phuse·virtual

Single Day Event

Multilingualism in Data Science: What's Your Second Programming Language?

phuse.global

The Languages of Knowledge Graphs

Introduction Tim Williams

- **Statistical Solutions Lead**
  - **UCB Biosciences**, Raleigh N.C.
- **Knowledge Graph Project Lead @ PHUSE**
- Knowledge Graphs since 2011
- ~ 20 years experience in Pharmaceutical Industry
- Systems Administrator, Systems Validation & Deployment, Research Associate (Epidemiology), Programmer.

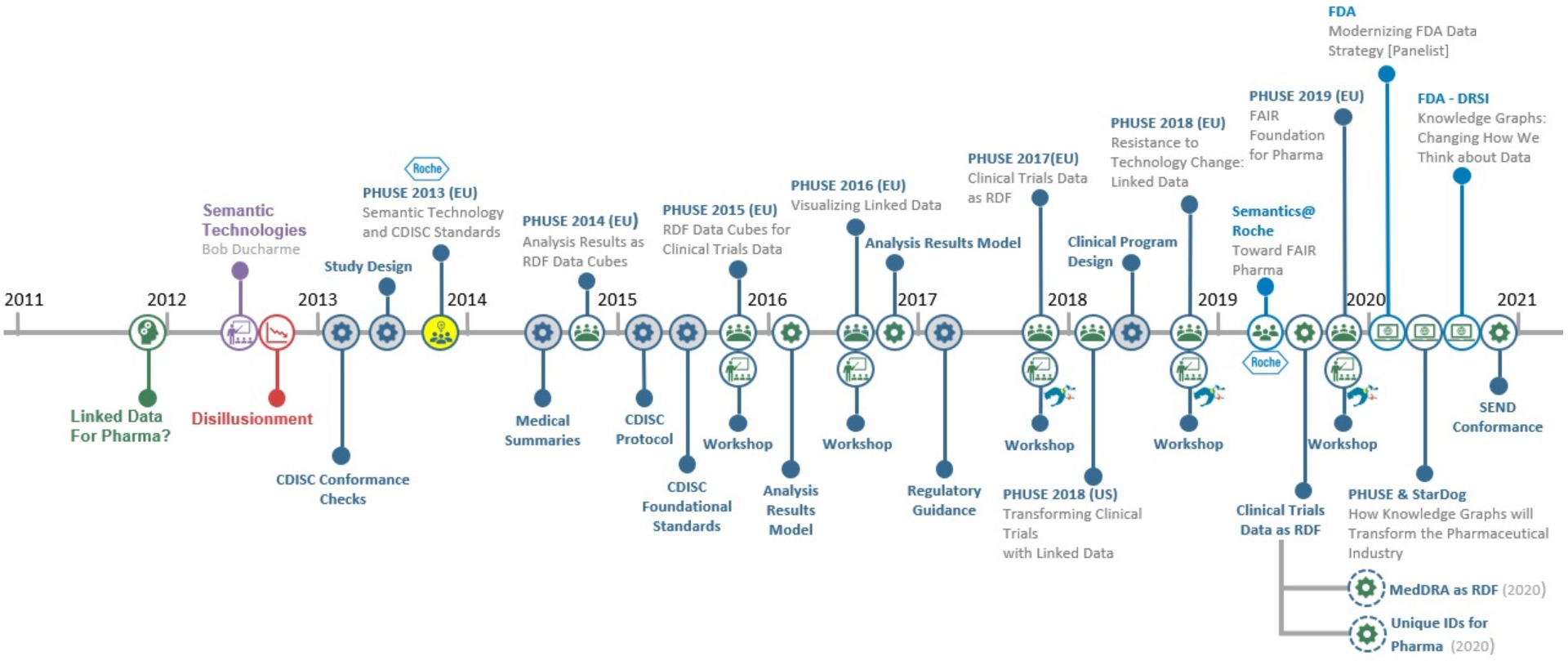
LinkedDataTim@gmail.com

@NovasTaylor

https://www.linkedin.com/in/timpwilliams

Opinions are my own

My Knowledge Graph Timeline



Select Graph Concepts and Languages

General Concepts
- Linked Data
- Semantic Web
- Knowledge Graph
- Reasoner & Inference
- Nodes, Edges
- URIs / IRIs
- Graph Database
- Triplestore
- F.A.I.R. Principles

Graph Types
- Resource Description Framework (RDF)
  - RDF-Star
- Property Graph

Rules and Validation Languages
- SPARQL Inferencing Notation (SPIN)
- Semantic Web Rule Language (SWRL)
- Rule Interchange Format (RIF)
- Shape Expressions (ShEx)
- Shape Constraint Language (SHACL)

Query Languages
- SPARQL
  - SPARQL-Star
- Cypher
- GraphQL

Mapping Languages
- RML
- R2RML
  - SMS

Miscellaneous
- LDFlex
- LDN (Linked Data Notifications)

