

A COMPARATIVE STUDY OF RECENT ONTOLOGY VISUALIZATION TOOLS WITH A CASE OF DIABETES DATA

V. Swaminathan¹, R. Sivakumar²

¹Department of Computer Science, A.V.V.M. Sri Pushpam College, Bharathidasan University, Trichirappalli, India

Email: vswaminathanthanjavur@yahoo.com

²Department of Computer Science, A.V.V.M. Sri Pushpam College, Bharathidasan University, Trichirappalli, India

Email: rskumar.avvmc@gmail.com

Abstract: *Ontology is a conceptualization of a domain into machine readable format. 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. A number of ontology visualizations exist that have been embedded in ontology management tools. The primary goal of this paper is to analyze recently implemented ontology visualization tools and their contributions in the enrichment of users' cognitive support. This work also presents the preliminary results of an evaluation of three visualization tools to determine the suitability of each method for end user applications where ontologies are used as browsing aids with a case of Diabetes data.*

Keywords: *Ontology Visualization, Semantic Web, Knowledge retrieval, Reasoner.*

I. INTRODUCTION

An ontology is a formal, explicit specification of a shared conceptualization. Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group. To build ontologies is complex and time consuming, and it is even more if ontology developers have to implement them directly in an ontology language, without any kind of tool support. To ease this task, in the mid – 1990s the first ontology building environments were created. The provided interfaces that helped users carry out some of the main activities of the ontology development process, such as conceptualization, implementation, consistency

checking, and documentation. In the last few years, the number of ontology tools has greatly increased and they have been diversified. Gomez-perez [1] distinguishes the following groups:

Ontology development tools group includes tools and integrated suites that can be used to build a new ontology from scratch. In addition to the common edition and browsing functions, these tools usually give support to ontology documentation, ontology export and import to/from different formats and ontology languages, ontology graphical edition, ontology library management, etc.

Ontology evaluation tools are used to evaluate the content of ontologies and their related technologies. Ontology content evaluation tries to reduce problems when one needs to integrate and use ontologies and ontology-based technology in other information systems. Ontology merge and alignment tools are to solve the problem of merging and aligning different ontologies in the same domain. With Ontology - based annotation tools users can insert instances of concepts and of relations in ontologies and maintain (semi)automatically ontology-based markups in web pages. Most of these tools appeared recently, in the context of the semantic Web. Ontology querying tools and inference engines allow querying ontologies easily and performing inferences with them. Normally, they are strongly related to the language used to implement ontologies.

Ontology learning tools can derive ontologies (semi)automatically from natural language texts, as well as semi-structured sources and databases, by means of machine learning and natural language analysis techniques.

In recent years, number of ontology tools have been designed and implemented with the support of visualization. The area of cognitive assistance much requires visualization techniques for its improvement in performance. 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. A number of ontology visualizations [2] [3] exist that have been embedded in ontology management tools (e.g. <http://protege.stanford.edu/> and <http://kaon.semanticweb.org/>) and are used as information retrieval aids in applications that use ontologies [4]. Evaluations of ontology visualization effectiveness, however, are up to this point scarce: [5] presents some user experiments focused on tree visualization systems, whereas [6] 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 [7] fisheye views [8], hyperbolic [9] and 3D hyperbolic layouts [10], aiming to help comprehend and analyse complex information structures. Preference of visualisation models vary according to the user's needs and query context [11]. It is also dependant of the type and extent of the visualised network. Using a combination of integrated visualisations of various types has shown to be sometimes beneficial [12][13]. Complex networks of multi-dimensional hierarchies and arbitrary relations are becoming common characteristics of current ontologies. Visualising large networks has always been challenging. [14][15] Surveyed a wide range of visualisation techniques and concluded that all existing algorithms have a size limit after which they cannot cope [16] and [17] 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 visualization should therefore turn some of these semantics more explicit [14] Spring-layout algorithms [18] 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 identifying developments of cognitive support in recently implemented ontology visualization tools and present a summary of various features and capabilities of those tools. 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.

