Knowledge Graph Construction from Heterogeneous Data Sources Exploiting Declarative Mapping Rules

PhD Defense

David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain

Supervisor: Oscar Corcho

dchaves@fi.upm.es dchavesf

16:00 - 28/06/2021 Online

cc BY SA

Contents

Contents

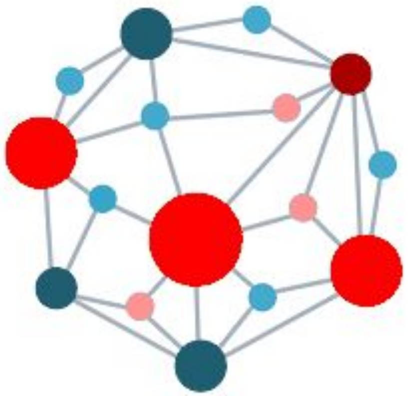
"Your impact will be as big as the quality of your pitch to explain the solution"

Pieter Colpaert

KG Embeddings

Explainable AI

Multilinguality

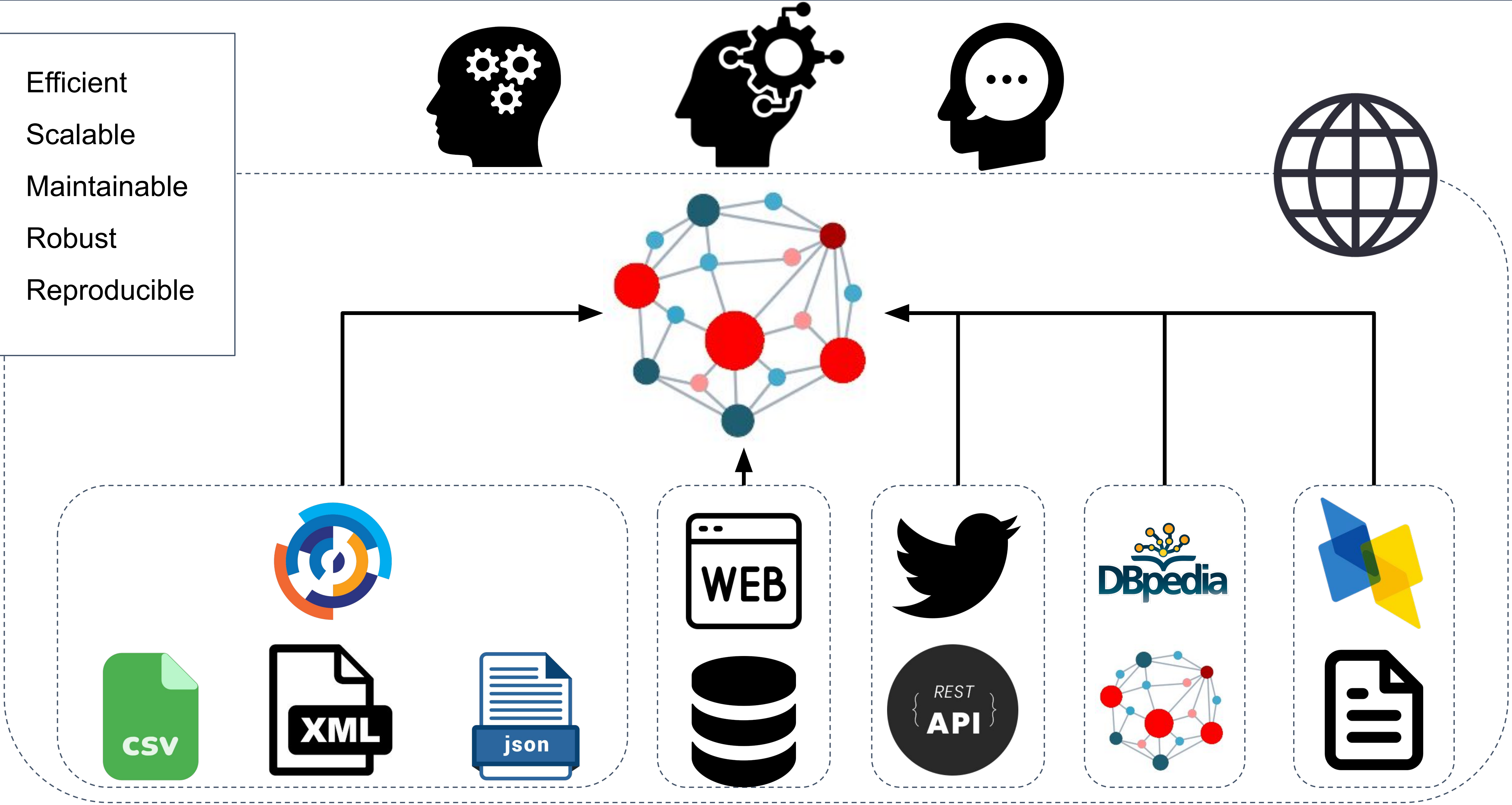


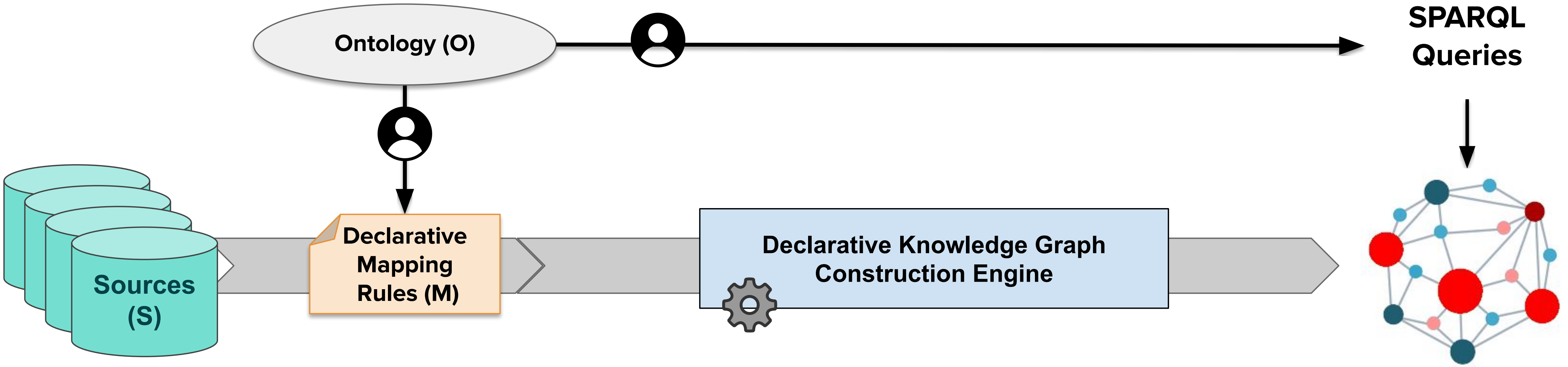
Question Answering Systems

Search Interfaces

Data Labelling

- Efficient
- Scalable
- Maintainable
- Robust
- Reproducible

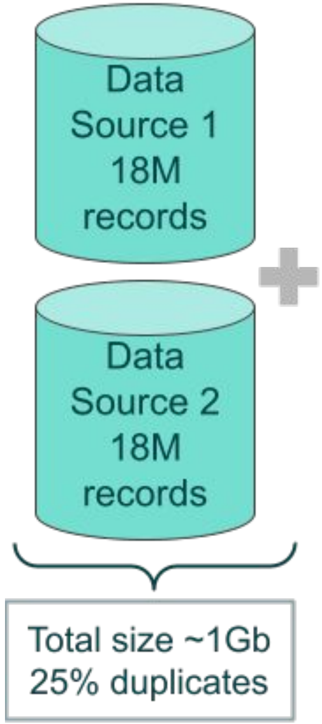




Knowledge Graph Construction = Data Integration System (DIS) = <S, M, O>

Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., & Rosati, R. (2008). Linking data to ontologies. In *Journal on data semantics X*
Lenzerini, M. Data integration: A theoretical perspective. In *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*

Challenge I: KG construction in complex DIS




```
<TriplesMap1>
 rml:logicalSource [ rml:source "dataSource1" ];
 rr:subjectMap [
   rr:template "http://iasis.eu/{acc}_{enst}";
   rr:class iasis:RBP_RNA_PhysicalInteraction ];

 rr:predicateObjectMap [
   rr:predicate iasis:interactionScore;
   rr:objectMap [ rml:reference "omixcore" ] ];

 rr:predicateObjectMap [
   rr:predicate iasis:interaction_involves_RBP;
   rr:objectMap [
     rr:parentTriplesMap <TriplesMap2> ] ].

<TriplesMap2>
 rml:logicalSource [ rml:source "dataSource1" ];
 rr:subjectMap [
   rr:template "http://iasis.eu/Protein/{acc}";
   rr:class iais:Protein] ;

 rr:predicateObjectMap [
   rr:predicate iasis:protein_isRelatedTo_exon;
   rr:objectMap [
     rr:parentTriplesMap <TriplesMap3>;
     rr:joinCondition [ rr:child "enst"; rr:parent "enst" ;];].

<TriplesMap3>
 rml:logicalSource [ rml:source "dataSource2" ];
 rr:subjectMap [
   rr:template "http://iasis.eu/Exon/{ense}";
   rr:class iais:Exon ] .
```

RDF Knowledge Graph Creation

RocketRML : Memory Failure

RMLMap

Efficiency
Scalability

Challenge II: Querying messy tabular data

Obtained | Expected

```
SELECT ?stop_name ?date1 ?date2
WHERE {
    ?stop1 gtfs:sameStop ?stop2
    ?stop1 gtfs:name ?stop_name
    ?stop1 gtfs:close_date ?date1
    ?stop2 gtfs:close_date ?date2
    FILTER (?date1 != ?date2)
}
```

?stop_name	?date1	?date2
Noviciado	20191225	20191231-20200101
Colonia_Jardin	2019-12-25	2019-12-31
Plaza_de_españa	2020-01-01	2020-01-06
Noviciado	2020-12-25	2019-12-31
Noviciado	2019-12-25	2020-01-01

```
BusStop(w(id))                  ← bus_stop(id,name,date)
MetroStop(w(id))                ← metro_stop(id,name,date,wheelchair)
name(w(id),name)                ← metro_stop(id,name,date,wheelchair)
close_date(w(id),close_date)    ← metro_stop(id,name,date,wheelcha
name(w(id),name)                ← bus_stop(id,name,date)
close_date(w(id),close_date)    ← bus_stop(id,name,dat
wheelchair(w(id),wheelchair)    ← metro_stop(id,name
sameStop(w(id),u(id))           ← metro_stop(name), b
```

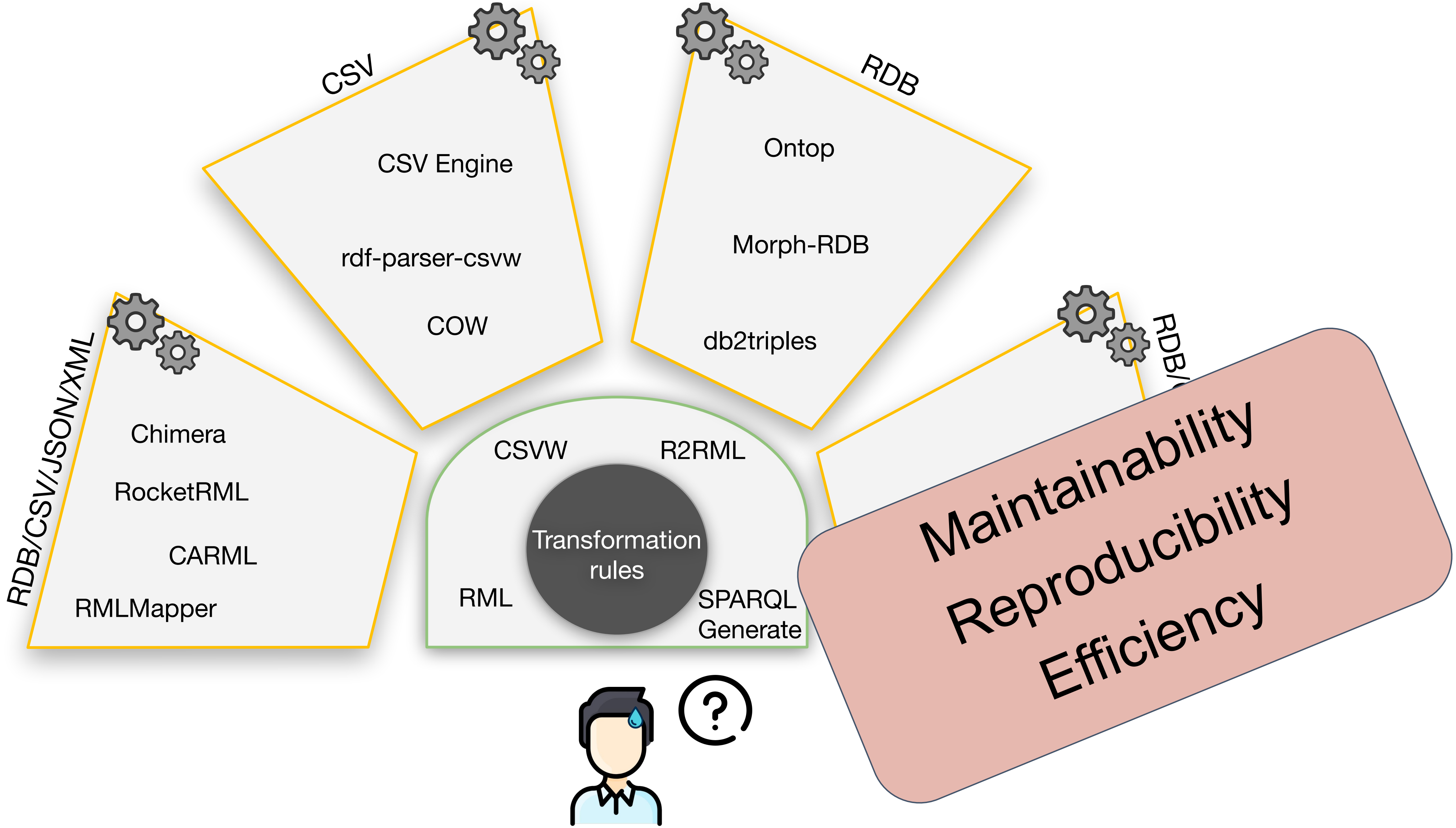
metro_stop.csv

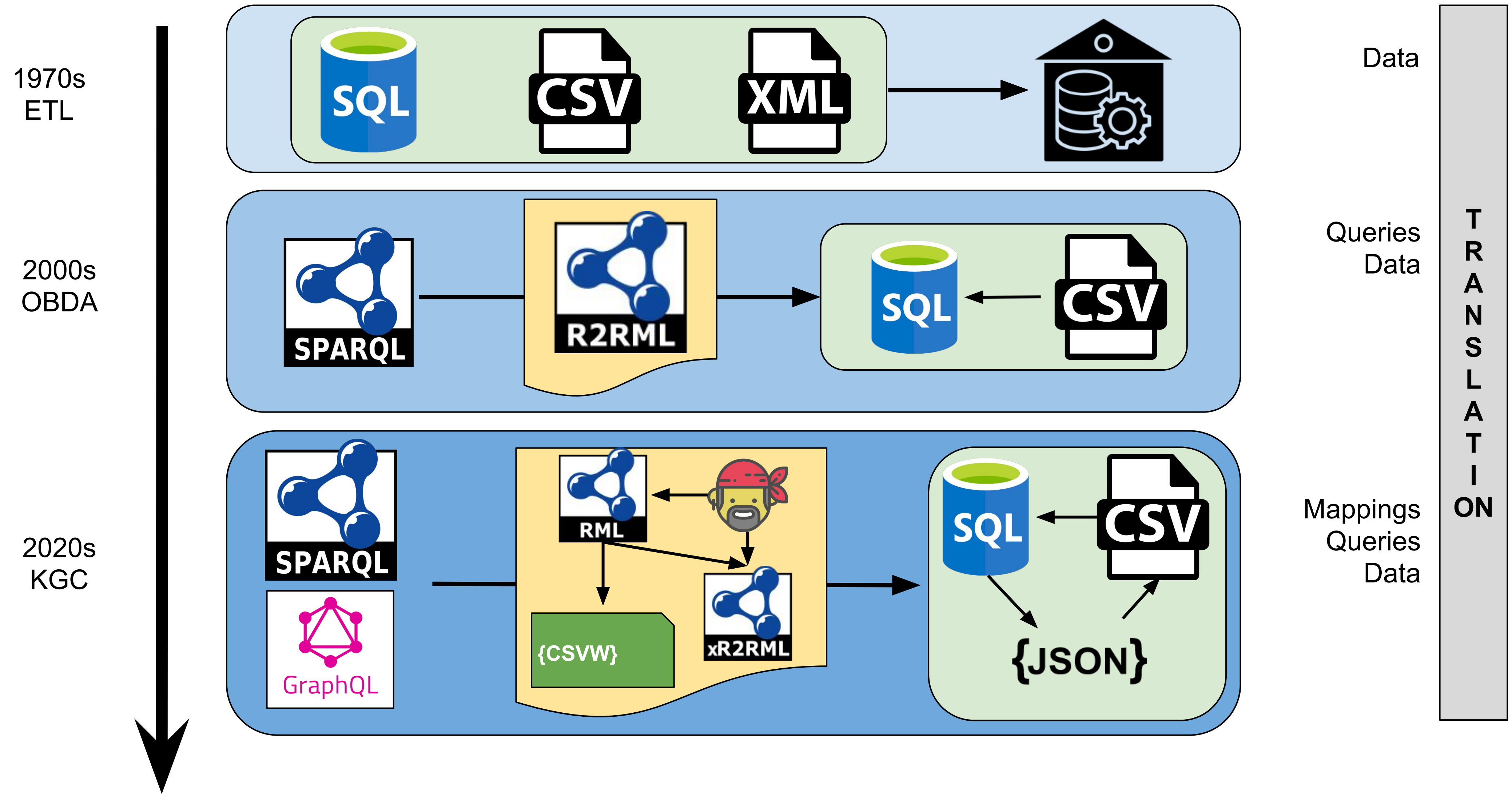
id	name	date	wheelchair
1	Colonia_jardin	20191225	0
2	Plaza_de_españa	20200101	1
3	Noviciado	20191225	0

id	nan	
1	Colonia Ja	231
2	Plaza De Esp	0101-20200106
3	Noviciado	20191231-20200101

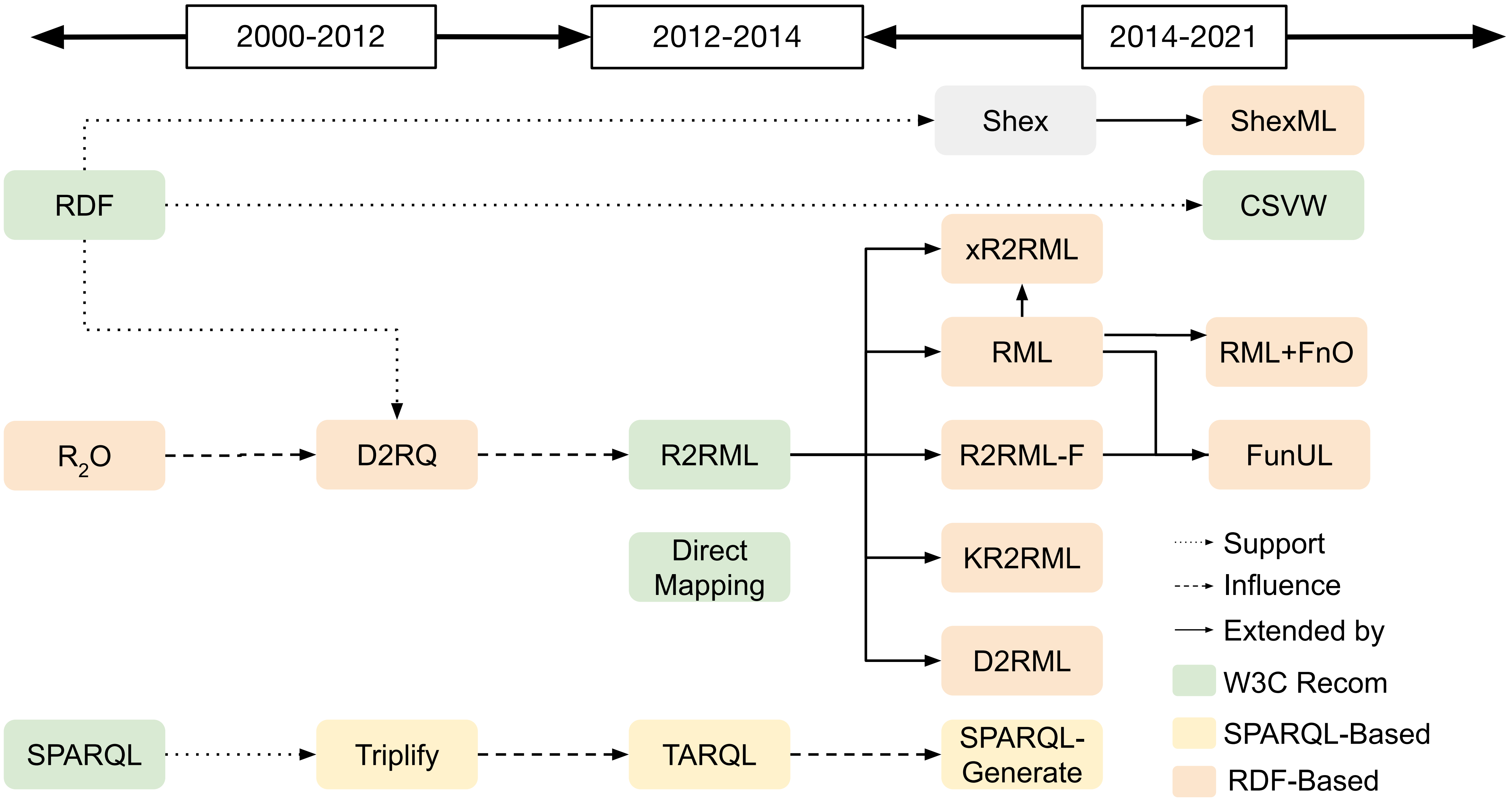
Maintainability
Reproducibility
Robustness

