SQL & C++ Sources Generator

Anthony Pietromartire - *pietromartire.anthony@nist.gov*
Zeid Kootbally - *zeid.kootbally@nist.gov*
National Institute of Standards and Technology

July 26, 2012

Contents

1 Introduction

The Generator tool is a graphical user interface developed in Java, allowing the user to store data from OWL files into a MySQL database. This tool also permits the user to query the database using the C++ function calls. The tool Generator is composed of the following functionalities:

1. Convert OWL documents into SQL syntaxes (OWL to SQL).
2. Translate SQL syntaxes to OWL language in order to modify an OWL document (SQL to OWL).
3. Convert the OWL language into C++ classes (OWL to C++).

To date, only steps 1. and 3. have been implemented and will be covered in this document.

2 Prequisites

The description of the Generator tool is given for a Ubuntu Linux system. To run and use the Generator tool, different applications must be installed on the system.

2.1 Java Runtime Environment

The Generator tool comes as a jar file. As such, the Java Runtime Environment should be installed on your system. This application can be found at *java.com*

2.2 MySQL Server and Client

The MySQL server and client should be installed and running on your system.

- *sudo apt-get update* (Update the package management tools)
- *sudo apt-get dist-upgrade* (Install the latest software)
- *sudo apt-get install mysql-server mysql-client* (Install the MySQL server and client packages)

When done, you have a MySQL database ready to run. However, you need to set a root password, for starters. MySQL has it's own user accounts, which are not related to the user accounts on your Linux machine. By default, the root account of the MySQL Server is empty. You need to set it. Please replace the following occurrences of *'mypassword'* with your actual password and *myhostname* with your actual hostname (localhost if you are installing on the local machine).

- *sudo mysqladmin -u root -h myhostname password 'mypassword'*

Finally, we need the plugin `libmysqlcppconn-dev` which allows C++ to connect to MySQL databases. It can be installed as follows:

- *sudo apt-get install libmysqlcppconn-dev*

3 How to Run the Generator Tool

The Generator tool can be launched using either one of these two following methods:

1. java -jar Generator.jar
2. Right-click on Generator.jar and select the option "Open With OpenJDK Java 6 Runtime". Note that this message will be different for future releases of the Java Runtime Environment.

4 Functionalities

As mentioned in the Introduction, we are covering only steps 1. and 3. in the rest of this document, i.e., *OWL to SQL* and *OWL to C++*, respectively.

4.1 OWL to SQL

To convert OWL classes and instances to SQL, the `Owl to SQL` tab should be selected (see Figure 1). The different fields are:

4.1.1 Generate SQL Files

- Ontology Path: This field requires the file `kittingInstances.owl`. Before doing so, you need to modify one line in this file. Open it with a text editor and find the line `Import(<file:kittingClasses.owl>)`. Modify this line by giving the absolute path to the file `kittingClasses.owl`. You should should have something that looks like `Import(<file:/home/username/NIST/ipmas/Generator/kittingClasses.owl>)`. When this is done, save the file, and browse to `kittingInstances.owl` using the "Browse" button.
- Browse to the directory where you want to save the SQL files.

Once the two previous steps are done, click on "Generate SQL". You should receive a message confirming the generation of the SQL files: `kittingInstances.owlCreateTable.sql` and `kittingInstances.owlInsertInto.sql`. The former is used to create tables, the latter is used to populate these tables;