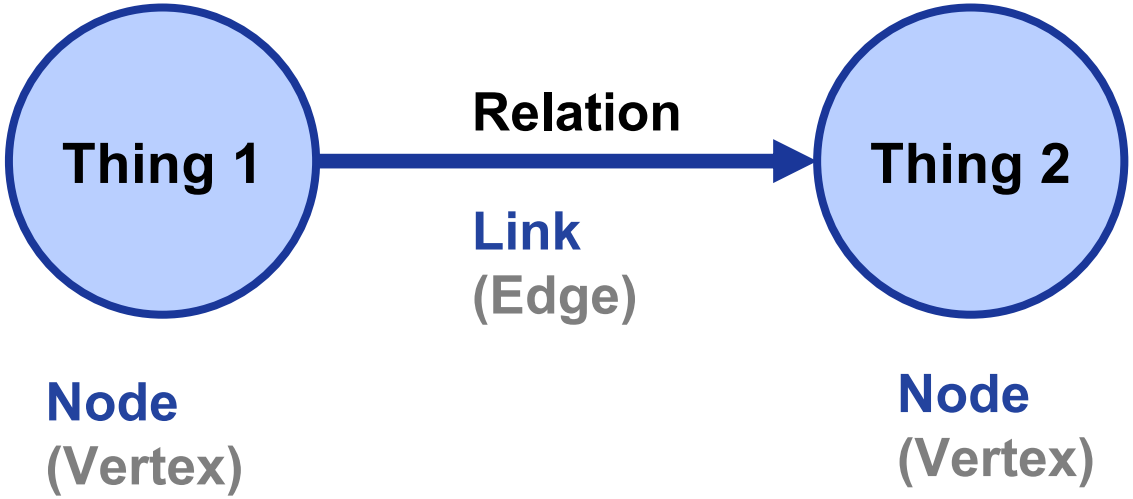
Languages for Working with Graph Data
- **Haskell**
- **JavaScript**
- **Perl**
- **Python**
- **Ruby**
- **SAS**
- **Java**
- **C++**
- **Scala**
- **Go**
- **Rust**
- **Elixir**
- **Erlang**
- **PHP**
- **R**
- **D**
- **Clojure**

...and many more

Ontology and Schema
- Web Ontology Language (OWL)
- Resource Description Framework in Schema (RDFs)
- schema.org
- Ontologies for Life Sciences

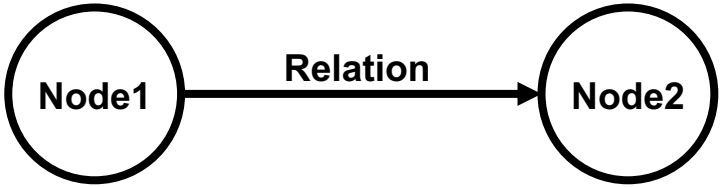
RDF Syntax
- RDF-XML
- RDFa (attributes)
- Notation3 (N3)
- N-Triples
- Turtle (TTL)

KG Foundation: Graph Data

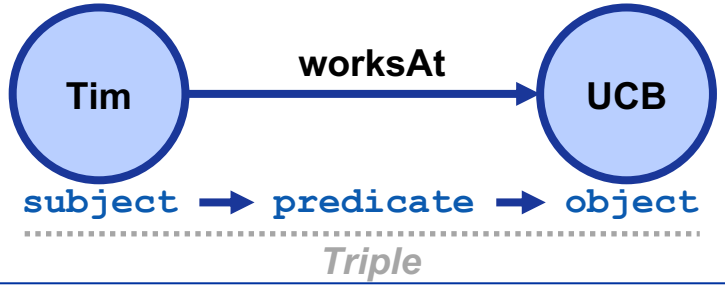


Two Main Types of Graphs

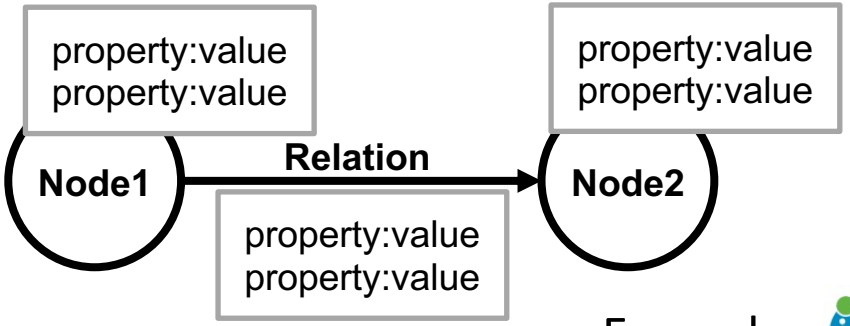
Resource Description Framework (RDF)



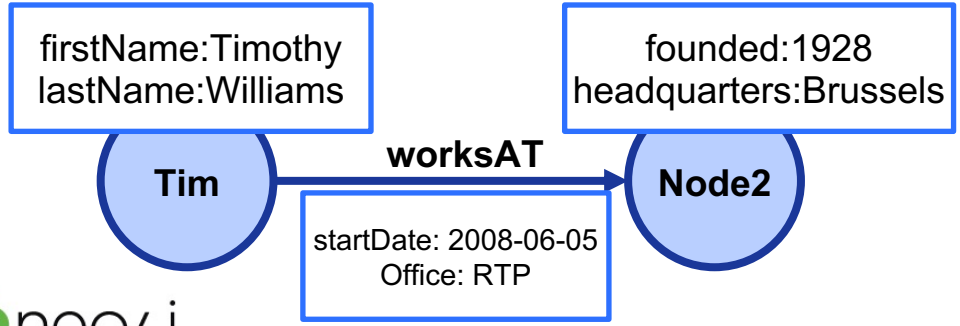
Example: Stardog



Labeled Property Graph (LPG)



