CO-Protégé: A Groupware Tool for Supporting Collaborative Ontology Design with Divergence

Alicia Díaz[1,2], Guillermo Baldo[1]

[1] Lifia, UNLP, La Plata, Argentina
[2] Loria, Nancy, France

Co-Protégé: Motivation

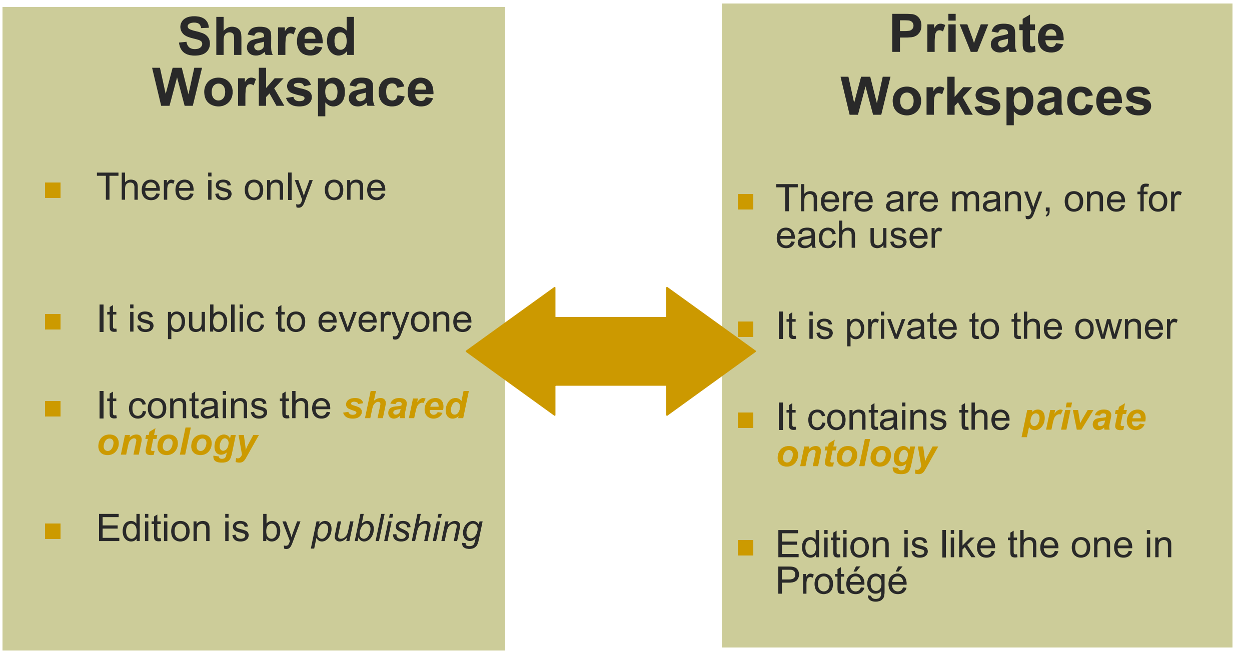
- To have a platform to develop an ontology in a collaborative fashion
- People can carry out this collaborative activity by means of
  - using of a private and a shared workspace,
  - supporting the discussion of alternative designs and
  - being aware of the collaborative activity

Lifia

Co-Protégé: the groupware

- Process-Oriented Workspace
  - Differentiates between a private and a shared workspace
  - Management of a private and a shared ontologies
- Discussion support
  - To allow the needed reflection among users to arrive at a unique ontology
- Awareness Information
  - To keep users up-to-date about the shared ontology evolution

The Workspace (WS)



Shared Workspace

- There is only one
- It is public to everyone
- It contains the ***shared ontology***
- Edition is by *publishing*

Private Workspaces

- There are many, one for each user
- It is private to the owner
- It contains the ***private ontology***
- Edition is like the one in Protégé

Discussion support in Co-Protégé

- It supports a discussion thread by
  - objecting shared ontological designs
  - giving alternative ontological designs
  - giving arguments to support or object different positions

Awareness information

- To keep people aware of
  - the activity in the shared workspace
    - the changes of the shared ontology (differences) and
    - the actions performed

