DATA SCIENCE MASTER

SEMANTIC KNOWLEDGE REPRESENTATION

Ontology Formalisation and Protégé

Mounira Harzallah

mounira.harzallah@univ-nantes.fr

Tél: 02 28 09 21 27

Plan

- Formalisation of ontology with description logics (DLs)
- Reasoning with DL on ontology
- Formalisation of ontology with OWL/Protégé

Description Logics

Description logics are logical formalisms for :

- Defining concepts and their relations/roles (**Terminologies)**
- Specifying properties of individuals (**Assertions)**

Main advantages:

- **Well-defined semantic**: *defined* by interpreting with subsets of individuals
- **Flexibility:** Formalisation of large kinds of situations

Today DLs represent one of the main language of knowledge representation, with applications in a number of areas of computer sciences

Semantic web: DLs are a foundation of ontology languages (OWL) which have been standardised by W3C group

Software engineering: one can translate UML diagrams into DL

Database theory: one can translate E/R-models into DL

DL constructs for ontology formalisation

Atomic Concepts denote subsets of the domain that are not defined (primitive concepts). All other concepts are defined in terms of the primitive concepts

Example: Human, Animal, Plant, Female

Roles denote **binary relations on the domain**

Example: HasChild, HasCost, HasPart, IsAuthorOf

Concepts constructors denote operations for combining and building complex concepts.

Example: Conjunction ($\sqcap$) , Disjunction ($\sqcup$)

Role constructors denotes operations for building complex roles

Example: transitive closure (*) and R* is a transitive closure on R

$R^*(x)=y \iff$ It exists z where $R(x)= z$ and $R^*(z)= y$

DL Constructs for ontology formalisation

Example

- Atomic/primitive concepts: Human, Animal, Plant, Female

- Role: HasChild, Marryto

- Complex Concept s :

  $\text{Animal} \sqcap \text{Plant}$

  $\text{Human} \sqcap \exists \text{HasChild}.\top$

  $\text{Human} \sqcap \exists \text{HasChild}.\top \sqcap \forall \text{HasChild}.\text{Female}$

  $\text{Human} \sqcap \neg \text{Female}$

- Complex Role: (HasChild)*= HasDescendant

HasDescendant (x)= HasChild*(x)=y<=>

It exists z where Haschild(x)= z and HasDescendant(z)= y

DL goals

DL is a language of description :

• How to constitute concepts and roles,

• Mechanisms to **specify knowledge on concepts and roles, using** Terminological axioms i.e .**TBox :** collection of definitions of concepts and their relations

• Mechanisms to **specify the properties of objects/individuals** using Assertional axioms i.e. **Abox:** specification of individual properties , where the properties are those defined in the TBox

• A set of mechanisms for reasoning

DL goals

Example

- Complex concepts:
  $Human \sqcap Male \sqcap \exists HasChild \sqcap \forall\ HasChild.Doctor$
- Tbox
  $T=\{Father \equiv Human \sqcap Male \sqcap \exists hasChild,$
  $HappyFather \subseteq Father \sqcap \forall\ HasChild.Doctor\}$

- Abox
  $A= \{HappyFather(Jean), (Jean, Marie) \in Haschild^I\}$

- Example of reasoning
  $T \models HappyFather \subseteq \exists HasChild.Doctor$
  $T \cup A \models Doctor(Marie)$

Semantic of DL constructs

- An interpretation I is a pair (F, Δ), where
  - Δ is a non-empty set
  - Fc : AtomicConcepts → Pow(Δ)
  - Fr : Roles → Pow(Δx Δ)

- Fc(C) = C^I is the interpretation of C

The set of individuals of C is one interpretation of C

The set of ordered pairs of individuals of R is one interpretation of R

Interpretation is a mean to explain

- the semantic of DL constructs
- the reasoning made

Interpretation and Reasoning

- An interpretation I satisfies a Tbox, if it satisfies every axioms in the Tbox

  denoted by $I \models$ Tbox

- **Tbox $\models$ Ai => all the** interpretations of the Tbox should satisfy Ai, Ai is an axiom

  If $\exists$ an interpretation of **Tbox that dosent' satisfy Ai => Tbox $\not\models$ Ai**

Example

Tbox= {$C \sqcup D \equiv B$}, I1 : $C^{I1}=\{1, 2, 3\}$, $D^{I1}=\{1, 4\}$, $B^{I1}=\{1, 2, 3,4\}$

$\{1, 2, 3\} \cup \{1, 4\} = \{1, 2, 3,4\}$ **=> I1** $\models$ Tbox

A1 : C $\sqcap D \equiv \bot$, Tbox $\not\models$ A1 because $C^{I1} \cap D^{I1} = \{1\}$ is not empty

Semantic of DL constructs

Construct	Syntax	Example	Semantics
atomic concept	A	Doctor	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
atomic role	P	hasChild	$P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
atomic negation	$\neg A$	¬Doctor	$\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$
conjunction	$C \sqcap D$	Hum ⊓ Male	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
(unqual.) exist. res.	$\exists R$	∃hasChild	$\{ a \mid \exists b. (a, b) \in R^{\mathcal{I}} \}$
value restriction	$\forall R.C$	∀hasChild.Male	$\{a \mid \forall b. (a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$
bottom	$\bot$		$\emptyset$

