Plug-in for Protégé 2000 which supports Sesame

Kalid Askar, Srinivas Vemuri, Yella Siril and Mark Dougherty
Department of Computer Engineering, Dalarna University, Sweden
{kar, v03srive, v02sirye, mdo}@du.se

Abstract

This paper presents a solution for a new plug-in named Passerelle for Protégé 2000. Passerelle makes it possible to connect Protégé to Sesame architecture, to store and query RDF(s) data. With Passerelle, Protégé becomes a stronger ontology editor. It gives an ontology developer, the possibility of using RDF query Language (RQL) as it enables much stronger and advanced query possibilities than Protégé axiomatic language (PAL). The aim of this work is to bridge the gap between the two entities Protégé and Sesame, by cutting down the existing differences of namespaces, respectively.

Introduction

In our work, we aim to bridge the gap between the use phase and the design phase of the product life cycle, so that the design of the new products is augmented. We aim to achieve this by developing an intelligent distributed knowledge management system to create and manage data flows between maintenance, warranty and design functions. Enhance the support and collaboration between the distributed players in the product life cycle is another objective. We propose to do this by building an agent based system supported by a unifying ontology both at vehicle and enterprise level[1].

We have decided to use Protégé[1] and Sesame[2] in the project. Protégé is an integrated software tool that is used by ontology developer's to develop knowledge based systems. Applications that have been developed in Protégé are used in problem-solving and decision making problems. Sesame is a web-based architecture that allows the storing of RQL data and schema information and needs to have a storehouse for important RDF data. A Database Management System (DBMS) is needed to keep Sesame independent of any particular database and also to fit the best DBMS to store data. Sesame uses a Repository Abstraction Layer (RAL), where all the DBMS code is saved. RAL offers a Resource Description Framework (RDF) specific method to the client, so it can call a specific DBMS. It gives a selection of the storehouse without changing any Sesame components. Sesame functional modules are clients to RAL. There are three modules:

- RQL Query Module – It calculates the RQL queries which are created by the user.

- RDF administration module – Allows the uploading of RDF data and schema information and also allows deletion of information.

- RDF export module – It removes the complete schema and data from the model in RDF format.

Sesame supports many different ways of communication. It is possible to communicate over HTTP while using the web environment and, also by using Remote Method Invocation (RMI) or Simple Object Access Protocol (SOAP) [2].

One of the problems which we encountered during the project was to connect Protégé to the Sesame architecture. The reason we could identify was, both Protégé and Sesame use RDFS files but Protégé uses an old namespace version of RDF compared to Sesame. The obvious solution to solve this problem was to write a simple program to pre-process the RDF file from Protégé before it is loaded in to Sesame. This problem has lead to the idea of developing a simple and alternative solution which connects Protégé and Sesame, the plug-in called Passerelle. Passerelle in French means a connection between two entities. The paper commences with a brief motivation of the work and introduction to Sesame. The next section presents a description of the Passerelle and methodology involved in designing it. Finally, concluding remarks and necessary future research are presented.

---

[1] http://protege.stanford.edu/
[2] http://www.openrdf.org/

Description of Passerelle

The overview of the present work is shown in Figure1. In this section we give an overview of the main components. Passerelle bridges the gap between the two entities Protégé and Sesame. It exports the required Resource Description Framework (RDF) files to Sesame by communicating with the respective modules of Sesame. For efficient storage of this data Sesame needs a scalable repository. Database Management Systems have been used for decades for storing large quantities of data. To keep Sesame DBMS-independent, all the DBMS-specific code is concentrated in a single Repository Abstraction Layer called Storage and Inference Layer (SAIL). The SAIL is an application programming interface (API) which offers RDF specific methods to its clients. Sesame functional modules are clients of the SAIL API. Currently the functional modules which are clients of SAIL API are RQL query engine, RDF admin module and RDF export module [2].
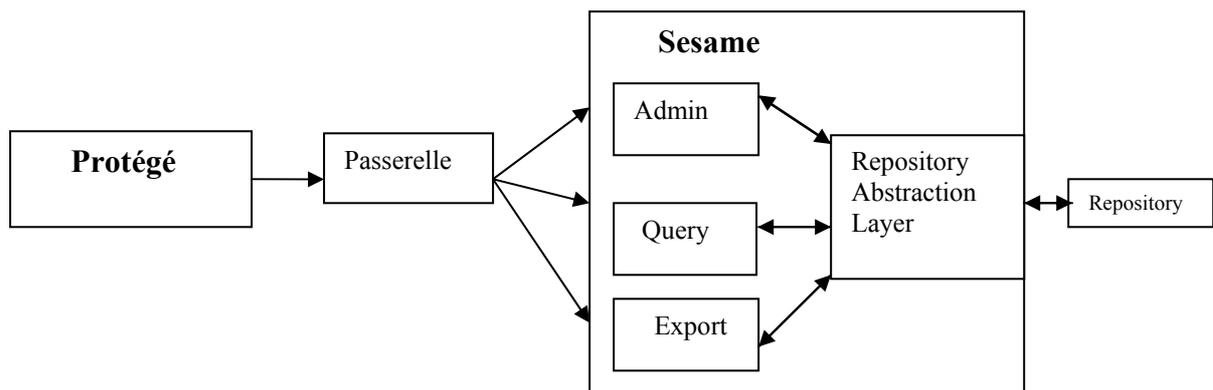

