



Propagating Ontology Changes to Declarative Mappings in Construction of Knowledge Graphs

Diego Conde¹, Lise Stork², Romana Pernisch^{2,3}, María Poveda Villalón¹, Oscar Corcho¹, David Chaves-Fraga⁴

¹Ontology Engineering Group, Universidad Politécnica de Madrid

²Vrije Universiteit Amsterdam, Department of Computer Science, Amsterdam

³Elsevier, Discovery Lab, Amsterdam

⁴CiTIUS, Universidade de Santiago de Compostela

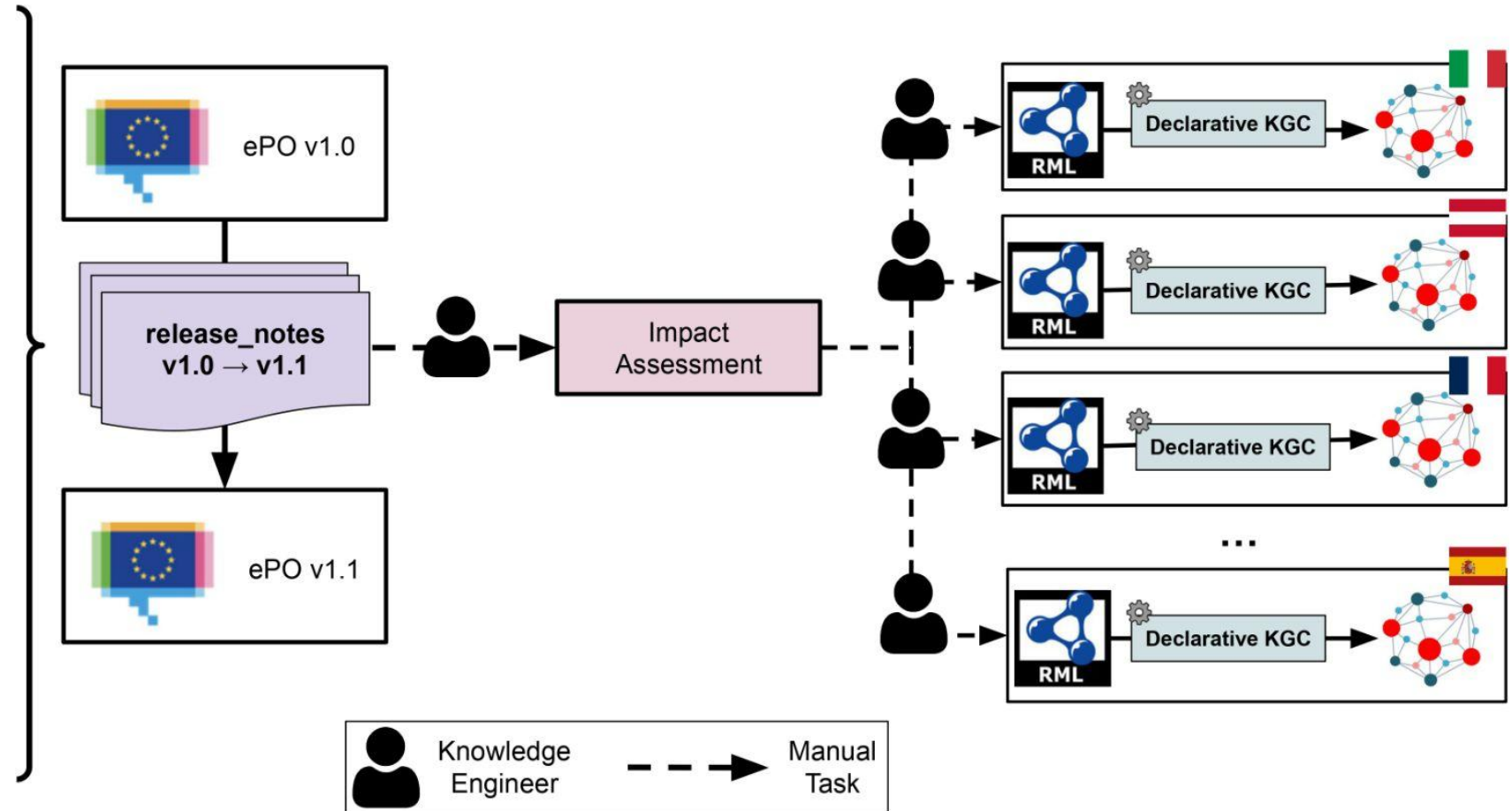
Changed classes

class	added attributes
epo:ProcurementElement	epo:hasDescription (moved from epo:ProcurementObject)
epo:ProcurementElement	epo:hasTitle (moved from epo:ProcurementObject)
epo:ProcurementObject	epo:hasAdditionalInformation (moved from epo:Lot)