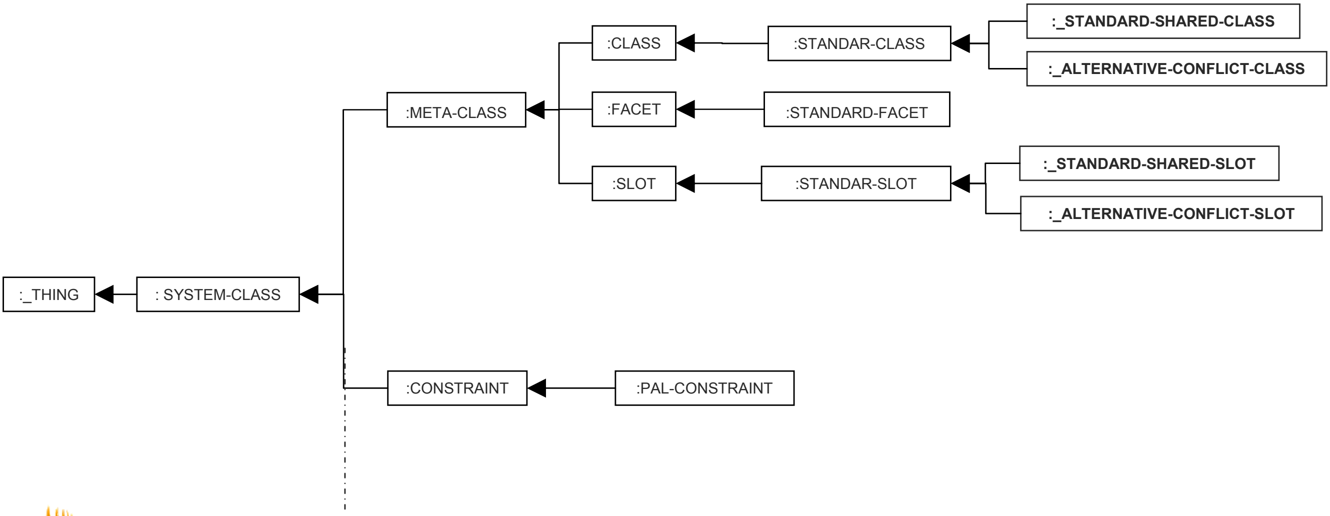
- Demo

Next

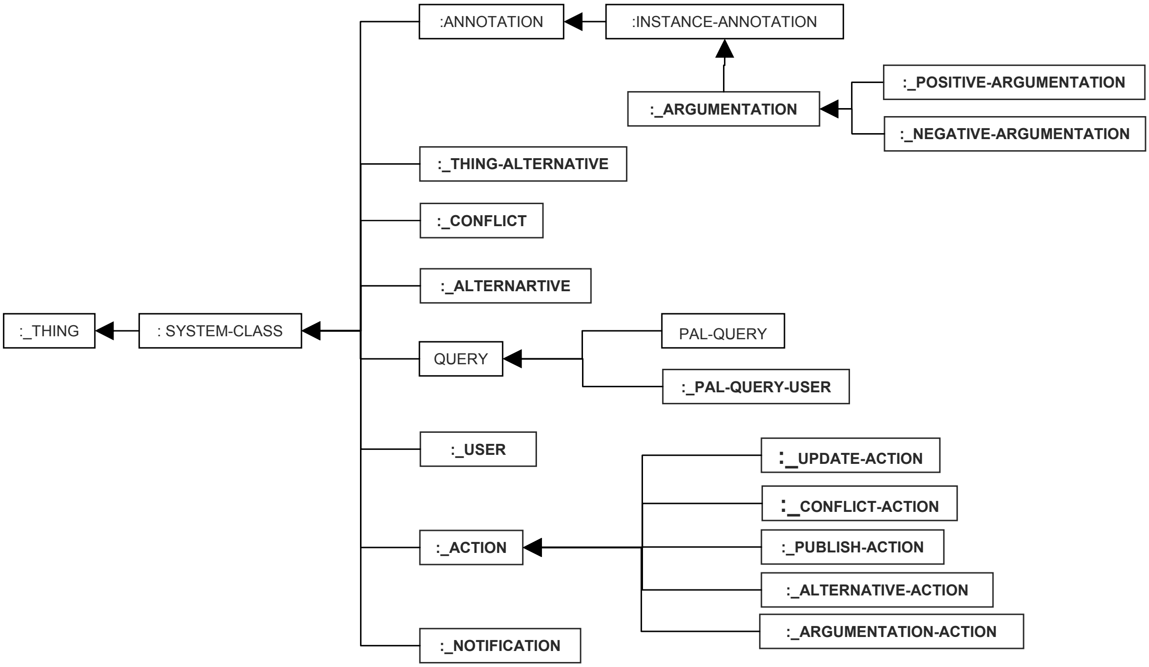
- Policy to close the discussions
- To improve visualization: awareness and alternative
- Performance features: large ontologies, many users
- OWL

