UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Semantic Data Management

Lab 3: Knowledge Graphs

by

Luis Alfredo León Villapún
Zyad Abduljabbar Moqbel Al-Azazi

Section B

Section B.1: TBOX Definition

For this section, we decided to create the graph TBOX schema first using Jena. This script can be triggered on the project repo at src->main->java->TBox.java. A copy of this script can also be found at the root of the repo with the naming convention required for the lab as: Group12D-SubSectionB1_TBOXCopy-LeonAlAzazi.java. The graphical representation of the TBOX was created with the Grafo tool. The resulting ontology file is saved as an OWL file (final-ontology.owl) and can also be found inside the project repo with the required name for the lab as: Group12D-SubSectionB1_TBOX-LeonAlAzazi.owl .

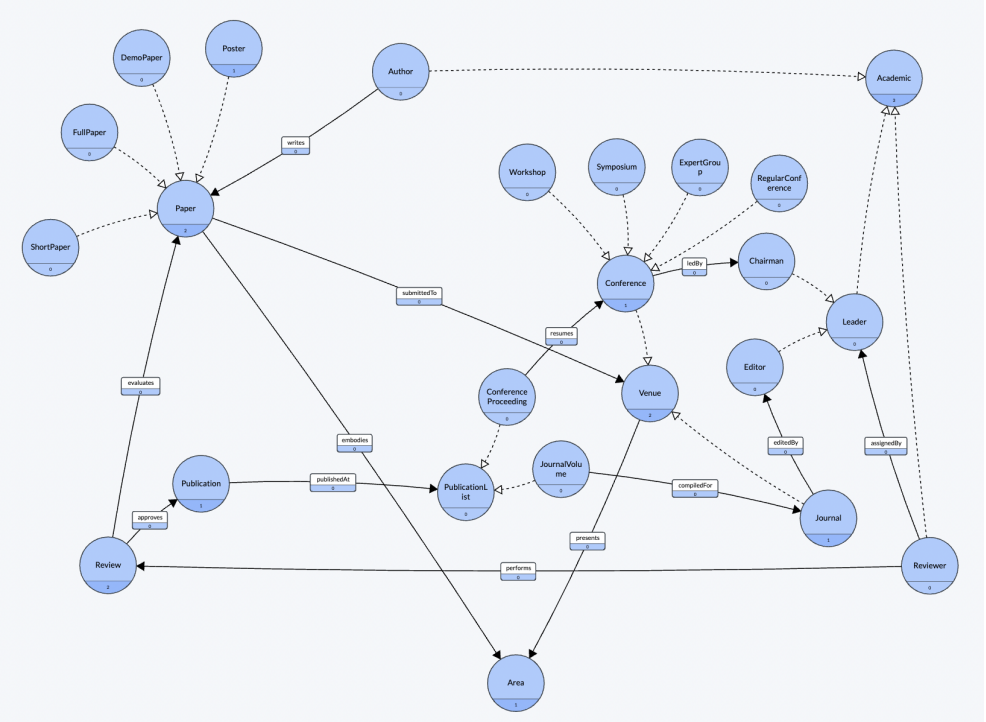

Figure 1. TBOX Graphical Representation.

The knowledge graph language we decided to use was OWL due to it being a standard language highly supported by the framework we chose to work – Jena API. In addition to the reasoning and inference capabilities that the language offers since it is based on formal semantics, which, in turn, enables reasoning and consistency checking.

As for our conceptual model, we created the superclass “Academic,” which represents an academic researcher and has the following subclasses: “Author”, “Reviewer” and “Leader.” The last class has two subclasses further: “Chairman” and “Editor.” This hierarchy comes from the assumption that all the personnels involved in the publication process are all academic researchers. The class “Paper” represents the written work written by an author and it has the subclass: “Full Paper”, “Short Paper”, “Demo Paper” and “Poster.” A paper, in this case, represents the different types of submission accepted in the academic publication process and paper is submitted to a “Venue”, which could be a “Conference” or a “Journal.” A “Conference” has four different subclasses: “Regular Conference”, “Expert Group”, “Symposium” and “Workshop” and the main reason for creating these subclasses is

to facilitate the incorporation of further specialized attributes for each of the subclasses in the future. A "Conference" is led by a "Chairman" and a "Journal" is edited by an "Editor." Another class is "Review" representing the review performed by a reviewer and evaluating a paper; it includes the final decision and justification attributes. Since two reviews with the final decisions of acceptance from reviewers assigned by the venue leader of which the paper was submitted to are needed for a paper to be published, we created the class "Publication." The class "PublicationList" represents the list of publications associated with a certain venue; it has two subclasses: "ConferenceProceeding" and "JournalVolume." "ConferenceProceeding" is linked to "Conference" using the property "resumes", while the "JournalVolume" is linked to the "Journal" using the property "compiledFor." Finally, the class "Area" represents the scientific branch/ area a paper's topic is from and the area a certain venue is about. "Venue" presents "Area", i.e. a conference or a journal presents publications in a certain area; whereas "Paper" is linked to "Area" through the property embodies, which represents how a paper embodies a work in a certain area (an assumption we made in this case is that the area and key words of "Publication" are the same as "Paper", they never change). "Area" has one attribute – keyword, which includes the keywords of the paper.

