

AN INTRODUCTION TO ONTOLOGY DEVELOPMENT

Stefano Bragaglia - May 14, 2010

ADDITIONAL NOTES

- ⊙ Developing an ontology is an ***iterative process***: adding new concepts often leads to refactoring
- ⊙ It is a good practice ***not*** to ***include unneeded concepts*** into the ontology (if it ain't broken, don't fix it)
- ⊙ Some concepts are difficult to express (i.e.: time relationships, equivalence, etc.) but can be easily handled by ***introducing ad hoc properties and classes***
 - ⊙ Time relationship: ***TimePeriod*** with ***isPrior*** or ***isAfter***
 - ⊙ Equivalence: ***Person*** with ***samePerson***

MULTIPLE INHERITANCE

- ⊙ One of the reasons ontologies are so popular is that they support ***multiple inheritance***
- ⊙ A ***YoungWoman*** may be defined as a subclass of ***Person, Young*** and ***Female***
- ⊙ This is as ***powerful*** as ***risky***, since it often collides with disjointness declarations
- ⊙ Watch out for inconsistencies!

MULTIPLE INHERITANCE

The screenshot displays the Protégé 3.4.4 interface with three main panes illustrating multiple inheritance:

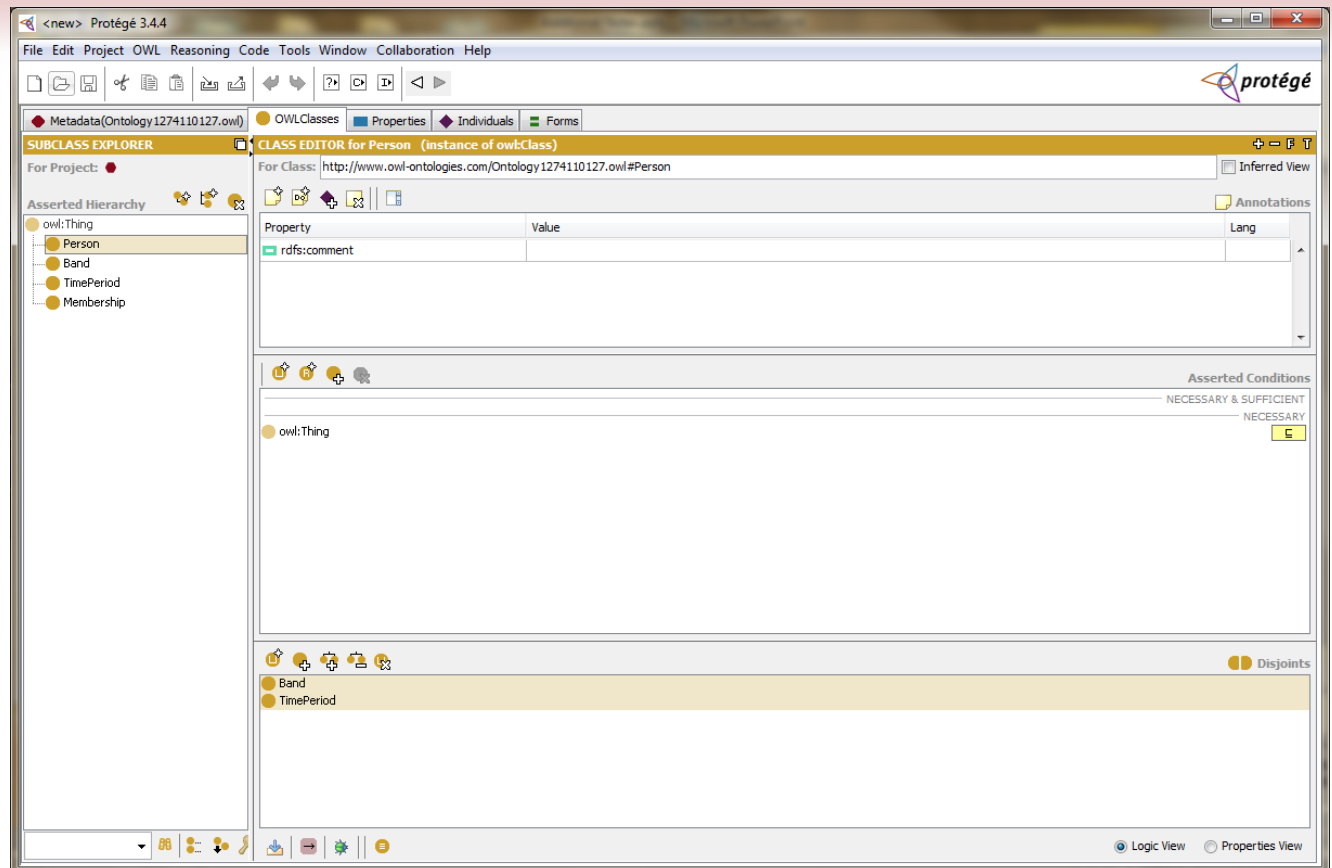
- Left Pane (Subclass Explorer):** Shows the "Asserted Hierarchy" for the project. The class hierarchy is: owl:Thing (parent) → Person (child) → Gender (child of Person) → Male (child of Gender) → Female (child of Gender) → YoungFemale (child of Gender). Other classes like Age, Young, Adult, and Elder are also shown as children of owl:Thing.
- Middle Pane (Subclass Explorer):** Shows the "Inferred Hierarchy". It highlights that "YoungFemale" is inferred as a subclass of "Male" because "Male" is a subclass of "Gender" and "YoungFemale" is a subclass of "Gender".
- Right Pane (Class Editor for Male):** Shows the "CLASS EDITOR for Male (instance of owl:Class)". The "Property" list includes "rdfs:comment". The "Value" field is empty. The "Lang" field is also empty. Below the property list, the "Asserted Conditions" section shows "Gender" as a necessary and sufficient condition for "Male".

