

Protégé 4.1

UNIK9710

Protégé 3x vs 4x

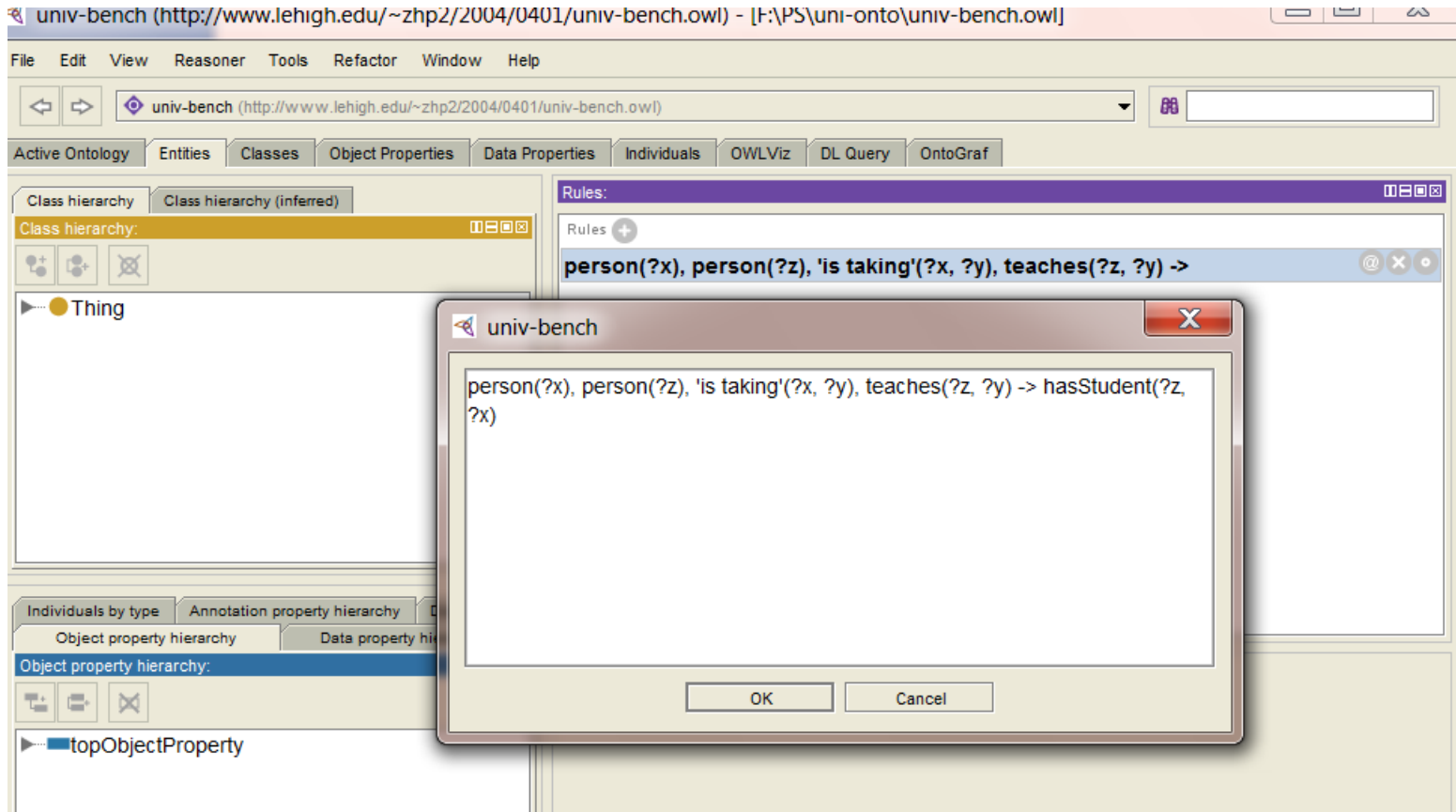
	Protégé 3x	Protégé 4x
Developers	<ul style="list-style-type: none">• cannot cleanly break away from RDF	<ul style="list-style-type: none">• purely OWL applications• larger developer community• supports OWL 2.0
Users	<ul style="list-style-type: none">• features of the platform are well known• the behaviour is predictable• many plugins available that are not currently available for P4.x• SWRL support	<ul style="list-style-type: none">• purely OWL ontologies or needing OWL2.0 features

