# Declarative Description of Knowledge Graphs Construction Automation: Status & Challenges

David Chaves-Fraga, UPM & KULeuven & Flanders Make
Anastasia Dimou, KULeuven & Flanders Make & Leuven.AI

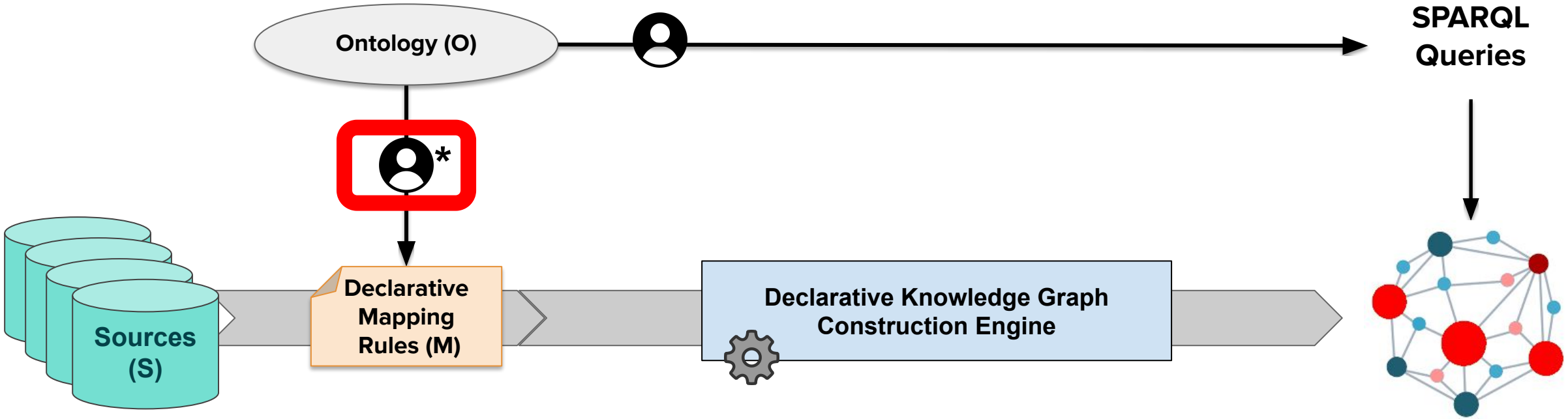
david.chaves@upm.es, anastasia.dimou@kuleuven.be
@dchavesf @natadimou

# First things first…

DO NOT QUOTE ME ON THIS PRESENTATION

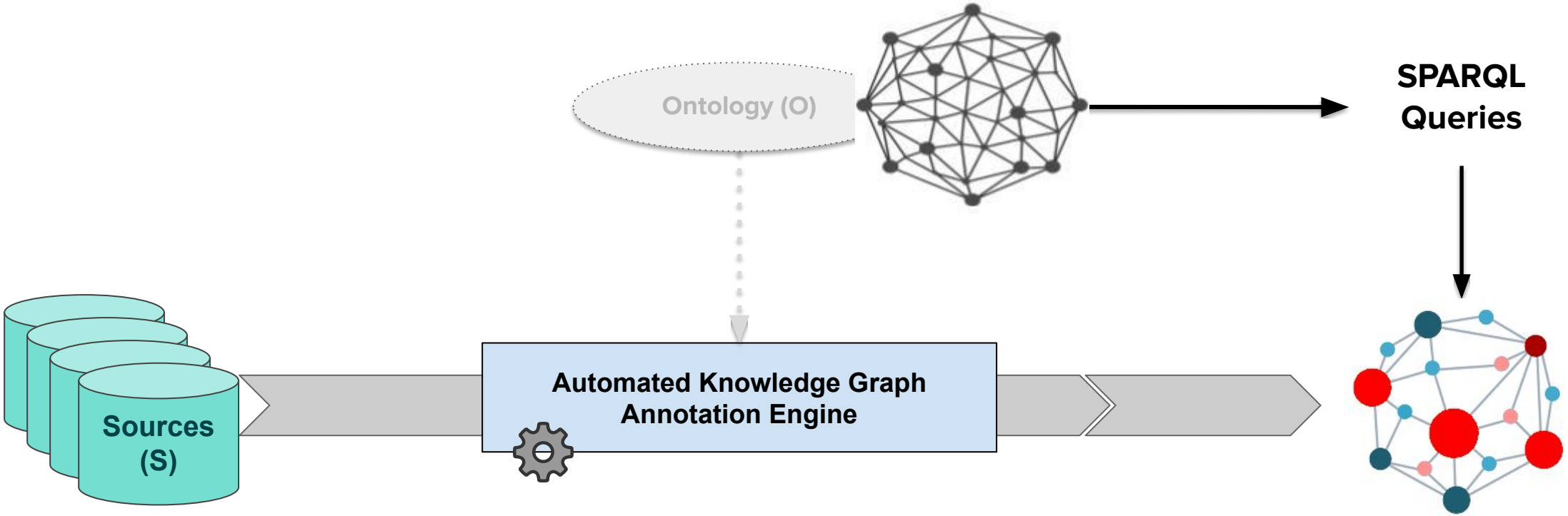
And interrupt me during the presentation to discuss