Challenge III: Choosing mappings and engines



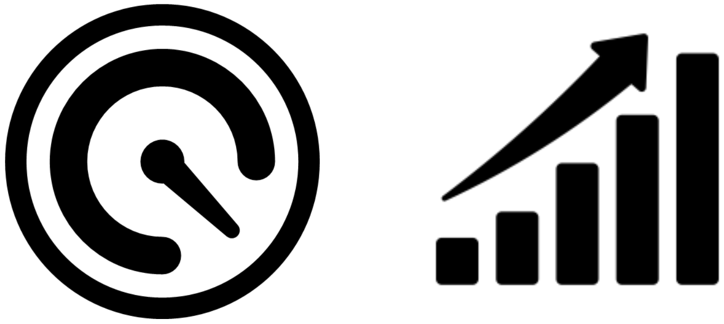

1970s
ETL
2000s
OBDA
2020s
KGC
SQL
CSV
XML
SPARQL
R2RML
SQL
CSV
SPARQL
GraphQL
RML
{CSVW}
xR2RML
SQL
CSV
{JSON}
Data
Queries
Data
Mappings
Queries
Data
TRANSLATION

Contents

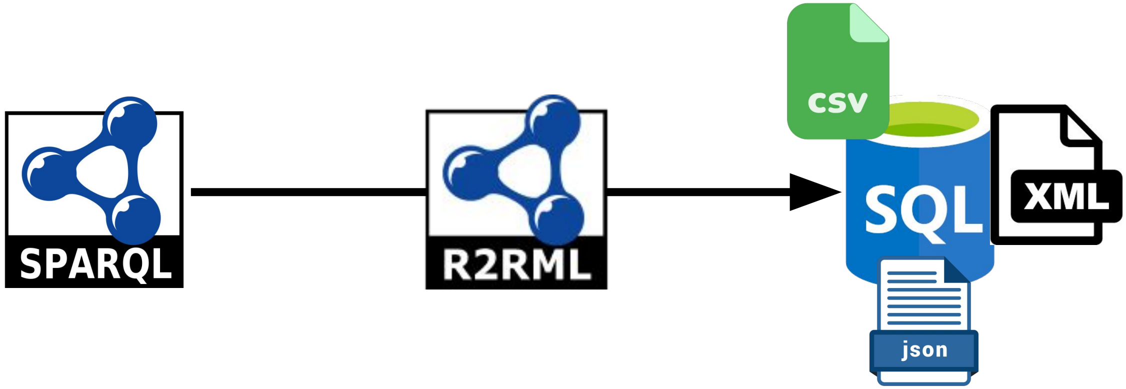

2000-2012
2012-2014
2014-2021
Shex
ShexML
RDF
CSVW
xR2RML
RML
RML+FnO
R2O
D2RQ
R2RML
R2RML-F
FunUL
Direct Mapping
KR2RML
D2RML
SPARQL
Triplify
TARQL
SPARQL-Generate
Support
Influence
Extended by
W3C Recom
SPARQL-Based
RDF-Based

Engine	Type	Mapping Rules	Written In	Evaluation	Limitation(s)
RMLMapper	Materializer	RML (+FnO) + CSVW	Java	-	Performance&Scalability
RocketRML	Materializer	RML (+FnO)	JavaScript	Ad-hoc	Memory consumption
CARML	Materializer	RML (+FnO)	Java	-	Scalability
Chimera	Materializer	RML	Java	-	Memory consumption
SPARQL-Generate	Materializer	SPARQL-Generate	Java	Ad-hoc	Scalability
CSV2RDF	Materializer	CSVW	Java	-	Scalability
COW	Materializer	CSVW	Python	-	Scalability
Ontop	Materializer & Virtualizer	R2RML & OBDA	Java	BSBM, NPD	Support for RDB
Morph-xR2RML	Materializer & Virtualizer	xR2RML	Java	-	Scalability & SPARQL
Morph-RDB	Materializer & Virtualizer	R2RML	Java	BSBM	Support for RDB
Squerall	Virtualizer	RML (+FnO)	Java	BSBM	SPARQL operators
Ontario	Virtualizer	RML	Python	LSLOD	SPARQL operators

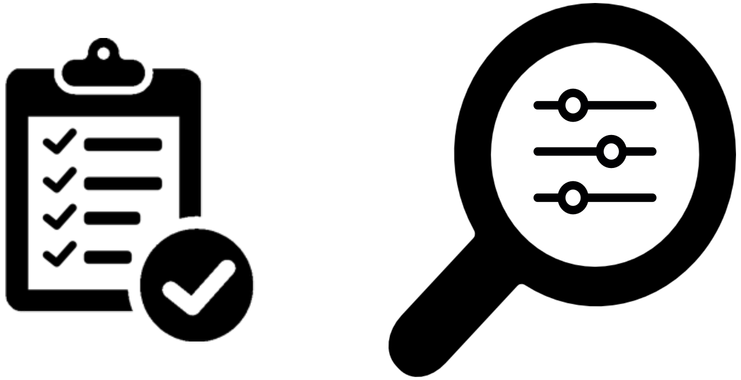
GLOBAL ADOPTION OF DECLARATIVE KNOWLEDGE GRAPH CONSTRUCTION SYSTEMS



Limitation I: Performance and scalability issues



Limitation II: Virtual Knowledge Graph Access Beyond RDB



Limitation III: Lack of comprehensive and unified evaluation

Limitation IV: Multiple mappings specs with specific features

- Introduction
- State of the Art
- **Research Methodology & Thesis Objectives**
- Contributions
  - C1: Knowledge Graph Construction at Scale
  - C2: Evaluation Framework for Knowledge Graph Construction
- Conclusions and Future Work

"As a computer scientist, the most important part of a research is the what, not the how"

Maria-Esther Vidal

Identification of Limitations

Four limitations based on the analysis of the state of the art

Definition of the objectives

Objective 1
Scalable and efficient construction of Knowledge Graphs

Objective 2
Evaluation methodologies of KGC systems

↓

Global adoption of declarative knowledge graph construction approaches

Definition of the hypotheses

Hypotheses 1,2,4,5:
Four hypotheses associated to the first objective

Hypothesis 3:
One hypothesis associated to the second objective

↓

Mapping rules are the key to integrate knowledge from heterogeneous data sources

Definition of the contribution

Contributions 1.[1-5]:
Five contributions associated to the first objective

Contributions 2.[1-3]:
Three contributions associated to the second objective

↓

New generation of knowledge graph construction systems

Research Methodology

Objective 1: Scalable and efficient construction of Knowledge Graphs

Hypothesis 1: It is possible to **translate declarative mapping rules** among different specifications.

Hypothesis 2:
The exploitation of **declarative annotations can enhance current virtual KGC** systems

Hypothesis 4:
Physical data structures and operators can be defined **for scaling up KGC engines**

Hypothesis 5:
Optimizations for functional mapping rules can be applied to scale up the KGC

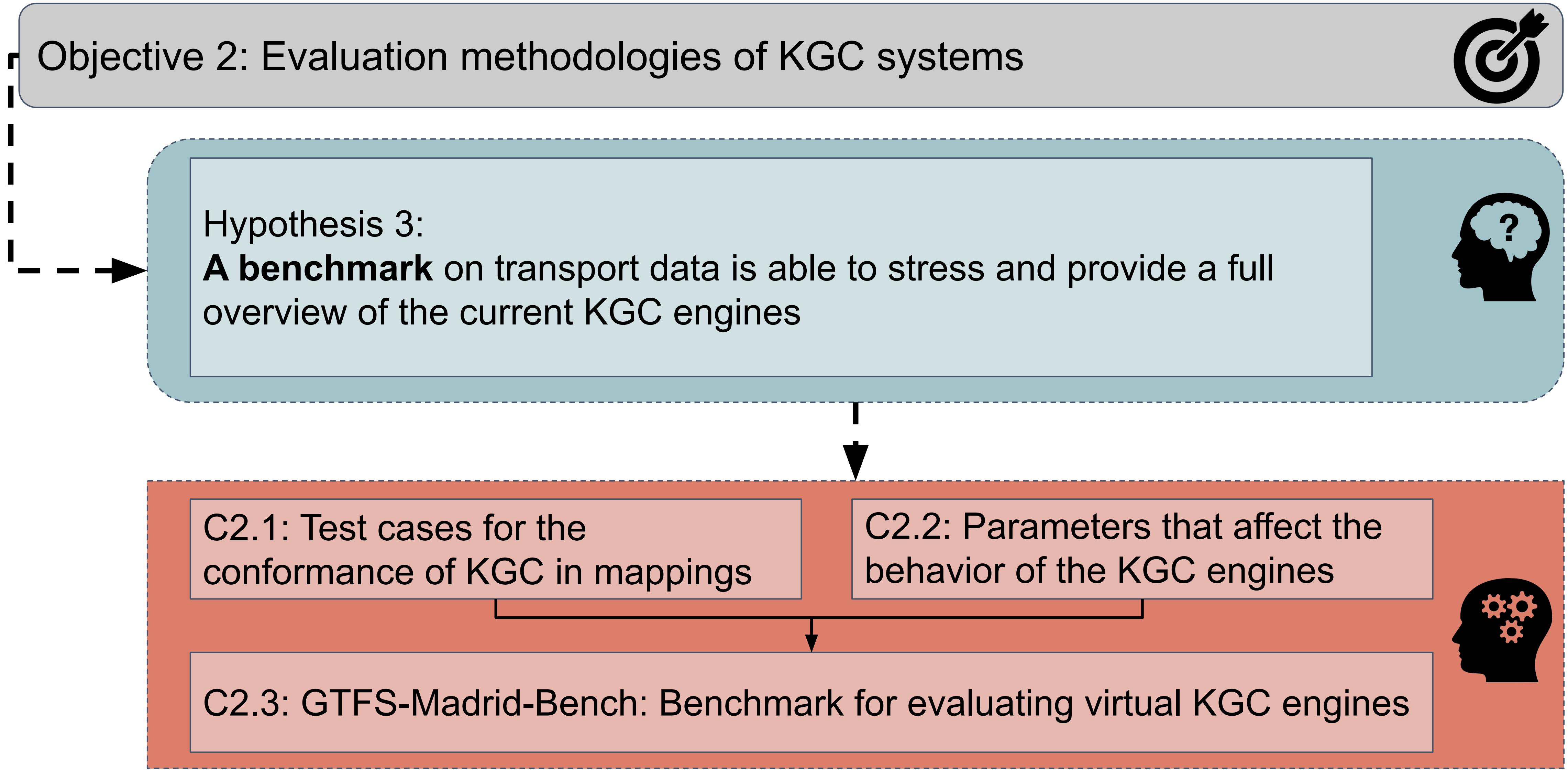
Contribution 1.2:
Extracting constraints from declarative mapping for enhancing VKGC

Contribution 1.3:
Automatic creation of functional wrappers from declarative mapping rules

Contribution 1.4:
Physical structures and associated operators for optimizing KGC

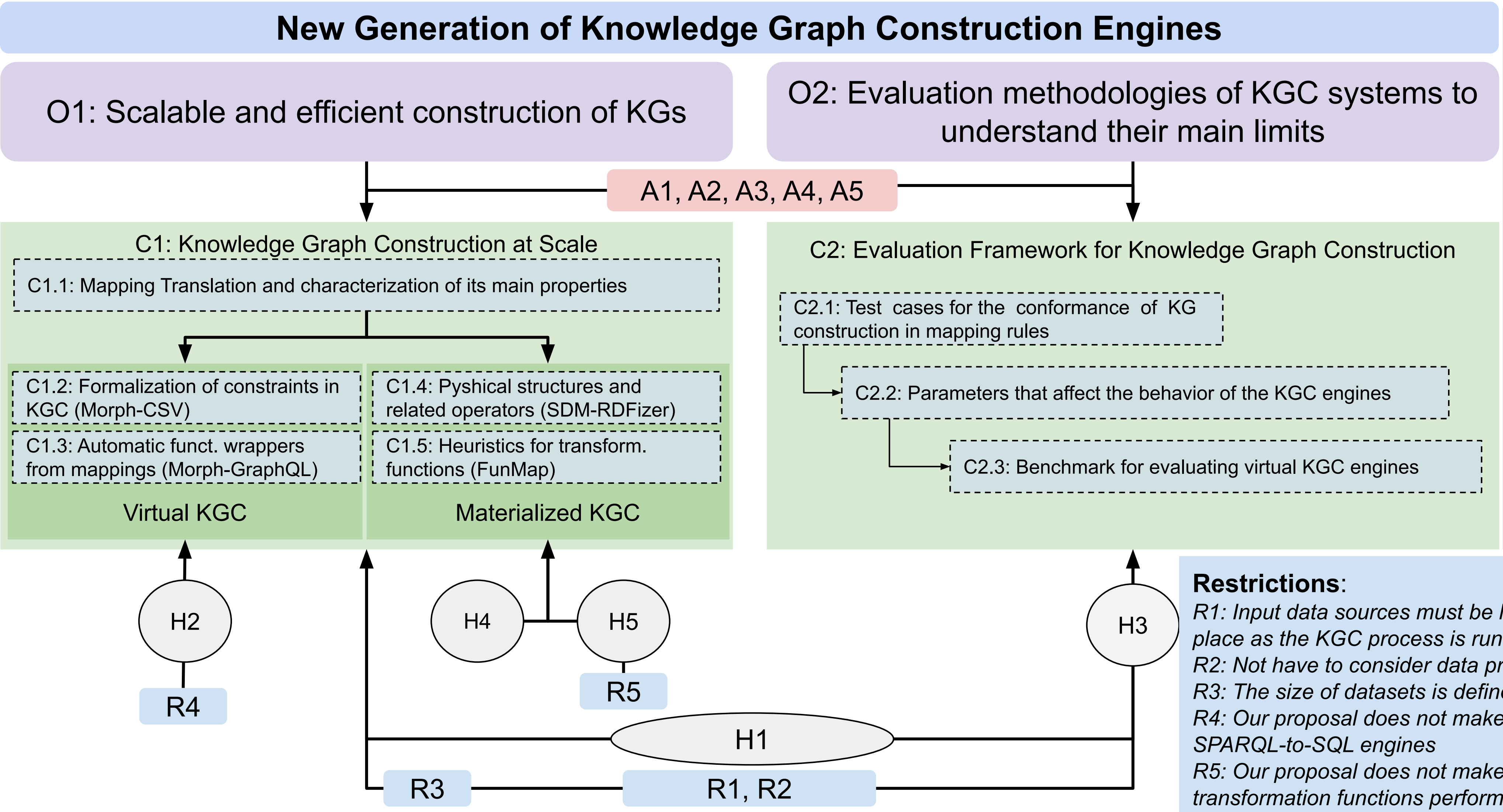
Contribution 1.5:
Heuristic-based approach for transformation functions in mapping rules

Contribution 1.1: **Mapping translation concept** and characterization of its main properties


Objective 2: Evaluation methodologies of KGC systems
Hypothesis 3:
A benchmark on transport data is able to stress and provide a full overview of the current KGC engines
C2.1: Test cases for the conformance of KGC in mappings
C2.2: Parameters that affect the behavior of the KGC engines
C2.3: GTFS-Madrid-Bench: Benchmark for evaluating virtual KGC engines

Summary of objectives and contributions

Assumptions:
A1: Mapping rules and metadata descriptions are declarative and follow W3C standards
A2: The ontology for integrating the source data is available and is implemented in OWL.
A3: Mapping rules and metadata are available
A4: Data are represented in formats that are not RDF
A5: Datasets are static, not streams.



Hypotheses:
H1: It is possible to translate declarative mapping rules among different specifications.
H2: The exploitation of declarative annotations can enhance current virtual KGC systems
H3: A benchmark on transport data is able to stress and provide a full overview of the current state of different KGC engines
H4: Physical data structures and operators can be defined for scaling up KGC engines
H5: Optimizations for functional mapping rules can be applied to scale up the construction of KGC

Restrictions:
R1: Input data sources must be located in the same physical place as the KGC process is run.
R2: Not have to consider data protection nor access restrictions.
R3: The size of datasets is defined in terms of Gigabytes
R4: Our proposal does not make use of the capabilities of SPARQL-to-SQL engines
R5: Our proposal does not make use of the features of the transformation functions performed over the input data sources.

Contents

"The primary goal of a researcher should be to continually challenge science with creative ideas"

Oscar Corcho

- Introduction
- State of the Art
- Research Methodology & Thesis Objectives
- **Contributions**
  - C1: Knowledge Graph Construction at Scale
  - **C2: Evaluation Framework for Knowledge Graph Construction**
- Conclusions and Future Work

C2: Evaluation Framework for Knowledge Graph Construction

Engine Conformance

Engine Behavior

Engine Performance & Scalability

Evaluation Framework for Declarative Knowledge Graph Construction Engines from Heterogeneous Data Sources

Conformance Test Cases for the RDF Mapping Language (RML)

Heyvaert, P., **Chaves-Fraga, D.**, Priyatna, F., Corcho, O., Mannens, E., Verborgh, R., & Dimou, A. (2019). Conformance test cases for the RDF mapping language (RML). In *Iberoamerican KGSWC*. This contribution are the result of joint collaboration with Ghent University as a result of research visits and collaboration work in the context of the W3C

From R2RML to RML test-cases

297 Test Cases covering:

- CSV, JSON, XML, MySQL, PostgreSQL, SQLServer
- Translated **from R2RML Test Cases**
- Semantically described by:
  - Evaluation and Report Language (EARL) 1.0 Schema
  - Test case manifest vocabulary
  - Test Metadata vocabulary
  - Data Catalog vocabulary
- Each Test Case includes: Data + Mapping + Expected Output KG
- Full info: http://rml.io/test-cases/ & https://rml.io/implementation-report/

carml RMLStreamer RocketRML SDM-RDFizer RMLMapper Chimera

Analysis of parameters that affect the construction of Knowledge Graphs

Chaves-Fraga, D., Endris, K. M., Iglesias, E., Corcho, O., & Vidal, M. E. (2019). What are the Parameters that Affect the Construction of a Knowledge Graph?. In *ODBASE.* This contribution is one of the result of joint collaboration with the Scientific Data Management Group from German National Library of Science and Technology (TIB), as a result of a research stay in the institution

Independent Variables

Knowledge Graph Construction

- **Mapping**
  - # triplesMap
  - # POM
  - # predicates
  - # objects
  - # joins
  - # named graphs
  - join selectivity
  - relation type
  - TermMap type
  - mapping order
- **Data**
  - dataset size
  - frequency dist
  - partitioning
  - data format
- **Platform**
  - cache on/off
  - RAM
  - # processors
- **Source**
  - dist data transfer
  - initial delay
  - access limitation
- **Output**
  - serialisation
  - duplicates
  - generation type

Engines (from RML-Implementation-Report):

- RMLMapper
- SDM-RDFizer

Datasets:

- Type: Naïve, Join-Duplicates, Relation-Type, Join-Selectivity
- Size: 1K, 10K, 50K rows
- Format: CSV

Comparison using Pearson's correlations:

Negative correlation (between 0 and -1)