Semantic of DL constructs

Construct	Syntax	Semantics
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
top	$\top$	$\Delta^{\mathcal{I}}$
qual. exist. res.	$\exists R.C$	$\{ a \mid \exists b. (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}} \}$
(full) negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
number	$(\geq k\, R)$	$\{ a \mid \#\{b \mid (a,b) \in R^{\mathcal{I}}\} \geq k \}$
restrictions	$(\leq k\, R)$	$\{ a \mid \#\{b \mid (a,b) \in R^{\mathcal{I}}\} \leq k \}$
qual. number	$(\geq k\, R.C)$	$\{ a \mid \#\{b \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \geq k \}$
restrictions	$(\leq k\, R.C)$	$\{ a \mid \#\{b \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \leq k \}$
inverse role	R^{-}	$\{ (a,b) \mid (b,a) \in R^{\mathcal{I}} \}$

Semantic of LD constructs

$C \sqcup D = B \Rightarrow C \sqsubseteq B$

Equivalences of the Negation

$\neg(C \sqcap D) = \neg C \sqcup \neg D$

$\neg(C \sqcup D) = \neg C \sqcap \neg D$

$\neg(\exists R.C) = \forall R.\neg C$

$\neg(\forall R.C) = \exists R.\neg C$

$\neg\neg C = C$

For reasoning, we can use interpretations and/or associate a first-order formula with free-variable x, y

For an atomic concept A, $A(x) = A(x)$

For a role R, we use a binary relation symbol $R(x, y)$

$\top(x) = P(x) \lor \neg P(x)$ and $\bot(x) = P(x) \land \neg P(x)$

$C \sqcap D(x) = C(x) \land D(x)$

$C \sqcup D(x) = C(x) \lor D(x)$

$\exists R.C(x) = \exists y(R(x, y) \land C(y))$

$\forall R.C(x) = \forall y(R(x, y) \rightarrow C(y))$

Exercices

1. Write "Course with at most 20 participants, all of which are master or Ph.D students", (primitive concepts: Course, MasterStudent, PhDStudent, role: HasParticipant).

2. $T=\{Father \equiv Human \sqcap Male \sqcap \exists hasChild,$

   $HappyFather \subseteq Father \sqcap \forall\ HasChild.Doctor\}$

   $A= \{HappyFather(Jean), Haschild(Jean, Marie)\}$

   To infer using interpratations :

   $$T \models HappyFather \subseteq \exists HasChild.Doctor$$

   $$T \cup A \models Doctor(Marie)$$

Exercices

$T = \{Father \equiv Human \sqcap Male \sqcap \exists hasChild,$

$HappyFather \subseteq Father \sqcap \forall HasChild.Doctor\}$

$A = \{HappyFather(Jean), Haschild(Jean, Marie)\}$

To infer using interprataions :

$$T \models HappyFather \subseteq \exists HasChild.Doctor$$

$$T \cup A \models Doctor(Marie)$$

$HappyFather \subseteq \exists hasChild \sqcap \forall HasChild.Doctor$

$\rightarrow (\forall x, Happyfather(x) \rightarrow (\exists y, HasChild(x,y)) \wedge (\forall y\ HasChild(x,y) \rightarrow Doctor(y))$

$\rightarrow (\forall x, Happyfather(x) \rightarrow \exists y, HasChild(x,y) \wedge Doctor(y))$

$\rightarrow HappyFather \subseteq \exists hasChild.Doctor$

$HappyFather(Jean) \rightarrow \forall HasChild.Doctor(Jean) \wedge$

$Haschild(Jean, Marie) \wedge \forall HasChild.Doctor(Jean) \rightarrow Doctor(Marie)$

How to formalise ontology

Primitive Concepts

Human, Male

Complex Concept definitions

Man $\subseteq$ Human

Man $\equiv$ (Human $\sqcap$ Male)

