



# Towards the Definition of a Language-Independent Mapping Template for Knowledge Graph Creation

**Oscar Corcho, Ontology Engineering Group**  
**Universidad Politécnica de Madrid, Spain**  
Ana Iglesias-Molina, David Chaves-Fraga and Freddy  
Priyatna, OEG - UPM

✉ [ocorcho@fi.upm.es](mailto:ocorcho@fi.upm.es)

🐦 @ocorcho

📅 19/11/2019

📍 SciKnow 2019



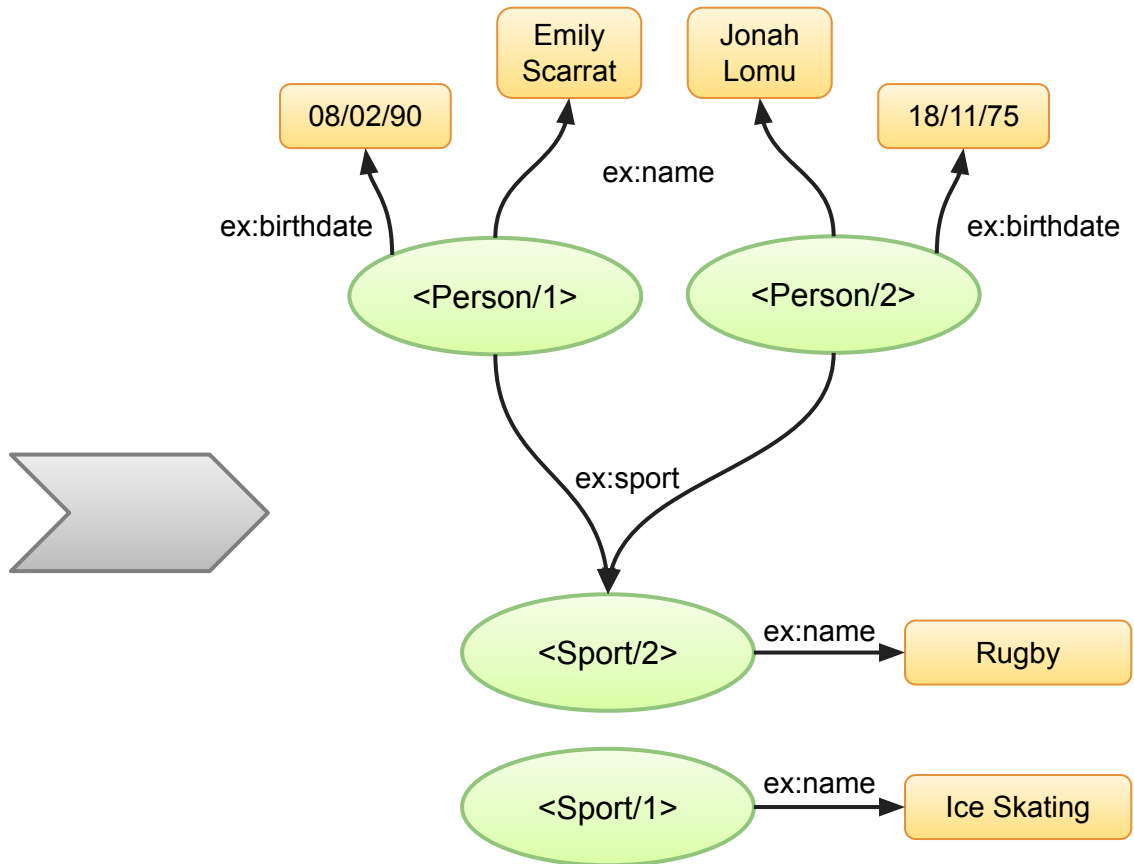
# Creating Knowledge Graphs?