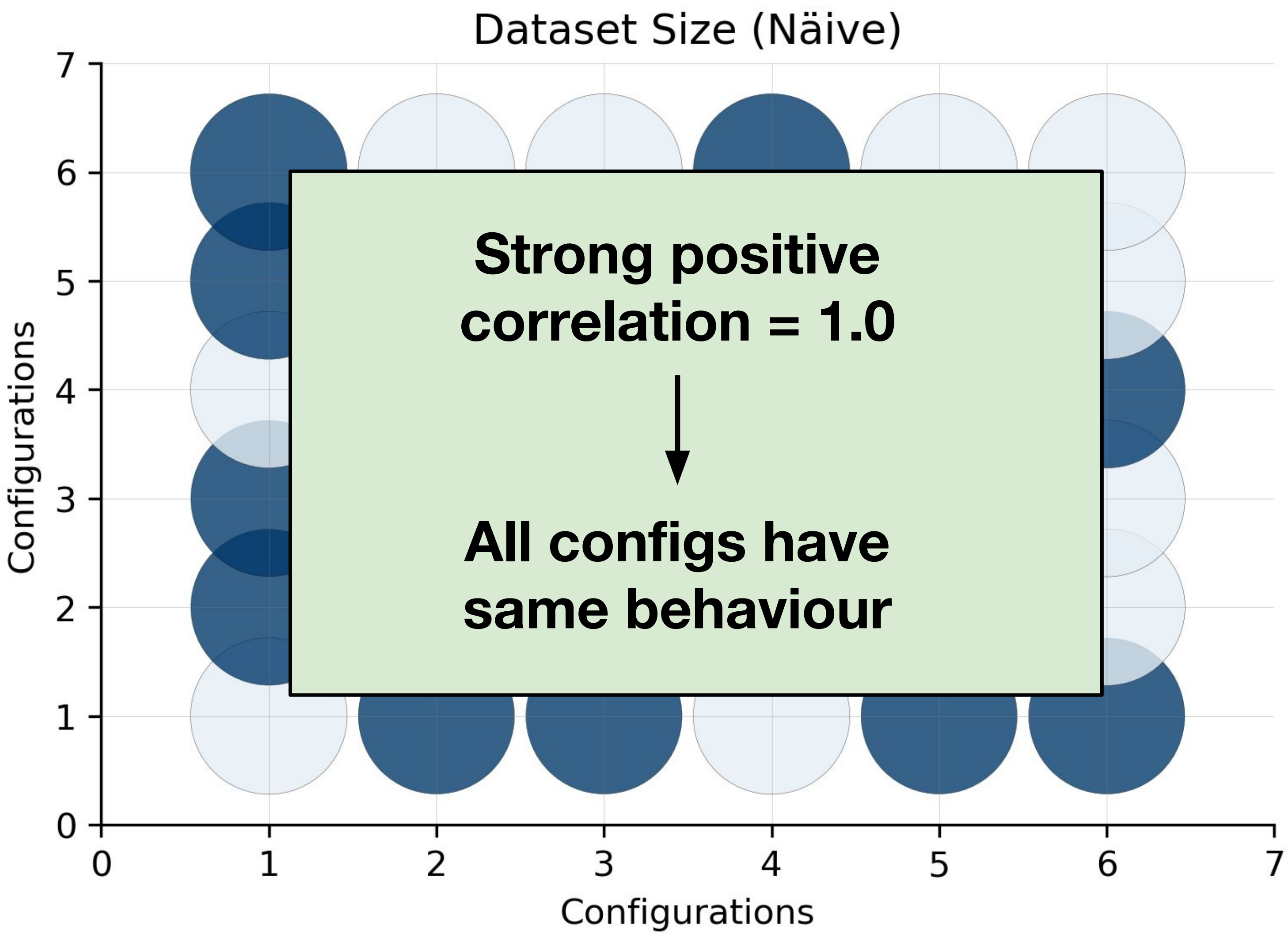
Trends of execution time of the tools are **opposite**

Positive correlation (between 0 and 1)

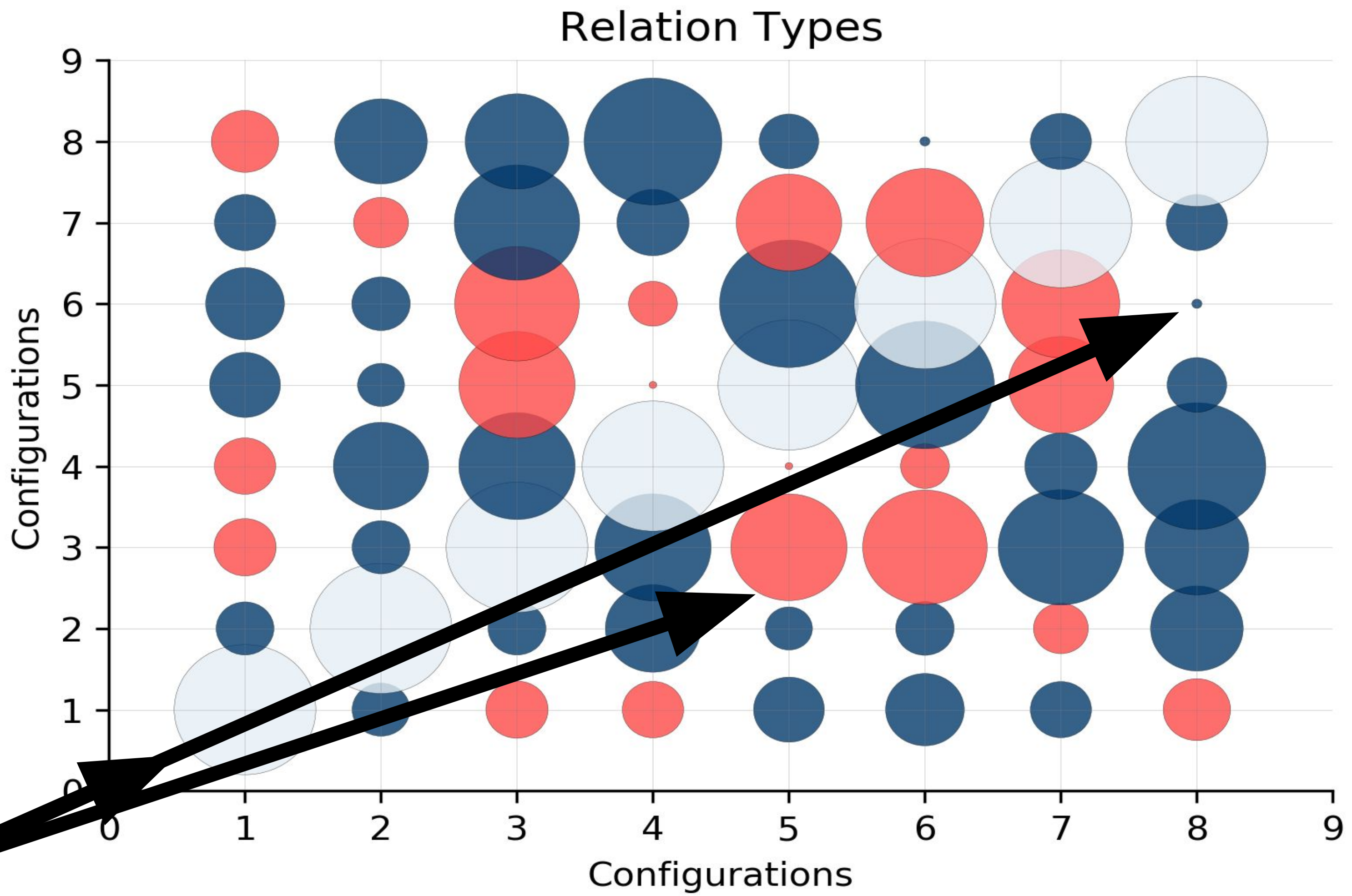
Trends of execution time of the tools are **similar**

Total positive correlation (1)

Comparing **same** configuration



Configurations 1-3: SDM-RDFizer on 1K, 10K and 50K rows
Configurations 4-6: RMLMapper on 1K, 10K and 50K rows



Comparing same configuration. Correlation ≃ 1.0

Weak positive correlation ≃ 0.1

Configurations 1-4: SDM-RDFizer on 1-N, N-1, N-M and combination
Configurations 5-8: RMLMapper on 1-N, N-1, N-M and combination

GTFS-Madrid-Bench: A Benchmark for (Virtual) Knowledge Graph Construction Engines

Chaves-Fraga, D., Priyatna, F., Cimmino, A., Toledo, J., Ruckhaus, E., & Corcho, O. (2020). GTFS-Madrid-Bench: A benchmark for virtual knowledge graph access in the transport domain. *Journal of Web Semantics* (Q2).

A comprehensive benchmark for (virtual) knowledge graph access

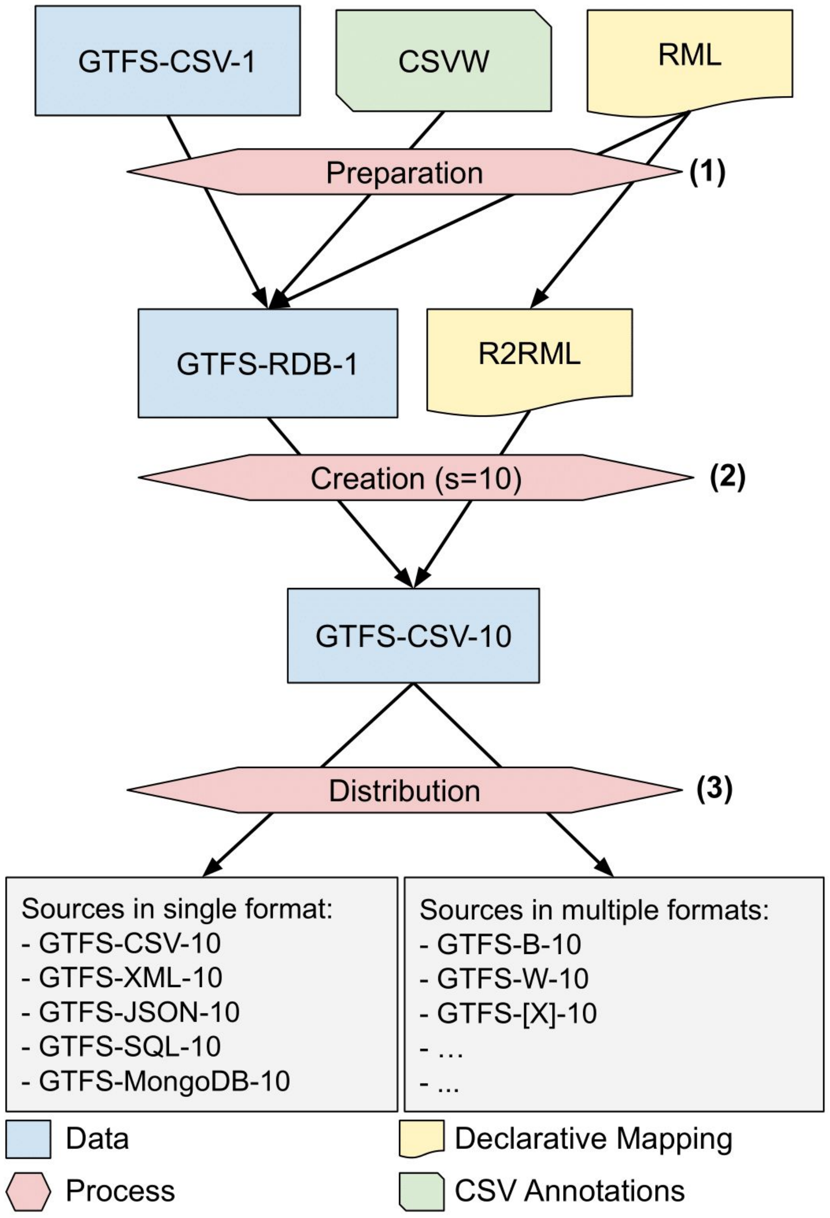
- Query translation over heterogeneous data sources
- Transport Domain (GTFS)
- Unified evaluation framework for heterogeneous OBDA/OBDI engines
- Tested over 5 tools from the state of the art
- Highly influenced by BSBM (queries) and NPD (data generation)

Variable	Requirement
Ontology	The ontology should include classes with data and object properties
Dataset	The virtual instance should maintain the constraints defined in the original dataset
Dataset	The virtual instance should be based on real world data
Dataset	The virtual instance should be distributed in different data formats
Mappings	The mappings should be able to indicate the format of the source
Mappings	The mappings should be expressed using well known mapping languages
Queries	The query set should be based on actual user queries
Queries	The query set should be complex enough with relations among same but also different data sources
Metrics	The metrics should provide relevant general information but also specific measures for each defined phase

(1) Morph-CSV to generate GTFS-RDB-1

(2) VIG to scale-up the RDB

(3) Distribution based on user preferences



TriplesMap	Source	Classes	# POM	# Predicates	# Objects	#ROM
shapes	shapes	gtfs:Shape	4	4	4	0
trips	trips	gtfs:Trip	8	8	5	3
calendar_rules	calendar	gtfs:Calendar	9	9	9	0
calendar_rules_dates	calendar_dates	gtfs:CalendarDateRule	2	2	2	0
stops	stops	gtfs:Stop	12	12	11	1
stoptimes	stop_times	gtfs:StopTime	9	9	7	2
routes	routes	gtfs:Route	8	8	7	1
agency	agency	gtfs:Agency	6	6	6	0
frequencies	frequencies	gtfs:Frequency	5	5	5	1
feed	feed_info	gtfs:Feed	6	6	6	0
service1	calendar	gtfs:Service	1	1	0	1
service2	calendar_dates	gtfs:Service	1	1	0	1

1 R2RML, 5 RML (YARRRML serialization),
1 xR2RML, 1 CSVW annotations + RML-Mapping generator

Query	#Triple Patterns	#Sources	OPTIONAL	Aggregation	Other features	FILTER		#Star-shaped groups	
						equal to	relational	w/o constants	w/constants
q1	4	1						1	0
q2	5	1	yes				yes	0	1
q3	5	1	yes			yes		0	1
q4	9	2	yes					2	0
q5	5	2					yes	1	1
q6	3	2		yes		yes		0	2
q7	15	4	yes		DISTINCT	yes		1	3
q8	14	5	yes					5	0
q9	7	2	yes				yes	1	1
q10	4	2		yes	DISTINCT		yes	1	1
q11	12	3			NOT EXISTS		yes	3	2
q12	10	4		yes	GROUP BY			3	1
q13	6	1	yes					0	1
q14	8	3	yes		ORDER BY			3	0
q15	3	1				yes		0	1
q16	8	3					yes	2	1
q17	9	3						3	0
q18	8	5			UNION			4	1

Dataset:

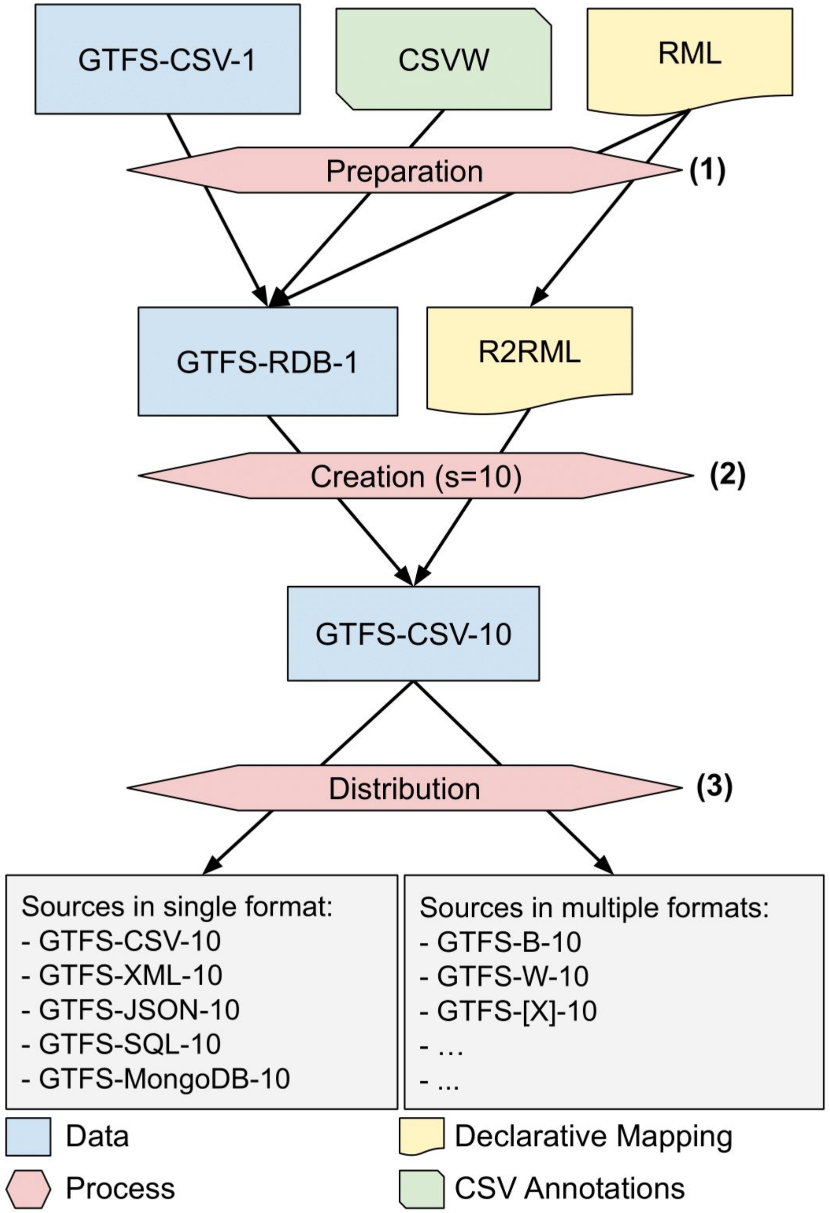
- Morph-CSV to generate GTFS-RDB
- VIG to scale-up
- Distribution based on user preferences

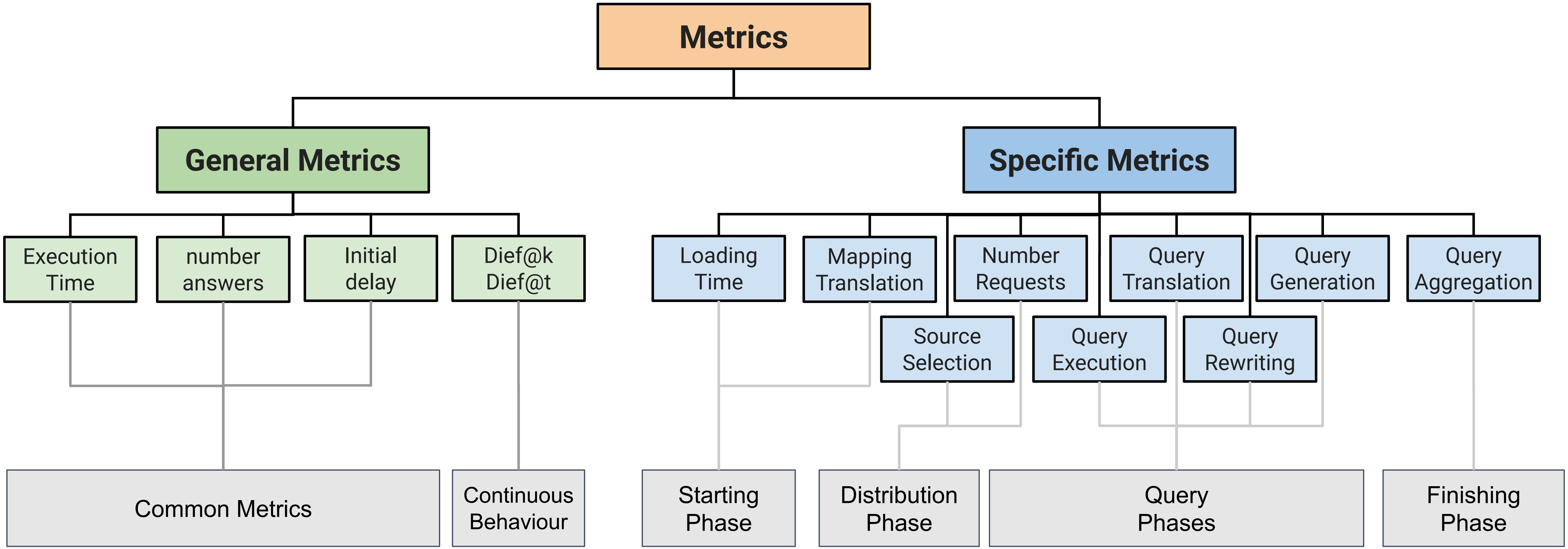
Queries:

- 18 queries covering different configurations and SPARQL operators
- Aligned with user stories in Madrid's transport domain
- Triple patterns: from 3 to 15; Sources: 1 to 5
- Single and chain star-shaped groups

Mappings:

- 10 Sources, 12 TriplesMap (12 Classes), 71 POM (70 P), 60 SOM, 11 ROM
- 1 R2RML, 5 RML (YARRRML serialization), 1 xR2RML, 1 CSVW annotations + RML-Mapping generator




Metrics
General Metrics
Specific Metrics
Execution Time
number answers
Initial delay
Dief@k Dief@t
Loading Time
Mapping Translation
Number Requests
Query Translation
Query Generation
Query Aggregation
Source Selection
Query Execution
Query Rewriting
Common Metrics
Continuous Behaviour
Starting Phase
Distribution Phase
Query Phases
Finishing Phase

Evaluation

TO (TimeOut), W (wrong nº results), E (error executing the query)

