Visualization Module of Vijjana, a Pragmatic Model for Collaborative, Self-organizing, Domain Centric Knowledge Networks

Suhasini Kalluru

Problem Report submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Electrical Engineering

Approved by

Dr. Yenumula V. Reddy, Ph.D., Chair
Dr. James D. Mooney, Ph.D.
Dr. Sumitra Reddy, Ph.D.

Lane Department of Computer Science and Electrical Engineering
Morgantown, West Virginia
2008

Keywords: Vijjana, Visualization, Knowledge base, XML, Prefuse
ABSTRACT

Visualization Module of Vijjana, a Pragmatic Model for Collaborative, Self-organizing, Domain Centric Knowledge Networks

Suhasini Kalluru

Humans are capable of doing many things, but a computer cannot. This happened to be a word in the past. Now the world is heading towards a Semantic Web. The Semantic Web was originated from the thoughts of Tim Berners-Lee, inventor of the WWW, URIs, HTTP, and HTML. Semantic web links the data that is easily recognized by machines. Two strings spelled similarly but used in different contexts can be found easily without a tedious search among many irrelevant results given by a regular search engine now.

As a part of the semantic web a small knowledge network, ‘Vijjana’ is being evolved. Vijjana is defined as a Pragmatic Model for Collaborative, Self-organizing, Domain Centric Knowledge Networks. The term knowledge network refers to the fact that the links referred to as the JAN’s in Vijjana are organized in a predefined taxonomy and linked semantically. The visualization module of Vijjana is presented in the current report. The Vijjana knowledge network database source is mainly the Open Directory Project that provides classified data. The classified data is categorized based on their relationships between the data and neatly organized. The ultimate data was visualized using certain tool like Prefuse. XML’s were generated using programming languages and used for the visualization. Different visualizations were extracted to provide a clear view of the integrated data and a more enhanced search for the users.
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who supported me in completing this problem report. Formerly I thank my Committee Chair, Dr. Yenumula V Reddy for his constant supervision and support. Vijjana was born from his thoughts and ideas of the Vijjana team. His guidance and help was always there to implement our ideas and come up with new ones.

I thank Dr.James Mooney and Dr. Sumitra Reddy for their encouragement and for accepting to serve on the committee.

I thank all the members in the Vijjana team who helped me in completing this report. I would like to specially thank Luyi Wang, Amara Satish, Krishna Mohan, and Gaurav Narkhede who extended out their selfless help in completion of my project.

I would like to mention my friends Raj Kumar Gondle, Rupesh Kumar Gondle, Hari Krishna Sallana, Shilpa Bollineni, Divya Doma, Shruthi Ratnagirishwaran, Shashidhar Muppidi, Nagaanupama Akuraju, Priyanka Namburi who were always supportive and inspiring.

Finally I am very thankful to my parents Mr.&Mrs Siva Shankar Reddy.K, and my sister Sirisha.K without whom I would not have reached this stage of my life.
# Table of Contents

ABSTRACT ................................................................................................................................. ii

ACKNOWLEDGEMENTS ............................................................................................................... iii

1. INTRODUCTION ..................................................................................................................... 1
   1.1 Background ......................................................................................................................... 1
   1.1 What is Vijnana? ............................................................................................................... 2
   1.2 Problem Statement ........................................................................................................... 2
   1.3 Implementation of the visualization ................................................................................ 3

2. Vijnana Model .......................................................................................................................... 4
   2.1 Vijnana Structure .............................................................................................................. 4
   2.2 Elements of Vijnana ......................................................................................................... 5
      2.2.1 Taxonomy and the Semantic Net of Knowledge – ‘T’ and ‘R’ ............................. 5
      2.2.2 The Discovery Agent – ‘dA’ ................................................................................... 5
      2.2.3 The Organizing Agent – ‘oA’ .................................................................................. 6
      2.2.4 The Consistency and Completeness Agent – ‘cA’ ................................................. 7
      2.2.5 The Visualization Agent – ‘vA’ ............................................................................. 7
      2.2.5 Search Agent – ‘sA’ ............................................................................................... 8
      2.2.6 Rating Agent – ‘rA’ ............................................................................................... 9
   2.3 Vijnana Architecture ....................................................................................................... 9