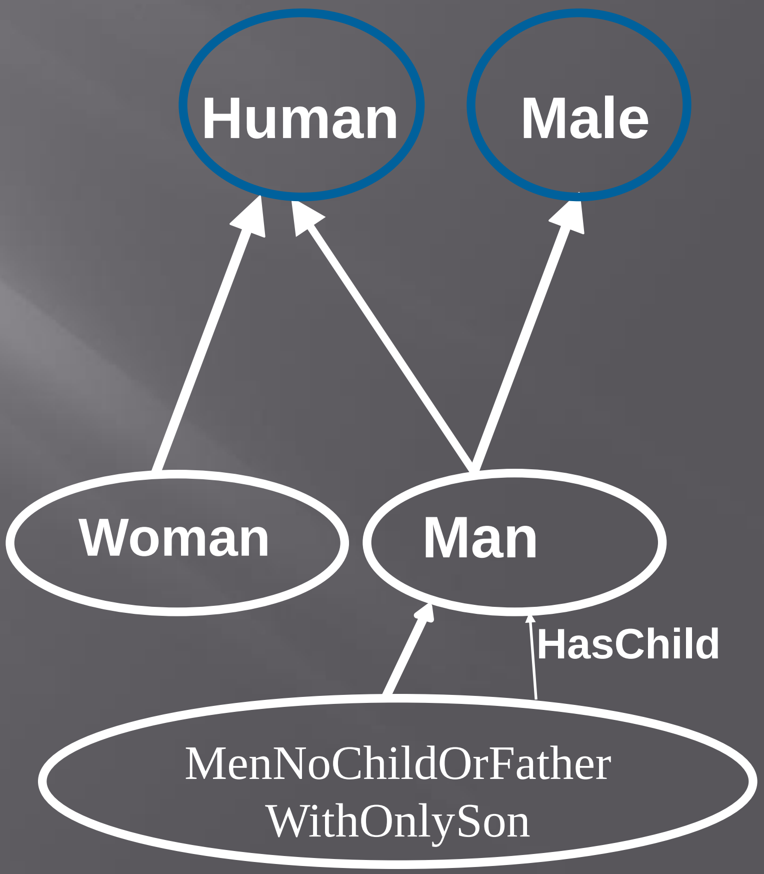
Woman$\equiv$ Human $\sqcap$ ($\neg$ Male)

Father $\subseteq$ $\exists$HasChild

Father $\equiv$ (Man $\sqcap$ $\exists$ HasChild)

MenNoChildOrFatherWithOnlySon $\equiv$

(Man $\sqcap$ $\forall$ HasChild.Man)



Reasoning on ontology

- Classification: classification of a concept within a taxonomy
- Improving a classification: a better classification of a concept)
- Inference of new properties of concepts, individuals or roles.
- Determination of unsatisfiability
- Identification of logical inconsistencies

Reasoning. Classification of concepts

A concept C should be classed under the concept C1 that is the most close to it.

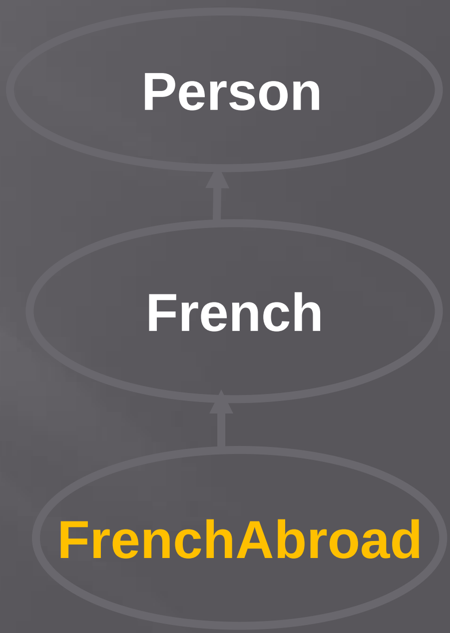
C1 is the most close to C, if C is a sub class of C1 where there is not another concept C2 (distinct of C1) where $C \subseteq C2 \subseteq C1$

Person

CitizenOf(Person, County)

LivesIn(Person, Country)

French $\equiv$ (Person $\sqcap$ $\exists$ Citizen.{France})



FrenchAbroad is a new concept defined as

$\equiv$ (Person $\sqcap$ ($\exists$ Citizen.{France}
$\sqcap$ $\exists$ LivesIn.$\neg${France})

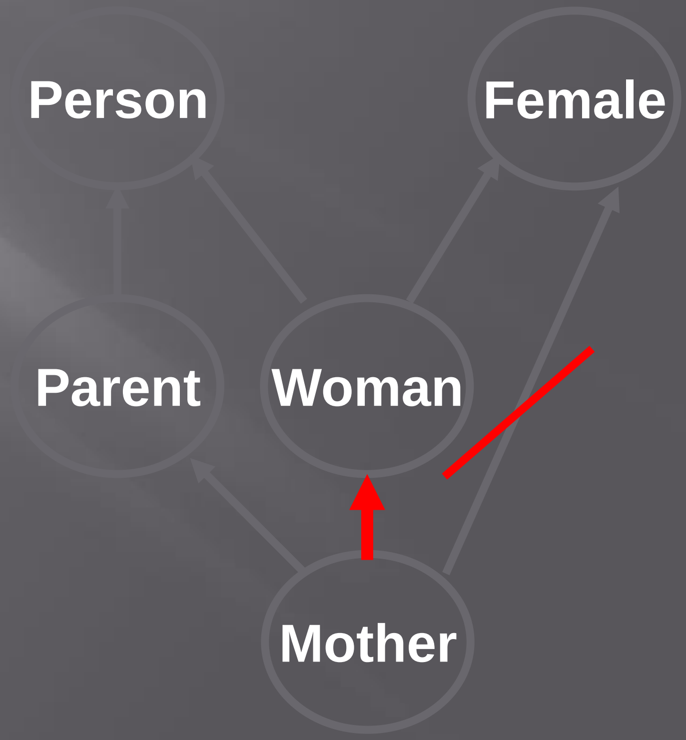
Then FrenchAbroad $\subseteq$ French

Reasoning. Taxonomy improving

- Person
- Parent ≡ (Person ⊓ ∃ HasChild.Person)
- Woman ≡ (Person ⊓ Female)
- Mother ≡ (Parent ⊓ Female)