## 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



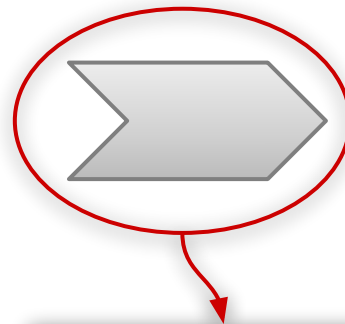
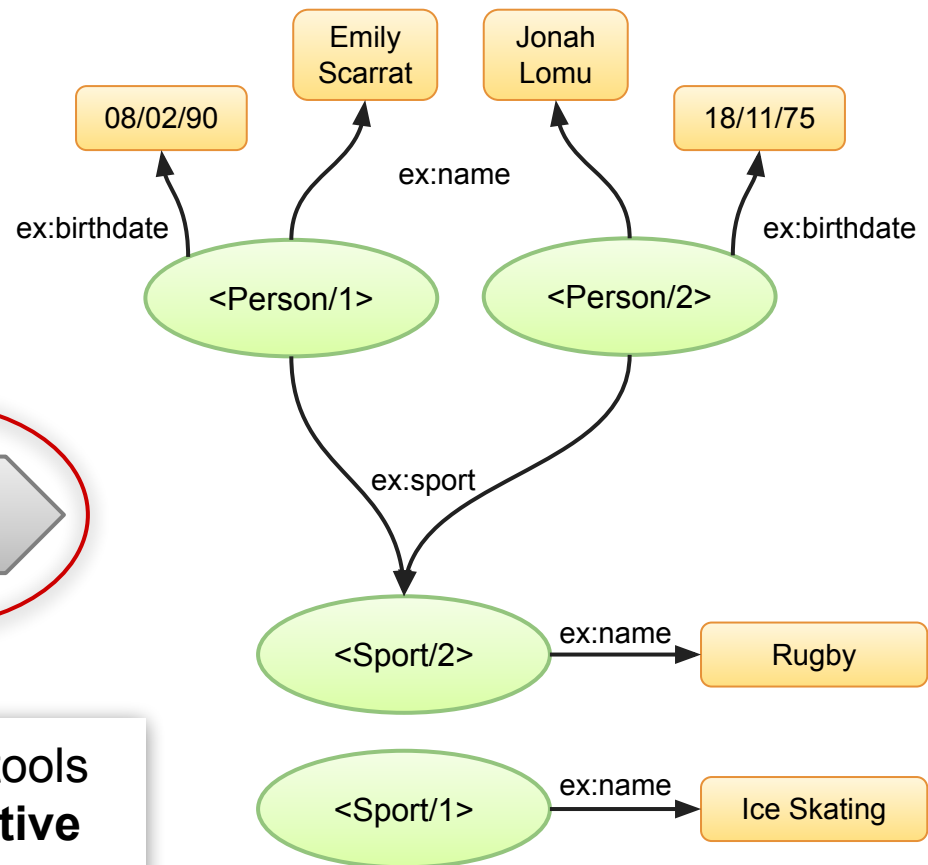
# Creating Knowledge Graphs?

