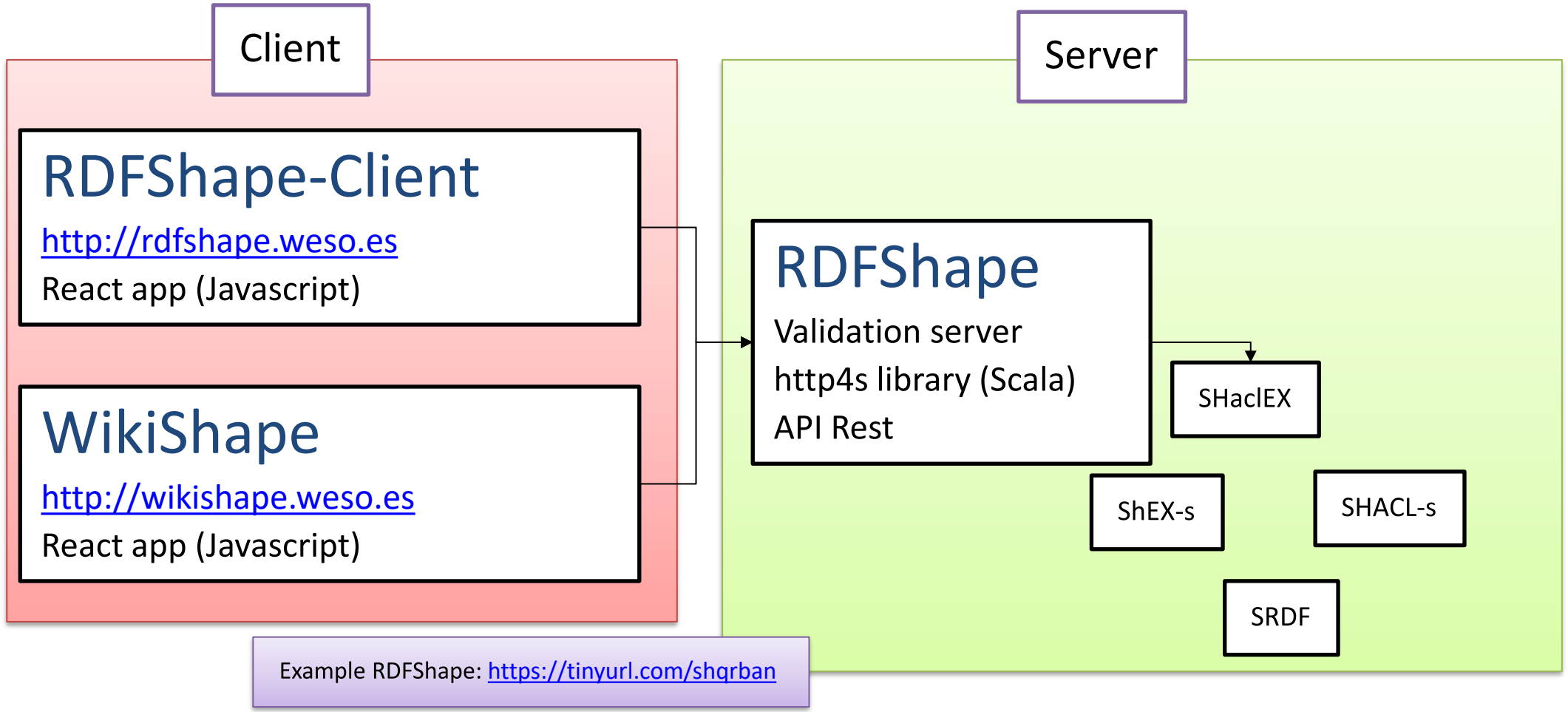
Examples at WESO



All libraries are available at: https://github.com/weso/

Online demos

Web Demos and playgrounds



Integrating shapes with other tools

TopBraid Composer

https://www.topquadrant.com/technology/shacl/

Ontology editors

SHACL plugin for protégé: https://github.com/fekaputra/shacl-plugin

ShEx plugin for protégé: https://github.com/weso/protegeShEx

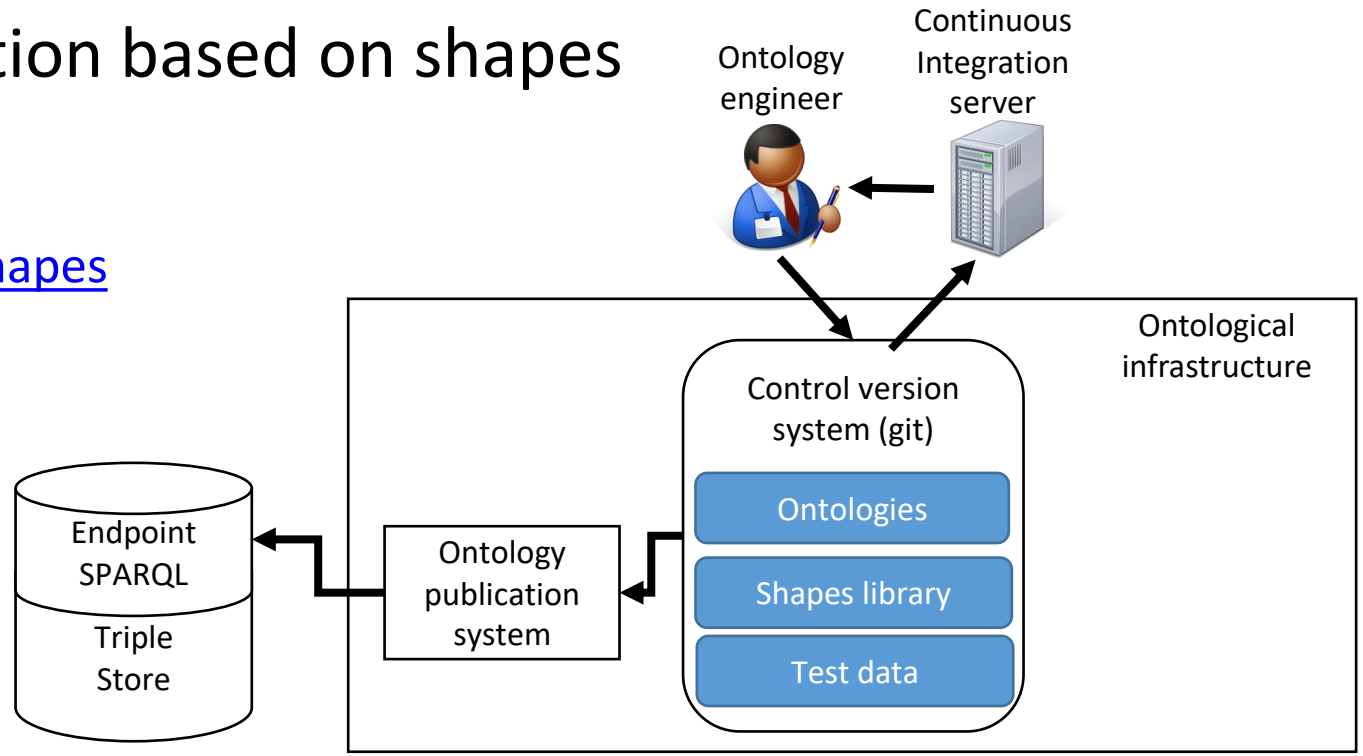
Continuous integration with Shapes

Coexistence between ontologies/shapes

- Shapes can validate the behaviour of inference systems
- Shapes pre- and post- inference
- TDD and continuous integration based on shapes

