Passerelle – a plug-in that connects Protégé to Sesame

Kalid Askar, Yella Siril, Mark Dougherty
Department of Computer Engineering, Dalarna University, Sweden
kar@du.se, sye@du.se, mdo@du.se

Introduction

Passerelle is a plug-in developed for Protégé to support Sesame. It provides Protégé users with the possibility to connect their ontologies with Sesame architecture in order to store and query resource description framework (RDF) data from Protégé [1]. We have identified a lack of a strong query language in the previous versions of Protégé (versions preceding Protégé 3.0), which we suppose is the main reason that has attracted a large number of people to use Protégé together with Sesame. Sesame is a web-based architecture which facilitates storage of RDF Query Language (RQL) data and schema information. A Database Management System (DBMS) is needed to keep Sesame independent of any particular database. Sesame uses a Repository Abstraction Layer (RAL), where all the DBMS code is saved. RAL offers a resource description framework specific method to the client, so that it can call a specific DBMS. 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]. The concept of a strong query language within Protégé by connecting with Sesame, led to the idea of developing a plug-in which could bridge the gap between the entities. Problems such as usage of incompatible namespaces between the two entities and manual configurations of namespaces to connect the two entities strengthened our idea and led to the development of Passerelle. This rest of the paper is organized as follows. Description of Passerelle is provided and after that we describe the ontology that uses Passerelle. The paper finally present concluding remarks together with future research directions.

Passerelle

Passerelle bridges the gap between the two entities, Protégé and Sesame, and it exports the required Resource Description Framework (RDF) files to Sesame by communicating with the respective modules of Sesame [1]. Passerelle was developed as a tab-widget plug-in. The functionality of Passarelle can be classified into two phases, loading the RDFS file and convertion of namespace respectively. We have concentrated our work on the Admin module and have implemented Sesame's repository API so that it can communicate with the Sesame server package. We have then created a Remote Service and finally configured the appropriate parameters such as username, password and database to be used to parse the input file, see Figure 1 [1].

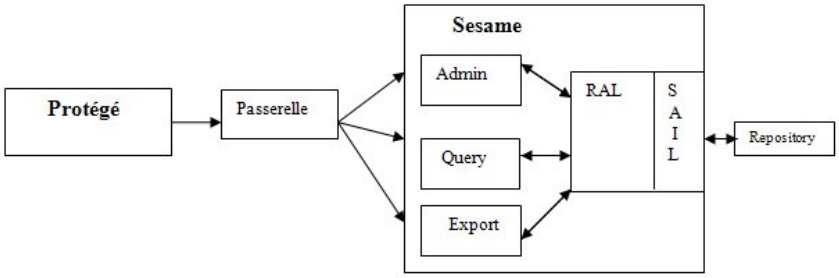

Figure 1 Architecture of Passerelle

IMPROV ontology uses Passerelle to connect to Sesame from Protégé

By way of background, we aim to support and collaborate between the distributed players in a product life cycle, so that the design of new products is augmented. We propose to do this by building an agent based system supported by a unifying ontology at both product and enterprise level. The ontology, see Figure 2 named IMPROV [3] was actually developed for a refrigeration truck system. The subclass Failure Unit will receive information from different Agent when problems exist in the unit. Condition is the subclass of Failure unit class and it has all the values from the different agents in the units [3]. The slot Error Code Id describes the type of error and the entire description of the Error Code Id exists in a table in the database so the service personal knows what type of error it is. The class Failure Unit also sends information to class Test, so the design engineer of the units can compare the values with the test values. This will allow the design engineer to have feedback, of how the unit is acting in real situation and it also provides him with the knowledge of the unit's behavior in different situations [3].

The idea is to use this knowledge to develop a better model in the future. This will allow the design engineer to obtain feedback about how the unit is performing in real conditions and in different situations.