## 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

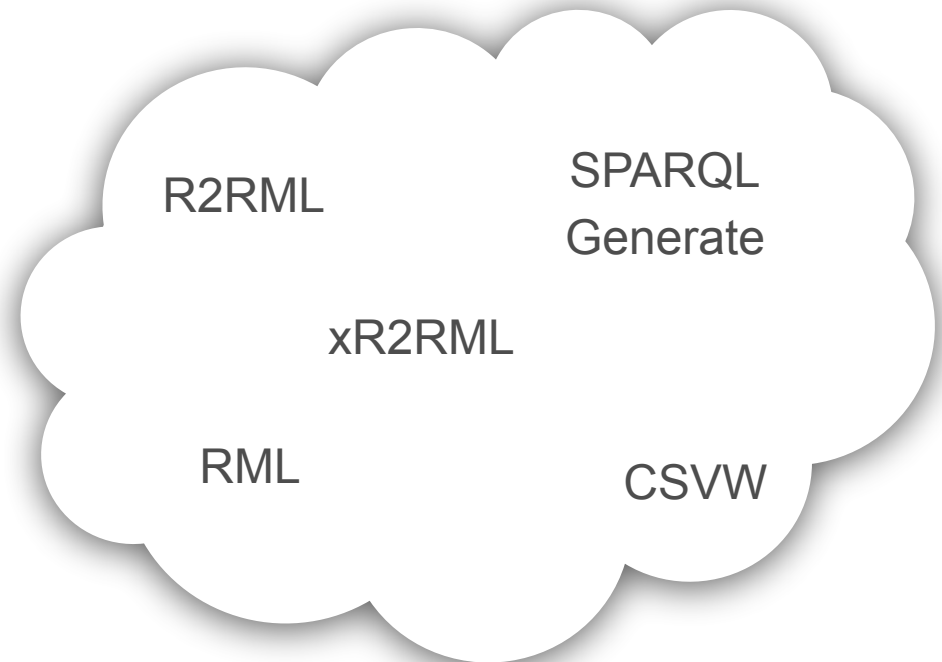
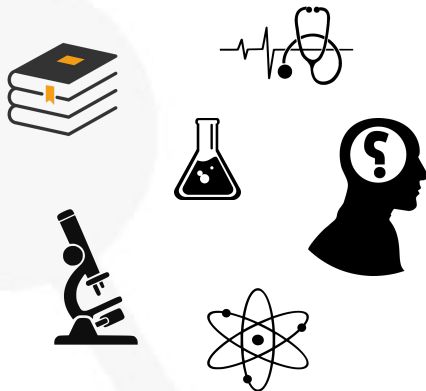


- Ad-hoc tools
- **Declarative mappings**

# Declarative Mappings

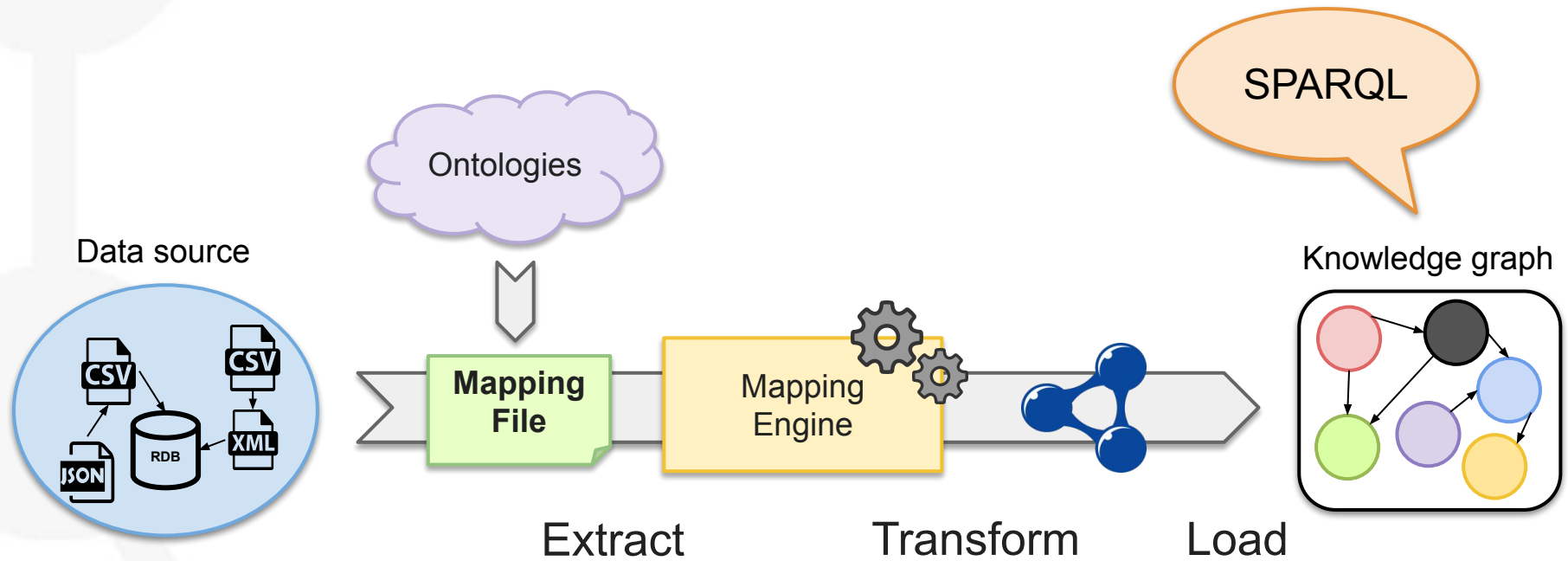
- Establish relationships between ontology and data
- Several languages:

Scientific  
Knowledge



**Easy way to create and choose among them?**

# Knowledge Graph Creation



# Mapping structure (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" ];
  ];

```

## Triples Map



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

→	ex:Person/1	a	ex:Person .
→	ex:Person/1	ex:name	"Emily Scarrat" .
→	ex:Person/1	ex:birthdate	"08/02/1990" .
→	ex:Person/2	a	ex:Person .
→	ex:Person/2	ex:name	"Jonah Lomu" .
→	ex:Person/2	ex:birthdate	"18/11/1975" .

# Mapping structure (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 "sport_id"; rr:parent "id" ];
    ];
  ];
];
```

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

```
<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/{sport}";
  ];
  rr:predicateObjectMap [
    rr:predicateMap [ rr:constant ex:name ];
    rr:objectMap [ rml:reference "sport" ];
  ];
];
```

**Join condition**

# Mapping structure (RML)

```
<PERSON>
  rml:logicalS
  rml:sc
  rml:r
];
rr:sub
rr:cl
rr:te
];
rr:pre
rr:pr
rr:ol
];
rr:pre
rr:pr
rr:ol
];
rr:pre
rr:pr
rr:ol
rr:jo
];
];
```

people.csv			
ID	Name	Birthdate	SportID

sports.csv	
ID	Sport
1	Ice Skating
	Rugby

```
ex:Person/1      a      ex:Person .
ex:Person/1      ex:name  "Emily Scarrat" .
ex:Person/1      ex:birthdate "08/02/1990" .
ex:Person/1      ex:sport  ex:Sport/2 .
ex:Person/2      a      ex:Person .
ex:Person/2      ex:name  "Jonah Lomu" .
ex:Person/2      ex:birthdate "18/11/1975" .
ex:Person/2      ex:sport  ex:Sport/2 .

ex:Sport/1       a      ex:Sport .
ex:Sport/1       ex:name  "Ice Skating" .
ex:Sport/2       a      ex:Sport .
ex:Sport/2       ex:name  "Rugby" .
```

```
csv" ;
port}";
ame ];
t" ];
```



- Gathering and declaration of mapping rules in spreadsheets
- Each sheet contains an essential element describing the data:
  - Prefixes
  - Source data
  - Subject
  - Predicate-Object
  - Functions

- **Objective:** language-independent, no need to know a mapping language
- **Target user:** Non mapping experts
- **Advantages:** Improves rule visualization, enables using functions of spreadsheets, guides the writing