Gene Ontology Shapes:

https://github.com/geneontology/go-shapes



Continuous integration with Shapes

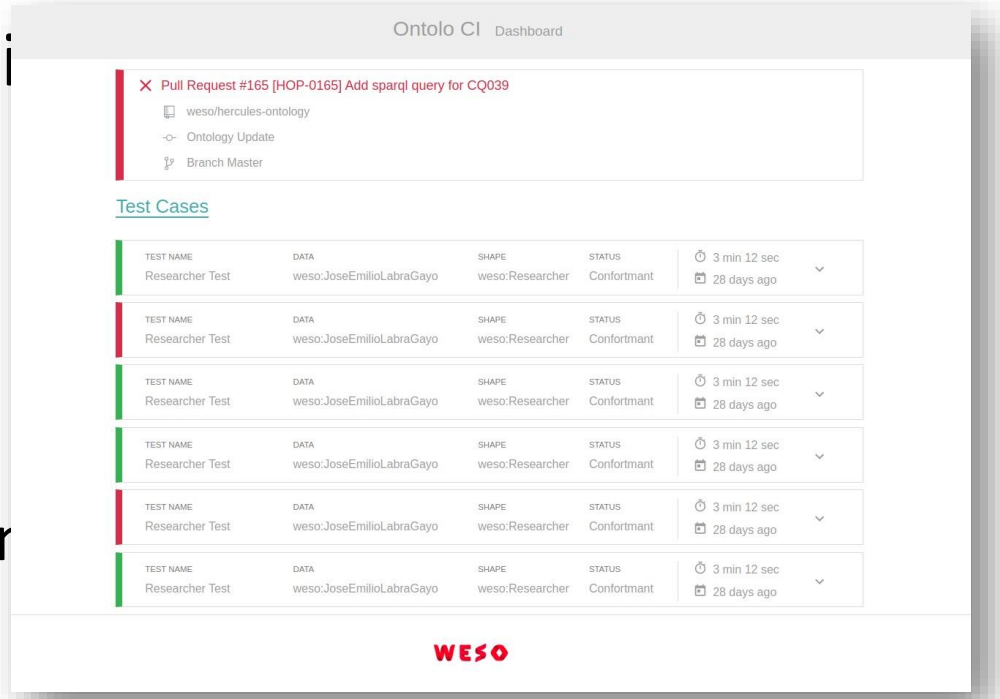
Ontolo-ci: https://github.com/weso/ontolo-ci

Developed as part of HERCULES-Ontology

Test-Driven-Development applied to Ontologi

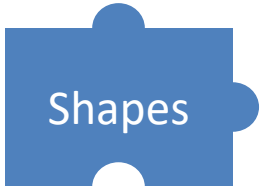
Input:

- Ontologies
- Shapes
- Test data
- Input shape map (SPARQL competency questior
- Expected result shape map



Creating shapes

- Shapes editors
  - Text-based editors
  - Visual editors and visualizers
- Obtaining shapes from...
  - Spreadsheets
  - RDF data
  - Ontologies
  - Other schemas (XML Schema)



