ONTOLOGY VISUALIZATION PROTÉGÉ TOOLS – A REVIEW

R. Sivakumar[1] and P.V. Arivoli[2]

[1] Associate Professor, Department of Computer Science,
A.V.V.M. Sri Pushpam College, Bharathidasan University, Trichirappalli, India
rskumar.avvmspc@gmail.com

[2]Project Fellow, Department of Computer Science, A.V.V.M. Sri Pushpam College,
Bharathidasan University, Trichirappalli, India
pva.tvr@gmail.com

ABSTRACT

Protégé is one of the most popular tools of the ontology visualization. The "Protégé" tools are being applied for further development in various disciplines for better understanding of knowledge. These tools commonly use four methods of ontology visualization, namely, indented list, node-link and tree, zoomable, and focus+context. The purpose of this work is to present a study on application of these four methods in the development of different kinds of protégé visualization tools and categorize their characteristics and features so that it assists in method selection and promotes further future research in the area of ontology visualization.

KEYWORDS

Ontology visualization techniques, Protégé tools, Knowledge bases, Knowledge models.

1. INTRODUCTION

An ontology is a conceptualization of a domain into machine readable format [7]. Ontologies are becoming increasingly popular modelling schemas for knowledge management services and applications. Focus on developing tools to graphically visualise ontologies is rising to aid their assessment and analysis. Graph visualisation helps to browse and comprehend the structure of ontologies. Protégé [21] is one of the most widely used ontology development tools that were developed at Stanford University. Protégé provides an intuitive editor for ontologies and has extensions for ontology visualization, project management, software engineering and other modelling tasks. An ontology, according to the definition in [20] is a formal explicit description of a domain, consisting of classes, which are the concepts found in the domain. Classes are organized in a specialization/generalization hierarchy through is-a (or inheritance) links, where each class is allowed to have zero, one or multiple parent classes. Each class has properties (or slots) describing various features of the modelled class. Slots are typed, and allowed types are either simple types (strings, numbers, booleans or enumerations) or instances of other classes (references); restriction on the value ranges of slots (e.g. integers from 1 to 10) may also be defined. Finally, instantiation may be applied to classes to produce items corresponding to individual objects in the domain of discourse (instances). Each instance has a concrete value for each property of the class it belongs to. Classes, together with instances are said to constitute the knowledge base. From the definition above, it is evident that the task of visualizing the full set of ontology features is not an easy one. The properties of ontology are summarized as follows:

DOI : 10.5121/ijait.2011.1401

- ***Hierarchy.*** A type of organization that, like a tree, branches into more specific units, each of which is "owned" by the higher-level unit immediately above.
- ***Properties representation.*** More than a hierarchy, as it concepts are described by using restrictions on properties.
- ***Level of detail.*** Possibility to choose till which level an ontology to be provided.
- ***History.*** The concepts that were chosen in the previous steps.
- ***Filtering.*** Ontologies could contain hundreds of properties. The user can be interested in only the subset of the ontology, based on the central concept and the properties of the user's choice.
- ***Multiple geometrical views.*** The representation of the graph in different geometrical models to better understand the structure of ontology.
- ***Zoom semantic/geometric.*** To see more or less details during ontology exploration. With the geometric zoom the visualized object is scaled when the user zooms in/out. The semantic zoom provides the possibility to see more/less details of the object by zooming in/out.

