KNOWLEDGE GRAPHS LAB

BDMA: NAZGUL RAKHIMZHANOVA, YANJIAN ZHANG

INTRODUCTION

To complete this laboratory work we used data files that were generated for Property graph laboratory.
Although we haven't included some concepts as Keywords and Abstract.

B.1 TBOX definition

To define the TBOX we have chosen OWL language, as it is more flexible in terms of setting restrictions and programming using the Python language.
Firstly we visualized our TBOX using a generic diagram (not pure UML class) (fig. 1) and later programmatically created an OWL file with TBOX, using owlready2 package for Python.

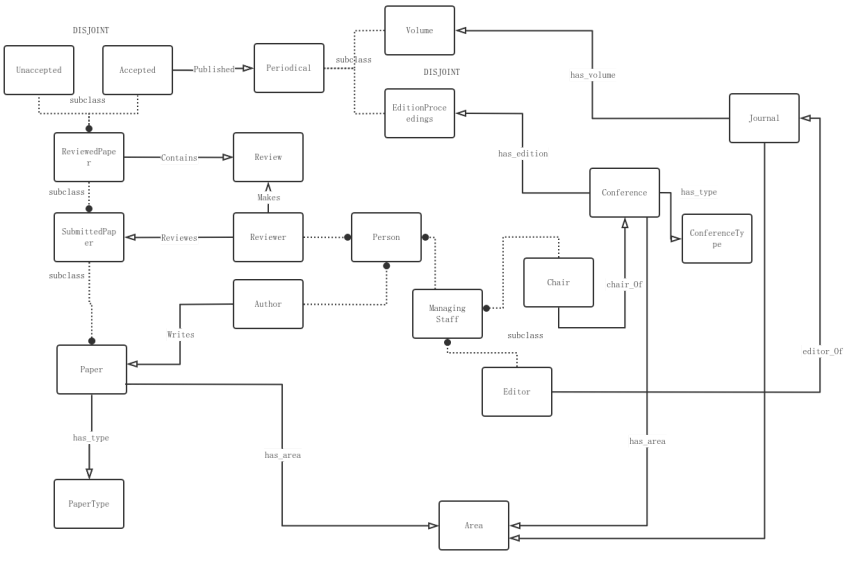


Figure 1. TBOX visual representation

In our TBOX we defined the following assumptions:

- Conference has a property as ConferenceType(Symposium, Workshop, Expert Group and Regular Conference)
- A conference has one or more Editions, and each Edition has Edition Proceedings (here, to keep our graph simple we haven't included Editions of Conference as node,

Code Link:
https://github.com/yanjianzhang/FIB-Course-Project/tree/master/Semantic_Data_Management/Project_3

but in real case application we do recommend to add Edition as a subclass of conference and link Edition Proceedings to Edition, rather than to Conference)

- A conference has a chair
- A journal has one or more Volumes
- A journal has a editor
- Both chair and editor are subclasses of ManagingStaff class (here, we haven't specify these classes as disjoint, as a person who is in chair board of conference CAN be an editor of a journal)
- Journal Volume and Conference edition's proceedings are both subclasses of a class Periodical
- Authors, Reviewers, Chair and Editors are subclasses of class Person
- Author writes a paper
- Chair and Editor assign Reviewer
- Reviewer makes a review for a paper
- Paper has a property as a PaperType(short, full, demo, and poster) (here, we decided to define paper type as a property instead of subclass, as we wanted to define submitted paper as a Paper subclass. To keep consistent approach we also defined conference type as a property instead of defining is as a subclasses)
- Paper has a subclass as SubmittedPaper
    - SubmittedPaper has a subclass as ReviewedPaper
        - ReviewedPaper has 2 disjoint subclasses as - Accepted_Paper and Unaccepted_Paper