## Prefix sheet

Prefix	URI
ex	http://ex.com/
sql	http://w3.org/ns/sql#

## Subject sheet

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

## Source sheet

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

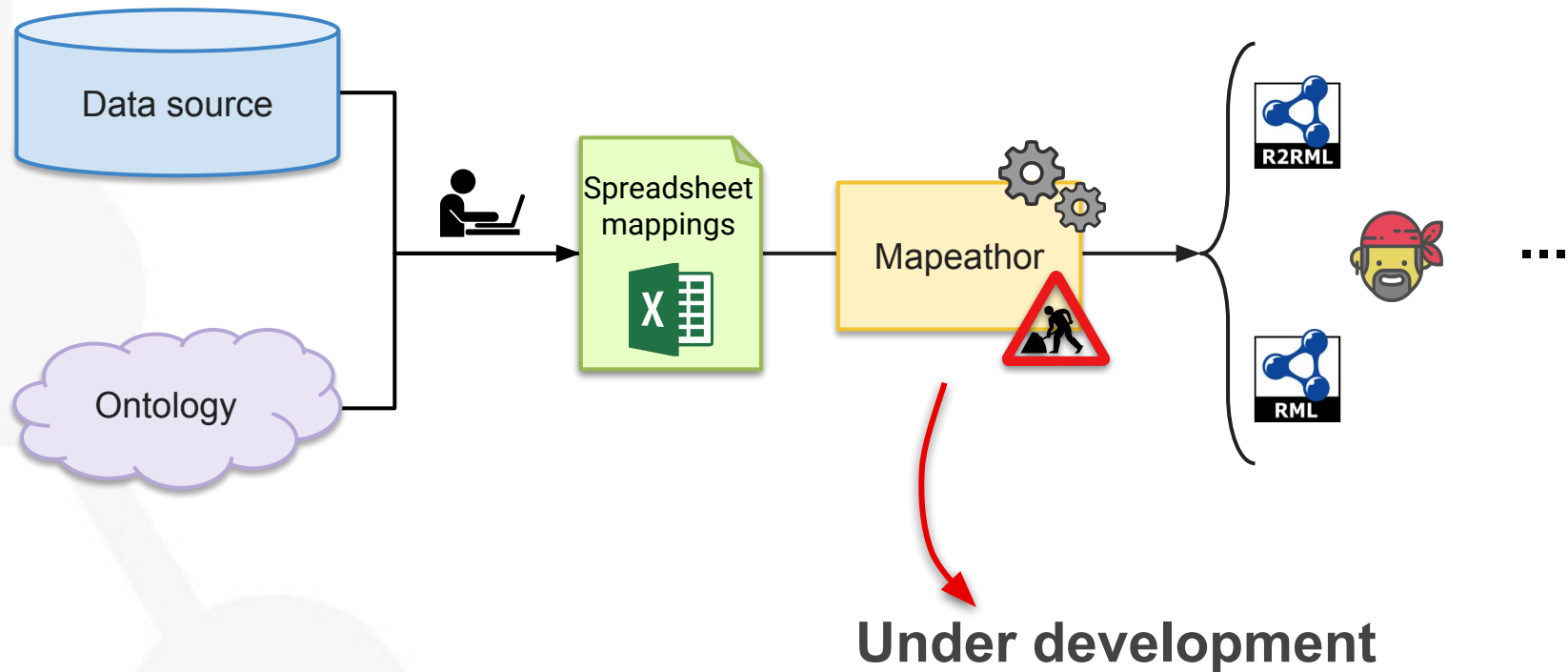
## Predicate-Object sheet

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
ex:code	{ID}	integer				SPORT
ex:comment	<Fun1>					SPORT

## Function sheet

FunctionID	Function	Params
<Fun1>	sql:concat	{ID}, <Fun2>
<Fun2>	sql:upper	{sport}

- Once the mapping rules are declared, they are translated into the most suitable language



<https://github.com/oeg-upm/Mapeathor>

## Conclusions:

- **Spreadsheet design** to capture scientific knowledge from experts improving KG creations
- **Mapeathor**: from spreadsheets to any mapping language
- Used in projects like H2020-Bimmer and Bio2RDF

## Future work:

- Improve spreadsheet design towards actual language-independence
- Develop Mapeathor with the spreadsheet
- Widen the language translation



# Towards the Definition of a Language-Independent Mapping Template for Knowledge Graph Creation

**Oscar Corcho, Ontology Engineering Group**  
**Universidad Politécnica de Madrid, Spain**  
Ana Iglesias-Molina, David Chaves-Fraga and Freddy Priyatna, OEG - UPM

✉ [ocorcho@fi.upm.es](mailto:ocorcho@fi.upm.es)

🐦 [@ocorcho](https://twitter.com/ocorcho)

📅 19/11/2019

📍 SciKnow 2019

