How to Develop Mapping Rules for Knowledge Graph Creation

Ana Iglesias-Molina & David Chaves-Fraga

✉ dataintegration@delicias.dia.fi.upm.es

@oeg-upm

07/11/2019

OEG Talk

Motivation

- Most common methods to generate RDF:
  - OpenRefine
  - Ad-hoc solutions (scripts)
- Reproducibility?
- Maintainability?
- Big Data? Volume, Variety, Velocity

Motivation

- Most common methods to generate RDF:
  - OpenRefine
  - Ad-hoc solutions (scripts)
- Reproducibility?
- Maintainability?
- Big Data? Volume, Variety, Velocity

→ Declarative Mappings

Table of Contents

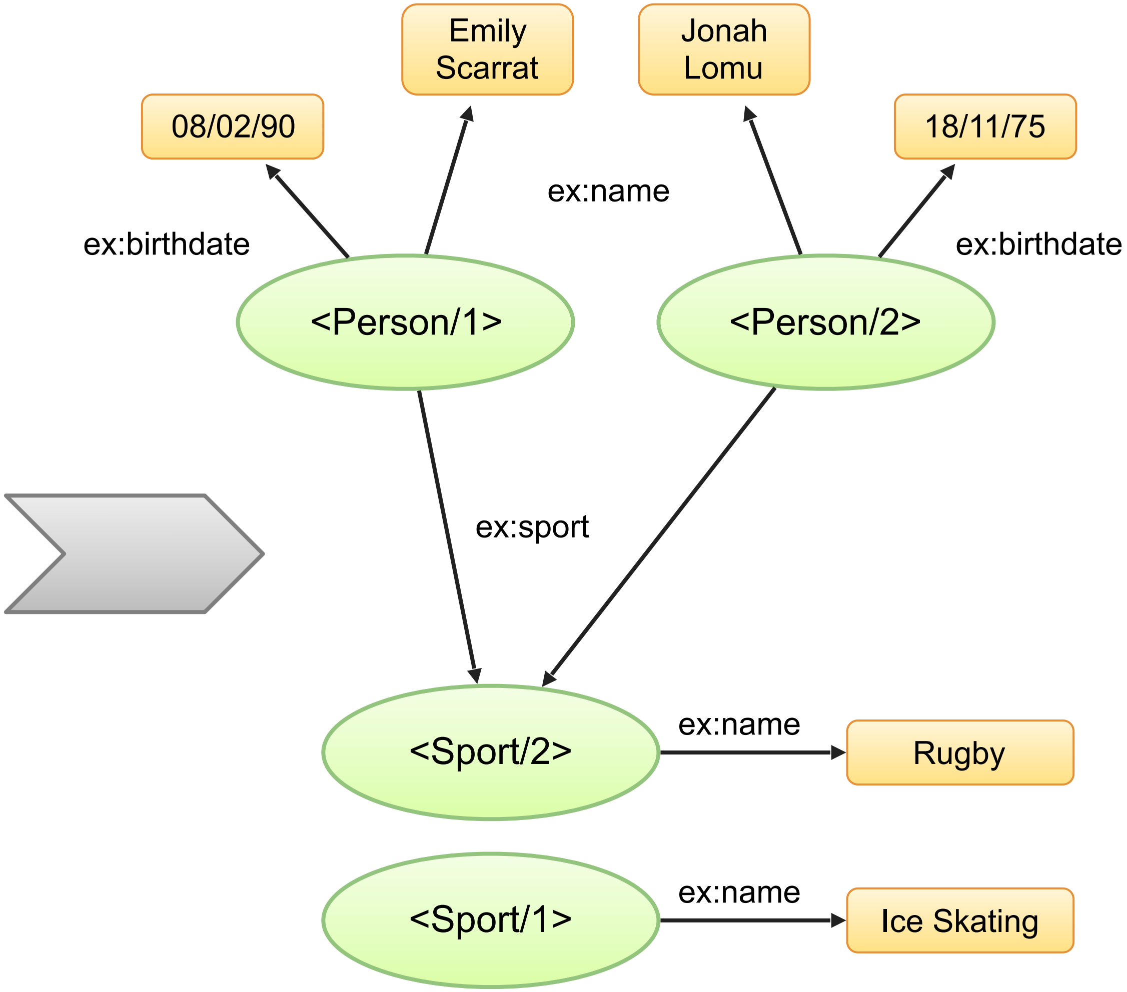
- Introduction
- History
- Structure
- User-friendly options
  - YARRRML
  - Domain Specific Languages
  - Spreadsheets as mappings

Source Data

People			
ID	**Name**	**Birthdate**	**SportID**
1	Emily Scarrat	19900208	2
2	Jonah Lomu	19751118	2