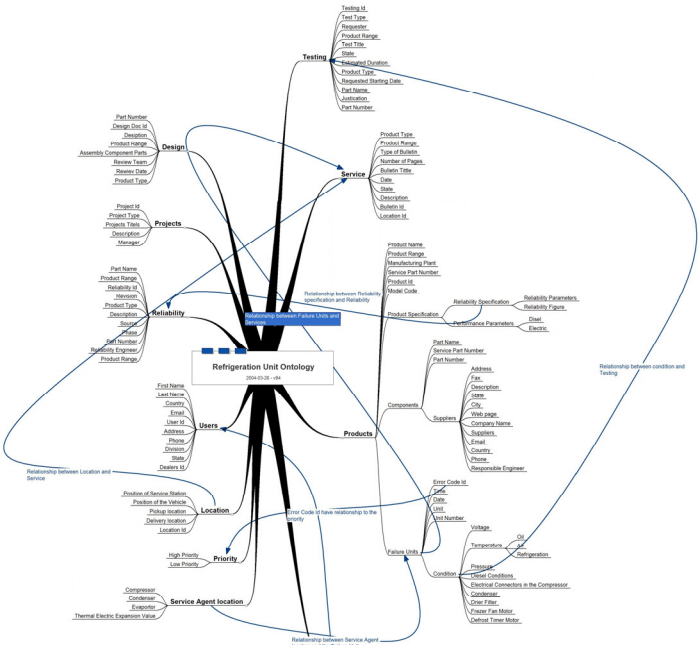

Figure 2 the mind map picture describing the IMPROV ontology

The ontology was developed using Protégé. The project has been chosen to be saved as an RDF format. This RDF file is then loaded into Passerelle and is converted to suit the Sesame configuration, see Figure 3.

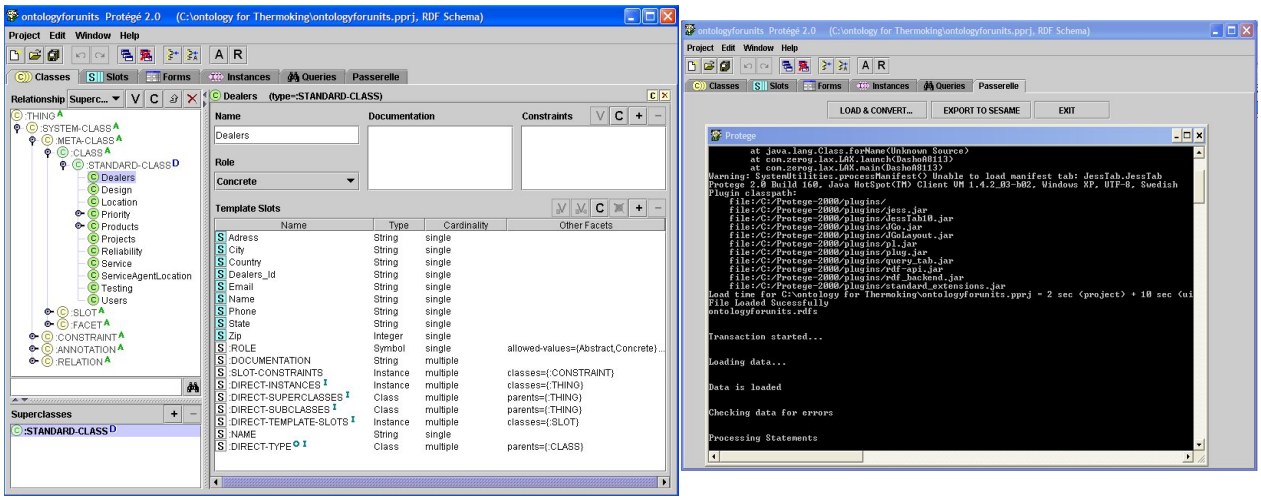

Figure 3 Convertion of RDF files and exporting them to Sesame

The database for IMPROV was developed using MySQL. MySQL was selected for the purpose since Sesame supports PostgreSQL, Oracle and MySQL. Moreover, Mysql is faster than PostgreSQL when cooperating with Sesame and the authors found it was not necessary to have Oracle in IMPROV because the database does not need advanced database systems. One other reason was the open availability of MySQL. If in the future we decide to use Oracle as a database system it will not be a problem because it is possible to implement Sesame in

a variety of repositories without the need to make changes within the Sesame API [2]. Since IMPROV is under construction we are for the moment querying our ontology through the web interface which Sesame offers. Figure 3 shows the different RQL which are possible to use in Sesame. The one we are using is called SeRQL. Before making an RQL query, one must check for the right configuration in the repository, check whether the Tomcat server is running and also check whether the database is already started. As you can see in Figure 4 the results are in html format on the query from the IMPROV databases, but it also possible to get the result as an XML format or in RDF format.

