Rapport — Ontologie (OWL / Turtle)

Exercice : modéliser des cours, devoirs et enseignants

Étudiant : Kadouaoui Abdelmoiz

Outil : Protégé Desktop (OWL 2) • Export : .ttl (Turtle)

Année / date : 2025-12-31

1. Énoncé (texte à transformer en ontologie)

There are courses and laboratory courses.

Homeworks are part of courses.

Courses are organized by teachers.

Teachers are either professors or assistants.

Professors teach courses while assistants only teach laboratory courses.

2. Modèle conceptuel

Concepts et relations extraits du texte

Concepts (classes)

•Course (Cours)

•LaboratoryCourse (Cours de laboratoire) ⟶ sous-classe de Course

•Homework (Devoir)

•Teacher (Enseignant)

•Professor (Professeur) ⟶ sous-classe de Teacher

•Assistant (Assistant) ⟶ sous-classe de Teacher

Relations (propriétés objet)

•organizedBy : *Course* ⟶ *Teacher* (un cours est organisé par un enseignant)

•partOf : *Homework* ⟶ *Course* (un devoir fait partie d’un cours)

•hasHomework : *Course* ⟶ *Homework* (un cours a des devoirs) — inverse de *partOf*

•teaches : *Teacher* ⟶ *Course* (un enseignant enseigne un cours)

3. Formalisation OWL (axiomes)

Traduction directe des phrases en contraintes

- $\text{LaboratoryCourse} \sqsubseteq \text{Course}$
- $\text{Homework} \sqsubseteq \text{partOf some Course}$
- $\text{Course} \sqsubseteq \text{organizedBy some Teacher}$
- $\text{Teacher} \equiv \text{Professor} \sqcup \text{Assistant}$
- $\text{Professor} \sqcap \text{Assistant} \sqsubseteq \bot$ (disjoint)
- $\text{Professor} \sqsubseteq \text{teaches some Course}$
- $\text{Assistant} \sqsubseteq \text{teaches only LaboratoryCourse}$

