Selected Results of a Comparative Study of Four Ontology Visualization Methods for Information Retrieval tasks

Akrivi Katifori[1], Elena Torou[1], Costas Vassilakis[2], Georgios Lepouras[2], Constantin Halatsis[1]
[1] Department of Informatics and Telecommunications, University of Athens, Greece
[2] Department of Computer Science and Technology, University of Peloponnese, Greece
vivi@di.uoa.gr, etorou@di.uoa.gr, costas@uop.gr, gl@uop.gr, halatsis@di.uoa.gr

***Abstract*— The need for effective ontology visualization for design, management and browsing has arisen as a result of the progress in the areas of Semantic Web and Personal Information Management. There are several ontology visualizations available through 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 selected results of an evaluation of four visualization methods in Protégé.**

***Index Terms*— Visualization, Information retrieval, User interfaces**

I. Introduction

THE need for more effective information retrieval has recently lead to the creation of the notions of the semantic web and personalized information management. In many of the proposed solutions in this field, it is common to include the use of an ontology. Consequently, the need for effective ontology visualization for design, management and browsing has arisen.

Numerous research attempts and ontology management tool development efforts have targeted the area of ontology visualization, proposing different approaches to visualizing and facilitating user interaction with ontologies. This work investigates the suitability of four ontology visualization methods for tasks not directly related to ontology management, but rather to information retrieval and for users that are not familiar with the specific content and structure of the visualized ontology. Four commonly used visualization methods, which are representative of the major ontology visualization approaches [8], have been chosen for the presented experiment. In order to assess the appropriateness of each visualization method for different user tasks, we formulated a set of information retrieval tasks and asked a group of users to carry them out, recording task completion times and success rates, as well as other user comments and reactions. The tasks within the set were chosen so as to cover the major task types identified in [8], while their complexity also varied. Tasks involving temporal characteristics (e.g. entity evolution) were also included in the task set, since such tasks often occur in certain contexts, such as information retrieval with the aid of historical archive material ontologies.

The rest of the paper is organized as follows: in section 2 related work is surveyed and useful definitions for ontologies are listed. Section 3 includes brief presentations of the evaluated methods and their characteristics, while section 4 describes the evaluation method and the results obtained from the experiment. Finally, section 6 concludes the paper and outlines future work.

II. Background and Definitions

An ontology, according to the definition in [1] 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 modeled 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. A number of ontology visualizations exist that have been embedded in ontology management tools (e.g. [2] and [6]) and/or are used as information retrieval aids in applications that use ontologies [8]. Evaluations of ontology visualization effectiveness, however, are up to this point scarce: [9] presents some user experiments focused on tree visualization systems, whereas [10] reports on preliminary results from a user study involving four visualization methods. This work aims to further investigate the effectiveness of visualization methods with different characteristics in diverse contexts and with user groups of varying computer expertise and ontology domain knowledge. The evaluation procedure is focused on information retrieval tasks, and not on ontology editing functionality.

III. Ontology Visualization Methods

Most ontology management environments, such as Protégé [2] or Kaon [6], include multiple visualization methods which users may employ to view and interact with the ontology. In our evaluation we opted to use a single tool only, so as to ascertain that the obtained results will not be affected by the functionality offered by the tools (which varies), but will reflect just the effectiveness of the visualization methods.

The visualization methods included in the experiment are the following:

1. Protégé Class Browser [2]
2. Jambalaya [5]
3. TGViz [3]
4. OntoViz [4]

These methods are briefly described in the following paragraphs.

A. Class Browser

Class Browser is a simple visualization technique, representing is-a inheritance relationships through the indented-list paradigm, with subclasses appearing below their superclasses and indented to the right. Users may navigate within the class hierarchy and expand or retract branches; when a class (or multiple classes) are selected in the hierarchy pane, the corresponding instances are shown in the "Instance browser" pane.

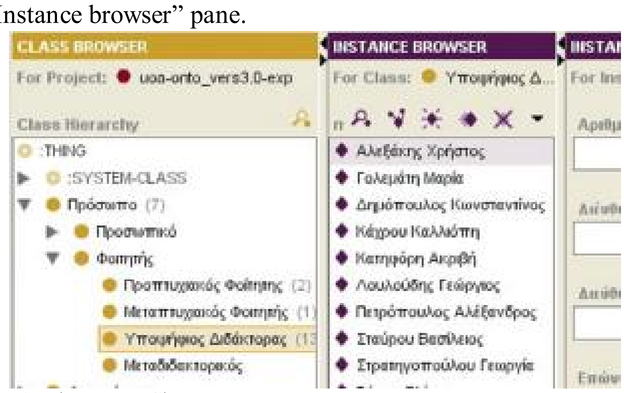

Fig. 1. The Protégé Class Browser.

