Shapes applications and tools

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

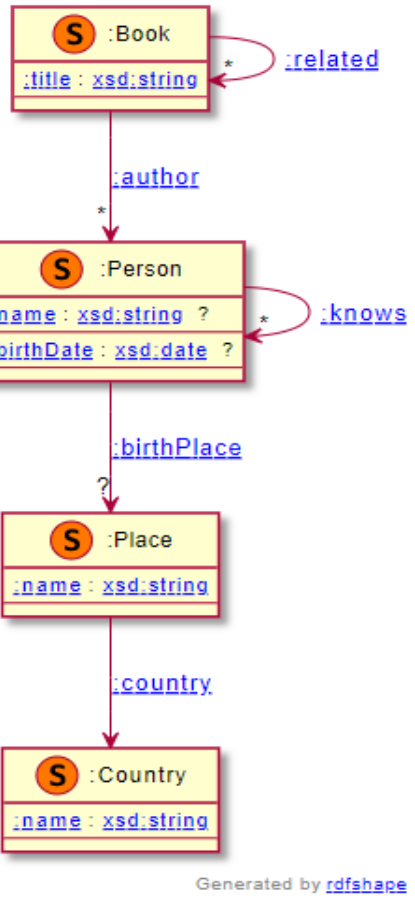
Contents

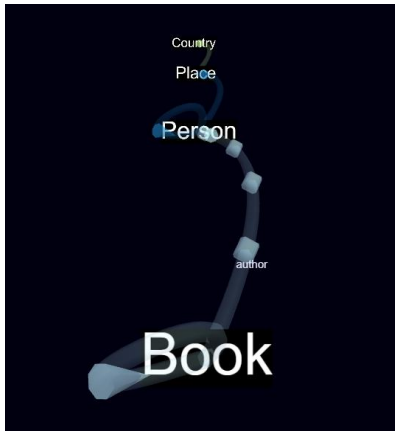
Some applications of Shapes

- Traditional application: Validate RDF data
- Tools to understand/manage the data models
  - Visualizations, HTML page generation
  - Editors
  - Obtaining shapes from data
- Creating subsets from Shapes
- Other applications:
  - Continuous integration
  - Generate code, SPARQL queries
  - Optimize SPARQL queries or triplestores based on their shapes
  - . . .

Tools to understand/edit shapes schemas

- UML-like diagram visualizations
  - Implemented in rdfshape and rudof
  - Edit the data models using UML editors (XMI prototype)
- HTML pages
  - Partial visualizations to improve usability browsing large data models
- 3D prototype visualization of shapes schemas





Continuous integration with Shapes

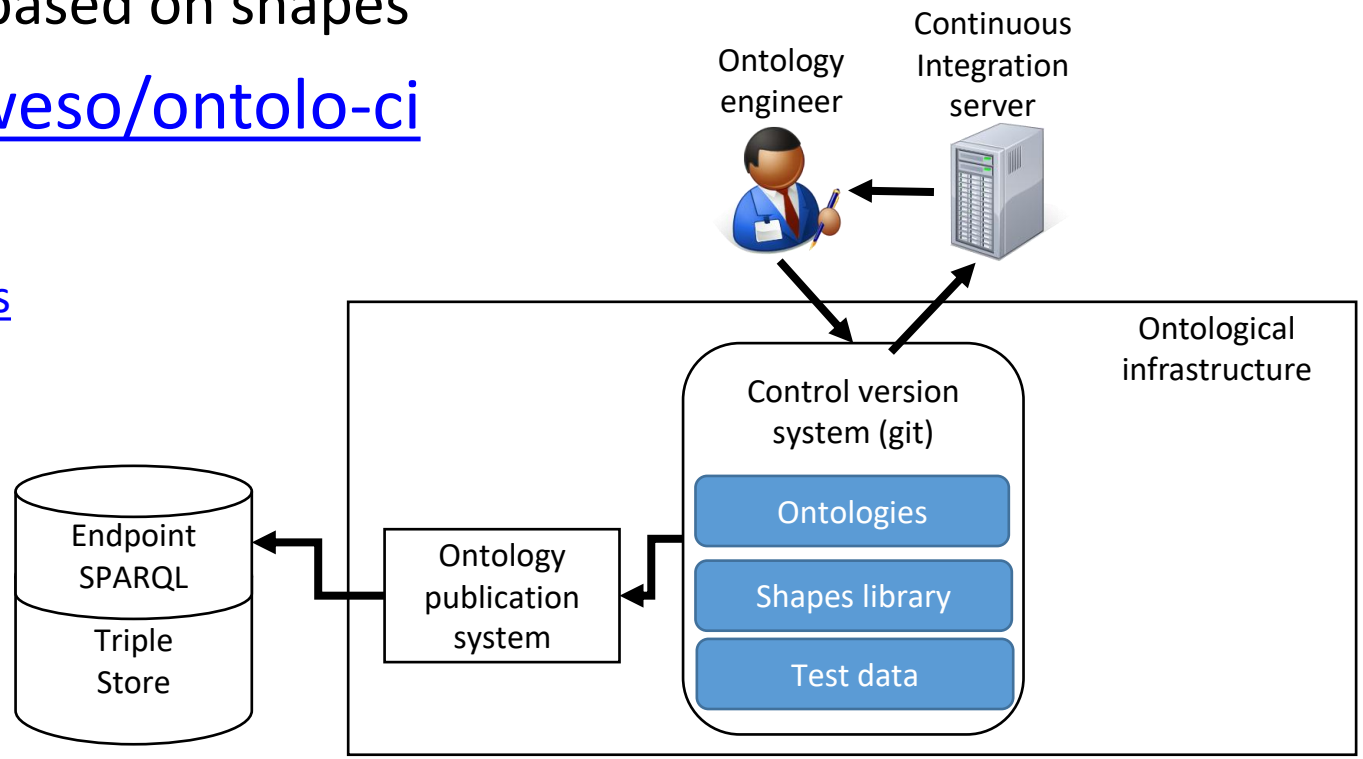
Coexistence between ontologies/shapes

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

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

Gene Ontology Shapes:

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



Continuous integration with Shapes

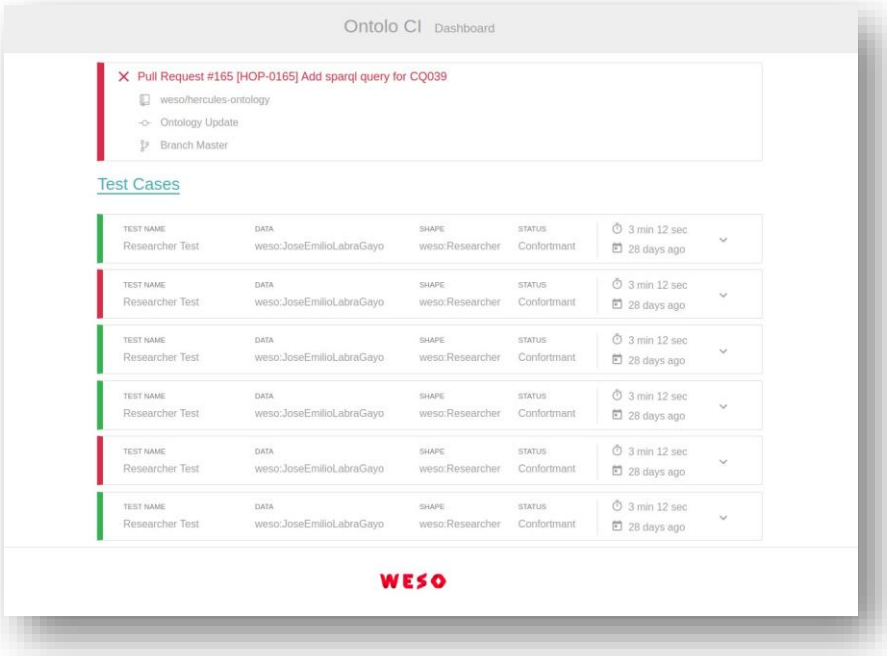
Ontolo-ci: https://github.com/weso/ontolo-ci

Developed as part of HERCULES-Ontology

Test-Driven-Development applied to Ontologies

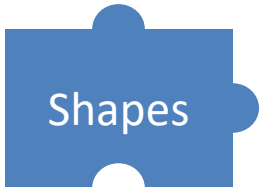
Input:

- Ontologies
- Shapes
- Test data
- Input shape map (SPARQL competency question)
- 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)



