**Knowledge Graphs Lab**

Semantic Data Management

Lab Report

Víctor Diví

Pietro Ferrazzi

May, 2022



# B.1: TBOX Definition

The TBOX has been generated programmatically using the Jena API, the code can be found in the file “Pietro Victor-B1-FerrazziDivi.java” or inside the executable project as the TBox class. All the triples can be found in Turtle format in the file “Pietro Victor-B1-FerrazziDivi.ttl”.

Two TBOX versions can be created, a base version that includes all the classes and properties, as well as some relationships between properties (inverse properties), and an extended version, which includes several additional OWL restrictions, because fuck it, why not. Since the extended version is apparently out of scope for this project, and we did it because it was fun, we have just added the code that generates it. It contains a lot of restrictions related to class disjointness and completeness, cardinality constraints, and property domain restrictions. From now on, every time we refer to the TBOX we mean the base version.

The implemented TBOX use the following OWL elements:

* owl:DatatypeProperty
* owl:ObjectProperty
* owl:inverseOf

Figure 1 below shows a simplified version of the class diagram implemented (in particular, Datatype Properties are not being shown).

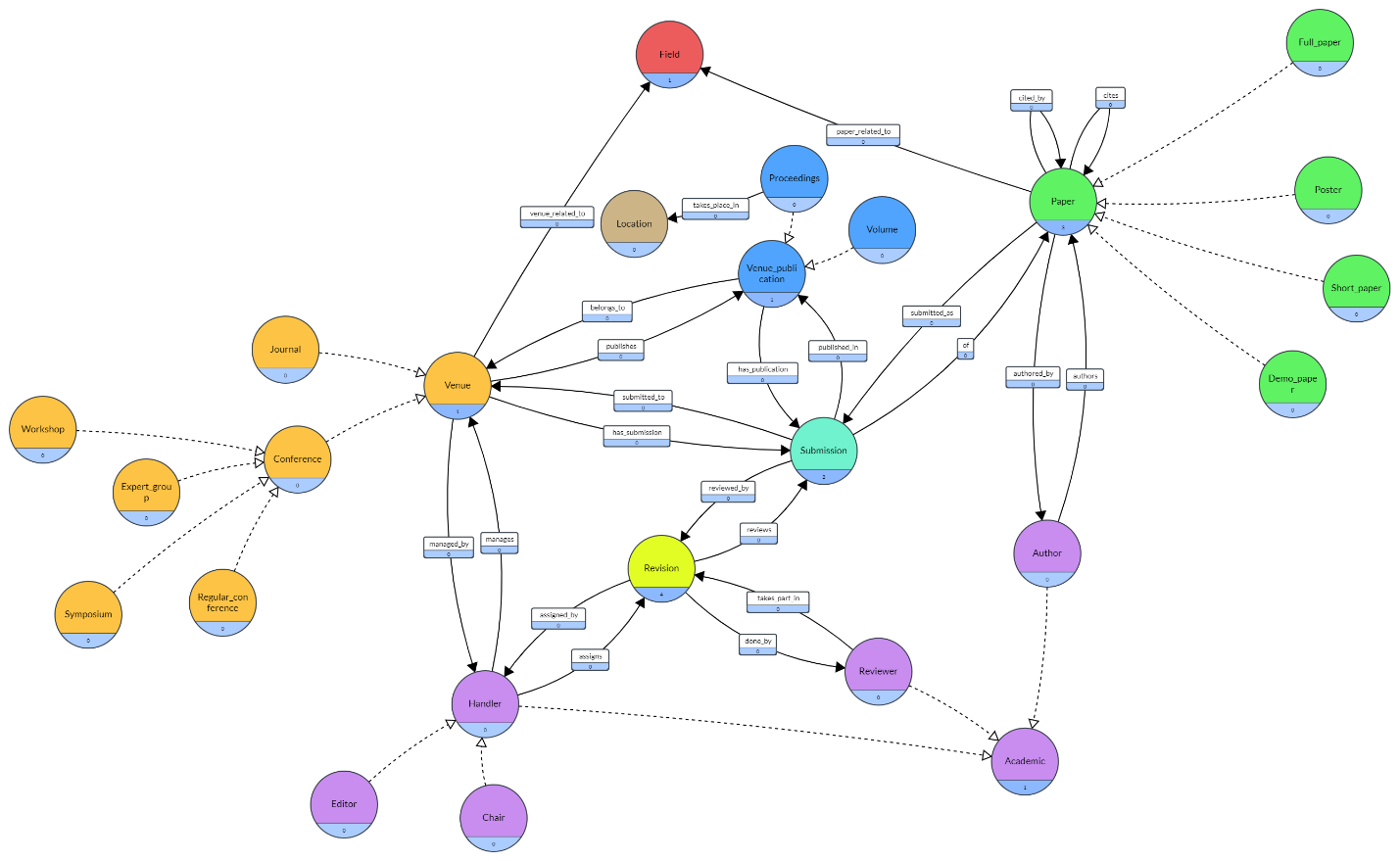


Figure 1 TBOX class diagram (Classes and Object Properties only)

As can be seen by the colors, we have 4 hierarchies: Paper, Academic, Venue, and Venue Publication; and 4 single classes: Revision, Submission, Field and Location. All classes and properties are local, except for Location, which is from the dbpedia ontology (<http://dbpedia.org/Ontology/Location>).