Limitations:

We are aware that our TBOX design has shortcomings and lacks some things under the assumptions that we made. For example, we did not model the presentation of the paper in the conference and who present it as we assumed that the authors present it and the property will just be a redundancy. As for some of the other attributes that exist in the TBOX with no ABOX instances, it is because of the lack of real life data, in addition to the same classes having these attributes already having attributes in both the TBOX and the ABOX. Our design of the subclasses for "Paper" and "Conference" may seem to have been better if we represented the subclasses as attributes; however, our choice of using subclasses comes from our consideration to future potential development of the graph that could allow defining more attributes to these subclasses in a way that represents their actual differences in real life. A final note regarding the constraints required, we implemented them in ABOX level to avoid any drastic changes in our modeling that could have complicated our TBOX (such as modeling "Poster" as a separate class for it to be published only on conferences).

Section B.2: ABOX Definition

To create the ABOX, we used an enriched version of the kaggle dataset BYU Publications in Scopus 2017-2021. To enrich the dataset, we first located the parts of the TBOX that would require extra data to be fulfilled. These sections were, in particular: reviewer, chairman, editor, leader, type of paper, type of conference, review, review decision, etc. All this data was randomly generated with a Jupyter notebook, included in the project repository. The enriched csv is called *triplets_csv_parsed.csv.* This csv can also be found as a copy at the root of the repo with the required naming conventio

	Authors	Author(s) ID	Title	Year	Source title	General conference name	Volume	Issue	Art. No.	Page start	...	Access Type	Sou
0	Khah F.S., Rybkowski Z.K., Ray Pentecost A., S...	57215333932;34868658300;57215305002;7410171017...	Development and testing of an innovative archi...	2019	27th Annual Conference of the International Gr...	Annual Conference of the International Group f...	IGLC1	NaN	NaN	515.0	...	Open Access	Scop
1	Nielsen J., Beard R.W.	56824897200;35562442700;	Relative target estimation using a cascade of ...	2017	30th International Technical Meeting of the Sa...	International Technical Meeting of the Satelli...	4	NaN	NaN	2273.0	...	NaN	Scop

Figure 2. Enriched csv.

After generating the enriched csv, we created the ABox.java class included in the project repo. A copy can also be accessed with the desired naming convention at the root of the repo as: "Group12D-SubSectionB2_ABOXCopy-LeonAlAzazi.java".

This class performs the following actions:

- Loads the TBOX ontology classes, properties, etc. into a Jena Model object.
- Implements a CSV iterator to specifically parse through the rows of the enriched csv that we created (PaperIterator).
- Creates the required triplets of subject, predicate, object for each of the rows. I.e: To create the triplets of author -> writes -> paper, we splitted the list of authors per row, and then for each we create the corresponding triplet to their paper.
- Saves the resulting model in an owl file (final-infered-ontology.owl). A copy can be found at the root of the repo:
    - “Group12D-SubSectionB2andB3_ABOXreconciledTBOX-LeonAlAzazi.owl”

Section B.3: Final Ontology

We decided to do all the connections and reconcile the TBOX with the ABOX programmatically taking advantage of Jena. The ABox.java class mentioned in the previous section achieves this by loading all data into the same model, we later on process the triplets by iterating through the original CSV file and finally we test the inferences using the InfModel class from Jena. The resulting reconciled file is named “final-infered-ontology.owl”, which also can be located at the root of the repo under the naming convention required for the lab as:
“Group12D-SubSectionB2andB3_ABOXreconciledTBOX-LeonAlAzazi.owl”.
As for the inference regime entailment, we can observe how we are saving multiple connections of rdf:type just by the use of inference:

- Paper subclasses (Short Paper, Full Paper, Demo Paper, Poster) are directly inferred to type Paper as well, therefore generating all the connections entailed by a Paper in Figure 1.
- Same scenario with Academic, Leader, Author, Chairman, Editor and Reviewer. By using inference we all know they all are academics. In the case of Chairman and Editor, they are directly associated as Leaders .
- The same happens with Conference and its subclasses (Workshop, Symposium, ExpertGroup or RegularConference).
- An interesting case happens with Conference and Journal, which are inferred to be Venues, and therefore are connected to a Paper or Publication.
- A similar case happens with PublicationList and Conference Proceeding or JournalVolume.