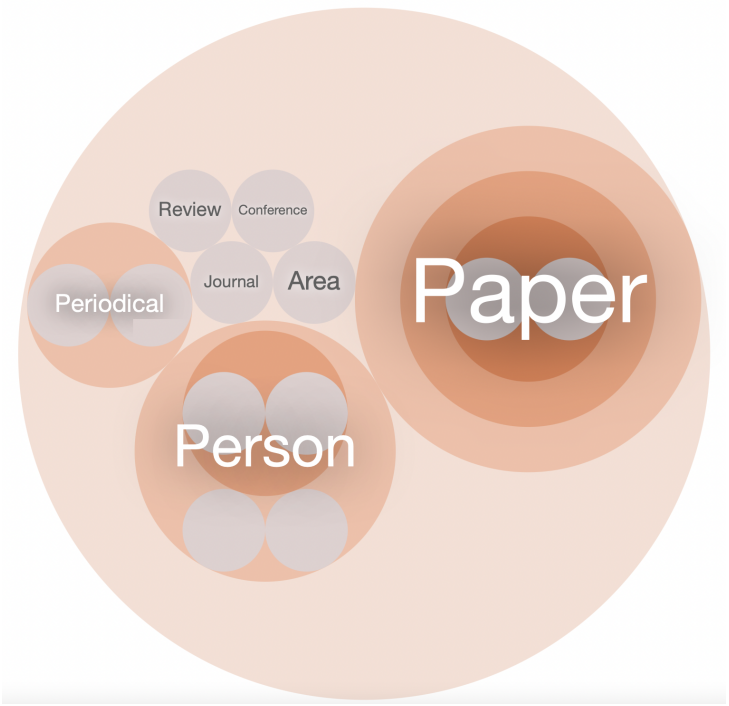


Figure 2. Publications graph hierarchy

B.2 ABOX DEFINITION

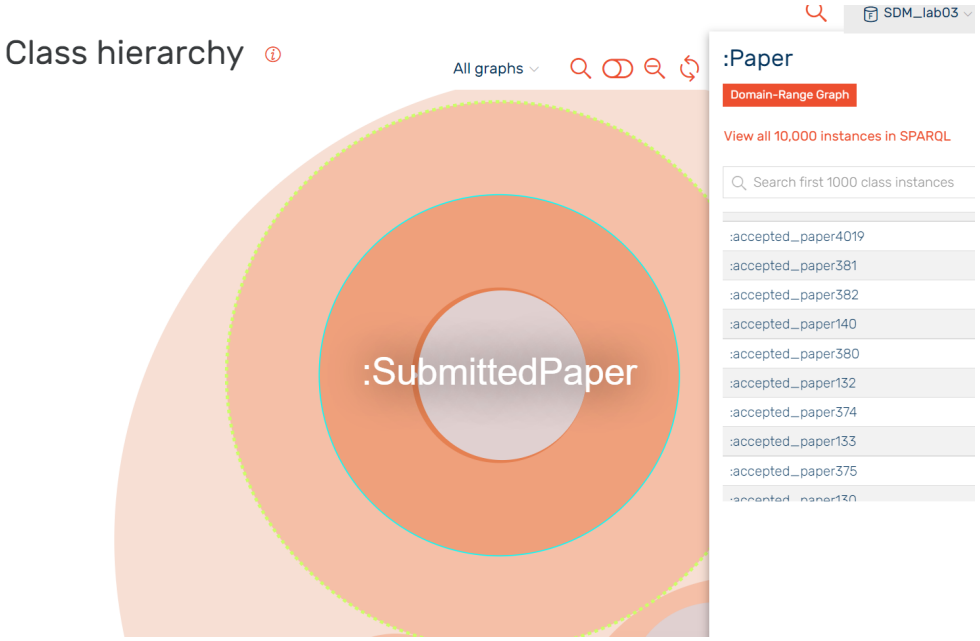


Figure 3. Instances of Paper loaded to graph with ABOX

We create ABOX with the following steps:

- Load the conference information table and create the instances of Edition Proceedings, Conference, Area and Chairs.
- Load the paper-conference table and create the instances of Paper, Author, Reviewer, Review and Area. Randomly assign paper into accepted paper and unaccepted paper with 1% unaccepted rate.
- Load the journal information table and create the instances of Volume, Journal, Areas and Editors.
- Load the paper-journal table and create the instances of Paper, Author, Reviewer, Review and Area, . Randomly assign paper into accepted paper and unaccepted paper with 1% unaccepted rate.