A number of ontology visualizations exist that have been embedded in ontology management tools (e.g. [23] and [11]) and are used as information retrieval aids in applications that use ontologies [13]. Evaluations of ontology visualization effectiveness, however, are up to this point scarce: [14] presents some user experiments focused on tree visualization systems, whereas [12] reports on preliminary results from a user study involving four visualization methods. They are *indented list, node-link and tree, zoomable*, and *focus+context*. A number of visualisation techniques have been described over the years, such as spanning tree layouts, tree-maps [10], fisheye views [4], hyperbolic [15] and 3D hyperbolic layouts [17], aiming to help comprehend and analyse complex information structures. Preference of visualisation models vary according to users needs and query context [5]. It is also dependant on the type and extent of the visualised network. Using a combination of integrated visualisations of various types has shown to be sometimes beneficial [19][24]. Complex networks of multi dimensional hierarchies and arbitrary relations are becoming common characteristics of current ontologies. Tools that discriminate against some of these features, for example by supporting spanning trees or hierarchical relations only, might not be appropriate for comprehensive ontology visualisation. Ontologies, together with their Knowledge Bases (KBs), could grow into very large information networks, especially if aimed at providing scalable services for the Semantic Web [1]. Visualising large networks has always been challenging [26]. [8] surveyed a wide range of visualisation techniques and concluded that all existing algorithms have a size limit afterwhich they cannot cope. [18] and [8] stressed the importance of reducing the visualised graphs into smaller sized sub graphs that users can browse to visualise other parts of the network. Ontologies are semantically rich by definition. Ontology visualisers should therefore turn some of these semanticsmore explicit [26]. Spring-layout algorithms [3] are example techniques that display semantically similar nodes closer to each other. Such layouts could help users to quickly recognise dense areas and interrelated objects in their ontologies and KBs. In this paper we conduct a survey on existing ontology visualization techniques & protégé ontology tools and present a summary of various features and capabilities of the existing techniques & tools respectively. The purpose of this work is to assist the researchers of ontologies to understand more about this domain and to extend their research activities in new directions.

This paper is structured as follows: Section 2 gives an overview of protégé and its services. Next, Section 3 presents the complete survey of different ontology visualization methods and their usage in the design of protégé tools. A discussion is also presented in Section 4 on comparison of various features and capabilities of different existing protégé tools. Finally, Section 5 concludes this paper.

2. PROTÉGÉ

Protégé-2000 [16] is a very popular knowledge-modelling tool developed at Stanford University. Ontologies and knowledge-bases can be edited interactively within Protégé and accessed with a graphical user interface and Java API. Protégé can be extended with pluggable components to add new functionalities and services. There exists an increasing number of plug-ins offering a variety of additional features, such as extra ontology management tools, multimedia support, querying and reasoning engines, problem solving methods, etc. Protégé implements a rich set of knowledge-modelling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé gives support for building the ontologies that are frame-based, in accordance with the Open Knowledge Base Connectivity protocol (OKBC). The extended version of frame based system was introduced in 2003 to support OWL with an advantage of semantic web version. There are various forms such as RDF(s), OWL and XML Schema in which protégé ontology can be exported. The following are various plug-ins available in protégé.

- ***JSave*** – To generate java class defintion stubs for protégé classes (http://protege.stanford.edu/plugins/ jsave/).
- ***Protégé Web Browser*** – Allows users to share protégé ontologies over the intenet .
- ***WordNet plug-in*** – An interface to WordNet knowledge base (http://protege.stanford.edu/plug-ins/wordnettab/wordnet_tab.html).
- ***XML Schema*** – Transforms a protégé knowledge into XML (http://faculty.washington.edu/gennari/Protege-plugins/ XMLBackend / XMLBackend.html).
- ***UML Plug-in*** – Generates an XML schema file describing protégé knowledge model and an XML file (http://protege.stanford.edu/plug-ins/uml/).
- ***DataGenie*** – Allows reading and creating knowledge model from RDBMS using JDBC (http://faculty.washington.edu/gennari/Protege-plugins/ DataGenie/index.html).
- ***Docgen*** – To create reports describing ontologies (http://protege-docgen.sourceforge.net/).
- ***JessTab*** – Provides a Jess console window to interact with Jess while running protégé (http://www.ida.liu.se/~her/JessTab/).
- ***Algernon*** – Performs forward and backward inference of frame based knowledge bases (http:/algernonj.sourceforge.net/doc/algernon-protege.html).
- ***PROMT*** – Allows to manage multiple ontologies with in protégé.
- ***OWL-S Editor*** – Allows loading, creating, managing and visualizing OWL-S services(http://owlseditor.semwebcentral.org/).