Dataset	Processor		Query																	
	Cache	**Name**	**q1**	**q2**	**q3**	**q4**	**q5**	**q6**	**q7**	**q8**	**q9**	**q10**	**q11**	**q12**	**q13**	**q14**	**q15**	**q16**	**q17**	**q18**
GTFS-SQL-1	Warm	Morph-RDB	5.85	02.07	E	1.82	W	1.86	1.97	E	26.02	1.80	E	1.81	2.06	W	1.89	E	2.11	E
	Cold	Ontario	18.02	E	TO	E	E	E	E	W	E	E	E	E	E	W	E	E	E	E
		Morph-RDB	7.14	2.65	E	2.42	W	2.36	2.43	E	28.65	2.38	E	2.41	2.69	W	2.58	E	2.68	E
		Ontop	8.37	05.04	5.18	E	W	E	W	E	16.56	E	E	E	05.06	W	5.10	W	5.00	W
GTFS-MongoDB-1	Warm	Morph-xR2RML	W	W	W	W	W	W	W	W	W	W	W	W	W	28.67	W	W	6.52	W
	Cold	Morph-xR2RML	W	W	W	W	W	W	W	W	W	W	W	W	W	28.17	W	W	6.96	W
GTFS-CSV-1	Cold	Morph-RDB	6.94	03.04	E	2.78	E	2.78	TO	E	TO	2.97	E	6.23	3.97	E	E	E	3.14	W
		Morph-CSV	15.11	10.88	E	10.72	E	9.95	10.84	E	40.90	10.70	E	11.60	11.82	E	E	E	11.48	W
		Ontario	W	E	17.34	E	E	E	E	W	E	E	E	E	E	W	E	E	E	E
GTFS-XML-1	Cold	Ontario	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
GTFS-JSON-1	Cold	Ontario	18.04	E	17.14	E	E	E	E	W	E	E	E	E	E	W	E	E	E	E
GTFS-MINEXT-1	Cold	Ontario	W	E	E	E	E	E	E	W	E	E	E	E	E	W	E	E	E	E
GTFS-MAXEXT-1	Cold	Ontario	W	E	17.14	E	E	E	E	W	E	E	E	E	E	W	E	E	E	E

- Only the SPARQL-to-SQL engines provide an acceptable support for SPARQL operators
- Virtual KGC proposals beyond relational databases are not mature enough and more research is needed
- The problem of translating SPARQL queries for querying raw data (CSV, JSON, XML) should not be understood as a technical case

GTFS-Madrid-Bench:

- First proposal able to provide a unique point for evaluating heterogeneous KGC engines*
- Open context in comparison with previous proposals (BSBM, NPD)
- Used in H2020-SPRINT to test KGC scalability and performance for NAP
- High support through open science principles

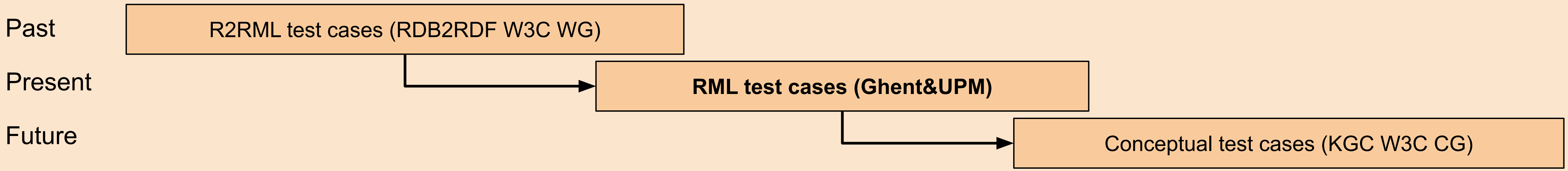
Necessary Improvements:

- Automatize generation of mapping rules beyond RML through mapping translation
- Incorporation of semantics in data generation process (e.g., start date before end date)
- Evaluate the impact of the benchmark parameters in KGC engines behavior
- Testing new capabilities of KGC engines (e.g., RDF-star gen, transformation functions, lists, etc.)

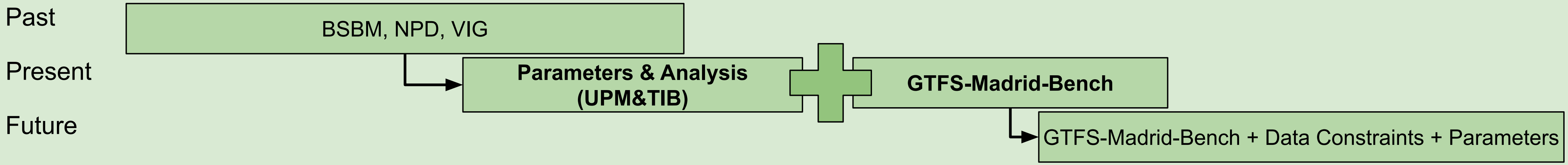
*Arenas-Guerrero, J., Scrocca, M., Iglesias-Molina, A., Toledo, J., Pozo-Gilo, L., Dona, D., Corcho, O., & **Chaves-Fraga, D.** (2021). Knowledge Graph Construction with R2RML and RML: An ETL System-based Overview. *In Proceedings of the 2nd International Workshop on Knowledge Graph Construction.*

Evaluation framework for Knowledge Graph Construction Engines (Past&Present&Future)

Conformance with the mapping language



Performance & Scalability



Contents

Mapping Translation: Concept and desirable properties

Corcho, O., Priyatna, F., & **Chaves-Fraga, D.** (2020). Towards a new generation of ontology based data access. *Semantic Web* (Q1)

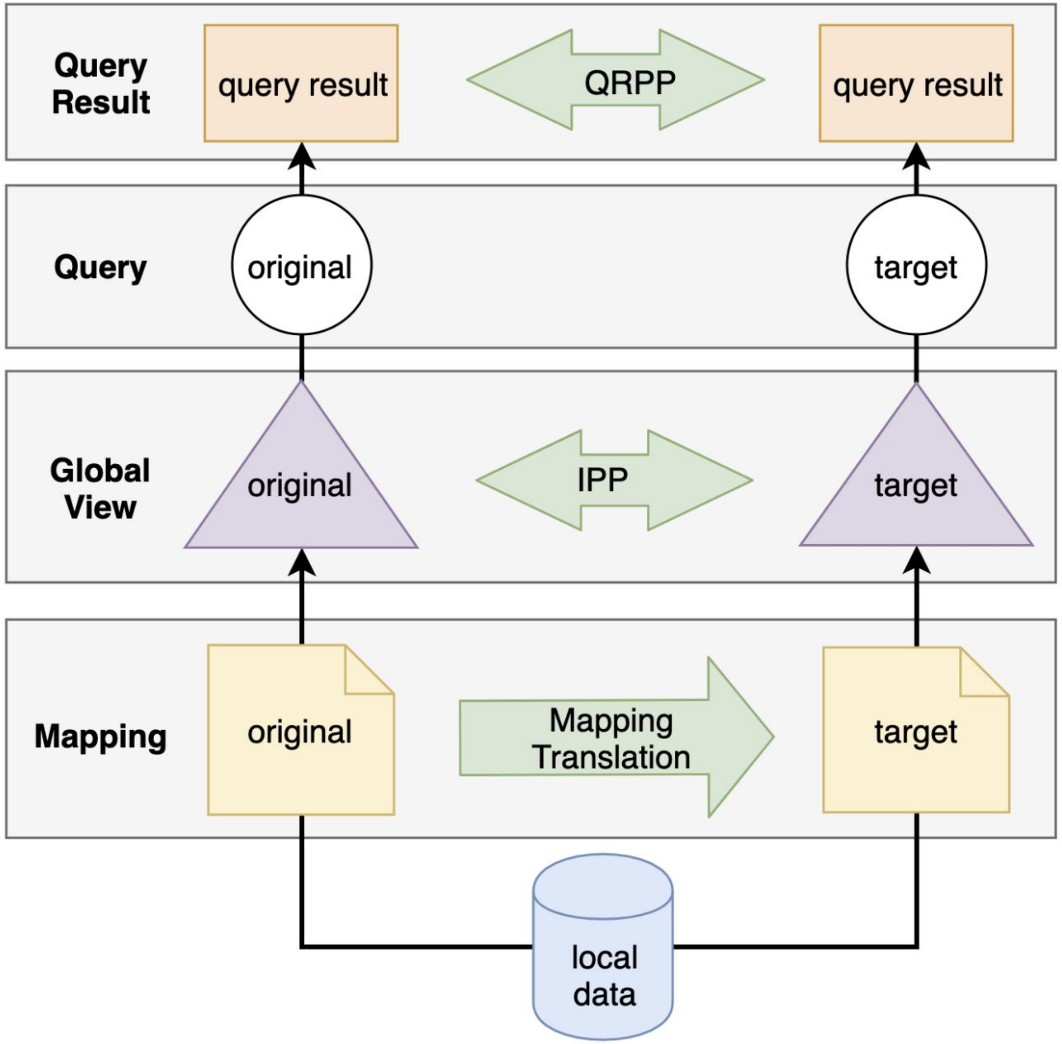
New generation of Knowledge Graph Construction Engines

Main properties:

- IPP: Information Preservation Property
- QRPP: Query Result Preservation Property

Main use-cases

Aim	Engine	Translation
Access / Optimizations	**Morph-CSV**	**RML+FnO-to-R2RML**
Optimizations	**FunMap**	**RML+FnO-to-RML**
Maintenance / Schema	**Morph-GraphQL**	**R2RML-to-GraphQL**
Maintenance	**Morph-CSV**	**R2RMLIter-to-R2RML**
Maintenance	ShExML	ShExML-to-RML
Maintenance	yarrrml-parser	YARRRML-to-RML
Maintenance	Mapeathor	Excel-to-[R2]RML
Optimizations / Semantics	ontop	R2RML-to-OBDA



Exploiting Declarative Annotations for Virtual Knowledge Graph Construction

morph

Virtual Knowledge Graph Construction over Tabular Data

Chaves-Fraga, D., Ruckhaus, E., Priyatna, F., Vidal, M. E., & Corcho, O. (2021). Enhancing Virtual Ontology Based Access over Tabular Data with Morph-CSV. *Semantic Web* (Q1).

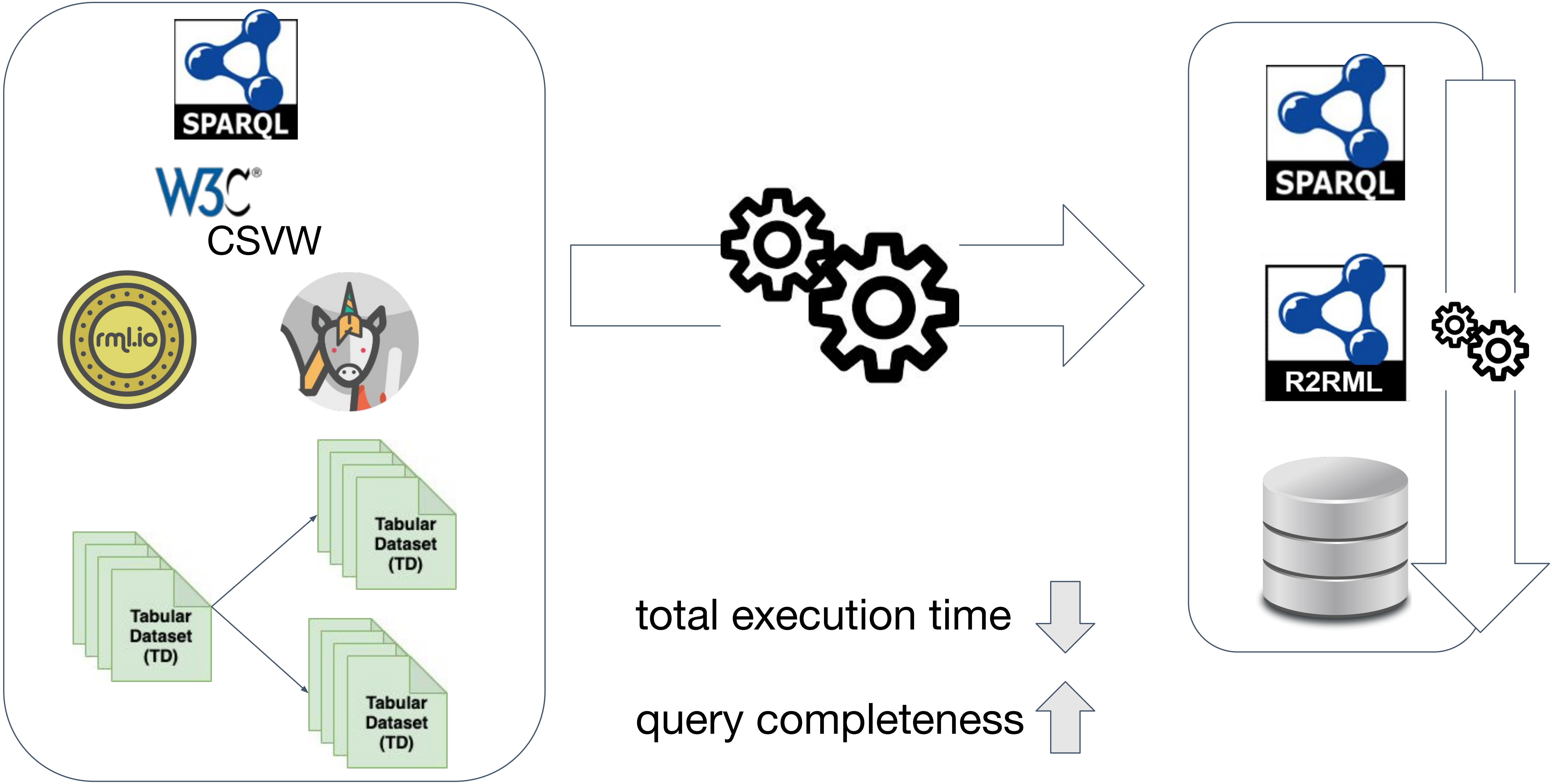
Chaves-Fraga, D., Pozo-Gilo, L., Toledo, J., Ruckhaus, E., & Corcho, O. (2020). Morph-CSV: Virtual Knowledge Graph Access for Tabular Data. In *International Semantic Web Conference (P&D)* (Core A).

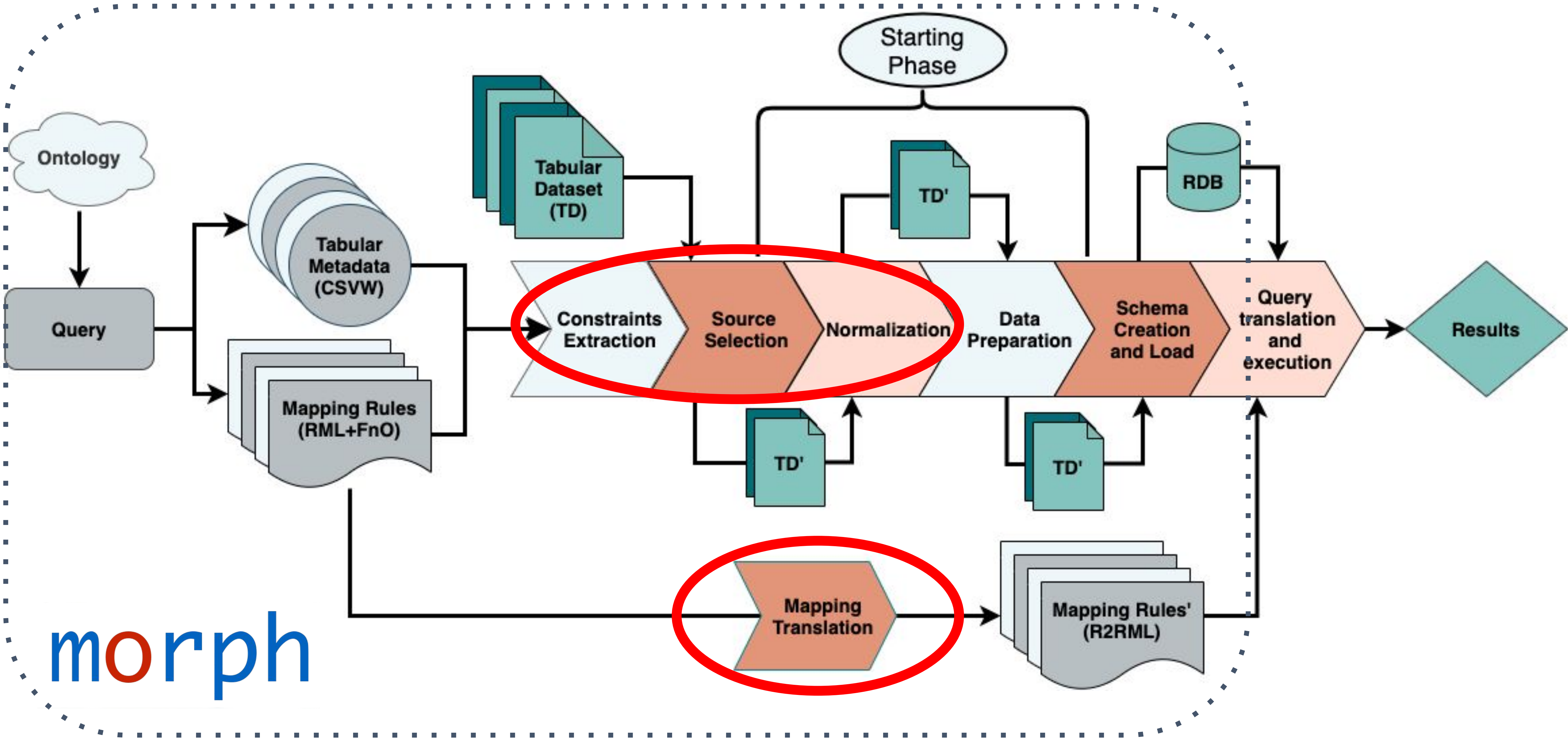
Can we reuse existing optimizations proposed
in SPARQL-to-SQL VKG construction?

SPARQL-to-SQL optimizations assumptions:

- There is a **native query language** for the input data source.
- There is an schema and typically includes **integrity and domain constraints** (cleaned and normalized).
- The data source is an **RDB instance or is a NoSQL database** instance with an RDB wrapper.