Text-based editors

YaSHE: Forked from YASGUI: http://www.weso.es/YASHE/

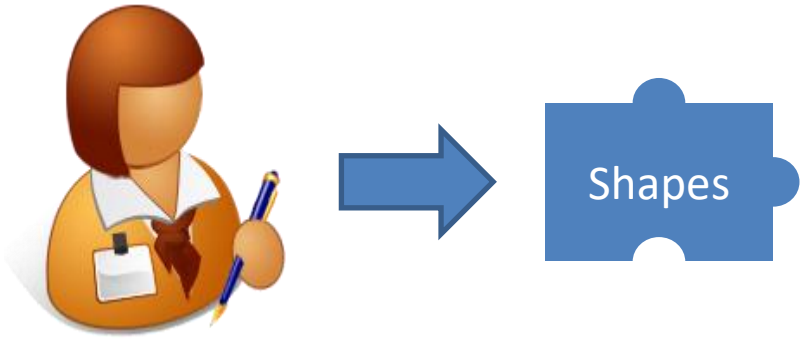
- Syntax highlighting
- Auto-completion

Shapes author tools

Top Braid Composer

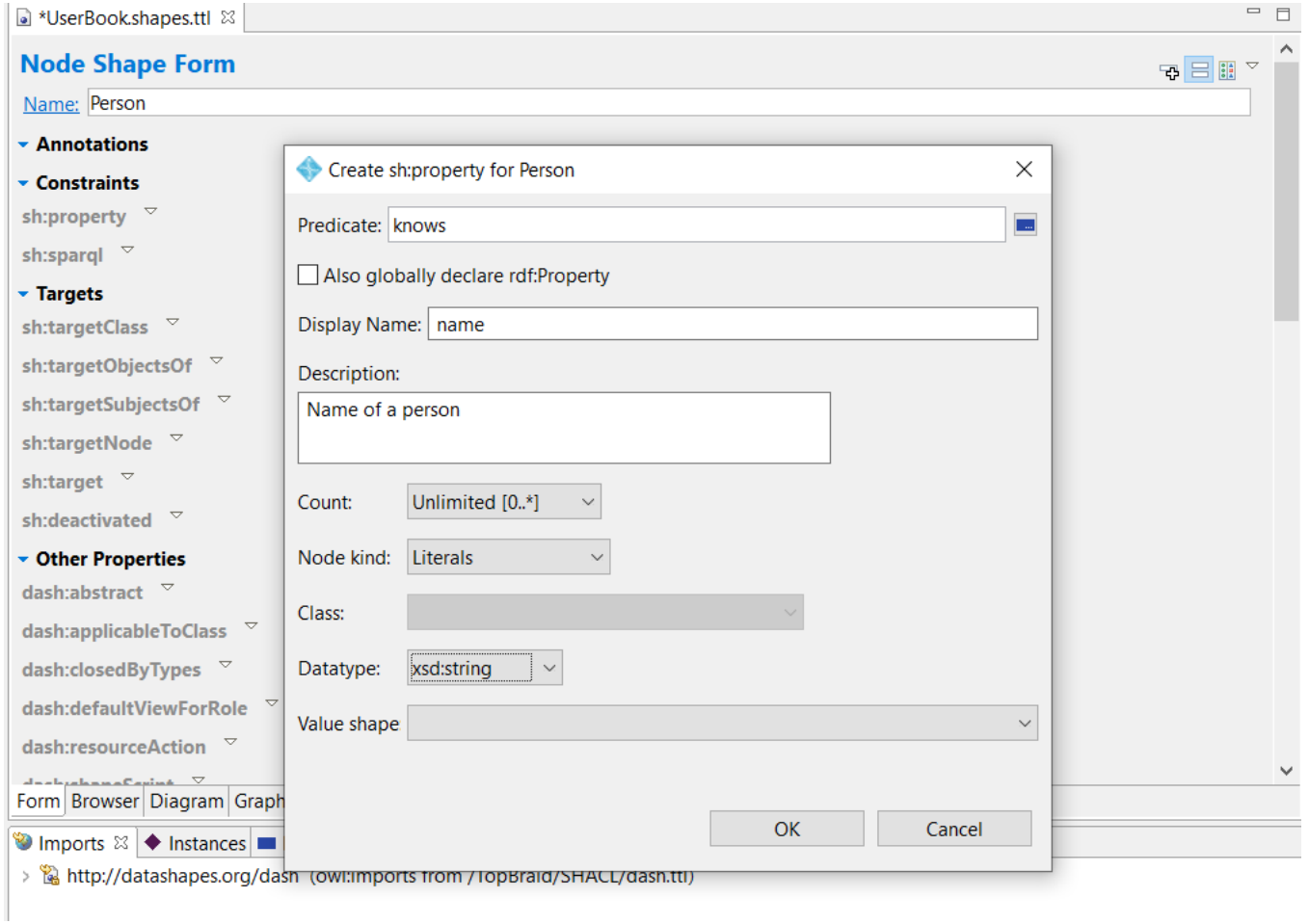
UnSHACLed

ShEx-Author



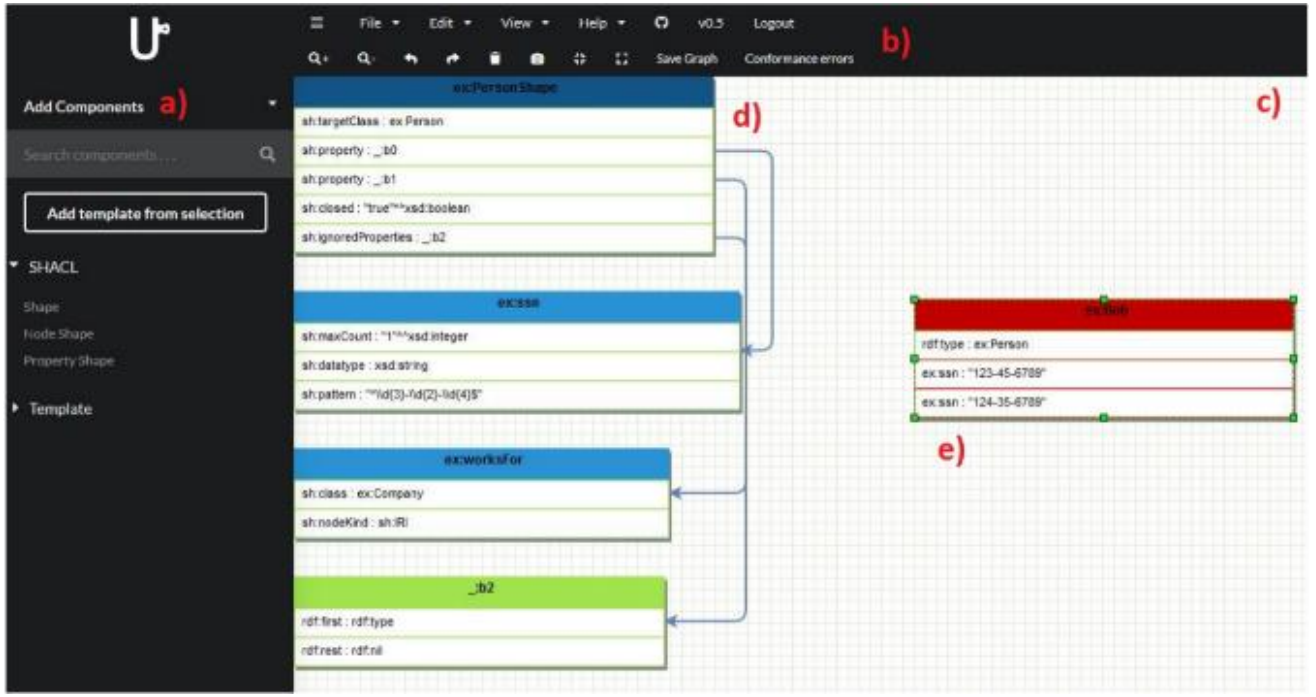
Shapes author tools: Top Braid Composer

Form based editor
Integrated with Top Braid product



Shapes author tools: UnSHACLed

Visual SHACL Editor in Javascript

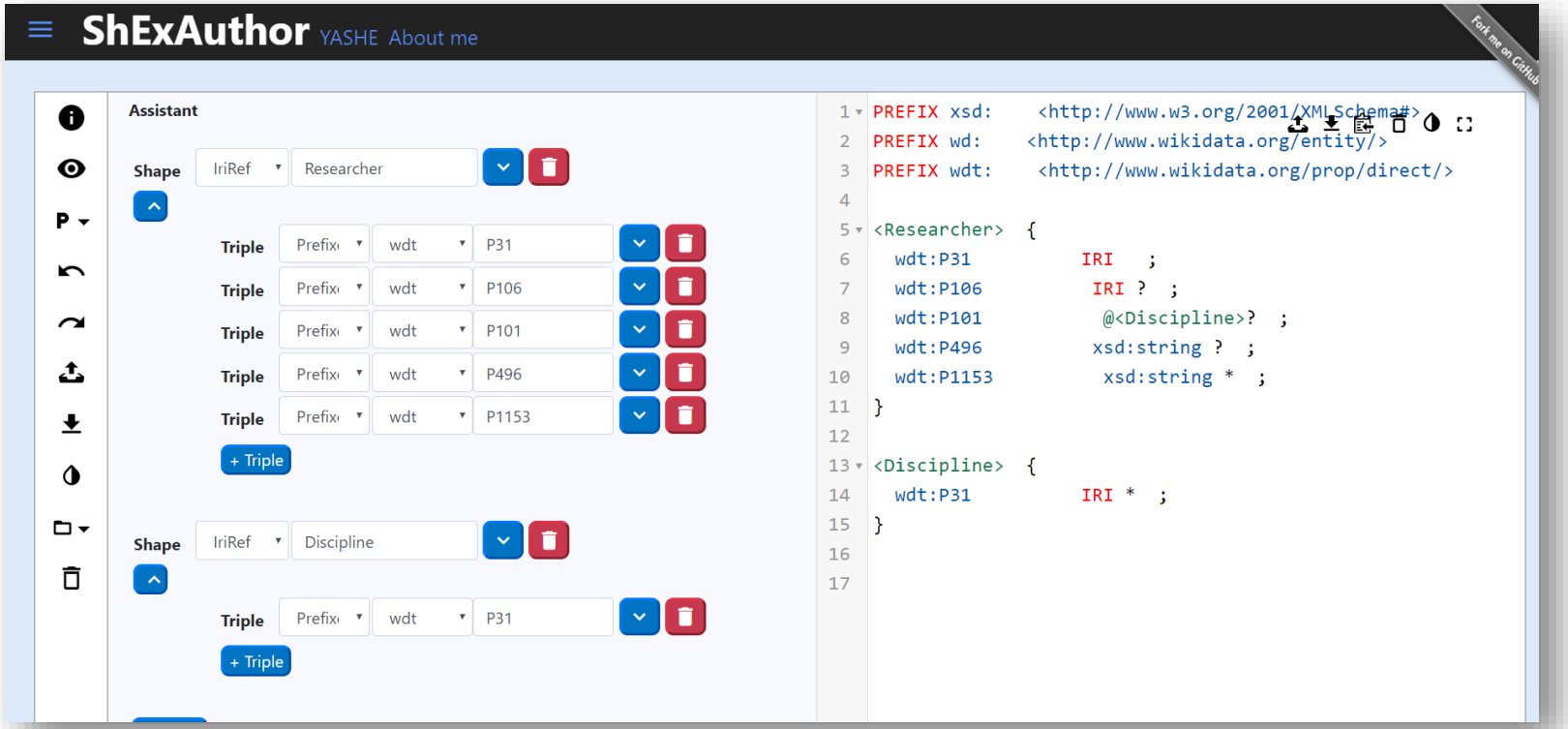


B. De Meester, P. Heyvaert, A. Dimou, and R. Verborgh, “Towards a Uniform User Interface for Editing Data Shapes,” in Proceedings of the 4th International Workshop on Visualization and Interaction for Ontologies and Linked Data, 2018, vol. 2187.

