

A Comparative Study of Four Ontology Visualization Techniques in Protégé: Experiment Setup and Preliminary Results

Katifori Akrivi <i>Dept. of Informatics and Telecommunications, University of Athens,</i> vivi@di.uoa.gr	Torou Elena <i>Dept. of Informatics and Telecommunications, University of Athens,</i> etorou@di.uoa.gr	Halatsis Constantin <i>Dept. of Informatics and Telecommunications, University of Athens,</i> halatsis@di.uoa.gr	Lepouras Georgios <i>Dept. of Computer Science and Technology, University of Peloponnese,</i> gl@uop.gr	Vassilakis Costas <i>Dept. of Computer Science and Technology, University of Peloponnese,</i> costas@uop.gr
----------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------

Abstract

The continuing need for more effective information retrieval has led to the creation of the notions of the semantic web and personalized information management, areas of study that very often employ ontologies to represent the semantic context of a domain. Consequently, the need for effective ontology visualization for design, management and browsing has arisen. There are several ontology visualizations available through the existing ontology management tools, but not as many evaluations to determine their advantages and disadvantages and their suitability for various ontologies and user groups. This work presents the preliminary results of an evaluation of four visualization methods in Protégé.

1. Introduction

Recently, the continuing progress in network technologies and data storage has enabled the digitization and dissemination of huge amounts of documents. The need for more effective information retrieval has led to the creation of the notions of the semantic web and personalized information management, areas of study that exploit the semantic context of documents to facilitate their management. In many of the proposed solutions in this field, it is common to include the use of an ontology; therefore, the need for effective ontology visualization for design, management and browsing has arisen.

In the field of ontology visualization there exist several works, employed by the existing ontology management tools and the applications that use ontologies as search aids. The current work presents the preliminary results of the evaluation of selected

visualization methods by a group of users with various degrees of computer expertise. Four commonly used visualization methods, which are representative of the major ontology visualization approaches, have been chosen for the presented experiment. Information retrieval (IR) tasks of different types and varying complexity were posed to users in this evaluation, in order to investigate the strong and weak points of the methods for each IR task class. Special attention is given to queries involving temporal characteristics (e.g. evolution of an entity), which are very useful in certain contexts, such as that of historical archive material ontologies.

The following sections provide useful definitions and related work, followed by a brief presentation of the evaluated methods and their characteristics. The evaluation description and results are then presented with conclusions and future work in the last section.

2. Background and Related Work

As defined in [1], an ontology is a formal explicit description of a domain, consisting of classes, which are the concepts found in the domain (also called entities). Each class may have one or more parent classes (*inheritance* or *is-a* links), formulating thus a specialization/generalization hierarchy, has properties (also called roles or attributes) describing various features of the modeled concept, and restrictions on the slots (called facets or role descriptions). Each slot, in turn, has a type and could have a restricted number of allowed values, which may be of simple types (strings, numbers, booleans or enumerations) or instances of other classes. Allowed classes for slots of type Instance are often called the range of the slot. The classes to which a slot is attached are called the domain of the slot. Classes may have *instances*, which correspond to

individual objects in the domain of discourse; each instance has a concrete value for each property of the class it belongs to. An ontology together with a set of individual instances of classes constitutes a knowledge base.

Various ontology extensions have been incorporated into ontology management tools, as is the case of the *relationships* class in Protégé [2], which is effectively a role relationship that is allowed to have properties.

As it is evident from the ontology definition, the complete visualization of all the ontology features is not an easy task. An obvious approach is to structure the ontology as a tree, based on the is-a links and visualize this tree, while role relations are made available only through the respective slots. This approach is however incomplete, since non-inheritance links that provide useful insight related to the relations between classes are not represented visually. On the other hand, a more complete visualization may offer the missing relationship information, but tends to clutter beyond a certain number of classes and become somewhat unintelligible. Another important issue is that of the representation of the instances, as to the most prominent way of presenting them. Again visualizing all instances of all classes simultaneously can be a major challenge.