Example: neo4j



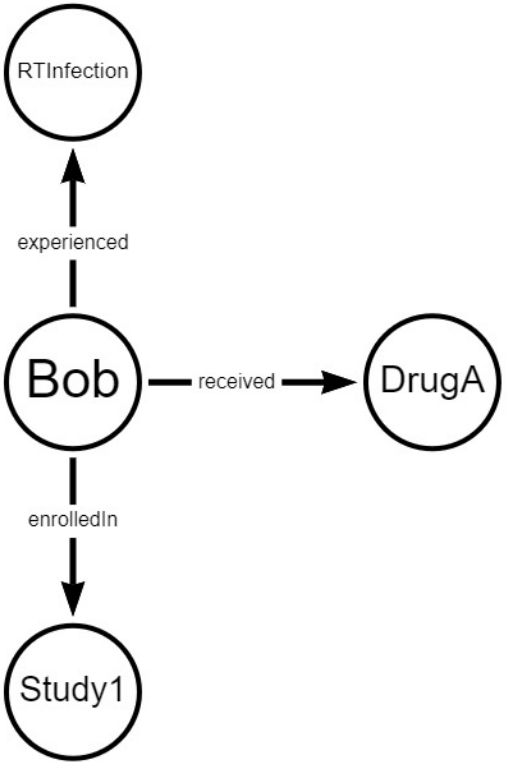
Which Type of Graph is Best?

~~*We will not have this fight today*~~.

There will be some "fighting words" today.

RDF bias in effect!

Whiteboard Language = Graph



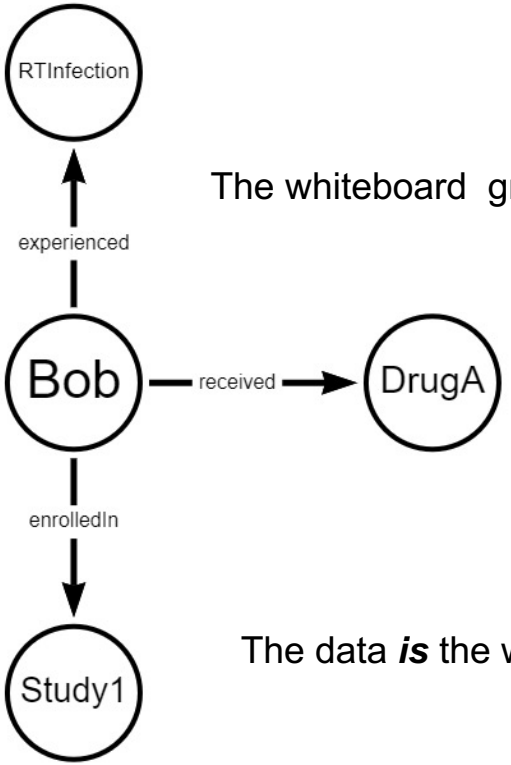
https://arrows.app/

Whiteboard Language *is* the Data Language

The graph is the *data*,
The data is the *graph*.

Terse Triple Language (TTL)

__RDF__ Graph data in abbreviated
__T__erse __T__riple __L__anguage (TTL)



The whiteboard graph ***is*** the data

```
:Bob   :enrolledIn   :Study1 ;
       :received     :DrugA ;
       :experienced  :RTInfection. .
```

The data ***is*** the whiteboard graph

RDF/XML

TTL

```
:Bob :enrolledIn   :Study1 ;
     :received     :DrugA ;
     :experienced  :RTInfection .
```

- Lightweight
- Human readable
- Based on JSON format
- Ideal for web services, programming, unstructured databases

RDF/XML

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:ns0="https:example.org/foo#">

  <rdf:Description rdf:about="https:example.org/foo#Bob">
    <ns0:enrolledIn rdf:resource="https:example.org/foo#Study1"/>
    <ns0:received rdf:resource="https:example.org/foo#DrugA"/>
    <ns0:experienced rdf:resource="https:example.org/foo#RTInfection"/>
  </rdf:Description>
</rdf:RDF>
```

JSON-LD

TTL

```
:Bob :enrolledIn   :Study1 ;
     :received     :DrugA ;
     :experienced  :RTInfection .
```

https://json-ld.org

- Lightweight
- Human readable
- Based on JSON format
- Ideal for web services, programming, unstructured databases

JSON-LD

```
[ {"@id":":Bob",
    ":enrolledIn":  [{"@id":":Study1"}],
    ":received":    [{"@id":":DrugA"}],
    ":experienced":[{"@id":":RTInfection"}]
  }
]
```

JSON-LD Languages

C#
- dotNetRDF
- json-ld.net

Erlang / Elixir
- jsonld-ex

Go
- JSON-goLD

Java
- Titanium
- JSONLD-JAVA

Javascript
- jsonld.js
- jsonld-streaming-parser.js
- jsonld-streaming-serializer.js
- rdf-parse.js

PHP
- php-json-ld
- JsonLD

Python
- PyLD
- RDFLib-jsonld

R
- jsonld

Ruby
- JSON-LD for RDF.rb

TypeScript
- jsonld-lint

Rust
- json-ld

Source: json-ld.org

RDFa Resource Description Framework in Attributes

RDFa in Web Page about DrugA