3. ONTOLOGY VISUALIZATION METHODS IN PROTÉGÉ

3.1. The Indented list method

The *indented list* methods represent the taxonomy of the ontology following the file system explorer-tree view. These methods are intuitive and simple to implement, representing *is-a* inheritance relationships through the *indented list* paradigm, with subclasses appearing below their super classes and indented to the right. Users may navigate within the class hierarchy and expand or retract branches; when a class (or multiple classes) is (are) selected in the hierarchy pane as also confirmed by the results of a user evaluation [12]. The following figure 1 is example of the *indented list method.*

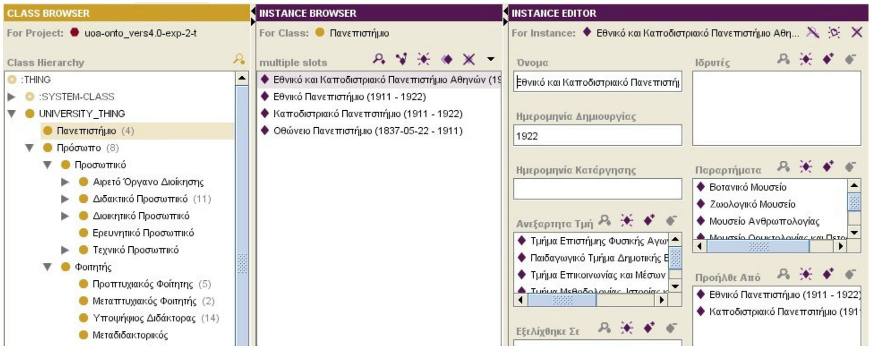
CLASS BROWSER
For Project: uoa-onto_vers4.0-exp-2-t
Class Hierarchy
:THING
:SYSTEM-CLASS
UNIVERSITY_THING
Πανεπιστήμιο (4)
Πρόσωπο (8)
Προσωπικό
Αιρετό Όργανο Διοίκησης
Διδακτικό Προσωπικό (11)
Διοικητικό Προσωπικό
Ερευνητικό Προσωπικό
Τεχνικό Προσωπικό
Φοιτητής
Προπτυχιακός Φοιτητής (5)
Μεταπτυχιακός Φοιτητής (2)
Υποψήφιος Διδάκτορας (14)
Μεταδιδακτορικός
INSTANCE BROWSER
For Class: Πανεπιστήμιο
multiple slots
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών (19
Εθνικό Πανεπιστήμιο (1911 - 1922)
Καποδιστριακό Πανεπιστήμιο (1911 - 1922)
Οθώνειο Πανεπιστήμιο (1837-05-22 - 1911)
INSTANCE EDITOR
For Instance: Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθη...
Όνομα
Εθνικό και Καποδιστριακό Πανεπιστή
Ιδρυτές
Ημερομηνία Δημιουργίας
1922
Ημερομηνία Κατάργησης
Παραρτήματα
Βοτανικό Μουσείο
Ζωολογικό Μουσείο
Μουσείο Ανθρωπολογίας
Ανεξάρτητα Τμή
Τμήμα Επιστήμης Φυσικής Αγω
Παιδαγωγικό Τμήμα Δημοτικής Ε
Τμήμα Επικοινωνίας και Μέσων
Προήλθε Από
Εθνικό Πανεπιστήμιο (1911 - 1922)
Καποδιστριακό Πανεπιστήμιο (191
Εξελίχθηκε Σε

Figure 1: Protégé Class browser

3.1.1. Protégé Class browser

It is a tool developed by protégé using *indented list* method. The protégé class browser consists in its simplicity of representation, and also familiarity to the user. Secondly it offers a clear view of all the class names and their hierarchy. Thirdly, its retraction and expand of nodes in useful features specific for focusing on specific parts of the hierarchy, especially for large hierarchies. Also the open source software is readily available of this. Figure 1 sums up its main characteristics. However, the protégé class browser has certain limitations. Its technique is that it can represent a tree and not a graph. Consequently it can display inheritance (*is-a*) relations, and not role relations. This does not support visual representation of the role relations. It cannot expand all and, not retract all buttons in protégé class browser. Above all, there will be zoomable view and no overview window display.