There exist a number of ontology visualizations that are being used in the context of ontology management tools or as information retrieval aids in applications that use ontologies. Some interesting ontology management tool surveys may be found in the Protégé web pages [2]. [7] presents the preliminary results of a survey using questionnaires related to ontology editing tools and ontology visualization. However, up to this point, comparative evaluations concerning their effectiveness in different contexts and with different user groups have not been conducted.

The purpose of this work, is to evaluate ontology visualization methods with different characteristics and investigate their effectiveness. The focus of the tests is mostly on information retrieval, and not on ontology editing tools.

3. Ontology Visualization Methods

Most ontology tools, like Protégé [2] or Kaon [6], offer more than one visualization methods for the representation of the ontology, by accommodating *visualization plug-ins*. We chose to use only one tool for the evaluation in order to limit the evaluation to the method characteristics and not the tool functionality. Furthermore, our aim is to identify the advantages and weaknesses of techniques that are already in use in order to contribute to their improvement.

A group of the Protégé ontology visualization methods was selected for the presented experiment,

because it offers a range of different characteristics. Furthermore, Protégé is a very widely used ontology tool and its open source environment presents many possibilities for improvement or creation of new functionality in the form of plug-ins.

The methods chosen for the evaluation are Protégé Class Browser, Jambalaya, TGVizTab and OntoViz and are briefly described in the following paragraphs.

The **Class Browser** is a simple visualization technique that offers a Windows Explorer - like view of the ontology. In this view, the taxonomy of the ontology (as dictated by the *is-a* inheritance relationships) is represented as a tree. It displays the class hierarchy with the lower-level nodes presented as a list under their parent and indented to its right. Classes with more than one parents (multiple inheritance) appear under all their parents. The lists of child nodes may be retracted or expanded at will by clicking or double clicking on their parent. A node may be located using the Search feature available, which, however, only locates classes that are already visible.

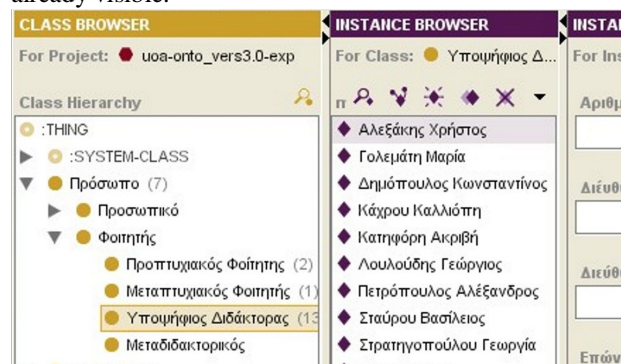


Figure 1. The Protégé Class Browser

The instances of a selected class are displayed in a separate pane to the right of the Class Browser. A search utility (distinct from the class search feature) is also available for searching between the instances already displayed in the Instance Pane.

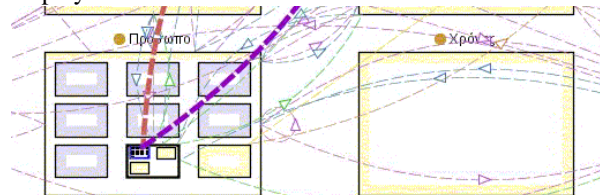


Figure 2. The Jambalaya tab in Protégé

Jambalaya [5] is a visualization plug-in for the Protégé ontology tool [2] that uses the SHriMP (Simple Hierarchical Multi-Perspective) 2D visualization technique. SHriMP uses a nested graph view and the concept of nested interchangeable views, combined with geometric, fisheye and semantic zooming.

According to this method, nested nodes are used to express the inheritance relations between the classes, as sub-classes are nested inside parent classes. Instances are also represented as nested nodes in their corresponding class in the graph. Instance nodes are distinguished from the class ones by their color. Role relations between classes or instances are represented in the graph using directed links between the related nodes.