```
<a about="https://example.org/foo#DrugA"
   property="https://example.org/foo#adverseReaction"
   href="https://example.org/foo#RTInfection">RT Infections as DrugA AE's
</a>
```

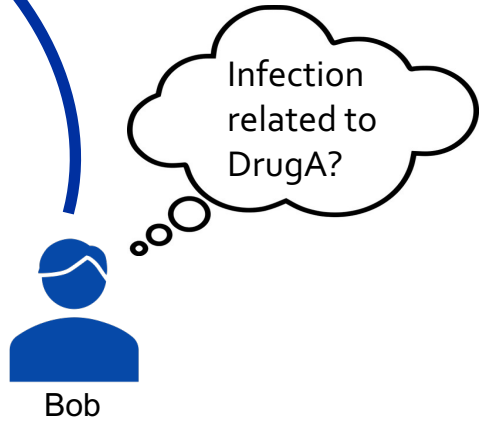
- Web page information consumable by the machine
- ***Structured Data*** added to web pages.

Extracted TTL

```
:DrugA :adverseReaction :RTInfection
```

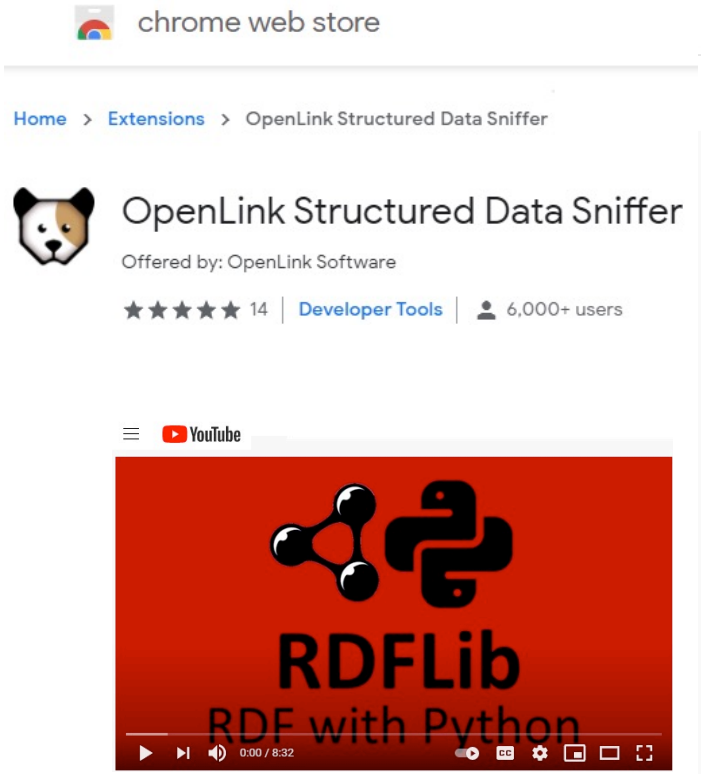
Study TTL

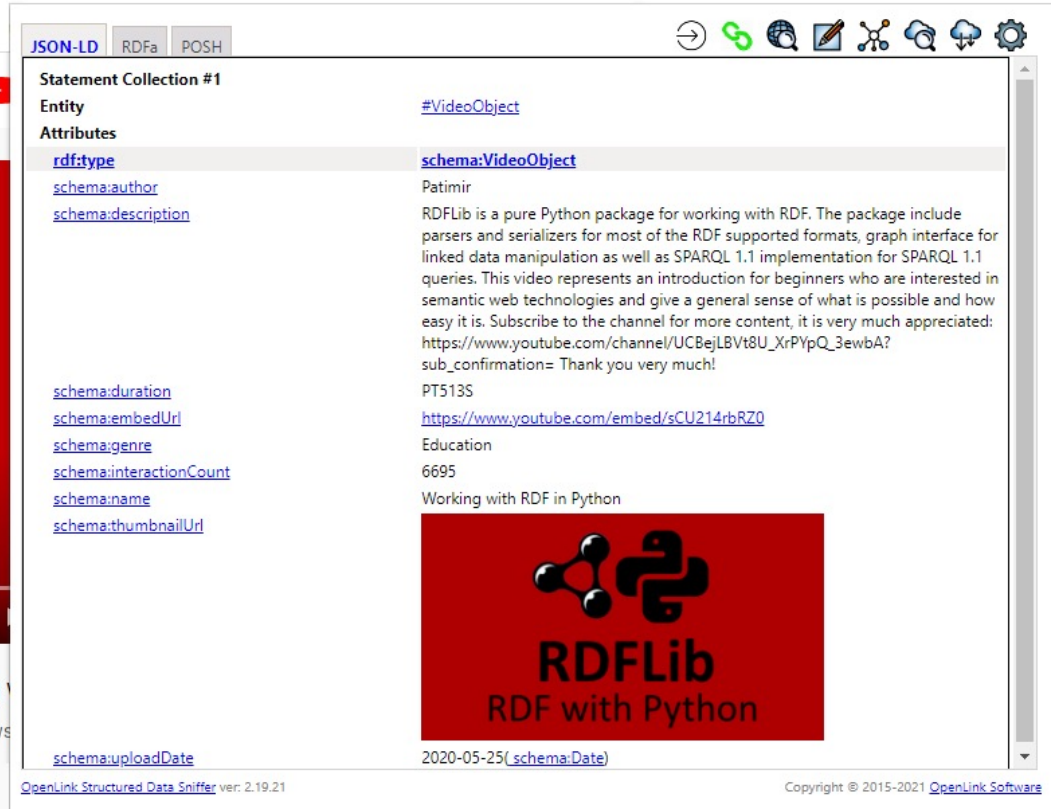
```
:Bob :enrolledIn   :Study1 ;
     :received     :DrugA ;
     :experienced  :RTInfection .
```



https://www.w3.org/TR/rdfa-primer/

Finding RDF in Web Pages





Graph Query Languages

Query Languages

RDF

- SPARQL
- “Learning SPARQL”

Bob Ducharme
@bobdc
www.bobdc.com

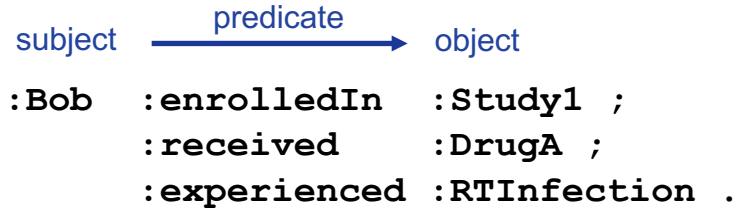
Labeled-Property Graph (LPG)

- GQL
  - ISO certification process underway (2021)
- Numerous Open or Proprietary languages

Neo4j	Cypher openCypher
TigerGraph	GSQL
Oracle	PGQL
Apache Tinkerpop	Gremlin
Linked Data Benchmark Council (LDBC)	G-CORE

SPARQL Query Language

Data



Query

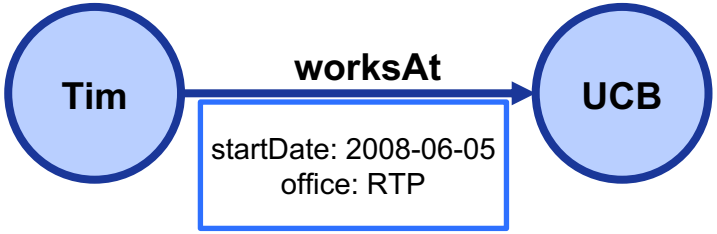
```
SELECT ?from ?relation ?to
  WHERE {
    ?from ?relation ?to  .
}
```

Query Result

from	relation	to
:Bob	:enrolledIn	:Study1
:Bob	:received	:DrugA
:Bob	:experienced	:RTInfection

RDF* (*RDF-Star) and SPARQL* (SPARQL-Star)

When you need to say something about triples



Data