Information visualization [19] represents information in a manner which aids in communication and facilitates understanding and exploration. The view of tools requires the user to first understand what the view is attempting to show. Most innovative visualizations suffer from lack of industry acceptance. This lack of adoption restricts meaningful evaluation of the ideas in the tool. With this concept of simplicity in mind, we need to make conscious and deliberate efforts to focus in identifying a tool which would focus on the key goals of communication and understanding, leveraging the techniques that best facilitate those goals. Hence the next purpose of this paper is to

evaluate the degree of cognitive support offered in selected three different ontology visualization tools with the help of end user groups.

The remaining part of this paper is organized as follows: Section 2 gives the introduction to Protégé. Next Section 3 gives a survey on features developments in recent ontology visualization tools. Followed which, section 4 comes out with evaluation of selected tools to determine their effectiveness in cognitive support. Section 5 concluded with future work.

II. PROTEGE -2000

Protege -2000 [20] is the latest version of the protégé line of tools, created by the Stanford Medical Informatics (SIM) group at Stanford University. The first protégé tool was created in 1987[21]; its main aim was to simplify the knowledge acquisition process for expert systems. To achieve this objective, it used the knowledge acquired in previous stages of the process to generate customized forms for acquiring more knowledge. Since then, Protégé has gone through several releases and has focused on different aspects of knowledge acquisition (knowledge bases, problem solving methods, ontologies, etc.), the result of which is protégé-2000. The history of the protégé line of tools was described by Gennari and colleagues [22]. It has around 7000 registered users. Protégé-2000 is oriented to the task of ontology and knowledge-base development. It is freely available for downloading under the Mozilla open-source license. The current version is 1.8(April 2003). Protégé-2000 is a Java-based standalone application to be installed and run in a local computer. The core of this application is the ontology editor, described further. Protégé-2000 has an extensible architecture for creating and integrating easily new extensions (aka plug-ins). These extensions usually perform functions not provided by the protégé-2000 standard distribution (other types of visualization, new import and export formats, etc.), implement Applications that use protégé-2000 ontologies, or allow configuring the ontology editor. Most of the plug-ins are available in the protégé Plug-in Library, where contributions from many different research groups can be found.

The three groups of plug-ins that can be developed for protégé-2000 with actual examples of such types of plug-ins are described below:

The first one, Tab Plug-ins are the most common types in Protégé-2000, and provide functions that are not covered by the standard distribution of the ontology editor. To perform their task, tab plug-ins extends the ontology editor with an additional tab so that users can access its functions from it. The following functions are covered by some of the plug-ins available: ontology graphical visualization (Jambalaya tab and Onto Viz tab), ontology merge and versioning (PROMT tab), management of large on-line knowledge sources (UMLS and WordNet tab), OKBC

ontology access (OKBC tab), constraint building and execution (PAL tab), and inference engines using Jess[23], Prolog, FLogic, FaCT, and Algemon (Jess, Prolog, FLORA, OIL and Algemon tabs respectively).

The next one, Slot widgets are used to display and edit slot values without the default display and edit facilities. There are also slot widgets for displaying images, video and audio, and for managing dates, for measurement units, for swapping values between slots, etc.

Finally, Backends enables users to export and import ontologies in different formats: RDF Schema, XML, XML Schema, etc. There is a backend for storing and retrieving ontologies from databases so that not only ontologies can be stored as CLIPS files (the default storage format used by Protégé-2000) but they can also be stored in any database JDBC compatible. Recently a backend to export and import ontologies in XML has been made available

III. FEATURES OF RECENT ONTOLOGY VISUALIZATION TOOLS

This section surveys the recently implemented ontology visualization tools with a scope of extended features.

A. OWL2Query (2011)

This is a conjunction query cum meta-query engine and visualization plug-in. It facilitates the creation of queries using SPARQL or intuitive graph based syntax and evaluates them using any OWL API compliant reasoner. This tool belongs to the category Tab Widget and application. This tool incorporates several new features such as toolbar, prefix editor, variable editor, layout editor, query graph, SPARQL query view, SPARQL-DL preview, result panel, edge editor, property editor and Abox, Tbox & Rbox node editors. These help the development of effectiveness of user's cognitive support.