Deleted classes

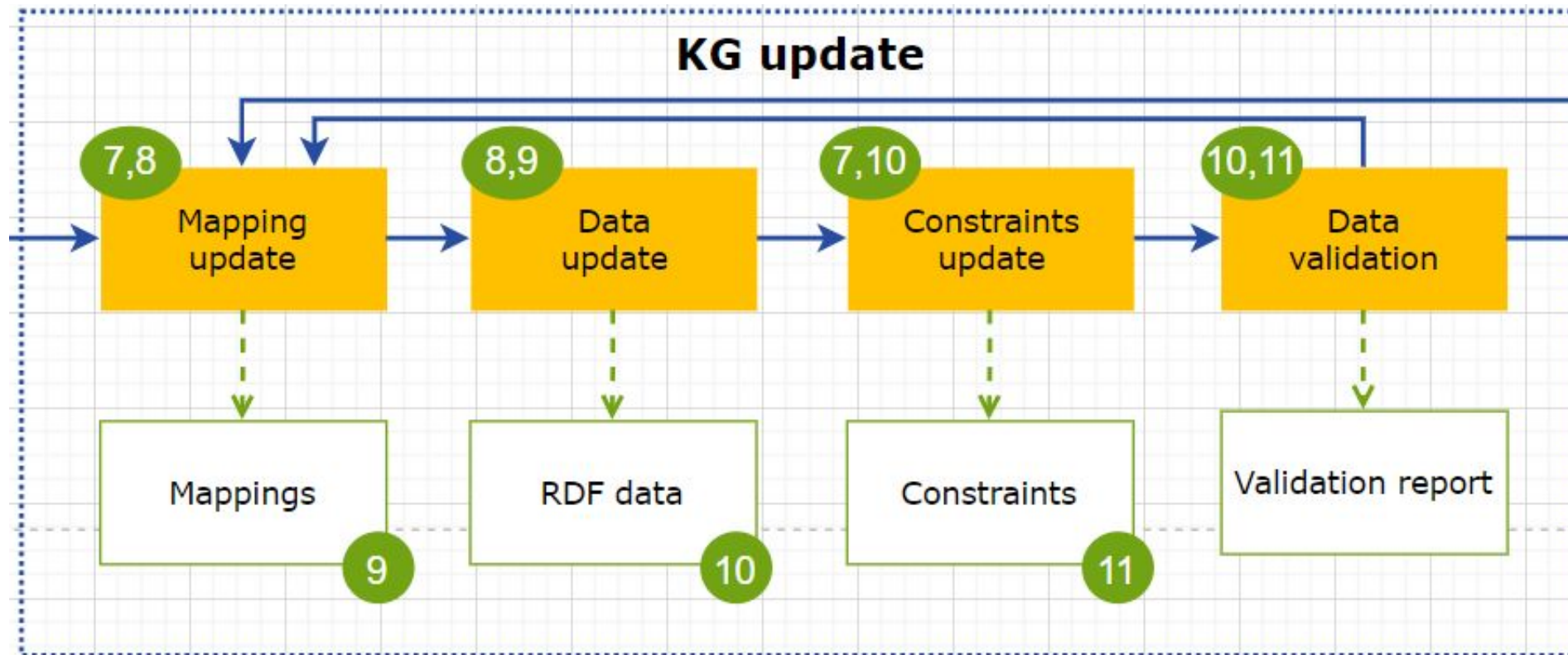
ePO core

- epo:BuyerSideSignatory
- epo:ContractSignatory
- epo:ContractorSideSignatory
- epo:GroupLeader
- epo:InformationProvider
- epo:PrimaryRole
- epo:SecondaryRole
- epo:TertiaryRole



https://single-market-economy.ec.europa.eu/single-market/public-procurement/digital-procurement/public-procurement-data-space-ppds_en

Whenever the ontology is changed the data has to be regenerated, and mappings, shapes, and queries, have to be **manually updated**



Pernisch, R., Poveda-Villalón, M., Conde-Herreros, D., Chaves-Fraga, D., & Stork, L. When Ontologies met Knowledge Graphs: Tale of a Methodology. Poster accepted at ESWC 2024

Previous work in **Ontology Evolution**:

- Defining **change operations** for the ontology evolution (Hartung, M. Rahm, E., 2013)
- Developing **ontologies** for describing said changes (Palma, R et al. 2009)
- Creating new ontology engineering **methodologies** for ontology evolution (Zablith, F et al. 2015)
- **Theoretical studies** on how ontology evolution impacts KG construction mappings. (Lembo D, et al. 2017)
- Change history **management frameworks** (Khattak, A. M et al. 2013)

Hartung, M., Groß, A., & Rahm, E. (2013). COnto-Diff: generation of complex evolution mappings for life science ontologies. *Journal of biomedical informatics*, 46(1), 15-32.