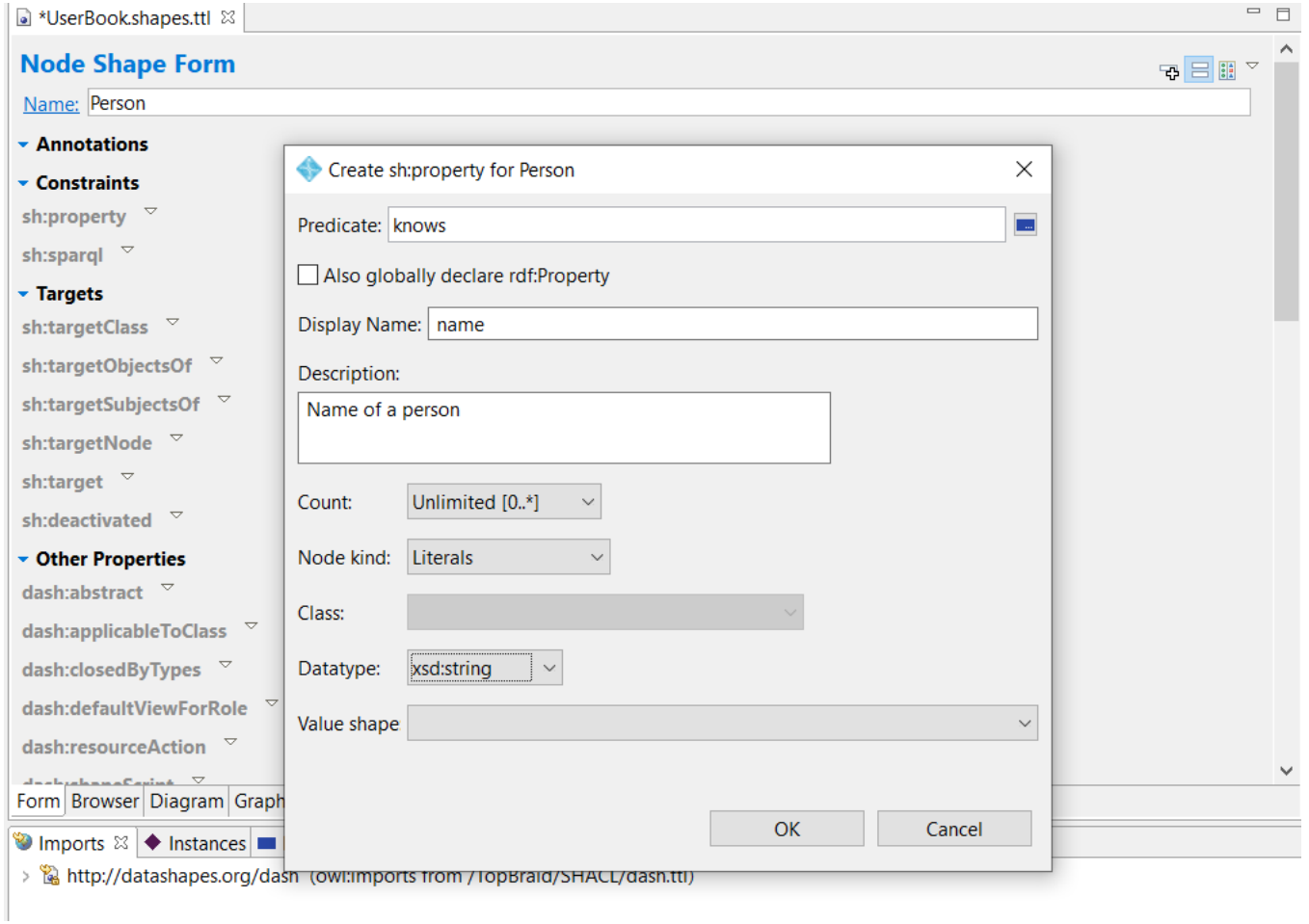
Text-based editors

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

- Syntax highlighting
- Auto-completion

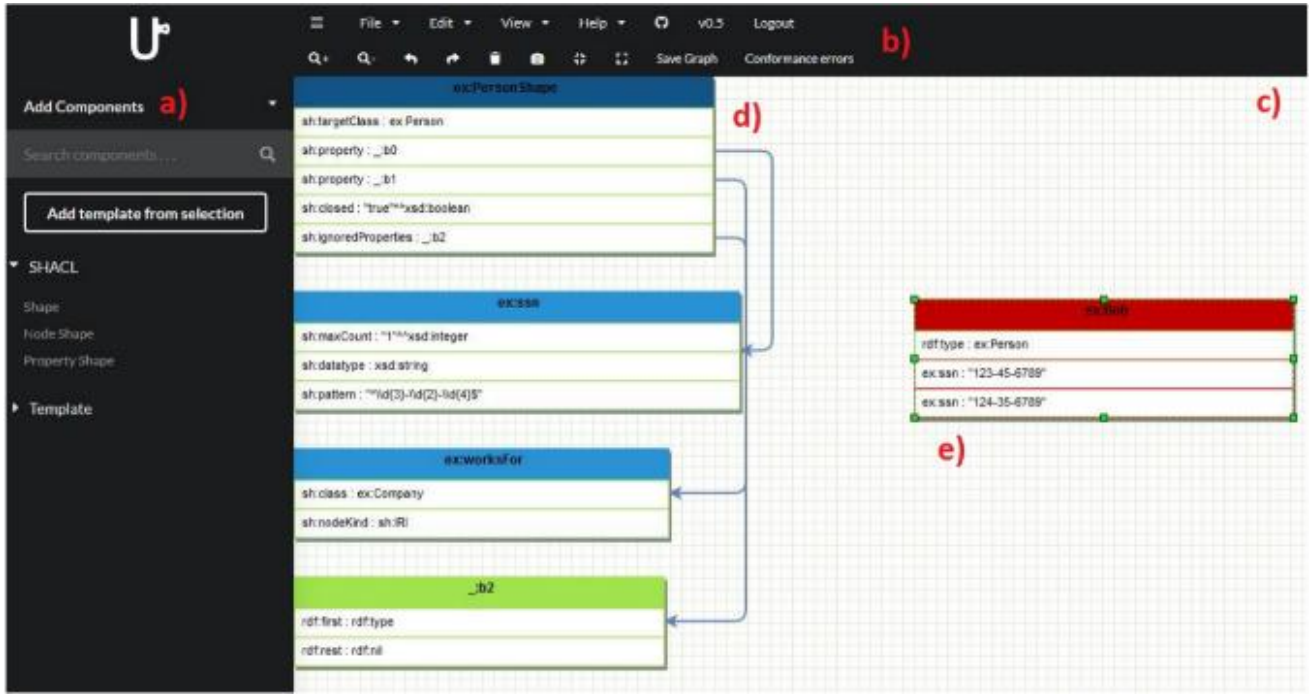
Shapes author tools: Top Braid Composer

Form based editor
Integrated with Top Braid product



Shapes author tools: UnSHACLed

Visual SHACL Editor in Javascript

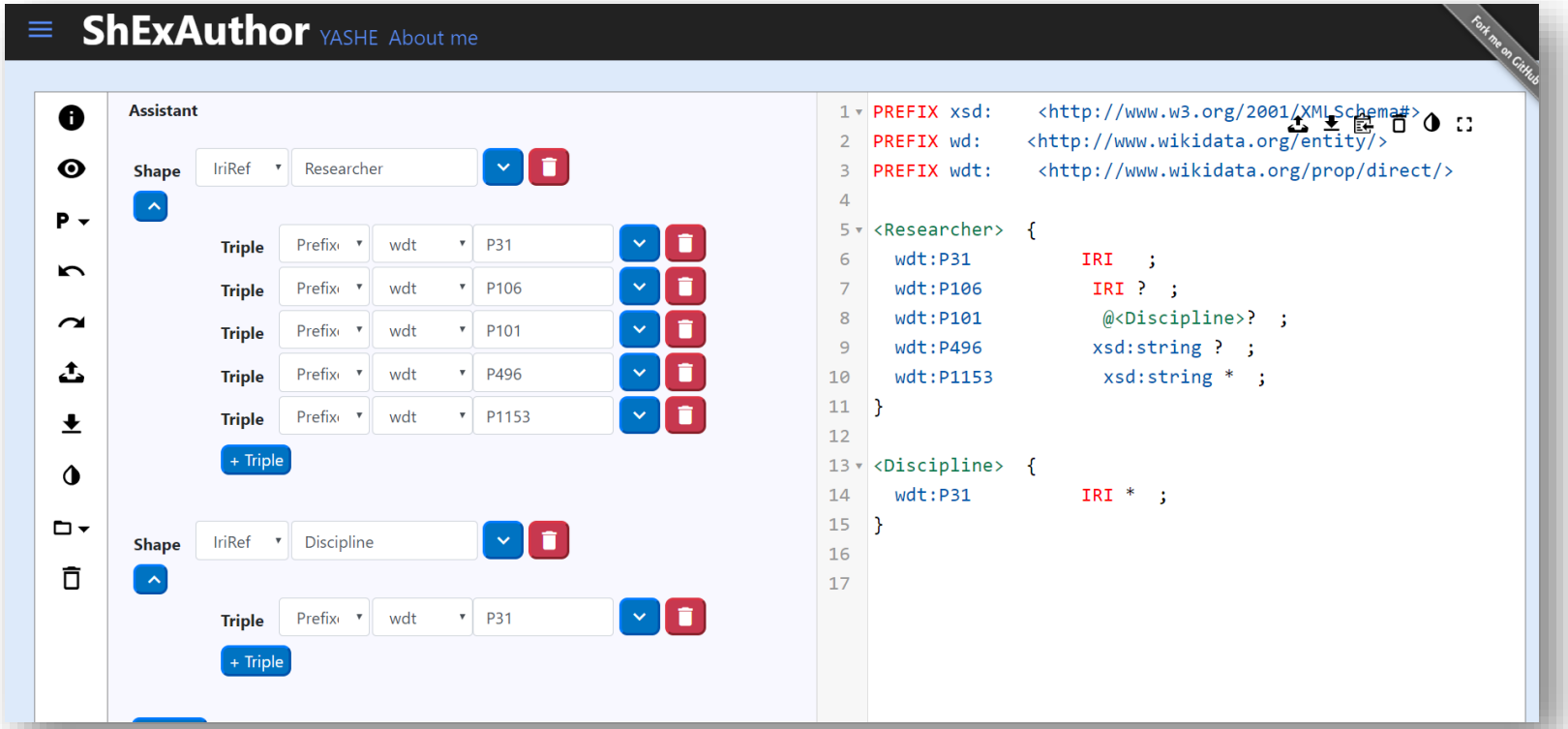


B. De Meester, P. Heyvaert, A. Dimou, and R. Verborgh, “Towards a Uniform User Interface for Editing Data Shapes,” in Proceedings of the 4th International Workshop on Visualization and Interaction for Ontologies and Linked Data, 2018, vol. 2187.