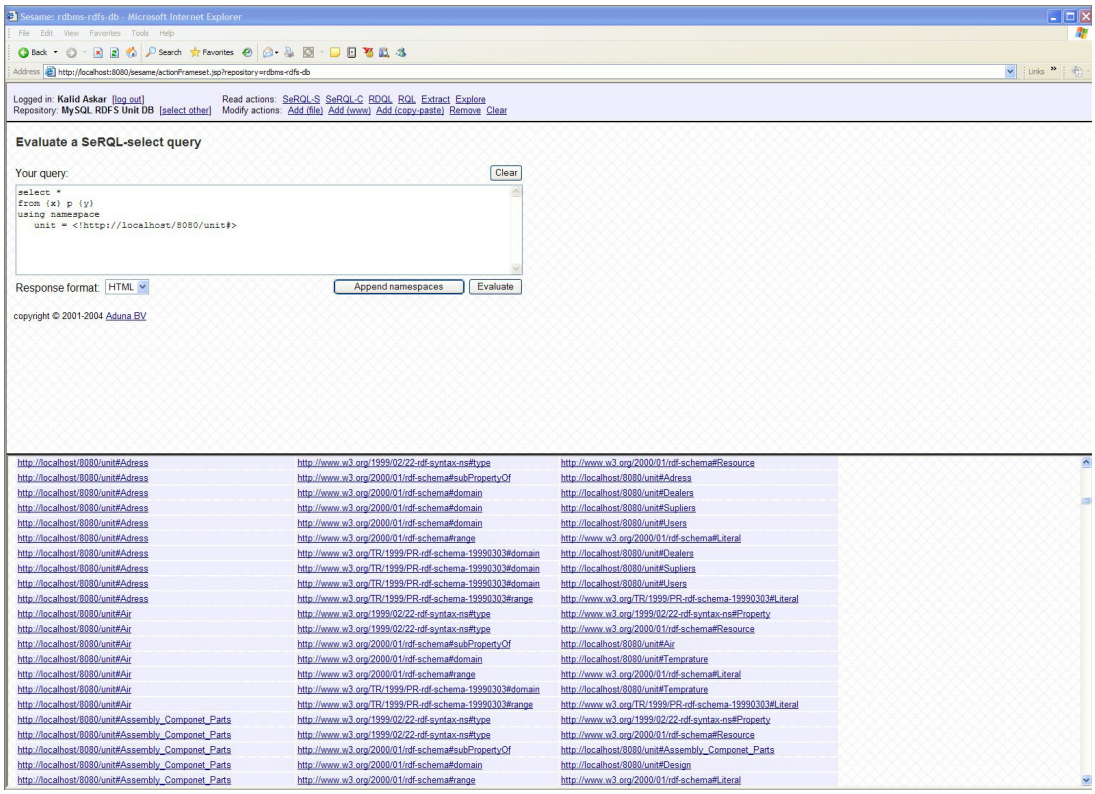

Figure 4 Using RQL to query IMPROV ontology

Conclusion

While we were developing Passerelle, there was no possibility to make RQL queries in Protégé. Hence many groups were using Protégé in combination with Sesame. But the problem of incompatible namespace always existed. Passerelle aimed at solving this problem by removing barriers to integrate between the entities. Passerelle offers ontology developers the flexibility of using Sesame while continuing to use Protégé as their ontology editor. It is important to support Protégé rather than switch to another ontology editor, since Protégé offers other advantages, such as being the only ontology editor which facilitates the development of plug-ins.

Future Work

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, which facilitates the usage of RQL without struggling with other tools. This will relieve the ontology developers of the need to perform manual configurations and will as a result, attract a wider audience to use Sesame together with Protégé.

References

[1.] Askar, K., et al. Plug-in for Protege 2000 witch supports Sesame. in 3 international Semantic Web Conference, Demos. 2004. Hiroshima, Japan.

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

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