Palma, R., Haase, P., Corcho, O., & Gómez-Pérez, A. (2009). Change representation for OWL 2 ontologies.

Zablith, F., Antoniou, G., d'Aquin, M., Flouris, G., Kondylakis, H., Motta, E., ... & Sabou, M. (2015). Ontology evolution: a process-centric survey. *The knowledge engineering review*, 30(1), 45-75.

Lembo, D., Rosati, R., Santarelli, V., Savo, D. F., & Thorstensen, E. (2017). Mapping repair in ontology-based data access evolving systems. In *IJCAI International Joint Conference on Artificial Intelligence* (pp. 1160-1166). International Joint Conference on Artificial Intelligence.

Khattak, A. M., Latif, K., & Lee, S. (2013). Change management in evolving web ontologies. *Knowledge-Based Systems*, 37, 1-18.

Previous work in **Ontology Evolution**:

- Defining **change operations** (Hartung, M., Groß, A., & Rahm, E., 2013)
- Developing **ontology evolution languages** (Palma, R et al. 2009)
- Creating new **ontology evolution models** (Zablith, F et al. 2015)
- **Theoretical studies** on ontology evolution (Lembo, D., Rosati, R., Santarelli, V., Savo, D. F., & Thorstensen, E. 2017) KG construction
- **Change history management** in **ontology networks** (Khattak, A. M et al. 2013)

**NO STANDARDISED
VOCABULARY FOR
CHANGE
DOCUMENTATION**

Hartung, M., Groß, A., & Rahm, E. (2013). COnto-Diff: generation of complex evolution mappings for life science ontologies. *Journal of biomedical informatics*, 46(1), 15-32.

Palma, R., Haase, P., Corcho, O., & Gómez-Pérez, A. (2009). Change representation for OWL 2 ontologies.

Zablith, F., Antoniou, G., d'Aquin, M., Flouris, G., Kondylakis, H., Motta, E., ... & Sabou, M. (2015). Ontology evolution: a process-centric survey. *The knowledge engineering review*, 30(1), 45-75.

Lembo, D., Rosati, R., Santarelli, V., Savo, D. F., & Thorstensen, E. (2017). Mapping repair in ontology-based data access evolving systems. In *IJCAI International Joint Conference on Artificial Intelligence* (pp. 1160-1166). International Joint Conference on Artificial Intelligence.

Khattak, A. M., Latif, K., & Lee, S. (2013). Change management in evolving web ontologies. *Knowledge-Based Systems*, 37, 1-18.

A framework for the automation of the propagation of ontology changes to mappings improves the following aspects of KGC:

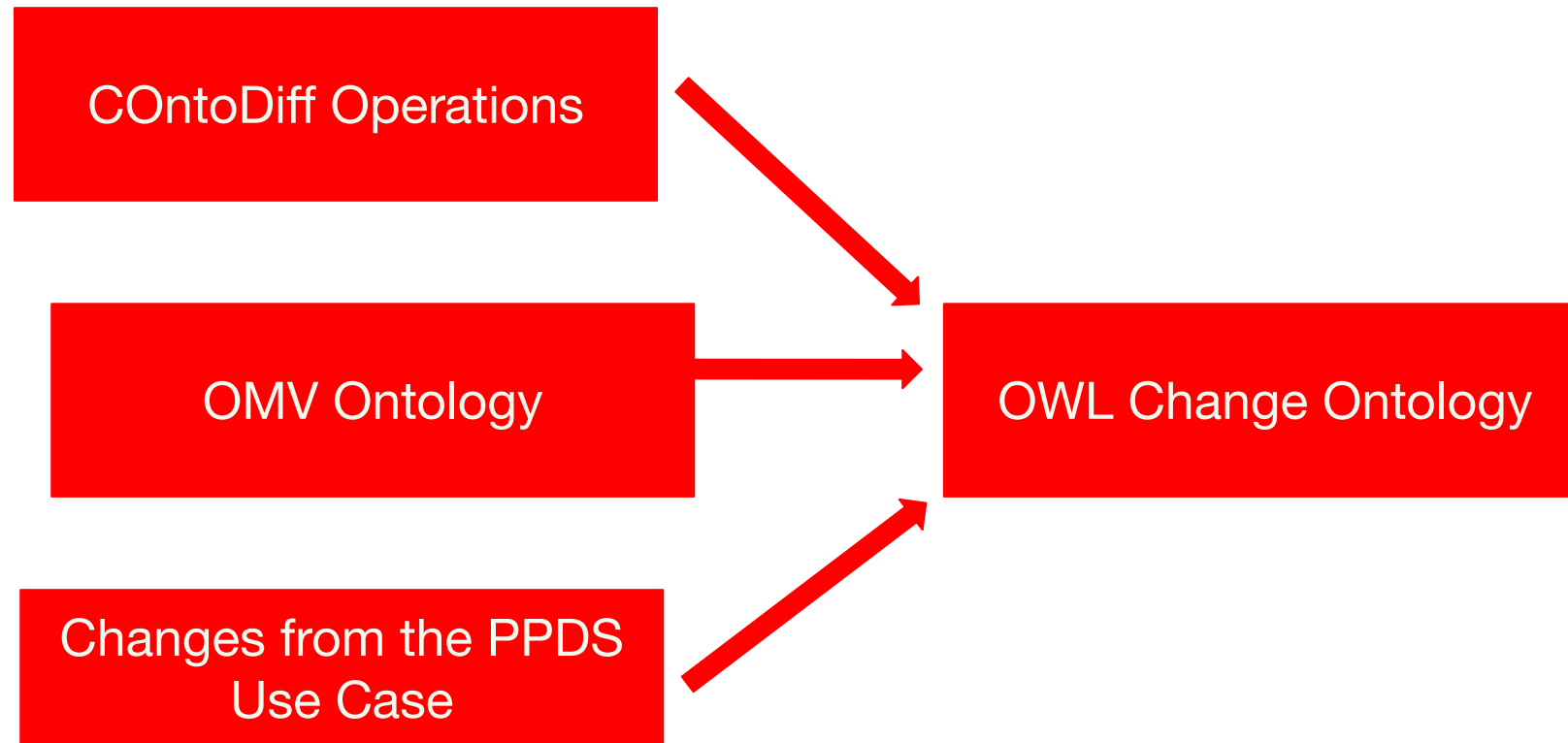
1

The manual work done by the knowledge engineers will be **drastically reduced**, only required when new knowledge needs to be integrated (e.g., adding a new class to the ontology).

2

The **KG construction pipeline can be informed about new rules to be processed**, avoiding the necessity of re-generating the complete knowledge graph from scratch, which is currently the common practice. The latter will additionally reduce execution time and memory consumption.

Our proposed vocabulary comes from a variety of sources:

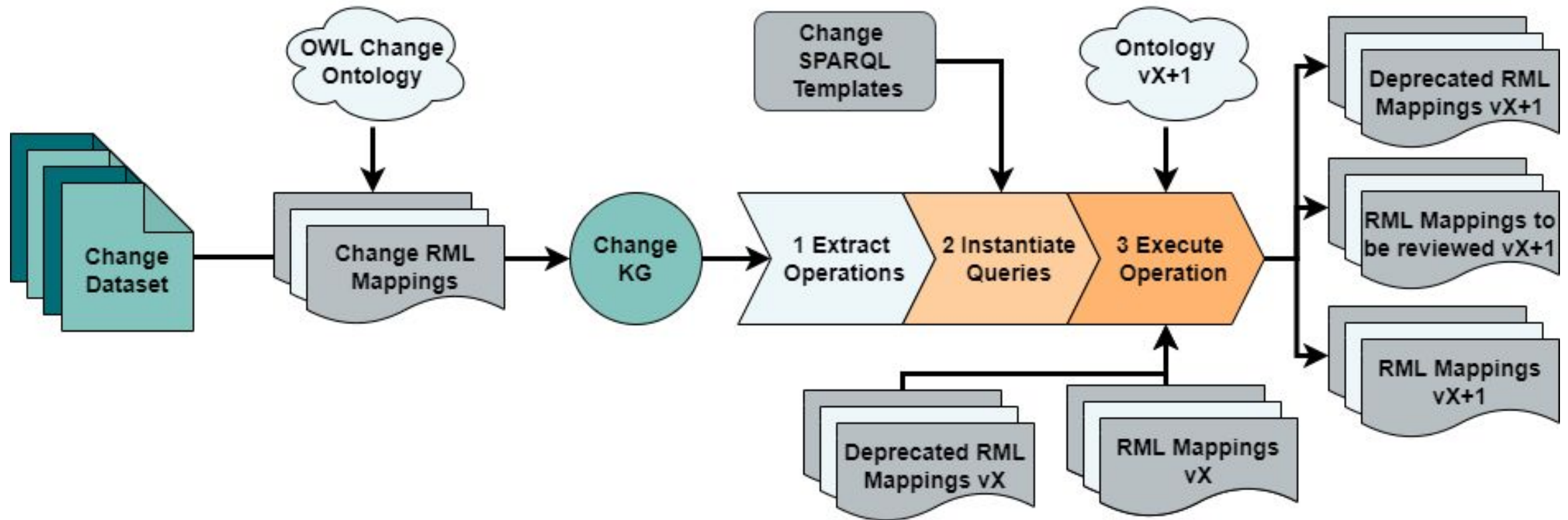


Hartung, M., Groß, A., & Rahm, E. (2013). COnto-Diff: generation of complex evolution mappings for life science ontologies. *Journal of biomedical informatics*, 46(1), 15-32.
Palma, R., Haase, P., Corcho, O., & Gómez-Pérez, A. (2009). Change representation for OWL 2 ontologies.

List of change operations with their effect on mappings

Operations	Changes	KE Intervention
AddClass(C)	Adds TriplesMap	YES
RemoveClass(C)	Removes TriplesMap and POM	NO
RenameTerm(T)	Replaces URI	NO
AddSubClass(C,D)	Adds Class to child & POM	NO
RemoveSubClass(C,D)	Removes Class from child & POM	NO
AddObjectProperty(C1,P,C2)	Adds POM	YES
RemoveObjectProperty(C1,P,C2)	Removes POM	NO
AddDataProperty(C,P)	Adds POM	YES
RemoveDataProperty(C,P)	Removes POM	NO
AddSubProperty(P,Q)	Adds rml:predicate	NO
RemoveSubProperty(P,Q)	Removes rml:predicate	NO
DeprecateElement(E)	Removes instances of Class or Property	NO
RevokeDeprecate(E)	Puts back instances of Class or Property	NO

POM = Predicate Object Map



<https://github.com/oeg-upm/ocp2kg>

List of change operations with their effect on mappings

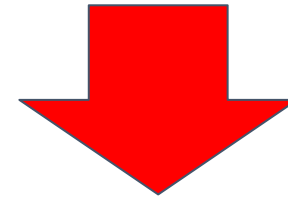
Operations	Changes	KE Intervention
AddClass(C)	Adds TriplesMap	YES
RemoveClass(C)	Removes TriplesMap and POM	NO
RenameTerm(T)	Replaces URI	NO
AddSubClass(C,D)	Adds Class to child & POM	NO
RemoveSubClass(C,D)	Removes Class from child & POM	NO
AddObjectProperty(C1,P,C2)	Adds POM	YES
RemoveObjectProperty(C1,P,C2)	Removes POM	NO
AddDataProperty(C,P)	Adds POM	YES
RemoveDataProperty(C,P)	Removes POM	NO
AddSubProperty(P,Q)	Adds rml:predicate	NO
RemoveSubProperty(P,Q)	Removes rml:predicate	NO
DeprecateElement(E)	Removes instances of Class or Property	NO
RevokeDeprecate(E)	Puts back instances of Class or Property	NO

POM = Predicate Object Map

From the 3.0.1 changelog:

New classes

- epo:ConcessionEstimate



```
epochanges:ACConcessionEstimate rdf:type och:AddClass .  
epochanges:ACConcessionEstimate och:addedClass epo:ConcessionEstimate .
```

```
epo:ConcessionEstimate:  
  sources:  
  - [XXXX~xxxx]  
  s: $(XXXX)  
  po:  
  - [rdf:type, epo:ConcessionEstimate]
```



List of change operations with their effect on mappings

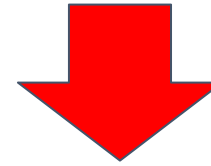
Operations	Changes	KE Intervention
AddClass(C)	Adds TriplesMap	YES
RemoveClass(C)	Removes TriplesMap and POM	NO
RenameTerm(T)	Replaces URI	NO
AddSubClass(C,D)	Adds Class to child & POM	NO
RemoveSubClass(C,D)	Removes Class from child & POM	NO
AddObjectProperty(C1,P,C2)	Adds POM	YES
RemoveObjectProperty(C1,P,C2)	Removes POM	NO
AddDataProperty(C,P)	Adds POM	YES
RemoveDataProperty(C,P)	Removes POM	NO
AddSubProperty(P,Q)	Adds rml:predicate	NO
RemoveSubProperty(P,Q)	Removes rml:predicate	NO
DeprecateElement(E)	Removes instances of Class or Property	NO
RevokeDeprecate(E)	Puts back instances of Class or Property	NO