Sports	
ID	**Sport**
1	Ice Skating
2	Rugby

Knowledge Graph



Source Data

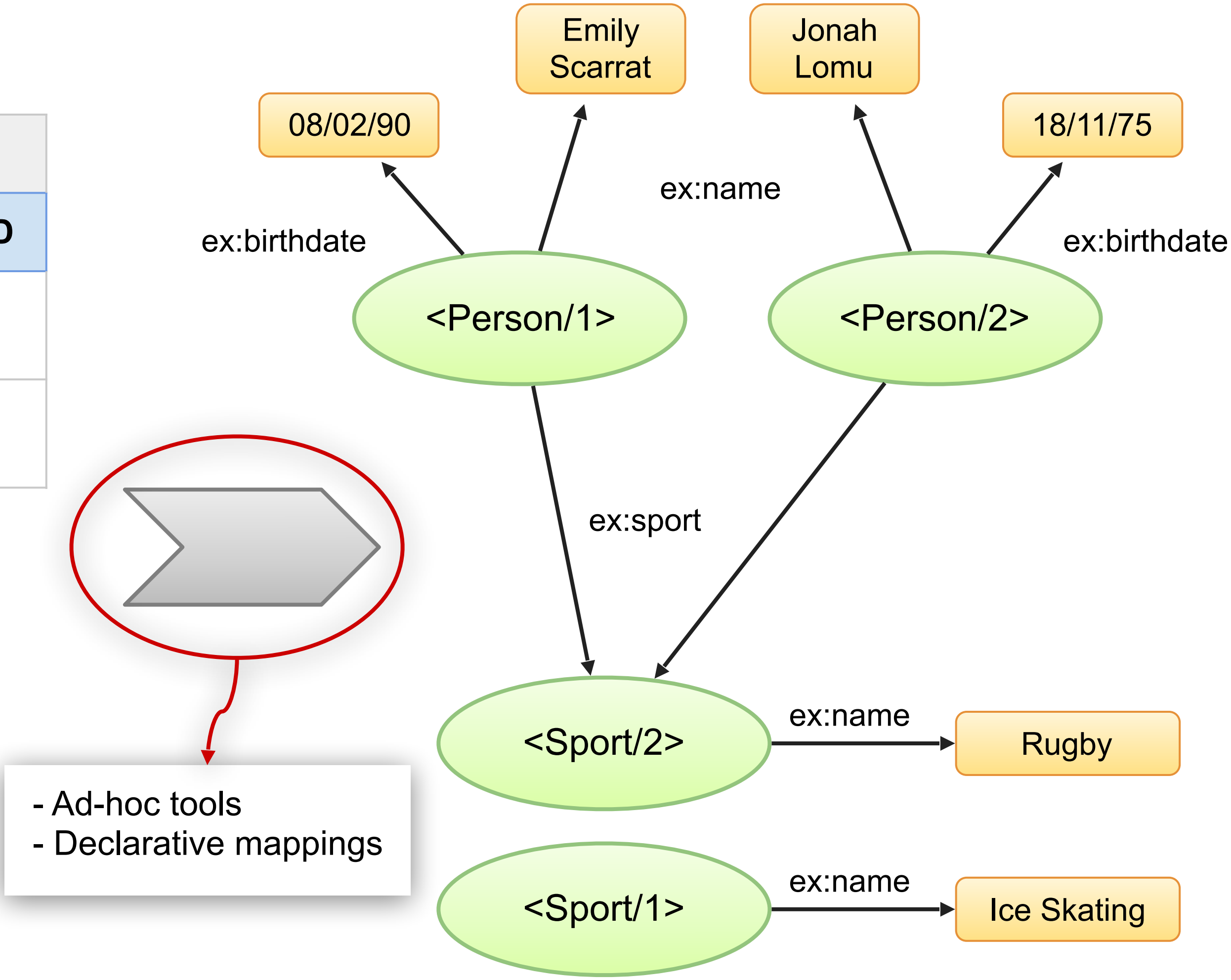
People

ID	Name	Birthdate	SportID
1	Emily Scarrat	19900208	2
2	Jonah Lomu	19751118	2

Sports

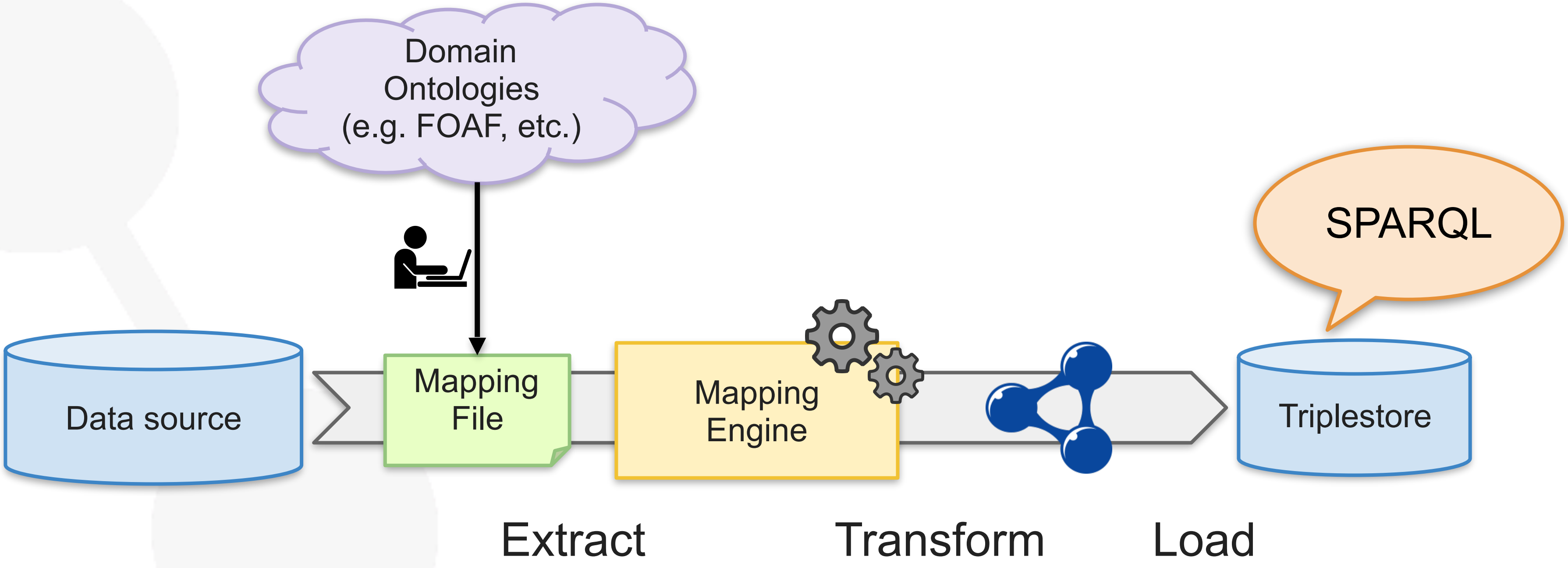
ID	Sport
1	Ice Skating
2	Rugby

Knowledge Graph



- Ad-hoc tools
- Declarative mappings

- Input
  - Data sources
  - Target ontologies
  - Mappings between the database and target ontologies
- Output
  - RDF graph