Shapes author tools: ShEx Author

ShEx-Author: Inspired by Wikidata Query Service

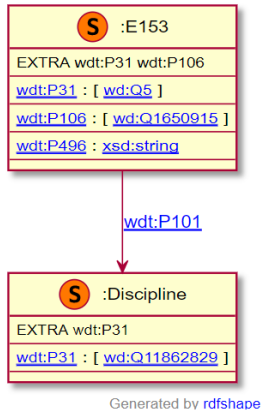
2 column: Visual one synchronized with text based

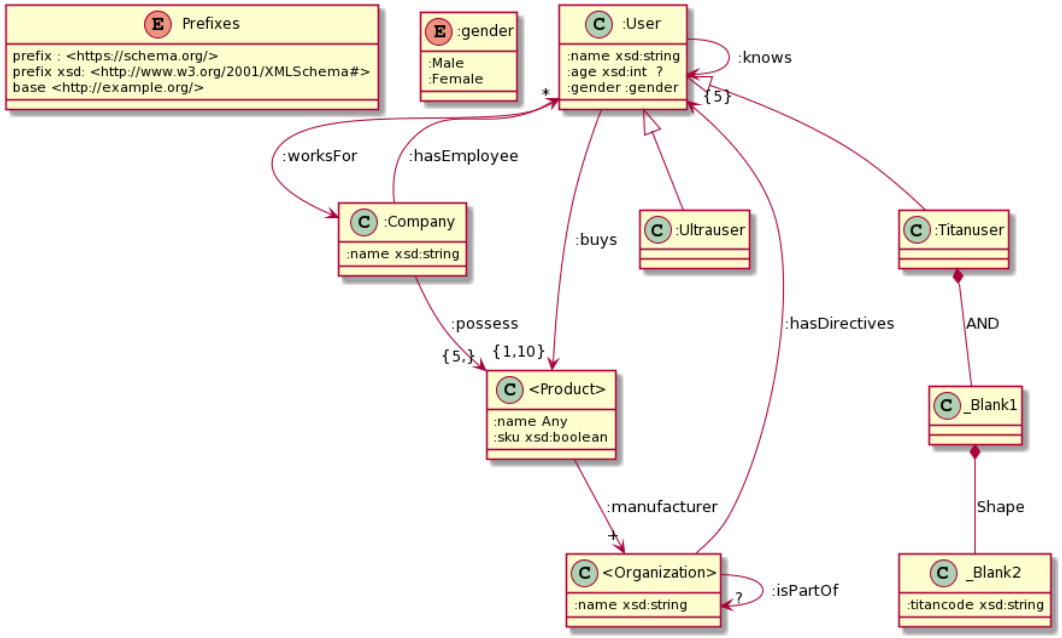


Shapes visualization

Integrated in RDFShape/Wikishape

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





Shapes from spreadsheets

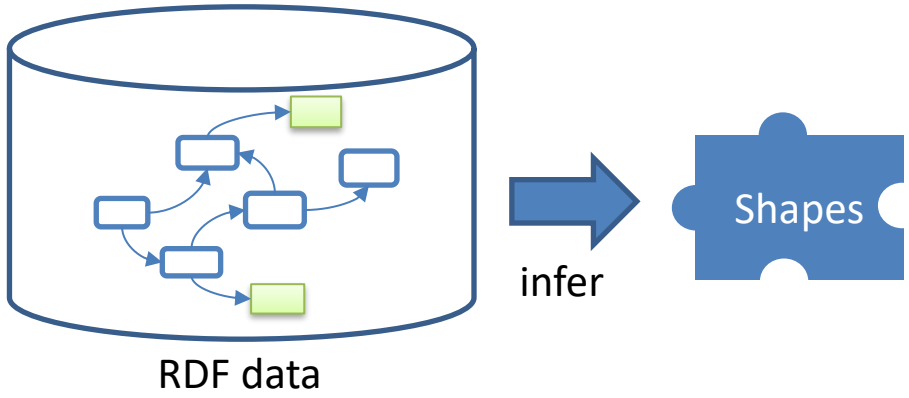
SKOS-Play was used at ELI to generate SHACL shapes from Excel
ShExstatements: https://shexstatements.toolforge.org/
ShExCSV: CSV representation of Shapes
Hermes: ShExCSV processor, https://github.com/weso/hermes
Creating data models using spreadsheets
DCTAP data models (templates in XLSX, CSV), recently added to rudof

Generating Shapes from RDF data

Useful use case in practice

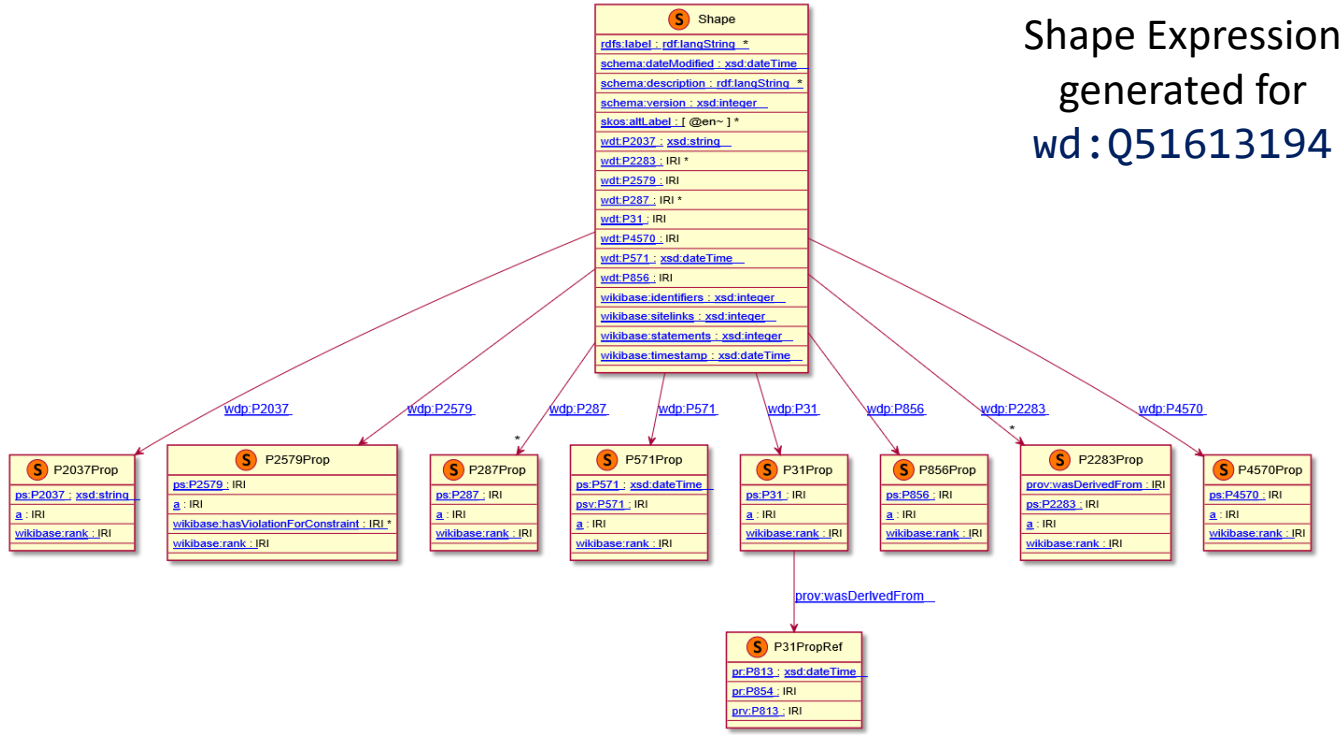
Some prototypes

- RDFShape: http://rdfshape.weso.es
- sheXer: http://shexer.weso.es/



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: https://github.com/DaniFdezAlvarez/shexer

Implemented as a Python library

Supports Shapes generation from large SPARQL endpoints

- Can generate shapes from sampling

ShEx consolidator can be used for large RDF data

Application example:
Creating Knowledge Graphs subsets

Some current problems…

- Large knowledge graphs are difficult to handle
  - SPARQL queries require computational resources
    - Endpoints usually impose limits (timeouts…)
- Contents are continually evolving
  - Results of SPARQL queries now may be different later
  - Research based on large KGs difficult to be reproducible

Some applications for KG subsets (1)

Performance

- Enable SPARQL queries that don’t work with whole KG

Data integration

- Create domain specific subsets and integrate with other data

Reproducibility

- Snapshots of KG contents that can be cited & reused

Some applications for KGs subsets (2)

Transformation and enrichment

- Add and integrate content from different KGs

KG in your pocket

- Create mobile apps based on some subset

KG analysis

- Create and compare subsets from historical dumps

License combination

- Combine subsets from KG with some license (CC0) with others

Motivating use case: GeneWiki project

SuLab / WikidataIntegrator

Code | Issues 4 | Pull requests 1 | Projects 0 | Pulse | Graphs

A Wikidata Python module integrating the MediaWiki API and the Wikidata SPARQL endpoint

397 commits | 2 branches | 1 release

Branch: master | New pull request

sebotic fixed an omission where new items don't get created when domain not s…

doc	Wikidata to Wikipedia mapping prototype for diseases added.
wikidataintegrator	fixed an omission where new items don't get created when domain not s…
.git-commit-hash	crop article title to 250 chars
.gitignore	pep8 package name, imports, reformat
LICENSE.txt	change to MIT license