Protégé 3x vs 4x

Feature	Protégé 3x	Protégé 4x
OWL	OWL 1.0 OWL and RDF(S) support	OWL 2.0 Pure OWL framework
API	Protege-OWL API	OWL API
SPARQL	Yes	Not yet
SWRL	SWRLTab	through a basic editor and pellet for reasoning
meta-modeling	allowing OWL Full	No OWL Full
Reasoner	through HTTP DIG interface or direct connection	Direct connection to FaCT++, Pellet and other DL reasoners
Multi-user Support	using the client-server version of Protege	None (Protege Client-Server has not been migrated yet)
Database Storage Model	Ability to store ontologies in a database provided by the JDBC database back-end	None (database back-end has not been migrated yet)

SWRL, Pellet and Protégé 4.1

- SWRL is supported through a basic editor and pellet for reasoning.
 - Window → Views → Ontology views → Rules



univ-bench (http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl)

Active Ontology | Entities | Classes | Object Properties | Data Properties | Individuals | OWLViz | DL Query | OntoGraf

Class hierarchy (inferred)

Class hierarchy

Class hierarchy: person

- Thing
 - Entity
 - Work
 - organization
 - person
 - publication
 - schedule

Members list | Members list (inferred)

Members list: JosefNoll

- ◆ AndreasPrinz
- ◆ DafferiantoTrinugroho
- ◆ JosefNoll
- ◆ Mauricelsabwe
- ◆ RolfNossum
- ◆ TrinhNguyen

Rules:

Rules +

person(?x), person(?z), 'is taking'(?x, ?y), teaches(?z, ?y) -> hasStudent(?z, ?x)

Annotations | Usage | Rules

Annotations: JosefNoll

Annotations +

Description: JosefNoll

Types +

- 'full professor'
- person

Same individuals +

Different individuals +

Property assertions: JosefNoll

Object property assertions +

- 'has a doctoral degree from' University_of_Bochum
- 'Works For' University_of_Oslo
- 'Works For' University_Graduate_Center
- teaches UNIK9710

Data property assertions +

- 'can be reached at' "josef@unik.no"^^string
- 'telephone number' "+4790838066"^^string
- title "Professor"^^string

Negative object property assertions +

No Reasoner set. Select a reasoner from the Reasoner menu Show Inferences

Class hierarchy (Inferred)

Class hierarchy: person

- Thing
 - Entity
 - Work
 - organization
 - person
 - publication
 - schedule

Members list: JosefNoll

- AndreasPrinz
- DafferiantoTrinugroho
- JosefNoll
- Mauricelsabwe
- RolfNossum
- TrinhNguyen

Rules

```
person(?x), person(?z), 'is taking'(?x, ?y), teaches(?z, ?y) -> hasStudent(?z, ?x)
```

Annotations: JosefNoll

Description: JosefNoll

Types

- 'full professor'
- person

Same individuals

Different individuals

Property assertions: JosefNoll

Object property assertions

- 'has a doctoral degree from' University_of_Bochum
- 'Works For' University_of_Oslo
- 'Works For' University_Graduate_Center
- teaches UNIK9710
- hasStudent Mauricelsabwe
- hasStudent TrinhNguyen
- hasStudent DafferiantoTrinugroho
- 'member of' University_Graduate_Center
- 'member of' University_of_Oslo
- 'has a degree from' University_of_Bochum

Data property assertions

- 'can be reached at' "josef@unik.no"^^string
- 'telephone number' "+4790838066"^^string
- 'title' "Professor"^^string

Negative object property assertions

How to retract or remove information from an ontology?

- Like OWL, SWRL supports **monotonic** inference only. Hence, SWRL rules **cannot be** used to modify existing information in an ontology. For this reason, SWRL rules **can not** retract or remove information from an ontology.

`Driver(?d) ^ hasAge(?d, ?age) ^ swrlb:greaterThan(?age, 25) -> isInsurable(?d, true)`

2 cases:

- If `isInsurable` is functional: OWL reasoner will indicate an inconsistency when applied to the resulting ontology
- Otherwise, `isInsurable` has 2 values