3.2. Node link and Tree

The *node-link and tree* methods represent another approach frequently used for ontology visualization. The ontology is represented as a set of interconnected nodes. They offer a good overview of the hierarchy and connections, but may produce cluttered displays when used to visualize more than hundred of nodes. The figure 2 and figure 3 are examples of *node-link and tree method.*

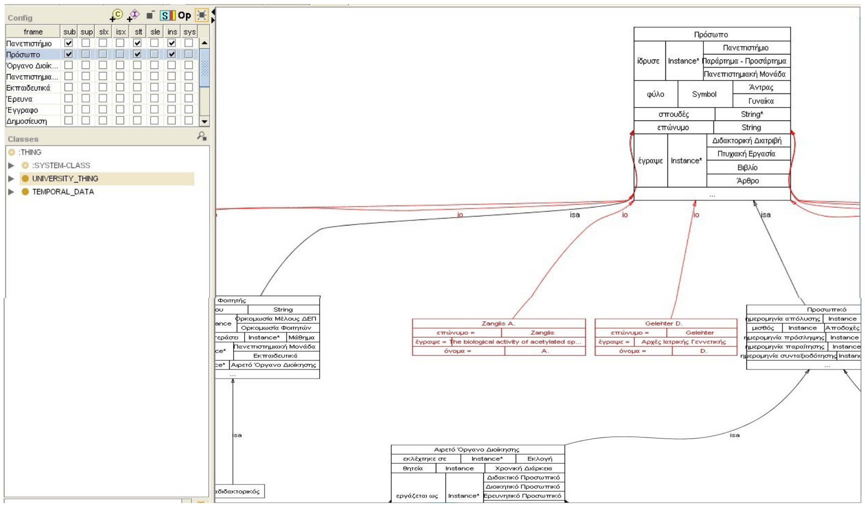

Figure 2: Protégé OntoViz visualization.

3.2.1. Protégé OntoViz visualization

The Protégé OntoViz is a tool based on the *node-link and tree* method with 2D view. The Protégé OntoViz has many merits. It provides an intuitive way for hierarchy representation. The ontology is presented as a 2D graph. It has the capability of each class to present, apart from the name, its properties and inheritance and also relations. Its instances are displayed in different colors. A right clicking on the graph allows the user to *zoom-in* as well as *zoom-out*. Another merit is that it has an overview window display. The above figure 2 represents the Protégé OntoViz tab visualization. However, the Protégé OntoViz has two serious limitations. It has no keyword search availability and relation link. Also it has no retraction and expansion of nodes.

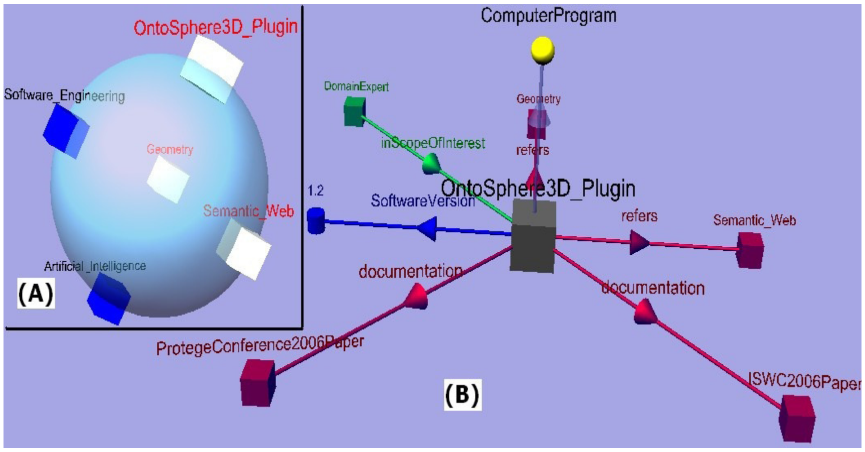

Figure 3: OntoSphere visualization (a) Root Focus view (b) Tree Focus view.

3.2.1. OntoSphere

The Protégé OntoSphere is another tool using the *node-link and tree* method with 3D view. The Protégé OntoSphere has one important merit. It makes three different views available for the user viz. Root focus scene, tree focus scene and concept focus scene. However there are three limitations. Overview taxonomy structure cannot be viewed. Properties cannot be visualized and also there is no overview window display. The figure 3 represents the OntoSphere visualization tab. The figure 3(a) represents Root focus view and figure 3(b) represents Tree focus view.