Jenkins

Jenkins › Running

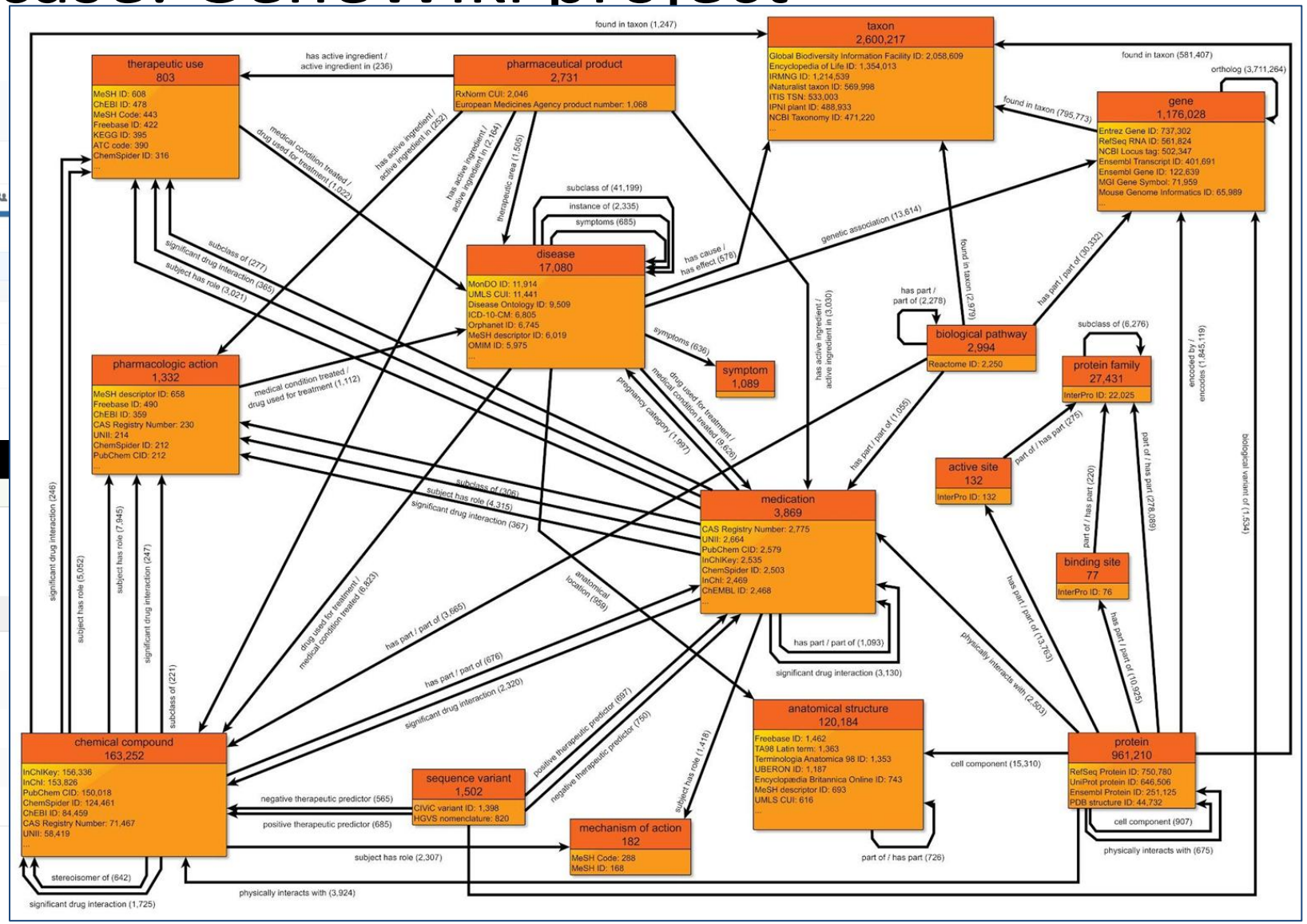
- New Item
- People
- Build History
- Edit View
- Delete View
- Manage Jenkins
- My Views
- Credentials

Build Queue

Running Bots

All | Running | +

S	Name
	ProteinBot_homo_sapiens
	GOBot_bigmem
	GeneBot_Homo_sapiens
	Disease_Ontology
	GeneDiseaseBot

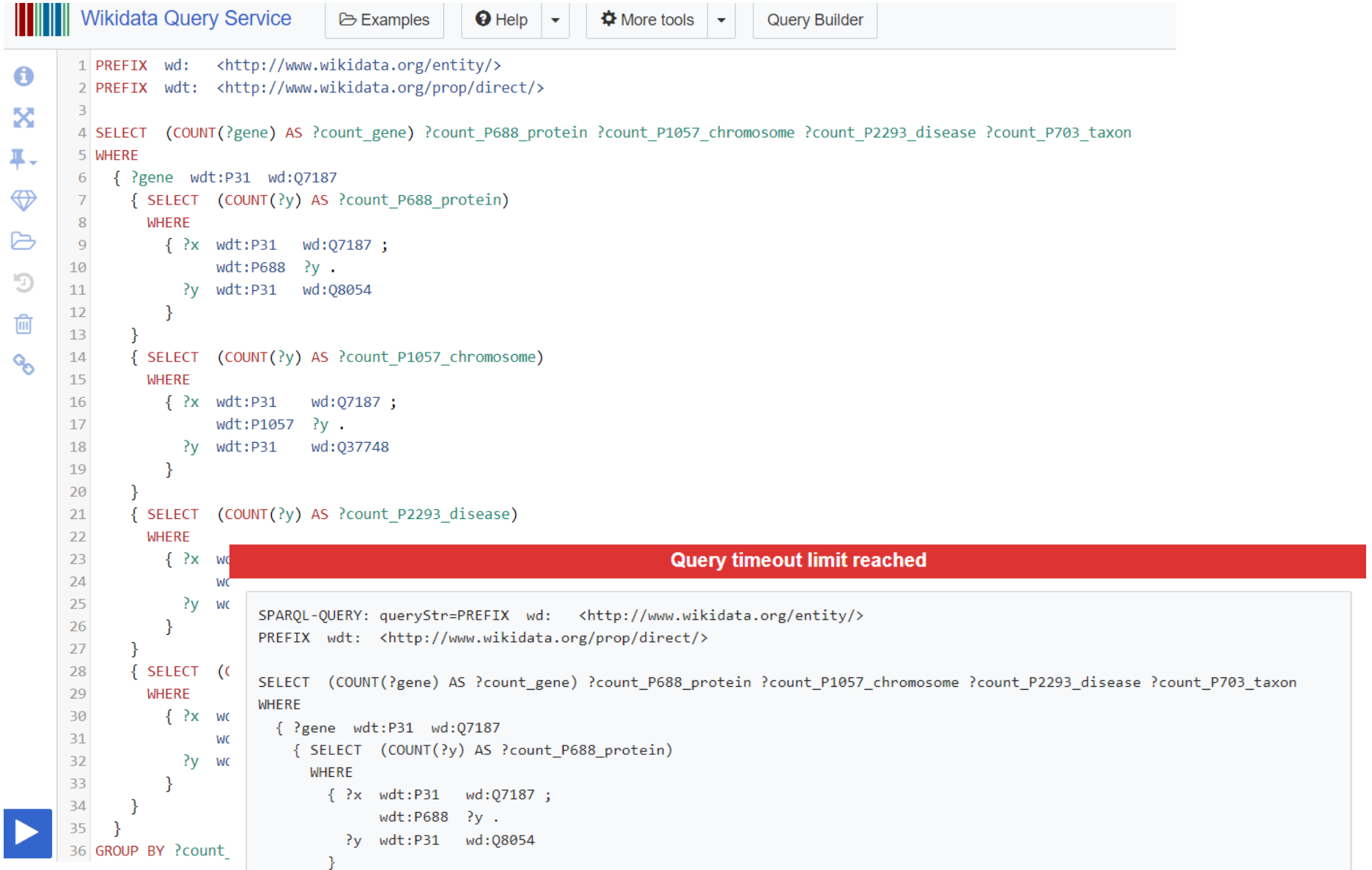


Source: Wikidata as a knowledge graph for the life sciences, A. Waagmeester et al, https://elifesciences.org/articles/52614

Reusing turned out to be difficult

Example: counting number of genes and associated elements:

- Proteins
- Chromosomes
- Disease
- Taxons



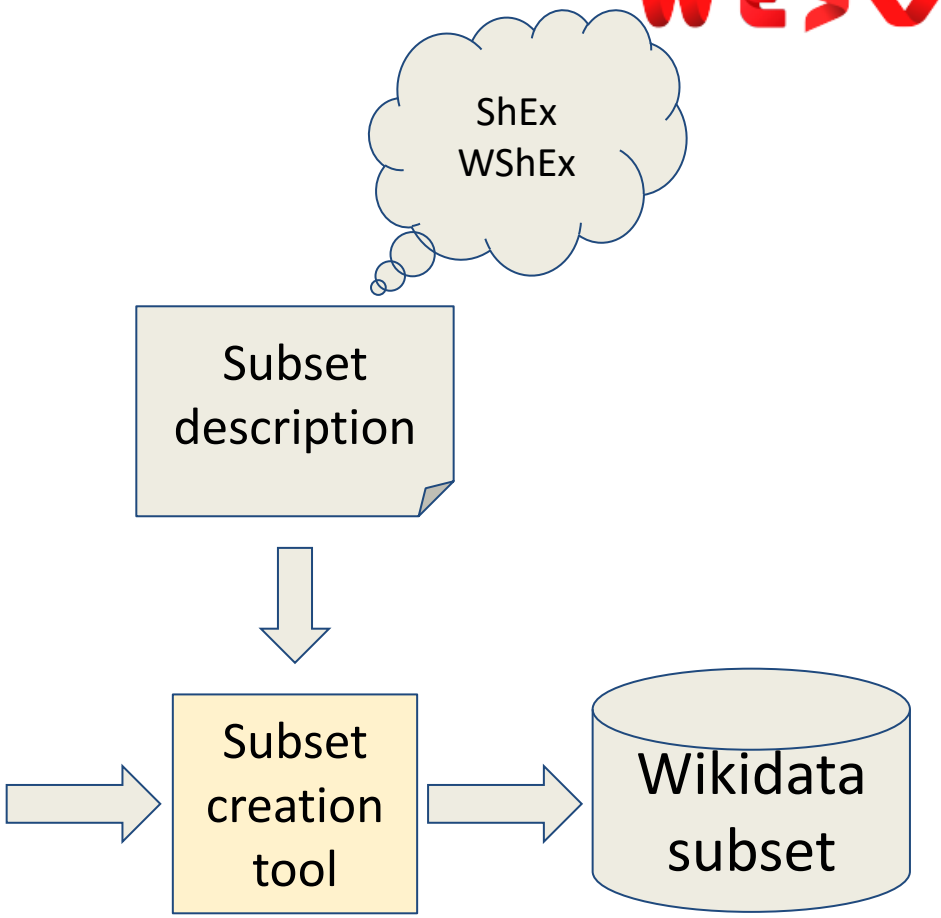
Wikidata subsetting efforts

Collaborative research driven by a practical need

Most of the advances were triggered by SWAT4HCLS and Biohackathons

- 12th SWAT4HCLS conference, 2019. Wikidata:WikiProject Schemas/Subsetting - Wikidata
- Europe Biohackathon, 2020, project 35 [preprint]
- Europe Biohackathon, 2021, project 21
- Europe Biohackathon, 2022 project 11, [preprint]
- Japan Biohackathon, 2023 [preprint]
- 2023, Paper: Wikidata subsetting: approaches, tools and evaluation, accepted at Semantic Web Journal