POM = Predicate Object Map

From the 3.0.1 changelog

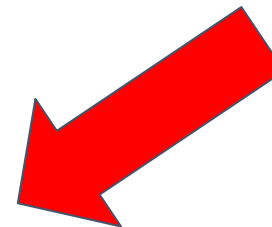
epo:SubmissionTerm

generalisation → epo:ProcedureSpecificTerm



```
epochanges:ASCProcedureSpecificTerm och:subAddSubClass epo:SubmissionTerm .
epochanges:ASCProcedureSpecificTerm rdf:type omv:AddSubClass .
epochanges:ASCProcedureSpecificTerm och:objAddSubClass epo:ProcedureSpecificTerm .
```

```
SubmissionTerm:
  sources:
  - [%1$s~xpath', /%2$s/PROCEDURE]
  s: $(if(exists(DATETIME_RECEIPT_TENDERS)
  po:
  - [rdf:type, epo:ProcedureSpecificTerm]
  - [rdf:type, epo:SubmissionTerm]
```



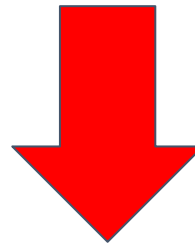
List of change operations with their effect on mappings

Operations	Changes	KE Intervention
AddClass(C)	Adds TriplesMap	YES
RemoveClass(C)	Removes TriplesMap and POM	NO
RenameTerm(T)	Replaces URI	NO
AddSubClass(C,D)	Adds Class to child & POM	NO
RemoveSubClass(C,D)	Removes Class from child & POM	NO
AddObjectProperty(C1,P,C2)	Adds POM	YES
RemoveObjectProperty(C1,P,C2)	Removes POM	NO
AddDataProperty(C,P)	Adds POM	YES
RemoveDataProperty(C,P)	Removes POM	NO
AddSubProperty(P,Q)	Adds rml:predicate	NO
RemoveSubProperty(P,Q)	Removes rml:predicate	NO
DeprecateElement(E)	Removes instances of Class or Property	NO
RevokeDeprecate(E)	Puts back instances of Class or Property	NO

POM = Predicate Object Map

From the 4.0 changelog:

class	removed property
epo:LotAwardOutcome	epo:hasAwardedEstimatedValue → epo:MonetaryValue

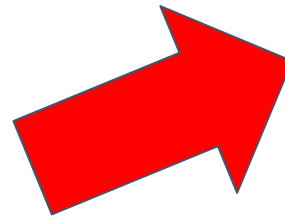


```
epochanges:ROPhasEstimatedValue1 och:rangeRemoveObjectProperty epo:MonetaryValue .  
epochanges:ROPhasEstimatedValue1 och:propertyRemoveObjectProperty epo:hasEstimatedValue .  
epochanges:ROPhasEstimatedValue1 rdf:type och:RemoveObjectProperty .  
epochanges:ROPhasEstimatedValue1 och:domainRemoveObjectProperty epo:LotGroup .
```

```
mappings:  
  LotAwardOutcomeModification:  
    sources:  
    - ['%1$s~xpath', /%2$s/MODIFICATIONS  
s: http://data.europa.eu/a4g/resource  
po:  
- [rdf:type, epo:LotAwardOutcome]
```

```
mappings:  
  LotAwardOutcomeModification: You 2 months ago + minor change  
  p: epo:hasRestatedAwardedValue  
  o:  
    mapping: LotAwardOutcomeAwardedMonetaryValueModification  
    condition:  
      function: equal  
      parameters:  
      - [str1, $(XXXX)]  
      - [str2, $(XXXX)]  
- p: epo:hasRestatedAwardedValue  
  o:  
    mapping: LotAwardOutcomeAwardedMonetaryValue  
    condition:  
      function: equal  
      parameters:  
      - [str1, $(XXXX)]  
      - [str2, $(XXXX)]
```

Both predicate object maps
that correspond to the
epo:MonetaryValue
TriplesMaps are removed



- We performed evaluation performing a reduced set of the change operations of the ePO for easy visualization
 - We measure the performed operations in terms of the **number and type of term maps** that the updated mappings have.

Operations performed in evaluation

Operations	Parameters
<i>AddClass</i>	<i>epo:ConcessionEstimate</i>
<i>AddSubClass</i>	<i>epo:SubmissionTerm,epo:ProcedureSpecificTerm</i>
<i>AddObjectProperty</i>	<i>org:Organization,epo:hasMainActivity,at-voc:main-activity</i>
<i>RemoveObjectProperty</i>	<i>org:Organization,epo:hasMainActivityType,at-voc:main-activity</i>
<i>AddDataProperty</i>	<i>org:Organization,epo:hasBuyerLegalTypeDescription</i>
<i>RemoveDataProperty</i>	<i>org:Organization,epo:hasBuyerTypeDescription</i>