3.3. Zoomable method

The *zoomable* methods present the nodes in the lower levels of the hierarchy nested inside their parents and with smaller size. The user may *zoom-in* to the child nodes to enlarge them. Node nesting is also used for *instance-of relationships*, thus a class node contains both its subclasses and its instances; the user may distinguish between the *class-type* and *instance-type* nodes through their color. These methods seem to be successful for browsing to locate specific nodes. The figure 4 is example of *zoomable method.*

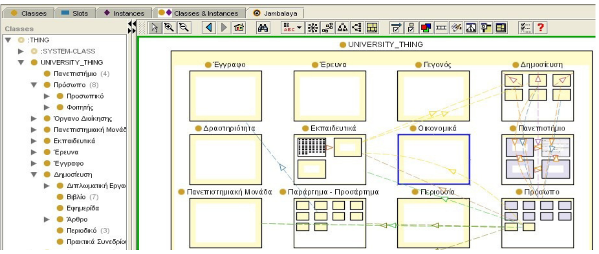

Figure 4:. The Jambalaya tab in Protégé with Class Browser on the left.

3.3.1. Jambalaya

The Jambalaya protégé tool uses the visualization method of *zoomable with 2D* view. It has the following merits: it makes all animated transitions available; it uses the ShriMP [Simple Hierarchical Multi-Perspective] 3D visualization technique; it can display overview window display; it helps to view the history/ back and forward. Also it has certain limitations. Properties can be displayed in embedded form only when the selected node is zoomed-in. There will be no retraction and expansion of nodes. Further movement and / or rotation of graphs are not available. The above figure 4 shows the Jambalaya protégé tool.

3.4. Focus+Context method

The *focus + context* methods present the node on the focus in the centre and the connected nodes around it, usually reduced in size. According to this technique nodes (classes) repel one another, whereas the edges (links) attract them, thus nodes that are semantically similar are placed closed to one another. This technique is effective to provide global overviews, to focus on specific nodes, and for quick browsing.

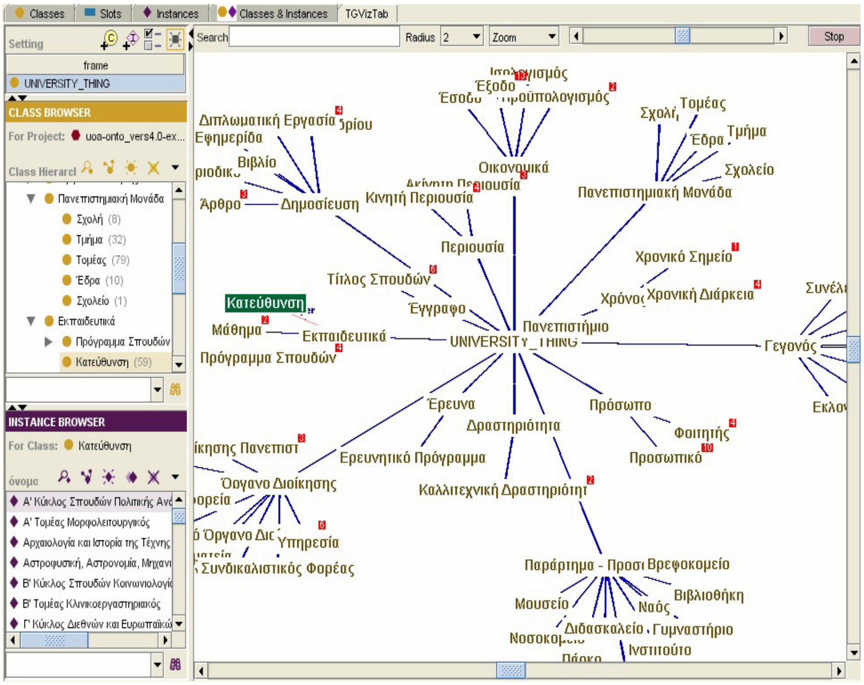

Figure 5: Protégé TGVizTab

3.4.1. Protégé TGVizTab

This tool is designed based on *focus+Context* method. One of its special characteristics includes interactive and adjust to the user commands as nodes move. It makes animated transitions available. The figure 5 represents the protégé TGVizTab. It has also some limitations. First there is no consensus regarding its visualization. Second, it makes a chaotic view of hierarchy. Next, there is no extraction and retraction view and movement and /or rotation of the graph. Finally, there is no relational link representation.