3. Vijnana Database & XML .................................................................................................... 14
   3.1 Introduction ......................................................................................................................... 14
   3.2 Pictorial Representation of Vijnana database ................................................................. 14
   3.3 Description of the Structure ........................................................................................... 16
      3.3.1 Properties ............................................................................................................... 16
      3.3.2 Status ....................................................................................................................... 17
      3.3.2 User ......................................................................................................................... 17
      3.3.4 User and the Jan .................................................................................................... 17
   3.4 XML Technology ......................................................................................................... 18
      3.4.1 Sample structure of XML .................................................................................... 19

4. Visualization ......................................................................................................................... 21
4.1 Prefuse Toolkit .................................................................................................................... 21
4.2. Different visualizations ...................................................................................................... 22
  4.2.1 Tree View...................................................................................................................... 22
  4.2.2 Radial Graph view ....................................................................................................... 22
  4.2.3 Graph View ................................................................................................................ 22
4.3 Prefuse Structure and its implementation in Vijjana ........................................................... 23
4.4 Programming Technique ..................................................................................................... 24
  4.4.1 Initialization phase ........................................................................................................ 25
  4.4.2 Connection phase .......................................................................................................... 25
  4.4.3 Query Phase .................................................................................................................. 25
  4.4.4 Graph view Phase ......................................................................................................... 27
  4.4.5 Visualization Phase ....................................................................................................... 28
4.5 Embedding into the web page ............................................................................................. 28
5. Implementation of Visualization ........................................................................................... 29
  5.1 Prefuse Toolkit .................................................................................................................... 29
  5.2 MySQL Database ................................................................................................................ 29
  5.3 NetBeans IDE ..................................................................................................................... 29
  5.4 Eclipse ................................................................................................................................ 29
  5.5 Platform ............................................................................................................................. 29
6. RESULTS.................................................................................................................................. 30
  6.1 Prefuse Tree view ................................................................................................................ 30
  6.2 Prefuse Radial Graph view ................................................................................................ 31
  6.4 Web Embedded views ....................................................................................................... 32
    6.4.1 Tree ............................................................................................................................... 33
    6.4.2 Radial Graph View ....................................................................................................... 34
7. Future Work .......................................................................................................................... 35
8. CONCLUSION ......................................................................................................................... 36
References ..................................................................................................................................... 37
Table of Figures

Figure 1: A proposed View of Vijjana Knowledge Network Web Interface adding a JAN .......... 6
Figure 2: A Sample Graph View .............................................................................................. 7
Figure 3: A Sample Tree view .................................................................................................. 8
Figure 4: Vijjana Architecture .................................................................................................. 9
Figure 5: Overall architecture of Vijjana client program .......................................................... 11
Figure 6: Present view of Vijjana ............................................................................................ 12
Figure 7: Proposed web interface view of Vijjana ..................................................................... 13
Figure 8: Overall representation of the database ...................................................................... 15
Figure 9: View of a sample XML ............................................................................................. 20
Figure 10: Sample Graph view .................................................................................................. 23
Figure 11: Diagram depicting the relation of different prefuse packages and classes to the infovis reference model. ............................................................................................................................ 23
Figure 12: Diagram explaining the node table and edge table .................................................... 27
Figure 13: Prefuse Tree View .................................................................................................... 30
Figure 14: Prefuse Radial Graph View ....................................................................................... 31
Figure 15: Embedding the views in two buttons ....................................................................... 32
Figure 16: Web Embedded Tree View ...................................................................................... 33
Figure 17: Web Embedded Radial Graph View ........................................................................ 34
1. INTRODUCTION

1.1 Background

Google is a powerful search engine that is being used around the world extensively. The other search engines like Yahoo, Answers.com have also made their mark. Types in the keyword you want, the search engines will pop-up results related to them. But some of them would be irrelevant making the search complicated. It would be a better idea to group the similar things into one group i.e the web pages are linked semantically.