History

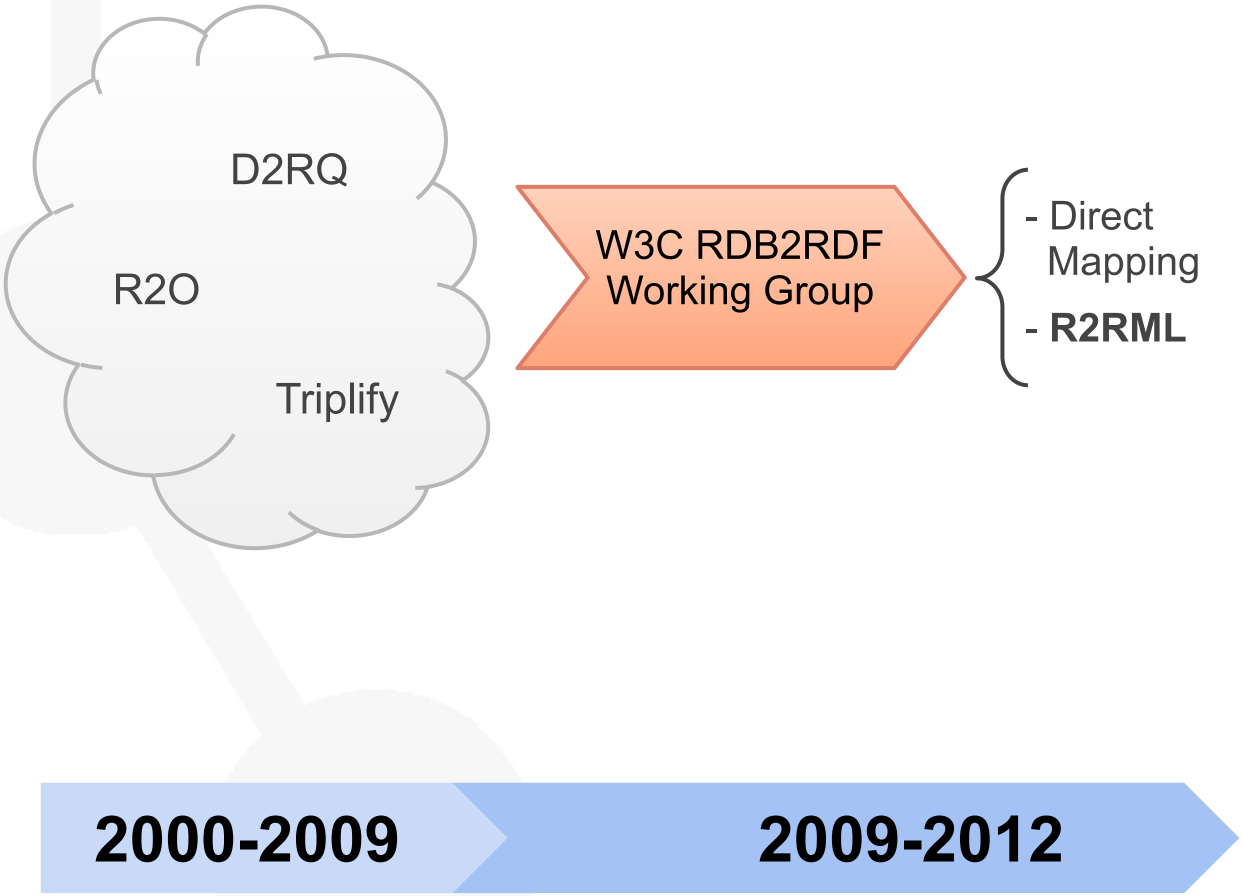
D2RQ

R2O

Triplify

2000-2009

History



D2RQ

R2O

Triplify

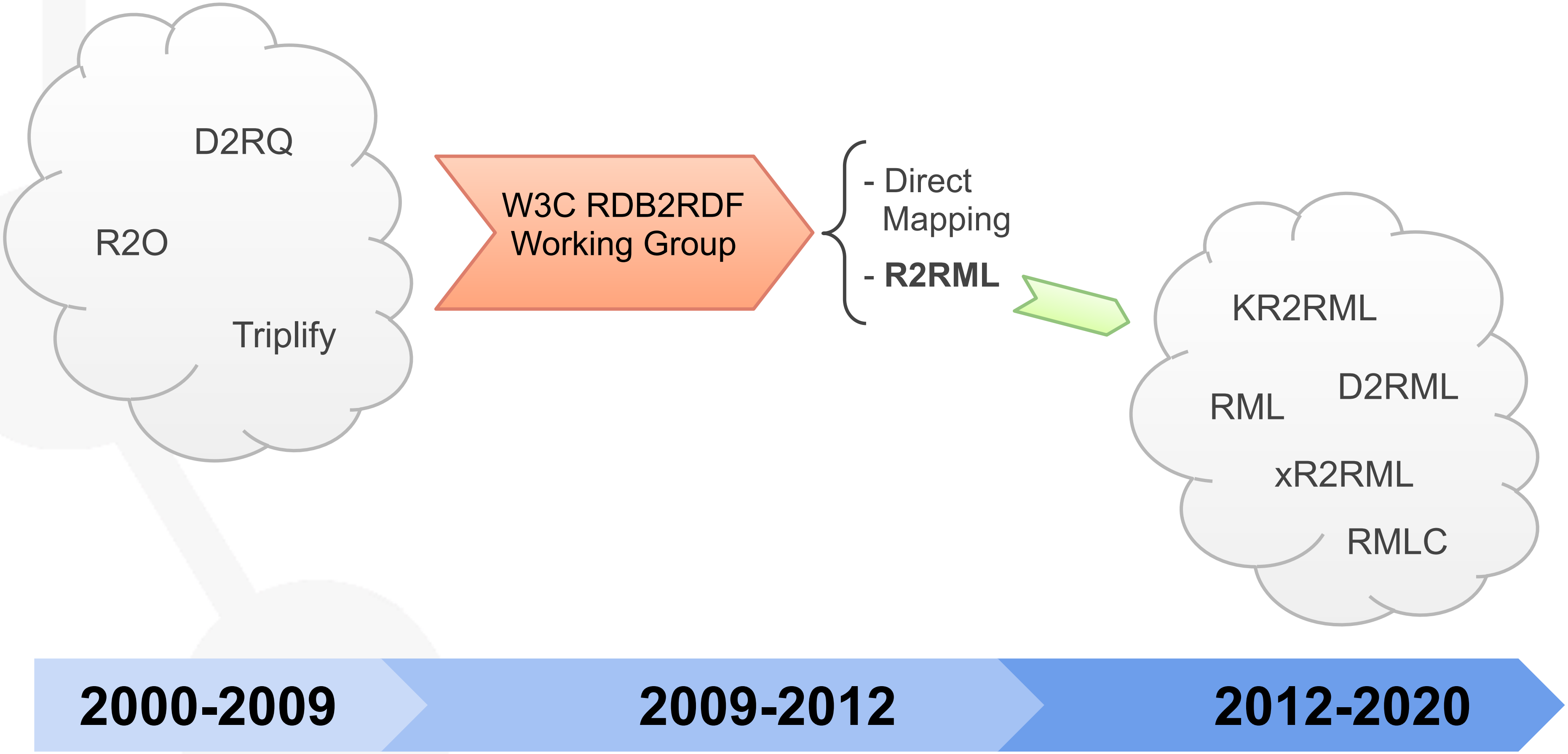
W3C RDB2RDF Working Group

- Direct Mapping
- **R2RML**

2000-2009

2009-2012

History



D2RQ

R2O

Triplify

W3C RDB2RDF Working Group

- Direct Mapping
- **R2RML**

KR2RML

RML

D2RML

xR2RML

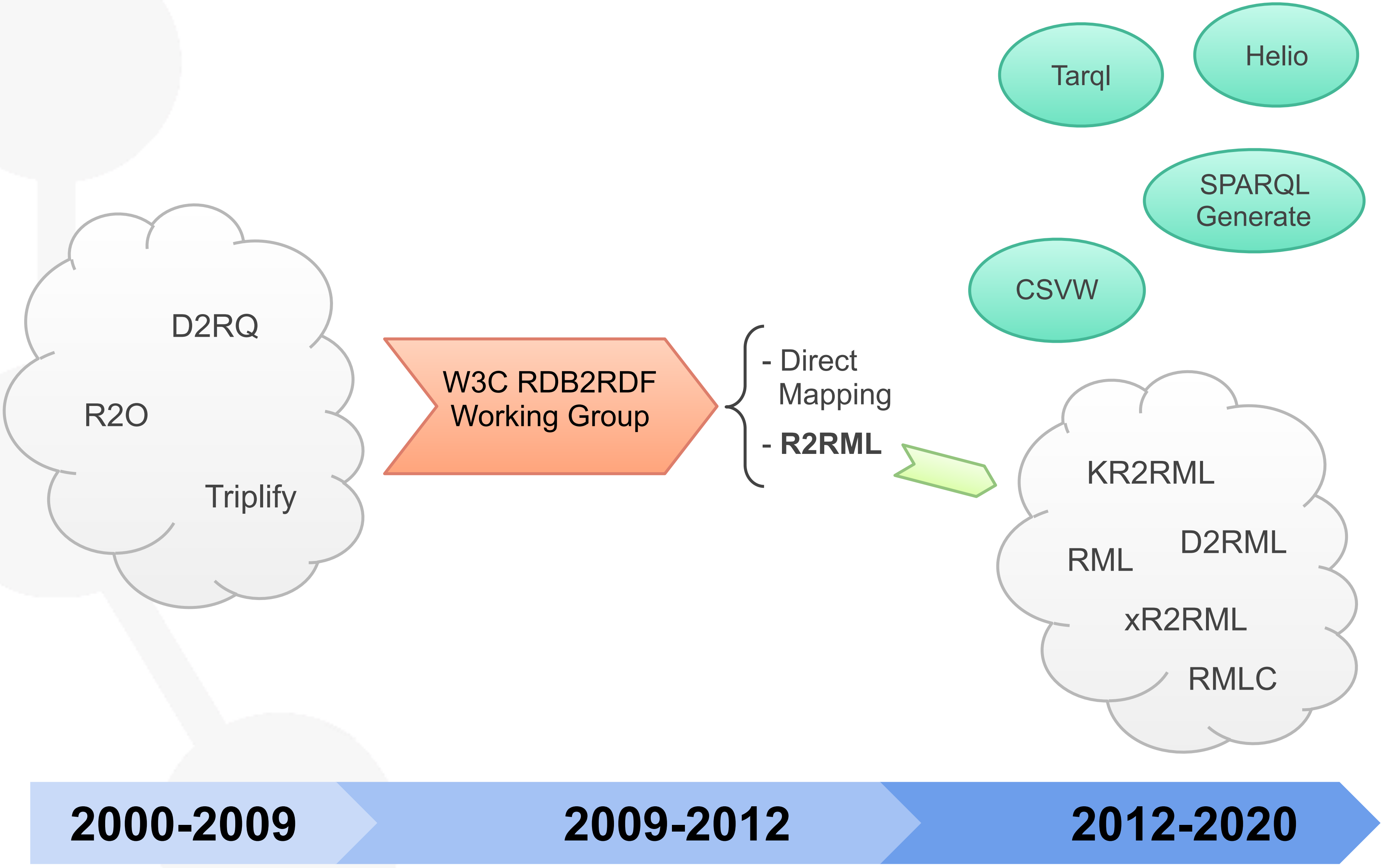
RMLC

2000-2009

2009-2012

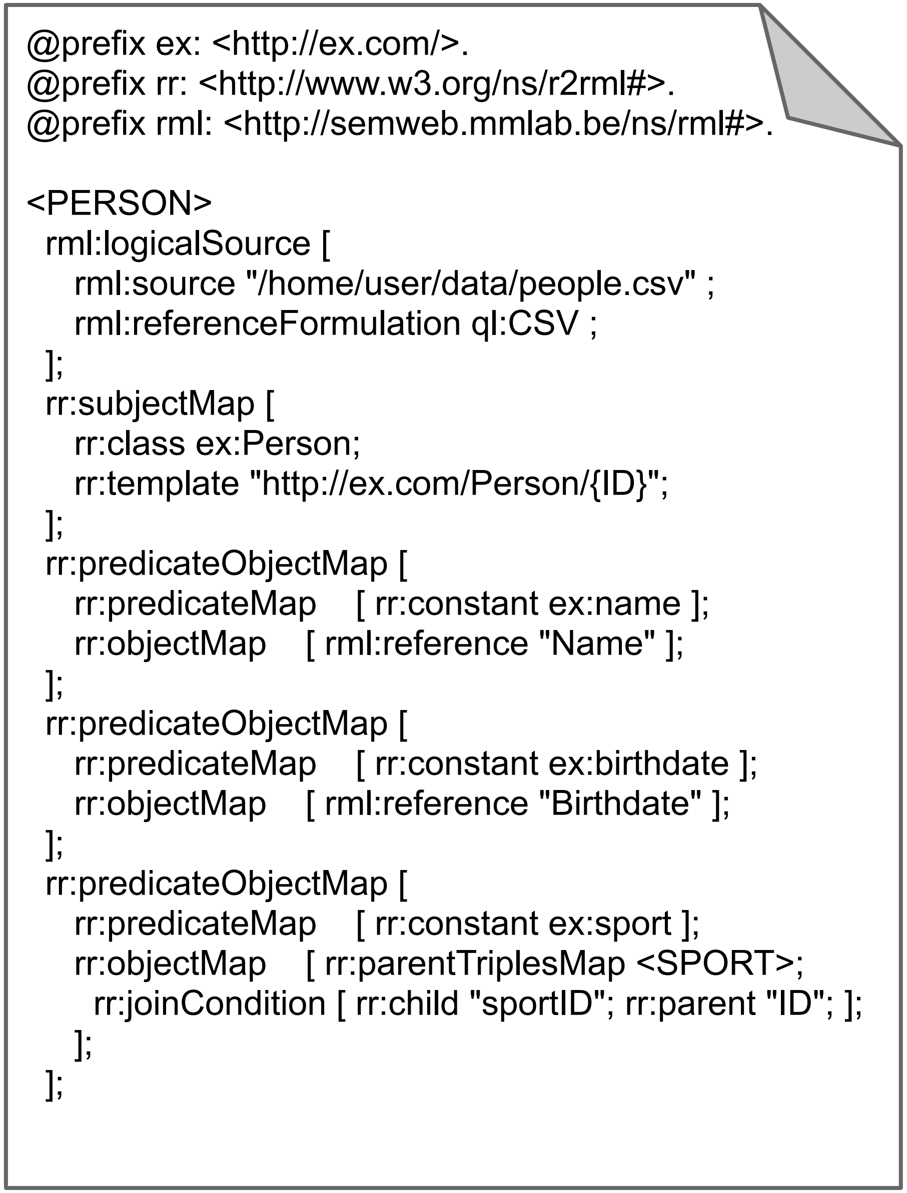
2012-2020

History


D2RQ
R2O
Triplify
W3C RDB2RDF Working Group
- Direct Mapping
- **R2RML**
Tarql
Helio
SPARQL Generate
CSVW
KR2RML
RML
D2RML
xR2RML
RMLC
2000-2009
2009-2012
2012-2020

Mapping structure (RML)

