SPARQLViz High Level Design

Werkcollege Informatiesystemen – FEW2091

Date: Tuesday, 14 March 2006 **Version:** 0.3
Team 5: Jethro Borsje Hanno Embregts Cosmas Fonville

Table of Contents

1. Introduction

For the project “Querying and Visualizing RDF data” for the course Werkcollege Informatiesystemen we will create an application. This document discusses the necessary details for building the program. We write this document so that it is clear for both parties (this group and the tutor) what the design goals are. The document also provides information which forms a basis for the application.
Our application will be a plugin for the existing program IsaViz. IsaViz is a visual environment for browsing and authoring RDF models represented as directed graphs. With our plugin we want to add a new functionality to IsaViz: the visualization of SPARQL queries. We are unaware of a current program which has the same functionality of IsaViz and SPARQLViz combined. At the time of writing there is certainly no plugin with this functionality available for IsaViz.
The plugin is useful for everyone who wants to visualize a SPARQL query with respect to the original dataset. A reason for visualizing such a query can be because the normal query results are just data and can be far from clear.

The goal of SPARQLViz in one line is: making SPARQL queries on an RDF model in IsaViz possible and visualizing the resultset with respect to the original dataset.

2. Architecture

2.1. Introduction

Our program will be a plugin for the existing program IsaViz. Because IsaViz is very important to our plugin we will describe the features of IsaViz.

IsaViz is a visual environment for browsing and authoring RDF models represented as directed graphs. It features:

- *a 2.5D user interface allowing smooth zooming and navigation in the graph*
- *creation and editing of graphs by drawing ellipses, boxes and arcs*
- *RDF/XML, Notation 3 and N-Triple import*
- *RDF/XML, Notation 3 and N-Triple export, but also SVG and PNG export*

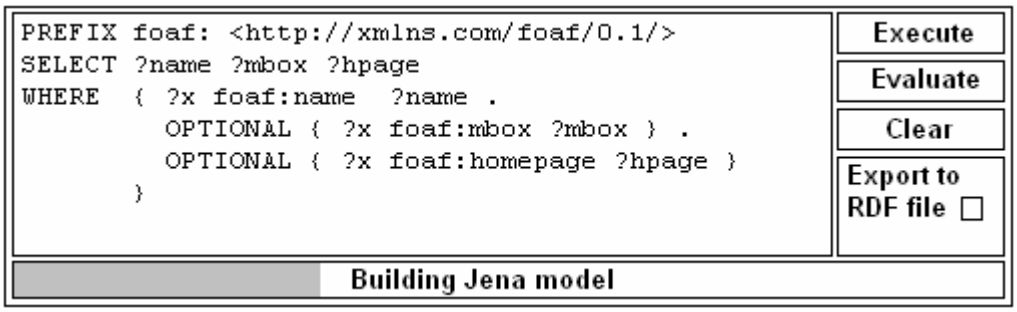
Our program is a plugin, so we can use all the functionalities of IsaViz and add the functionality of our plugin to IsaViz.

IsaViz is written in Java so it is platform independent; our plugin will also have these properties.
The input for our plugin is an RDF dataset imported in IsaViz. After this, a SPARQL query can be entered into our plugin. The output of our plugin is the visualization of the query in IsaViz with respect to the original dataset. It is also possible to output data results.
SPARQLViz will have a Graphical User Interface (GUI), which offers the following functionalities:

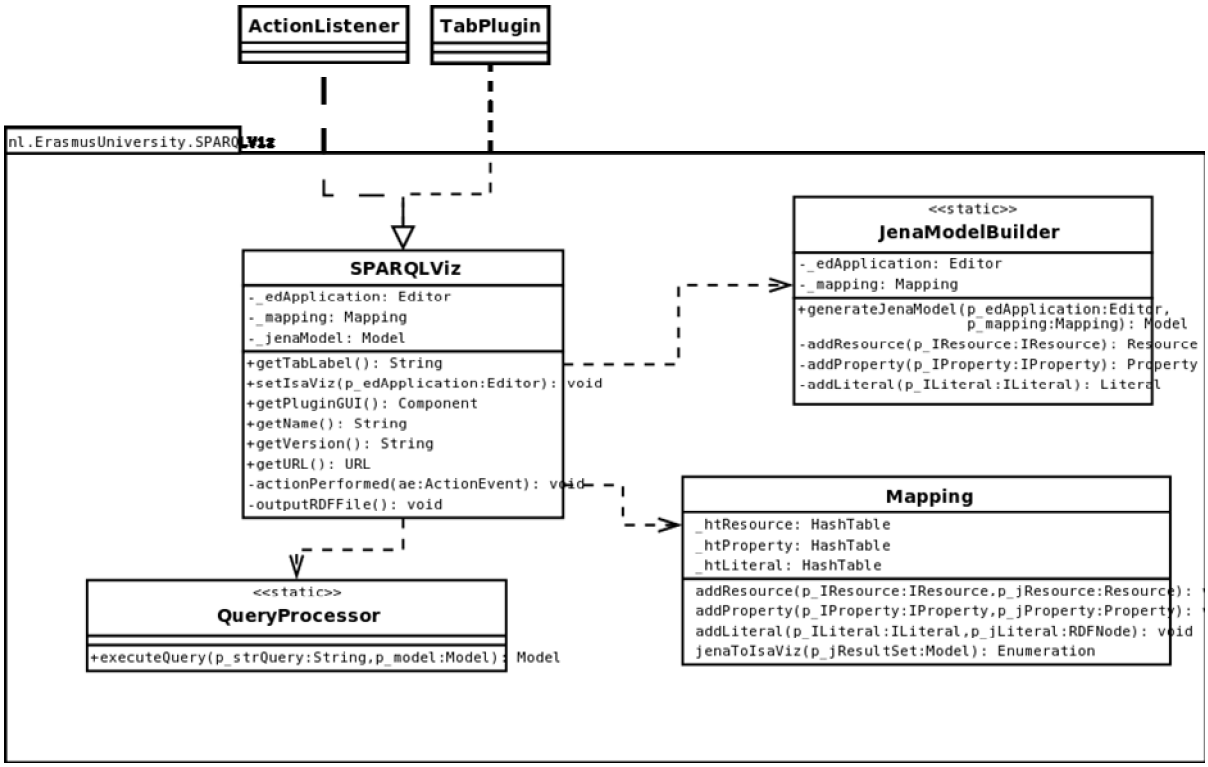
Type	**Function**
BASIC	Entering a query
BASIC	Execute the query
BASIC	Possibility to enter a new query
BASIC	Checkbox to get resultset as RDF document
OPNIONAL	Show the progress of the query (progress bar)
OPTIONAL	Validating the syntax of the query

Functionalities marked [BASIC] must be implemented in the final release. Functionalities marked [OPTIONAL] will be implemented when there is time left.

The GUI will look like this:

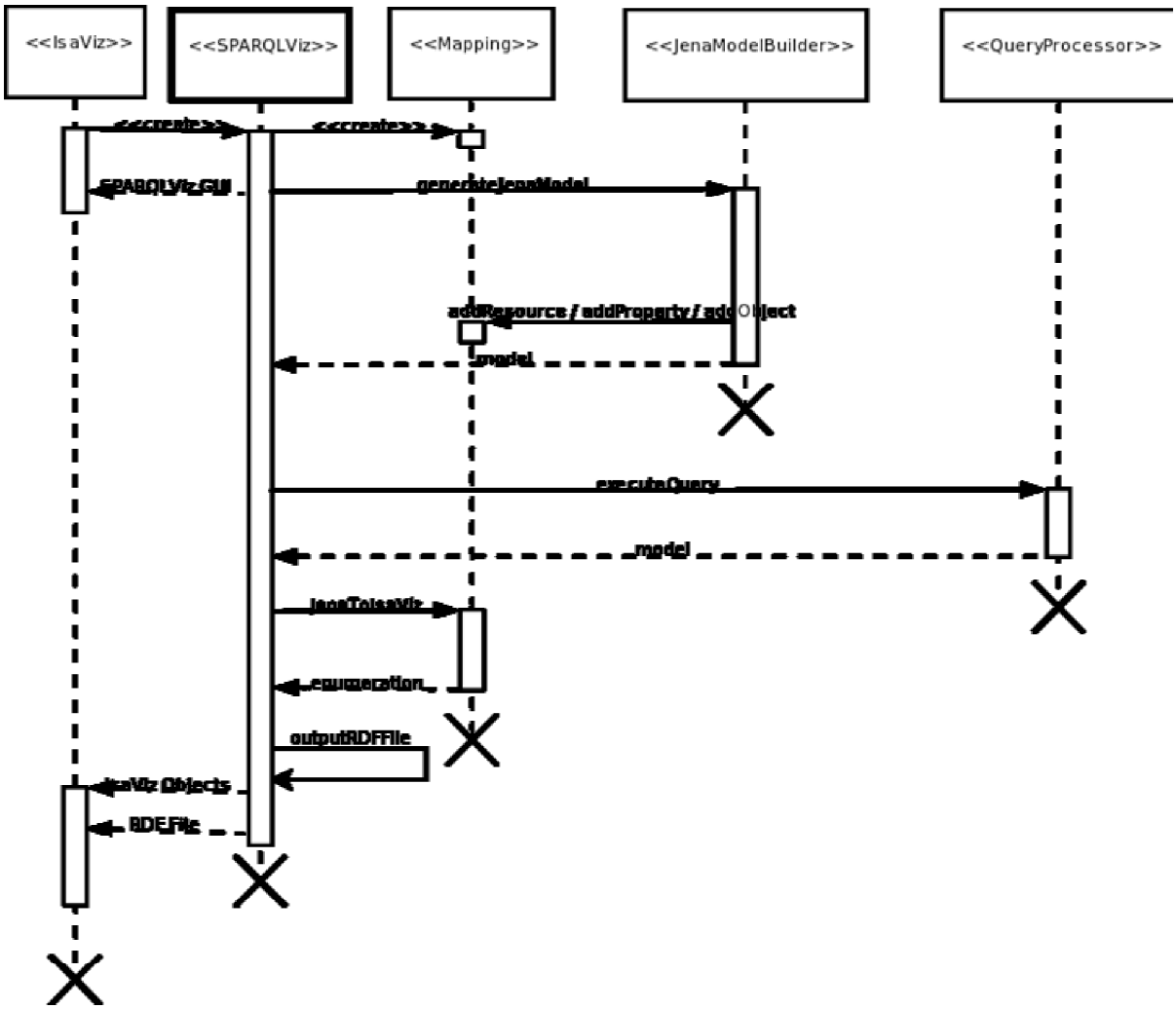


The structure of our plugin is shown in the following UML class diagram:



2.2. Communication

The working of the plugin will be explained here; however it is useful to look at the UML Sequence diagram on the next page first.
When IsaViz is started it will look for plugins in its “plugins” directory and find the SPARQLViz plugin (placed there as a .jar file). The GUI will now be loaded into IsaViz. The actual work for our program begins when the user presses a button in our GUI. Assuming the action is ‘executing a query’, there will be a Mapping object created for the mapping between IsaViz and Jena. This is necessary because both use their own way of handling RDF data. IsaViz uses an internal data structure, called IObjects. SPARQLViz will work with Jena. While SPARQLViz builds its internal Jena model, the mapping between the Jena model and the IObjects is made. This will be done in the class JenaModelBuilder. Eventually a whole Jena model is created and will be passed through to the SPARQLViz main class of our plugin. The plugin will now use the QueryProcessor, which is able to execute SPARQL queries on a Jena model. The result is a new model which is returned to the SPARQLViz main class. SPARQLViz now wants to know what the results of the query are in IsaViz (IObjects), so the SPARQL query results (which are in Jena) are now converted back to IsaViz with the mapping object. The mapping object knows which Jena object matches to which IObject because of the earlier made mapping. The result is sent back to the SPARQLViz main class and can be viewed in IsaViz.