Challenges in SPARQL-to-CSV: Updated results, lightweight schema (no native query language), heterogeneity (messy data), not normalized

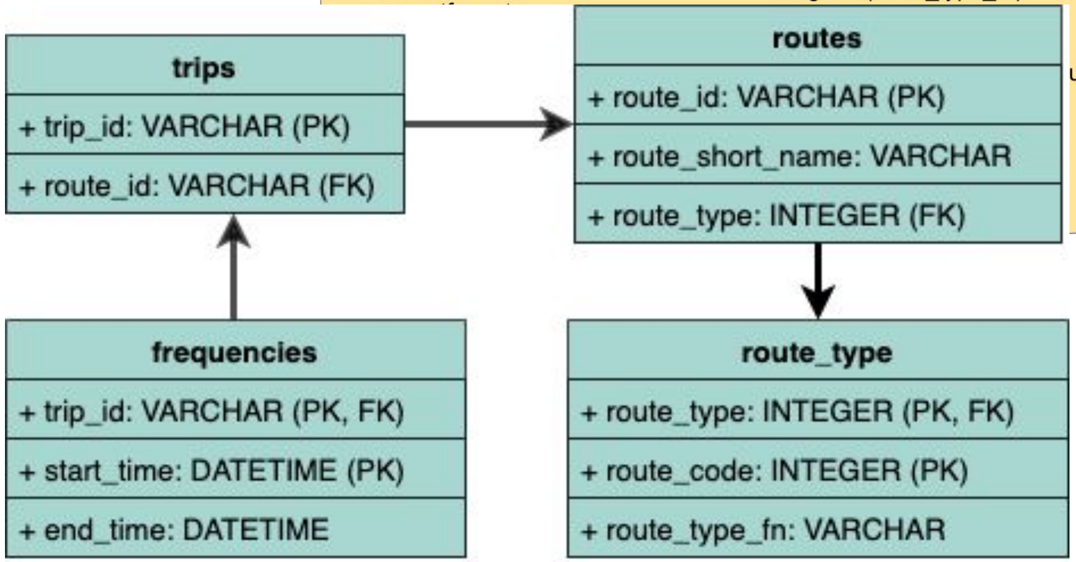

SPARQL
W3C
CSVW
rml.io
Tabular Dataset (TD)
Tabular Dataset (TD)
Tabular Dataset (TD)
total execution time
query completeness
SPARQL
R2RML


Starting Phase
Ontology
Query
Tabular Metadata (CSVW)
Mapping Rules (RML+FnO)
Tabular Dataset (TD)
Constraints Extraction
Source Selection
Normalization
Data Preparation
Schema Creation and Load
Query translation and execution
Results
TD'
TD'
TD'
RDB
Mapping Translation
Mapping Rules' (R2RML)
morph

```
frequencies:
  sources:
    - [frequencies.csv~csv]
  s: mbench:freq/$(trip_id)-$(start_time)
  po:
    - [a, gtfs:Frequency]
    - [gtfs:startTime,$(start_time)]
    - [gtfs:endTime,$(end_time)]
    - [gtfs:headSecs,$(headway_secs)]
    - [gtfs:exactTimes,$(exact_times)]
    - p: gtfs:trip
      o:
        - mapping: trips
          condition:
            function: equal
            parameters:
              - [str1, $(trip_id)]
              - [str2, $(trip_id)]
trips:
  sources:
    - [trips.csv~csv]
  s: mbench:trips/$(trip_id)
  po:
    - [a, gtfs:Trip]
    - [gtfs:headsign, $(trip_headsign)]
    - [gtfs:shortName, $(trip_short_name)]
    - [gtfs:direction, $(direction_id)]
    - [gtfs:block, $(block_id)]
    - p: gtfs:route
      o:
        - mapping: routes
          condition:
            function: equal
            parameters:
              - [str1, $(route_id)]
              - [str2, $(route_id)]
routes:
  sources:
    - [routes.csv~csv]
  s: mbench:routes/$(route_id)
  po:
    - [a, gtfs:Route]
    - [gtfs:shortName, $(route_short_name)]
    - [gtfs:longName, $(route_long_name)]
    - [dct:description, $(route_desc)]
    - [gtfs:routeUrl, $(route_url)~iri]
    - [gtfs:color, $(route_color)]
    - [gtfs:textColor, $(route_text_color)]
    - p: gtfs:agency
      o:
        - mapping: agency
          condition:
            function: equal
            parameters:
              - [str1, $(agency_id)]
              - [str2, $(agency_id)]
    - p: gtfs:RouteType
      o:
        - mapping: route-type
          condition:
            function: equal
            parameters:
              - [str1, $(route_type)]
              - [str2, $(route_type)]
route-type:
  sources:
    - [routes.csv~csv]
  s: CONCAT(gtfs:,TRANS($(route_type)))
  po:
    - [a, gtfs:RouteType]
    - [gtfs:routeTypeCode,$(route_code)]
```

R2RML

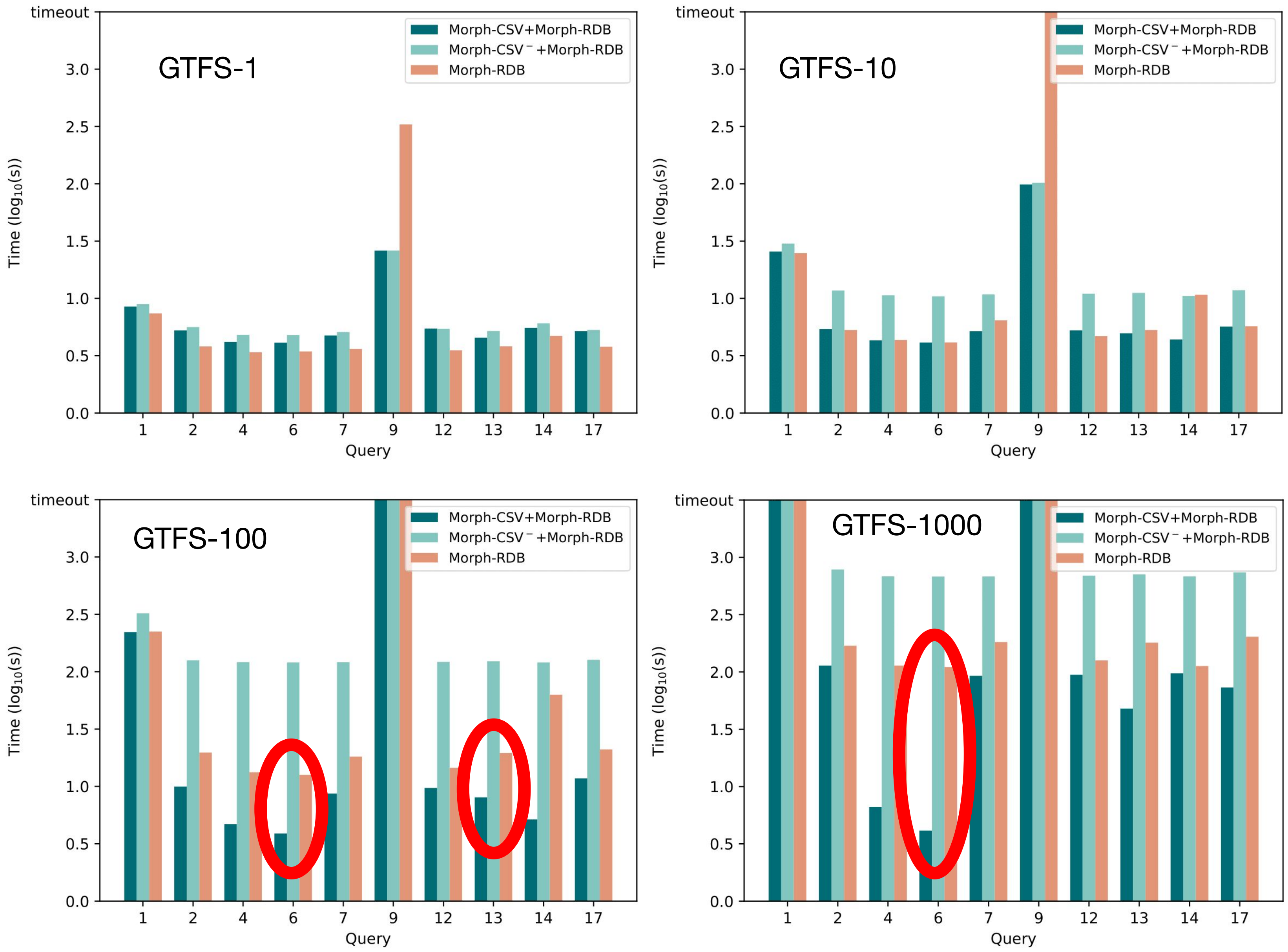
```
frequencies:
  tables:
    - [frequencies]
  s: mbench:freq/$(trip_id)-$(start_time)
  po:
    - [a, gtfs:Frequency]
    - [gtfs:startTime,$(start_time)]
    - [gtfs:endTime,$(end_time)]
    - p: gtfs:trip
      o:
        - mapping: trips
          condition:
            function: equal
            parameters:
              - [str1, $(trip_id)]
              - [str2, $(trip_id)]
trips:
  tables:
    - [trips]
  s: mbench:trips/$(trip_id)
  po:
    - [a, gtfs:Trip]
routes:
  tables:
    - [route]
  s: mbench:routes/$(route_id)
  po:
    - [a, gtfs:Route]
    - [gtfs:longName, $(route_long_name)]
    - p: gtfs:RouteType
      o:
        - mapping: route-type
          condition:
            function: equal
            parameters:
              - [str1, $(route_type)]
              - [str2, $(route_type)]
route-type:
  tables:
    - [route_type]
  s: gtfs:$(route_type_fn)
ute_code)]
```



Baseline:
Morph-RDB

Approach 1 (complete RDB):
Morph-CSV⁻ + Morph-RDB

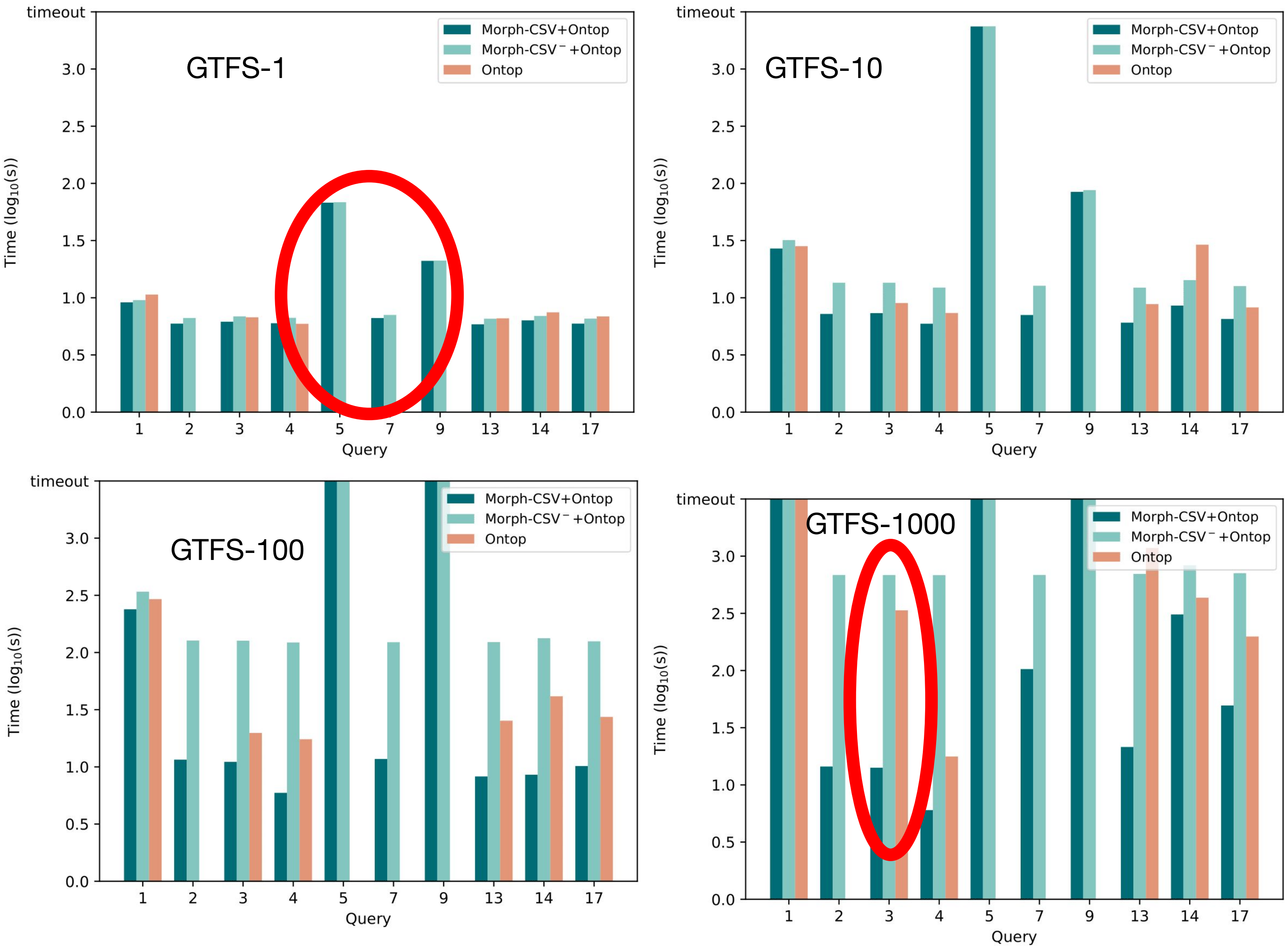
Approach 2 (source selection):
Morph-CSV + Morph-RDB



Baseline:
Ontop

Approach 1 (complete RDB):
Morph-CSV⁻ + Ontop

Approach 2 (source selection):
Morph-CSV + Ontop



morph

GraphQL Server Generation from Declarative Mappings

Chaves-Fraga, D., Priyatna, F., Alobaid, A., & Corcho, O. (2020). Exploiting Declarative Mapping Rules for Generating GraphQL Servers with Morph-GraphQL. *International Journal of Software Engineering and Knowledge Engineering* (Q4)

Priyatna, F., **Chaves-Fraga, D.**, Alobaid, A., & Corcho, Ó. (2019). morph-GraphQL: GraphQL Servers Generation from R2RML Mappings. In *International Conference on Software Engineering and Knowledge Engineering* (Core B).

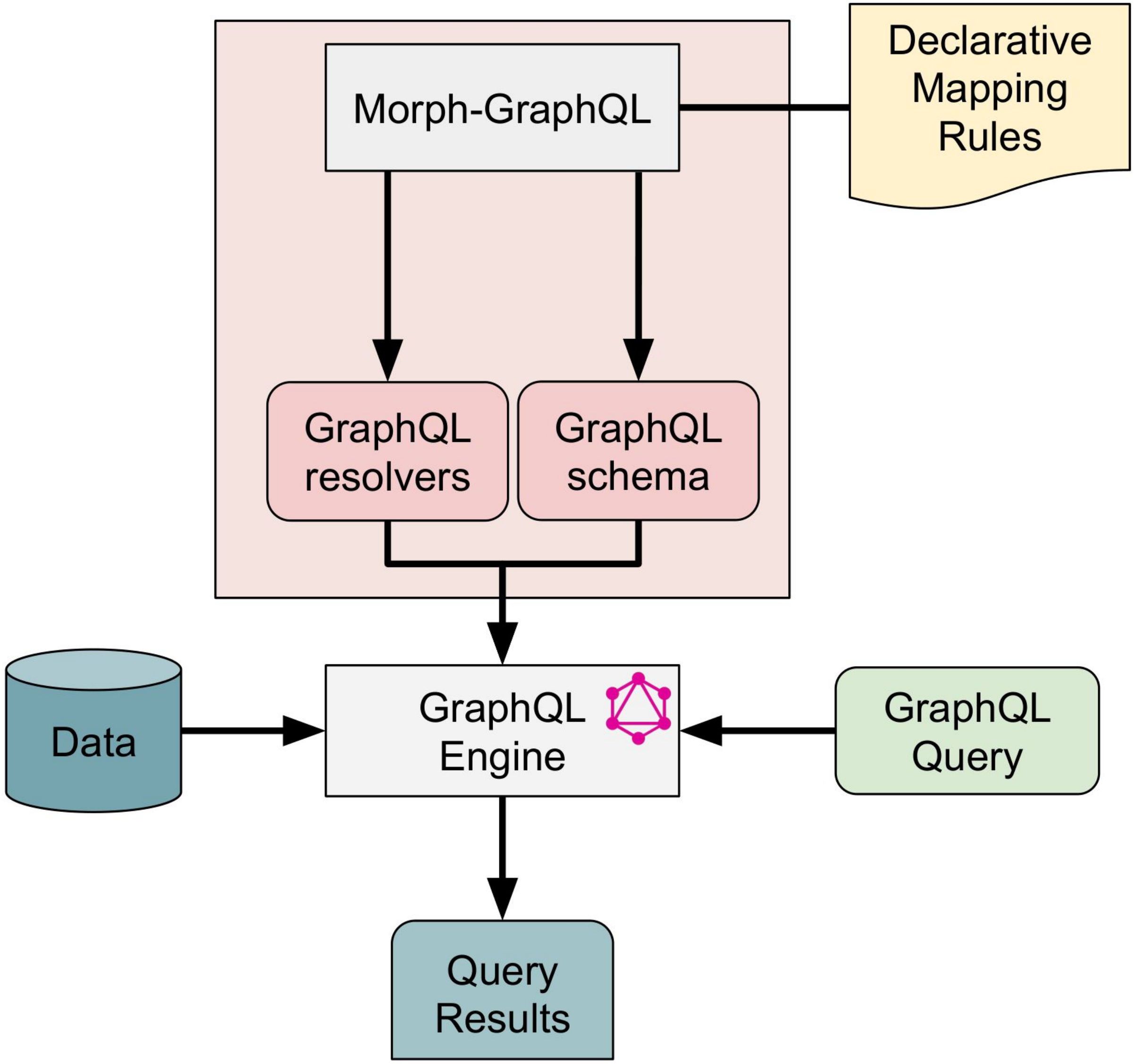
Alobaid, A., **Chaves-Fraga, D.**, Priyatna, F., & Corcho, Ó. (2019). GraphQL Servers generation from R2RML with morph-GraphQL (Demo). In I*nternational Conference on Software Engineering and Knowledge Engineering* (Core B). Best Demo Award

Main motivations:

- Common representation of mappings between source and GraphQL schema
- Shared vocabularies to avoid data silos
- Bootstrapping GraphQL development

Main features:

- Translation based on Chebotko's SPARQL-to-SQL algorithm
- Experimentation with Linköping GraphQL Benchmark v0.1
- Similar results as Virtuoso or VKGC (Morph-RDB)



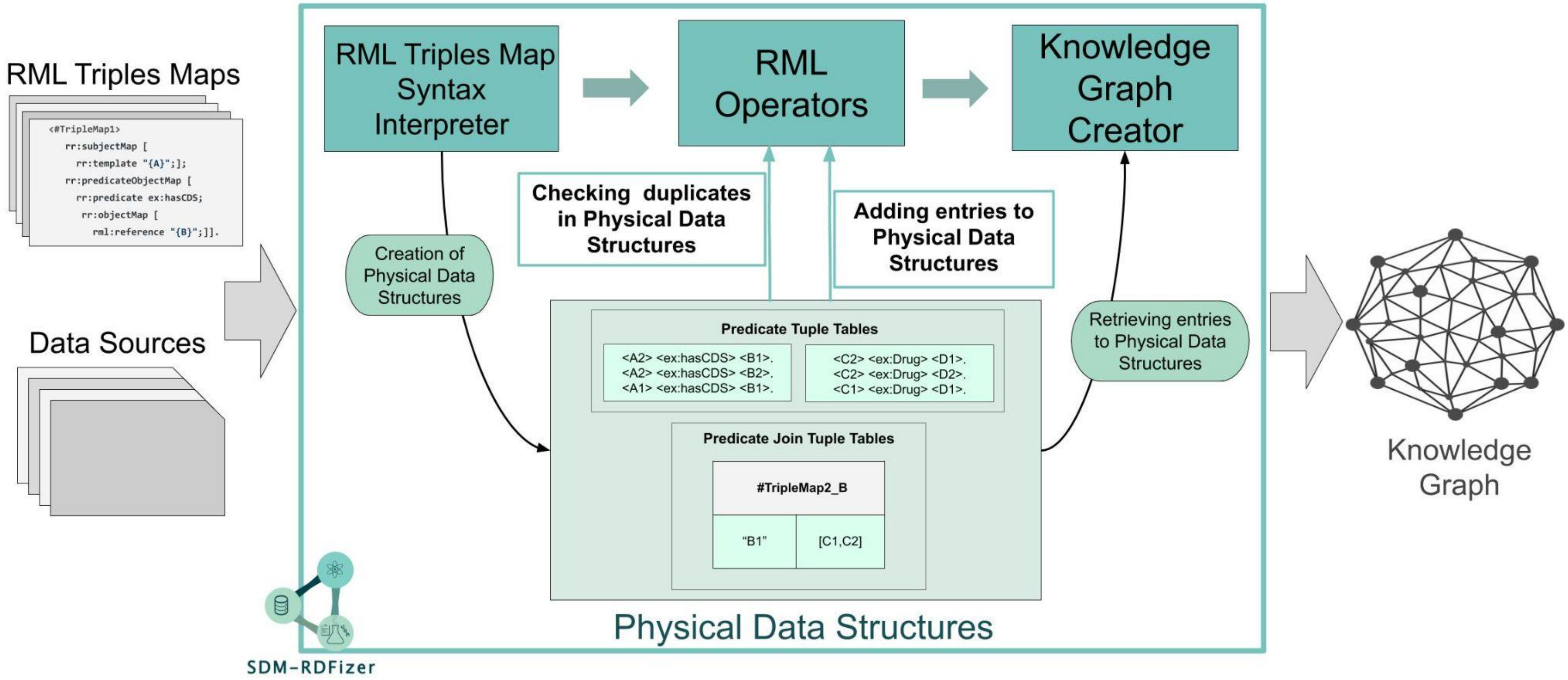
Optimizations for Scaling-up Materialized Knowledge Graph Construction Techniques*

* These contributions are the result of joint collaboration with the Scientific Data Management Group from German National Library of Science and Technology (TIB), as a result of a research stay in the institution.

SDM-RDFizer

SDM-RDFizer: An RML Interpreter for the Efficient Creation of RDF Knowledge Graphs.

Iglesias, E., Jozashoori, S., **Chaves-Fraga, D**., Collarana, D., & Vidal, M. E. (2020). SDM-RDFizer: An RML interpreter for the efficient creation of rdf knowledge graphs. In *Proceedings of the 29th ACM CIKM* (Core A). First 3 authors contributed equally to the research


RML Triples Maps
<#TripleMap1>
rr:subjectMap [
rr:template "{A}";];
rr:predicateObjectMap [
rr:predicate ex:hasCDS;
rr:objectMap [
rml:reference "{B}";]].
Data Sources
RML Triples Map Syntax Interpreter
RML Operators
Knowledge Graph Creator
Creation of Physical Data Structures
Checking duplicates in Physical Data Structures
Adding entries to Physical Data Structures
Retrieving entries to Physical Data Structures
Predicate Tuple Tables
<A2> <ex:hasCDS> <B1>.
<A2> <ex:hasCDS> <B2>.
<A1> <ex:hasCDS> <B1>.
<C2> <ex:Drug> <D1>.
<C2> <ex:Drug> <D2>.
<C1> <ex:Drug> <D1>.
Predicate Join Tuple Tables
#TripleMap2_B
"B1"
[C1,C2]
Physical Data Structures
SDM-RDFizer
Knowledge Graph

Predicate Tuple Table (PTT)

- stores RDF triples for a predicate generated so far

Hash table:

- **Key** encoding **subject** and **object**
- **Value** the RDF triple.