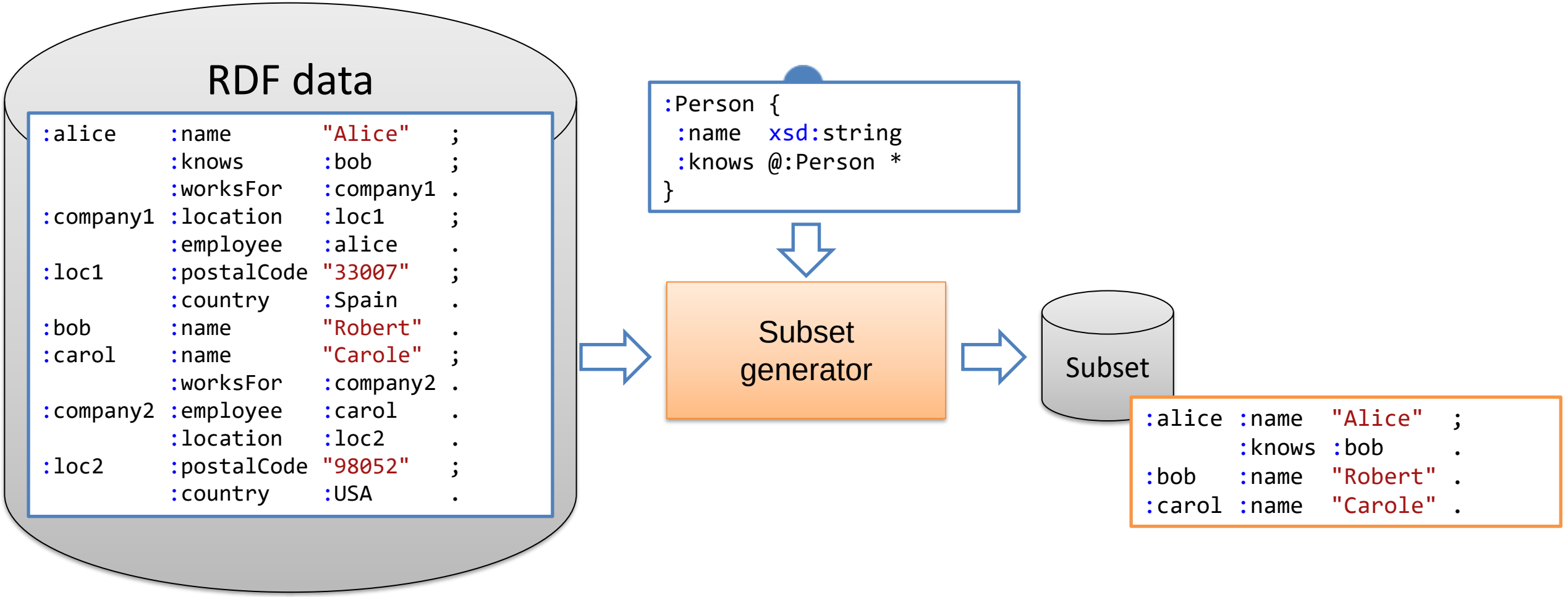
Problem statement



Subsetting based on ShEx

Generate subsets from ShEx

ShEx describes the contents of the expected subset



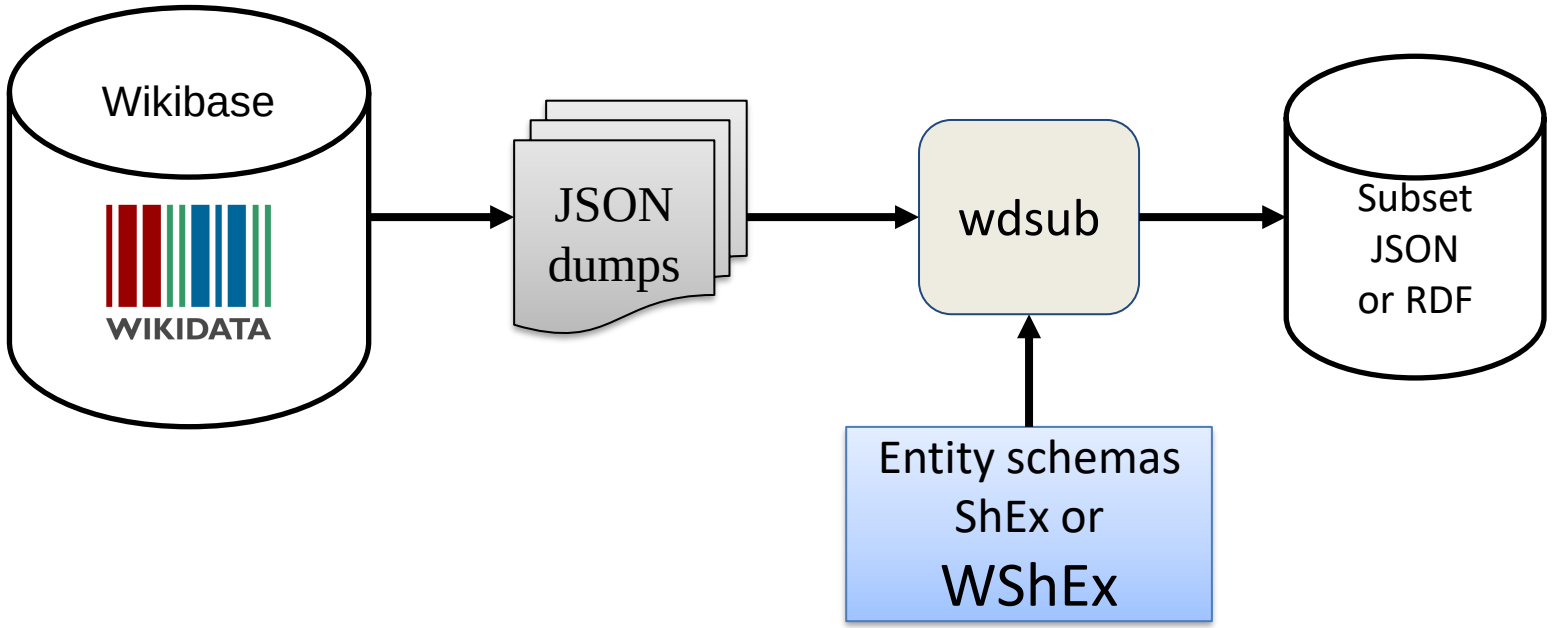
wdsub

Input:

- ShEx/WShEx schema
- Wikidata dumps in JSON

Output

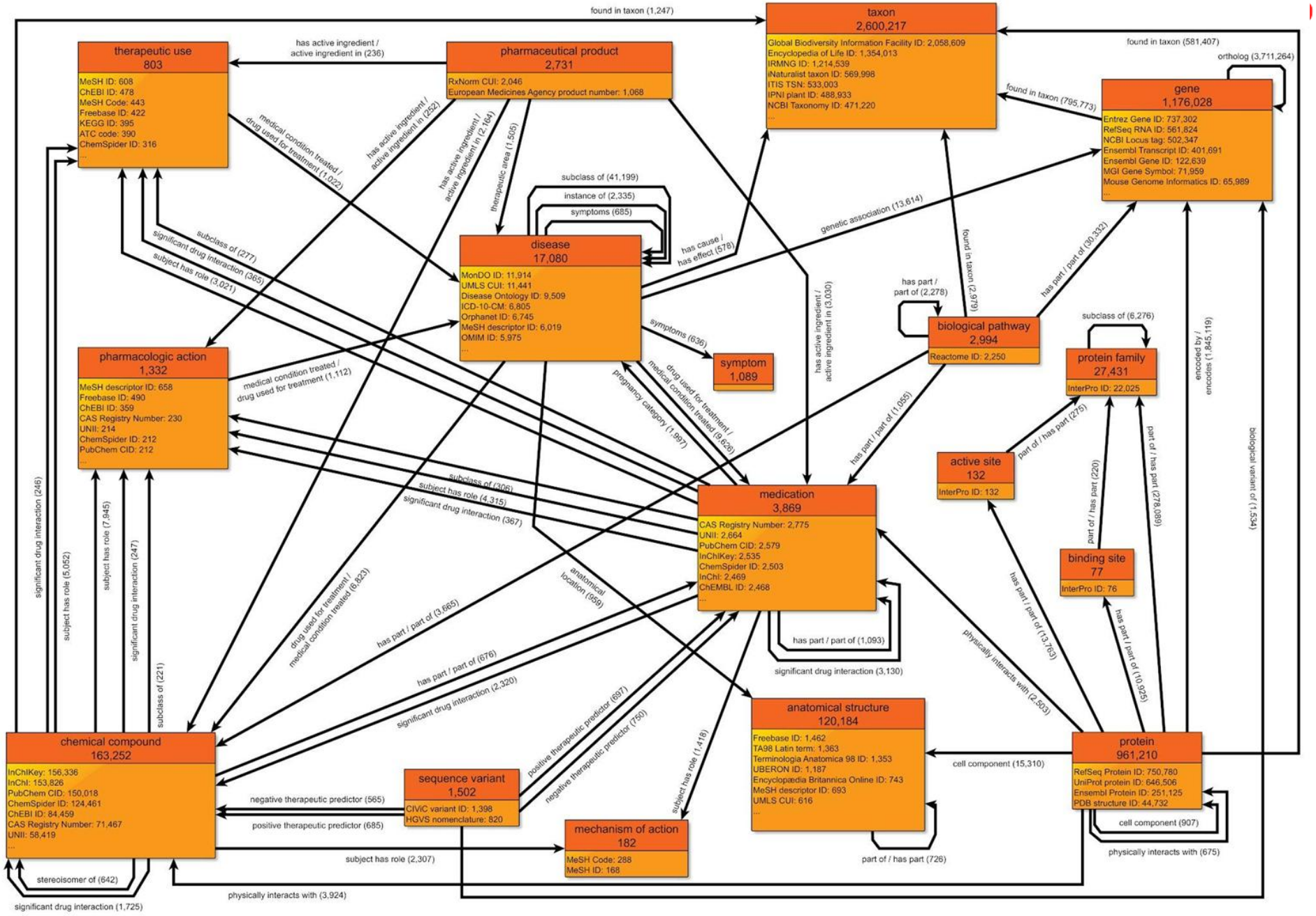
- Dumps in JSON/RDF



Link: https://github.com/weso/wdsub
Docker (CLI): wesogroup/wdsub:0.0.32

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

Other examples

The paper contains other other studies about

Multilingual extraction

```
start= @:gene OR
       @:taxon

:gene EXTRA wdt:P31 {
  rdfs:label          [ @en @es @fa @nl ]      ;
  schema:description  [ @en @es @fa @nl ]      ;
  skos:altLabel       [ @en @es @fa @nl ] *    ;
  wdt:P31             [ wd:Q7187 ] ;
  wdt:P703           @:taxon *    ;
}

:taxon EXTRA wdt:P31 {
  rdfs:label          [ @en @es @fa @nl ] ;
  schema:description  [ @en @es @fa @nl ] ;
  skos:altLabel       [ @en @es @fa @nl ] * ;
  wdt:P31             [ wd:Q16521 ] ;
}
```

Qualifiers and references

```
. . .
<gene> EXTRA wdt:P31 {
  rdfs:label [ @en ] ? ;
  wdt:P31 [ wd:Q7187 ] * ;    # is instance of (P31) gene (Q7187)
  wdt:P351 . ? ;              # has one or no Entrez Gene ID (P351)
  p:P2293 {
    ps:P2293 @<disease> + ;
    prov:wasDerivedFrom @<References> + ;
  } + ;
}
. . .
```

More information at paper: 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

Subsets based on Pregel algorithm

Pregel algorithm = parallel algorithm proposed at Google

"think like a vertex"

2 prototype implementations (work in progress)

- Scala: SparkWDSub
  - We were able to create subsets in 36 minutes with large cluster (512 cores)
- Rust: pregel-rs based on Polars and
  - It can process Wikidata-dumps
  - Recently applied to large RDF data: UniProt

More information at paper:

- Creating Knowledge Graphs Subsets using Shape Expressions, J. Labra, arXiv:2110.11709
- Using Pregel to create Knowledge Graphs subsets described by non-recursive Shape Expressions, A. Préstamo, J. Labra, accepted at KGSWC' 23

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:

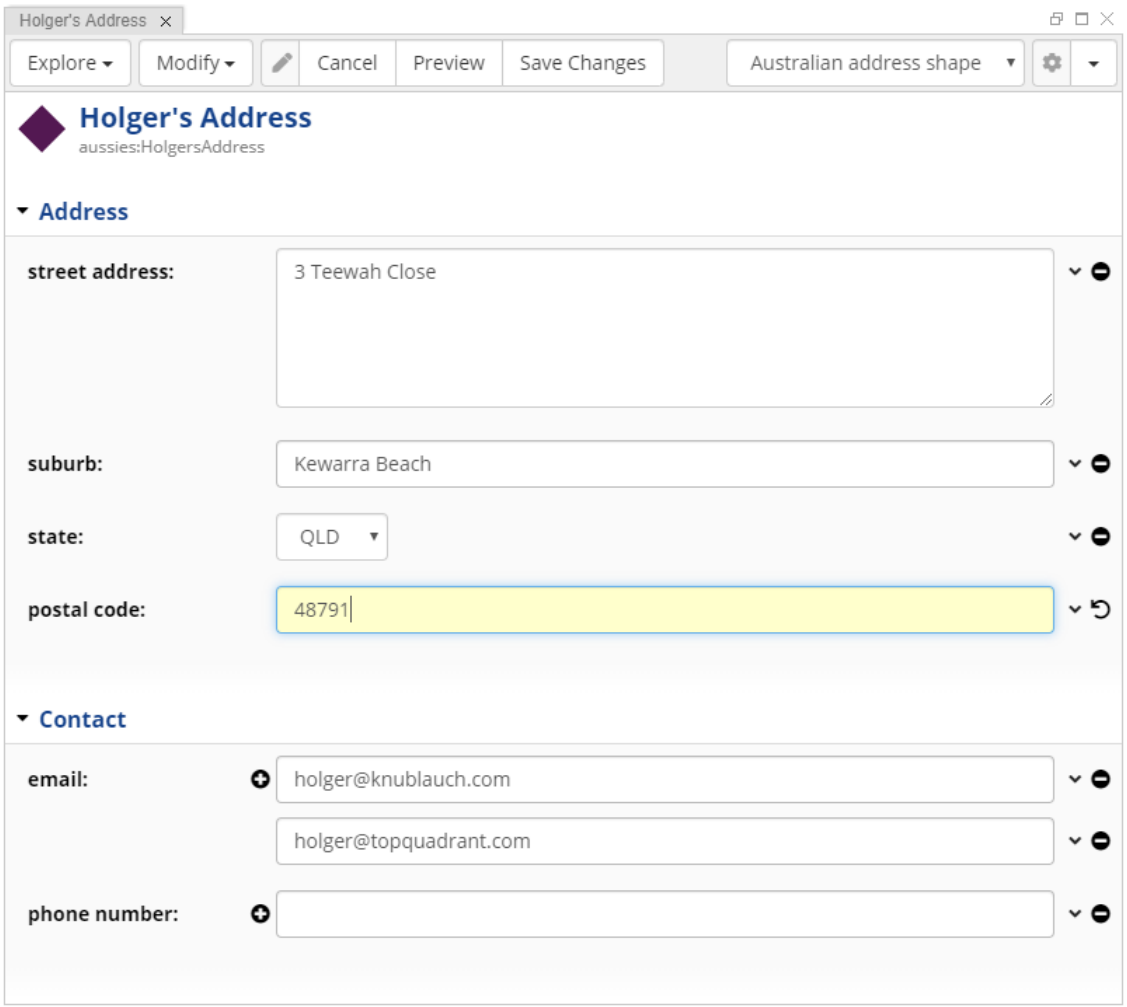
https://ericprud.github.io/shex-form/?manifestURL=examples/manifest.json

UIs and shapes: TopQuadrant

Form generation from SHACL

DASH vocabulary:

http://datashapes.org/forms.html

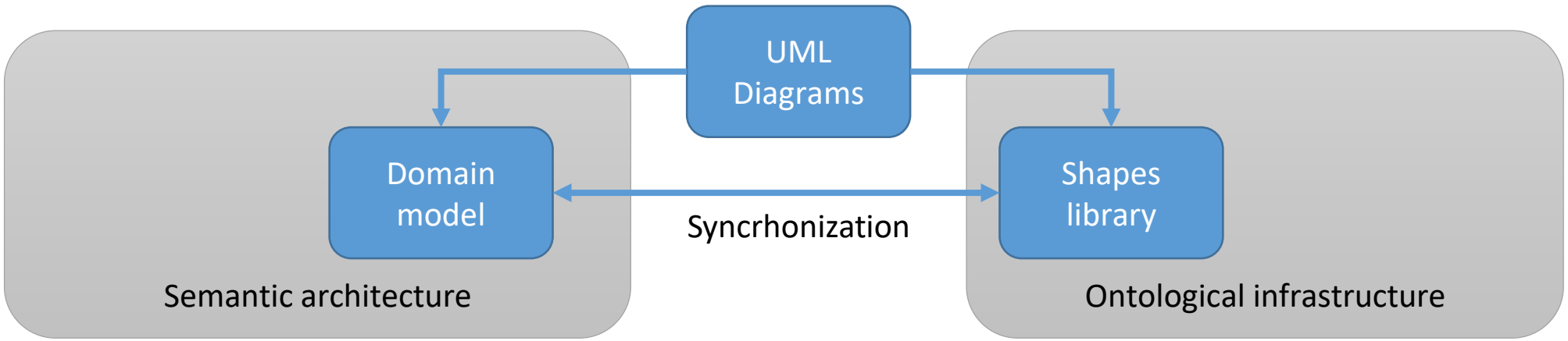


Generating code from shapes

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 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

Shapes can be a very important aspect of Knowledge Graphs

Graph data is flexible and shapes also

Shapes ecosystems with:

- Prescriptive shapes
- Descriptive shapes
- Suggestive shapes (can *suggest* properties to employ)

Domain experts are a key aspect of this

END OF PRESENTATION

Acknowldgments

Awesome Semantic Shapes:

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

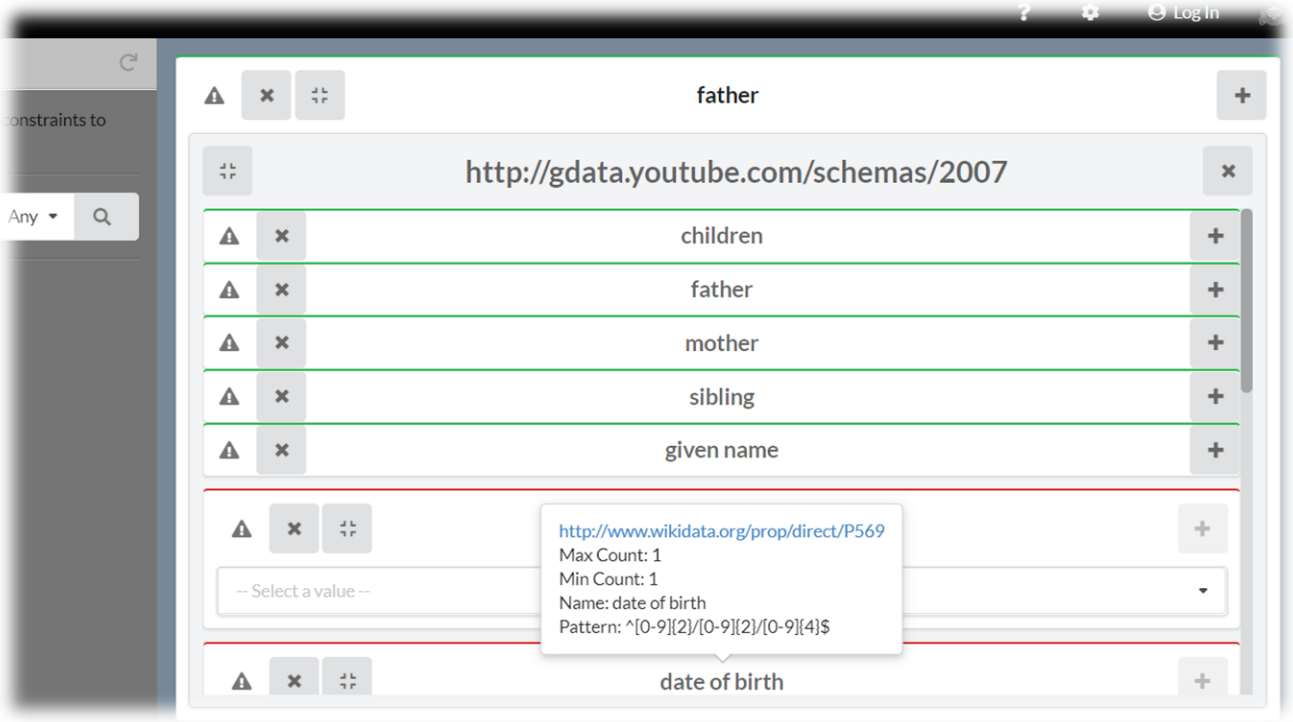
Special thanks to Vladimir Alexiev for starting it