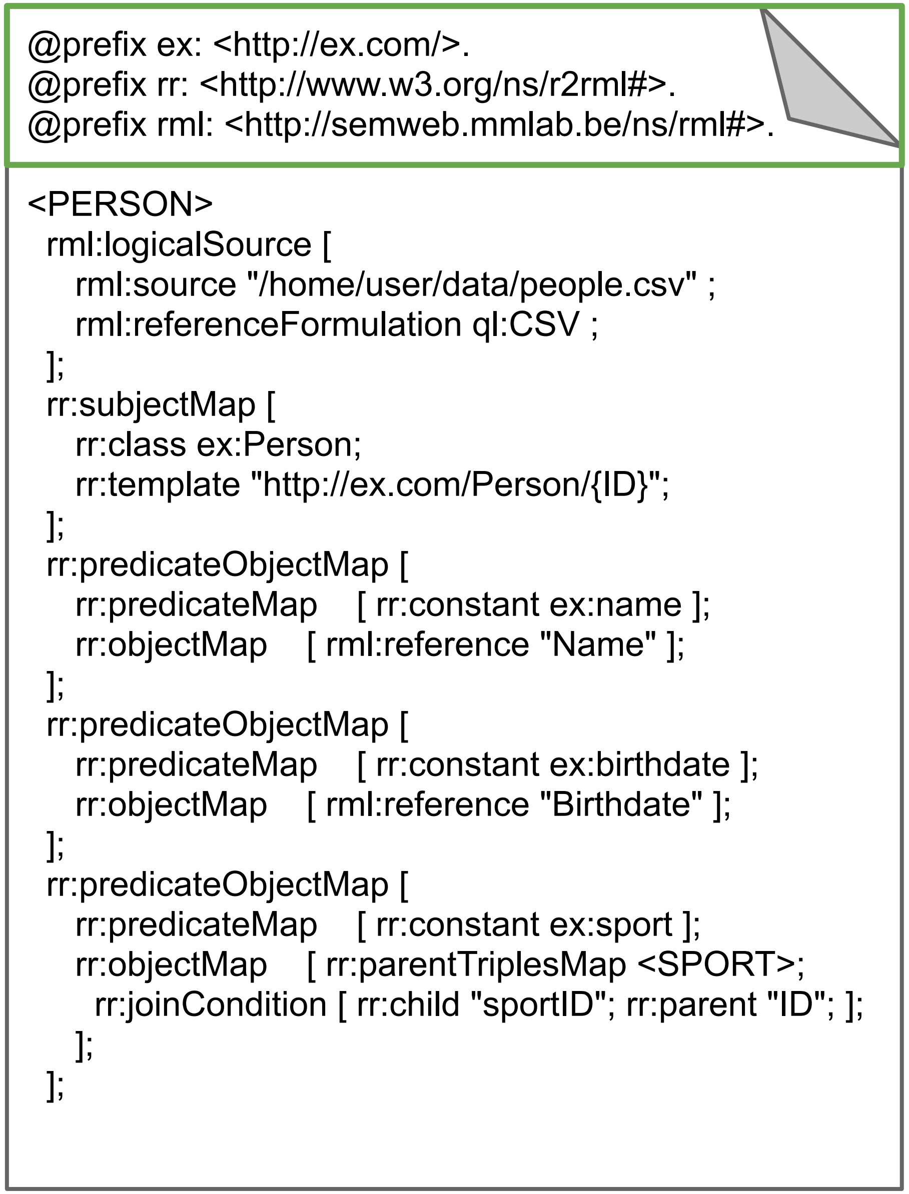
- Prefixes
- Triple Map
  - Source
  - Subject
  - Predicate-Object
  - Reference Object



```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
  rml:logicalSource [
     rml:source "/home/user/data/people.csv" ;
     rml:referenceFormulation ql:CSV ;
  ];
  rr:subjectMap [
     rr:class ex:Person;
     rr:template "http://ex.com/Person/{ID}";
  ];
  rr:predicateObjectMap [
     rr:predicateMap    [ rr:constant ex:name ];
     rr:objectMap    [ rml:reference "Name" ];
  ];
  rr:predicateObjectMap [
     rr:predicateMap    [ rr:constant ex:birthdate ];
     rr:objectMap    [ rml:reference "Birthdate" ];
  ];
  rr:predicateObjectMap [
     rr:predicateMap    [ rr:constant ex:sport ];
     rr:objectMap    [ rr:parentTriplesMap <SPORT>;
       rr:joinCondition [ rr:child "sportID"; rr:parent "ID"; ];
     ];
  ];
```

Mapping structure (RML)

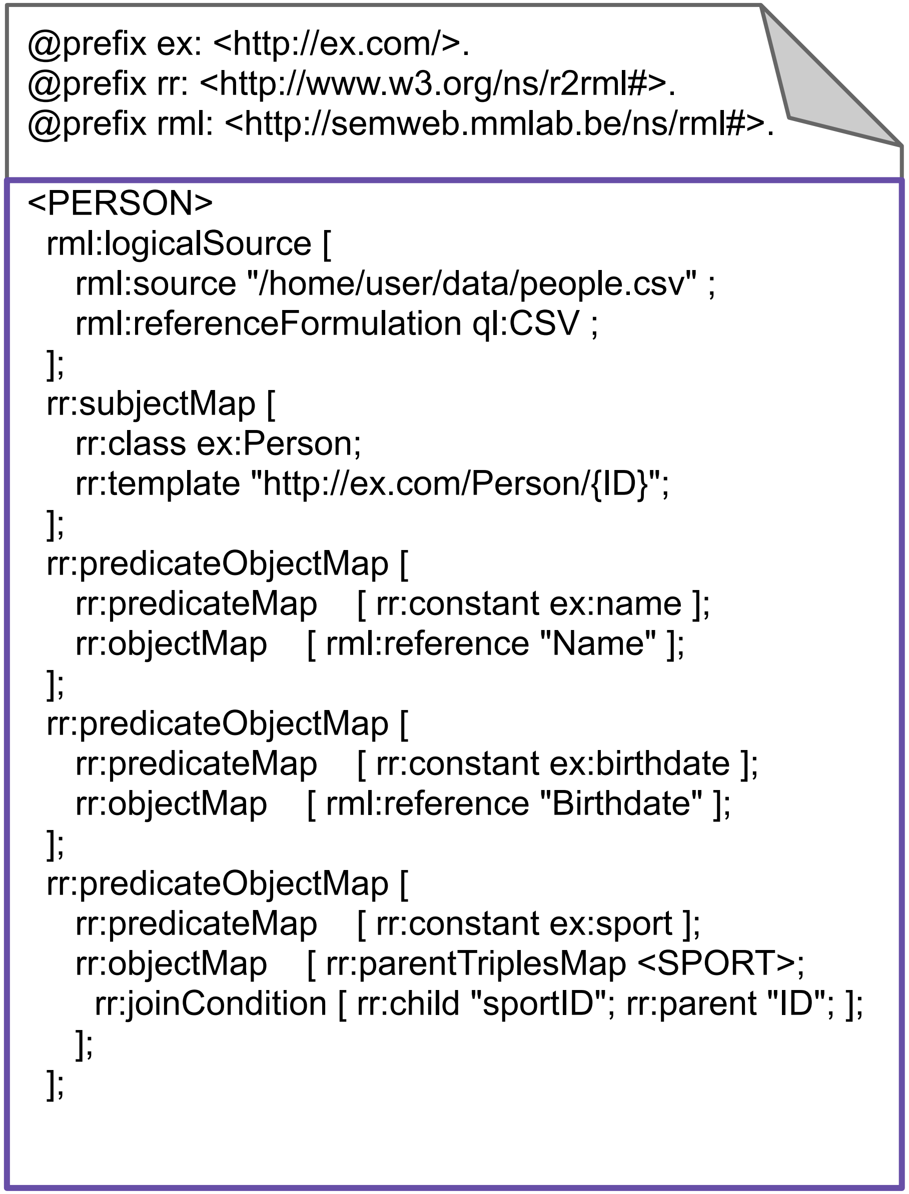
- Prefixes
- Triple Map
  - Source
  - Subject
  - Predicate-Object
  - Reference Object



```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
  rml:logicalSource [
    rml:source "/home/user/data/people.csv" ;
    rml:referenceFormulation ql:CSV ;
  ];
  rr:subjectMap [
    rr:class ex:Person;
    rr:template "http://ex.com/Person/{ID}";
  ];
  rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:name ];
    rr:objectMap    [ rml:reference "Name" ];
  ];
  rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:birthdate ];
    rr:objectMap    [ rml:reference "Birthdate" ];
  ];
  rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:sport ];
    rr:objectMap    [ rr:parentTriplesMap <SPORT>;
      rr:joinCondition [ rr:child "sportID"; rr:parent "ID"; ];
    ];
  ];
```

Mapping structure (RML)