```
<< :Tim  :worskAt :UCB >>
  :startDate  "2008-06-05"^xsd:date ;
  :office         :RTP .
```

Query

```
SELECT ?name ?company
?companyOffice ?startDate
WHERE{
  <<?name :worksAt ?company>>
   :startDate ?startedOn ;
   :office  ?companyOffice .
}
```

Query Result

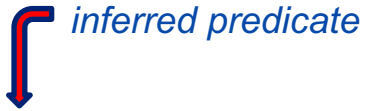
name	company	companyOffice	startDate
:Tim	:UCB	:RTP	2008-06-05

RDF* (*RDF-Star) and SPARQL* (SPARQL-Star)

Cautions

- New
- Is not a W3C Standard (as of 2021-09)
- Implementation details vary across platforms

  Eg: Inference not yet supported on edge properties with Stardog.The query below will not work because `:hasPrevious` is an inferred predicate that cannot be used to obtain the second set of embedded triples in the query:




```
WHERE{
  << :FileA :writesFile :FileB >> :hasPrevious+ ?prevStep .
   <<?file1 ?action ?file2 >> :hasStep ?prevStep
}
```

fails

Graph Query Language Future

The future of query languages is tied to the future of Graph data.

- Likely a combination of concepts from RDF and LPGs.
  - Example: Vaticle (https://vaticle.com/) using TypeDB/TypeQL

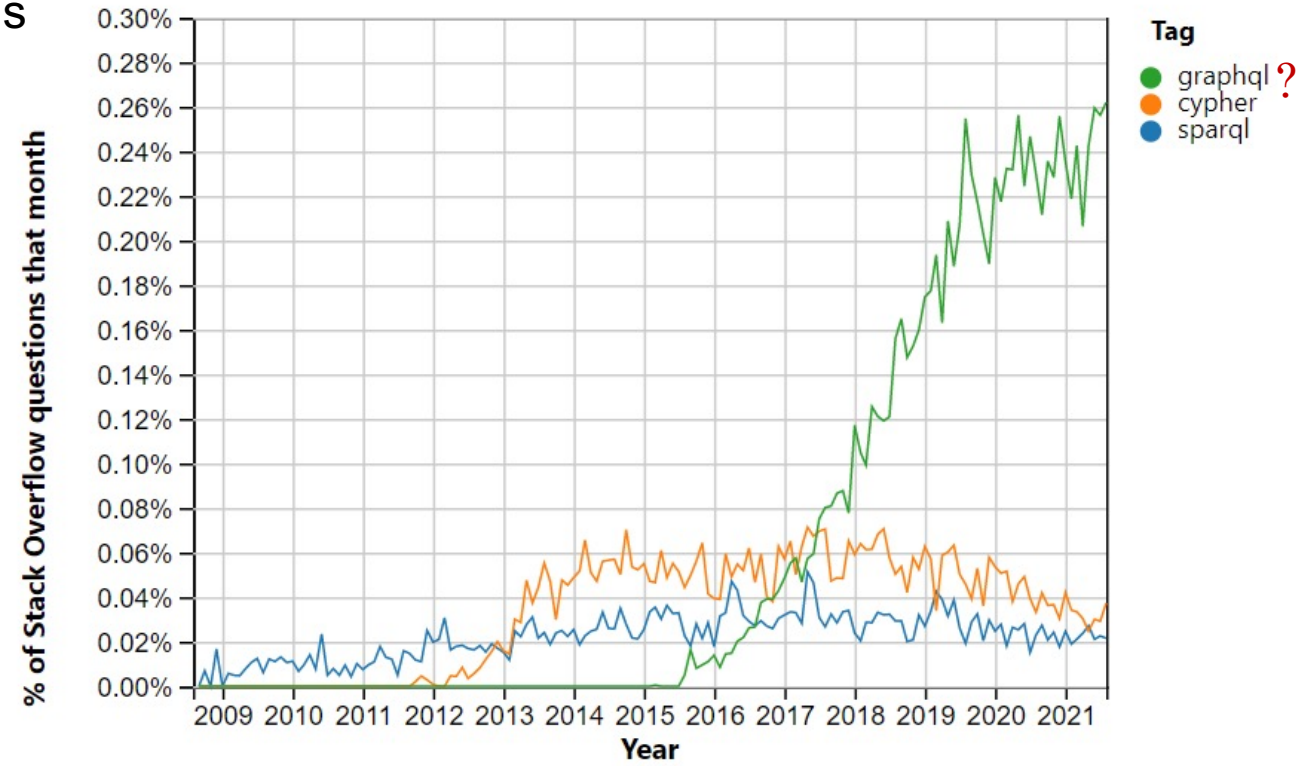
Graph Query Language Trends

Stack Overflow Trends

Tags

sparql × cypher ×

graphql × ?



☆GraphQL

- “data query and manipulation language for APIs” - Wikipedia
- Great for Web APIs
- Not the best for “graph-type” queries.
- Example: Unable to retrieve all parents of a node.

Modeling Languages

Modeling Languages

RDF Shema

- Built-in using RDF primitives (S,P,O)
  - Semantic. Lives with the data
- RDF Schema Language (RDFs)
  - Classes and Subclasses
- Web Ontology Language (OWL)
- SHACL
- Tools:
  - Text Editor
  - Graphical Editor : Gra.fo = graphical editor by data.world
  - Structured View – **Protege**, Topbraid
  - Taxonomy focus - PoolParty

LPG Schema

- No widely accepted schema language*
  - Work in progress by the Property Graph Schema Working Group community

* Source: Sequeda and Lassiila, 2021

Ontology

What is an Ontology?

- **Dictionary** – terms and definitions
- **Taxonomy** – class hierarchy
- **Thesaurus** - relationships between terms
  - Use a reasoner to ***infer values and relationships*** not in the original data