Shapes author tools: ShEx Author

ShEx-Author: Inspired by Wikidata Query Service

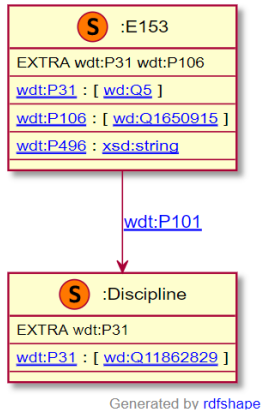
2 column: Visual one synchronized with text based

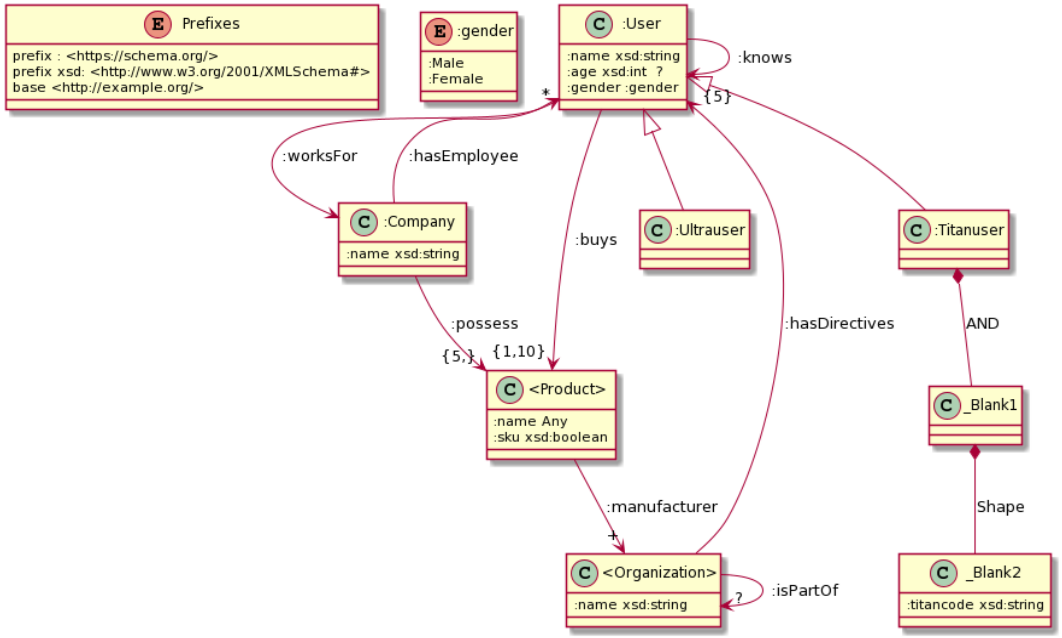


Shapes visualization

Integrated in RDFShape/Wikishape

- UMLSHaclEX UML diagrams for ShEx
- ShUMLex: Conversion to UML through XMI





Shapes from spreadsheets

SKOS-Play was used at ELI to generate SHACL shapes from Excel

ShExstatements: https://shexstatements.toolforge.org/

ShExCSV: CSV representation of Shapes

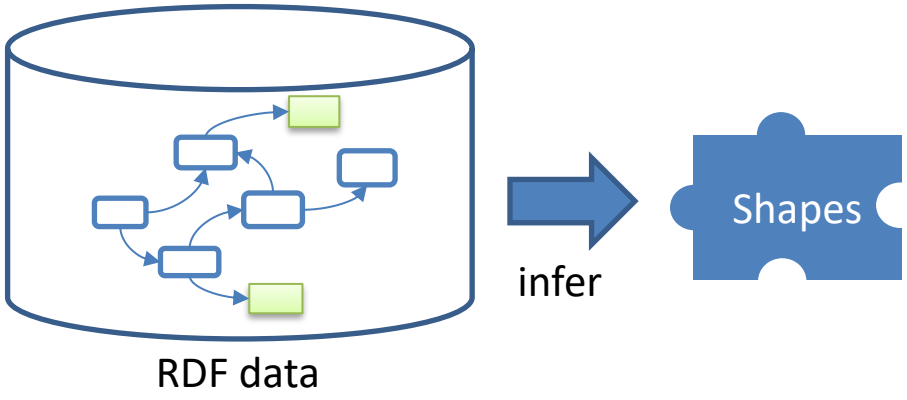
Hermes: ShExCSV processor, https://github.com/weso/hermes

Generating Shapes from RDF data

Useful use case in practice

Some prototypes

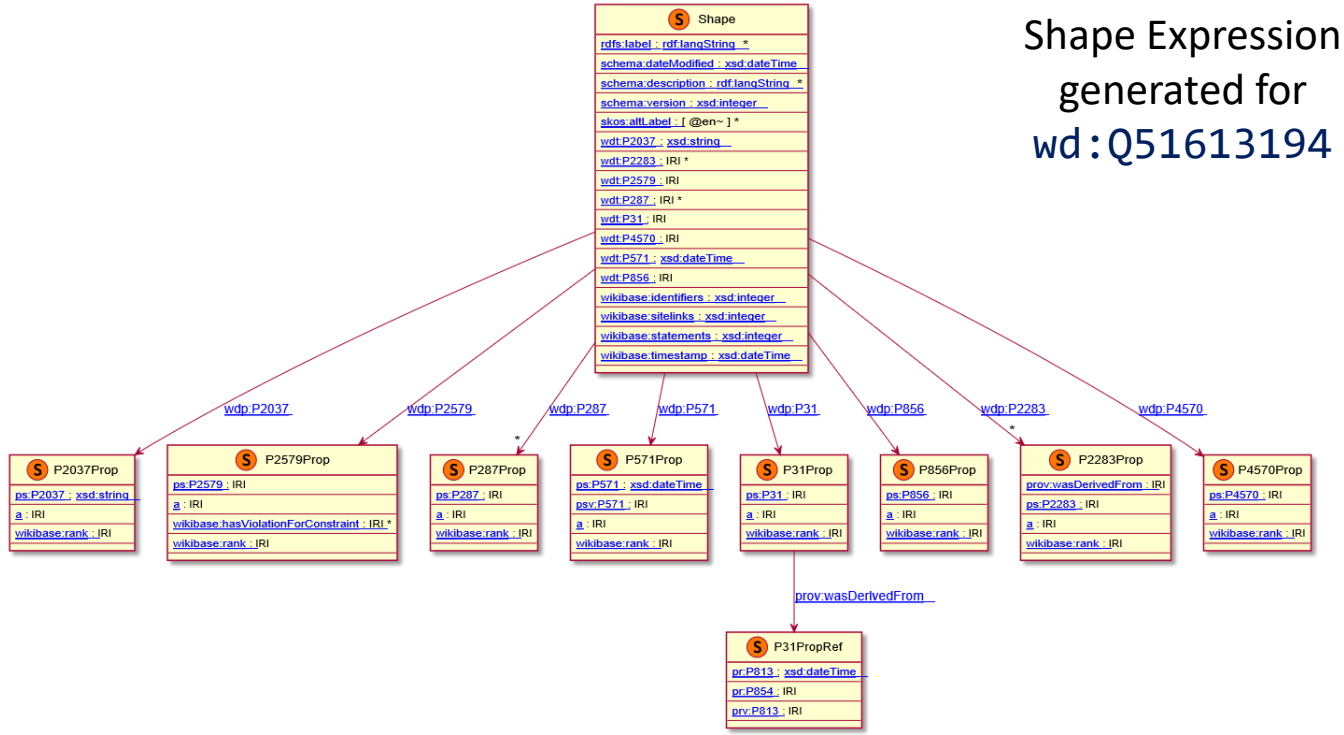
- sheXer: http://shexer.weso.es/
- RDFShape: http://rdfshape.weso.es
- ShapeDesigner: https://gitlab.inria.fr/jdusart/shexjapp



