

Software to Transform a Knowledge Graph into an Ontology

ABDOULHOUSSEN Hamza
CRESSANT Killian
ROCHU Hadrien

May 18, 2022



- 1 Introduction
- 2 State of art
 - Reasoner
- 3 From ontology to knowledge graph
 - The chess ontology
 - Visualisation on *Protégé*
 - Ontology on python
 - Conversion into RDF
 - Constraints issues
 - Flags on the script
 - Triples representation
- 4 From knowledge graph to ontology
- 5 Conclusion

Introduction

From this...

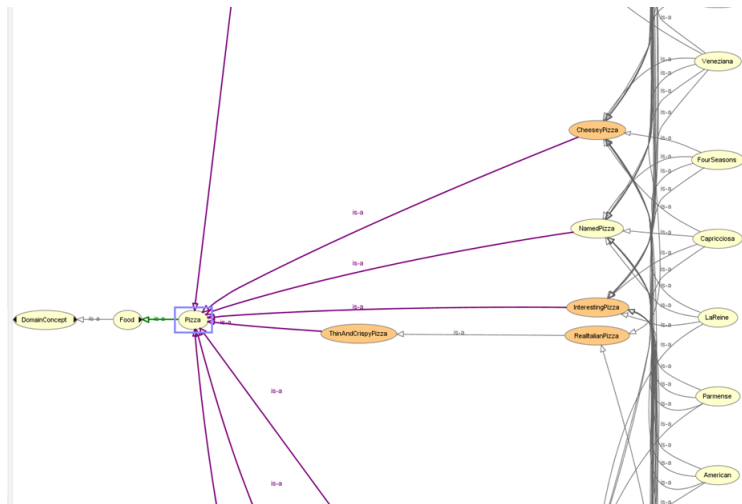


Figure: Part of Pizza ontology

...To this !

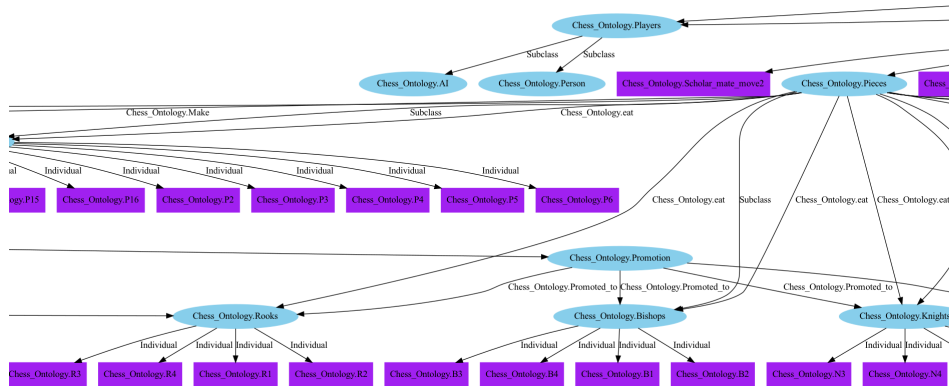


Figure: Part of a Knowledge graph of Chess game

- constraints modelisation

State of art

Ontology: explicit specification of a conceptualization.



Ontology:

- was created by the W3C,
- manipulates knowledge,
- Initially for the web, currently in web semantics, AI biomedical field

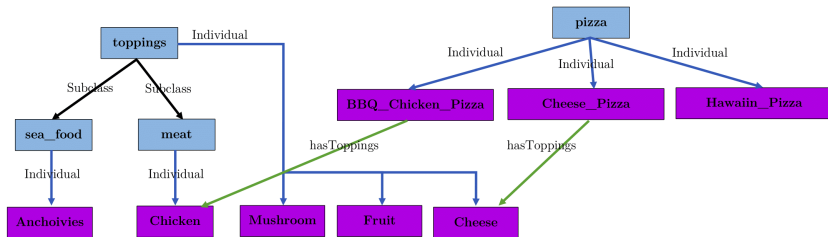


Figure: example pizza_some

$$\begin{aligned}
 & \bigcup_{e \in \text{IndividualOf}(\text{meat})} \text{hasToppings}^{-1}(\{e\}) \\
 &= \text{hasToppings}^{-1}(\{\text{Chicken}\}) \\
 &= \{\text{BBQ_Chicken_Pizza}\}
 \end{aligned} \tag{1}$$

Knowledge graph

S

From ontology to knowledge graph

The chess ontology

Visualisation on *Protégé*

Plugins on *Protégé* for visualisation :

- OWLViz

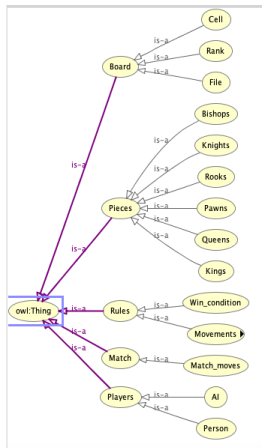
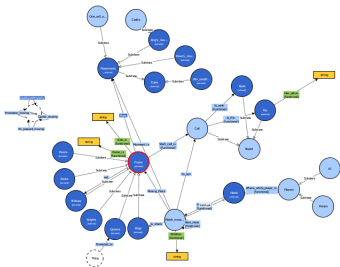