Implementation

- metamodel



Implementation: generic ontologies



:_THING

: SYSTEM-CLASS

:ANNOTATION

:INSTANCE-ANNOTATION

:_ARGUMENTATION

:_POSITIVE-ARGUMENTATION

:_NEGATIVE-ARGUMENTATION

:_THING-ALTERNATIVE

:_CONFLICT

:_ALTERNARTIVE

QUERY

PAL-QUERY

:_PAL-QUERY-USER

:_USER

:_ACTION

:_UPDATE-ACTION

:_CONFLICT-ACTION

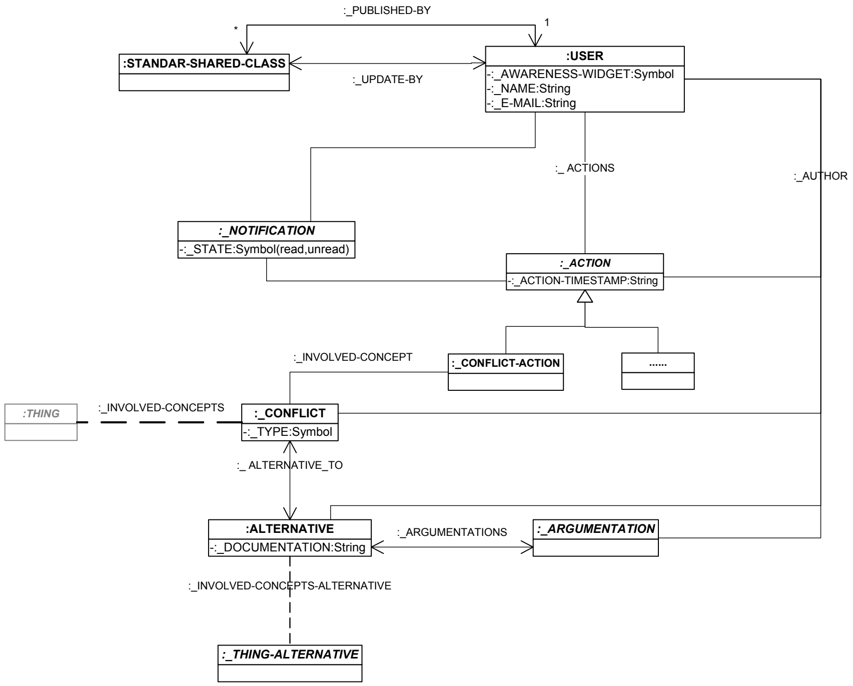
:_PUBLISH-ACTION

:_ALTERNATIVE-ACTION

:_ARGUMENTATION-ACTION

:_NOTIFICATION

Co-Protégé’s generic ontologies-2



:_PUBLISHED-BY

*

1

:STANDAR-SHARED-CLASS

:_UPDATE-BY

:USER
-:_AWARENESS-WIDGET:Symbol
-:_NAME:String
-:_E-MAIL:String

:_ ACTIONS

:_AUTHOR

:_NOTIFICATION
-:_STATE:Symbol(read,unread)

:_ACTION
-:_ACTION-TIMESTAMP:String

:_INVOLVED-CONCEPT

:_CONFLICT-ACTION

......