B.3 CREATE THE FINAL ONTOLOGY

We used OWL QL reasoner and it inferred the following knowledge from ontology:

- Person is an Author if it has a predicate writes, Person is a Reviewer if it has a predicate reviews, and respectively for Chair and Editor
- A Paper is infered to be accepted if it has been Reviewed and Accepted

To get amount of different class instances we run a couple of SPARQL queries, as

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX : <http://localhost:7200/SDM_lab03#>
SELECT (COUNT(?s) AS ?CNT_of_class_Instances)
   WHERE {
     ?s a :Conference .
     ?s :has_type ?type .
```

```
filter contains(?type, "workshop")
}
```

#	Class	Type	Amount of entities in resulting graph	Subclass			Amount of entities in resulting graph
1	**PERSON**	-	-	Author			17263
				Reviewer			12623
				Managing Staff		Chair	135
						Editor	10
2	**PAPER**	Short paper	2904	Submitted	Reviewed	Accepted Paper	9904
		Full paper	2967				
		Demo paper	2870			Unaccepted Paper	96
		Poster paper	1259				
3	**CONFERENCE**	Workshop	2	-	-	-	-
		Expert Group	1	-	-	-	-
		Symposium	2	-	-	-	-
		Regular Conference	2	-	-	-	-
4	**PERIODICAL**	-	-	Volume			660
				Edition Proceedings			27
5	**JOURNAL**		10	-	-	-	-
6	**AREA**		18	-	-	-	-
7	**REVIEW**		30000				

Table 1. Classes breakdown

To get the number of predicates we run the following query