  - **Rules and Restrictions** - Group membership, exclusions, types

Free e-book
A Data Engineer's Guide to Semantic Modelling
- Ilaria Maresi (June 2020)
http://blog.thehyve.nl/news/ebook-semantic-model

Ontologies for Life Sciences

BioPortal – Biomedical Ontologies
https://bioontology.org/

Open Biological and Biomedical Ontology (OBO Foundry)
http://www.obofoundry.org/

Beware the Ontology Design Quagmire

Follow a design philosophy

Ontology Development 101 - Noy and McGuinness (2001)

Mapping Languages

From Relational to Graph

RDF

- R2RML
  - SMS (Stardog)
- Commercial
  - Gra.fo (data.world)
  - Metaphactory
  - Data Lens
- Open Source
  - - Karma https://github.com/usc-isi-i2/Web-Karma
  - - RML https://github.com/RMLio/rmlmapper-java
  - - ....and others
- “Roll your own”
  - - R, Python...

LPG

- No known mapping languages*
- “Roll your own”

* Source: Sequeda and Lassiila, 2021

Validation Languages

RDF

- RDFs
  - - classes , subclasses
- SPARQL
- OWL
  - - classes (RDFs), rules, domain, range...
- **SHA**pes **C**onstraint **L**anguage (SHACL)
- **Sh**ape **Ex**pressions (ShEx)

LPG

- No non-proprietary validation language*

* Source: Sequeda and Lassiila, 2021

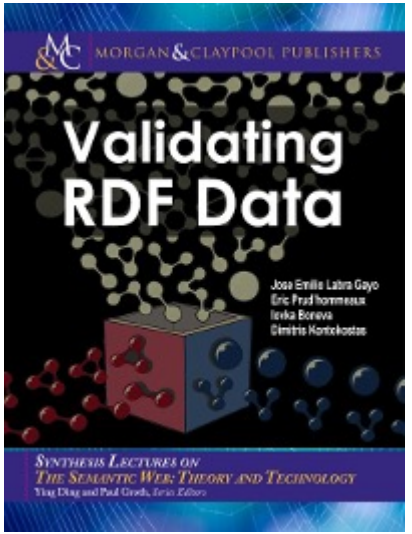
Validation Languages

Validating RDF Data (Book)

- *Free online or hardcopy*

https://book.validatingrdf.com/

- *ShEx and SHACL*
- *Good introduction to RDF*
  - *Not a good first book for RDF*



SHACL

- Graph data is compared and validated against a graph pattern (shape)
- Shape = set of constraints the graph data must/should satisfy

Shapes can apply to

- **Nodes** (Subject, Object)
- **Relations** (Predicates)
  - property definitions/constraints

Languages for working with KG's

- Haskell
- JavaScript
- Perl
- Python
- Ruby
- SAS
- Java
- C++
- Scala
- Go
- Rust
- Elixir
- Erlang
- PHP
- R
- D
- Clojure

...and many more

R For *Creating* RDF

Libraries

- rdflib
- redland
- jsonld
- digest # SHA1 hashing for IRI formation
- tools # md5sum hashing: IRI formation, file content changes

R For *Creating* RDF

1. Data to R dataframe
2. Data Massage
   a. Create IRIs in data
   b. Create RDF according to required model

Option 1: Traditional R Dataframe loop