The bottom of the interface shows the "Disjoints" section with "Female" listed. The status bar at the bottom indicates "Logic View" is selected.

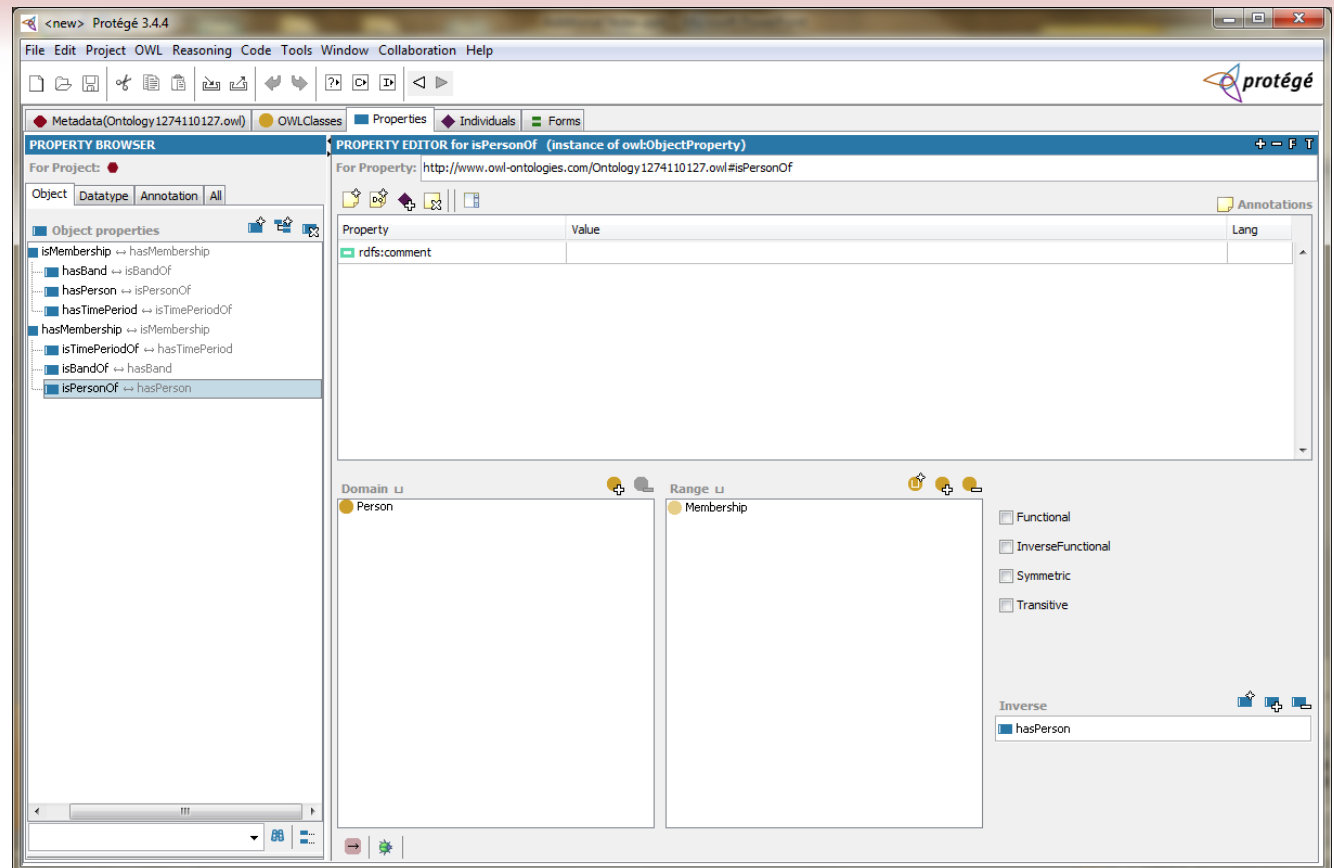
THE SITUATION PATTERN

- ⊙ Sometimes stating concepts separately is not enough since they have to be considered as a whole
 - ⊙ Being member of a certain band during a certain time period is an example
- ⊙ Such a case is generally named Situation and requires the definition the context, a class, and several object properties linking the context to each concept and vice versa