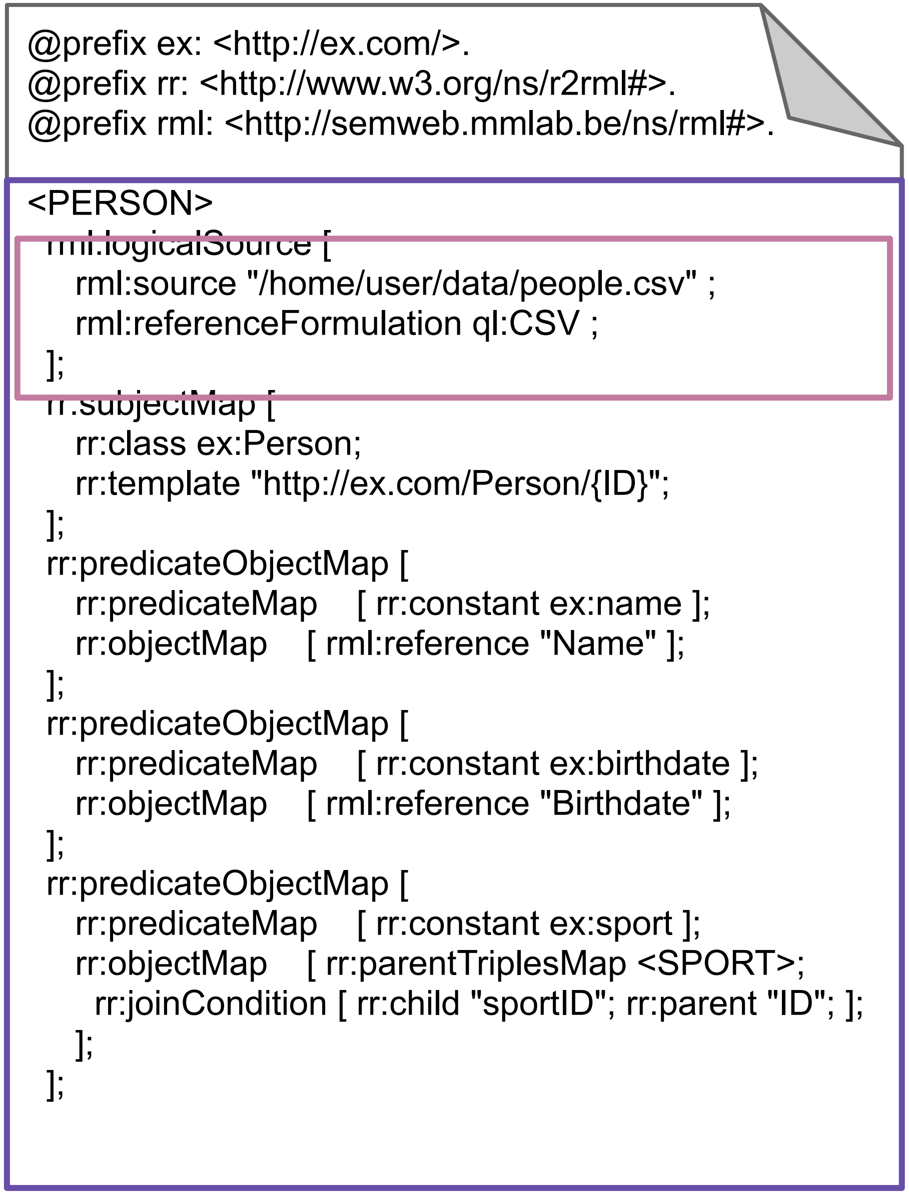
- Prefixes
- Triple Map
  - Source
  - Subject
  - Predicate-Object
  - Reference Object



```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
  rml:logicalSource [
     rml:source "/home/user/data/people.csv" ;
     rml:referenceFormulation ql:CSV ;
  ];
  rr:subjectMap [
     rr:class ex:Person;
     rr:template "http://ex.com/Person/{ID}";
  ];
  rr:predicateObjectMap [
     rr:predicateMap    [ rr:constant ex:name ];
     rr:objectMap    [ rml:reference "Name" ];
  ];
  rr:predicateObjectMap [
     rr:predicateMap    [ rr:constant ex:birthdate ];
     rr:objectMap    [ rml:reference "Birthdate" ];
  ];
  rr:predicateObjectMap [
     rr:predicateMap    [ rr:constant ex:sport ];
     rr:objectMap    [ rr:parentTriplesMap <SPORT>;
       rr:joinCondition [ rr:child "sportID"; rr:parent "ID"; ];
     ];
  ];
```

Mapping structure (RML)

- Prefixes
- Triple Map
    - Source
    - Subject
    - Predicate-Object
    - Reference Object



```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
 rml:logicalSource [
   rml:source "/home/user/data/people.csv" ;
   rml:referenceFormulation ql:CSV ;
 ];
 rr:subjectMap [
   rr:class ex:Person;
   rr:template "http://ex.com/Person/{ID}";
 ];
 rr:predicateObjectMap [
   rr:predicateMap    [ rr:constant ex:name ];
   rr:objectMap    [ rml:reference "Name" ];
 ];
 rr:predicateObjectMap [
   rr:predicateMap    [ rr:constant ex:birthdate ];
   rr:objectMap    [ rml:reference "Birthdate" ];
 ];
 rr:predicateObjectMap [
   rr:predicateMap    [ rr:constant ex:sport ];
   rr:objectMap    [ rr:parentTriplesMap <SPORT>;
     rr:joinCondition [ rr:child "sportID"; rr:parent "ID"; ];
   ];
 ];
```

Mapping structure (RML)

- Prefixes
- Triple Map
    - Source
    - Subject
    - Predicate-Object
    - Reference Object

```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
  rml:logicalSource [
     rml:source "/home/user/data/people.csv" ;
     rml:referenceFormulation ql:CSV ;
  ];
  rr:subjectMap [
     rr:class ex:Person;
     rr:template "http://ex.com/Person/{ID}";
  ];
  rr:predicateObjectMap [
     rr:predicateMap    [ rr:constant ex:name ];
     rr:objectMap    [ rml:reference "Name" ];
  ];
  rr:predicateObjectMap [
     rr:predicateMap    [ rr:constant ex:birthdate ];
     rr:objectMap    [ rml:reference "Birthdate" ];
  ];
  rr:predicateObjectMap [
     rr:predicateMap    [ rr:constant ex:sport ];
     rr:objectMap    [ rr:parentTriplesMap <SPORT>;
       rr:joinCondition [ rr:child "sportID"; rr:parent "ID"; ];
     ];
  ];
```

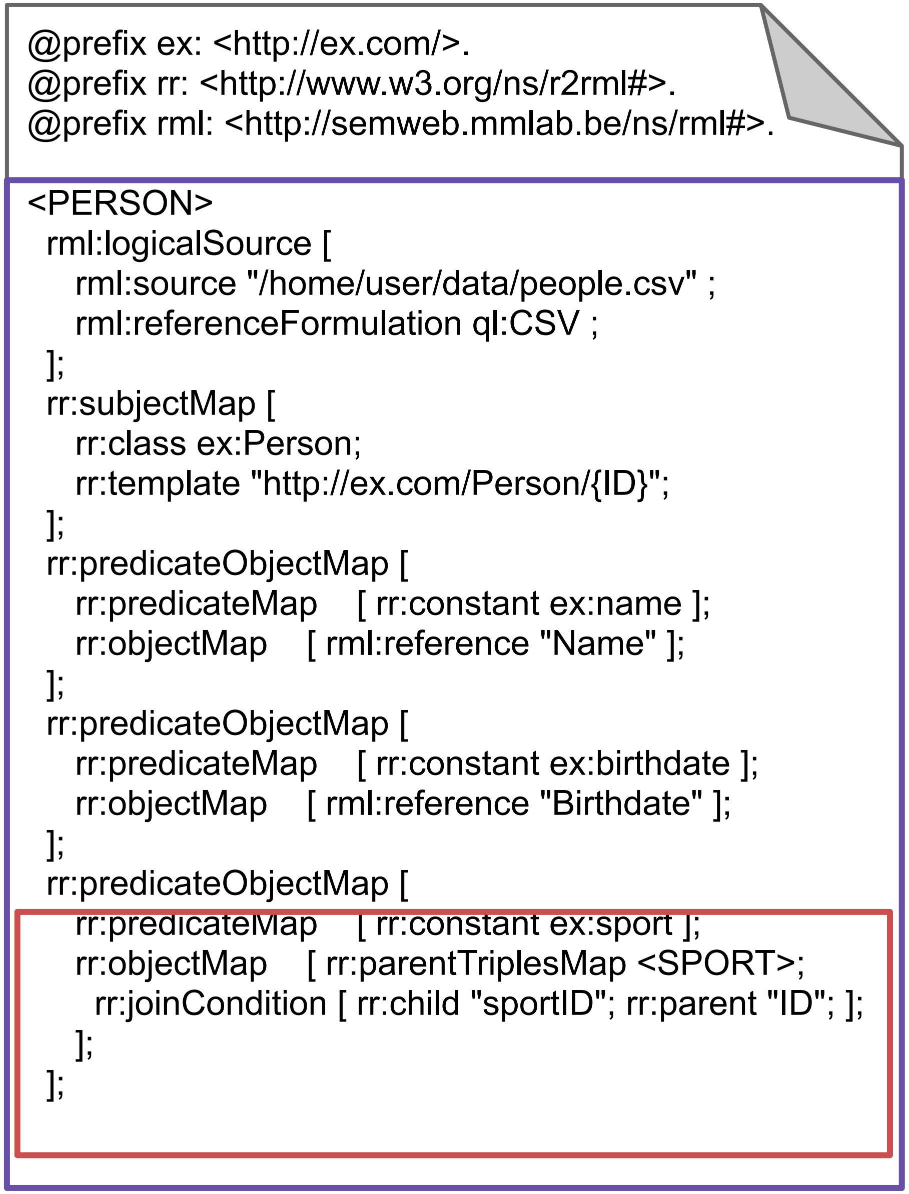
Mapping structure (RML)

- Prefixes
- Triple Map
  - Source
  - Subject
  - Predicate-Object
  - Reference Object

```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
  rml:logicalSource [
    rml:source "/home/user/data/people.csv" ;
    rml:referenceFormulation ql:CSV ;
  ];
  rr:subjectMap [
    rr:class ex:Person;
    rr:template "http://ex.com/Person/{ID}";
  ];
  rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:name ];
    rr:objectMap    [ rml:reference "Name" ];
  ];
  rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:birthdate ];
    rr:objectMap    [ rml:reference "Birthdate" ];
  ];
  rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:sport ];
    rr:objectMap    [ rr:parentTriplesMap <SPORT>;
      rr:joinCondition [ rr:child "sportID"; rr:parent "ID"; ];
    ];
  ];
```

Mapping structure (RML)