Imagine a scenario, a software programmer was assigned a new task. The project was developing SOAP based web services for a client. When he/she tries to search through the search engines, he/she might get results related to detergents, facial soaps etc. Because of the different semantic associations of the word “soap,” the results you receive are varied in relevance.

However, in a Semantic Web-enabled environment, you could use a Semantic Web agent to search the Web for “SOAP” where SOAP is a type of technology specification used in Web services. This will give relevant results this time. The semantic web agent also finds out if there has been any ‘soap’ related research and if anyone has worked on it earlier. It also finds the related technologies to ‘soap’. Now that one understands that WSDL, XML, and URI are all technologies related to SOAP, they can start off the project with a basic research on these technologies. Armed with the information returned by your Semantic Web agent, you read the related technology specifications and send emails to the colleagues who have worked on SOAP-related materials available on network to ask for their input before starting your new project.
As stated in Dr.Reddy’s et.al paper [1] consider the example of the two professors who are searching for a common thing. It would a better idea if the two could meet up regarding the common subject. To meet these Vijjana was developed as a platform for an efficient, effective, time saving search.

1.1 What is Vijjana?

‘Vijjana’ is a Sanskrit word that represents collective knowledge created through classification and analysis. Vijjana is a Pragmatic Model for Collaborative, Self-organizing, Domain Centric Knowledge Networks. The metadata of the related web pages is stored in a central database. The framework of Vijjana is designed in such a way that the users can also act as contributors. It also provides a base for sharing, contributing and discussing. The Vijjana database grows as more users share their knowledge.

Regular search engines like Google cannot understand the keywords it searches for. The main idea behind Vijjana is to make machines understand as what we do. It provides a personal space for each individual registered as a user. The knowledge network is so designed and exposed such that the semantic relations embedded are obvious. The phrase knowledge network refers to a collections of URLs organized according to a predefined taxonomy and associated link semantics.

1.2 Problem Statement

‘Vijjana’ is a social network that is designed aiming to make the easy search. User interactive controls are provided only when the core part of the knowledge network is efficiently and smartly shown to the users. The main idea was to develop a program that focuses on presenting
the database on a global space, so that the users can easily use and manage their own personal space. The visualization is the main medium that influences and conveys the actual goal and usefulness of the knowledge network. The visualization ways is many, of which few are described later in the document.

1.3 Implementation of the visualization

Following are the steps for the implementation.

1. The database was schemed using the relational database schema. The data was fetched from the database.

2. The data was then divided into two tables- the node table and the edge table.

3. The SQL queries are issued on the database.

4. The data was grabbed into the tables that are initialized earlier.

5. This gives an output as a graph in the XML.

6. The graph is then given as an input to the prefuse tool.

7. An embedded class is used to view this graph in the visualized form.

8. An applet is used to create the view in a separate window.
2. Vijjana Model

2.1 Vijjana Structure

The Vijjana was modeled using the following elements.

\[ \text{Vijjana-} \{ \text{J, T, R, dA, oA, cA, vA, sA, rA} \} \]

where

- ‘X’ = the domain name
- ‘J’ = the collection of Jans in the Vijjana-X
- ‘T’ = the Taxonomy used for classification of Jans
- ‘R’ = the domain specific relations
- ‘dA’ = the discovery agent which discovers relevant Jans
- ‘oA’ = the organizing agent which interlinks the Jans based on R
- ‘cA’ = the consistency/completeness agent
- ‘vA’ = the visualization agent
- ‘sA’ = the search agent
- ‘rA’ = the rating agent

The different agents and the parts of the Vijjana are described in section 2.2.
2.2. Elements of Vijjana

2.2.1 Taxonomy and the Semantic Net of Knowledge – ‘T’ and ‘R’

The first stage was to define the Jans and interrelate them to form a semantic net of the URL’s. The taxonomy is defined in order to develop the knowledge database. A Jan can be related to many others to form many groups of Jans. Hence the Vijjana model has wide variety of taxonomies considering them as parameters that are modified as the knowledge network develops. The database can be imported or exported easily then.