4. DISCUSSION

Tables 1(a) and 1(b) summarize ontology visualization characteristics in protégé tools. In Protégé class browser, classes are presented as nodes in an indented, expandable and rectangle tree. Instances are displayed in a separate window. In Protégé Ontoviz, classes are represented in rectangle nodes with different colors. In OntoSphere, they are represented as spheres. When it comes to Jambalaya, they are represented as rectangles inside their parent nodes. In Protégé TGVizTab, classes and instances are represented as labels of different colors. In Protégé class browser, hierarchy can not be displayed. In Protégé Ontoviz visualization, the child nodes are placed under the parent ones and linked with an "*is-a*" link. In OntoSphere, there are two views. In tree focus view, child nodes are under their parent. In root focus view, the roots can be shown. In Jambalaya, lower levels are represented with smaller size rectangles and placed inside their parent nodes. Finally TGVizTab, lower level nodes are displayed around their parents and connected with them, with a "*sub*" link.

Table1(a): Summary of the ontology visualization characteristics in protégé tools.

Protégé Tool name	*Protégé Class browser*	*Protégé OntoViz visualization*	*OntoSphere*	*Jambalaya*	*Protégé TGVizTab*
Ontology technique used	indented list	node-link and tree	node-link and tree	zoomable	focus+context
Dimension	2D	2D	3D	2D	2D
Classes & Instances	Classes - nodes of indented, expandable and retractable tree. Instances - in a separate window.	Rectangle nodes with different color for classes and instances.	Classes and instances are represented as spheres.	Represented as rectangles inside their Parent node.	Classes and instances are represented as labels of different colors.
Class hierarchy	Not available.	The child nodes are placed under the parent ones and linked with an "*isa*" link.	In the Tree Focus View child nodes are placed under their parent.	Lower levels are represented with smaller size rectangles and placed inside their parent nodes.	Lower level nodes are displayed around their parent and connected with it, with a "*sub*" link.
Multiple inheritance	Child nodes are placed under both parents.	The child node is Linked with all the parents.	The child node is connected to both its parents in TreeFocus View.	Child nodes are placed under both parents.	There is a link from the node to both its parents.

With the case of multiple inheritance, in protégé class browser, child nodes can be placed under both parents. When it comes to protégé Ontoviz visualization, the child node can be linked with all the parents. In OntoSphere, the child node can be connected to both its parents in tree focus view. In Jambalaya, child nodes can be placed under both parents. In TGVizTab, there will be a link from the node to both its parents.

Table1(b): Summary of the ontology visualization characteristics in protégé tools.

Protégé Tool name	*Protégé Class browser*	*Protégé OntoViz visualization*	*OntoSphere*	*Jambalaya*	*Protégé TGVizTab*
Role relations	Supported through the properties window only.	They are represented with labelled links.	In Concept Focus View links are used to denote role relations.	Supported through the propertied and as directed links with their label visible as a tool tip.	Links with labels on mouse over are used.
Properties	Displayed in a separate window.	Displayed on the node	Not available.	Displayed as an embedded form if the selected node is *zoomed-in*.	Displayed in a separate window.
Keyword Search and Filtering	Available only for the already visible nodes in the class and instance windows.	Not available.	Not available .	Possibility to select the type of the searched item and search between the results.	Only works on the part of the ontology that is already visible.
No. of nodes supported	1000 to 10000 nodes	300 nodes	1000 to 10000 nodes	Up to 1000	1000 to 10000 nodes
Software availability	Open Source, available at [Protégé].	Available as a Protégé [Protégé Project] plug-in.	Available as a Protégé plug-in in [OntoSphere].	Open source, available as a Protégé [Protégé] plug-in.	Open Source, available as a Protégé [Protégé] plug-in.