Predicate Join Tuple Table (PJTT)

- stores values generated during execution of a join condition.

Hash table:

- **Key** encoding the **value** of the **attributes** in the **join**
- **Value** set with the subject values associated with the values of the attributes in the hash key

<http://example.org/trips/1> <schema:name> “Sol-Aluche”.
<http://example.org/trips/2> <schema:name> “Sol-Retiro”.
<http://example.org/trips/1> <schema:name> “Sol-Aluche”.

Key	Value
http://example.org/trips/1_Sol-Aluche	<http://example.org/trips/1> <schema:name> “Sol-Aluche”
http://example.org/trips/2_Sol-Retiro	<http://example.org/trips/2> <schema:name> “Sol-Retiro”

PTT schema:name

trips.csv

trip_name	trip_id
“Sol-Aluche”	2193351
“Sol-Retiro”	2193351

routes.csv

trip_id	route_id
2193351	1455465
2193351	2064548
2196270	2061629

TripleMap_Trips_Routes	
“2193351”	[1455465,2064548]
“2196270”	[2061629]

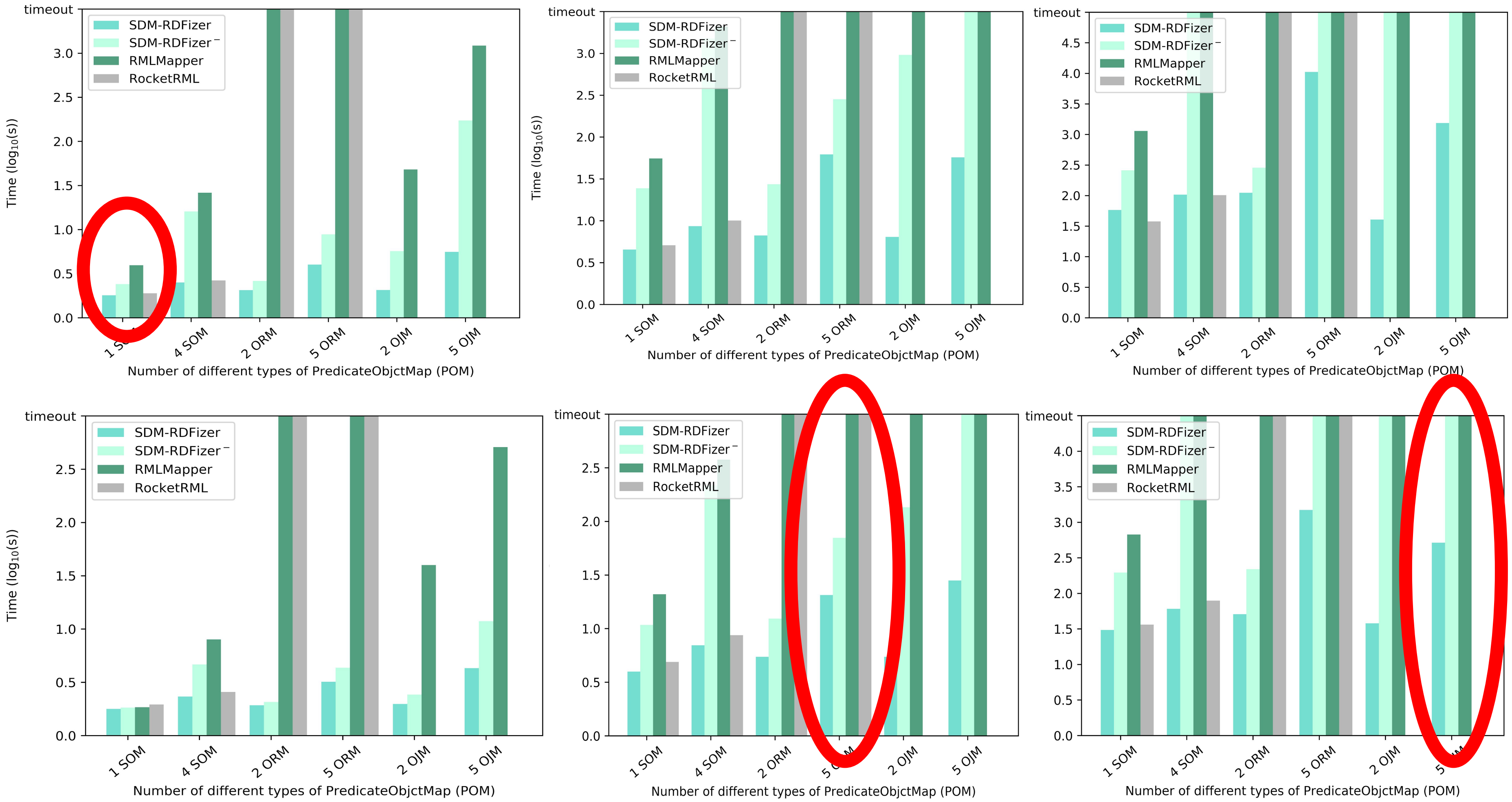
JPTT TripleMap2_trips_routes

What is the impact of data duplication rate, data size and triples map types in the execution time of a knowledge graph creation approach?

10K rows | 100K rows | 1M rows

25% Dup

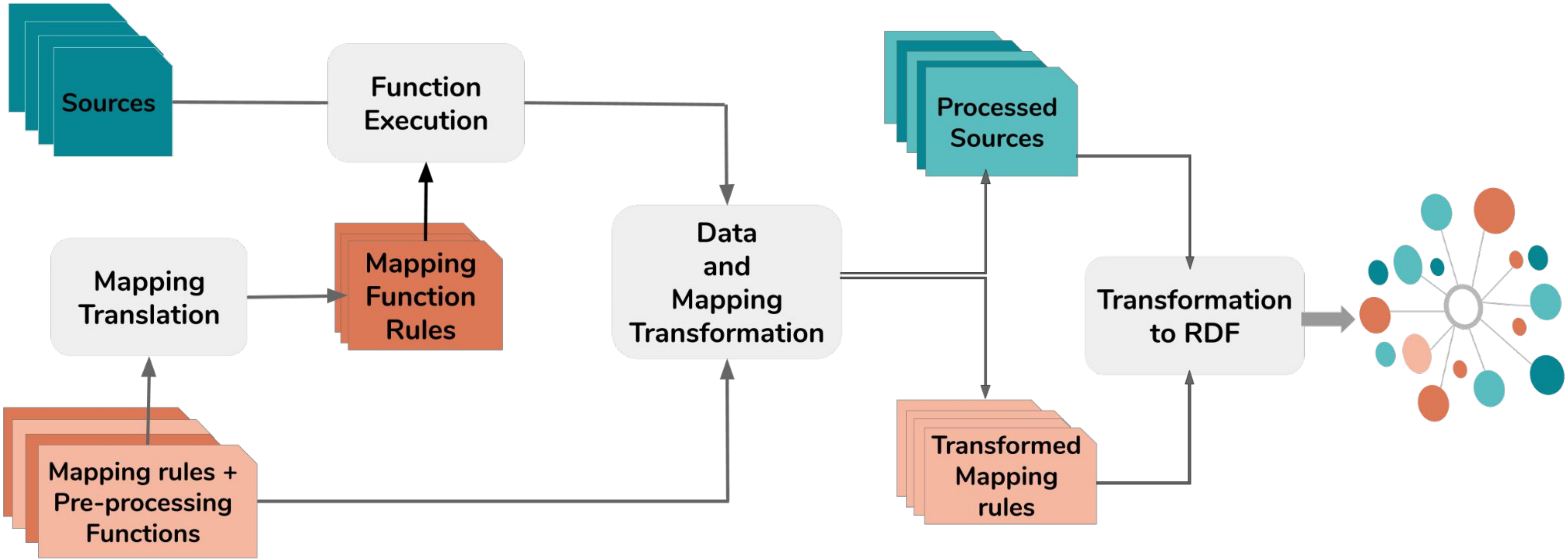
75% Dup



FunMap

FunMap: Efficient Execution of Functional Mappings for Scaled-Up Knowledge Graph Creation

Jozashoori, S., **Chaves-Fraga, D**., Iglesias, E., Vidal, M. E., & Corcho, O. (2020, November). FunMap: Efficient Execution of Functional Mappings for Knowledge Graph Creation. In *International Semantic Web Conference* (Core A). First two authors contributed equally to the research. Fully reproduced paper


Sources
Function Execution
Mapping Translation
Mapping Function Rules
Mapping rules + Pre-processing Functions
Data and Mapping Transformation
Processed Sources
Transformed Mapping rules
Transformation to RDF

Function in an ObjectMap:

- Output file with function applied
- References used in function for join conditions
- Projecting input sources

```
<#TriplesMap1>
   rml:logicalSource [ rml:source "source1.csv";
                       rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rr:template "ias:/Mutation/{GENOMIC_MUTATION_ID}";
       rr:class ias:Mutation;];
   rr:predicateObjectMap [
       rr:predicate iasis:isLocatedIn;
       rr:objectMap <#FunctionMap1> ];
   rr:predicateObjectMap [
       rr:predicate iasis:tissue;
       rr:objectMap [
           rml:reference "Primary site" ]].
<#TriplesMap2>
   rml:logicalSource [ rml:source "source1.csv";
                       rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rr:template "ias:/Gene/{Gene name}";
       rr:class iasis:Gene;];
   rr:predicateObjectMap [
       rr:predicate iasis:isRelatedTo;
       rr:objectMap <#FunctionMap1>].
```

```
<#FunctionMap1>
   a fnml:FunctionTermMap;
   fnml:functionValue [
       rml:logicalSource [ rml:source "source1.csv";
       rml:referenceFormulation ql:CSV ];
       rr:predicateObjectMap [
           rr:predicate fno:executes ;
           rr:objectMap [
               rr:constant ex:replaceValue ]];
       rr:predicateObjectMap [
           rr:predicate ex:value;
           rr:objectMap [
               rml:reference "Mutation genome position"]];
       rr:predicateObjectMap [
           rr:predicate ex:value2;
           rr:objectMap [
               rr:constant "-"; ]];
       rr:predicateObjectMap [
           rr:predicate ex:value3;
           rr:objectMap [
               rr:constant ":"; ]];].
```

Transforms to

```
<#TriplesMap1>
   a rr:TriplesMap;
   rml:logicalSource [ rml:source "projected1.csv";
   rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rr:template "ias:/Mutation/{GENOMIC_MUTATION_ID}";
       rr:class ias:Mutation;];
   rr:predicateObjectMap [
       rr:predicate iasis:isLocatedIn;
       rr:objectMap [
       rr:parentTriplesMap <#TriplesMap3>;
       rr:joinCondition [
           rr:child "Mutation genome position";
           rr:parent "Mutation genome position"
       ;];];].

<#TriplesMap2>
   rml:logicalSource [ rml:source "projected2.csv";
                       rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rr:template "ias:/Gene/{Gene name}";
       rr:class iasis:Gene;];
   rr:predicateObjectMap [
       rr:predicate iasis:isRelatedTo;
       rr:objectMap [
       rr:parentTriplesMap <#TriplesMap3>;
       rr:joinCondition [
           rr:child "Mutation genome position";
           rr:parent "Mutation genome position"
       ;];];].

<#TriplesMap3>
   a rr:TriplesMap;
   rml:logicalSource [ rml:source "output1.csv";
             rml:referenceFormulation ql:CSV
          ];
   rr:subjectMap [
       rml:reference "functionOutput"
   ].
```

Function in an SubjectMap:

- **Main** output file with function applied
- References used in function for join conditions
- Projecting input sources

```
<#TriplesMap1>
   rml:logicalSource [ rml:source "source1.csv";
                       rml:referenceFormulation ql:CSV ];
   rr:subjectMap <#FunctionMap1> ;
   rr:predicateObjectMap [
       rr:predicate iasis:represents;
       rr:objectMap [
          rml:reference "Mutation" ]];
   rr:predicateObjectMap [
       rr:predicate iasis:tissue;
       rr:objectMap [
          rml:reference "Primary site" ]].
```

```
<#FunctionMap1>
   a fnml:FunctionTermMap;
   fnml:functionValue [
       rml:logicalSource [ rml:source "source1.csv";
       rml:referenceFormulation ql:CSV ];
       rr:predicateObjectMap [
           rr:predicate fno:executes ;
           rr:objectMap [
               rr:constant ex:replaceValue ]];
       rr:predicateObjectMap [
           rr:predicate ex:value;
           rr:objectMap [
               rml:reference "Mutation genome position"]];
       rr:predicateObjectMap [
           rr:predicate ex:value2;
           rr:objectMap [
               rr:constant "-"; ]];
       rr:predicateObjectMap [
           rr:predicate ex:value3;
           rr:objectMap [
               rr:constant ":"; ]];].
```

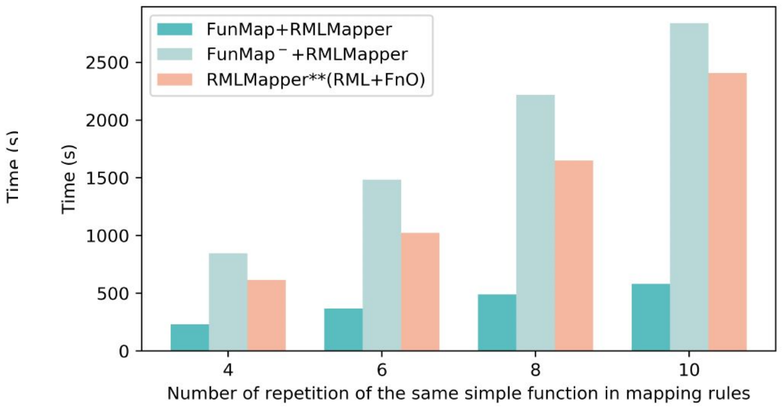
Transforms to

```
<#TriplesMap1>
   rml:logicalSource [ rml:source "output1.csv";
                       rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rml:reference "functionOutput" ];
   rr:predicateObjectMap [
       rr:predicate iasis:represents;
       rr:objectMap [
       rr:parentTriplesMap <#TriplesMap2> ;
       rr:joinCondition [
           rr:child "Mutation genome position";
           rr:parent "Mutation genome position";];];].
   rr:predicateObjectMap [
       rr:predicate
       rr:objectMap [
       rr:parentTriplesMap <#TriplesMap3> ;
       rr:joinCondition [
           rr:child "Mutation genome position";
           rr:parent "Mutation genome position";];];].
<#TriplesMap2>
   a rr:TriplesMap;
   rml:logicalSource [ rml:source "projected1.csv";
          rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rml:reference "Mutation" ].
<#TriplesMap3>
   a rr:TriplesMap;
   rml:logicalSource [ rml:source "projected1.csv";
          rml:referenceFormulation ql:CSV ];
   rr:subjectMap [
       rml:reference "Primary site" ].
```

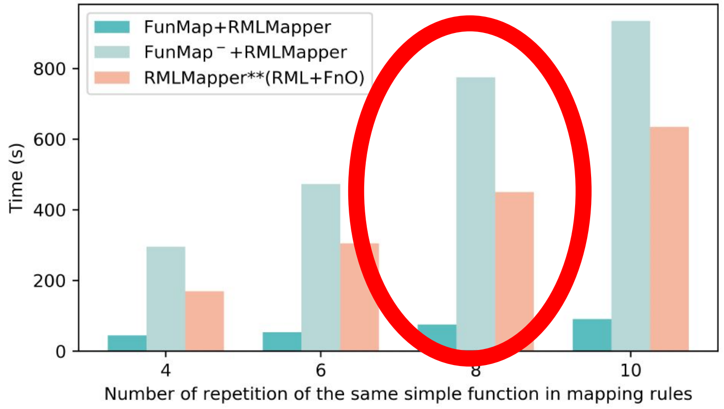
FunMap - Experimental results (COSMIC dataset)

What is the impact of data duplication rate and different types of complexity over transformation functions in the execution time of a knowledge graph construction approach?

Simple functions (lower, upper)

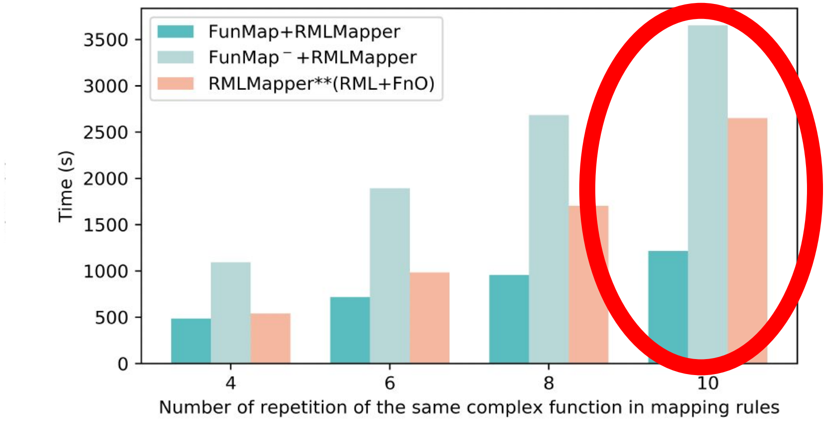


(c) RMLMapper - 25% of duplicates

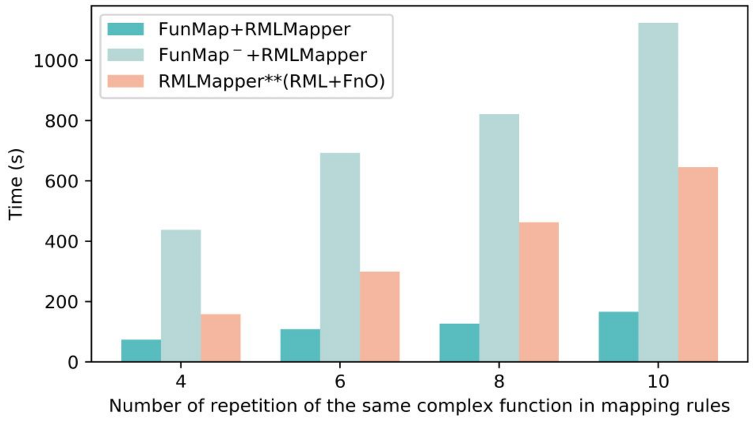


(d) RMLMapper - 75% of duplicates

Complex functions (if, replace, multiple columns)



(c) RMLMapper - 25% of duplicates



(d) RMLMapper - 75% of duplicates

Conclusions

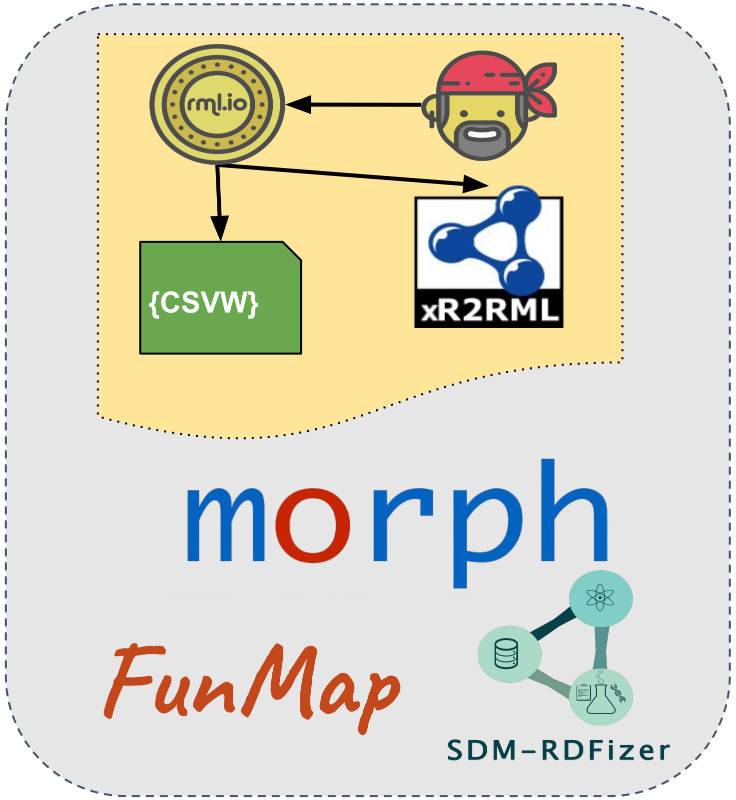
- Efficient techniques for **constructing KGs at scale**
- Empirical results indicate that **SDM-RDFizer outperforms the state of the art by up to three orders of magnitude**
- FunMap converts data integration systems in RML+FnO into **equivalent data integration systems specified in RML**
- FunMap generates data integration systems that **enhance RML-complaint engines**
- Empirical evaluations suggest that **FunMap execution time of RML+FnO is reduced by up to 20 times**

"The terms data cleaning and data pre-processing should be removed"

An Anonymous Data Engineer

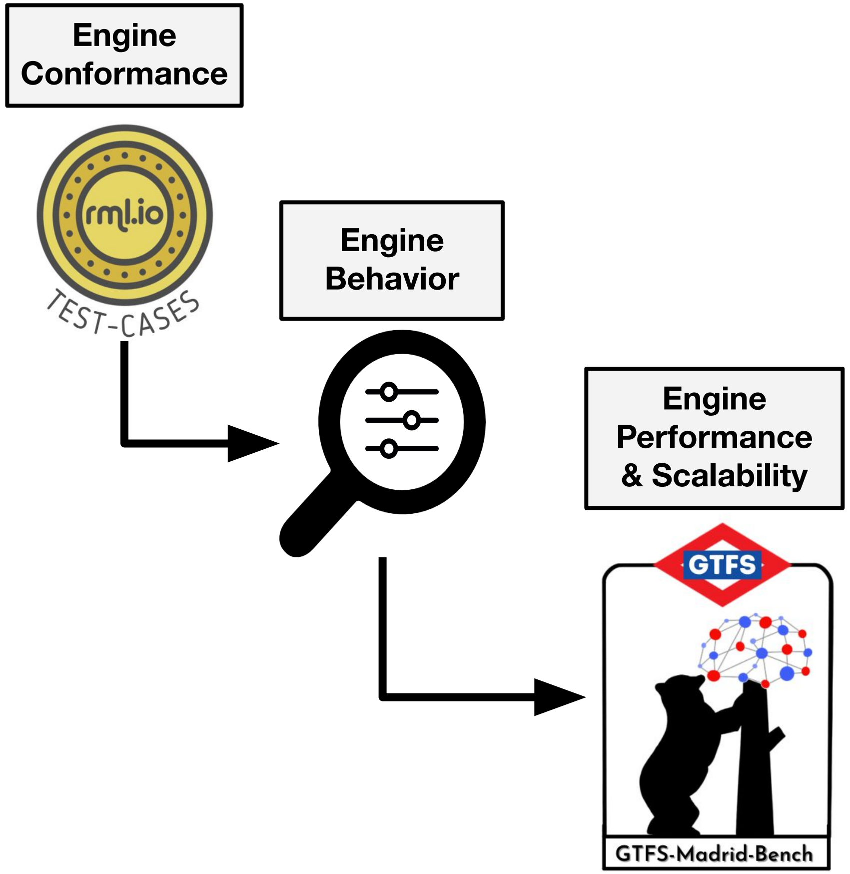
01 Knowledge Graph Construction at Scale

- **Exploitation of mapping rules** to enhance the construction of virtual and materialized knowledge graphs
- **Heterogeneity of mapping** specifications **is a benefit**, not a problem.
- **Pre-processing and data cleaning** steps need more attention.
- From **engineering to research** in materialization approaches.
- **Performance and scalability** is still an **issue** in KG construction.