B. Onto Graf (2010)

This tool is designed for Protégé-OWL application. It offers support for interactively navigating the relationships of created OWL ontologies. It supports various layouts for automatically organizing the structure of designed ontology. It is compatible with Protégé-OWL 4.1 and 4.2. It is available in various versions like 1.0.1, 0.0.5, 0.0.4, 0.0.3, 0.0.2 and 0.0.1. The 1.0.1 version fixed a weird generics problem with Java6. On the other hand the 0.0.5 version added support to export the current visible graph into a DOT language format file. The addition of new view for visualization OWL imports and adds support for pinning the tool tip display to show multiple tool tips at once and this is offered by version 0.0.4. By the mean-time 0.0.3 version added new tool tips exporting graphs as images and saving/opening graphs. This tool incorporates additional features such as focus on home, grid alphabet, radial, spring, tree-vertical & horizontal directed, zoom-in, zoom-out, no-zoom,

node-type, arc type and search (contains, start with, end with, exact match, reg exp).

C. 3.3 DL Query (2008)

It offers the facility of quick test definition of classes to see that they subsume the appropriate subclasses or to test for class membership of arbitrary descriptions without having to create named class. This tool belongs to the type Tab Widget. It is designed for the application of Protégé-OWL. This tool adds effectiveness to the user's cognitive support by introducing the following features: Query box, execute option, object properties, class hierarchy window, data properties, checkboxes for super class, ancestor class, equivalent class, subclass, descendent class and individuals and Query result frame.

IV. STUDY OF PERFORMANCE

An experiment was conducted in order to evaluate users' satisfaction on extended features implemented by three different ontology visualization tools. As a part of our real time project, we created ontologies for DIABETES patients' information. The experiment described in this work was designed in order to provide useful insight concerning three research areas, which are:

- The evaluation of three ontology visualization tools- DL Query, OntoGraf and OWL2Query.
- The strategies and techniques employed by the users while researching DIABETE information.
- The evaluation of the DIABETES ontology created by our research project.

This paper limits the discussion to the results concerning the advantages and disadvantages of the three ontology visualization tools. The focus of this experiment was not overall ontology management and editing, but rather information retrieval and assessing the stability of each method for end user applications where ontologies are used as browsing assisters. This section gives an overview of the performed evaluation, containing brief descriptions of the evaluation user group, the ontology used, the query types used for information retrieval through the ontology, the description of the evaluation sequence and the results.

A. Users for Evaluation

In order to examine the effectiveness of the evaluated ontology visualization, a user group with both computer skills and basic domain knowledge was chosen. The choice of ontology was such that all the users could have at least some familiarity with computers and domain. This fact ensured that there would not be significant differences in the performance of the users due to complete lack of knowledge of the domain. Then the ontology created for the "DIABETES" domain was chosen. Most of the users that participated to the experiment were the

research scholars of Computer Science and Information Science departments of Sri Pushpam College of Bharathidasan University, India. All these users have some knowledge both in computers and DIABETES domain as they are instructed to be familiar for testing. But they vary in degree of skills both in computer and domain knowledge respectively. The user group was divided into two commensurate groups of eight users one that got a short introduction in how to use these testing tools, and one without any prior knowledge about ontology editing.

B. Ontology Description

The ontology used in this experiment is an effort to describe the domain of the DIABETES - THANJAVUR. It presents the current DIABETES patients information under the treatment and also contains relevant information about the symptoms and causes of DIABETES. It contains 15 classes. It is populated with 257 instances. The maximum depth of the "is-a" taxonomy tree is 13 classes. Multiple inheritances have been employed for about 2 classes and no other classes are having more than one parent.

C. Experimental set-up

Before starting the evaluation process we had to perform some preliminary tests in order to decide upon the visualization method set-up to be used in the experiment.