In Protégé class browser, role relations cannot be viewed. But it is supported through the property window. In protégé Ontoviz visualization, it can be represented with labelled links. In OntoSphere, concept focus views are used to denote role relations. In Jambalaya, role relations are supported through the propertied and as directed links with their label visible as a tool tip. In protégé TGVizTab, role relations will be displayed when the mouse point is moved over the node. In protégé class browser, properties are displayed in a separate window. In protégé Ontoviz visualization, properties are displayed on the node. In Jambalaya, properties are displayed as an embedded form if the selected node is zoomed-in. In protégé TGVizTab, properties are displayed in a separate window. But in OntoSphere, the properties cannot be viewed. In protégé class browser, key word search and filtering option are available only for the visible nodes. In Jambalaya, key word search and filtering options are supported with the possibility to select the type of the searched item and search between the results. In protégé TGVizTab option works only on the part of the ontology that is already visible. In Ontoviz and OntoSphere key word search and filtering options are not provided. Table 2 gives a summary of the interaction and navigation support of protégé tools. The protégé tools are available in two different main versions such as protégé_3.4.4 and protégé_4.1_beta. The main difference is that the later version supports *ontograph* feature.

Table2: Summary of the interaction and navigation support of protégé tools.

Protégé Tool Name	*Protégé Class browser*	*Protégé OntoViz visualization*	*OntoSphere*	*Jambalaya*	*Protégé TGVizTab*
Retraction &Expansion of nodes / pruning	Yes	Yes	No	No	Yes
Movement and / or rotation of the graph	No	No	Yes	No	Yes
Movement and / or rotation of the view point	Yes	Yes	Yes	Yes	Yes
Zooming	No	Yes	No	Yes	Yes
Overview window	No	Yes	No	No	Yes
History back and forward	No	No	No	Yes	No
Animated transactions	No	No	No	Yes	Yes

5. CONCLUSION

Much work has been done in the field of graph and hierarchy visualization both in 2D and 3D. The visualization of ontologies using protégé tool is a particular sub problem of this area with many implications due to the various features that an ontology visualization should present. The current work is an attempt to summarize the research that has been done so far in this area, providing an overview of the protégé tools and their main advantages and limitations. As seen from the survey and the information provided in the tables 1 and 2, it can be concluded, there is no one specific method that seems to be the most appropriate for all applications and, consequently, a viable solution is providing the user with several visualizations, so as to be able to choose the one that is the most appropriate for one's current needs.

ACKNOWLEDGEMENTS

This work has been financed by University Grants Commission Major Research Project – Ref. No. 38-4/2009(SR) (India).

REFERENCES

[1] Berners-Lee, T., Handler, J., Lassila, O, (2001), The Semantic Web. *Scientific American.*

[2] Bosca, A., bomino, D., and Pellegrino, P. (2005). OntoSphere: more than a 3D ontology visualization tool. *In Proceedings of SWAP, the 2nd Italian Semantic Web Workshop*, Trento, Italy, December 14-16, CEUR, Workshop Proceedings, ISSN 1613-0073, online http://ceur-ws.org/Vol-166/70.pdf.

[3] Eades, P., (1984), A Heuristic for Graph Drawing. *Congressus Numerantium*,42:149-160.

[4] Furnas, G. W., (1986). The FISHEYE view: A new look at structured files. *Proc. of the Conf. on Human Factors in Computing Systems ACM*. pp 16-23.

[5] Graham, M., Kennedy, J., Benyon, D., 2000. Towards a Methodology for Developing Visualizations. Int. J. of Human-Computer Studies 53(5): 789-807.

[6] Gruber T R, (1995). Towards principles for the design of ontology used for knowledge sharing [J]. *International Journal of Human-Computer Studies.*1995 (43): 907-928.

[7] Guarino, N., Giaretta, P., (1995). Ontologies and Knowledge bases: towards a terminological clarification. *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, IOS, 25-32.