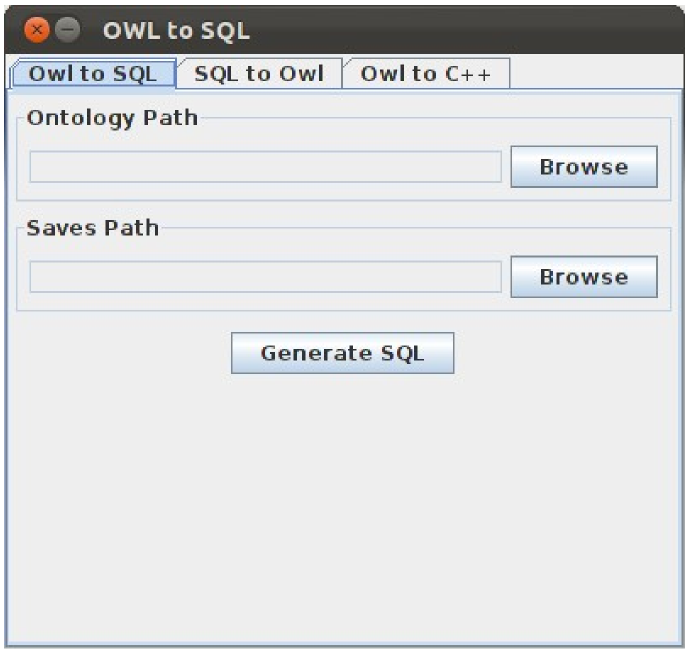


Figure 1: Owl to SQL tab.

4.1.2 SQL Tables and Insertions

The next step is to create a database and to populate it.

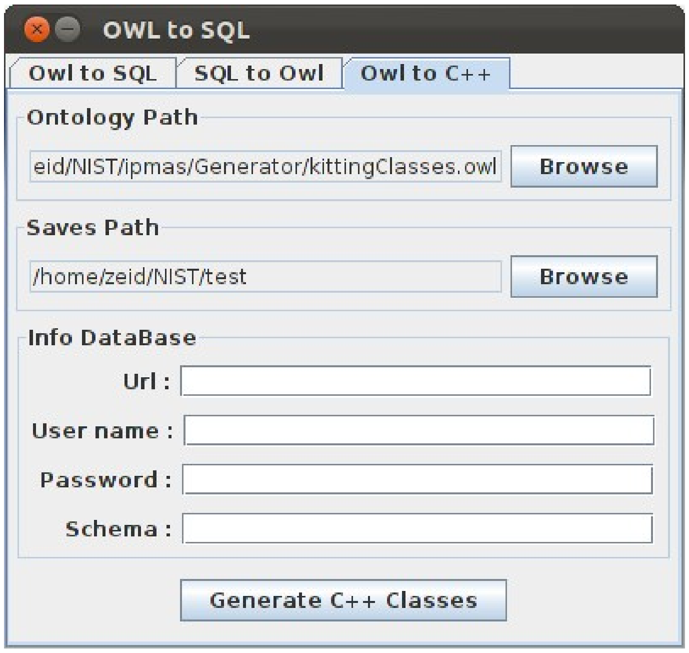
- Connect to mysql using *mysql -u root -p*, then enter your password. You should be in the mysql shell if this succeeded (*mysql>*).
- Delete a previous database (if you already used this tool and you want to replace the existing database with this new one) : `mysql>` *DROP DATABASE OWL;* (*OWL* is the name of the old database).
- Create a database:
  - `mysql>` *CREATE DATABASE OWL;*. Here, *OWL* is the name of the database (you can use a name of your choice).
  - Before performing the following commands, we need to tell MySQL which database we are planning to work with (*OWL* in our case). This is done using:

    `mysql>` *USE OWL*
- Populate the database with tables using `kittingInstances.owlCreateTable.sql`.
  - `mysql>` *source <path>/kittingInstances.owlCreateTable.sql;*
- Populate the tables with data using `kittingInstances.owlInsertInto.sql`:

☐ `mysql>` *source <path>/kittingInstances.owlInsertInto.sql;*

<path> designs the absolute path to the appropriate file.

4.2 OWL to C++



Ontology Path: eid/NIST/ipmas/Generator/kittingClasses.owl
Saves Path: /home/zeid/NIST/test
Url :
User name :
Password :
Schema :

Figure 2: Owl to C++ tab.

The “Owl to C++" tab (see Figure 2) is used to generate C++ classes and scripts allowing the connection between C++ and MySQL. The different fields are explained below:

- **Ontology Path**: This is the path to the ontology (`kittingClasses.owl` in our example).
- **Saves Path**: Directory where the C++ files and scripts will be generated.
- **Url**: This is the url of the database. It’s usually the IP address of the machine hosting the database (127.0.0.1 if it is local).
- **User name**: User name used to connect to the MySQL database.
- **Password**: Password associated to the user name to connect to the MySQL database.
- **Schema**: This is the name of the database (*OWL* in our example).

When all the fields are completed, click the “Generate C++ Classes" button to generate C++ and script files.