Bearing in mind that we were investigating the most suitable visualization not for ontology developers but for users that will use the ontology as an information retrieval aid, we had to keep the visualization method controls as simple as possible. Furthermore, for the size of the experiment ontology, some visualization set-ups were really cluttered and not at all useful for information retrieval. In the case of DLQuery, Query box, execute option; object properties, class hierarchy window, data properties, checkboxes for super class, ancestor class, equivalent class, subclass, descendent class and individuals and Query result frame were introduced to the users. In the case of OntoGraf, focus on home, grid alphabet, radial, spring, tree-vertical & horizontal directed, zoom-in, zoom-out, no-zoom, node-type, arc type and search (contains, start with, end with, exact match, reg exp) options were introduced to the users. Finally, for the case of OWL2Query, we introduced the following features to the users: toolbar, prefix editor, variable editor, layout editor, query graph, SPARQL query view, SPARQL-DL preview, result panel, edge editor, property editor and Abox, Tbox & Rbox node editors.

D. DIABETES ontology information retrieval tasks

This experiment constructed five different queries to retrieve information from DIABETES ontology. The queries are then grouped into different types according to ontology related criteria, such as the number of different classes they entail, if they are relevant to the ontology hierarchy or not, if they ask for the number of classes of instances with a common

characteristics etc. The identified query types are presented in the following text, along with brief description examples.

1. The user is given the value of a slot of an instance, and is asked to find the value of another slot of the same instance. For example, "What is the year of registration of the patientid 101?"

In this case the user has to locate a specific instance and then extract a slot value to find the answer to the query.

2. The user is given the value of a slot of an instance I1, and is asked to retrieve a slot value of some instance I2, linked to I1 through a role relationship. For example, "What is the year of identification of the Type 1 that a particular patient affected with?"

In this case the user should first locate I1, follow the role relationship to I2 and then extract a slot value to find the answer to the query.

3. Query related to the class hierarchy, the taxonomy. In this case, a class is described to the user and he/she is asked to retrieve its direct subclasses. For example, "What are the symptoms of the "Type 1"?"

In this example, the result is a set of class names, which should be organized hierarchically.

4. Querying for the number of instances of a specific class. For example, "What is the number of the drug manufacturer?"

In this case the result is a number, so the user has to locate the specific instances and count them or view their number if this feature is provided by the interface.

5. Retrieve the number of instances with a specific common slot value. For example, "What is the number of patients prescribed with a specific drug?"

In this case, as in query 4 of the previous section, the result is a number, so the user has to locate the specific instances and count them (or view their number if this feature is provided by the interface). However, this case is somewhat more complicated as not all the instances of an entity are requested, but a sub-set of them with a common slot value. The user groups were involved in retrieving the answers for those queries using DL Query, OWL2Query and OntoGraf.

E. Performance Evaluation

For each task (query), we measured the NASA Task Load Index (TLX) [24] and the time that was needed to perform the task. Figure 1 show the results of the comparison of OWL2Query, DLQuery and OntoGraf ontology visualization tools based on the visualization scores as perceived by the users. As we expected, we can see that the group that was given a short introduction into the tools, performed better on average. It scored lower TLX and shorter time.

For task1, there was almost no difference observed between the two groups with respect to Onto Graf. The figure 1 clearly shows Onto Graf let to a lower TLX on average in all five tasks compared to other two tools such as OWL2Query and DLQuery, that is, users were less frustrated and mentally stressed. That is very likely also an outcome of the reduced time users had to spend for each task. The figure shows that users could solve each task faster using Onto Graf. On average users spent approximately 58.9 % less time with Onto Graf and had a 24.9% lower TLX than when using OWL2Query and 32.8 % less time with Onto Graf and had a 15.9% lower TLX than when using DLQuery. The comparison only shows that Onto Graf is better suited for the use case of DIABETES ontology we described. DLQuery and OWL2Query are however much more feature enriched tools which make it difficult to use than Onto Graf for our experiment. We evaluated only information retrieval process that was possible with these three tools.