In the taxonomy Mother is-a Female

But we can deduce that
Mother is-a Woman



Reasoning. Logical inconsistency

Ontology inconsistency is when an ontology contains logical contradictions for which a model does not exist

Examples of ontology inconsistencies

- $O \models (Ci \sqcap Cj = \perp$ and $Ci^I \sqcap Cj^I = \{c\})$
- $O \models Ci = \neg Ci$

Reasoning. Unsatisfability

- **Satisfiability. A concept C is satisfiable if there exists an** interpretation I such that $C^I \neq \emptyset$

The following complex concept are satisfiables

- Parent $\equiv$ (Person $\sqcap$ $\exists$hasChild.Person)
- Woman$\equiv$ (Person $\sqcap$ Female)
- Man $\equiv$ (Person $\sqcap$ Male)
- Mother $\equiv$ (Female $\sqcap$ Parent)
- NonHumain $\equiv$ $\neg$ Woman $\sqcap$ $\neg$ Man
- OtheMother $\equiv$ (NonHumain $\sqcap$ Mother)

We can deduce that there is not a non empty set that can interpret OtherMother

OtherMother is Unsatisfiable

OWL (Ontology Web Language)

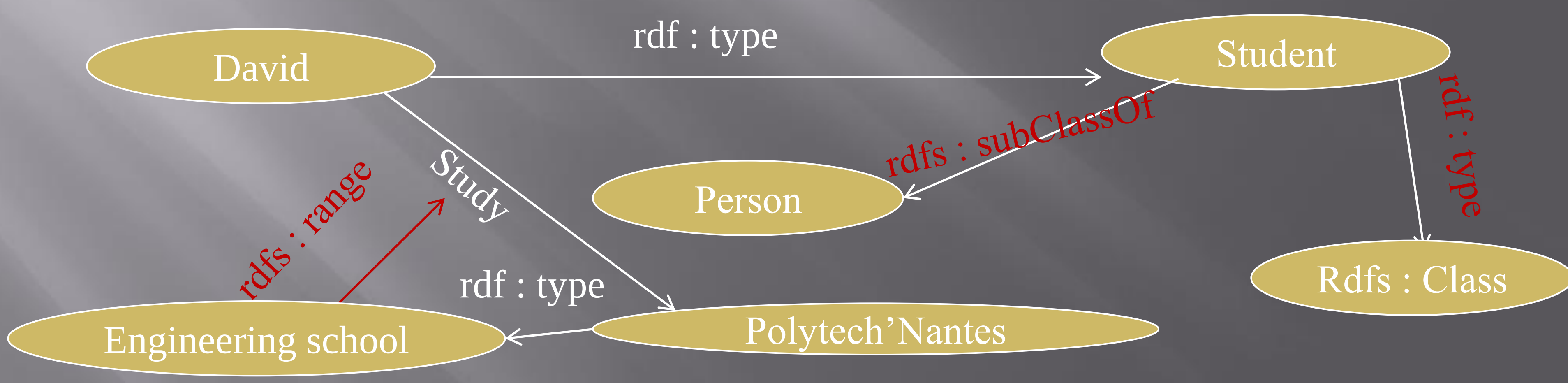
OWL is developed based on DL. It defines

- Hierarchies of Classes
- Hierarchies of relations (or properties)
- Axioms on concepts and relations

OWLs

RDF/RDFs are languages to describe ressources/data on the web

RDF defines the web ressources as triplets of (objet predicat objet)
RDFs : structures the ressources of RDF



RDF/RDFs are not suffisant to specify and formalize ontologies

OWL languages

- OWL is an extension of RDF/RDFs
- OWL langages: OWL Lite, OWL DL, OWL Full
- OWL languages support a variety of syntaxes.

OWL vs LD

OWL contructor	DL constructor	Example
intersectionOf	$C_1 \sqcap \cdots \sqcap C_n$	Human $\sqcap$ Male
unionOf	$C_1 \sqcup \cdots \sqcup C_n$	Doctor $\sqcup$ Lawyer
complementOf	$\neg C$	$\neg$Male
oneOf	$\{a_1\} \sqcup \cdots \sqcup \{a_n\}$	{john} $\sqcup$ {mary}
allValuesFrom	$\forall P.C$	$\forall$hasChild.Doctor
someValuesFrom	$\exists P.C$	$\exists$hasChild.Lawyer
maxCardinality	$(\leq n\, P)$	($\leq$ 1 hasChild)
minCardinality	$(\geq n\, P)$	($\geq$ 2 hasChild)

OWL vs LD

OWL axiom	DL syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Man $\sqsubseteq$ $\neg$Female
sameIndividualAs	$\{a_1\} \equiv \{a_2\}$	{presBush} $\equiv$ {G.W.Bush}
differentFrom	$\{a_1\} \sqsubseteq \neg\{a_2\}$	{john} $\sqsubseteq$ $\neg${peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	hasCost $\equiv$ hasPrice
inverseOf	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$