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

UIs and shapes: Schímatos

http://schimatos.org/

It will be presented at ISWC20

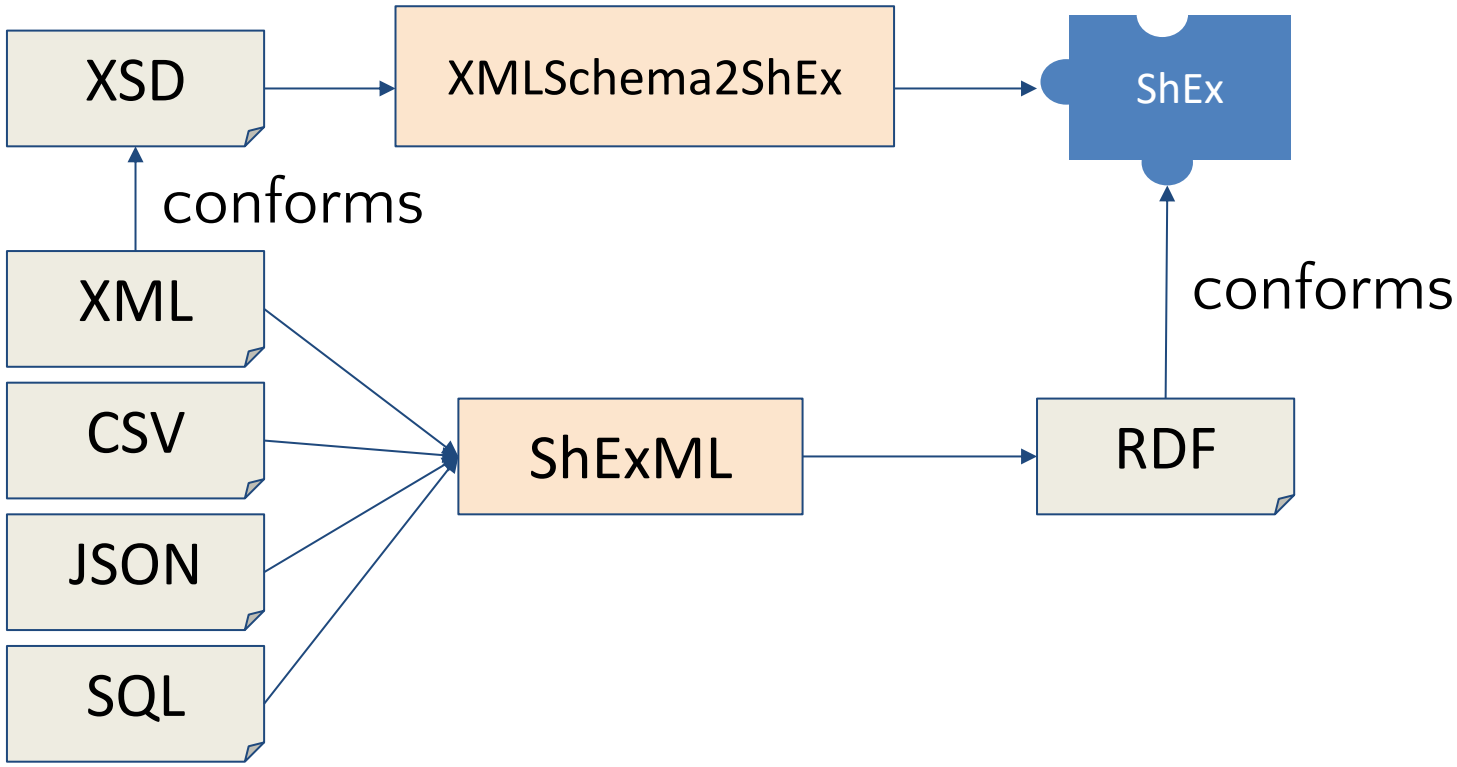


Shapes for data integration

XMLSchema2ShEx: Convert XML Schemas to shapes

ShExML: Domain specific language to convert data to RDF

Input formats: CSV, XML, JSON, SQL



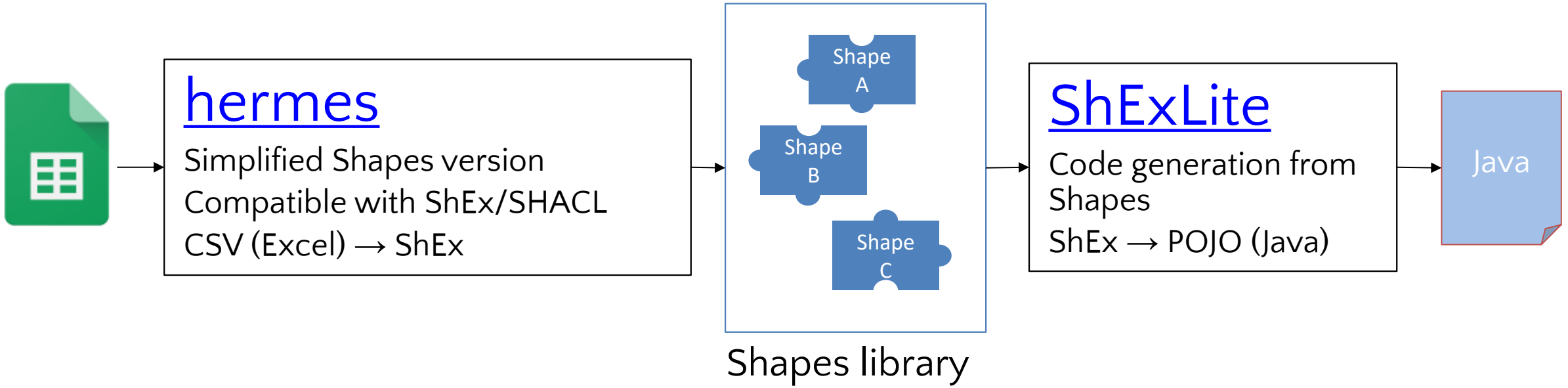
Generating code from shapes

Generate domain model from shapes

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

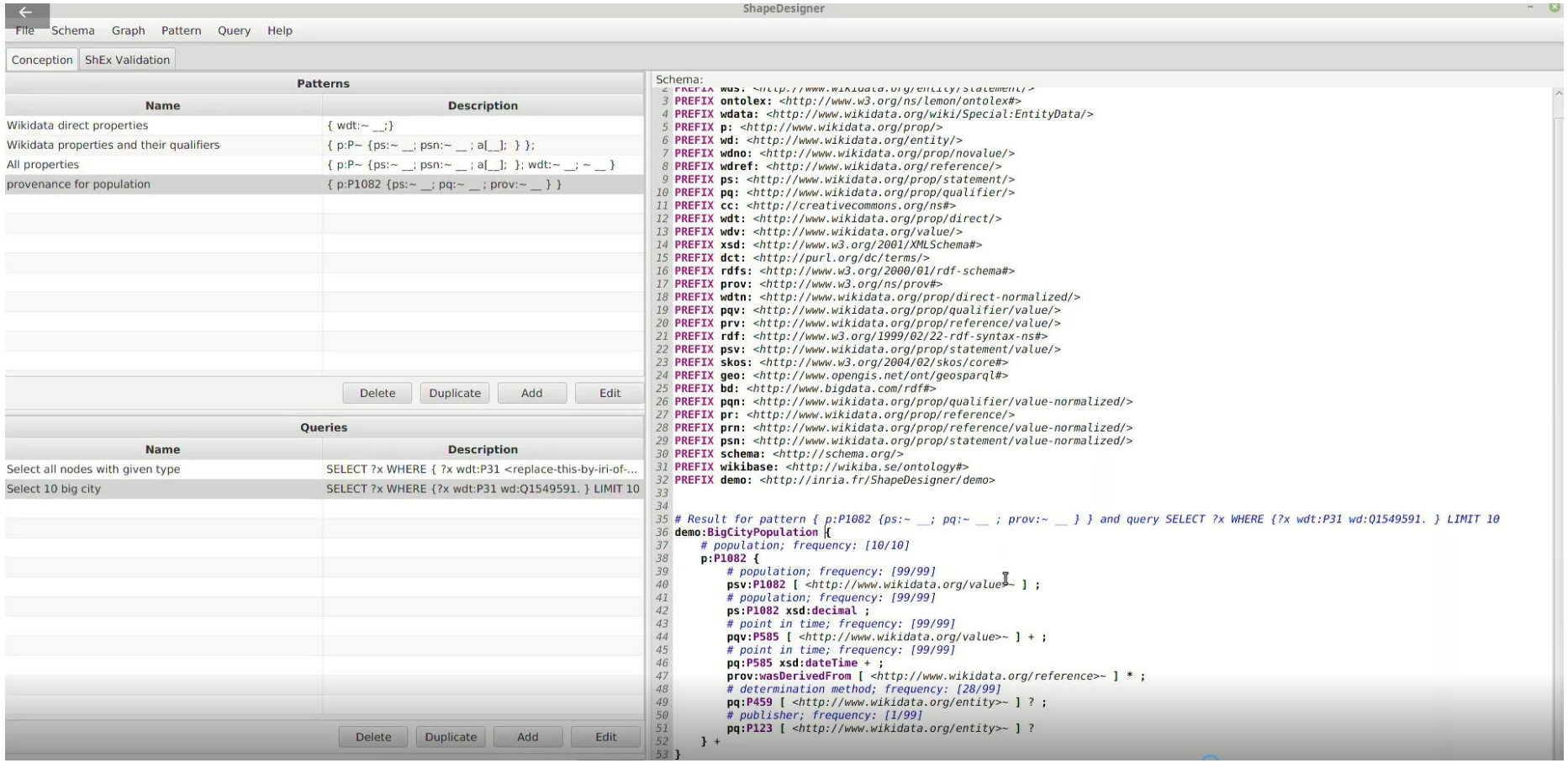
Shapes generation from those templates

Java code generation (POJOs) from those shapes



Shapes from data: ShapeDesigner

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



Other uses of Shapes

UIs and shapes

Generating code from Shapes

Shapes and rules

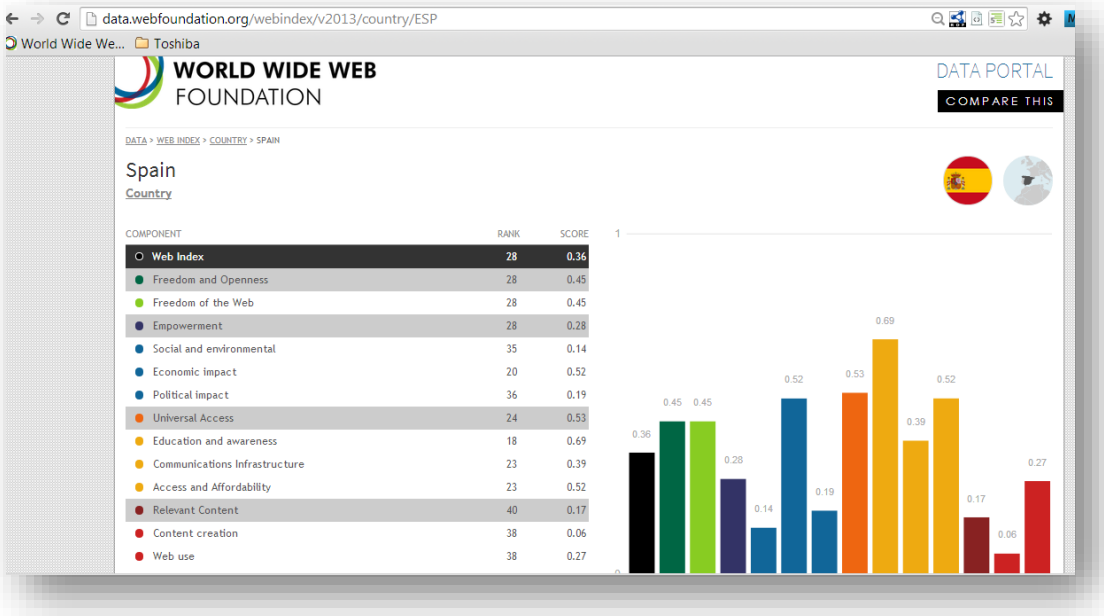
Data portals

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

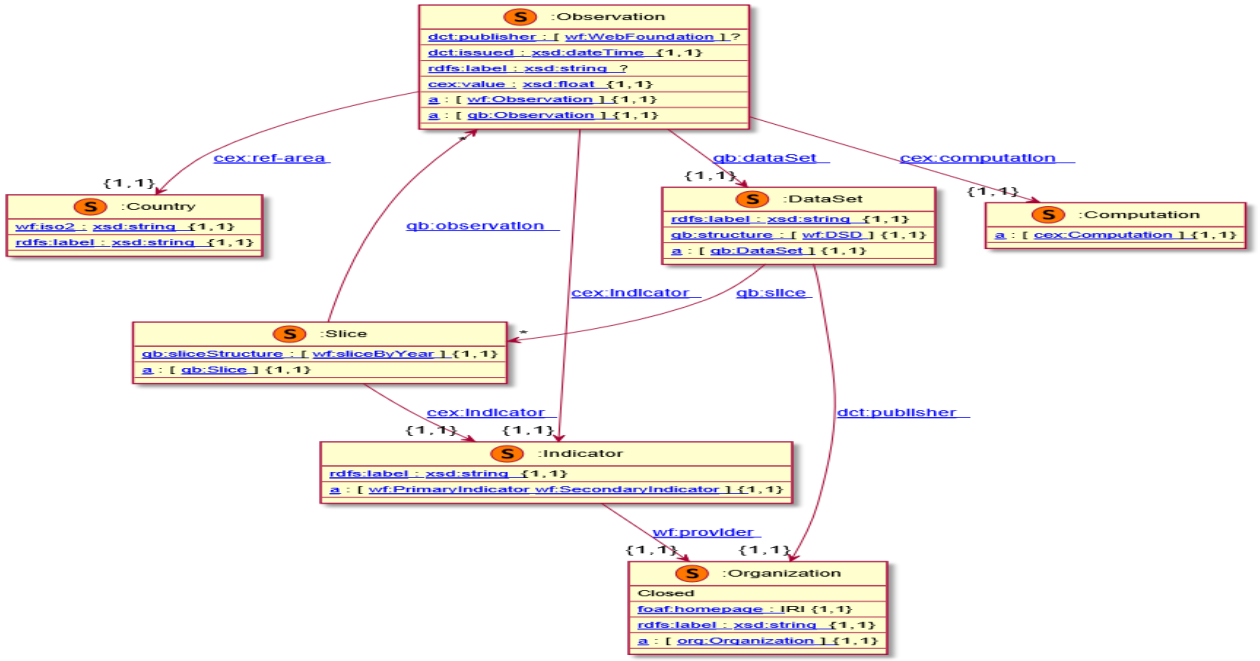
Examples: WebIndex (Web Foundation)

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

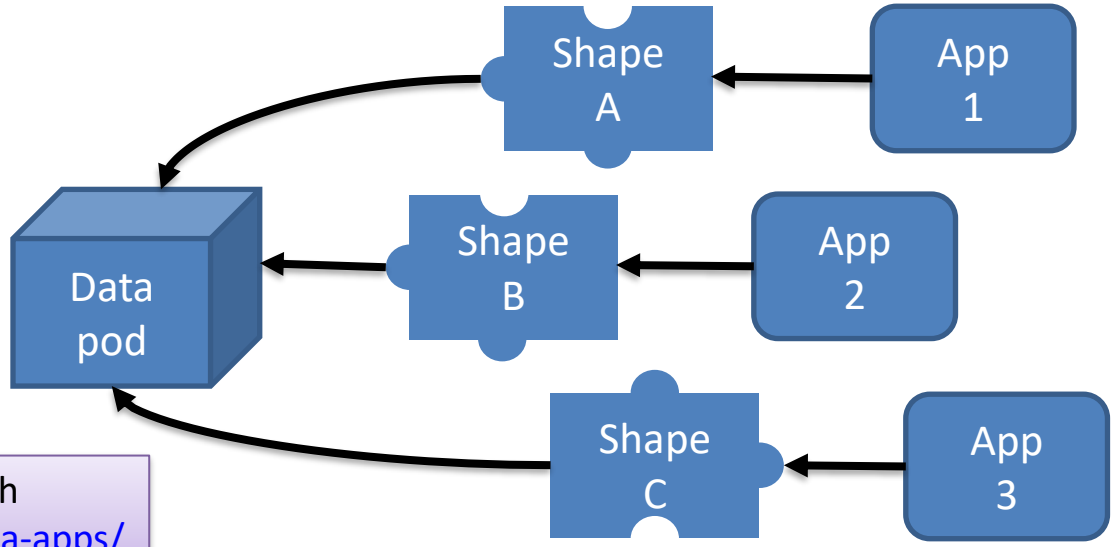
Solid project

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.

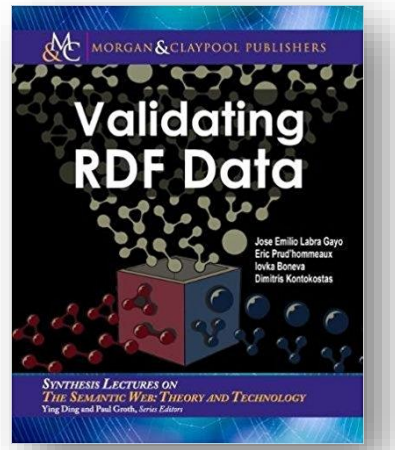
Example: https://www.hl7.org/fhir/observation.html

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