Operations	Parameters
<i>AddClass</i>	<i>epo:ConcessionEstimate</i>
<i>AddSubClass</i>	<i>epo:SubmissionTerm, epo:ProcedureSpecificTerm</i>
<i>AddObjectProperty</i>	<i>org:Organization, epo:hasMainActivity, at-voc:main-activity</i>
<i>RemoveObjectProperty</i>	<i>org:Organization, epo:hasMainActivityType, at-voc:main-activity</i>
<i>AddDataProperty</i>	<i>org:Organization, epo:hasBuyerLegalTypeDescription</i>
<i>RemoveDataProperty</i>	<i>org:Organization, epo:hasBuyerTypeDescription</i>



	Old	Updated
#TriplesMap	2	3
#LogicalSource	2	3
#SubjectMap	2	3
#PredicateObjectmap	3	3

Operations	Parameters
<i>AddClass</i>	<i>epo:ConcessionEstimate</i>
<i>AddSubClass</i>	<i>epo:SubmissionTerm, epo:ProcedureSpecificTerm</i>
<i>AddObjectProperty</i>	<i>org:Organization, epo:hasMainActivity, at-voc:main-activity</i>
<i>RemoveObjectProperty</i>	<i>org:Organization, epo:hasMainActivityType, at-voc:main-activity</i>
<i>AddDataProperty</i>	<i>org:Organization, epo:hasBuyerLegalTypeDescription</i>
<i>RemoveDataProperty</i>	<i>org:Organization, epo:hasBuyerTypeDescription</i>



	Old	Updated
#TriplesMap	2	3
#LogicalSource	2	3
#SubjectMap	2	3
#PredicateObjectMap	3	5

rr:class is added to the SubjectMap but no POMs are inherited since it is not within mappings

Operations	Parameters
<i>AddClass</i>	<i>epo:ConcessionEstimate</i>
<i>AddSubClass</i>	<i>epo:SubmissionTerm,epo:ProcedureSpecificTerm</i>
<i>AddObjectProperty</i>	<i>org:Organization,epo:hasMainActivity,at-voc:main-activity</i>
<i>RemoveObjectProperty</i>	<i>org:Organization,epo:hasMainActivityType,at-voc:main-activity</i>
<i>AddDataProperty</i>	<i>org:Organization,epo:hasBuyerLegalTypeDescription</i>
<i>RemoveDataProperty</i>	<i>org:Organization,epo:hasBuyerTypeDescription</i>



	Old	Updated
#TriplesMap	2	3
#LogicalSource	2	3
#SubjectMap	2	3
#PredicateObjectMap	3	5

Operations	Parameters
<i>AddClass</i>	<i>epo:ConcessionEstimate</i>
<i>AddSubClass</i>	<i>epo:SubmissionTerm,epo:ProcedureSpecificTerm</i>
<i>AddObjectProperty</i>	<i>org:Organization,epo:hasMainActivity,at-voc:main-activity</i>
<i>RemoveObjectProperty</i>	<i>org:Organization,epo:hasMainActivityType,at-voc:main-activity</i>
<i>AddDataProperty</i>	<i>org:Organization,epo:hasBuyerLegalTypeDescription</i>
<i>RemoveDataProperty</i>	<i>org:Organization,epo:hasBuyerTypeDescription</i>



	Old	Updated
#TriplesMap	2	3
#LogicalSource	2	3
#SubjectMap	2	3
#PredicateObjectMap	3	5

No POM are deleted since that property is not within the mappings

Operations	Parameters
<i>AddClass</i>	<i>epo:ConcessionEstimate</i>
<i>AddSubClass</i>	<i>epo:SubmissionTerm,epo:ProcedureSpecificTerm</i>
<i>AddObjectProperty</i>	<i>org:Organization,epo:hasMainActivity,at-voc:main-activity</i>
<i>RemoveObjectProperty</i>	<i>org:Organization epo:hasMainActivityType at-voc:main-activity</i>
<i>AddDataProperty</i>	<i>org:Organization,epo:hasBuyerLegalTypeDescription</i>
<i>RemoveDataProperty</i>	<i>org:Organization,epo:hasBuyerTypeDescription</i>



	Old	Updated
#TriplesMap	2	3
#LogicalSource	2	3
#SubjectMap	2	3
#PredicateObjectMap	3	5

Operations	Parameters
<i>AddClass</i>	<i>epo:ConcessionEstimate</i>
<i>AddSubClass</i>	<i>epo:SubmissionTerm,epo:ProcedureSpecificTerm</i>
<i>AddObjectProperty</i>	<i>org:Organization,epo:hasMainActivity,at-voc:main-activity</i>
<i>RemoveObjectProperty</i>	<i>org:Organization,epo:hasMainActivityType,at-voc:main-activity</i>
<i>AddDataProperty</i>	<i>org:Organization,epo:hasBuyerLegalTypeDescription</i>
<i>RemoveDataProperty</i>	<i>org:Organization,epo:hasBuyerTypeDescription</i>