 - ⊙ Given the classes Person, Band and TimePeriod, the situation requires the definition of the class Membership and of the object properties hasPerson, hasBand and hasTimePeriod (and their inverse properties)

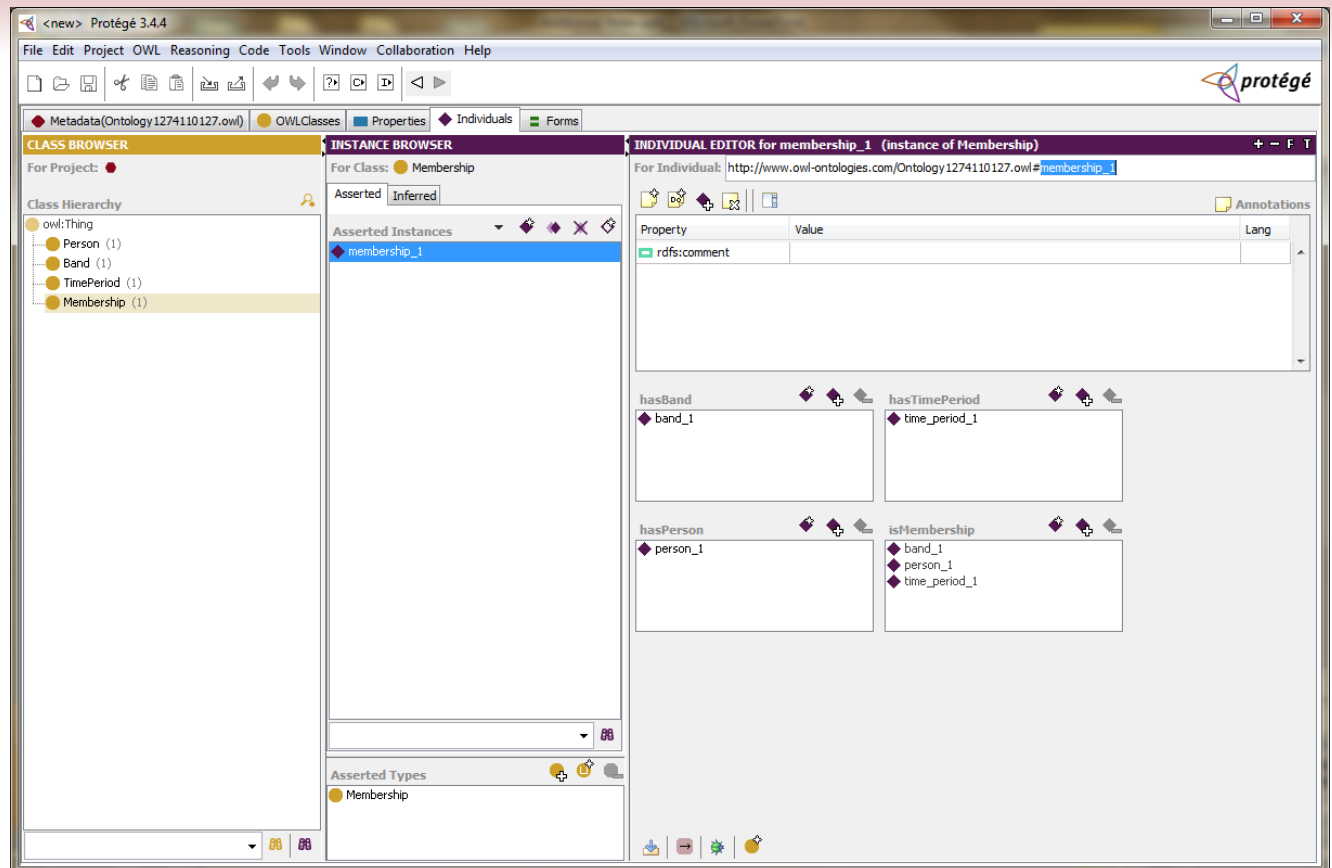
THE SITUATION PATTERN



THE SITUATION PATTERN



THE SITUATION PATTERN



- ◎ If you have any questions, please send me an email or call me:

E-mail: stefano.bragaglia@unibo.it

Tel: 051-20.93086