The database was designed in such a way that Vijjana database is updated as new Jans are added and visualizes as it grows into a bigger network. The links are removed when they are in active.

2.2.2 The Discovery Agent- ‘dA’

The discovery agent plays the role of collecting the Jans from the web world. The agent can either be human or computerized. Since Vijjana is in its initial stages of construction the discovery agent is a manual. The agent finds the agents and marks them so that they can be organized into a node (A node on a graphical taxonomy is one that displays its hierarchy when clicked on it). Another way of adding the Jans to the node is by an automated program or a script.

In the near future, one can subscribe to the RSS feeds on related websites to automatically add the Jan to the database network. Collaborators can contribute their relevant JAN’s by email, following a predefined syntax of the message to enable automatic parsing of Email.
2.2.3 The Organizing Agent – ‘oA’

The Jan obtained by the discovery agent is forwarded to the organizing agent. The Jans can be obtained in any of the following ways as described above- RSS feed, Email, using mark-up button by a user, Vijjana client interface. It is the responsibility of the agent to ensure that is an active link and comes from a trusted source, then classifies the Jan into its node. The organizing agent can also make reverse links with the information.

In a case, a Jan describes the capital city of a state. So, it is linked as the “capital of” relation with the state. In the reverse context the relation ‘has’ can be used to represent that the State has this city as it’s capital. When the capitals’ Jan is incorporated into the Vijjana network it is automatically linked using the forward and the reverse links. Thus the knowledge network grows. The Jan information is logged into a provenance file for generating the history. The ‘oA’
agent notifies the registered users that a particular new Jan has been added to the knowledge network. The rating agent provides the number of visits to a particular Jan to the organizing agent.

2.2.4 The Consistency and Completeness Agent – ‘cA’

The consistency and completeness agent ‘cA’ is responsible for the reliability of the network as a whole. It checks frequently for the dead links, removes them and tries to complete a jan that is to be linked to the network of Jans. He highlights the isolated link that it captures the attention of any user who might be able to provide information to complete its integrity with the knowledge network.

2.2.5 The Visualization Agent –‘vA’

The visualization agent ‘vA’ is responsible for the visual demo of the Vijjana knowledge network. The prefuse tool is the basis for the visualization in Vijjana that offers many kinds of visualizations. The tree view and the graph view are present below.

![Figure 2: A Sample Graph View [2]](image)
It can be seen that when the node is clicked it expands showing its children. Thus the network is seen growing as the further nodes are clicked.

2.2.5 Search Agent – ‘sA’

The user can make a keyword based search or an advanced search. A whole command can be used to make a search that satisfies the given criteria, the search agent would be able to provide you with the results. This however is only possible if the semantic links are completely specified.
2.2.6 Rating Agent – ‘rA’

The rating agent is a key to capture the attention of the users towards high value Jans. A particular Jan with many included links useful to many users can be highlighted to draw more attention. The value of the Jan can be based on the user rating that uses a combination of user specified values, number of “hits”, and the number of fully specified relational.

2.3 Vijjana Architecture

![Vijjana Architecture Diagram]

*Figure 4: Vijjana Architecture [1]*
Figure [2.4] illustrates the Vijjana architecture. System 1 is the administrator interface on the left side and System 2, the client interface is on the right hand side. The consistency agent ‘cA’, the organizing agent ‘oA’ constitute the administrator actions. The users of System 2 can view the entire knowledge network through the visualization provided by the visualizing agent ‘vA’. The users can browse through the knowledge network and in return might add more JANs to the network.

The Vijjana also provides a base for social networking. Users from different fields walking through the network can come across other users browsing the same JAN. This Vijjana helps the users to interact among themselves through the chat board discuss and might even rate a JAN collaboratively.