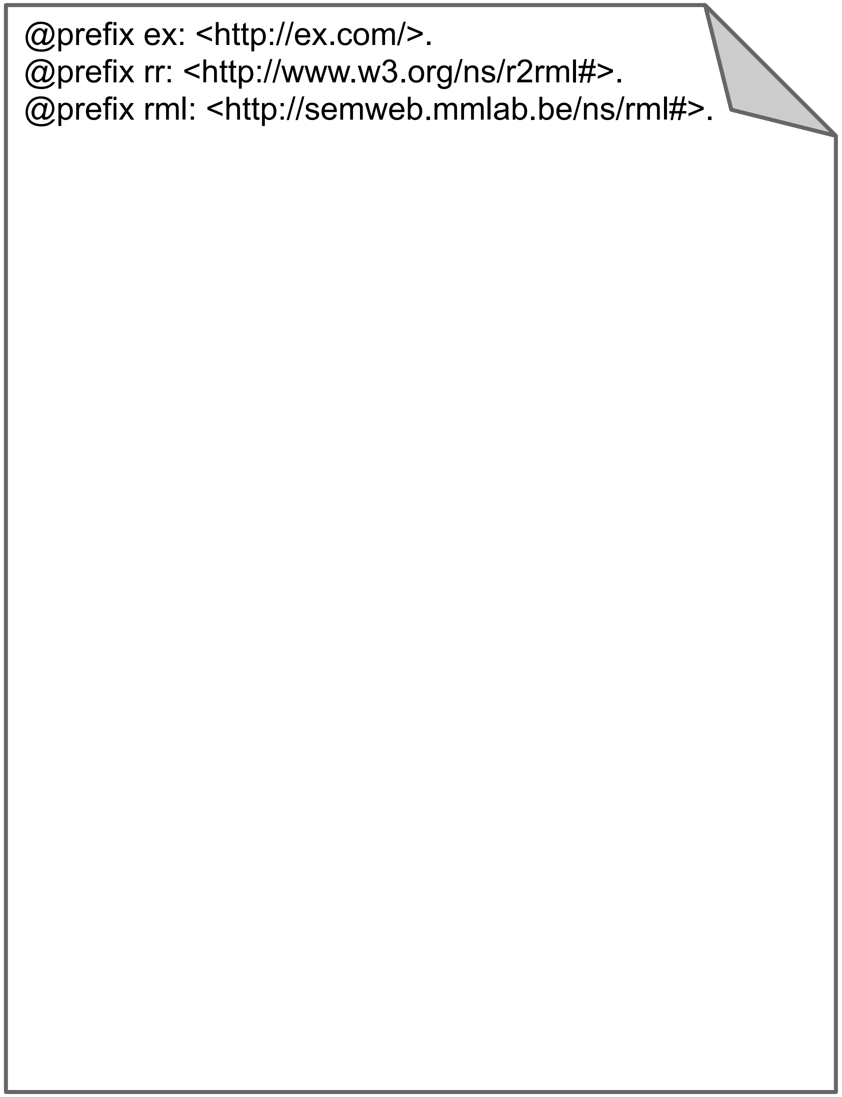
- Prefixes
- Triple Map
  - Source
  - Subject
  - Predicate-Object
  - Reference Object



```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
  rml:logicalSource [
    rml:source "/home/user/data/people.csv" ;
    rml:referenceFormulation ql:CSV ;
  ];
  rr:subjectMap [
    rr:class ex:Person;
    rr:template "http://ex.com/Person/{ID}";
  ];
  rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:name ];
    rr:objectMap    [ rml:reference "Name" ];
  ];
  rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:birthdate ];
    rr:objectMap    [ rml:reference "Birthdate" ];
  ];
  rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:sport ];
    rr:objectMap    [ rr:parentTriplesMap <SPORT>;
      rr:joinCondition [ rr:child "sportID"; rr:parent "ID"; ];
    ];
  ];
```

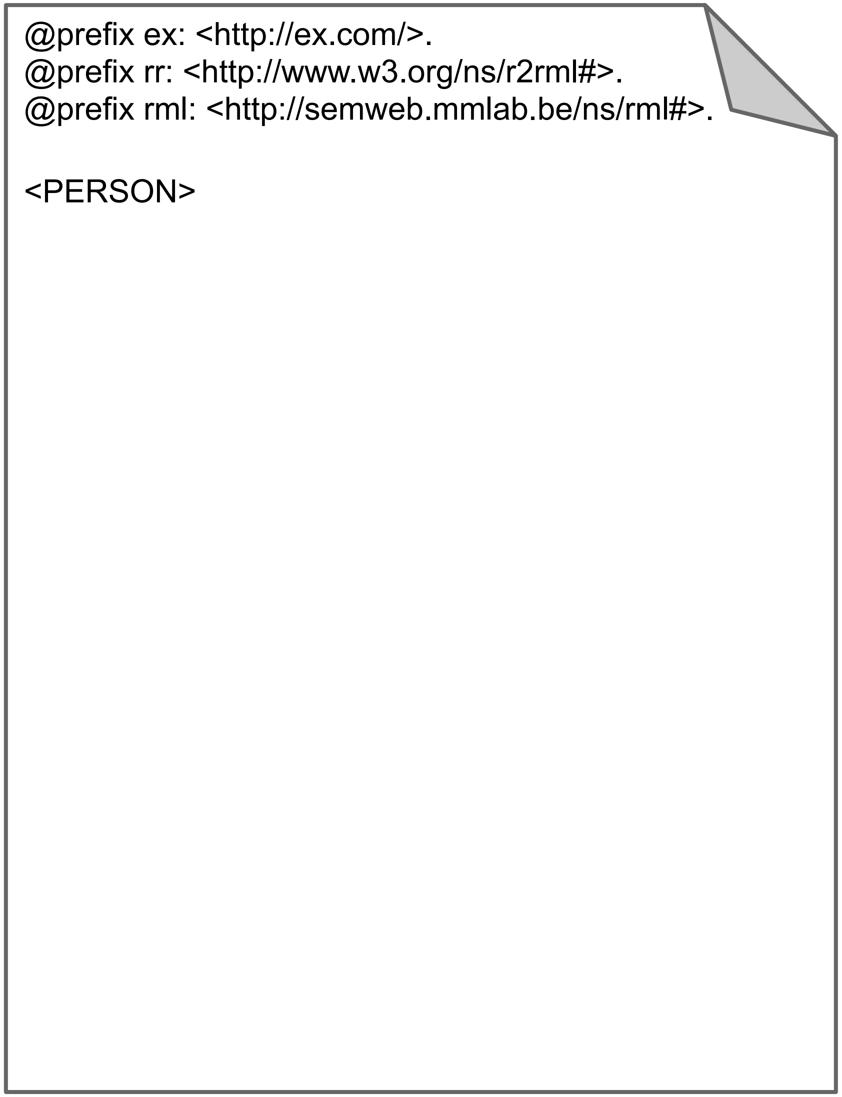
Mapping structure (RML)

Mapping structure (RML)



```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.
```

Mapping structure (RML)



```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
```

Mapping structure (RML)

people.csv

ID	Name	Birthdate	SportID
1	Emily Scarrat	19900208	2
2	Jonah Lomu	19751118	2

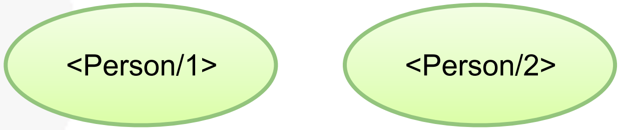
```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
 rml:logicalSource [
    rml:source "/home/user/data/people.csv" ;
    rml:referenceFormulation ql:CSV ;
 ];
```

Mapping structure (RML)

people.csv

ID	Name	Birthdate	SportID
1	Emily Scarrat	19900208	2
2	Jonah Lomu	19751118	2

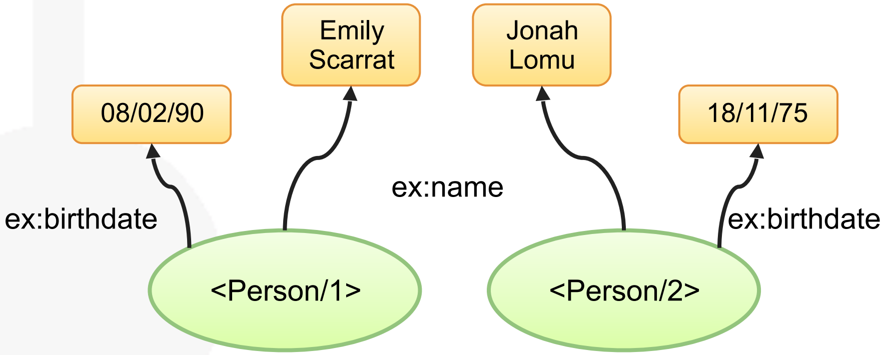
```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
 rml:logicalSource [
    rml:source "/home/user/data/people.csv" ;
    rml:referenceFormulation ql:CSV ;
 ];
 rr:subjectMap [
    rr:class ex:Person;
    rr:template "http://ex.com/Person/{ID}";
 ];
```