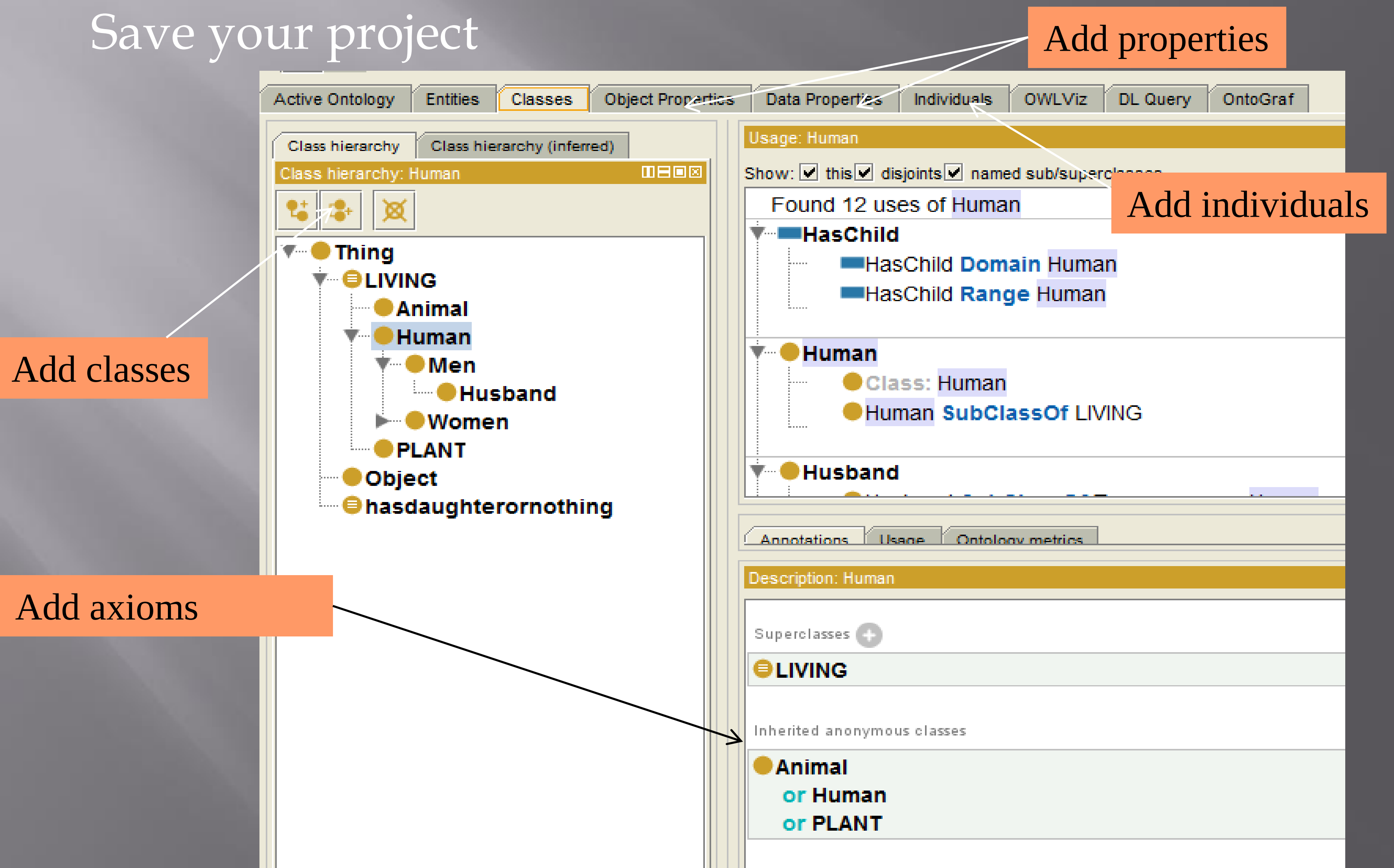
Protégé : ontology eidtor

- Developed with Java
- Specific language, but based on OWL (and then on DL)
- With protégé, we can edit an ontology in OWL (in several syntaxes)
- Possibility to integrate plug-in by adding applications on the ontology
- Ontology is defined as a hierarchy of classes (is-a), slots (relations), instances and axioms

Protege et OWL

Create a new OWL ontology → OWL/XML

Save your project



Definition of classes/subClasses and Instances with OWL

The relation Class/subClass is defined by the definition of a hierarchy

```
<owl:Class rdf:ID=“Woman">
      <rdfs:subClassOf rdf:resource="Human"/>
 </owl:Class>
 <owl:Class rdf:ID=“man">
      <rdfs:subClassOf rdf:resource="Human"/>
   </owl:Class>
```