[8] Herman, I., Melancon, G., Marshall, M.S., (2000). Graph Visualization and Navigation in Information Visualisa-tion: a Survey. *IEEE Transactions on Visualisation and Computer Graphics* 6(1): 24-43.

[9] Jambalaya, URL: http://www.thechiselgroup.org/jambalaya.

[10] Johnson, B., Shneiderman, B.: Treemaps, (1991). A space-filling approach to the visualisation of hierarchical in-formation structures. *Proc. 2nd Int. IEEE Visualization Conference*, San Diego.

[11] KAON, http://kaon.semanticweb.org/

[12] A. Katifori, E. Torou, C. Halatsis, G. Lepouras, C. Vassilakis, (2006). A Comparative Study of Four Ontology Visualization Techniques in Protégé: Experiment Setup and Preliminary Results, URL:http://ieeexplore.ieee.org /stamp/stamp.jsp?arnumber =01648294 .

[13] Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E., (2007). Ontology Visualization Methods - A Survey, *ACM Computing Surveys*, Vol. 39, Issue 4.

[14] Kobsa, A, (2004). User Experiments with Tree Visualization Systems. *In IEEE Symposium on Information Visualization (INFOVIS'04)*, 9-16.

[15] Lamping, J., Rao, R., Pirolli, P. (1995). A focus + context Technique based on Hyperbolic Geometry for Visual-izing Large Hierarchies. *ACM Conference on Human Factors in Computing Systems (CHI'95)*, New York, ACM Press. pp 404-408.

[16] Musen, M.A., Fergerson, R.W., Grosso, W.E., Noy, N.F., Grubezy, M.Y., Gennari, J.H., (2000). Component-based support for building knowledge-acquisition systems. *Proc. Intelligent Information Processing (IIP 2000) Conf. Int. Federation for Processing (IFIP), World Computer Congress (WCC'2000)*, Beijing, China, pp 18-22.

[17] Munzner, T., (1997). H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. *Proc. of the IEEE Symp. on Information Visualisation.*, Phoenix, USA.

[18] North, S. C., (1995). Incremental layout in DynaDAG. *Proc. of the Symposium on Graph Drawing GD '95*, Springer-Verlag, pp 409-418.

[19] North, C., Shneiderman, B., (2000). Snap-together visualisa-tion: can users construct and operate coordinated visu-alisations?. *Int. J. of Human-Computer Studies* 53: 715-739.

[20] Noy, N. F., McGuiness D. L., (2001). Ontology Development 101: A Guide to Creating Your First Ontology, *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.*

[21] Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W., & Musen, M.A., (2001). Creating Semantic Web Contents with Protege-2000.*IEEE Intelligent Systems*, 16, 60--71.

[22] OntoViz URL: http://protegewiki.stanford.edu/index.php/OntoViz.

[23] Protégé URL : http://protege.stanford.edu/

[24] Risden, K., Czerwinski, M.P., Munzner, T., Cook, D., (2000). An initial examination of ease of use for 2D and 3D information visualizations of web content. *Int. J. of Human-Computer Studies* 53: 695-714.

[25] TGVizTab Protégé plug-in URL: http://users.ecs.soton.ac.uk/ha/TGVizTab/

[26] Wills, G. J., (1997). NicheWorks - Interactive Visualisation of Very Large Graphs. *Proc. Graph Drawing* '97, Rome, Italy, Springer-Verlag. pp 403-414.

Authors

R. Sivakumar received his M.Phil degree (1996) in Computer Science from Regional Engineering College, India and Ph.D degree (2005) in Computer Science from Bharathidasan University, India. He has published a number of articles both in national level and international level journals. He has also completed a minor research project on Bioinformatics and currently working on developing visualization tools for health informatics. His areas of interest are Data mining, HCI, Ontology and Informatics Systems.

P.V. Arivoli received his M.Phil degree (2008) in Computer Science from Bharathidasan University, India. He is currently working as a research project fellow in "Development of Visualization Methods to Integrate Patient Records for the domain of Acquired Immune Deficiency Syndrome (AIDS)" at A.V.V.M. Sri Pushpam College, Bharathidasan University, India. His areas of interests are HCI and Ontology.