Try it with RDFShape:
https://tinyurl.com/y8pjcbyf

Shapes from data: RDFShape

RDFShape/Wikishape implement a basic prototype to derive Shapes from RDF data



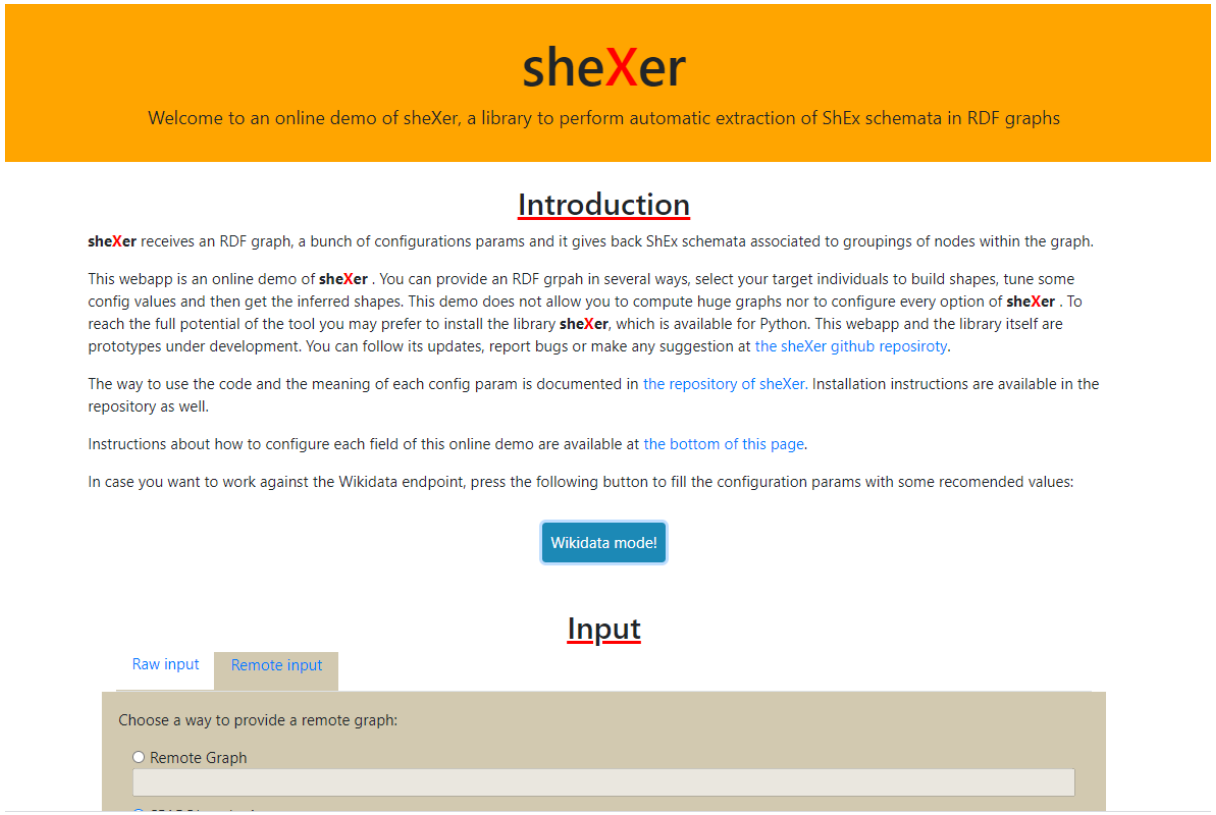
Shape Expression generated for `wd:Q51613194`