```
PREFIX : <http://localhost:7200/SDM_lab03#>
PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select ?p (count(?p) as ?totalCount)
where {
   ?s ?p ?o
}
group by ?p
order by DESC(?totalCount)
```

#	Predicate	Number
1	:fullname	30753
2	:has_review	30000
3	:writes	28125
4	:has_area	19344
5	:is_reviewedBy	15000
6	:makes	15000
7	:reviews	15000
8	:accepted	9904
9	:paper_type	10000
10	:published	9904
11	:title	10000
12	:periodicaltype	687
13	:has_volume	660
14	:is_volume	660
15	:chairOf	135
16	:has_edition	27
17	:is_edition	27
18	:conf_type	7
19	:editorOf	10
RDFS predicates		
20	rdf:first	8
21	rdf:rest	8
22	rdfs:subClassOf	100
23	rdfs:domain	55
24	rdfs:subPropertyOf	40
25	rdfs:range	28
OWL predicates		
26	owl:equivalentClass	16
27	owl:onProperty	13

28	owl:inverseOf	12
29	owl:hasValue	4
30	owl:intersectionOf	4
31	owl:onClass	4
32	owl:qualifiedCardinality	3
33	owl:disjointWith	2
34	owl:minQualifiedCardinality	1

Table 2. Properties breakdown

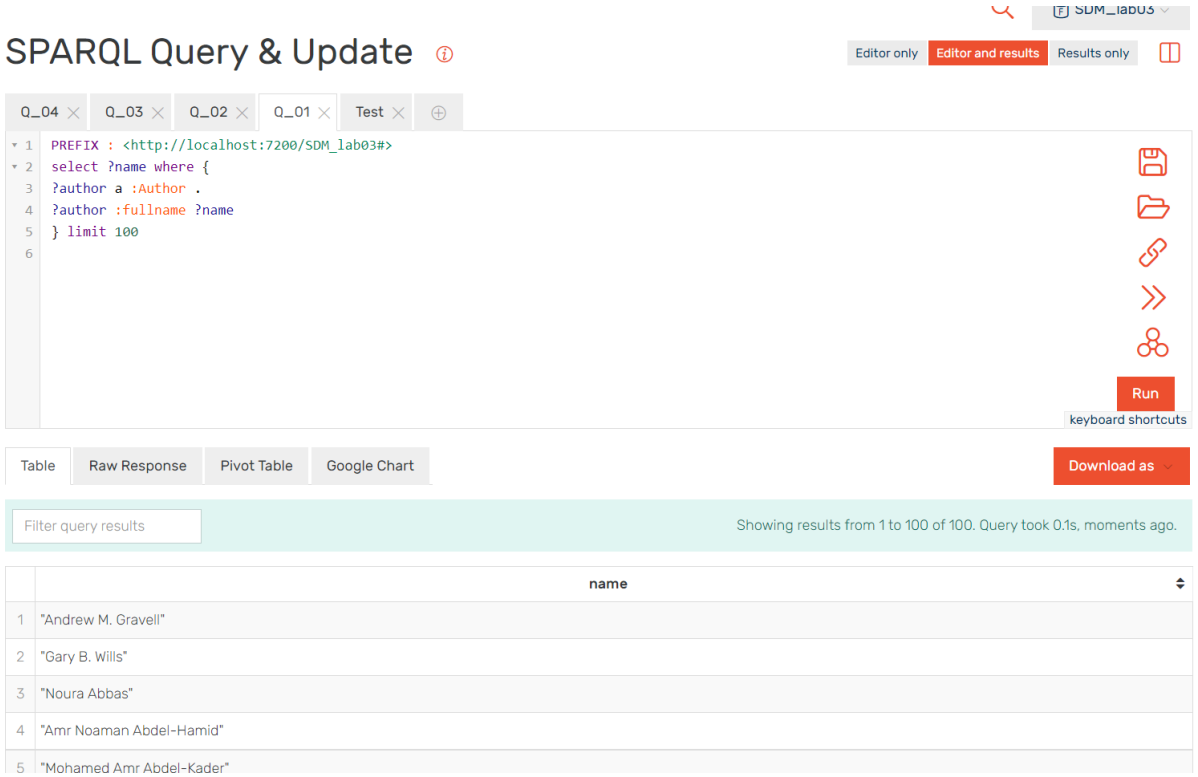
#	Class	Number of triplets
1	Paper	179255
2	Author	114440
3	Conference	98
4	Journal	770

Table 3. Main classes' triplets breakdown

B.4 QUERYING THE ONTOLOGY

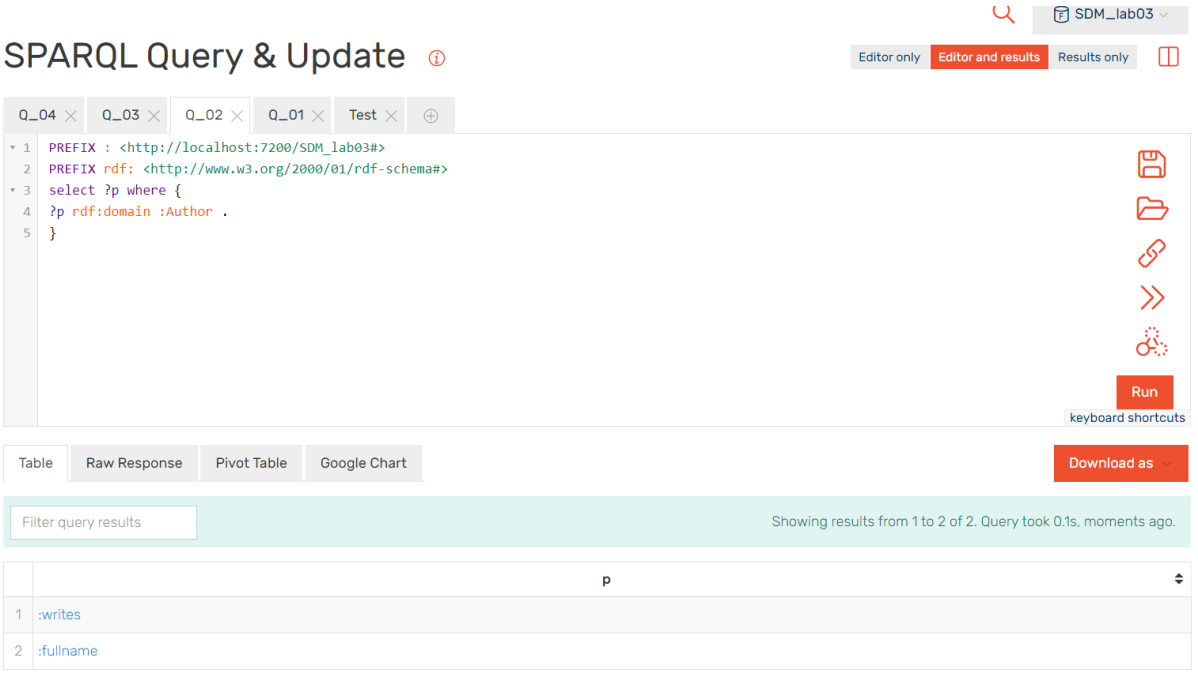
Query 1:

```
PREFIX : <http://localhost:7200/SDM_lab03#>
PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
select ?name where {
?author a :Author .
?author :fullname ?name
} limit 100
```



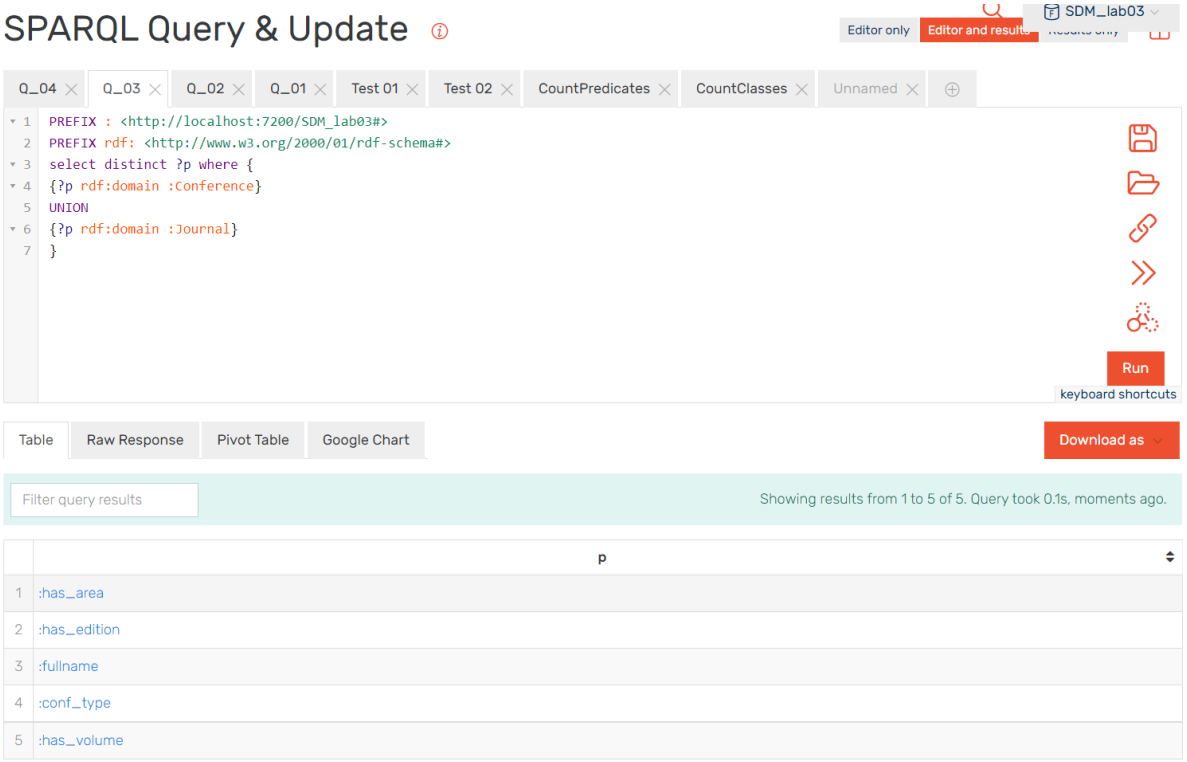
Query 2:

```
PREFIX : <http://localhost:7200/SDM_lab03#>
PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
select ?p where {
?p rdf:domain :Author .
}
```



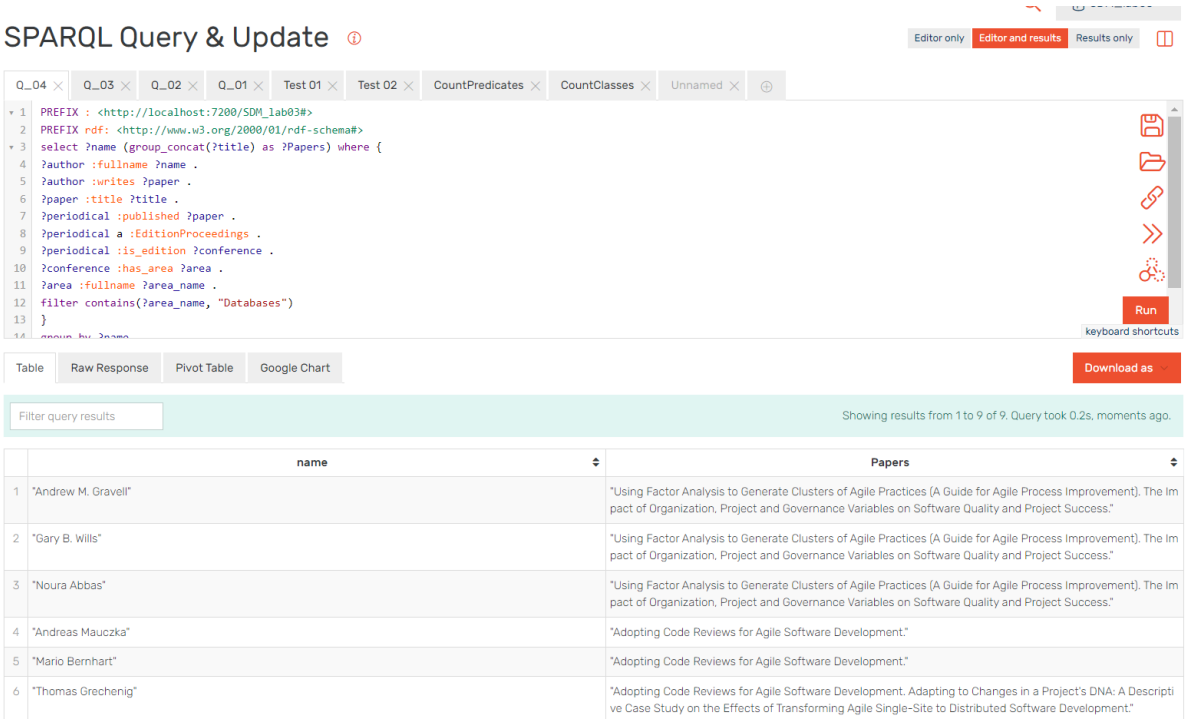
Query 3:

```
PREFIX : <http://localhost:7200/SDM_lab03#>
PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
select distinct ?p where {
{?p rdf:domain :Conference}
UNION
{?p rdf:domain :Journal}
}
```



Query 4:

```
PREFIX : <http://localhost:7200/SDM_lab03#>
PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
select ?name (group_concat(?title) as ?Papers) where {
?author :fullname ?name .
?author :writes ?paper .
?paper :title ?title .
?periodical :published ?paper .
?periodical a :EditionProceedings .
?periodical :is_edition ?conference .
?conference :has_area ?area .
?area :fullname ?area_name .
filter contains(?area_name, "Databases")
}
group by ?name
limit 9
```



CONCLUSION

This was the most interesting laboratory work where we were able to apply knowledge and skills we obtained during previous lab works and lectures.
The most challenging part was to define the TBOX, during completion of the laboratory work we have created 3 different TBOX structures for the Publications domain - each of them had its cons and pros. We believe that the one that we included to the final submission is the most elegant and simple one.

References

[1] Reasoning, Owlready2's documentation, 2022, https://owlready2.readthedocs.io/en/latest/reasoning.html
[2] SPARQL queries to obtain statistics, GITHub, https://gist.github.com/yayamamo/8052bd4620c1c58adff8