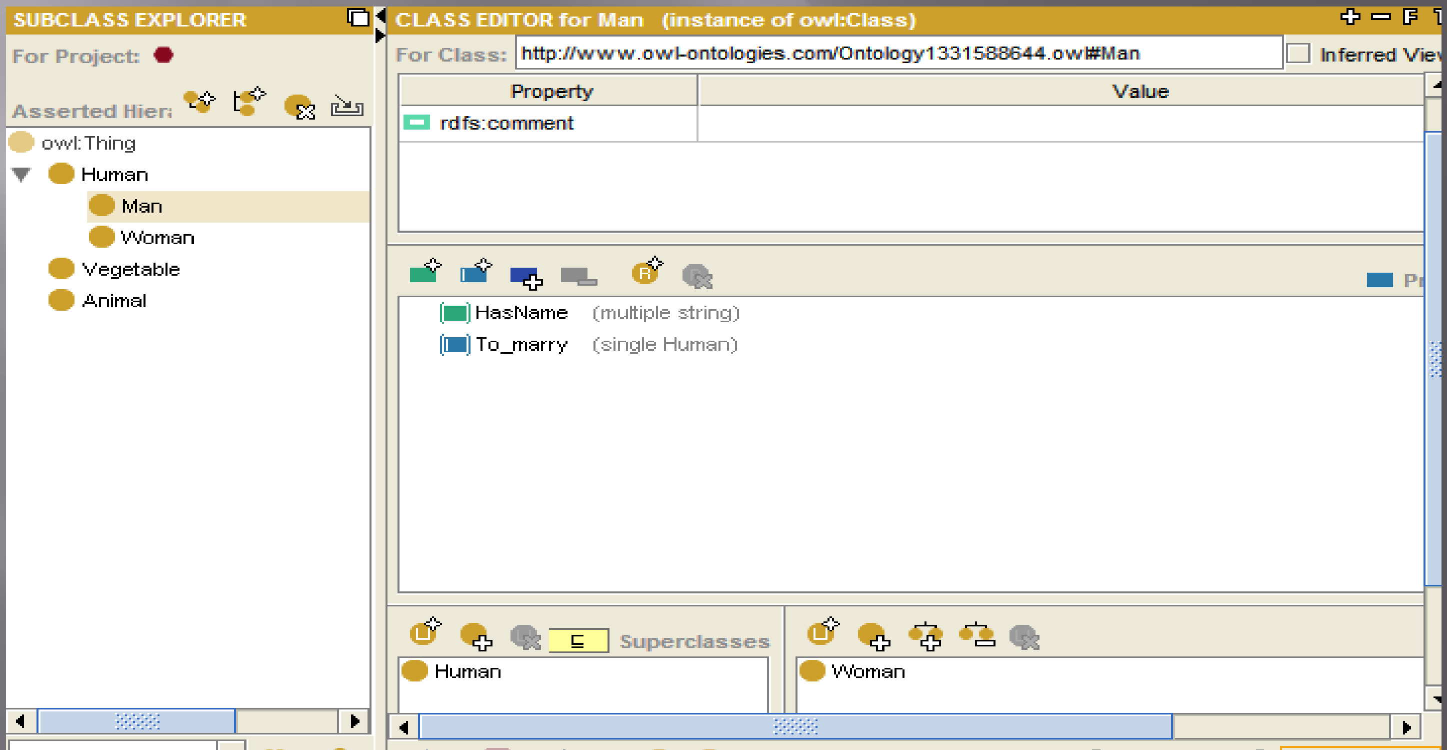
Definition of intances

```
<Woman rdf:ID="Mounira"/>
```

Definition of axioms with OWLProtege

Example : two classes are disjoints (i.e if it dosen’t exist an instance that belongs to the two classes)