Mapping structure (RML)

people.csv

ID	Name	Birthdate	SportID
1	Emily Scarrat	19900208	2
2	Jonah Lomu	19751118	2

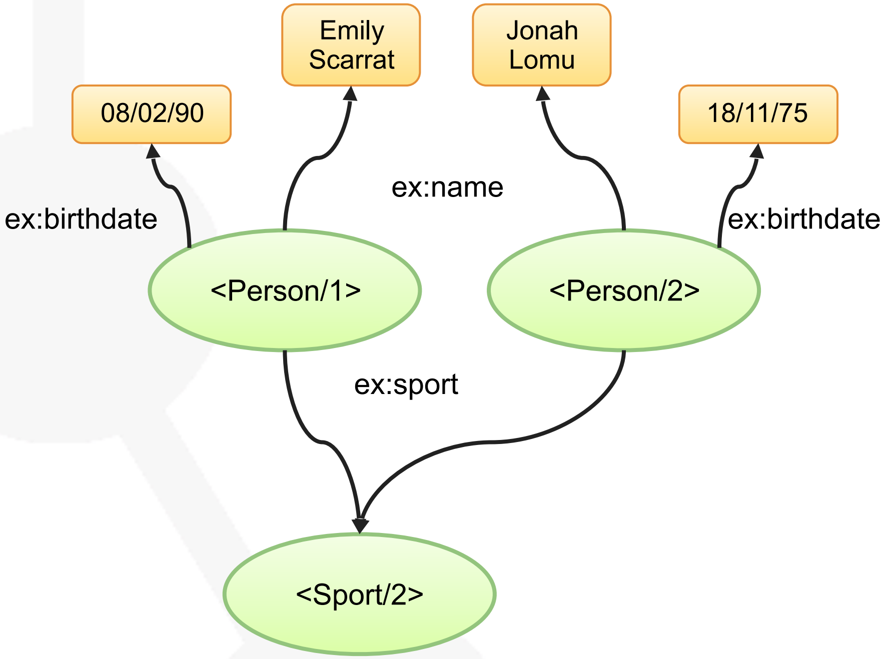


```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
 rml:logicalSource [
   rml:source "/home/user/data/people.csv" ;
   rml:referenceFormulation ql:CSV ;
 ];
 rr:subjectMap [
   rr:class ex:Person;
   rr:template "http://ex.com/Person/{ID}";
 ];
 rr:predicateObjectMap [
   rr:predicateMap    [ rr:constant ex:name ];
   rr:objectMap    [ rml:reference "Name" ];
 ];
 rr:predicateObjectMap [
   rr:predicateMap    [ rr:constant ex:birthdate ];
   rr:objectMap    [ rml:reference "Birthdate" ];
 ];
```

Mapping structure (RML)

people.csv

ID	Name	Birthdate	SportID
1	Emily Scarrat	19900208	2
2	Jonah Lomu	19751118	2



```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<PERSON>
 rml:logicalSource [
    rml:source "/home/user/data/people.csv" ;
    rml:referenceFormulation ql:CSV ;
 ];
 rr:subjectMap [
    rr:class ex:Person;
    rr:template "http://ex.com/Person/{ID}";
 ];
 rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:name ];
    rr:objectMap    [ rml:reference "Name" ];
 ];
 rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:birthdate ];
    rr:objectMap    [ rml:reference "Birthdate" ];
 ];
 rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:sport ];
    rr:objectMap    [ rr:parentTriplesMap <SPORT>;
      rr:joinCondition [ rr:child "sportID"; rr:parent "ID"; ];
    ];
 ];
```

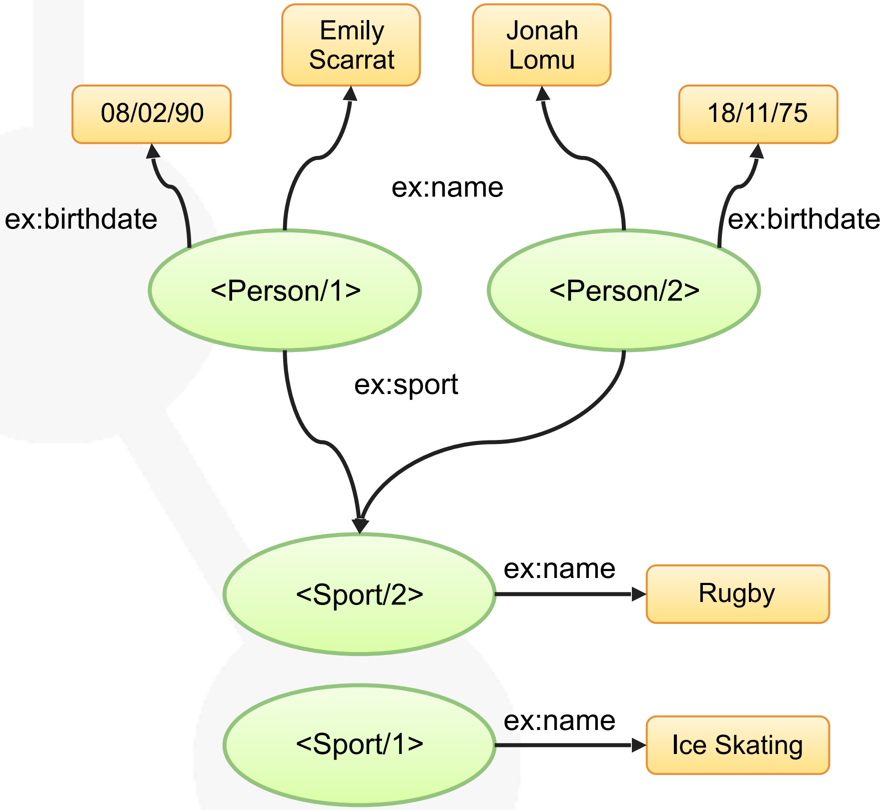
Mapping structure (RML)

people.csv

ID	Name	Birthdate	SportID
1	Emily Scarrat	19900208	2
2	Jonah Lomu	19751118	2

sports.csv

ID	Sport
1	Ice Skating
2	Rugby



```
@prefix ex: <http://ex.com/>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix rml: <http://semweb.mmlab.be/ns/rml#>.

<P
 rr

 ];
 rr

 ];
 rr

 ];
 rr
    rr:objectMap    [ rml:reference "Birthdate" ];
 ];
 rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:sport ];
    rr:objectMap    [ rr:parentTriplesMap <SPORT>;
      rr:joinCondition [ rr:child "sportID"; rr:parent "ID"; ];
    ];
 ];
```

```
<SPORT>
  rml:logicalSource [
    rml:source "/home/user/data/sports.csv" ;
    rml:referenceFormulation ql:CSV ;
  ];
  rr:subjectMap [
    rr:class ex:Sport;
    rr:template "http://ex.com/Sport/{ID}";
  ];
  rr:predicateObjectMap [
    rr:predicateMap    [ rr:constant ex:name ];
    rr:objectMap    [ rml:reference "sport" ];
  ];
```

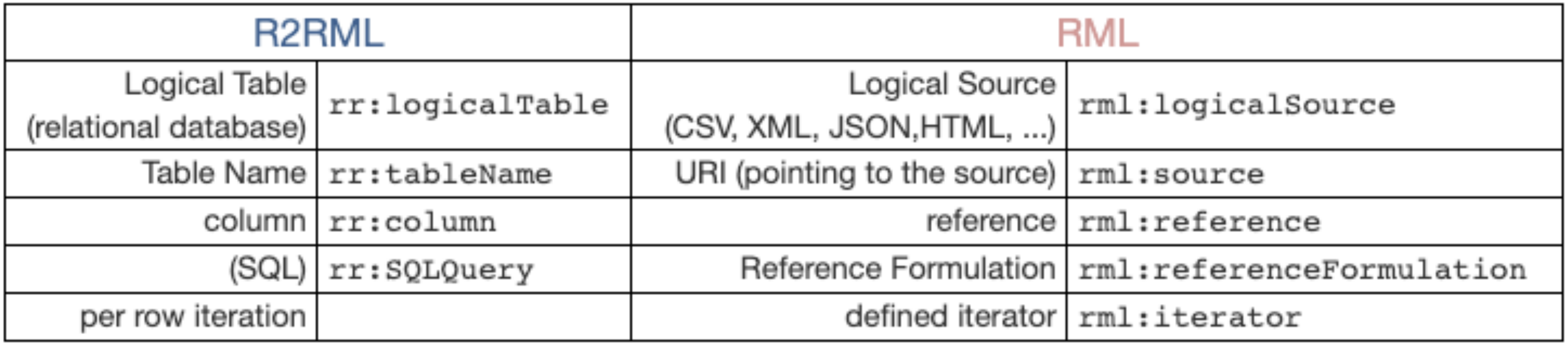
R2RML

- Input
  - RDB
  - Target ontologies
  - Mappings between the RDB and target ontologies
- Output
  - RDF graph
  - SPARQL result-set (query translation techniques)
- Engines: https://www.w3.org/TR/rdb2rdf-implementations/

https://www.w3.org/TR/r2rml/

RML

- Input
  - Data sources in any format
  - Target ontologies
  - Mappings between the data sources and target ontologies in RML
- Output
  - RDF graph
- Engines: http://rml.io/implementation-report/



R2RML		RML	
Logical Table (relational database)	`rr:logicalTable`	Logical Source (CSV, XML, JSON,HTML, ...)	`rml:logicalSource`
Table Name	`rr:tableName`	URI (pointing to the source)	`rml:source`
column	`rr:column`	reference	`rml:reference`
(SQL)	`rr:SQLQuery`	Reference Formulation	`rml:referenceFormulation`
per row iteration		defined iterator	`rml:iterator`

Info: http://rml.io/

Examples: https://bit.ly/36Be8r7

User-friendly - YARRRML

YARRRML

http://rml.io/yarrrml/matey/

People			
ID	**Name**	**Birthdate**	**SportID**
1	Emily Scarrat	19900208	2
2	Jonah Lomu	19751118	2

