RDF Gravity ++: Tool for Authoring and Visualization of Semantic Web Ontologies, Schemas and Repositories

Nitin Arora
Knowledge based Information Systems
Salzburg Research Forschungsgesellschaft m.b.H.
Salzburg Austria.

Abstract

The **RDF GRA**ph **VI**sualization **T**ool ++ (RDF Gravity ++) visualizes RDF, RDFS and OWL as directed graphs and provides authoring environment with a graphical user interface. The tool provides features like filtering, zooming, undo/ redo, relating selected or similar objects, querying and full text search, saving graphical state information, visualizing specific parts or fragments of RDF Graphs as sub-graphs and connectivity to relational databases.

1 Introduction

RDF Gravity ++[RDFG] is an authoring and visualization tool built on JUNG API [JUNG] and Jena [JENA] for building and viewing RDF model as a graph. The application provides an easy to configure graphical environment for constructing and viewing Resource Description Framework (RDF) [RDF] and Web Ontology Language (OWL) [OWL] as graphs.

The application displays the graphs using the basic geometrical shapes (Rectangles, Connectors or Edges, Triangles). It displays Instances as Triangles, rest of the RDF Nodes as Rectangles and RDF Properties which is referred from a certain vocabulary as connectors or edges. It also distinguishes between Concepts, Properties, Literals, URI Resources, Anonymous Nodes and Instances using separate icons for each of them. The graphical figures are assigned according to a color and shape scheme on the group to which it belongs (For e.g. Literal Rectangles are shown in green). The selection of different icons, colors and graphical shapes make the graph more comprehensive and very easy to interpret.

2 Salient Features

RDF Gravity ++ provides the following features of the tool:

2.1 Complete Authoring and Visualization Environment

The tool provides a complete authoring and visualization graphical user interface for addition, deletion and editing of RDF nodes and edges (RDF properties).

2.2 History

Users need support for undo, replay, and progressive refinement. It is therefore important to keep a history of user actions and allow the user to manipulate it. RDF Gravity ++ do keep a record of recent node and edge addition/deletion actions. The user can again perform the recent undone action by using redo utility of the tool (in unsupervised layout mode).RDF Gravity ++ uses a dynamic layout algorithm and so it does not keep a record of geometrical manipulation (like scramble, moving a node from one position to other etc.) Therefore, the geometrical manipulation actions can't be undone or redone.

2.3 Query

The results of visualization operations need to be extracted from the visualization. This might be a subset of the data or the query parameters and operations which produced the results. In this way, users can extract subsets of the data for further analysis and apply the results of an analysis to other data sets.

A sub graph on the basis of a query can be extracted from the entire RDF graph. There are no changes to visual representation or layout of the new sub graph.

RDF Gravity ++ provides support for free text as well as RDQL [RDQL] queries.

2.4 Zooming

The RDF Graph which is displayed on the screen can be zoomed geometrically using a slider button on the GUI. If the zoomed view crosses the screen boundaries, the screen is supported with scroll bars. The GUI support provided for zoom doesn't support area selection. In area selection zooming, the portion (area) of the screen to be zoomed can be selected using a mouse.

2.5 Filter

In addition to the graphical data set reduction accomplished by zooming, users often need to reduce the data set size by eliminating items based on their attributes. The application supports filtering based on following attributes:

- Filter or Show URI Resources, Anonymous Nodes, Concept Nodes, Property Nodes, Instance Nodes, Literal Nodes and zero edge nodes.

- The RDF Nodes in the graph are filtered based on the namespace and are grouped in a tree. Each RDF Node is represented as a tree element and can be filtered on selection.
- RDF Gravity ++ also filters out RDF Properties and organizes them in a tree based on their namespace.
- The RDF Nodes and Properties can be filtered out and shown in a list view based on the query fired to the RDF model.