```
srcDf <- patientData
for(i in 1:nrow(srcDf)){
  rdf_add(some_rdf,
    subject   = paste0(STUDY, "PATIENT_", srcDf[i,"patID_h"]),
    predicate = paste0(STUDY, "received"),
    object    = paste0(STUDY, srcDf[i,"drug"])
  )
}
```

Option 2: Tidyverse Approach

"A tidyverse lover's intro to RDF" - Carl Boettiger

3. Write out to TTL file
4. Validate TTL file with SHACL
4. Upload to graph database

R For *Querying* RDF

Library

- SPARQL

Option 1: Statement in R Code

```
sparql <-
  'SELECT  ?patientID ?treatment
   WHERE {
     ?patient :hasTreatment ?treatment .
   }'
foo <- rdf_query(rdf, sparql)
```

Option 2: Source external .RQ file

```
foo <- rdf_query(rdf,
                 read_file("C:/myKGProject/PatientTreatments.rq"))
```

R For *Analyzing* RDF

Similar to other data analysis workflows

1. Query result to R dataframe
2. Analysis like any other R Dataframe

“Analysis with benefits”

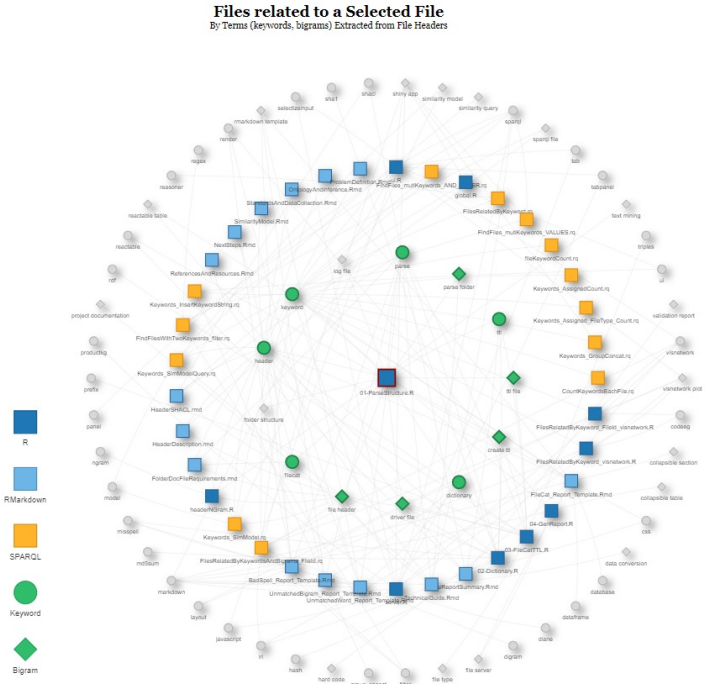
- *Values link to other values*
- *Semantic relationships*
