

## **Refined Problem Statement**

### **1. Problem**

This tool would be very useful for the end users who are interested in viewing specific regions in the ontology. The tool would have a class tree view and property tree view as in protégé. This tool would be very useful for the end users who are interested in viewing specific regions in the ontology. The user will be viewing the condensed instance graph initially. The tool would have a class tree view and property tree view as in protégé. The user would be able to choose a node and or property (relationship) from the instance graph either from the instance and property view or by other ways of providing input (could be a dialog box or through a select option in popup menu when right-clicked on a node or property). The user would be able to view all paths that the selected property was involved between any two nodes selected. Hence, the tool would help to find out the semantic associations between any two entities.

Browsing mechanisms: To help user in browsing ontology tool would have an option for browsing ontology neighborhood based on the root node and the number of hops or distance from the root node. Once the user selects the node & number of hops the query interface would be able to return the neighborhood of the selected node. Our tool would have option of being able to cluster certain complex instances based on the relationship (property) which the user would provide. The tool would provide an option of being able to cluster certain node based on the relationships which the user has selected to a single node (hyper graph or clustering of nodes) & later being able to expand the node by means of some zooming techniques.

Users would have option of being able to zoom-in & zoom-out from the clustered node. This option would be very useful in filtering unwanted edges in the instance graph and would help users to view relevant information. Additionally, the tool should be able to provide a class hierarchy view of a particular instance & also the property view of the instance. Property view would be helpful in knowing which relationships have been inherited & which other relationships belong to that class. The tool should be able to hide, collapse, a node or a given edge, the tool might have a fisheye view for viewing.

Currently, there isn't any existing tool that is able to provide these functionalities. This tool would be very handy to researchers in bioinformatics when they are trying to discovering biological networks – created through relationships between classes derived from instances.

## **2. Objectives**

The objectives of this project are:

- To help researcher view the contents of the ontology in a more meaningful manner.
- To help researcher view specific regions of the ontology either by means of a query or by providing better navigation techniques to browse through the ontology.
- To help researchers view the contextual paths that exist between two nodes based on certain relationship.

## **3. Functional Requirements**

- Researcher should be able to view the schema
- Researcher should be able to cluster or create a hypergraph node from the ontology schema graph based on certain relationships
- Researcher would be able to browse through the graph and view specific concepts or classes in the ontology
- Researcher would be able to view paths between two nodes based on the selected relationship. He would be able to view the entire path for which the selected relationship was involved between two nodes
- Provide an easy way of navigation through the ontology

## **4. Nonfunctional Requirements**

### **1. Usability:-**

The user of the system is a researcher and has knowledge of schema and instances. The User interface would be built by looking at other tools like protégé, Isaviz, protégé pugins Like OntoViz, Jumbalaya, TGvizTab etc. There other touch graph based tools available.

### **2. Reliability:-**

The system should be able to handle exceptions and should be able to display the graph without causing errors

### **3. Performance:-**

The system shouldn't take much time to display. The responsive time of the system would be dependent on the file size.

### **4. Supportability:-**

The extensions to this system would be the following:

- Being able to provide the instance view of the ontology
- Creating hypergraph of instances

- Given a instance the user should be able get the class hierarchy (i.e. its type & the class hierarchy till the root or thing in RDFS)
- Being able to view contextual path between a given two instances. The ranking of paths could be done based on the contextual weights of the edges & the node in the path.
- Being able to add different layouts for view large graphs
- Provide fisheye views for efficient view and navigation through the graph

### **5. Implementation:-**

There are no constraints with respect to the hardware platform the software would be running. We would be using Java Swings for building the GUI, Yfiles or JGraph libraries for drawing the graphs & layouts, Poseidon IDE for drawing UML, and NetBeans java editor. We might be using a temporary cache API available in Java i.e. JCACHE if required. Jena 2 API will be used for retrieving the RDF & OWL triples & for querying.

### **6. Interface:-**

Not required

### **7. Operation-**

The users of the system would be the one managing the tool.

8. Licenses would be provided for academic use only.

### **5. Target Environment**

- This is a standalone application and should be capable of running on a basic configuration computer.
- It should be able to run on any platform or operating system as the code would be written in java.
- If the user wants to load very large files and wants to visualize it, then he must use a top of the line computer with a memory of above 1GB.
- User should have jdk installed in his system and should be able to set the class path to class files, while installing if necessary.