

# JINSTANCE PLUGIN

Carmen Benavides, Isaías García, Héctor Aláiz, Ángel Alonso, and Fernando Lomas

Control and Systems Engineering Group, Electrical and Electronic Engineering Department, University of León, Escuela de Ingenierías, Campus de Vegazana, 24071 León (Spain)  
{diecbc dieigr, hmoreton, dieaaa}@unileon.es

One of the most important and difficult parts in the development of a Knowledge Based System is the domain knowledge acquisition process, we can do it using an ontology editor like Protégé. Protégé is a free, open source ontology editor and knowledge-base framework, it is based on Java, is extensible, and provides a foundation for customized knowledge-based applications.

Once we have designed and initialized the ontology with Protégé, there are several options to work with this knowledge, such as:

## 1. Interacting from the protege environment:

- a) Using one of the many plugins developed for protege, like jesstab, clipstab, Algernon Tab or Prolog Tab Widget.
- b) Building our own plugin to customize the interface and functionality.

## 2. Developing an entirely separate, external application to protégé, usually with java language:

- a) The external application can use the Protégé's API to open the Protégé project and retrieve or modify the classes of the ontology. In this case, the developer needs to know a new API, for example suppose we have a protege class called Car, with a slot called model, to create a new instance of the Car class

```
Cls myCar=kb.getCls("Car");  
Instance i=kb.createInstance(null, myCar);
```

- b) Using Jsave plug-in developed by Samson Tu, we get Java class definition stubs for Protégé classes. In this case, to create an instance of the class Car will use:

```
Car myCar=new Car();
```

From developer's point of view this is the most intuitive way to work, but we must know that we extract every classes of a project with Jsave but we don't extract its instances.

Our contribution consists in extending jsave to give the user the option of extract also all the instances that exist in the project, we have called it "jinstance". Jinstance permits building a Knowledge Based System from an ontology developed with Protégé like an independent application of Protégé that includes all original classes and instances of the project. Furthermore, we have provided a graphical interface to jsave and jinstance, in such a way that it appears like a Protégé tab, making easier to work with it.