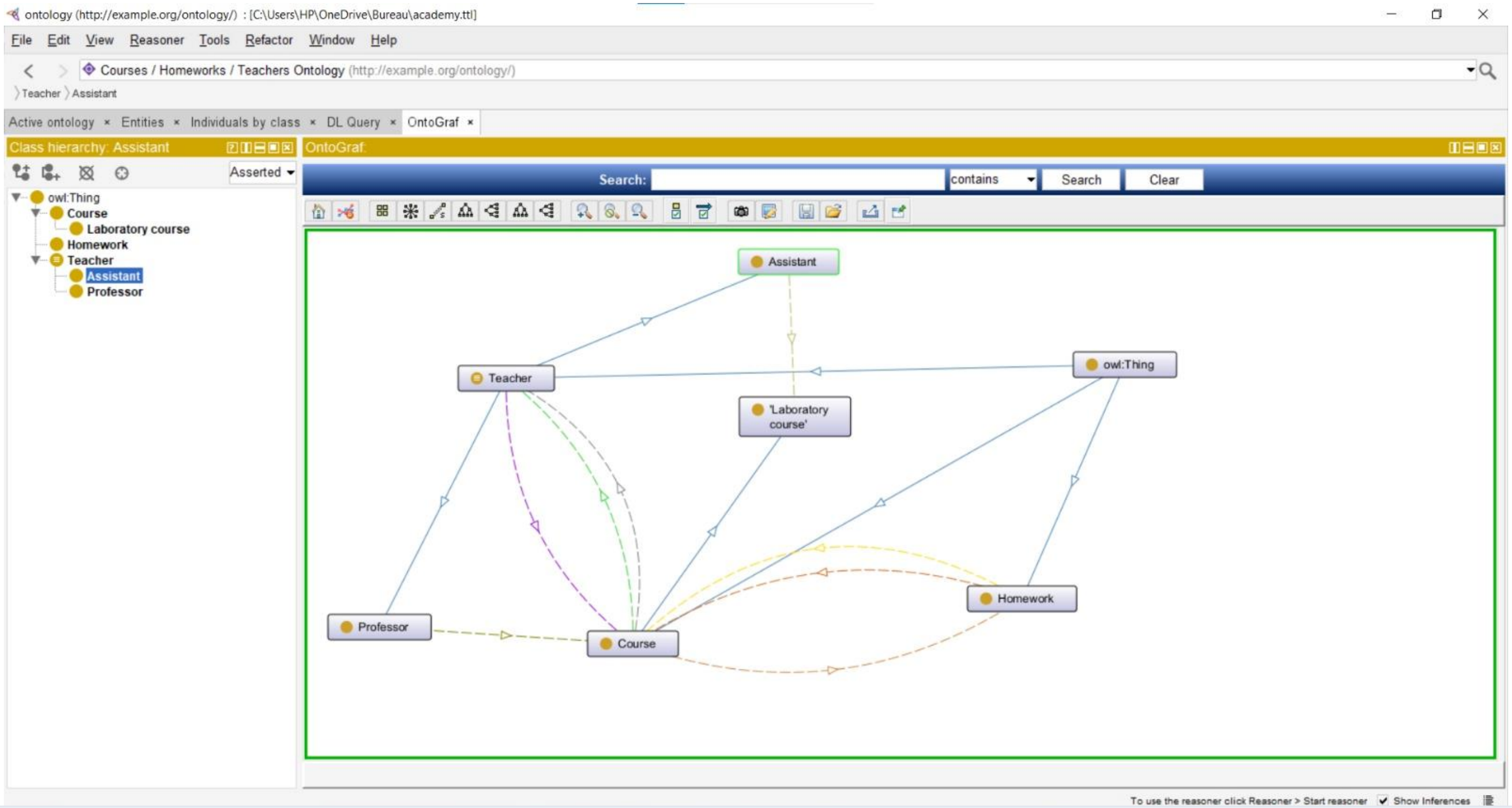
4. Implémentation (Turtle / .ttl)

```
@prefix ex:   <http://example.org/ontology/> .
@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#> .
ex: a owl:Ontology ;
  rdfs:label "Courses / Homeworks / Teachers Ontology"@en .
ex:Course a owl:Class ;
  rdfs:label "Course"@en ;
  # Courses are organized by teachers.
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty ex:organizedBy ;
    owl:someValuesFrom ex:Teacher
  ] .
ex:LaboratoryCourse a owl:Class ;
  rdfs:label "Laboratory course"@en ;
  rdfs:subClassOf ex:Course .
ex:Homework a owl:Class ;
  rdfs:label "Homework"@en ;
  # Homeworks are part of courses.
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty ex:partOf ;
    owl:someValuesFrom ex:Course
  ] .
ex:Teacher a owl:Class ;
  rdfs:label "Teacher"@en ;
  # Teachers are either professors or assistants.
  owl:equivalentClass [
    a owl:Class ;
    owl:unionOf ( ex:Professor ex:Assistant )
  ] .
ex:Professor a owl:Class ;
  rdfs:label "Professor"@en ;
  rdfs:subClassOf ex:Teacher ;
  # Professors teach courses.
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty ex:teaches ;
    owl:someValuesFrom ex:Course
  ] .
```

```
ex:Assistant a owl:Class ;
  rdfs:label "Assistant"@en ;
  rdfs:subClassOf ex:Teacher ;
  # Assistants only teach laboratory courses.
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty ex:teaches ;
    owl:allValuesFrom ex:LaboratoryCourse
  ] .
# Exclusive either/or (optional but consistent with "either ...
or")
ex:Professor owl:disjointWith ex:Assistant .
########################################
# Object properties
########################################
ex:partOf a owl:ObjectProperty ;
  rdfs:label "part of"@en ;
  rdfs:domain ex:Homework ;
  rdfs:range  ex:Course .
ex:hasHomework a owl:ObjectProperty ;
  rdfs:label "has homework"@en ;
  rdfs:domain ex:Course ;
  rdfs:range  ex:Homework ;
  owl:inverseOf ex:partOf .
ex:organizedBy a owl:ObjectProperty ;
  rdfs:label "organized by"@en ;
  rdfs:domain ex:Course ;
  rdfs:range  ex:Teacher .
ex:teaches a owl:ObjectProperty ;
  rdfs:label "teaches"@en ;
  rdfs:domain ex:Teacher ;
  rdfs:range  ex:Course .
```

6. Visualisation OntoGraf

Graphe généré dans Protégé



Lecture : traits pleins = hiérarchie (subClassOf) ; pointillés = restrictions OWL (some/only) ; arcs colorés = propriétés.

8. Conclusion

- Le texte a été traduit en une ontologie OWL : classes, propriétés et axiomes.
- Les contraintes clés sont formalisées : (Teacher = Professor ⊔ Assistant) et (Assistant teaches only LaboratoryCourse).
- Le format Turtle (.ttl) permet un export compact et standard.
- La visualisation OntoGraf et le reasoner (HermiT/Pellet) servent à valider cohérence et inférences.