2.6 Details on demand

The user can configure the application to show all edges and to show full labels. If the user has selected to show all edges, then the local name of the properties which are represented as connectors (Arrow Line) is shown on the screen with the connector. If the user chooses to view full labels, then the long labels like URI of URI resource are shown fully and not in an abbreviated form. The application also supports information presentation through roll over. If you roll your mouse over a RDF figure, then URI is shown for a resource (id is shown if it's a blank node), local name is shown for a property and literal value is shown for a literal.

2.7 Relate

If users discover an item of interest, they may need to know about other items with similar attributes. Alternatively, items in visualization could be related through being part of the same process or event (for e.g. a visit to the doctor in a medical database). Users would want to know about related events or items. Here, clicking on one item would highlight related ones.

RDF Gravity ++ supports "Relate" on the basis of connectivity i.e. "geometrical relate". There is no support provided for "semantic relate". Semantic relate can be exemplified like, if there exists a Concept about Mammals, then highlight all nodes which are somewhat related to mammalian activities or events. The above process can be achieved by firing some queries to the RDF. Relate can also be formulated using the concept of *Concise Bounded Description* [CBD]. In the RDF Gravity ++, when a RDF Node is double clicked, all the nodes which are directly connected to this node are also highlighted.

2.8 Graphical State information

Graphical state information such as location is tracked whenever a user performs changes and is saved whenever the user exits the session. This saved information is used to provide the same location and look and feel of the data when the same file is used.

2.9 Support for opening/ feeding from a URL

RDF Gravity ++ provides support for reading RDF/ OWL data from a URL and visualizes it.

2.10 Connectivity with Relational databases

RDF Gravity ++ provides support for relational database connectivity using the D2RQ [D2RQ] map which maps relational data to RDF for visualization.

3 Comparative Analysis

A comparative study [CompAnalysis] has been done with other visualization tools like HP Node Centric Visualization Tool [HPNCV] and ISA-VIZ [ISA-VIZ]. Overall RDF Gravity ++ compares favorably with its peers particularly through its authoring facilities and comprehensive visualization.

4 Conclusion

RDF Gravity ++ provides a complete authoring and visualization environment for software designers and developers in the semantic web community for developing ontologies, schemas and instances in RDF and OWL. The original motivation for building the tool was essentially to debug RDF schemas and RDF knowledge bases and to aid understanding of new/ unknown ontologies and schemas.

References

[RDFG] Sunil Goyal, Rupert Westenthaler, Nitin Arora, Salzburg Research Forschungsgesellschaft m.b.H., Salzburg. RDF Gravity (Only Visualization version); http://semweb.salzburgresearch.at/apps/rdf-gravity/.

[JUNG] Java Universal Network/Graph Framework.

http://jung.sourceforge.net/

[JENA] HP Labs Semantic Web Programme. Semantic Web Framework for Java.

http://jena.sourceforge.net

[RDF] Resource Description Framework.

http://www.w3.org/RDF/

[OWL] Deborah L. McGuinness (Knowledge Systems Laboratory, Stanford University), Frank van Harmelen (Vrije Universiteit, Amsterdam).Web Ontology Language. http://www.w3.org/TR/owl-features/

[RDQL] Andy Seaborne, HP Labs Bristol. A Query for RDF. http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/

[CBD] Patrick Stickler, Nokia. Concise Bounded Description. http://sw.nokia.com/uriqa/CBD.html.

[D2RQ] Chris Bizer - Institut für Produktion, Wirtschaftsinformatik, Freie Universität Berlin. In *Treating Non RDF Databases as Virtual RDF graphs.*

[CompAnalysis] Nitin Arora, Kunal Chawla, Salzburg Research Forschungsgesellschaft m.b.H. Salzburg. In *Comparative analysis of visualization tools for semantic web frameworks.* Work in progress.

[HPNCV] Sayers Craig, HP Labs Bristol. In *Node Centric RDF Graph Visualization*, HP Technical Reports.

[ISA-VIZ] A Visual Authoring Tool for RDF.

http://www.w3.org/2001/11/IsaViz/