System 3 is the interface between the discovery agent and the web through which the Jans are added. The discovery agent is responsible for adding the Jans to the database. System 4 is the interface between the database and the knowledge network. This database defines the actual taxonomies for the network and the relationships between the URL’s. It is a relational database built in MySQL.
Figure 2.5 illustrates the overall architecture of the Vijjana client program. The client or the user also contribute to the growing knowledge network by adding some Jans that they feel might be worth sharing with others. As seen from the fig.2.5 the Jan can be added by a client through a mail or through the markup. The parser parses the Jan to identify to which class of Jans it belongs to and passes on the information agent to the organizing agent that does the job. Future implementations can be tried to expand the present interface to include RSS feeds.
An RSS document referred to as a “feed”, “web feed” or a “channel” describes the summary of a web site. The XML file format information providing an outline of the content in the website automatically updates the subscribers to the feeds.

Figure 6: Present view of Vijiana [17]
Figure 7: Proposed web interface view of Vijjana [1]
3. Vijjana Database & XML

3.1 Introduction

Vijjana knowledge network provides a search that is conveyed by the meanings of the objects. The current search engines restrict the search to a certain extent. Vijjana has evolved to redo the things and provide a much better and effortless search. The basic element behind Vijjana is doing a search that concentrates on the meaning or the interpretation depending on where they are used rather than just by a bunch of key

The foundation for this knowledge network would be a database. A database is a structured collection of data that meets the needs of the users. The popular database model used is the relational model. The other models that are in use are the hierarchical and network models. Vijjana aims at a professionally organized database to quench the user’s thirst for knowledge.

The Vijjana database consists of three parts, basic data, user data (personal space), and the data on the website. ’ODP’, Open Directory project is the main source for the basic data. It is also known as the ‘dmoz’ is a large directory of the links all over the world owned by Netscape. The ODP uses a hierarchical ontology schema for the topic listings. All the related links of a particular topic are related into one category.

3.2 Pictorial Representation of Vijjana database

The Vijjana database structure is depicted in the Fig 3.1. It also shows the relationships between different elements of the structure.
Figure 8: Overall representation of the database
3.3 Description of the Structure

Different tables in the database are designed for different targets.

- **General, Identifier** and **Classification** are related to the Open Directory Project.
- The tables **Educational, Technical, Rights** describe the properties of the Jans.
- **Life Cycle** gives the status of the Jan.
- User information is given in the **Membership** table.
- Following tables relate the user and the JAN.
  - Comment
  - Comment Rating
  - Jan Rating
  - Keyword

3.3.1 Properties

3.3.1.1 Educational

This table gives the educational information to the users. This data is useful to all of those who are involved in teaching, learning writing etc. As we have seen earlier in the document the two professors can easily look through the database to find a information relevant to their teaching.

3.3.1.2 Rights

The table describes the rights that are assigned and allow users to use the information. It gives the terms and conditions under which the information can be used. The copyright information is provided in this table.
3.3.1.3 Technical

The technical data is all about the technical requirements of the learning object. It cites the software required to access that learning object and other hardware details.

3.3.2 Status

3.3.2.1 Life Cycle

This table describes the completion status or condition of this learning object.

3.3.2 User

3.3.2.1 Membership

The user information is made available in this table. A user can be registered into Vijjana. He needs a user and a password to access his personal space and to venture inside Vijjana. The registrations details, user info is all stored in this table.

3.3.4 User and the Jan

Certain activities describe the relation between the user and Jan.

3.3.4.1 Comment

Users can add Jans to the database. They can go through a Jan and also can comment them. These comments are displayed with the Jan. The user, the comment and the Jan are all identified by their own identification ID’s. It gives the update as to when the comment is added and updated.
3.3.4.2 Comment Rating

The Vijjana characterizes a useful feature of rating the comment on a particular Jan. A user can comment a Jan and another user can rate that comment. The comment is associated with an ID used to relate it to the Jan and display with it.