# KG Construction with Mapping Rules

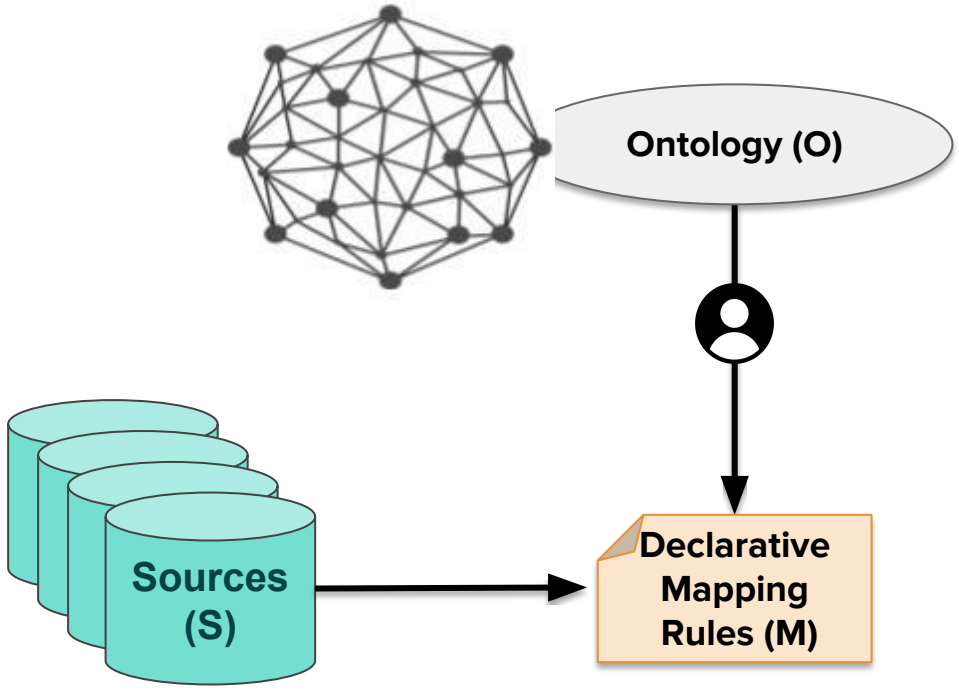



*** On average it takes 6 Person-Month to create the knowledge graph**

# Automation KG Construction



# KGC: Automation or/and rules?




**Rules**

- Declarative approach
- Understanding the domain
- Target KG
- Linear iteration
- Time consuming
- High quality KG
- Non-reproducible task
- Explainable