Shapes from data: sheXer

sheXer: http://shexer.weso.es/

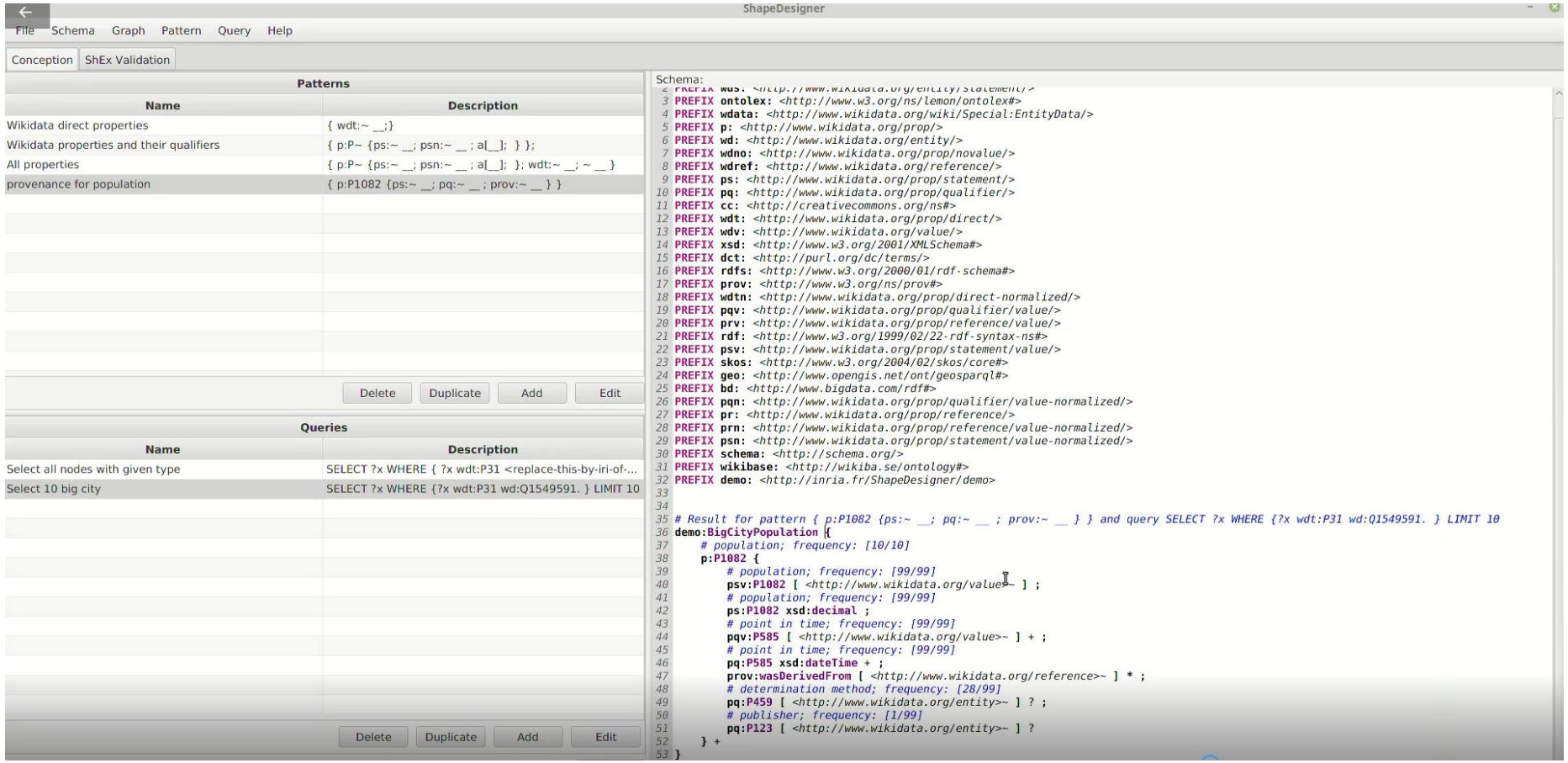
Implemented in Python

Configuration options



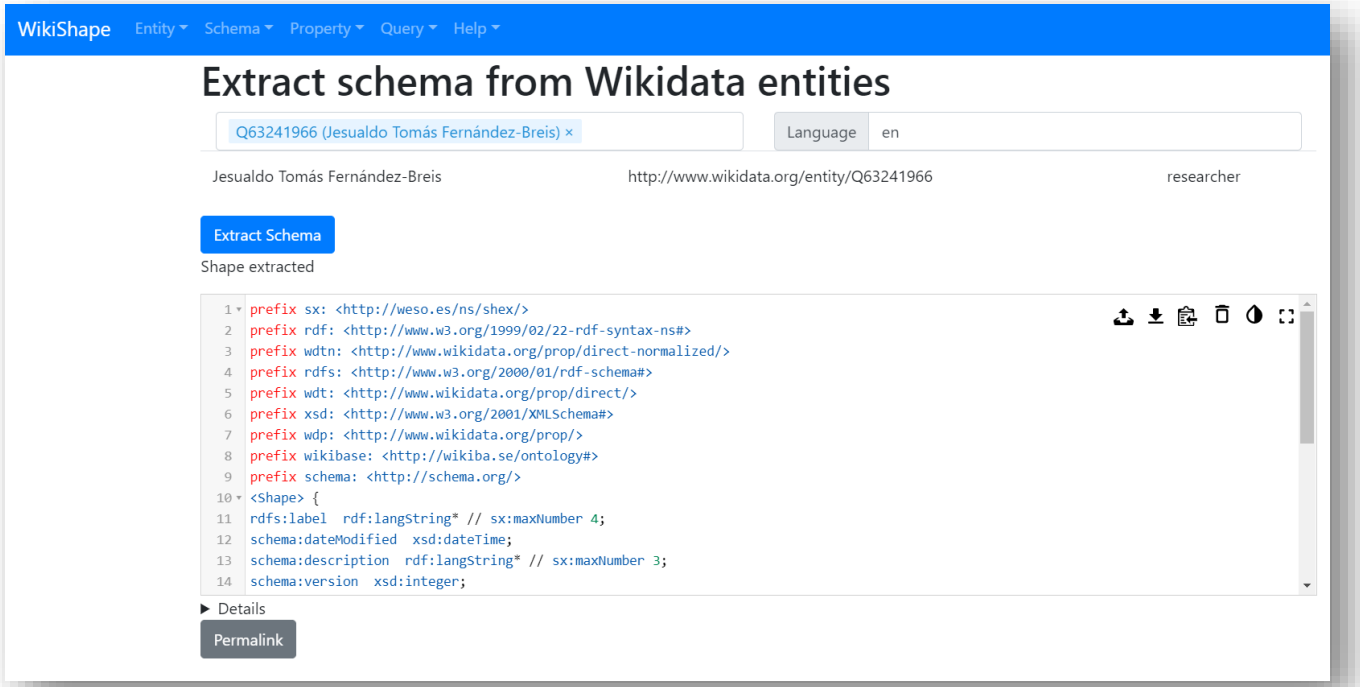
Shapes from data: ShapeDesigner

https://gitlab.inria.fr/jdusart/shexjapp



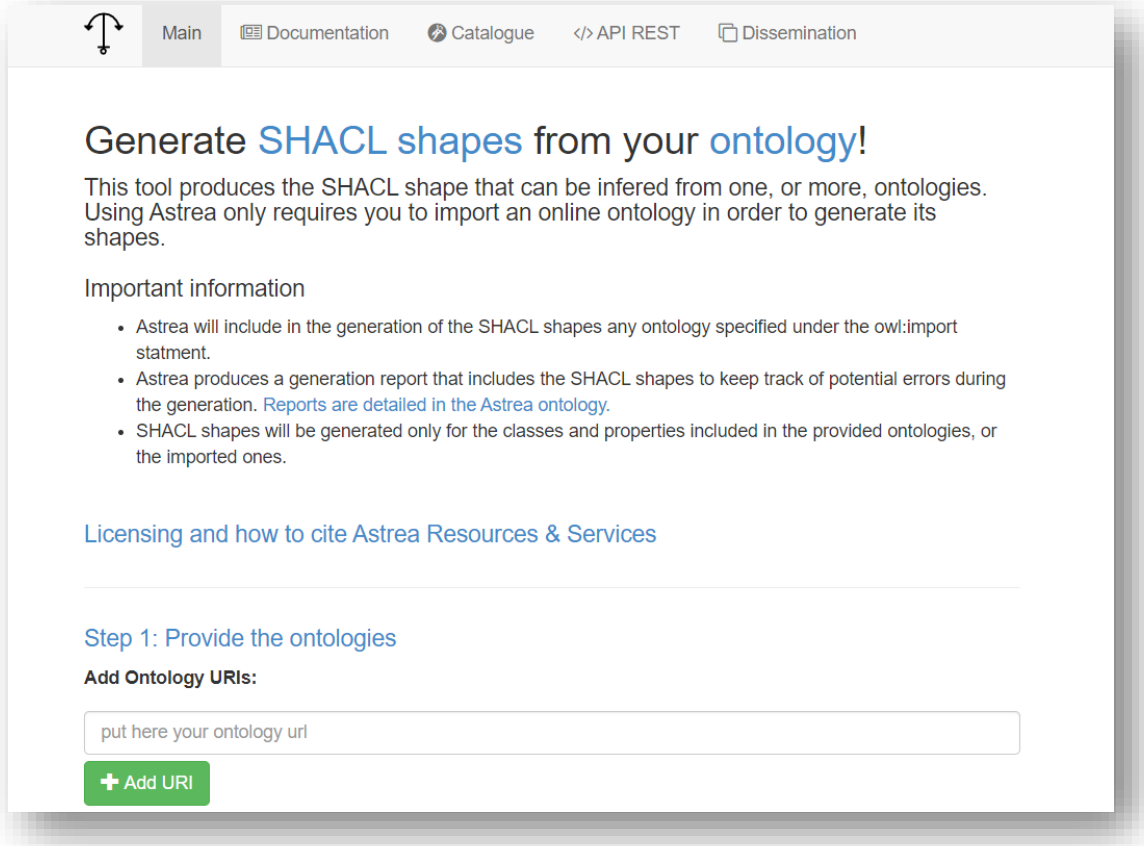
Shapes from RDF data

RDFShape allows to infer basic shapes automatically



Shapes from ontologies

- Astrea*: https://astrea.linkeddata.es/
- Generates SHACL shapes from OWL ontologies
- Mappings between ontology construct patterns to SHACL



*Cimmino, A., Fernández-Izquierdo, A., & García-Castro, R. (2020). Astrea: automatic generation of SHACL shapes from ontologies. In European Semantic Web Conference

Other uses of Shapes

UIs and shapes

Generating code from Shapes

Shapes and rules

UIs and shapes

Shapes can provide hints to generate user interfaces/forms

SHACL core defines a basic vocabulary: sh:group, sh:order, ...

ShEx annotations can also be used to define UI declarations

- Example: UI ontology annotations

UIs and Shapes: ShExPath and ShEx-Forms

ShEx Path can be used to point to parts of a ShEx schema

https://shexspec.github.io/spec/ShExPath

ShEx generated forms demo based on UI ontology:

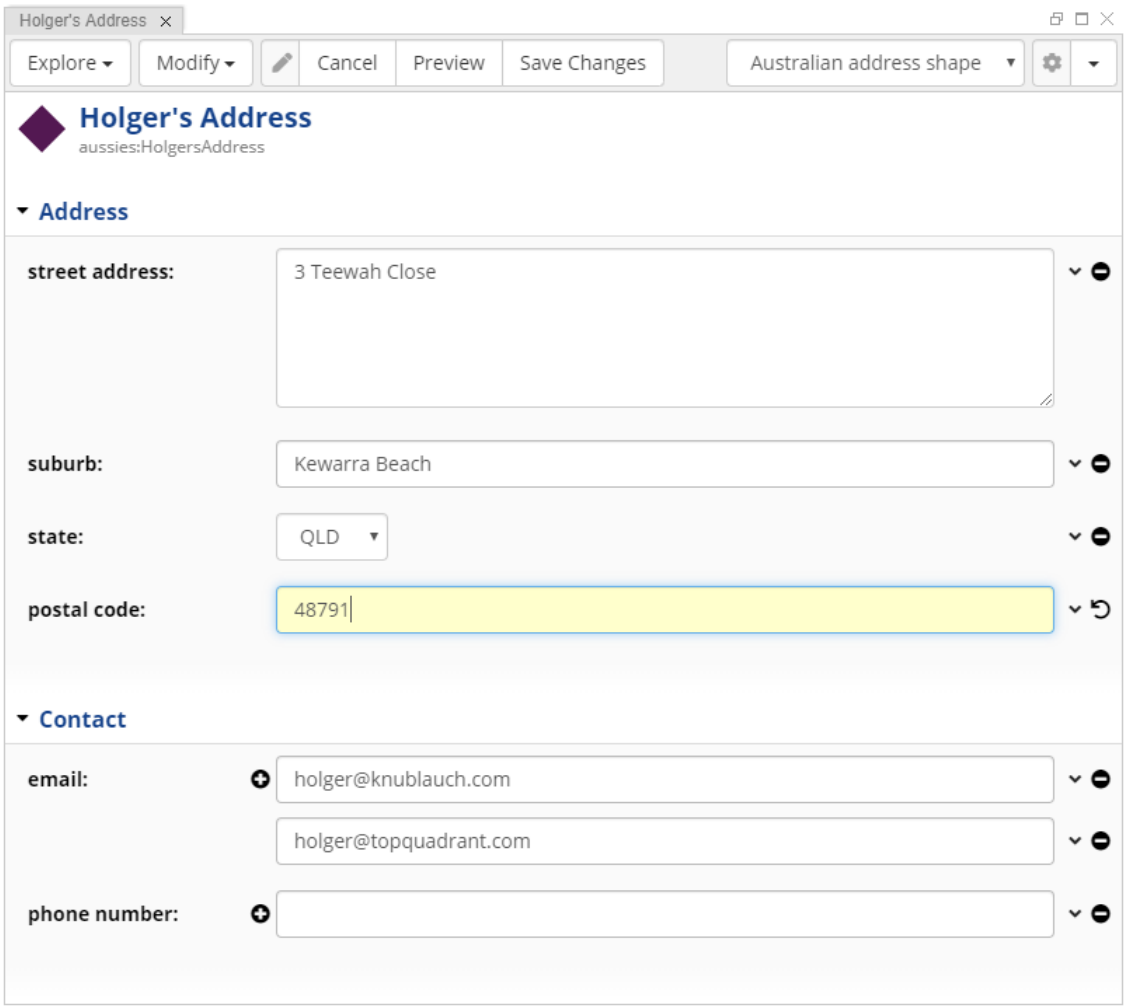
https://ericprud.github.io/shex-form/?manifestURL=examples/manifest.json

UIs and shapes: TopQuadrant

Form generation from SHACL

DASH vocabulary:

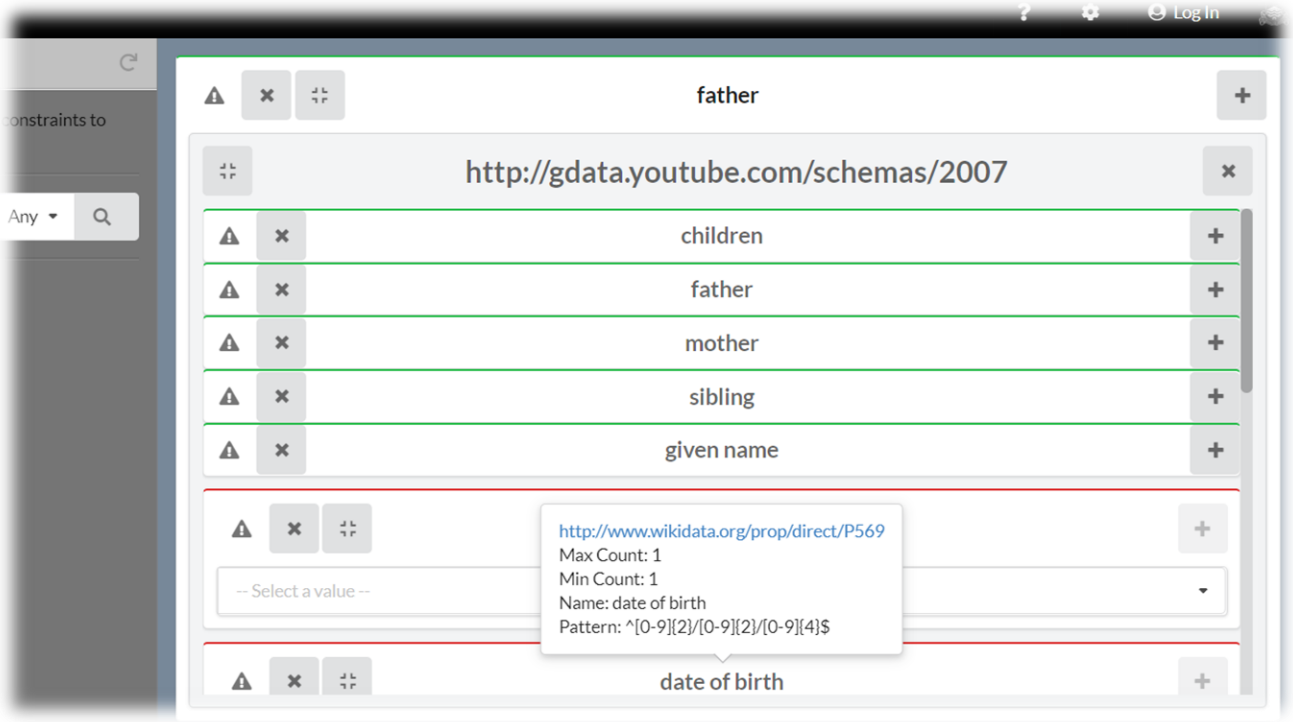
http://datashapes.org/forms.html



UIs and shapes: Schímatos

http://schimatos.org/

It will be presented at ISWC20



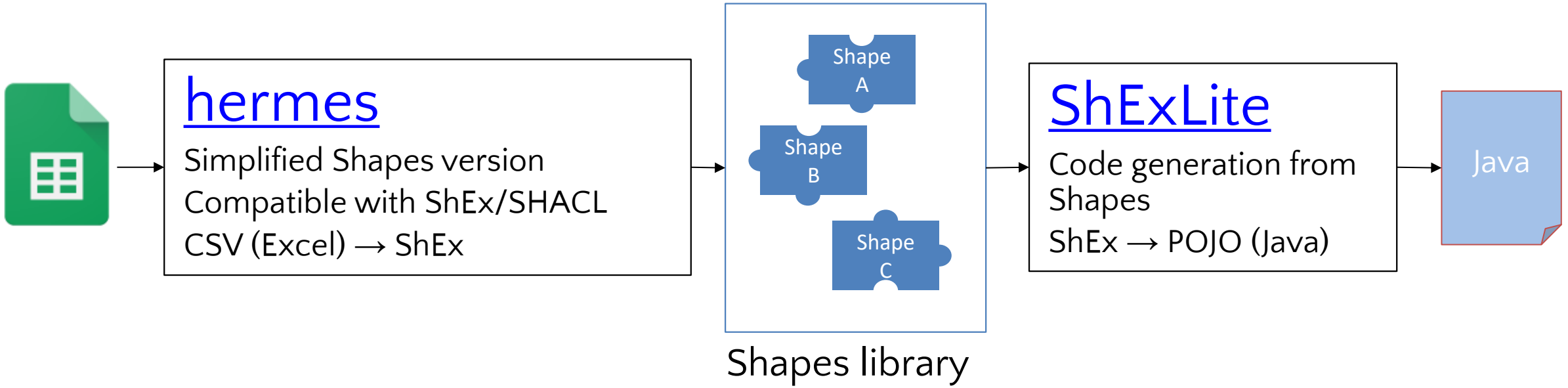
Generating code from shapes

Generate domain model from shapes

Entities (pseudo-shapes) defined with Excel (Google spreadsheets)