3.3.4.3 Jan Rating

This gives the rating of a particular Jan. Any user can rate any Jan depending on the degree of its usefulness as to their opinion. A Jan might be rated highest by one user while another user might judge and rate it lower.

3.4 XML Technology

XML is a meta-language for describing markup languages. XML defines tags and structural relationships between them. So the XML defines the tags itself. XML is a markup language for documents with structured information. It identifies the structures in the document. It also defines a standardized way to add markup to the documents.

XML stands for Extensible Markup Language. XML is a simple, self-describing way of representing both the text and the data with ease. The data represented by XML can be easily understood and processed without much human or machine intelligence. XML data can be exchanged between platforms, languages, and applications. It was designed to transport data, store data focusing on what the data is. In a way it is different from
the HTML. HTML deals on how to display the data, while XML cares about transporting it. Extensible Stylesheet Language Transformations XSLT is the stylesheet language of XML.

### 3.4.1 Sample structure of XML

A simple structure of an XML can be the following

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="simple.xsl"?>
<breakfast_menu>
  <food>
    <name>Dosa</name>
    <price>$5.95</price>
    <description>A thin pancake made out of rice flour.</description>
  </food>
  <food>
    <name>vada</name>
    <price>$7.95</price>
    <description>A snack that is spicy, shaped like a doughnut and made from dal.</description>
  </food>
</breakfast_menu>
```

A result of the above XML can be the following.
Figure 9: View of a sample XML

The xml can be defined to get different views such as the Tree view or the Radial view. In the above xml, each food item is added as a child to the main node breakfast menu. A description is again a child element to the node each food item. As such the result expands as new food items are added to the breakfast menu.
4. Visualization

A picture is worth ten thousand words. A complicated picture might depict more information, especially in the fields where the operations are done in logical level rather than handling physical objects. Entity relationship diagrams, use case diagrams, database diagrams, organization hierarchies, and even a family tree are candidates for a visualization method that takes care of laying out the relationships so that they are easily distinguishable.

The visualization of the database is the critical part in the Vijjana knowledge network through which the user interacts with the knowledge network. Appealing and user friendly interaction visualization is implemented by Vijjana. For the visualization purpose, the Prefuse toolkit has been used.

4.1 Prefuse Toolkit

Prefuse is a set of software tools for creating rich interactive data visualizations. It supports features for modeling data, visualization, and interaction of data. The prefuse tool provides different visualizations such as the Tree view, Radial graph view and Graph view. Using different java based classes in prefuse, the visualization of the database can be designed.

Following are some of the features prefuse includes.

- Different visualizations that are user friendly.
- Techniques for layout, color, size, orientation.
- Search engines to offer a quick search.
- Support for SQL queries and to provide a visualization using these results.
API’s are easy to use and implement processing and interaction.

4.2. Different visualizations

4.2.1 Tree View

Tree view is the new web control tool with a parent child relation depiction. This representation has branches and nodes. With a new child added to the parent, the node, a new branch evolves to it. The tree can be build in two ways, either from left to right or from right to left. As each node is clicked its children are shown. If any of the child branches has any further child it shows up when it is clicked. When the earlier nodes are clicked the tree shrinks.

4.2.2 Radial Graph view

