

SPARQLViz Requirements Document

Werkcollege Informatiesystemen – FEW2091



Date: 14 March 2006

Version: 0.3

Team 5:

Jethro Borsje

Hanno Embregts

Cosmas Fonville

Table of Contents

Table of Contents	2
1. Introduction.....	3
2. Application Overview	4
2.1. Objectives.....	4
2.2. Business Process.....	4
2.3. User Roles and Responsibilities.....	4
2.4. Interactions with Other Systems.....	Fout! Bladwijzer niet gedefinieerd.
2.5. Production Rollout.....	4
2.6. Terminology.....	5
3. Functional Requirements	6
3.1. Statement of Functionality.....	6
3.2. Scope	6
3.3. Performance.....	6
3.4. Usability.....	6
4. Appendices	Fout! Bladwijzer niet gedefinieerd.
4.1. Planning	7

1. Introduction

For the project “Querying and Visualizing RDF data” for the course Werkcollege Informatiesystemen we will create an application. Currently there is an application for visualizing RDF data called “IsaViz”. However, this application does not support queries on the data. Our program will be a plugin for “IsaViz” adding these properties to the program. We think this is useful because it is common that you only see the resultset of the query, this makes it hard to relate it to the original dataset.

This document states *what* the software will do once it is finished. We write this document so that it is clear for both parties (this group and the tutor) what the application will do. The document also provides information which forms a basis for the project.

2. Application Overview

2.1. Objectives

We have chosen for this assignment because we think it is an interesting goal to write a program to visualize and query RDF data. This program will make the RDF data and queries more synoptic. We found a program called IsaViz which visualizes RDF data. We think it's a great idea to add functionality to IsaViz by writing a plugin for this program. By this way we can add the following functionality to IsaViz: visualizing SPARQL queries with respect to the original dataset. We think this is useful for people who query RDF datasets and want a more synoptic view of the query in respect with the original dataset.

We are doing this project for the course Werkcollege Informatiesystemen of the study "Informatics & Economics". By working on this application we will gain experience with Java, RDF, SPARQL, Jena, DOT and more. This is useful for our study and perhaps for future job opportunities.

2.2. Business Process

Currently it is hard to visualize the result of SPARQL queries with respect to the RDF dataset. It is common that you only see the resultset of the query and this makes it hard to relate it to the original dataset. This can make the resultset of a SPARQL query harder to understand. The application we are developing will add visualization to SPARQL queries in respect to the original dataset. With IsaViz it is possible to import a RDF dataset and visualize it. Our plugin makes it possible to execute a SPARQL query on the dataset. As a result you will see the resultset of the query with respect to the original dataset.

2.3. User Roles and Responsibilities

The users of the application are people who work with RDF data and who want a visualization of their SPARQL queries in respect to the original dataset. The users must have some knowledge of the semantic web, RDF data and SPARQL queries.

2.4. Resources

IsaViz is written in Java and is platform independent, our plugin SPARQLViz will also have these properties. The plugin uses Jena for making it possible to query the RDF dataset. SPARQL is used for the querying. IsaViz will display the output of our plugin. An overview of how the program interacts with IsaViz, Jena and SPARQL can be found in the Design Document.

2.5. Production Rollout

The milestones of the plugin production are listed in the next table.

Date	Milestone
23 January 2006	Start developing program, all the essentials will be programmed first.
14 February 2006	A beta build must be finished with the basic functionalities. See chapter 3.1 for the functionalities overview.
14 March 2006	Final program must be finished.

2.6. Terminology

RDF: The Resource Description Framework is a standard for describing resources.

RDF graph: a set of triples; each triple consists of a *subject*, a *predicate* and an *object*.

SPARQL: is a query language for getting information from RDF graphs.

Java: an object-oriented programming language which is platform independent.

Jena: a Java API which can be used to create and manipulate RDF graphs. Jena has object classes to represent graphs, resources, properties and literals.

3. Functional Requirements

3.1. Statement of Functionality

Here we determine what the program will do, which functions it will have. We also list additional functions; these will be implemented when there is time left. Here follows an overview.

Type	Function
BASIC	Read RDF files in XML/RDF serialization.
BASIC	Convert XML/RDF serialization to N-TRIPLE serialization
BASIC	Draw initial data set
BASIC	Textual input possibility of a SPARQL query
BASIC	Visualization of the query results in respect to the initial dataset, drawn in a graph
BASIC	GUI which includes above described functions
OPTIONAL	Visualize the process of creating the SPARQL query

Functions with a [BASIC] tag must be implemented and functions with an [OPTIONAL] tag will be implemented when there is time left.

3.2. Scope

The BASIC functions described in section 3.1 will be implemented as soon as possible and all BASIS functions will be included in the final version. The OPTIONAL functions will be implemented if there is time left but they are not required for the program to run properly. Our program *does* works with RDF files but *doesn't* work with OWL files.

3.3. Performance

The performance functional requirements will be filled in later when we have more experience with RDF / SPARQL visualization.

3.4. Usability

There will be a GUI for the plugin. The GUI will make it possible to easily enter and execute a SPARQL query. For an example of the GUI please see the design document.

4. Miscellaneous

4.1. Planning

Here you will find the finishing dates of activities concerning the application.

Date	Activity
31 January 2006	Finished requirements document
6 February 2006	Finished design document Finished UML diagrams
14 February 2006	Finished prototype of application
7 March 2006	Finished beta version of application
14 March 2006	Finished application