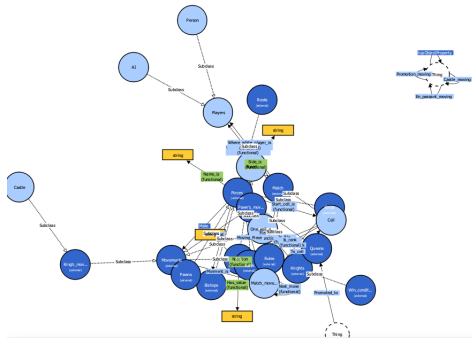
Figure: OWLViz visualisation

Visualisation on *Protégé*

- VOWL



VOWL visualisation



Unorganized VOWL visualisation

Ontology on python

Use python as object-oriented

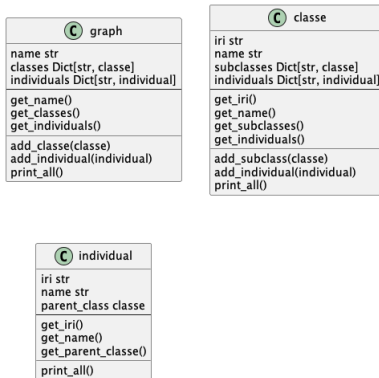


Figure: structure created on python

Ontology on python

```
1      ...
2  Chess_Ontology.Pawns : Pawns
3  Chess_Ontology.Board : Board
4  Chess_Ontology.Rules : Rules
5  Chess_Ontology.Win_condition : Win_condition
6  Chess_Ontology.King's_movement : King's_movement
7  Chess_Ontology.Pawn's_movement : Pawn's_movement
8  Chess_Ontology.AI : AI
9  Chess_Ontology.Castle : Castle
10 Chess_Ontology.Person : Person
11 23 classes
```

print_all output for the chess ontology

RDF template

```
1 <rdf:Description rdf:about="${subject}">
2   <ex:${predicate}>
3     <rdf:Description rdf:about="${object}"/>
4   </ex:${predicate}>
5 </rdf:Description>
```

Conversion into RDF

```
1 <owl:ObjectProperty rdf:about="...#First_move">
2   <rdf:type
3     ↪ rdf:resource="...#FunctionalProperty"/>
4   <rdfs:domain rdf:resource="...#Match"/>
5   <rdfs:range rdf:resource="...#Match_moves"/>
6 </owl:ObjectProperty>
```

Example in OWL

```
1 ('Chess_Ontology.Match',
   ↪ 'Chess_Ontology.First_move',
   ↪ 'Chess_Ontology.Match_moves')
```

python triple

Conversion into RDF

```
1 <rdf:Description rdf:about="Chess_Ontology.Match">
2   <ex:Chess_Ontology.First_move>
3     <rdf:Description
4       ↪rdf:about="Chess_Ontology.Match_moves"/>
5   </ex:Chess_Ontology.First_move>
  </rdf:Description>
```

output in RDF

Constraints issues

Format proposed to include constraints

- For SpicyPizza

```
1      Pizza
2          and (hasTopping some
3              (PizzaTopping
4                  and (hasSpiciness some Hot)))
```

Restriction in the ontology

"pizza.Pizza & pizza.hasTopping.some(pizza.PizzaTopping &
pizza.hasSpiciness.some(pizza.Hot))"

The equivalent element in the RDF file

Flags on the script

DESCRIPTION

- i [required] [need argument] it is to add the owl input file path
by default, the output is the input with '_output'
ex : -i "resource/pizza.owl"
- o [need argument] add the output file path
ex : -o "output/pizza"
- O [need argument] same as -o but overwrite the file if already exists
ex : -O "output/pizza"
- p To print the triple added
- s To create standard triple without restrictions
- r To add reasoner before
- kr [need argument] To keep the ontology made after the reasoner
the argument is the path of the new file
ex : -kr "reasoner/pizza.owl"
- Kr [need argument] same as -kr but overwrite the file if already exists
ex : -Kr "reasoner/pizza.owl"

Figure: Description of the script command

Triples representation

Triples representation from python using dot language

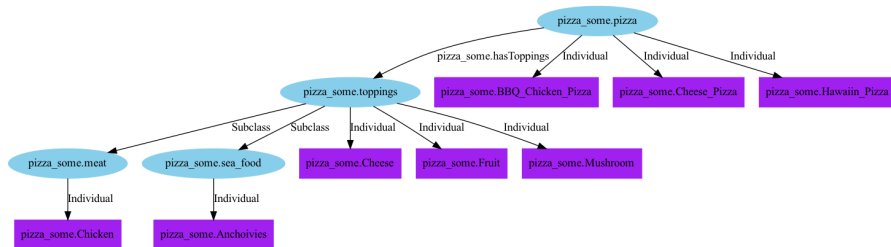


Figure: pizza_some representation

Triples representation

The representation is not really adapted



Figure: Chess_Ontology representation

From knowledge graph to ontology

Conclusion