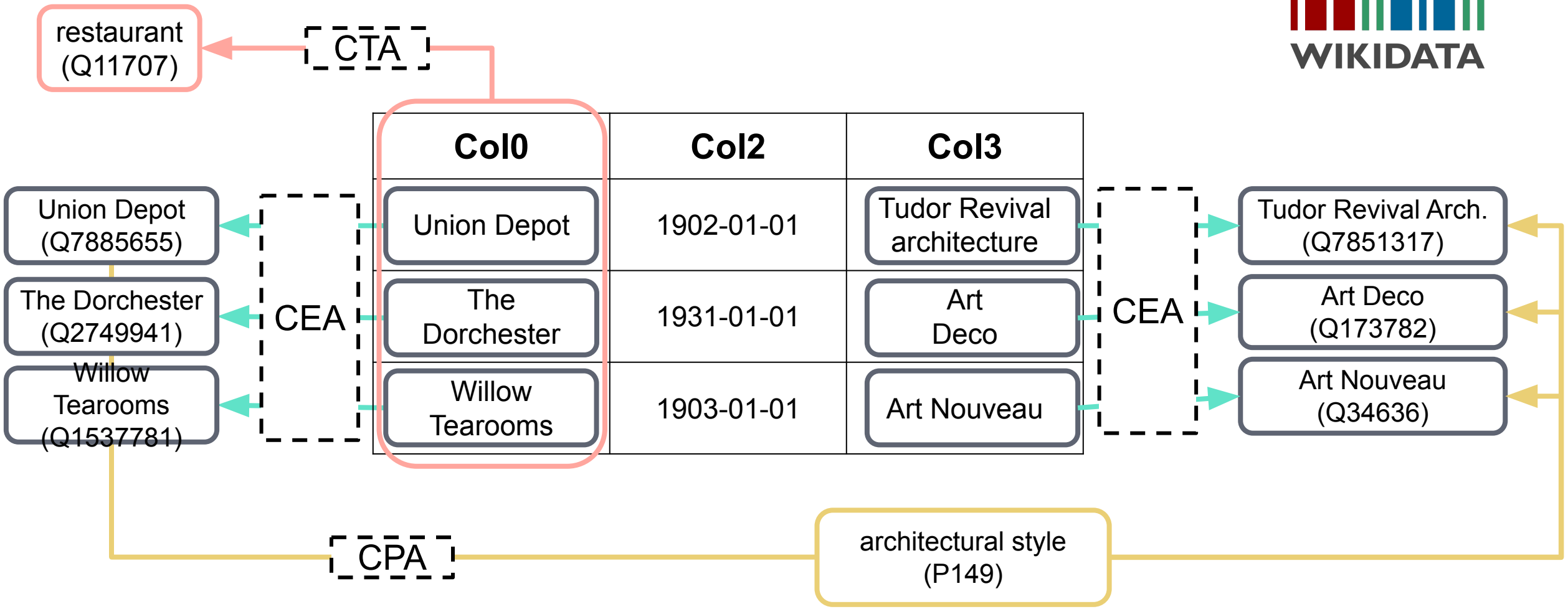
**Automation**

- No manual work
- No knowledge about the domain
- Target Annotation
- Multiple iterations
- Faster
- Quality can be compromised
- Reproducible tasks?
- Non-explainable

RQ1) Are hybrid approaches feasible to explain and optimize a knowledge graph construction process?

RQ2) Can we describe a knowledge graph construction automation process using declarative rules?