- *Integral metadata*
- *Superior data quality*
  - *Unique identifiers (IRIs)*
  - *SHACL*

R For *Visualizing* RDF

Libraries

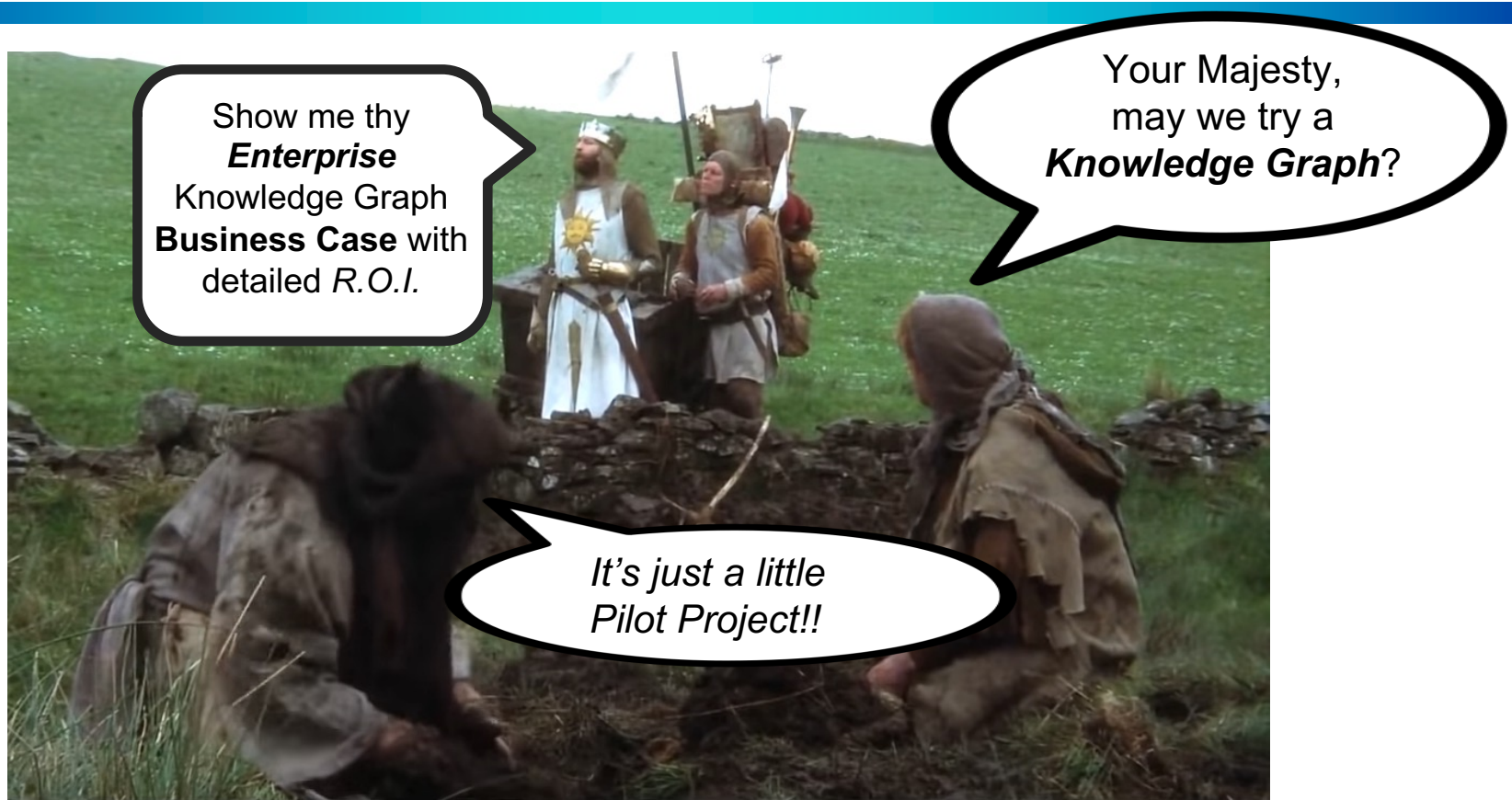
- visnetwork
- ggplot2
- plotly
- DiagrammeR
- r2d3 (for D3JS)
- json, jsonlite

  - Export to JSON for consumption by D3JS and others.



"Visualizing Knowledge Graphs: Information Solution or Ball of Confusion"
- Tim Williams, EUConnect21 (November)

Communication (Human Language)

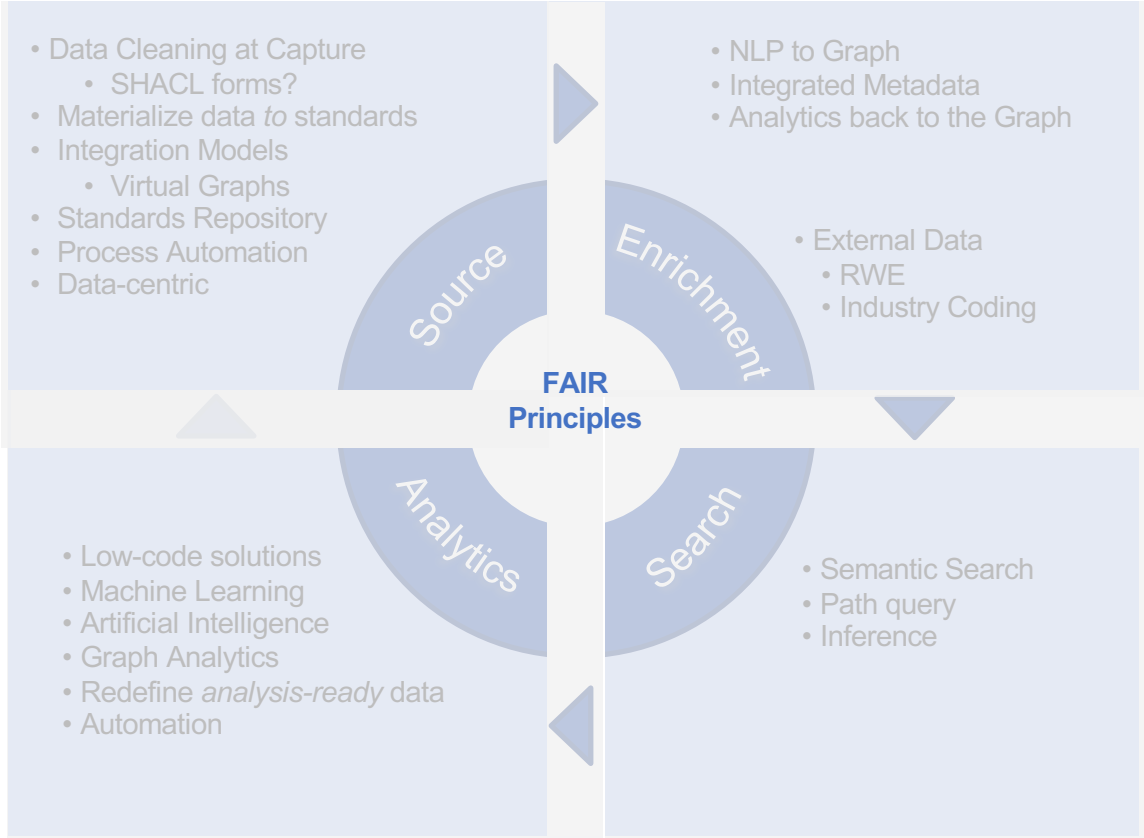


Learn Languages by Dogfooding

https://en.wikipedia.org/wiki/Eating_your_own_dog_food

Coming Soon(?) : "A Gateway to Knowledge Graphs using R" – Tim Williams (to be published online)

The Language of F.A.I.R. Data



Adapted from:
https://www.slideshare.net/AlanMorrison/datacentric-business-transformation-using-knowledge-graphs

Thank you!

The Global Healthcare Data Science Community

Contact Channels

UK +44 1843 609600
office@phuse.global
phuse.global

Social Media

@phusetwitta
/phusebook
/phusetube
/company/phuse