Users may navigate in the ontology through this visualization utilizing the selection and zoom tools. When a class or instance is selected by zooming on it, the SHriMP view focuses (using a focus technique with animation) on the selected node of the nested graph. When the class or instance is double-clicked, the view focuses on the clicked node and opens a form with the node information, embedded in the visualization. The visualization also offers extra navigation buttons like “back” or “home”.

Jambalaya contains a more advanced keyword search than the other methods, allowing the user to search the whole ontology (classes and instances alike) or limit the search scope by specifying the type of the searched item. Search refinement is also available by searching within the results.

TGVizTab (Touchgraph Visualization Tab) [3] uses a spring-layout technique where nodes repel one another, whereas the edges (links) attract them. This results in placing the semantically similar nodes close to one another. A characteristic of this technique is that it is especially interactive, as the nodes move and adjust to the user commands. The user may also expand or retract nodes, hide them, rotate the graph and change the zoom level.

Figure 3 presents the interface of the TGVizTab. The ontology is also presented as a tree structure on the left (Class Panel). For the purpose of the evaluation, this panel was hidden in order for the user to employ only the TouchGraph visualization.

The Touchgraph visualization displays classes and instances as nodes with different colors. The relations are represented as links. The user has full control on the color and visibility of the links. Each link has a label, which appears only on mouse-over, so as not to clutter the visualization. The is-a links are denoted as “sub” links and the role links have a label with the name of the relation they represent. By right-clicking on a node, the node menu appears, which allows the retraction and expansion of nodes, the display of node information as a form, and their deletion.

This visualization allows the users to navigate making visible gradually parts of the graph. A variable radius of visibility is used to limit the size of the displayed graph in smaller, more manageable sizes, or expand it to include more information. When the user double-clicks on another node, the graph is recreated

with the new node on focus. This transition is presented with animation. When a node is partially expanded, it has a label with the number of hidden links associated with it. The Instances of a selected class may also be presented in the Instance browser on the left. A separate keyword search is available for locating classes and instances; however as in the Class Browser case, it only searches within information that is already visible in the respective window.



Figure 3. The Protege TGVizTab

OntoViz [4] is another Protégé [2] visualization plug-in using a very simple 2D graph visualization method. The ontology is presented as a 2D graph with the capability for each class to present, apart from the name, its attributes slots and inheritance and role relations. The instances are displayed in different color. It is possible for the user to choose which ontology elements will be displayed from the configuration panel on the left. Right-clicking on the graph allows the user to zoom – in or zoom – out.

As can be concluded from the method descriptions, they are representative of the main 2D graph visualization categories. Class Browser represents the intended list taxonomy visualization, very common in the Windows – based applications. The tree structure is represented by OntoViz. TGVizTab offers a focus + context approach, presenting the selected node in the middle of the visualization with its related nodes located around it. Lastly, Jambalaya is an example of a zoomable interface, which allows the user to zoom in the part of the hierarchy that interests him/her.

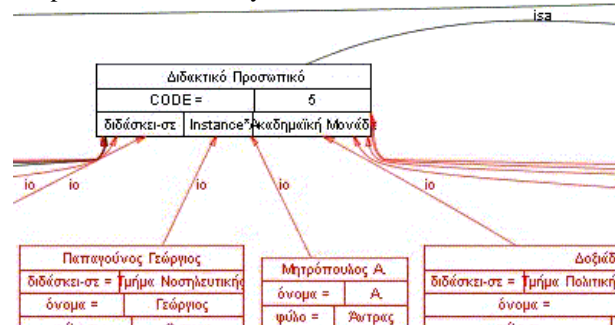


Figure 4. Protege OntoViz visualization.

4. Evaluation Description

The experiment described in this work was designed in order to provide useful insight concerning three research areas, which are:

- The evaluation of 4 ontology visualization methods.
- The strategies and techniques employed by the users while researching historical material.
- The evaluation of the University of Athens ontology created by our group.