	Old	Updated
#TriplesMap	2	3
#LogicalSource	2	3
#SubjectMap	2	3
#PredicateObjectMap	3	5

No POM are deleted since that property is not within the mappings

- We have created an **ontology for documenting changes** for OWL Ontologies.
- We have created a **tool that semi-automatically propagates ontology changes to mappings** with minimal KE intervention as seen in the PPDS evaluation. (<https://github.com/oeg-upm/ocp2kg>)
 - Its main limitation is that it **still requires KE intervention** whenever new knowledge is added.

- **Perform Experimental Study to study impact on execution time and memory consumption for KG construction engines.**

- Perform Experimental Study to study impact on execution time and memory consumption for KG construction engines.
- **Extend the propagation of ontology changes to SHACL shapes and SPARQL queries.**

- Perform Experimental Study to study impact on execution time and memory consumption for KG construction engines.
- Extend the propagation of ontology changes to SHACL shapes and SPARQL queries.
- **Study changes of other parts of the ETL pipeline, such as the source data or the RML specs.**

- Perform Experimental Study to study impact on execution time and memory consumption for KG construction engines.
- Extend the propagation of ontology changes to SHACL shapes and SPARQL queries.
- Study changes of other parts of the ETL pipeline, such as the source data or the RML specs.
- **Study the use of OWL inference to propagate changes (subClassOf and subPropertyOf mainly).**

When Ontologies met Knowledge Graphs: Tale of a Methodology

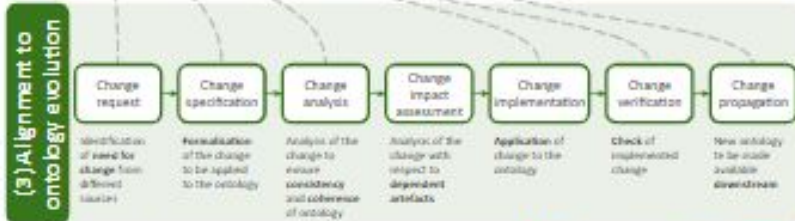
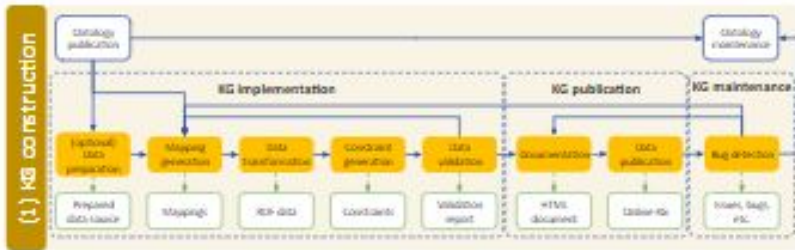
Romana Pernisch, María Poveda Villalón, Diego Conde-Herreros, David Chaves-Fraga, Lise Stork

Problem:

Do you miss a methodology to construct your knowledge graph? Do you suffer through ontology evolution in silence?

Solution:

- (1) definition of KG construction activities,
- (2) integration with LOT,
- (3) comparison to ontology evolution.



María Poveda Villalón is funded by the European Union's Horizon 2020 research and innovation programme under the grant agreement no. 101019161 (LOT4KG). David Chaves-Fraga is funded by the Galician University of Education, University and Professional Training and the European Regional Development Fund (ERDF) through grant PID1120/18/00 and PID1120/18/00. Diego Conde-Herreros is supported by the project Knowledge Spaces (KSP) PID1120/18/00 funded by MICINN/18/11364/PG1/18/00111/18 and by Spanish National R&D.



When Ontologies met Knowledge Graphs, Tale of a Methodology

Romana Pernisch, María Poveda-Villalón, Diego Conde-Herreros, David Chaves-Fraga, and Lise Stork

Another part of the work we are doing is the extension to the Linked Open Terms methodology for integrating Ontology and KG lifecycle

Please come see us!!!!



Propagating Ontology Changes to Declarative Mappings in Construction of Knowledge Graphs

Diego Conde¹, Lise Stork², Romana Pernisch^{2,3}, María Poveda Villalón¹, Oscar Corcho¹, David Chaves-Fraga⁴

¹Ontology Engineering Group, Universidad Politécnica de Madrid

²Vrije Universiteit Amsterdam, Department of Computer Science, Amsterdam

³Elsevier, Discovery Lab, Amsterdam

⁴CiTIUS, Universidade de Santiago de Compostela