Figure 1 shows the basic architecture of Passerelle

Sesame provides the Administration Module to insert RDF data and Schema information into a repository. The Admin module offers two main functions:

- Incrementally adding RDF data/schema information.
- Clearing the repository.

 The Admin module retrieves its information from an RDF file and parses it using streaming RDF parser. The parser delivers the information to the Admin module on a statement-by-statement basis. The admin module then tries to add this statement to the repository by communicating with the SAIL. It returns back to the user, if any errors or warnings have crept in to the process. In Sesame, RQL queries are translated into a set of calls to SAIL. The Query module handles the querying part of the Sesame. Firstly this module parses the given query and then builds a query tree model for it. This model is then fed to the query optimizer. Finally the query optimizer transforms the query model into an equivalent model which evaluates the given query more efficiently.

The RDF Export Module is the simplest of the available three functional modules. This module exports the contents of the repository. The contents which are to be exported are formatted in XML-serialized RDF. The main idea behind the development of this module is to supply a basis for using Sesame in combination with RDF tools. All the RDF Tools can read the data in this XML-serialized RDF format. The RDF Export Module can selectively export the data and schema or both.

Design

The entire project is divided into three main modules. Various sub-modules are employed under these main modules to accomplish the task. The three main modules which are designed are:

- Module concerning Protégé and its libraries
- Module concerning Sesame and its libraries
- Module concerning the total layout of the project

The module concerning Protégé is intended to extend the AbstractTabWidget class and implement an initialising method. The module then loads the required RDFS file. The Protégé generated RDFS file has an erroneous tag **(<! ENTITY rdfs 'http: //www.w3.org/TR/1999/PR-rdf-schema-19990303#'>)**. This creates an error while uploading the RDFS file as Sesame uses a different namespace. The present module replaces this erroneous tag by **(<!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>)** which is acceptable to Sesame's namespace.

The module concerning Sesame is mainly intended to export this changed file. For this purpose we have concentrated our work on the Admin module. We have implemented the Sesame's repository API such that it can communicate with the Sesame server package. We have then created a Remote Service and finally we have set the appropriate parameters like username, password, database to be used and parsers to parse the input file.

The module concerning the total layout gives the overall view of the project. The above two modules are added to this module which completes the total work. The execution cycle of the present work is shown in Figure 2.
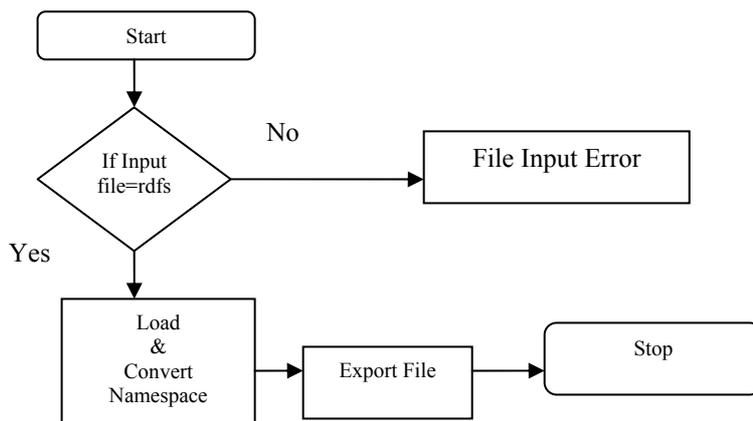