In this paper, we will limit our discussion to the results concerning the advantages and disadvantages of the four ontology visualization methods. Our focus was not overall ontology management and editing, but rather information retrieval and assessing the suitability of each method for end user applications where ontologies are used as browsing aids.

This section overviews 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.

4.1. Evaluation user group

In order to examine the effectiveness of the evaluated ontology visualizations, a user group of both computer experts and novice computer users was chosen. The choice of the ontology was such that all the users could have at least some familiarity with the concepts it contained. 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. To this end, an ontology created for the “university” domain was chosen.

Most of the users that participated to the experiment were students of history-related departments and researchers working in the Department of Informatics and Telecommunications of the University of Athens. All these users have some knowledge regarding the concepts of the “University” domain but a varying degree of computer expertise.

The user group was composed of 5 men and 8 women. 8 of them are students or researchers of computer science related departments, while the remaining 5 are students or researchers that have at least once visited the Athens University Historical Archive or another Archive for research purposes.

4.2. The ontology used

The ontology used in this experiment is a first effort to describe the domain of the University of Athens. It presents the current state of the university, but also contains information about the history of certain entities.

It contains 205 classes. It is not densely populated with instances, as about 2/3 of the classes do not have direct instances. The rest share 599 instances, not evenly distributed to the classes. The maximum depth of the is-

a taxonomy tree is 5 classes, whereas the mean depth is 2-3 classes. Multiple inheritance has been employed for about 20 classes; no class has more than 2 parents. More than half of the 176 slots describe relations between classes, for example Professor “teaches at” Department.

4.3. Visualization method set-up

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

Class Browser does not offer different presentation options, so it would be used as is. For the rest, we decided to introduce the users to a subset of the available functions. 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 Jambalaya, the zoom-in tools, the Back and Forward buttons, the Home button and the Search tool were introduced to the users. In the case of Ontoviz, the size of the ontology precluded the presentation of all relation types and nodes at once (due to excessive cluttering), so it was decided to initially present the users with only a subset of classes and their instances being visible. Users were allowed to reconfigure the display, by selecting to display the super-classes of the classes of interest for each specific IR task. For TGVizTab we faced a similar problem with role relations and instances, so we limited the graph to display only inheritance relations. Additionally, the instances were configured to appear only in the Instance Browser pane, and not on the graph.

4.4. Ontology information retrieval tasks

In order to draw more useful results, we pursued to use in this experiment information retrieval tasks which are actually used in the real world. To this end, we studied the queries posed by researchers to the Historical Archive of the University of Athens, and grouped them into different types, according to ontology related criteria, such as the number of different classes they entail, whether they are relevant to the ontology hierarchy or not, if they ask for the number of classes or instances with a common characteristic etc. The identified query types are presented in the following text, along with a brief description and 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 birth of the Professor named Constantin Halatsis?”

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 founding of the Faculty that the department of Philosophy belongs to?” 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 Central University Administration Bodies?” 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 University Museums?”

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 departments of a specific Faculty?”

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.

6. Find instances that have evolved in a given way. For example, “Who became Full Professor after X years from the time they s/he was elected as Associate Professor?”

This query type, if not supported by a powerful query mechanism, needs effort from the part of the user, who has to look for the instances that satisfy the specific conditions making calculations for the time periods.

7. Querying for an Entity Timeline, i.e. for all the information relevant to a specific physical entity that may be located in the ontology. For example, “What are the biographical data present in the ontology related to a person with a specific name or a specific department?”

In this case, the user has to locate all the instances that may be relevant to a specific physical entity and record the related information. Note that

this IR task type differs from the previous one, since in the previous case the classes that should be examined are indirectly identified in the task description (e.g. Assistant professor and Full professor), while in this case any class in the ontology might contribute to the result.

4.5. Evaluation description

Before the beginning of the evaluation, users were trained briefly on the usage of each method. About one hour for each user was dedicated to explaining the concept of an ontology and its features, and to instruct them to the usage of the four techniques. To this end, a small ontology of 20 classes, 45 instances, and 46 slots was created in order for them to use for training; the training ontology was drawn from another domain, to avoid user familiarization to the experiment ontology.