Focus + context technique is used for visualizing and manipulating large hierarchies to give a radial graph view. The focused node will be displayed at the center and all other nodes are linked around the focused node. This will help us in displaying very large hierarchies.

4.2.3 Graph View

The graph view is visualized as a woven net. But it does not have a common node. One link is connected to couple of another related.
4.3 Prefuse Structure and its implementation in Vijjana

Figure 11: Diagram depicting the relation of different prefuse packages and classes to the infovis reference model. [2]

Figure 10: Sample Graph view
The prefuse.data package provides various structures like the Table, Graph and Tree data structures for visualizing the data. The Node and Edge classes are used as the members for the Graph and the Tree structures.

The prefuse.io provides classes for reading and writing the graph structures via GraphML and TreeML readers and writers. These classes are used to define the main graph required for viewing.

The prefuse has classes that support reading and writing the table, graph, and tree data from files. The data is derived from the database.

The visualization can be extracted by adding the data set to the ‘Visualization’ class. For the Node or Edge added a related VisualItem instance is created.

The Action modules provide the visual mappings. They provide different effects such as the setting item visibility, calculating layouts, defining color values etc.

Renderers are responsible for drawing the items and determining the item bounds. These renderers are used for drawing shapes, labels, and images.

The interactive views are embedded in the display component, acting on the visualized items. The items can be panned, zoomed and rotated as desired using these components.

The prefuse.controls package provides controls for focusing, dragging, panning etc of the view.

Interaction can also occur through the queries in the prefuse.data.query package.

4.4 Programming Technique

A brief overview of how the visualization is implemented.

The ‘vA’ fetches the data from the database agent.
The Node Table and the Edge Table created by the database agent are provided to the Prefuse program.

The data is retrieved using SQL Queries and the data is stored in an XML format.

The XML data is visualized using the Visual Instance classes in prefuse.

An applet is created with these views.

The views are embedded into the web page.

4.4.1 Initialization phase

The username and password are defined. The URL for connecting the database is also defined. The graph, the node table and the edge table are defined.

4.4.2 Connection phase

The database is connected. The parameters required for the connection are initialised. The username, password, URL, database are to be identified for this purpose. This returns a Databasedatasource. The Databasedatasource is then initialized to null. It is used to send queries to a relational database, processes the results and stores the results in prefuse Table instances. In order to use the ‘Class DatabaseDataSource’, the ConnectionFactory has to be initialized.

4.4.3 Query Phase

The Graphchildnode table is retrieved here. The SQL query is issued to retrieve the id, name and title for the node table.
The GraphEdgeTable is retrieved in this part. Similar to the above case queries are run to get the id, sid(source id), tid(target id) columns for the edge table. A new table t2 is created with these columns.

### Node Table

<table>
<thead>
<tr>
<th>Node ID</th>
<th>General ID</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>Computer Science</td>
</tr>
<tr>
<td>2</td>
<td>17</td>
<td>Hardware</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>Software</td>
</tr>
</tbody>
</table>

### Edge Table

<table>
<thead>
<tr>
<th>Source ID</th>
<th>Target ID</th>
<th>Edge ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

The above two tables represent the Node table and Edge table. So the tree would be built with the information in the above two tables.
4.4.4 Graph view Phase

An XML file ‘Graph’ is created results from the previous phases. The GraphWriter instance writes a tree file using the TreeML file format. The parameters for the writeGraph method in the TreeML Writer are the graph, and the os.

Sample XML for the previously mentioned tree format is shown below.