Shapes generation from those templates

Java code generation (POJOs) from those shapes

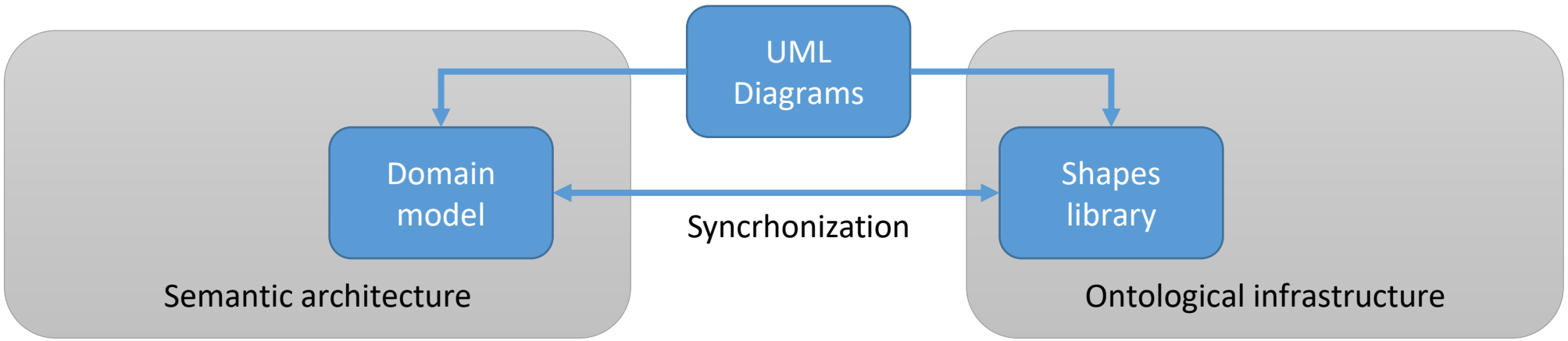


Generating code from shapes

Domain model based on Shapes

Clean architecture pattern

- Domain model as central element
- Simple classes (POJO): Plain Old Java Objects
- Shapes synchronization
- Application logic and services based on domain model



Shapes and rules

SHACL Advanced Features describes SHACL rules

```
:Rectangle a rdfs:Class, sh:NodeShape ;
  rdfs:label "Rectangle" ;
  sh:property [ sh:path :height ;
   sh:datatype xsd:integer ;
   sh:maxCount 1 ; sh:minCount 1 ;
   sh:name "height" ] ;
  sh:property [sh:path :width ;
   sh:datatype xsd:integer ;
   sh:maxCount 1 ; sh:minCount 1 ;
   sh:name "width" ; ] ;
  sh:rule [ a sh:TripleRule ;
   sh:subject sh:this ;
   sh:predicate rdf:type ;
   sh:object    :Square ;
   sh:condition :Rectangle ;
   sh:condition [
    sh:property [
     sh:path   :width ;
     sh:equals :height ;
] ; ] ; ] .
```

```
:I a :Rectangle .
:N a :Rectangle ;
   :height 2 ;
   :width  3 .
:S a :Rectangle ;
   :height 4 ;
   :width  4 .
```

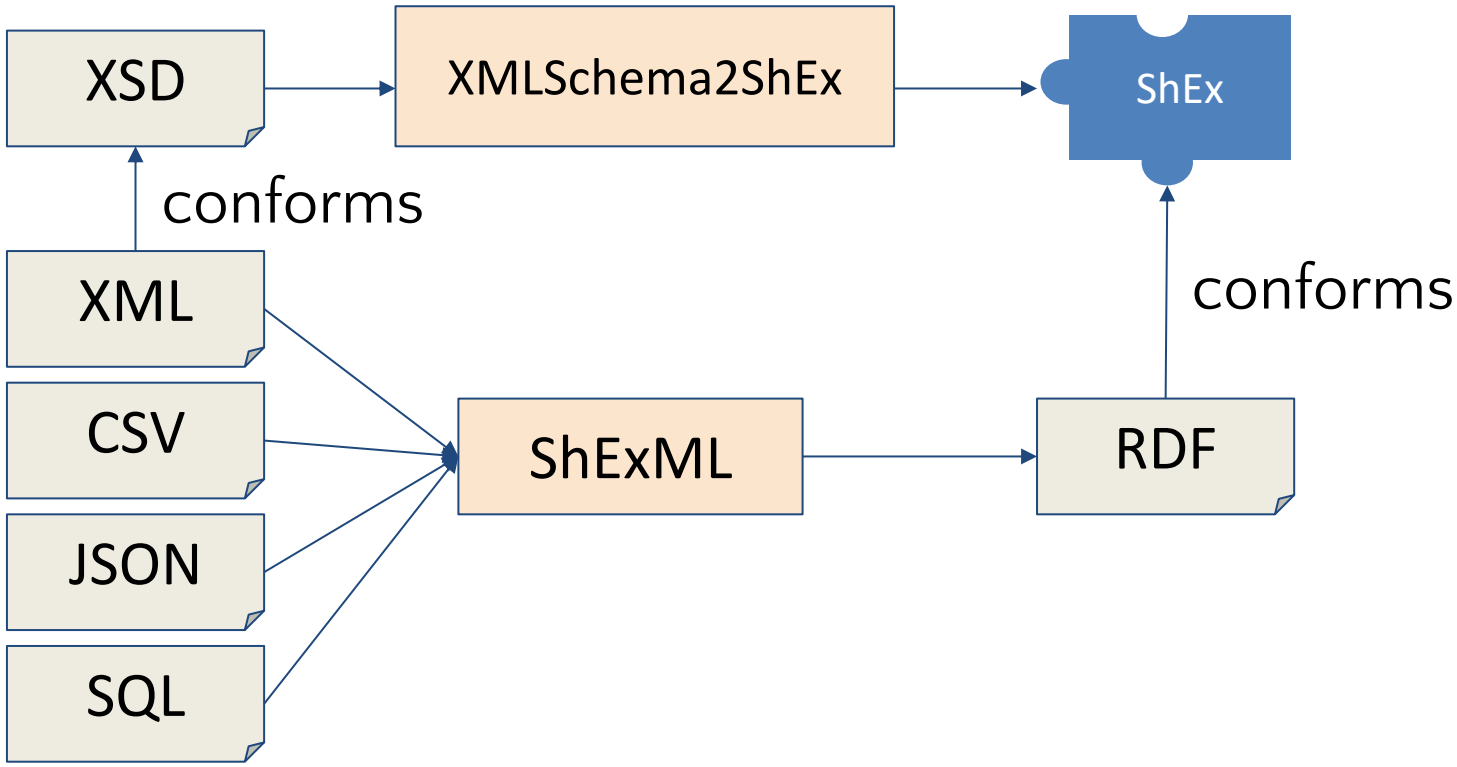
```
:S a :Square .
```

Shapes for data integration

XMLSchema2ShEx: Convert XML Schemas to shapes

ShExML: Domain specific language to convert data to RDF

Input formats: CSV, XML, JSON, SQL



Shapes ecosystems

- Wikidata provides a whole ShEx ecosystem
- Entity schemas can evolve and relate between each other
  - Directory: https://www.wikidata.org/wiki/Wikidata:Database_reports/EntitySchema_directory
  - Different schemas for the same entities?
    - Some schemas stress some aspects while others stress others
  - Evolution of schemas
  - Searching entity schemas

Conclusions

ShEx and SHACL have had a great level of adoption

But there are other types of Knowledge Graphs

- Much more work to do
- New tools and challenges

Acknowldgments

Awesome Semantic Shapes:

https://github.com/w3c-cg/awesome-semantic-shapes

Special thanks to Vladimir Alexiev for starting it

People from ShEx community group: Tom Baker, Kat Thornton, Andra Waagmeester,...