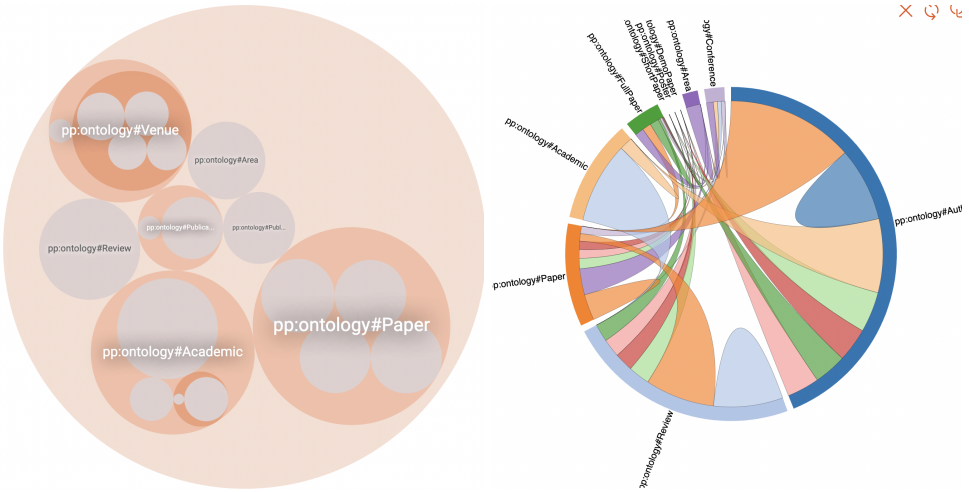

Figure 3. Class Hierarchy and Relationships of the final model in GraphDB.

After loading the resulting owl file onto GraphDB, we obtain the resulting images in Figure 3, where we can observe the relationships between the defined classes, as well as their dependencies and inferences.

We computed a simple query to count the number of classes, and we obtained 56 between those defined by owl and our custom designed ones. Figure 4 displays the first 10.

	class
1	gf:Academic
2	gf:Area
3	gf:Author
4	gf:Chairman
5	gf:Conference
6	gf:ConferenceProceeding
7	gf:DemoPaper
8	gf:Editor
9	gf:ExpertGroup
10	gf:FullPaper

Figure 4. First 10 classes.

For the number of properties, we also obtained 56, with the top 10 shown on Figure 5.

Filter query results — Showing results from 1 to 56 of 56. Query took 0.1s, moments ago.

	property
1	gf:abstract
2	gf:affiliation
3	gf:approves
4	gf:assigned-by
5	gf:compiled-for
6	gf:decision
7	gf:edited-by
8	gf:edition
9	gf:embodies
10	gf:evaluates

Figure 5. First 10 properties.

Additionally, we have computed the total number of instances of the main classes and properties, shown in Figure 6.

Class / Property	**Count**
Paper	564
Short Paper	174
Full Paper	144
Demo Paper	149
Poster	151

Academic	454
Author	434
Reviewer	20
Review	1128
Author -> writes -> Paper	579
abstract	571
Review -> evaluates -> Paper	1128

Figure 6. Counts of some of the basic classes and properties

Section B.4 Querying the ontology

Query 1. Find all authors

Query:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX gf: <http://www.publicationprocess.com/ontology#>
SELECT ?name WHERE {
    ?instance a gf:Author .
    ?instance gf:person-name ?name
} LIMIT 10
```

Result:

	name
1	"Bruening D.A."
2	"_venda V.G."
3	"Svenda V.G."
4	"Loiseau J."
5	"Nascimento M.R."
6	"Baker N.F."
7	"Fulda N."
8	"Johnson J."
9	"Pierce J."
10	"Barker B."

Figure 7. Query 1 results.

Query 2. Find all properties whose domain is Author.

Query:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX gf: <http://www.publicationprocess.com/ontology#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?property WHERE {
```

```
      ?property rdfs:domain gf:Author
}
```

Result:

	property
1	gf:writes

Figure 8. Query 2 results.

Query 3. Find all properties whose domain is either Conference or Journal..

Query:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX gf: <http://www.publicationprocess.com/ontology#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?property WHERE {
    VALUES ?domain { gf:Conference gf:Journal }
    ?property rdfs:domain ?domain .
}
```

Result:

	property
1	gf:led-by
2	gf:edition
3	gf:edited-by
4	gf:impact-factor

Figure 9. Query 3 results.

Query 4. Find all the papers written by a given author that where published in database conferences.

Query:

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX gf: <http://www.publicationprocess.com/ontology#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?paper ?auth ?kw
WHERE {
    ?author rdf:type gf:Author ;
            gf:writes ?paper ;
            gf:person-name ?auth .
```

```
    ?paper rdf:type gf:Paper ;
           gf:submitted-to ?venue .

       ?venue rdf:type gf:Conference .

    ?area rdf:type gf:Area ;
          gf:keyword ?kw .

    FILTER(?kw = " Database systems") .
    FILTER(?auth = "Heath D.") .
}
```

Result:

	paper	auth	kw
1	gf:Semanticstylecreation	"Heath D."	" Database systems"

Figure 10. Query 4 results.