After the training period the users were asked to complete a set of IR tasks using each of the visualization methods. The planning of the evaluation was for all users to use all four methods. However, after using the first method the users would get a notion of the ontology structure and contained entities, a fact that would affect their performance when using the subsequent methods. Another factor affecting objectivity was that a certain IR task could not be posed more than once to the same user, as the result would already be known to him/her.

In order to minimize the effect of the first factor, the order by which the methods were used was not the same for each user. To tackle the second problem,, four different sets of IR tasks were created, and each user was supplied a different set for each method.

Each set of IR tasks included one task from each task type described in section 4.4. The user had to find the answer and note it on the corresponding result form. After testing each method the user was asked to fill in a questionnaire, in order to record his/her impressions from using the method. After using all four methods s/he was asked to fill in a questionnaire with comparative questions related to the methods.

During the experiment, the time needed for a user to complete each IR task was recorded. Failure to complete some task, as well as any comments or reactions and difficulties that users may have had with certain tasks were recorded as well. Users were asked to think aloud, in order to record any comments on the visualizations as well as the users’ search strategies employed for finding the answers.

5. Evaluation Results

The user questionnaires, measured times per task and user actions and comments during the experiment are still being analyzed. This section presents the

preliminary results of this analysis. Firstly, the user comments and evaluator observations are briefly presented, followed by the first results of the statistical analysis being performed on the measured times.

5.1. User comments

The **Class Browser** received a very positive reaction from 11 of the users. Many commented the familiarity with the visualization due to the usage of the Windows Explorer. Some users had problems getting used to clicking on the arrows in order to expand or retract, they expected that clicking on the class label itself would have the same effect. Most of the users complained about the lack of an effective search tool for both instances and classes available in the same tab. Furthermore, some users suggested the addition of “Expand All” and “Retract All” buttons.

TGVizTab received intense but contrasting reactions. Some users disliked it and for some it was the best. The main reason users gave for this was the “spontaneous” movements of the ontology. Some users found it “playful”, “nice” or “funny” while others were not very content having to “chase the concept which is moving by itself” or found the effect “dizzying”. Some users commented that the visualization gave them a clear view of the hierarchy while others found it “chaotic”. It is interesting, however, that even the users that disliked TGVizTab performed well in it. On the other hand, almost all commented, like in Class Browser, the lack of an effective search tool accompanying the visualization and the fact the in some cases labels occlude the ones behind them.

Jambalaya in general got more positive reactions than TGVizTab. Most users commented positively on the effective search tool and the animated transition when double clicking on an instance or class, they liked “flying together with the visualization to locate the information”. However, some noted that they would like the animation to be faster (“I lose time waiting”) or slower (“not enough time to understand the transition”) or to display the steps of the transition to the side. It was interesting that none of the users tried to use the relation links visible and almost all noted as a negative point the appearance of the links and that fact that after browsing some concepts these relation links become so many that they obstruct the view to the visualization. They also noted that labels overlap in the case of many instances.

OntoViz on the other hand received very negative reactions. All users commented on the lack of interaction and had problems with the navigation. They had to drag the scrollbars to navigate. Furthermore, the zoom in and out commands and clicking accidentally on an instance, which resulted in focusing on its class, had as a result the loss of the item on focus. They found the

presentation “poor” and “chaotic” and commented on the lack of a search tool, the fact that some labels are not fully – visible, forcing the user to guess their meaning; absence of sorting (instances are not presented in alphabetical order) was also negatively commented. However, some users commented that the visualization could be effective for smaller ontologies or if the user is very familiar with the ontology as it seemed to them effective for the presentation of hierarchies.

Figure 5 presents the perceived effectiveness of the four methods as recorded by the users.