Figure 2 shows the execution cycle
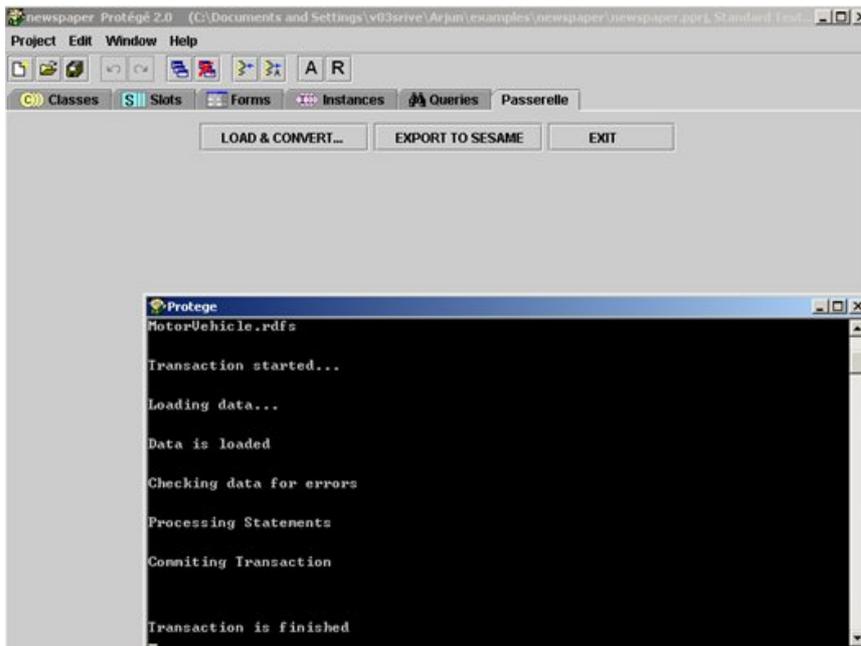
Screenshots of Passerelle

Figure 3 Screenshot of Passerele, the plug-in in Protégé

Figure 3 show the screenshots of the Passerelle. It allows the user to load the RDFS file which is developed from Protégé project file (pprj). By pressing the button *"Export to Sesame"* the program convert the standards and matches them to Sesame. When the file is loaded to Sesame, the RQL query get through sesame logic engine allows the query so the databases deliver the result to Sesame. This query can be used trough web environment[3], along with RMI[3] and SOAP[3] with the ontology developed in Protégé.

Test and validation

In this section the authors test and validate the effectiveness of Passerelle using the standard Newspaper example provided with Protégé (newspaper.pprj). The authors test, validate and check whether everything works right! First, we start with the newspaper example by manually converting the namespace of the generated RDFS file into Sesame acceptable form. We have then exported this file to the Sesame manually. Finally we have queried the database using RQL. And next, we have used Passerelle to export the converted RDFS files to the Sesame. We have just queried the database at Sesame and compared the results with the former manual implementation. And, it was proved that Passerelle works in par with the manual conversion. The important point to be noted is, Passerelle does the expected job of a plug-in, since the basic idea of any plug-in is mainly intended to simplify the work of knowledge engineers.

Conclusion

In this paper the authors have developed a trail version of a plug-in called Passerelle, that aid in solving the problem, raised during the development of an intelligent distributed knowledge management system by establishing communication between Protégé and Sesame. The development of Passerelle was an easy and alternate solution to convert the Protégé 2000 RDFS files such that, they can be supported by the Sesame environment. It also enables Protégé to use the services offered by Sesame, which makes it stronger and useful too. This avoids the necessity of developing a program to pre-process the RDF file from Protégé before it is loaded into Sesame.

Future work

The future work can be focused on how to process an RQL query within Protégé while still having the possibility of using the Sesame environment. This will result in a stronger version of Protégé. As, it facilitates the usage of RQL without struggling with other tools. We are also interested in making, Passerelle available on the internet and encourage users in the area to run the above plug-in to test and report errors, if any! This would aid in further development of the plug-in, leading to more simplified or sophisticated versions

References

1.   Askar, K., Dougherty,MS and Roche,T, *Agent Based System that support Reliability Transport Engineering.* 8th AATT 2004 Conference, Beijing, China, 2004: p. 456.

2.   Brokestra, J.a.A.K., *Query Language Defination: On-to-Knowledge (IST-1999-10132)*, in *Deleveriable.* 2001.

---

[3] http://www.openrdf.org/documentation.jsp