O2 Evaluation for Knowledge Graph Construction

Proposal	Benchmarking	Objectives	Domain	Testing Features
Morph-CSV	Madrid-GTFS-Bench, BSBM	Implicit Constraints on VKGC	Transportation, E-Commerce	1-1 relation between concepts and sources, Synthetic data generator
Morph-GraphQL	Linköping GraphQL	GraphQL-to-SQL	E-Commerce	1-1 relation between concepts and sources, synthetic data generator
SDM-RDFizer	COSMIC Dataset	Duplicates Removal + Joins	Biomedicine	N-1 relation between ontology concepts and sources, manual testbed generator
FunMap	COSMIC Dataset + Transformation Functions	Duplicates Removal + Function Execution	Biomedicine	N-1 relation between ontology concepts and sources, manual testbed generator, simple and complex functions



Publications: 12 peer-review papers

Contribution	Publication
C1.1: Mapping Translation	J1, J2, J3, J4, C1, C2, W1
C2: Evaluation Framework for KGC	J2, C4, C5
C1.[2|3]: Enhancing virtual KGs	J1, J4, C1, D1, D2
C1.[4|5]: Materialized KGs at scale	C2, C3

J = Journal | C = Conference | D = Demo | W = Workshop

J1 2021 — **Chaves-Fraga, D.**, Ruckhaus, E., Priyatna, F., Vidal, M. E., & Corcho, O. (2021). Enhancing virtual ontology based access over tabular data with Morph-CSV. *Semantic Web* (Q1).

J2 2020 — **Chaves-Fraga, D.**, Priyatna, F., Cimmino, A., Toledo, J., Ruckhaus, E., & Corcho, O. (2020). GTFS-Madrid-Bench: A benchmark for virtual knowledge graph access in the transport domain. *JoWS* (Q2).

J3 2020 — Corcho, O., Priyatna, F., & **Chaves-Fraga, D.** (2020). Towards a new generation of ontology based data access. *Semantic Web* (Q1)

J4 2020 — **Chaves-Fraga, D.**, Priyatna, F., Alobaid, A., & Corcho, O. (2020). Exploiting Declarative Mapping Rules for Generating GraphQL Servers with Morph-GraphQL. *IJSEKE* (Q4)

D1 2020 — **Chaves-Fraga, D.**, Pozo-Gilo, L., Toledo, J., Ruckhaus, E., & Corcho, O. (2020). Morph-CSV: Virtual Knowledge Graph Access for Tabular Data. In *ISWC* (Core A).

C1 2019 — Priyatna, F., **Chaves-Fraga, D.**, Alobaid, A., & Corcho, Ó. (2019). morph-GraphQL: GraphQL Servers Generation from R2RML Mappings. In *SEKE* (Core B).

C2 2020 — Jozashoori, S., **Chaves-Fraga, D.**, Iglesias, E., Vidal, M. E., & Corcho, O. (2020, November). FunMap: Efficient Execution of Functional Mappings for Knowledge Graph Creation. In *International Semantic Web Conference* (Core A)

C3 2020 — Iglesias, E., Jozashoori, S., **Chaves-Fraga, D.**, Collarana, D., & Vidal, M. E. (2020). SDM-RDFizer: An RML interpreter for the efficient creation of RDF knowledge graphs. In *Proceedings of the 29th ACM CIKM* (Core A)

C4 2019 — **Chaves-Fraga, D.**, Endris, K. M., Iglesias, E., Corcho, O., & Vidal, M. E. (2019). What are the Parameters that Affect the Construction of a Knowledge Graph?. In *ODBASE.*

C5 2019 — Heyvaert, P., **Chaves-Fraga, D.**, Priyatna, F., Corcho, O., Mannens, E., Verborgh, R., & Dimou, A. (2019). Conformance test cases for the RDF mapping language (RML). In *Iberoamerican KGSWC*

W1 2018 — **Chaves-Fraga, D.**, Priyatna, F., Santana-Pérez, I., & Corcho, O. (2018). Virtual Statistics Knowledge Graph Generation from CSV files. In *SemStats-ISWC* (Best Workshop Papers)

D2 2019 — Alobaid, A., **Chaves-Fraga, D.**, Priyatna, F., & Corcho, Ó. (2019). GraphQL Servers generation from R2RML with morph-GraphQL (D). In *SEKE* (Core B). Best Demo Award

14 related papers

J Goncalves, M., **Chaves-Fraga, D.**, & Corcho, O. (2021). Handling Qualitative Preferences in SPARQL over Virtual Ontology-Based Data Access. In *Semantic Web (Under Review)*

C Corcho, O., **Chaves-Fraga, D.**, et al (2021). A High-Level Ontology Network for ICT Infrastructures. In *International Semantic Web Conference (Resource Track)*.

C Goncalves, M., **Chaves-Fraga, D.**, & Corcho, O. (2020). Morph-Skyline: Virtual Ontology-Based Data Access for Skyline Queries. In *International Joint Conference On Web Intelligence And Intelligent Agent Technology (WI-IAT'20)*

C Iglesias-Molina, A., **Chaves-Fraga, D.**, Priyatna, F., & Corcho, O. (2019). Enhancing the Maintainability of the Bio2RDF Project Using Declarative Mappings. In *Proceedings of the 12th International Conference on Semantic Web Applications and Tools for Healthcare and Life Sciences*.

D Goncalves, M., **Chaves-Fraga, D.**, & Corcho, O. (2020). Morph-Skyline: Skyline Queries for Virtual Knowledge Graph Access. In *International Semantic Web Conference (Posters, Demos & Industry Tracks)*.

D Rojas, J., **Chaves-Fraga, D.**, Colpaert, P., Verborgh, R., & Mannens, E. (2017). Providing Reliable Access to Real-Time and Historic Public Transport Data Using Linked Connections. In *International Semantic Web Conference (Posters, Demos & Industry Tracks)*.

D Iglesias-Molina, A., Pozo-Gilo, L., Dona, D., Ruckhaus, E., **Chaves-Fraga, D.**, & Corcho, Ó. (2020). Mapeathor: Simplifying the Specification of Declarative Rules for Knowledge Graph Construction. In *International Semantic Web Conference (Posters, Demos & Industry Tracks)*.

W Arenas-Guerrero, J., Scrocca, M., Iglesias-Molina, A., Toledo, J., Pozo-Gilo, L., Dona, D., Corcho, O., & **Chaves-Fraga, D.** (2021). Knowledge Graph Construction with R2RML and RML: An ETL System-based Overview. *In Proceedings of the 2nd International Workshop on Knowledge Graph Construction (ESWC)*.

W **Chaves-Fraga, D.**, Antón, A., Toledo, J., & Corcho, O. (2019). ONETT: Systematic Knowledge Graph Generation for National Access Points. In *1st International Workshop on Semantics for Transport (SEMANTICS)*.

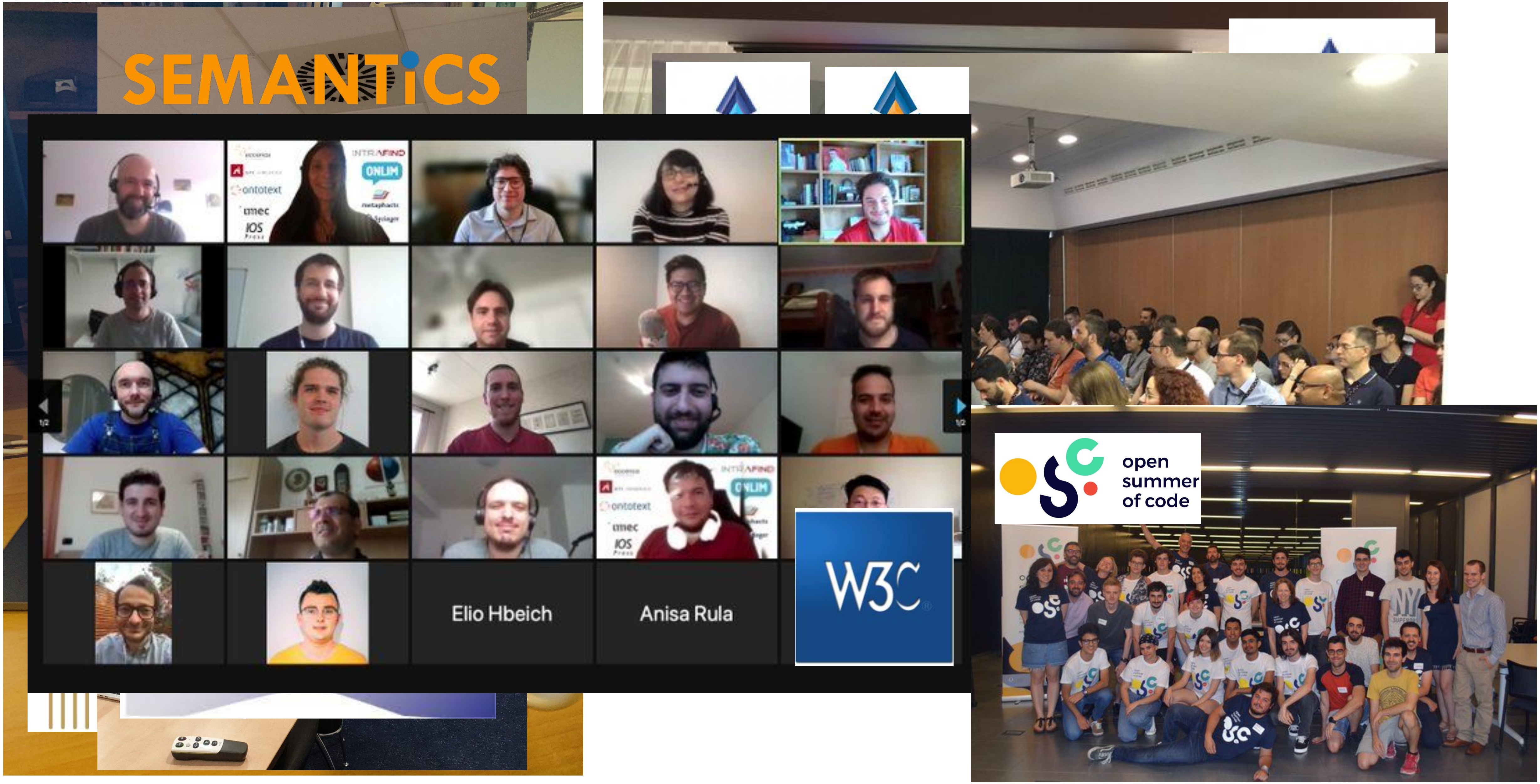
W **Chaves-Fraga, D.**, Rojas, J., Vandenberghe, P. J., Colpaert, P., & Corcho, O. (2017). The tripscore Linked Data client: calculating specific summaries over large time series. *In Proceedings of the 1st International Workshop on Decentralizing the Semantic Web (ISWC).*

W **Chaves-Fraga, D.**, Gutiérrez, C., & Corcho, O. (2017). On the Role of the GRAPH Clause in the Performance of Federated SPARQL Queries. In *International Workshop on Dataset PROFIling & Search (ISWC.*)

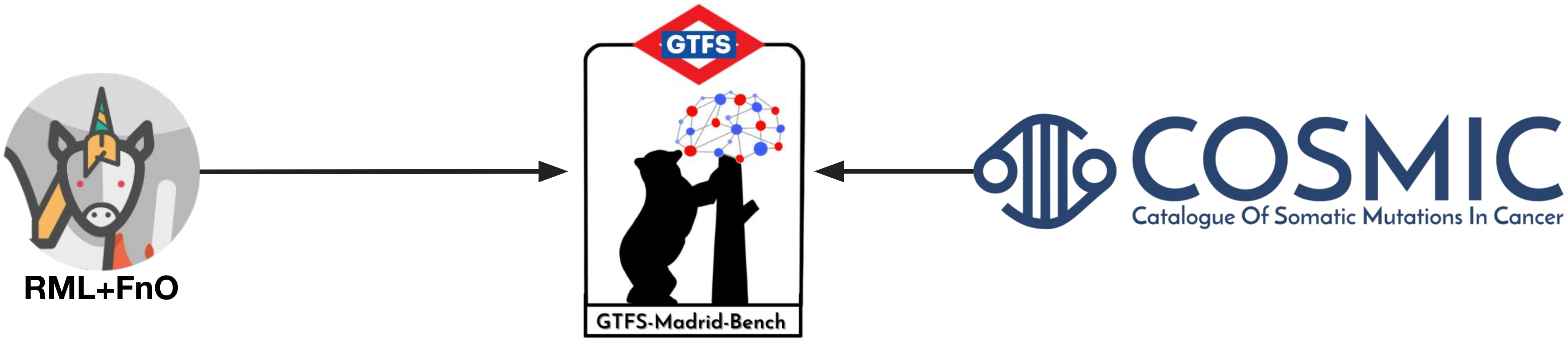
W Iglesias-Molina, A., **Chaves-Fraga, D.**, Priyatna, F., & Corcho, O. (2019). Towards the definition of a language-independent mapping template for knowledge graph creation. In *Proceedings of the Third International Workshop on Capturing Scientific Knowledge co-located with the 10th International Conference on Knowledge Capture (K-CAP 2019)*.

P Carriero Valentina, **Chaves-Fraga D.** et al. (2018). The Jedi Approach: Using The Force to Solve Linked Data Incompleteness. In *Linked Open Data Validity - A Technical Report from ISWS 2018*

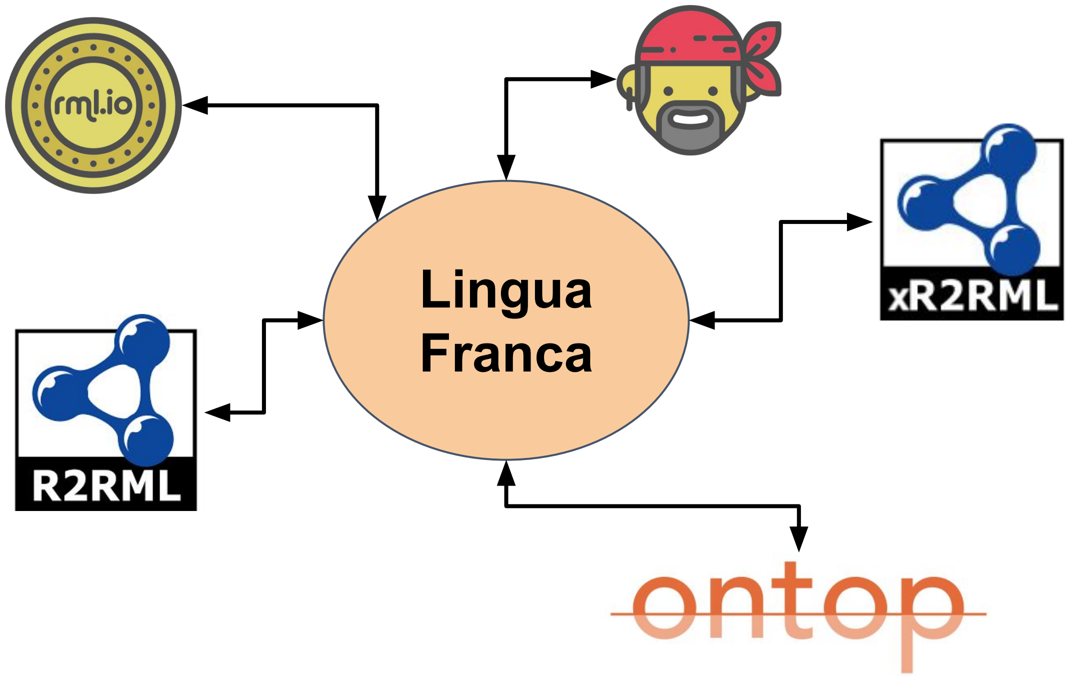
P Badenes-Olmedo, C., **Chaves-Fraga, D.**, et al. (2020). Drugs4Covid: Drug-driven Knowledge Exploitation based on Scientific Publications. *arXiv preprint arXiv:2012.01953*


SEMANTiCS
ontotext
imec
IOS Press
INTRAFIND
ONLIM
Elio Hbeich
Anisa Rula
W3C
open summer of code

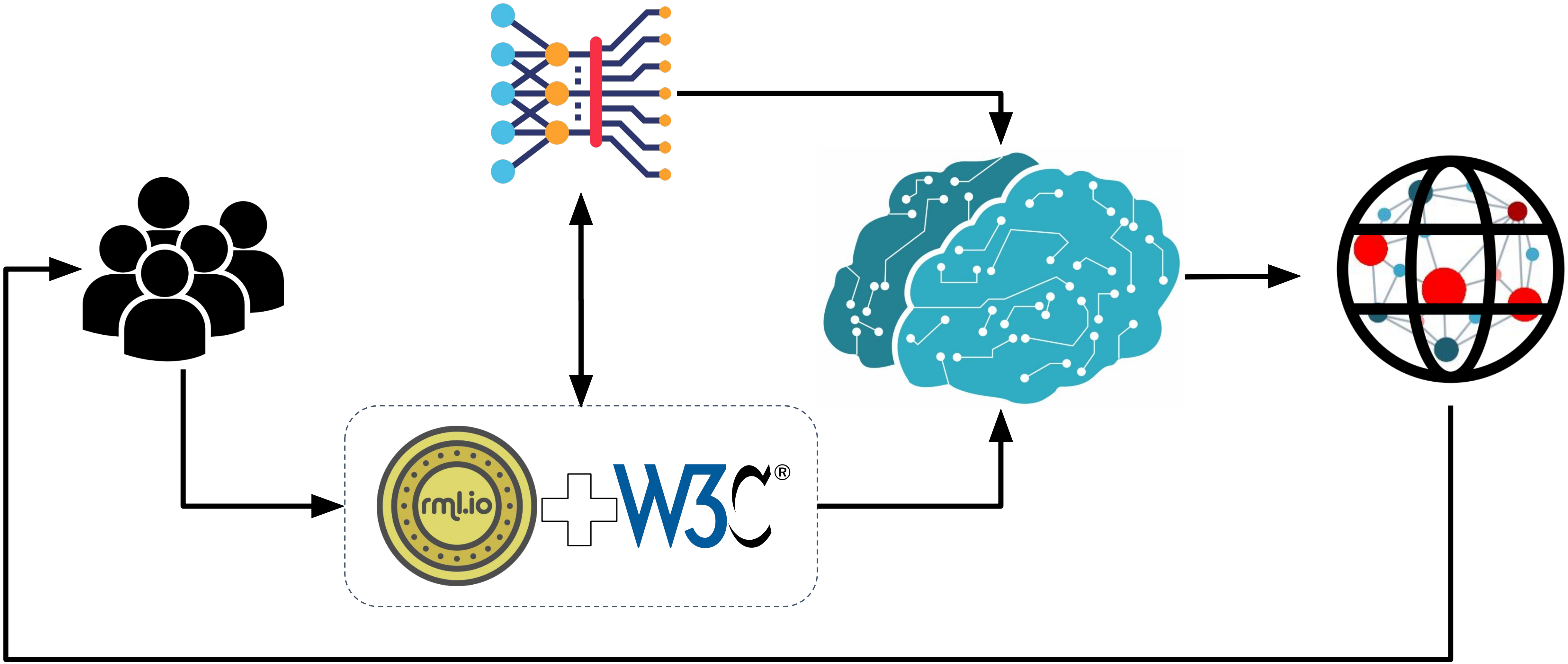
Future Work


RML+FnO
GTFS
GTFS-Madrid-Bench
COSMIC
Catalogue Of Somatic Mutations In Cancer


morph
FunMap
SDM-RDFizer


rml.io
Lingua Franca
xR2RML
R2RML
ontop

Users in the loop for hybrid knowledge graph construction solutions



Knowledge Graph Construction from Heterogeneous Data Sources Exploiting Declarative Mapping Rules

