Automatic generation of Object-Relational Databases from the Protegè Ontologies. *PostgreSQLGen* and *WebIntGen* Plugins

C. Albanese[2] (albanese@marscenter.it), F. Mele[1,3] (f.mele@cib.na.cnr.it), G. Minei[2,3] (minei@marscenter.it), F. Pagano[2] (pagano@marscenter.it)
1 CNR – Istituto di Cibernetica - Pozzuoli – Napoli – Italia; 2 MARS Center Napoli – Italia; 3 Università degli Studi di Napoli Federico II Facoltà di MM. FF. NN., C.L. Informatica Napoli – Italia

Introduction – The frame based formalisms have many common theoretical aspects with the models of object-relational database. In our work in order to define and develop the plugins *PostgreSQLGen* and *WebIntGen*, we have executed an analysis of the correspondences among Frame Based systems and Object Relational Databases. *PostgreSQLGen* generates an object relational database (ProstgreSQL database) in automatic way from an Protegé ontology, while *WebIntGen* creates a Web Interface for the management of the generated database. The plugins have been developed at MARS Center in Naples (Italy) and used, in an intensive way, in the frame of an industrial project of Alenia Aeronautica. In this project an ontology, for concepts of the domain of Non Destructive Controls (NDC), has been developed. In such ontology all the entities, related to the control of the materials (hardware and software tools, plattforms for the diagnostic investigations, peoples, processes,.., etc), have been represented. The ontology incluses about one hundred classes and a thousand of slots. The plugin PostgreSQLGen allowed to export in automatic way the ontological structure in the form of a database PostgreSQL. The plugin *JspWebGen* has been used for the automatic generation of web interfaces to support expert users in NDC, but with a low level of knowledge about the databases.

Correspondences between Protegé and PostgreSQL – In order to generate a database schema in PostgreSQL using Protegè, we have analysed the relationships which exist between two systems. Our aim was find a general method to demonstrate that these relations could be valid for all the knowledge systems based on ontologies and for all the databases based on object relational models. In Protegé the basic elements are Classes (Frames) each Class is described by a set of Slots. In PostgreSQL the basic elements are the tables, with theattributes as the structural elements associated each of to them. For this reason we have defined two main correspondences between Protegé and PostgreSQL: we have associated to frames and slots of the ontologies respectively tables and attributes of the object relational databases.

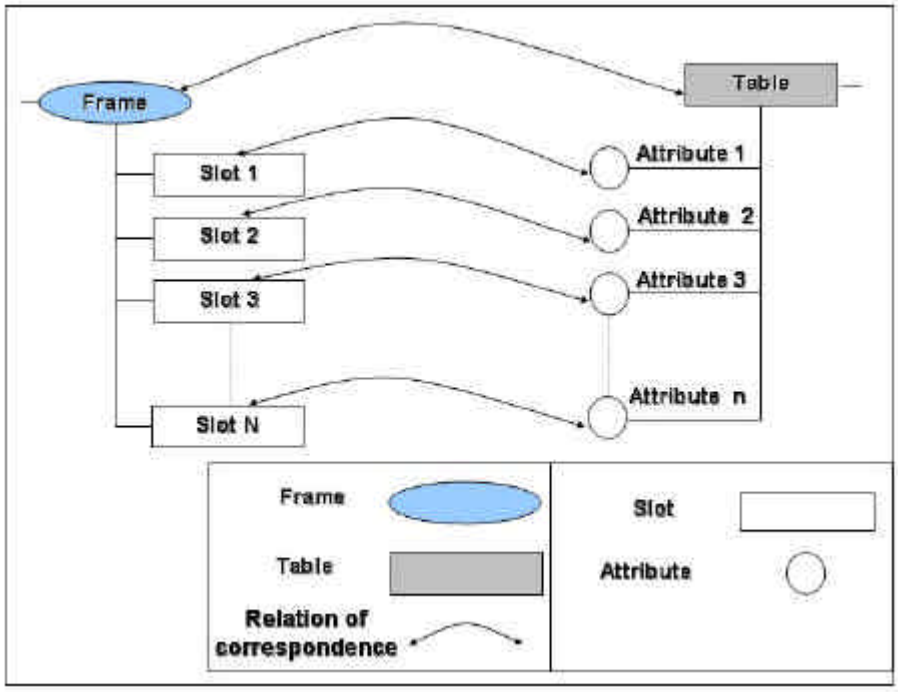

Figure 1 Fundamental correspondences

From the analysis of the *types* of Protegé Slots (Name, Documention, Template Values, Cardinality, Default, Value Type), we have defined other correspondences between Protegé and PostgreSQL, building analogous types on the PostgreSQL attributes. We show this correspondences in figure 2:

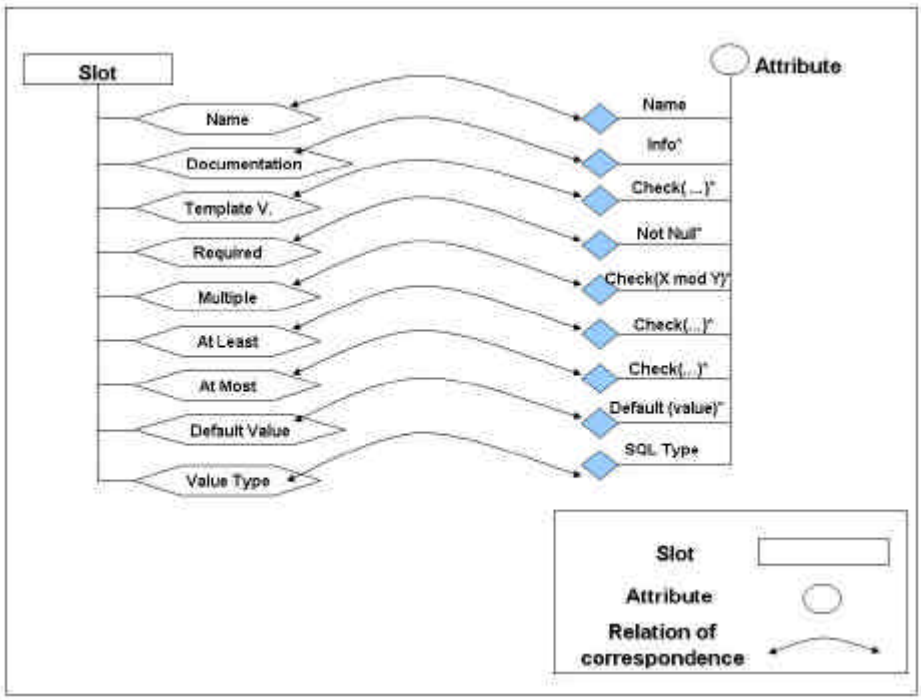

Figure 2. Correspondences among types

There are three cases in which it is not possible to define a simple correspondence between Protegé and PostgreSQL. These cases regard to the following Protegé types: Any, Class and Instance. Only for these three types, we have defined correspondence rules based on additional structures generated in PostgreSQL .

System architecture - In our approach the system architecture is composed by a set of modules. Starting from the definition of an ontological schema in Protegé, this schema is translated into a PostgreSQL database, tanks to the veloped plugin of Protegé (PostgreSQLGen).

The plugin consists in a Java application, which allows to reproduce the some operations on an ontological schema in Protegé, on the database exported in PostgreSQL.

In particular this first plugin is composed by the following modules:

- a module for executing operations (modification of a class, with respect to inheritance constraints; cancellation of an existing class; addition or cancellation of a slot, etc);

- a module for updating operations (creation of a new class instance; modification of an instance values; cancellation of an instance; etc);

- a second plugin for quering the generated DB, starting from the ontological structure.

The functionalities, above described, have been implemented by a set of algorithms. implemented as Java methods, they represent the main components of the plugin.

Two others modules are part of the architetture; a Java library which allows the access to Protegé ontological data and metadata structures; a JDBC Manager which allows the access the database, independently from the platform (PostgreSQL, Access, Oracle, etc).

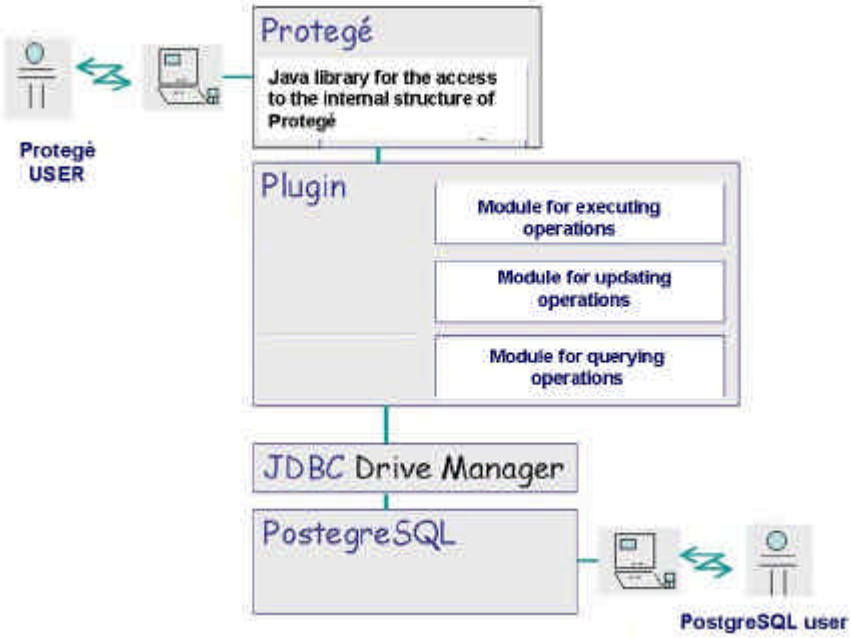

Figure 3 Main software components of the PostgreSQLGen plugin

Implementation - The plugin, PostgreSQLGenTab, has been implemented as a Tab-widget plugin. In particular, we have defined user interfaces that allows to share operations executed on ontological schema with a defined PostgreSQL database whose structure is related to Protegé ontological structure.

A second plugin, JspWebGenTab, has been implemented. Starting from a Protegé ontological project, the plugin automatically generates, a set of WEB user interfaces based on technology JSP. These interfaces can support users in managing a PostgreSQL database using a WEB browser, instead of Protegé system interface.

The interfaces created by the plugin are organized with respect to Protegé ontological structure related to PostgreSQL database. JSP interfaces are managed by Tomcat as a web application.