```xml
<GraphXML xmlns="http://graphml.graphdrawing.org/xmlns">
  <graph edgedefault="undirected">
    <key id="name" for="node" attr.name="name" attr.type="string"/>
    <node id="1">
      <data key="name">Computer_Science</data>
    </node>
    <node id="2">
      <data key="name">Software</data>
    </node>
  </graph>
</GraphXML>
```
4.4.5 Visualization Phase

The graph is extracted and visualized using the visual instance of the prefuse package. The prefuse tool provides an applet that is implemented and used to get the visualization in a separate pane.

4.5 Embedding into the web page

- Create a class file of the views.
- Integrate the class file and prefuse.jar file in the website code.
- An html file is required to display the applet. The html file is the page on which you run the applet.

All the files are stored in a common directory from where in the code when executed in the browser accesses the files and generated the required views.
5. Implementation of Visualization

Any model is appealing when it is attractive. This model of Vijjana can be brought out near to users through visualization that provides them with an idea of the knowledge network niched. All the tools used in Vijjana are open source and some programming languages.

5.1 Prefuse Toolkit

Prefuse Beta released in 2007.1.21 has been used for implementing the visualization of Vijjana.

5.2 MySQL Database

The version of the MySQL 5.0.22 version was used as the web server. The database was imported to the local server (due to the restricted ports in the university) and connected locally. Either of the Netbeans IDE or the Eclipse environments could be used to implement the programs.

5.3 NetBeans IDE

The 6.1 version used is an open source Integrated Development Environment. Supports Java language (mainly used in programming), C/C++ etc.

5.4 Eclipse

Eclipse Version: 3.4.0 has been used in Vijjana for the visualization. The java programs developed have been worked out in this environment.

5.5 Platform

Windows Vista has been used as a platform for all the tools and the softwares.
6. RESULTS

The test results when executed in Eclipse are the following.

Following is the tree view of the small part extracted from the database.

6.1 Prefuse Tree view

![Prefuse Tree View](image)

*Figure 13: Prefuse Tree View*
6.2 Prefuse Radial Graph view

The radial graph view is the following.

![Prefuse Radial Graph View](image)

*Figure 14: Prefuse Radial Graph View*
6.4 Web Embedded views
These views give a basic idea of how the Jans are structured in Vijjana. These views are embedded in the web interface of Vijjana. They are hidden in the “Visualization” on the left hand side of the main home page. When clicked on the button, two visualizations are shown as below.

Figure 15: Embedding the views in two buttons
6.4.1 Tree

The tree view when embedded in the web page is

![Web Embedded Tree View](image)

*Figure 16: Web Embedded Tree View*
6.4.2 Radial Graph View

The radial graph view would be

![Radial Graph View Image]

Figure 17: Web Embedded Radial Graph View
7. Future Work

The basic foundation for Vijjana has been laid. A full structured database has been developed. But still it needs to be more developed and more new impressing features can be integrated. Because of the large amount of data, the visualization has been restricted to a small part of it. There can be a solution designed for it in the future. The earlier visualization was for a small database, now extended to the larger one.

The present work involves an XML created for the entire domain. But in the near future, Vijjana expects the generation of XML with respect to the user choice. The static visualization created now should be made dynamic.

Integration of the Visualization with the database is a major work where in when a new Jan is added to the knowledge network not only the database but also the visualization is updated. Designing different visualizations can also be a good work. Clicking on a node giving the URL’s is another important and very useful work that Vijjana needs to get more attractive. Highlighting different nodes such as recently added, recently updated, isolated and the dead. Grouping the same categorized links into a net color can also be an expected future work.
8. CONCLUSION

Once, computers worked only on numbers. They are being improved from listening to understanding like humans. This brought into the term ‘Semanticweb’ technology. It is still in building the relationships between the information. Once, the matured semweb comes into light, as it has promised, makes life easier. Checking bank accounts on the calendar dates, scheduling appointments on calendar etc., are expected.

Making Vijjana implementable slowly would one day bring it into light solving earlier non-semantic issues and making a semantic search possible. If Vijjana would be a huge success, it makes its own mark in the web world with its features and finds a place among the other search engines.
References

3. ‘Visualization module of Vijjana, a Pragmatic Model for Collaborative, Self-organizing,  
   Domain Centric Knowledge Networks’, a problem report submitted to CEMR by Sasanka Babu Gottipati
4. Jeffrey Heer, Large-Scale Online Social Network Visualization [Online]  
   http://www.cs.berkeley.edu/~jheer/socialnet/
6. Twine: A social network built on the semantic web dls interview, by Brad Linder,  
   http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58
11. August 2009: How Google beat Amazon and Ebay to the Semantic Web by Paul Ford,  
12. Munzner, Tamara. Interactive Visualization of Large Graphs and Networks. Stanford  
13. “Database module of Vijjana, a Pragmatic Model for Collaborative, Self-organizing,  
    Domain Centric Knowledge Networks”, a problem report submitted by, Amara Satish Kumar
15. Database Agent and Visualization of Vijjana, a Pragmatic Model for Collaborative, Self-organizing, Domain Centric Knowledge Networks, problem report submitted by, Durga Vardhineedi
17. Organization module of Vijjana, a Pragmatic Model for Collaborative, Self-organizing, Domain Centric Knowledge Networks, problem report submitted by Gaurav Sudhakar Narkhede