import/export format, graph view, consistency check, version, dependencies, type, multi-user type, web support, library support and etc., The result of this survey and analysis provides comprehensive understanding of new features that enhance cognitive support. Finally, we presented some preliminary results from a comparative evaluation of selected three visualization tools. The results are being further analyzed in order to extract interesting patterns. Furthermore, the results of this evaluation are being analyzed with respect to the other two aspects of the experiment, i.e. the evaluation of the ontology itself and a test of the methods users employ for dealing with various information retrieval types. This work can be extended with other tools/ frameworks for the complete ontology management operations

VI. REFERENCES

- [1] Gomez-perez A Survey on ontology Tools, OntoWeb deliverable D1.3. http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D13_1-0.zip.
- [2] 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.
- [3] Boinski T, Bundnik L, Jaworska A, Mrozinski J, Mazurkiewicz, K (2009) OCS-Domain Oriented Ontology Creation System. Polish Journal of Environmental Studies. 18:3B. pp 35-38.
- [4] Benjamin B, Emmanuel P, Ilaria L, Gennady L (2011) OntoTrix: A Hybrid Visualization for Populated Ontologies. Proceedings of WWW2011, Hyderabad, India
- [5] Katifori A, Halatsis C, Lepouras G, Vassilakis C, Giannopoulou E (2007) Ontology Visualization Methods - A Survey, ACM Computing Surveys, Vol. 39, Issue 4. doi:10.1145/1287620.1287621
- [6] Kobsa A (2004) User Experiments with Tree Visualization Systems. In IEEE Symposium on Information Visualization (INFOVIS'04), 9-16. doi:10.1109/INFVIS.2004.70
- [7] Katifori A, Torou E, Halatsis C, Lepouras G, Vassilakis C (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>
- [8] Johnson B, Shneiderman B (1991) Treemaps: a space-filling approach to the visualisation of hierarchical information structures. Proc. 2nd Int. IEEE Visualization Conference, San Diego.
- [9] Furnas GW (1986) The FISHEYE view: A new look at structured files. Proc. of the Conf. on Human Factors in Computing Systems ACM. pp 16-23.
- [10] Lamping J, Rao R, Pirolli P (1995) A focus + context Technique based on Hyperbolic Geometry for Visualizing Large Hierarchies. ACM Conference on Human Factors in Computing Systems (CHI'95), New York, ACM Press. pp 404-408.

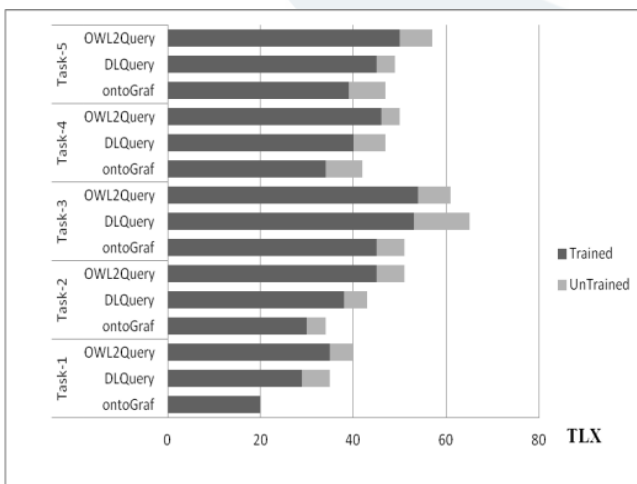
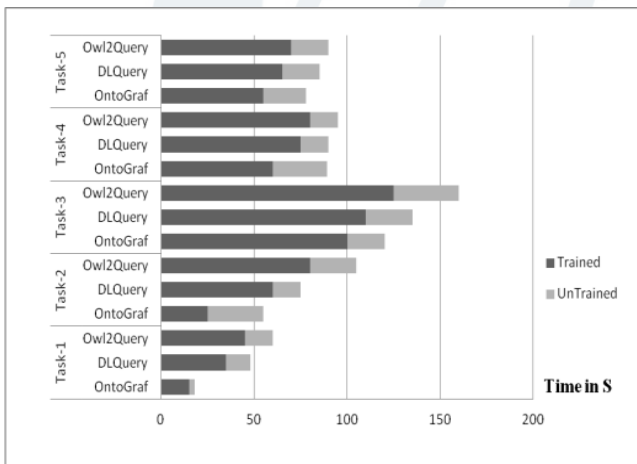