:THING

:_INVOLVED-CONCEPTS

:_CONFLICT
-:_TYPE:Symbol

:_ ALTERNATIVE_TO

:ALTERNATIVE
-:_DOCUMENTATION:String

:_ARGUMENTATIONS

:_ARGUMENTATION

:_INVOLVED-CONCEPTS-ALTERNATIVE

:_THING-ALTERNATIVE

Co-Protégé: A Groupware Tool for Supporting Collaborative Ontology Design with Divergence

Co-Protégé, is a platform to develop an ontology in a collaborative fashion, where people can carry out this collaborative activity by means of

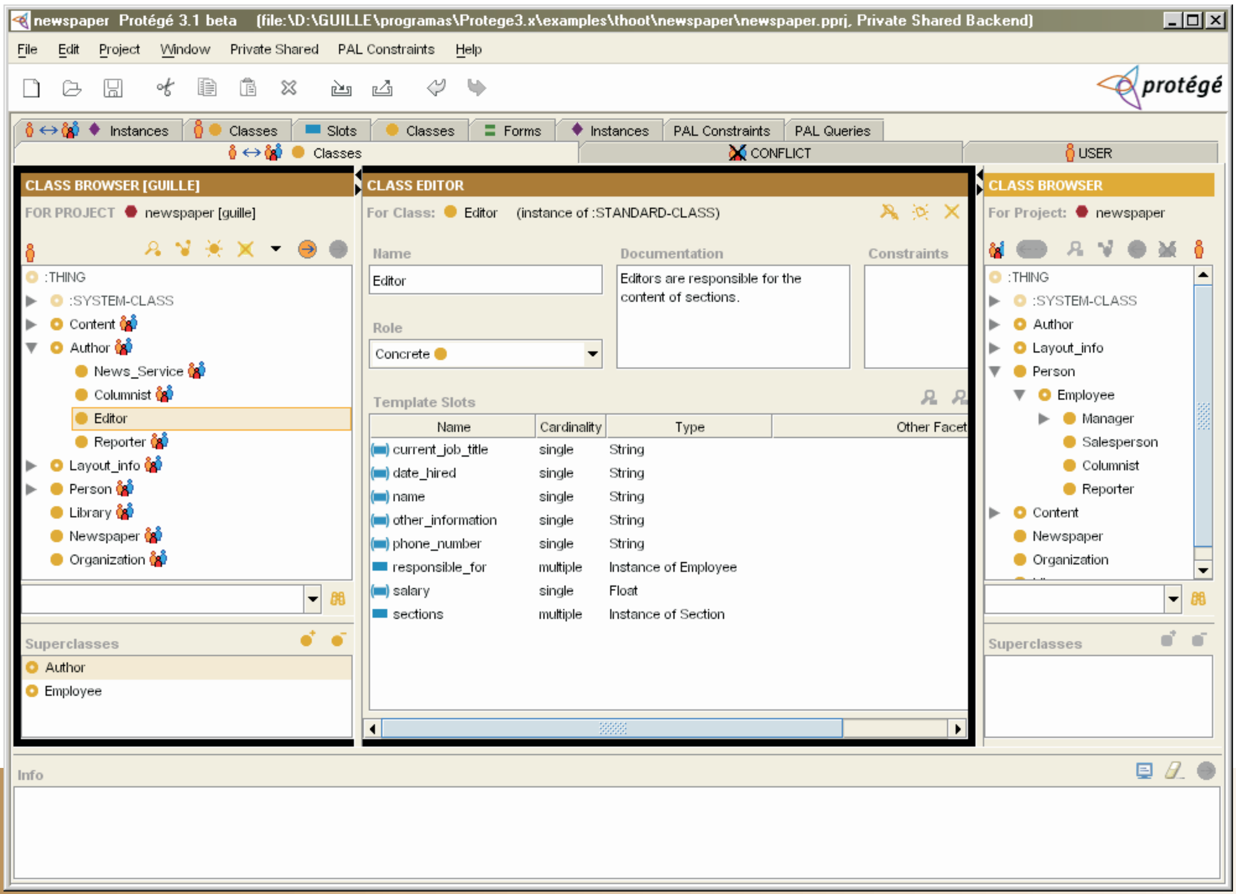
- using a private and a shared workspace
- supporting the discussion of alternative conceptualizations and
- being aware of the collaborative activity

Co-Protégé imposes its own manner of carrying out the collaborative development of an ontology. In Co-Protégé people do not make a direct edition of the shared ontology, instead they change it by means of the publication of an ontological artifact.

In Co-Protégé users manage simultaneously two ontologies: the private ontology and the shared one. There is also a shared workspace where the shared ontology is updated by means of the ontological contributions. Users edit the private ontology in the private workspace as users were working in Protégé-2000 stand-alone fashion. Then, they can publish ontological artifacts to the shared ontology. The shared ontology is manipulated in the shared workspace and its "edition" is carried out through publications.