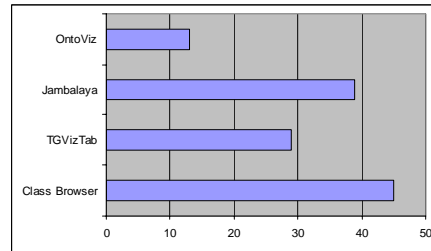


Figure 5. Visualization effectiveness scores as perceived by the users.

5.2. Comparative measured times

This section presents some preliminary results from the statistical analysis performed on the measured times. The analysis was applied to the first 5 task types because not more than 4 users concluded successfully the remaining two task types (6 and 7), thus no statistically sound conclusions can be drawn. This ratio (< 40%) shows that none of the visualization methods is adequate for complex queries that involve time periods or the history of an entity.

Table 1 presents the average and standard deviation of measured times for each of the 1 to 5 tasks and each visualization.

Table 1. Method average times and standard deviations for the first 5 IR tasks.

IR task	Class-Browser		TGViz		Jambalaya		OntoViz	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
1	66	43	65	26	114	119	210	85
2	57	57	123	146	121	94	281	207
3	59	60	90	83	65	56	122	86
4	50	72	76	56	72	62	89	63
5	49	31	41	21	81	107	165	102
All tasks	56	-	79	-	90	-	173	-

At a first glance, the collective average for all 5 tasks shows that the users performed better in Class Browser with TGVizTab and Jambalaya following and OntoViz at the end. The non-parametric Kruskal-Wallis test showed that there is significant difference (confidence >

95%) in these means. Further analysis is needed here however.

The application of non – parametric Mann-Whitney tests on each pair of methods for each task type showed Class Browser was significantly better than Jambalaya and OntoViz for tasks of type 2 (retrieving a slot value of some instance I2, linked to I1 through a role relationship). Class Browser performed better than OntoViz for tasks of type 1 (finding the value of an instance slot given another slot value of the specific instance) and of type 4 (Querying for the number of instances of a specific class).

There were no significant differences between Jambalaya and TGVizTab, however both performed better than OntoViz in tasks of type 1 (see above) and 5 (retrieving the number of instances having a slot with a specific value). TGVizTab performed better than the Class Browser in tasks of type 5, as well.

Finally, the standard deviations reveal that users perform more “uniformly” with Class Browser, while significant variances can be noticed in the times needed to complete certain tasks using the other three visualization methods.

6. Conclusions and Future Work

In this paper we presented some preliminary results from a comparative evaluation of four visualization methods. The results are being further analyzed in order to extract interesting patterns. Besides elaborating on additional aspects of the methods, an increase on the statistical sample is planned in order to get more conclusive statistical results as well as user comments..

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 an investigation of the methods users employ for dealing with various query types. Lastly, we are currently developing a new visualization to be used as a tool for time related queries on ontologies.

7. References

1. Noy, N. F., McGuinness D. L., *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, (March 2001)
2. Protégé project, Stanford University, <http://protege.stanford.edu>.
3. Alani, H., TGVizTab: An Ontology Visualisation Extension for Protégé, *Proceedings of Knowledge Capture (K-Cap'03), Workshop on Visualization Information in Knowledge Engineering*, Sanibel Island, Florida, USA, (2003)

4. Sintek, M., *Ontoviz tab: Visualizing Protégé ontologies*, 2003, <http://protege.stanford.edu/plugins/ontoviz/ontoviz.html>
5. Storey, M.-A., Mussen, M., Silva, J., Best, C., Ernst, N., Fergerson, R., Noy, N, *Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé*, *Workshop on Interactive Tools for Knowledge Capture, K-CAP-2001*, Victoria, BC, Canada, (2001), <http://www.thechiselgroup.org/jambalaya>
6. KAON, <http://kaon.semanticweb.org/>
7. Ernst, N. A., and M.-A. Storey, "A Preliminary Analysis of Visualization Requirements in Knowledge Engineering Tools", University of Victoria, Victoria, CHISEL Technical Report August 19, 2003