Figure 1. TLX and Time comparison for five different tasks in OntoGraf, DLQuery and OWL2Query

V. CONCLUSION

This paper reported the Performance analysis of advances of cognitive support in recent ontology visualization tools. The review includes both research and commercial category tools. Tools of similar purpose are chosen to analyze their features. This work also included in its analysis information about the

- [11] Munzner T (1997) H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space. Proc. of the IEEE Symp. on Information Visualisation., Phoenix, USA. doi:10.1109/INFVIS.1997.636718
- [12] Graham M, Kennedy J, Benyon D (2000) Towards a Methodology for Developing Visualizations. *Int. J. of Human-Computer Studies* 53(5): 789-807. doi:10.1006/ijhc.2000.0415
- [13] North C, Shneiderman B (2000) Snap-together visualisation: can users construct and operate coordinated visu-alisations?. *Int. J. of Human-Computer Studies* 53: 715-739. doi:10.1006/ijhc.2000.0418
- [14] Ridsen K, Czerwinski MP, 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. doi:10.1006/ijhc.2000.0413
- [15] Wills GJ (1997) NicheWorks - Interactive Visualisation of Very Large Graphs. Proc. Graph Drawing '97, Rome, Italy, Springer-Verlag, pp 403-414.
- [16] Herman I, Melancon G, Marshall MS (2000) Graph Visualization and Navigation in Information Visualisation: a Survey. *IEEE Transactions on Visualisation and Computer Graphics* 6(1): 24-43. doi:10.1109/2945.841119
- [17] North SC (1995) Incremental layout in DynaDAG. Proc. of the Symposium on Graph Drawing GD '95, Springer-Verlag, pp 409-418.
- [18] Eades P (1984) A Heuristic for Graph Drawing. *Congressus Numerantium*,42:149-160.
- [19] Boinski T, Jaworska A, Kleczkowski R, Kunowski P (2010) Ontology Visualization. Proceedings of the 2nd International Conference on Information Technology, ICIT.
- [20] Musen MA (1989) Automated support for building and extending expert models *Machine Learning* 4:347 – 376. doi:10.1007/BF0013071
- [21] Noy NF, Musen MA (2000) PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: Rosenbloom P, Kautz HA, Porter B, Dechter R, Sutton R, Mittal V (eds) 17th National Conference on Artificial Intelligence (AAAI'00). Austin, Texas, pp 450-455.
- [22] Gennari JH, Musen MA, Ferguson RW, Grosso WE, Crubezy M, Eriksson H, Noy NF, Tu SW (2003) The Evolution of Protégé: An Environment for knowledge-Based Systems Development. *International Journal of Human-Computer Studies* 58(1):89-123. doi:10.1016/S1071-5819(02)00127-1
- [23] Friedman – Hill E (2003) *Jess in Action: Java Rule-Based Systems*. Manning Publications Company, Greenwich, Connecticut.
- [24] Musen MA, Ferguson RW, Grosso WE, Noy NF, Grubezy MY, Gennari JH (2000) Component -based support for building knowledge-acquisition systems.
- [25] Proc. Intelligent Information Processing (IIP 2000) Conf. Int. Federation for Processing (IFIP), World Computer Congress (WCC'2000), Beijing, China, pp 18-22.
- [26] DLQuery
<http://protegewiki.stanford.edu/wiki/OWL2Query>
- [27] OntoGraf
: http://protegewiki.stanford.edu/wiki/DL_Query
- [28] OWL2Query:
<http://protegewiki.stanford.edu/wiki/OntoGraf>

How to cite

V. Swaminathan, R. Sivakumar, "A Comparative Study of Recent Ontology Visualization Tools with a Case of Diabetes Data". *International Journal of Research in Computer Science*, 2 (3): pp. 31-36, April 2012. doi:10.7815/ijorcs.23.2012.026