# Automation KGC: SemTab Challenge



# Analysis of current SemTab tools

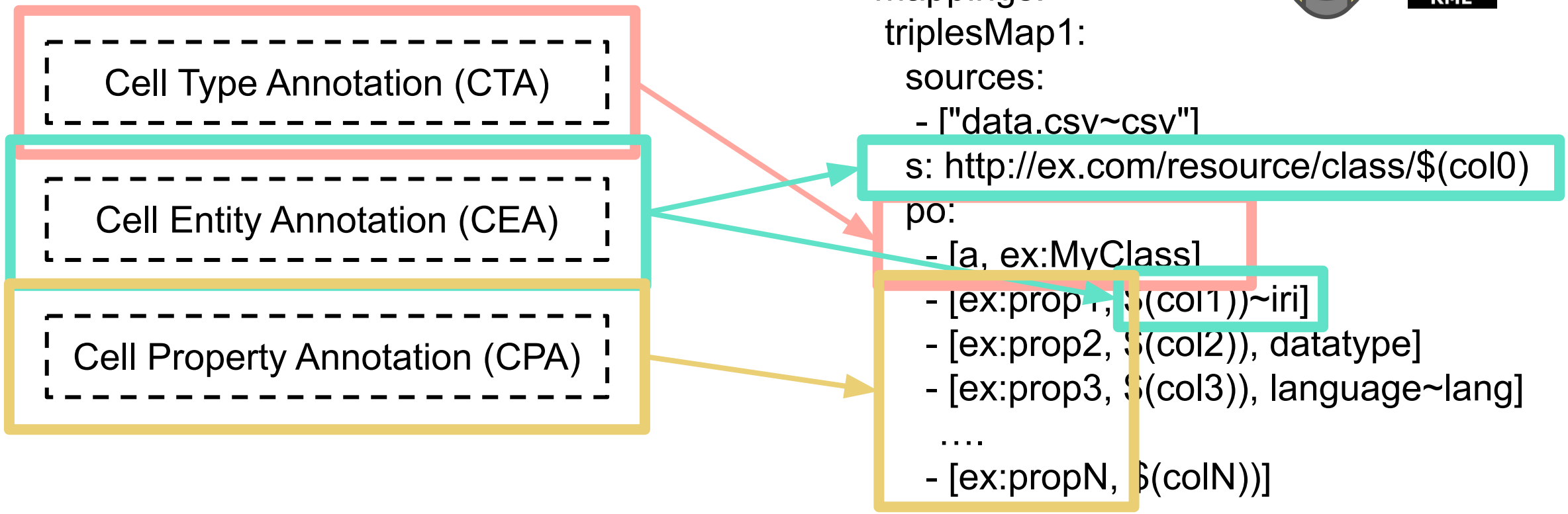
**We tried to compare the SemTab annotators…**

**Open Source Tools:** JenTab, MTab and Mantis V

**Outcomes:**

- Similar steps (e.g., KGs lookup, preprocessing, datatype prediction)
- Common procedures (e.g., majority vote/levenshtein distance)
- Blackboxes/Not explainable
- Iterative process

# SemTab and RML



# More questions than answers (I)

What happened to the **RDB2RDF automation** approaches (e.g., MIRROR, AutoMap4OBDA)? **adapt/extend them to this new generation**?

# More questions than answers (II)

Should we **extend current mapping languages**
to **describe more complex tasks**
beyond triples generation?

# More questions than answers (III)

Should we use **declarative description of functions**
to enhance the **transparency** & **explainability**
of current SemTab solutions?

# More questions than answers (IV)

Are **declarative mapping languages**
the ideal way of **representing automation**
despite the difference among **paradigms**?

# What do **YOU** think?

"*The annotation process is iterative by nature,*
*but **not sure if this iterative process should be included within the mappings**.*
*I would **give the final results of the CEA, CTA & CPA to the mappings** component.*"

Ernesto Jimenez-Ruiz

"*The **mappings should deal with the types & relationships (CTA, CPA)** detected by the matching process,*
*& leave CEA to the result of the matching process.*
*The inner workings of the -iterative- matching process are avoided =>*
*limit the extensions to the mapping language.*"

Francois Scharffe

*"in the end, it boils down to refactoring the code of the existing approaches to be integrated in RML,*
*but if **the whole functionality is then hidden in those function**, what would be the ultimate advantage?*
*An advantage could be mix-and-match experimentation,*
*i.e., use CEA from system 1 combined with CTA from system 2"*

Heiko Paulheim

# Conclusions / Vision

- Benefits when automation tasks are declaratively described, with respect to **maintenability, sustainability, and reproducibility**
- Directly aligning the automatic solutions with the declarative solutions might be **technically and conceptually challenging**
- Use **declarative descriptions of workflows** instead of mapping rules
- Would the automatic and declarative KG construction methods keep on growing in **different directions**?

# Declarative Description of Knowledge Graphs Construction Automation: Status & Challenges

David Chaves-Fraga, UPM & KULeuven & Flanders Make
Anastasia Dimou, KULeuven & Flanders Make & Leuven.AI

david.chaves@upm.es, anastasia.dimou@kuleuven.be
@dchavesf @natadimou