Divergent contributions make possible the coexistence of many conceptualizations simultaneously at the shared ontology . They are the resource to discuss the design of an ontology.

Most collaborative edition functionalities are provided through a special tab, the shared-private tab that overlaps both workspaces and allows users to manipulate simultaneously their private ontology and the shared one.


newspaper Protégé 3.1 beta (file:\D:\GUILLE\programas\Protege3.x\examples\thoot\newspaper\newspaper.pprj, Private Shared Backend)

For Class: Editor (instance of :STANDARD-CLASS)

Name: Editor

Documentation: Editors are responsible for the content of sections.

Constraints:

Role: Concrete

Template Slots

Name	Cardinality	Type	Other Facets
current_job_title	single	String	
date_hired	single	String	
name	single	String	
other_information	single	String	
phone_number	single	String	
responsible_for	multiple	Instance of Employee	
salary	single	Float	
sections	multiple	Instance of Section	

A snapshot of Co-Protégé. Both private and shared ontologies can be appreciated simultaneously. The black rectangle remarks the associated property pane to the current ontology.

Shared-Private tabs

Co-Protégé proposes tabs that "overlap" both workspaces in the same tab. This kind of tab joins both the private and the shared ontologies, easily achieving to a direct manipulation of the two ontologies. Alternating both ontologies is easy, because they are visually overlapped.

Only the private side of a shared-private tab has the same functionality as Protégé-2000 to edit a single ontology; the shared side cancels them because the shared ontology is updated by publications.

There is one tab for each kind of frame (class, slot and instance) and each one shows the two ontology versions: the private and the shared ones. They are the classes-shared-private tab, the slots-shared-private tab and the instances-shared-private tab.

There is a series of operations that allows making contributions from one side to the other. They are organized in two groups: publications from the private to the shared side and publications from the shared to the private side. The system makes compatibility checking each time a publication is performed, and it is completed only if the publication is “augmentative”. Whatever the checking result is, Co-Protégé informs this result on the bottom the shared-private workspace tab. Besides, at the shared-private tab it is also possible to open conflicts.

Conflict tab

A discussion is created in the shared-private tab by selecting the set of frames that will be put in conflict. After that, the frame is shown as "in conflict" To facilitate the visualization of conflict, it was decided to separate the conflict management form the shared-private tab. The conflict tab defines a space where users can browse and develop a conflict. Once a conflict was created, it becomes part of the conflict list, where all currently open conflicts are listed. Users can add alternatives and argumentations. Alternative are created with frames from the private ontology. It is the mechanism that enables to publish contributions that did not pass the compatibility checking.

User tab

This is the tab to manipulate the user's profile, to update users' interests. Users' interests can point to any kind of frame described by the metamodel of Co-Protégé, that is, elements of the shared ontology, other users, conflicts and conflict components. There are some cases where the system is in charge of changing users' profile, by tracking user activity. Then, this may be used to adapt delivered awareness information. Co-Protégé provides a set of suggestions that helps users to complete their profile.

Implementation Features

- Co-Protégé extends Protégé through the definition of some plugins. The resulting architecture looks like any other Protégé extension because it follows the Protégé extension philosophy. In Co-Protégé a project is made up of the shared ontology plus a private ontology (one for each user). Both kinds of ontologies are stored in Standard Text File format.

- Co-Protégé is a client-server application, where a project is defined as a Protégé's metaproject. In this metaproject every ontology is defined (the shared and each private) and the access permissions.

- Co-Protégé uses the Protégé-knowledge model: classes, slots, facets and instances. However, it uses two different metamodels to represent both the private and the shared ontologies. Any private ontology is considered as a Protégé-2000 project. However Co-Protégé proposes its own metaclass architecture, which is an extension of the Protégé-2000, to model the primitive frames of a shared ontology and conflict primitives. Besides, Co-Protégé defines a set of generic

Contact: Alicia Díaz

Lifia, Fac. Informática- UNLP
CC 11, 1900 La Plata, Argentina
+54 221 423 6585 / +54 221 422 8252
alicia.diaz@sol.info.unlp.edu.ar