2.3. Code

2.3.1. Introduction

SPARQLViz is a plugin for IsaViz. By our request the developer of IsaViz, Emmanuel Pietriga, made it possible for plugins to add GUI elements to IsaViz. Previously this was only possible for RDF data conversion plugins, but now all kinds of plugins can be imported. However, this is only possible in a beta version of IsaViz 3.0.
When reading the description of our program you should keep in mind that it is a plugin and not an independent program.

2.3.2. Classes

2.3.2.1. Class SPARQLViz

This is the main class and will be created by IsaViz when it is started. This class controls the working of our plugin, it is the controller class. This class creates instances of other classes. Its main function is receiving SPARQL queries and take the necessary steps to execute and visualize them.
The input of this module is a SPARQL query and the outputs are IsaViz objects which will be visualized.

This class is also responsible for showing our GUI in IsaViz.

2.3.2.2. Class Mapping

This class is created by the SPARQLViz class (main class) and its function is to map IObjects to Jena objects. It is necessary to map IObjects to Jena objects, because they differ and we need a Jena model to process the SPARQL queries. These can only be done on a Jena model. The mapping object knows which Jena object belongs to which IObject and vice versa.

2.3.2.3. Class JenaModelBuilder

This class is created by the SPARQLViz class (main class) and its function is to generate a Jena Model out of the IObjects. Each time a Jena object is created, the Mapping instance will be updated by creating a mapping between the IObject and the Jena object.
The inputs of this class are all IObjects of the current graph and the output is a Jena model.

2.3.2.4. Class QueryProcessor

This class is used by the SPARQLViz class (main class) and its function is to execute a SPARQL query on the generated Jena model. The input of this class is a Jena model and a SPARQL query. The output is the result of the query in a Jena model.

3. Operation

3.1. User Types

The users of our SPARQLViz are those who want to visualize RDF queries for a more synoptic view of the result of this query with respect to the original dataset.

3.2. Scenarios

Here follows a scenario of how a user can use our plugin.

1. User starts IsaViz
2. User imports RDF model in IsaViz
3. User clicks on our plugin tab in the Definitions window
4. User enters SPARQL query
5. User executes the SPARQL query
6. User sees a visualization of the query in IsaViz
7. User clears the query text-area

3.3. Installation

The plugin comes in a jar-file which can be placed in the plugin directory of IsaViz. IsaViz will detect the plugin and load the GUI automatically.

3.4. Uninstall

The uninstall is easy: just remove the SPARQLViz plugin from the plugin directory in IsaViz.

4. Development

4.1. Resources

We are working with three developers on this plugin. The developers should have enough knowledge and experience with Java to build this plugin. The tools we will be using are: a Java development environment (like NetBeans or JBuilder) with all the needed tools in it like an ANT and a CVS client, CVS possibilities for sharing the most recent version of the plugin, ProjectPath and an FTP-server for sharing information and files about this project.

4.2. Roadmap

Version	Change
0.1	Implement the basic plugin interface
0.2	Design an error handling framework
0.3	Build the input feature for the SPARQL query
0.4	Build conversion model for IObjects to a Jena model.
0.5	Build RDF-file return feature
0.6	Build a mapping object to map the Jena objects to the IObjects
0.7	Build SPARQL query execution processor
0.8	Map the SPARQL resultset to the IObjects
0.9	Mark IObjects in the original dataset
1.0 *beta*	Prototype
1.0	Final version
2.0	Final version implementing graphical construction of SPARQL queries (future research needed)

5. Miscellaneous

5.1. Conformance with standards

The plugin will be programmed in Java and will also be correctly commented using the JavaDoc standard. We have written a program rules document as a programming standard.

5.2. Debugging

The program may contain bugs, but if the program reaches version 1.0 beta no serious bugs are allowed. That's why there will be severe testing. Eventually the final version 1.0 will come out and also minor bugs will be filtered.

5.3. Security

By declaring the appropriate class availability, we want to secure our plugin as much as possible. This can be done by using the keywords `private`, `protected` and `public`.

5.4. Bibliography

Web site Jena: http://jena.sourceforge.net/
Web site IsaViz: http://www.w3.org/2001/11/IsaViz/