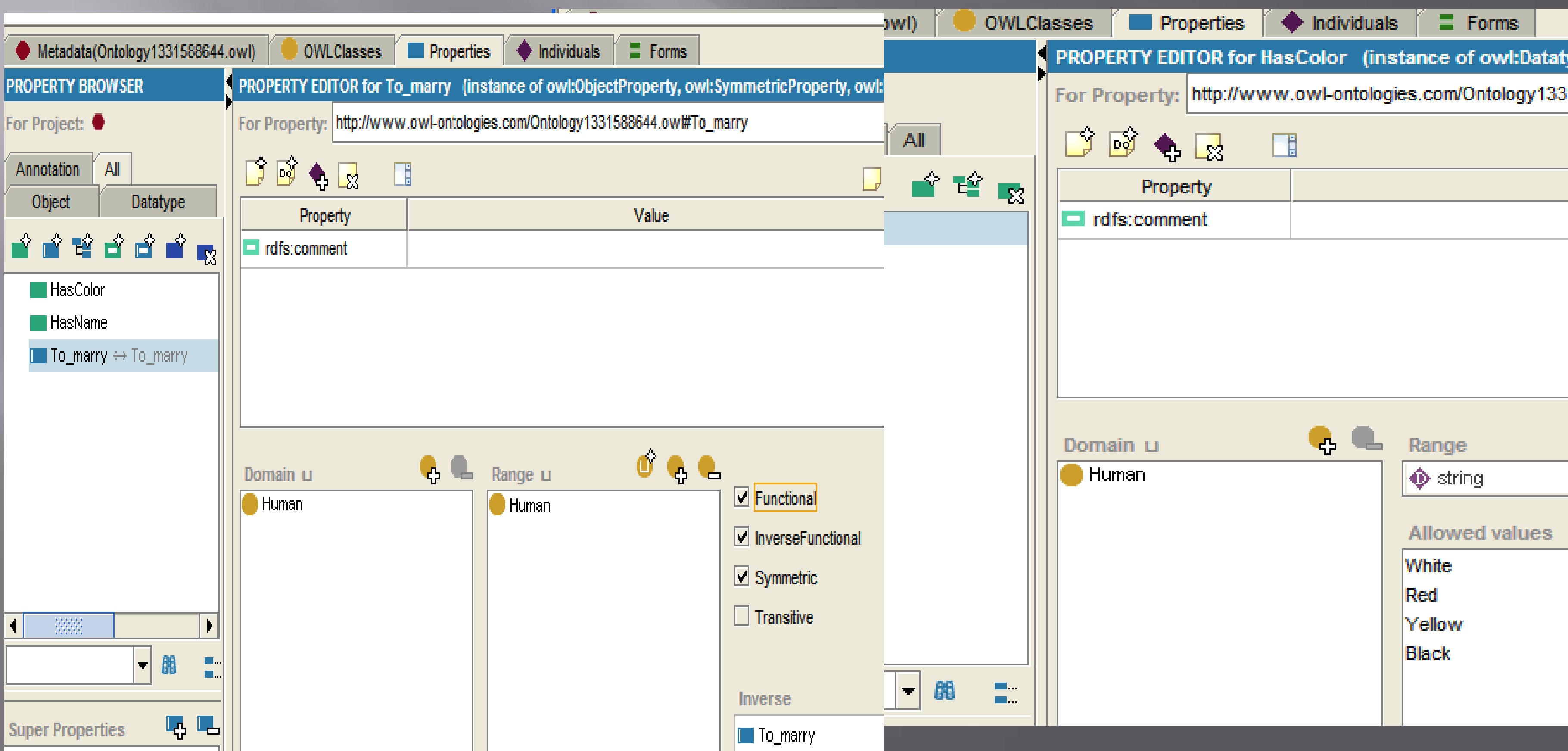
<owl:Class rdf:ID= "Woman">
<rdfs:subClassOf rdf:resource="Human"/>
<owl:disjointWith rdf:resource="Man"/>
</owl:Class>



Definition of properties with Protege/OWL

Two kinds of properties :

- Object properties : between two individuals
- Datatype properties : between an individual and a datatype



Definition of axioms OWL/Protege

Characteristics of properties :

- It can be the inverse of another property
- It can have only one value for a given instance: function
- It can be transitive
- It can be symmetric

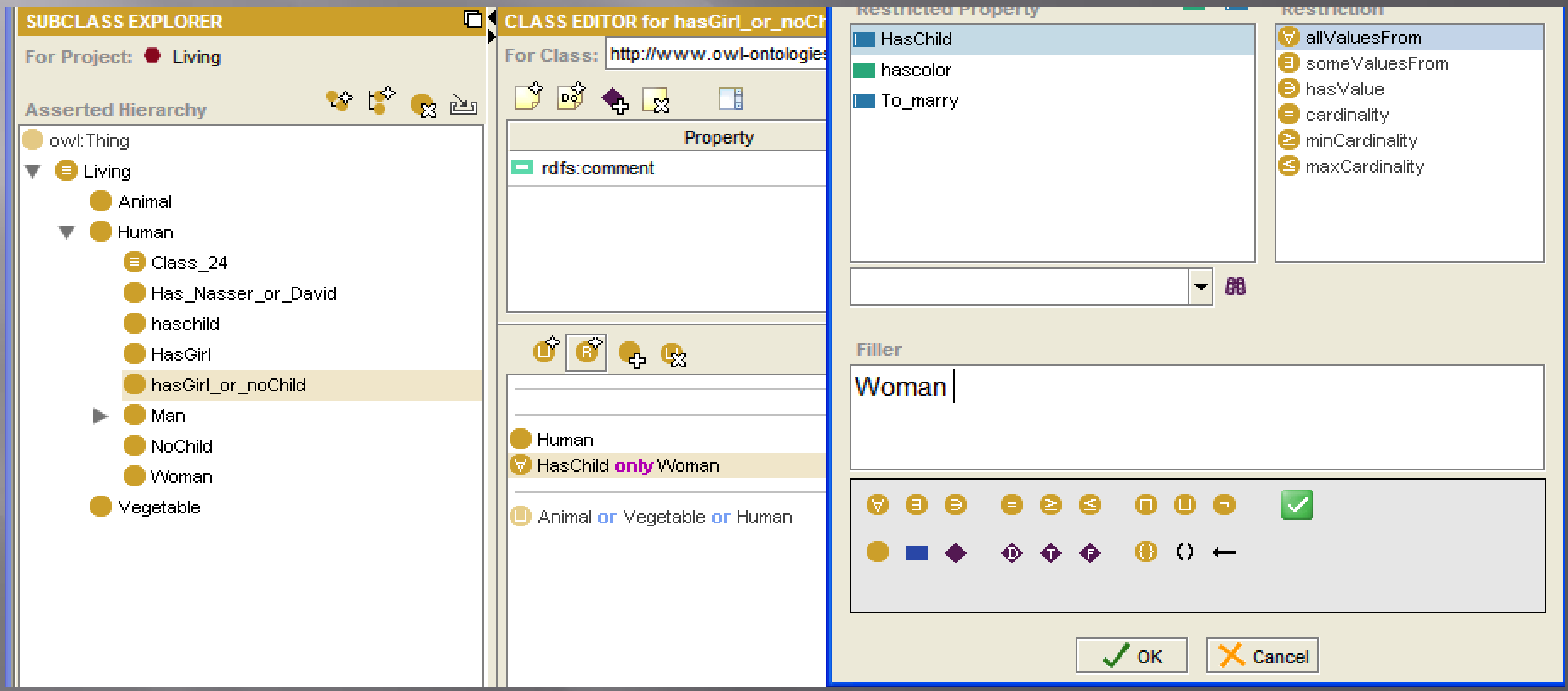
Restrictions on properties:

- Restrictions on the cardinalities
- Restrictions on the values (instances) (hasValue)
- Restrictions with quantifiers (existential and universal)

Definition of axioms with OWL/Protege

Restrictions with quantifiers

Example
"∀hasChild only Woman" : instances that have daughter or don't have child



Definition of axioms with OWL/Protege

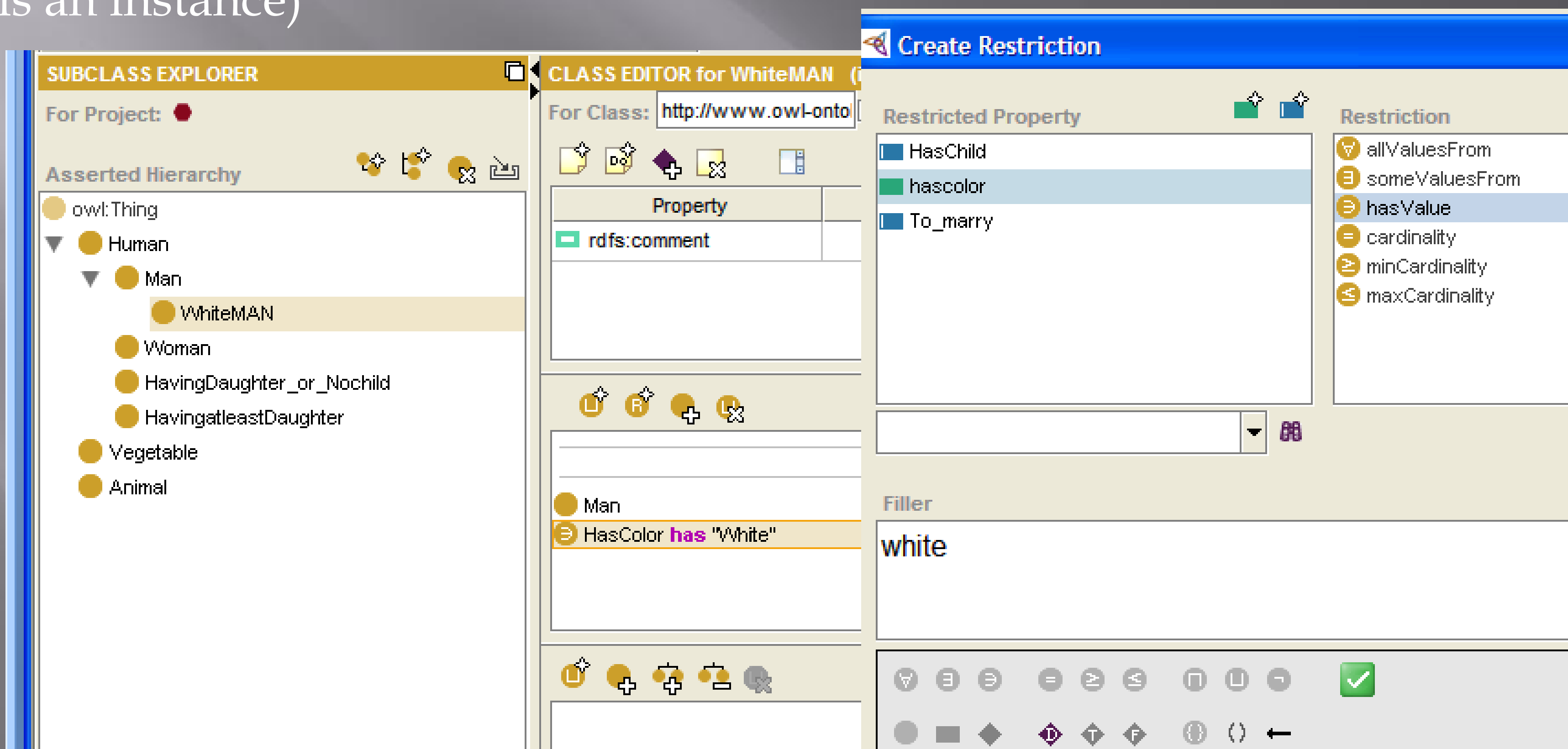
Restrictions with quantifiers

Examples :

∃HasChild some Woman : instances that has at least one daughter (someValuesFrom)

Restriction on Value (on instances)

∋HasColor has “White” : instances has at least a color white (here white is an instance)



Definition of axioms with OWL/Protege

Restrictions on cardinalities