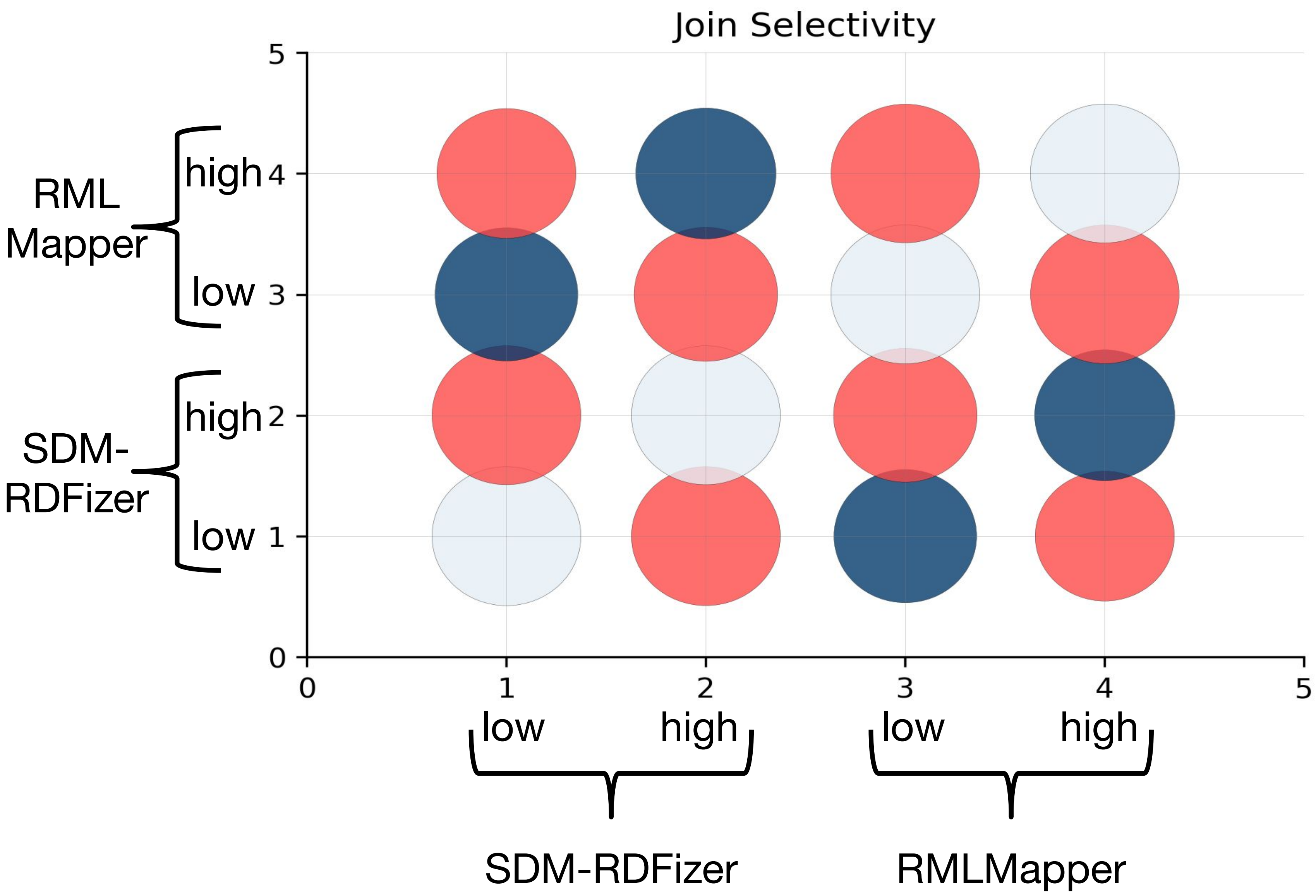
David Chaves-Fraga, Ontology Engineering Group
Universidad Politécnica de Madrid, Spain

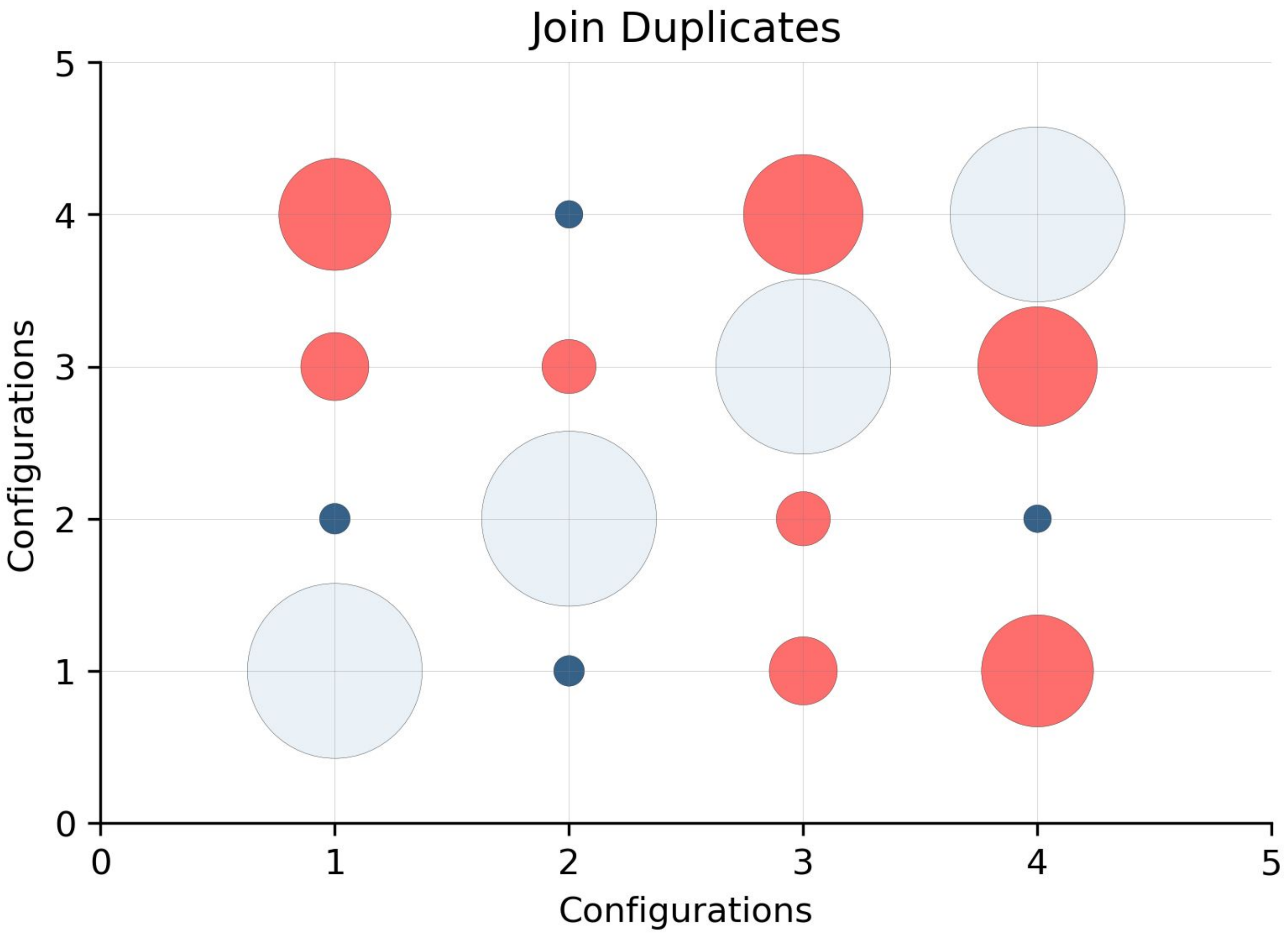
Supervisor: Oscar Corcho

dchaves@fi.upm.es dchavesf

28/06/2021 Online

Results IV: Join Selectivity on 1K





Configurations 1-2: SDM-RDFizer on low and high duplicates
Configurations 3-4: RMLMapper on low and high duplicates

Challenges VS Declarative Annotations in Tabular Data

Challenge	Detailed Challenge	Relevant properties
Updated results	Select relevant sources and columns	SPARQL + Mapping
Lightweight Schema	Describing concepts and properties	rr: class / rr:predicateMap
	Add header to a CSV file	csvw:rowTitles
	Column datatype	csvw:datatype
Heterogeneity	Domain values	csvw:minimum, csvw:maximum
	Specify the format of a column	csvw:format
	Transform value	fnml:functionValue
	Default for missing values	csvw:default
	Specify NULL values	csvw:null
	Specify NOT NULL constraint	csvw:required
Not Normalized Sources	Integrity Constraints (PK & FK)	csvw:primaryKey / csvw:foreignKey
	Relationships between columns	rr:parentTriplesMap + rr:joinCondition
	Multiple entities in one source	rr:TriplesMap with same LogicalSource
	Support for multiple values in one cell	csvw:separator

Step 1: Source selection

```
SELECT ?trip ?routeName ?routeType ?startTime ?endTime ?code WHERE {
  ?trip a gtfs:Trip .
  ?trip gtfs:route ?route .

  ?frequency a gtfs:Frequency .
  ?frequency gtfs:startTime ?startTime .
  ?frequency gtfs:endTime ?endTime .
  ?frequency gtfs:trip ?trip .

  ?route a gtfs:Route .
  ?route gtfs:shortName ?routeName .
  ?route gtfs:routeType ?routeType .

  ?routeType gtfs:routeTypeCode ?code   }
```

Select the relevant rules for the star shape groups of the input query

```
frequencies:
  sources:
    - [frequencies.csv~csv]
  s: mbench:freq/$(trip_id)-$(start_time)
  po:
    - [a, gtfs:Frequency]
    - [gtfs:startTime,$(start_time)]
    - [gtfs:endTime,$(end_time)]
    - [gtfs:headSecs,$(headway_secs)]
    - [gtfs:exactTimes,$(exact_times)]
    - p: gtfs:trip
      o:
        - mapping: trips
          condition:
            function: equal
            parameters:
              - [str1, $(trip_id)]
              - [str2, $(trip_id)]
trips:
  sources:
    - [trips.csv~csv]
  s: mbench:trips/$(trip_id)
  po:
    - [a, gtfs:Trip]
    - [gtfs:headsign, $(trip_headsign)]
    - [gtfs:shortName, $(trip_short_name)]
    - [gtfs:direction, $(direction_id)]
    - [gtfs:block, $(block_id)]
    - p: gtfs:route
      o:
        - mapping: routes
          condition:
            function: equal
            parameters:
              - [str1, $(route_id)]
              - [str2, $(route_id)]
routes:
  sources:
    - [routes.csv~csv]
  s: mbench:routes/$(route_id)
  po:
    - [a, gtfs:Route]
    - [gtfs:shortName, $(route_short_name)]
    - [gtfs:longName, $(route_long_name)]
    - [dct:description, $(route_desc)]
    - [gtfs:routeUrl, $(route_url)~iri]
    - [gtfs:color, $(route_color)]
    - [gtfs:textColor, $(route_text_color)]
    - p: gtfs:agency
      o:
        - mapping: agency
          condition:
            function: equal
            parameters:
              - [str1, $(agency_id)]
              - [str2, $(agency_id)]
    - p: gtfs:RouteType
      o:
        - mapping: route-type
          condition:
            function: equal
            parameters:
              - [str1, $(route_type)]
              - [str2, $(route_type)]
route-type:
  sources:
    - [routes.csv~csv]
  s: CONCAT(gtfs:,TRANS($(route_type)))
  po:
    - [a, gtfs:RouteType]
    - [gtfs:routeTypeCode,$(route_code)]
```

route_id	agency_id	route_short_name	route_long_name	route_type	route_code	route_url	route_color
4_1	CRTM	1	Chamartín-Valdecarros	1	401	http://crtm/metro/4_1	2DBEF0
4_2	CRTM	2	Las Rosas - C. Caminos	1	401	http://crtm/metro/4_2	ED1C24
4_3	CRTM	3	Villaverde Alto-Moncloa	1	401	http://crtm/metro/4_3	FFD000
4_4	CRTM	4	Chamartín-Argüelles	1	401	http://crtm/metro/4_4	B65518
5_C1	CRTM	C1	P.Pío-AeropuertoT4	2	109	http://crtm/train/5_1	4FB0E5
5_C2	CRTM	C2	Guadalajara-Chamartín	2	109	http://crtm/train/5_2	008B45
5_C3	CRTM	C3	Aranjuez-Escorial	2	109	http://crtm/train/5_3	9F2E86
5_C4	CRTM	C4	Parla-Colmenar Viejo	2	109	http://crtm/train/5_4	005AA3

Step 1: Source selection

```
frequencies:
  sources:
    - [frequencies.csv~csv]
  s: mbench:freq/$(trip_id)-$(start_time)
  po:
    - [a, gtfs:Frequency]
    - [gtfs:startTime,$(start_time)]
    - [gtfs:endTime,$(end_time)]
    - p: gtfs:trip
      o:
        - mapping: trips
          condition:
            function: equal
            parameters:
              - [str1, $(trip_id)]
              - [str2, $(trip_id)]
trips:
  sources:
    - [trips.csv~csv]
  s: mbench:trips/$(trip_id)
  po:
    - [a, gtfs:Trip]
    - p: gtfs:route
      o:
        - mapping: routes
          condition:
            function: equal
            parameters:
              - [str1, $(route_id)]
              - [str2, $(route_id)]
routes:
  sources:
    - [routes.csv~csv]
  s: mbench:routes/$(route_id)
  po:
    - [a, gtfs:Route]
    - [gtfs:longName, $(route_long_name)]
    - p: gtfs:RouteType
      o:
        - mapping: route-type
          condition:
            function: equal
            parameters:
              - [str1, $(route_type)]
              - [str2, $(route_type)]

route-type:
  sources:
    - [routes.csv~csv]
  s: CONCAT(gtfs:,TRANS($(route_type)))
  po:
    - [a, gtfs:RouteType]
    - [gtfs:routeTypeCode,$(route_code)]
```

Routes and route-types has the same LogicalSource

⇩

Source contains information from independent entities

route_id	route_long_name	route_type	route_code
4_1	Chamartín-Valdecarros	1	401
4_2	Las Rosas - C. Caminos	1	401
4_3	Villaverde Alto-Moncloa	1	401
4_4	Chamartín-Argüelles	1	401
5_C1	P.Pío-AeropuertoT4	2	109
5_C2	Guadalajara-Chamartín	2	109
5_C3	Aranjuez-Escorial	2	109
5_C4	Parla-Colmenar Viejo	2	109

Step 2: Normalization

route_id	route_long_name	route_type
4_1	Chamartín-Valdecarros	1
4_2	Las Rosas - C. Caminos	1
4_3	Villaverde Alto-Moncloa	1
4_4	Chamartín-Argüelles	1
5_C1	P.Pío-AeropuertoT4	2
5_C2	Guadalajara-Chamartín	2
5_C3	Aranjuez-Escorial	2
5_C4	Parla-Colmenar Viejo	2

route_type	route_code
1	401
1	401
1	401
1	401
2	109
2	109
2	109
2	109

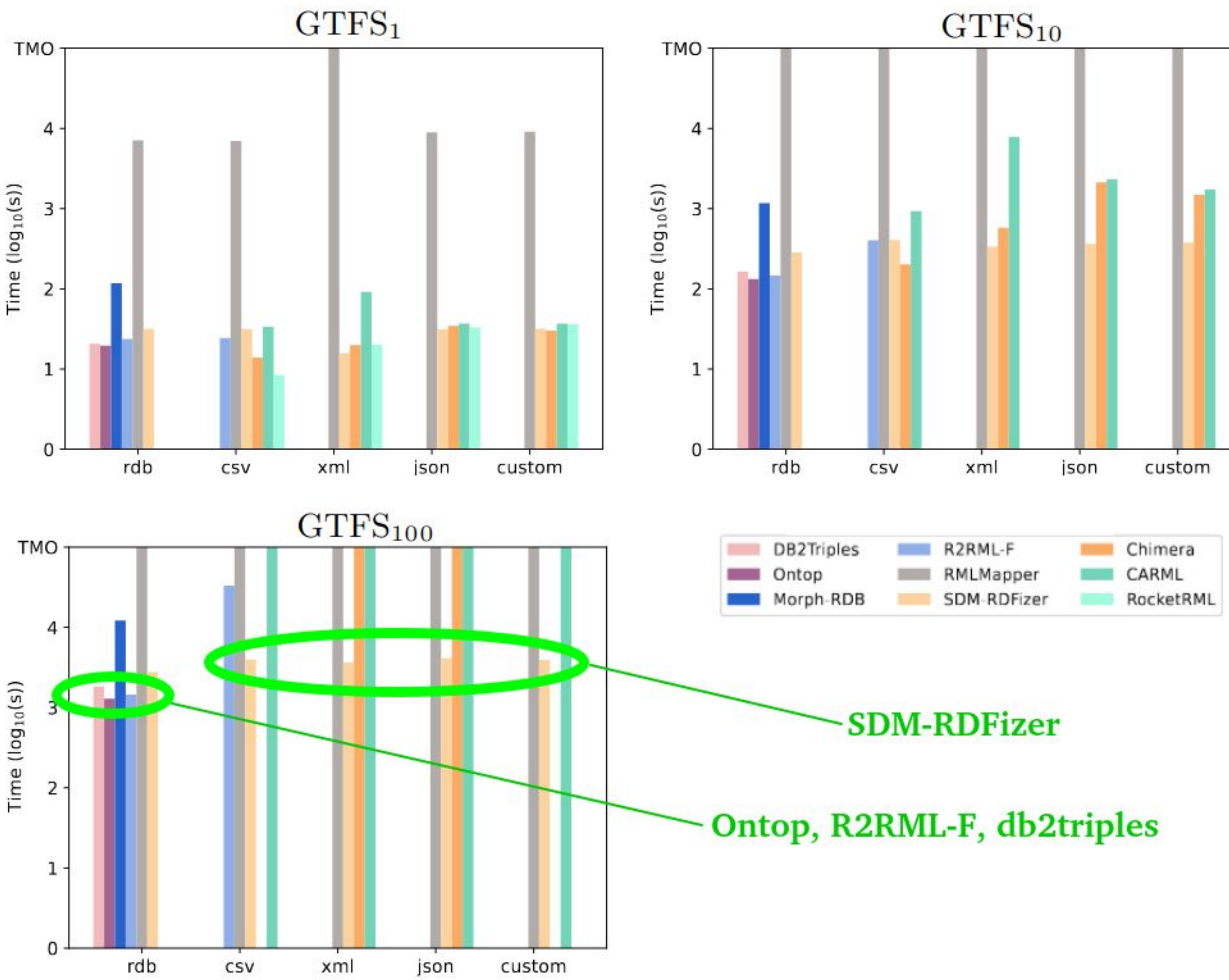
1) **route TriplesMap references:** route_id, route_long_name, route_type
2) **route_type TriplesMap references:** route_type, route_code

route_type	route_code	route_type_fn
1	401	Subway
2	109	Train

```
route-type:
   sources:
     - [routes_types.csv~csv]
   s: gtfs:$(route_type_fn)
    po:
     - [gtfs:routeTypeCode,$(route_code)]
```

1. Remove duplicates from the input sources (from 8 to 2 rows)
2. Substitute values from CSVW annotations (csvw:default, csvw:null, csvw:format) directly over tabular data
3. Generate the SQL actions for the ad-hoc transformation functions (fnml:functionValue)

```
UPDATE route_type_table SET route_type_fn = REPLACE(route_type, '1', 'Subway')
```



GTFS1
GTFS10
GTFS100
TMO
4
3
2
1
0
Time (log10(s))
rdb
csv
xml
json
custom
DB2Triples
Ontop
Morph-RDB
R2RML-F
RMLMapper
SDM-RDFizer
Chimera
CARML
RocketRML
SDM-RDFizer
Ontop, R2RML-F, db2triples

Memory Consumption