Sports	
ID	**Sport**
1	Ice Skating
2	Rugby

Target user: Scientist with technical background in data management

Advantage: easy to read and to create; translated to RML

```
prefixes:
  ex: "http://ex.com/"

mappings:
  PERSON:
    sources:
      - [/home/user/data/people.csv~CSV]
    s: http://ex.com/Person/$(ID)
    po:
      - [a, ex:Person]
      - [ex:name, $(Name)]
      - [ex:birthdate, $(Birthdate)]
      - p: ex:sport
        o:
          - mapping: SPORT
            condition:
              function: equal
              parameters:
                - [str1, $(SportID)]
                - [str2, $(ID)]
  SPORT:
    sources:
      - [/home/user/data/people.csv~CSV]
    s: http://ex.com/Sport/$(ID)
    po:
      - [a, ex:Sport]
      - [ex:name, $(Sport)]
```

Domain Specific Language (DSL)

https://zazuko.com/blog/rdf-and-dsl-a-perfect-match/

People			
ID	**Name**	**Birthdate**	**SportID**
1	Emily Scarrat	19900208	2
2	Jonah Lomu	19751118	2

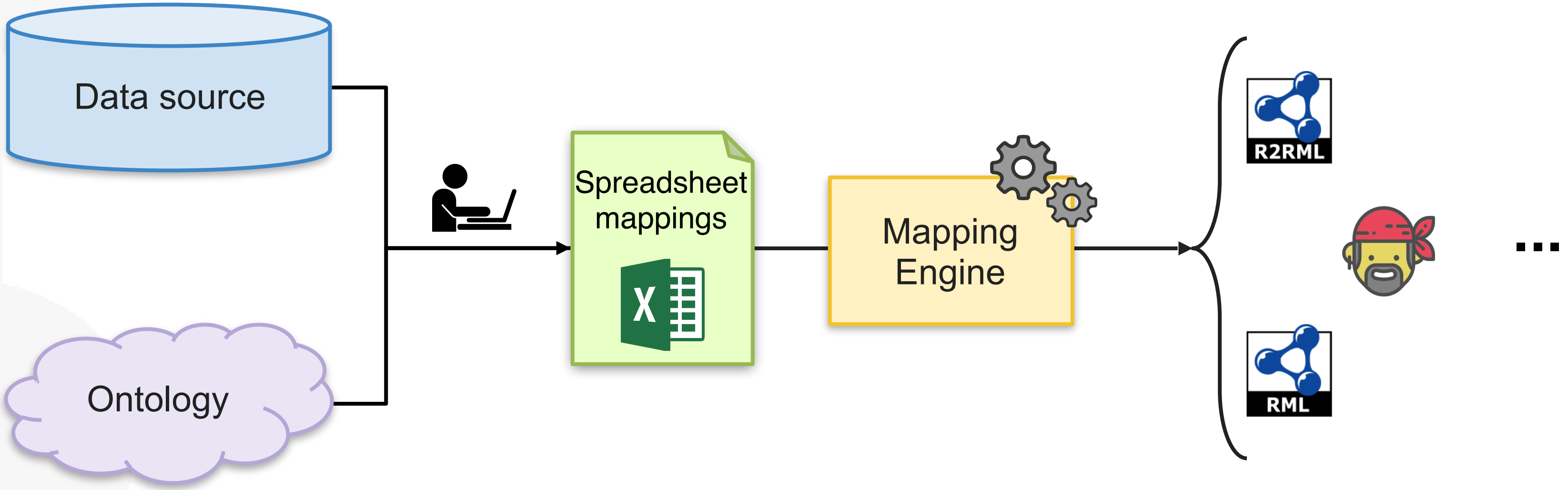
Sports	
ID	**Sport**
1	Ice Skating
2	Rugby

```
map PERSON from People {
  subject template http://ex.com/Person/{0} with ID;
  types ex.Person;

  properties
    ex.name from Name
    ex.birthdate from Birthdate

    ex.sport link SPORT with SportID
}

map SPORT from Sports {
  subject template http://ex.com/Sport/{0} with ID;
  types ex.Sport;

  properties
    ex.name from Sport
}
```

Target user: knowledge engineering with programming skills

Advantage: always generates syntactically valid mappings

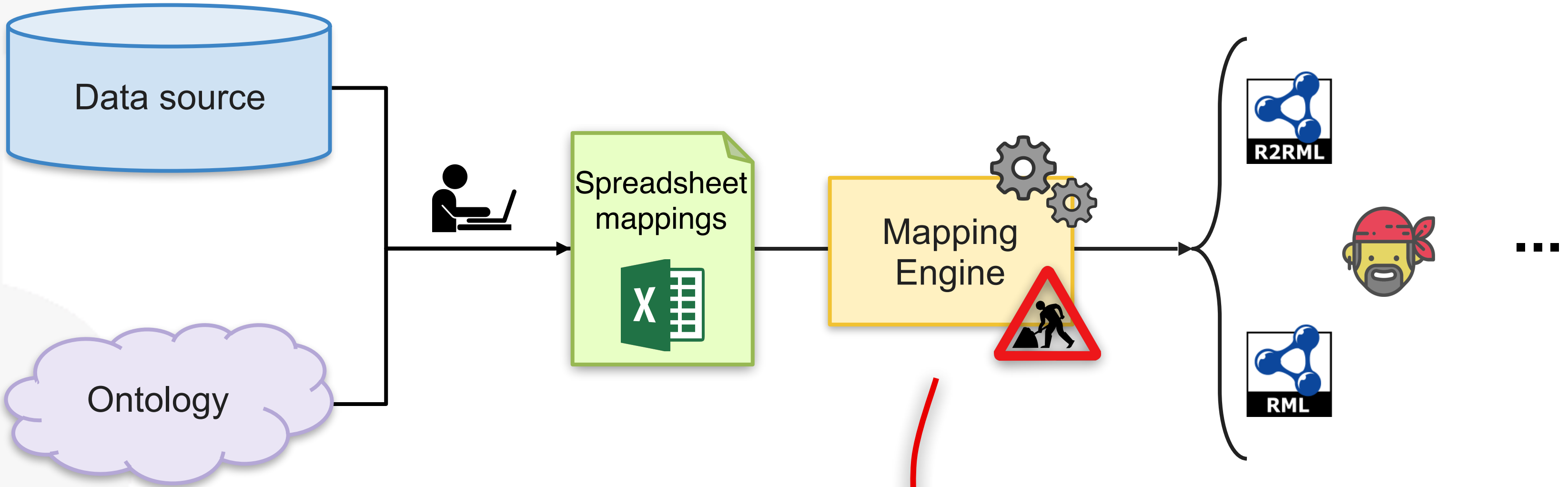
Spreadsheets as mappings



Ana Iglesias-Molina, David Chaves-Fraga, Freddy Priyatna, Óscar Corcho. "Towards the Definition of a Language-Independent Mapping Template for Knowledge Graph Creation". Third International Workshop on Capturing Scientific Knowledge (Sciknow 2019) co-located with the 10th International Conference on Knowledge Capture, 2019

https://github.com/oeg-upm/Excel-mapping-translator
https://github.com/w0xter/ABM2ALL

Spreadsheets as mappings



Looking for names!

Ana Iglesias-Molina, David Chaves-Fraga, Freddy Priyatna, Óscar Corcho. “Towards the Definition of a Language-Independent Mapping Template for Knowledge Graph Creation”. Third International Workshop on Capturing Scientific Knowledge (Sciknow 2019) co-located with the 10th International Conference on Knowledge Capture, 2019

https://github.com/oeg-upm/Excel-mapping-translator
https://github.com/w0xter/ABM2ALL

User-friendly - Spreadsheets

Prefix sheet

Prefix	URI
ex	http://ex.com/

Subject sheet

ID	Class	URI
PERSON	ex:Person	http://ex.com/Person/{ID}
SPORT	ex:Sport	http://ex.com/Sport/{ID}

Source sheet

ID	Feature	Value
PERSON	source	data/people.csv
PERSON	format	CSV
SPORT	source	data/sports.csv
SPORT	format	CSV

Predicate-Object sheet

Predicate	Object	DataType	ReferenceID	InnerRef	OurterRef	ID
ex:name	{name}	string				PERSON
ex:birthdate	{birthdate}	date				PERSON
ex:sport			SPORT	SportID	ID	PERSON
ex:name	{sport}	string				SPORT

User-friendly - Spreadsheets

- Following idea of mapping translation*
- Structured in sheets
- **Objective**: language-independent
- **Target user**: Non mapping experts
- **Advantages**: Improves rule visualization, enables using functions of spreadsheets

Oscar Corcho, Freddy Priyatna, and David Chaves-Fraga. 2019. Towards a New Generation of Ontology Based Data Access. *Semantic Web Journal* (2019).

Future Work

- Mapping translation:
  - From independent rules to specific language (ISWC 2020)
  - Web application to create mappings (ISWC 2020 Demo)
- Engines:
  - Web applications of morph tools (Demos)
  - Morph-CSV: efficient management tabular data exploiting mapping rules (ESWC 2020 and ISWC 2020)
- OTHER IDEAS?

Looking for names!

Help us decide:

1. SpreadPing (Spreadsheet mapPing)
2. MapSheet
3. Sheet2Map
4. S2M (Sheet to Mapping)
5. morph-translation
6. ABM2ALL (ABstract Mapping to ALL)
7. SpreadMap
8. Mapeator
9. Map-ATOMATIC
10. More ideas?

How to Develop Mapping Rules for Knowledge Graph Creation

Ana Iglesias-Molina & David Chaves-Fraga

✉ dataintegration@delicias.dia.fi.upm.es

@oeg-upm

📅 07/11/2019

📍 OEG Talk