Table 1 lists all the Object Properties shown previously in the diagram, showing also the relationship between them (inverse properties), as well as the theoretical cardinality each property should have, even though the actual restrictions are only present in the extended TBOX.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property | Domain | Range | inverseOf | Cardinality |
| authors | Author | Paper | authored\_by | ≥1 / ≥1 |
| cites | Paper | Paper | cited\_by | - / - |
| submitted\_as | Paper | Submission | of\_paper | - / =1 |
| submitted\_to | Submission | Venue | has\_submission | =1 / - |
| published\_in | Submission | VenuePublication | has\_publication | ≤1 / - |
| belongs\_to | VenuePublication | Venue | publishes | =1 / - |
| manages | Handler | Venue | managed\_by | ≥1 / ≥1 |
| assings | Handler | Revision | assigned\_by | - / ≥1 |
| done\_by | Revision | Reviewer | takes\_part\_in | ≥2 / ≥1 |
| reviews | Revision | Submission | reviewed\_by | ≥1 / =1 |
| paper\_related\_to | Paper | Field | - | ≥1 |
| venue\_related\_to | Venue | Field | - | ≥1 |
| takes\_place\_in | Proceedings | Location | - | ≥1 |

Table 1 List of Object Properties

Table 2 lists all the Datatype Properties of the TBOX that are missing in Figure 1 (following the same color code). The last column the theoretical cardinality each property should have, even though the actual restrictions are only present in the extended TBOX. Note that we accept having multiple values of most textual properties to be able to have translations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Domain | Property Name | Range (XSD) | SubPropertyOf | Cardinality |
| Academic | name | String | rdfs:label | ≥1 |
| Paper | doi | String | - | ≤1 |
| Paper | title | String | rdfs:label | ≥1 |
| Paper | abstract | String | rdfs:comment | ≥1 |
| Submission | submission\_date | Date | - | =1 |
| Submission | submission\_accepted\_date | Date | - | ≤1 |
| Revision | accepted | Boolean | - | ≤1 |
| Revision | review\_text | String | rdfs:comment | - |
| Revision | revision\_date\_start | Date | - | =1 |
| Revision | revision\_date\_end | Date | - | ≤1 |
| Field | keyword | String | rdfs:label | ≥1 |
| Venue | venue\_name | String | rdfs:label | ≥1 |
| VenuePublication | year | Year | - | =1 |

Table 2 List of Datatype Properties

The following assumptions were made in the TBOX design:

* In a revision of a submission, only one review text and final decision is made. I.e., there is one review text and decision per revision, not per reviewer.
* At least one attribute has been added per hierarchy, meaning that each concept is in the domain of at least one attribute, although it may not be directly (e.g., Handler has no attributes, but Academic does).
* Conferences may take place in different locations each year, and therefore that location is related to the Proceedings of that Conference.

The following restrictions should be followed to have a semantically valid model, although violating any of them doesn’t create any inconsistency (all of them except from the last three are covered in the extended TBOX):

* Journals (Conferences) can only publish Volumes (Proceedings). Conversely, Volumes (Proceedings) can only belong to Journals (Conferences).
* The seven main classes (Academic, Venue, VenuePublication, Paper, Submission, Revision and Field) are pairwise disjoint.
* Academic subclasses are complete (i.e., Academic is equivalent to the union of Author, Reviewer and Handler).
* Handler subclasses are complete.
* Paper subclasses are complete and pairwise disjoint.
* Venue subclasses are complete and pairwise disjoint.
* Conference subclasses are complete and pairwise disjoint.
* VenuePublication subclasses are complete and pairwise disjoint.
* Posters can only be submitted/published in Conferences
* A Submission can only be published if it is accepted by the Revision
* A Paper can only be published once

It’s possible that more restrictions could exist in the model, but since they have not been identified, they are considered out of scope for this project.

# B.2: ABOX Definition

# B.3: Ontology Creation

# B.4: Ontology Querying

All following queries can be found in “Pietro Victor-B4-FerrazziDivi.sparql” and make use of the following prefixes:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX fd: <https://ferrazzi.divi/#>

1. Find all Authors.

SELECT ?person WHERE {

?person rdf:type FD:Author

}

1. Find all properties whose domain is Author.

SELECT ?property WHERE {

?property rdfs:domain fd:Author

}

1. Find all properties whose domain is either Conference or Journal.

SELECT ?property WHERE {

{?property rdfs:domain fd:Conference}

UNION

{?property rdfs:domain fd:Journal}

}

1. Find all the papers written by a given author that were published in database conferences

SELECT ?paper WHERE {

?paper rdf:type fd:Paper ; #\*

fd:authored\_by fd:<insert authorId here> ;

fd:submitted\_as ?submission .

?submission rdf:type fd:Submission ; #\*

fd:published\_in ?proceedings .

?proceedings fd:belongs\_to ?conference .

?conference rdf:type fd:Conference ;

fd:venue\_related\_to ?field .

?field rdfs:label “databases” .

}

In this query, the lines marked with an \* are not really needed, since they are redundant with the properties also queried (fd:Paper is the domain of both fd:authored\_by and fd:submitted\_as, and fd:Submission is the range of fd:submitted\_as and domain of fd:published\_in)

# References

**There are no sources in the current document.**