B. Jabmalaya

Jambalaya [5] employs nested nodes to denote the is-a type relationships among classes. 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. Jambalaya offers the option to display user-defined relationships between classes or instances, which are shown using directed links between the related nodes. When a new node is set as the new visualization focus, an animation sequence is displayed illustrating the is-a relationship path beginning at the former visualization focus and leading to the newly selected one (movements both up and down the ontology hierarchy may be shown in this sequence).

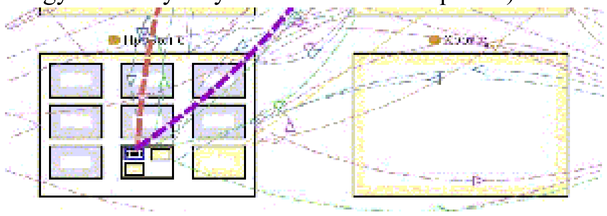

Figure 2. The Jambalaya tab in Protégé

C. TGViz

TGViz Tab (Touchgraph Visualization Tab) [3] depicts the ontology using a *spring–layout* technique. 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. It is worth noting that if the user changes the location of a node or hides/expands/retracts it, the attraction-repulsion forces between nodes are recalculated, resulting to a highly interactive display. rotate the graph and change the zoom level. Graph rotation and zoom options are also available.

The interface of TGViz is shown in Figure 3. Classes are presented in the spring layout area on the right, whereas instances of the selected class are listed in the Instance Browser area on the right.

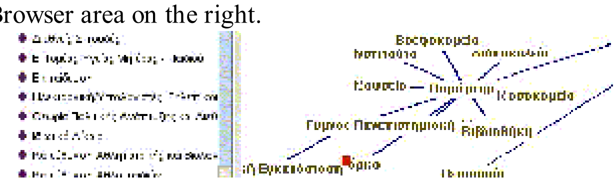

Figure 3. The Protégé TGVizTab

D. OntoViz

OntoViz [4] renders the ontology as a two-dimensional graph using a vertical tree layout where parent/child relationships are derived from the is-a links within the ontology. Ontoviz offers the option to show on the graph not only class names, but also selected slots (slot names together with their values) and relationships. Both classes and instances are included in the graph, and a different color is used for the nodes of different types. The user may also select specific classes/class hierarchies/instances to be visualized, instead of the whole ontology through a configuration panel. Finally, the user may zoom in or out the graph by right-clicking on it.

From the method descriptions presented above, it can be concluded that the chosen methods are representative of the main 2D graph visualization categories [8]. Table 1 presents the correspondence between visualization categories and visualization methods.

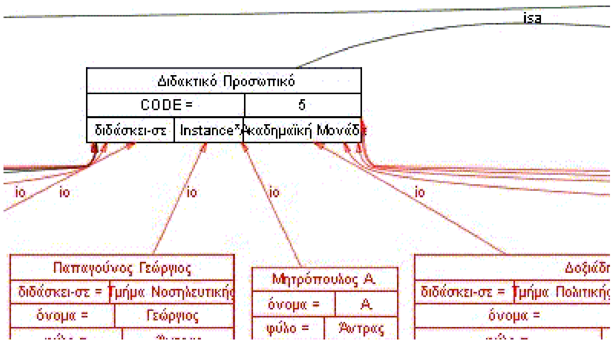

Figure 4. Protégé OntoViz visualization.

TABLE 1
VISUALIZATION CATEGORIES VS. VISUALIZATION METHODS

Visualization category	*Visualization method*
Intended list	Class Browser
Zoomable interface	Jambalaya
Focus + context	TGViz
Node-link/tree	Ontoviz

IV. EVALUATION DESCRIPTION

This section offers an overview of the performed evaluation; the evaluation user group is briefly described, the information retrieval tasks that the users had to carry out are presented and details are given on the procedures followed during user sessions.

A. *User group*

The user group consisted of 37 persons, 25 male and 12 female, the majority (31) of which were students, professors or research staff of departments related to computer science, while the remaining 6 belonged to the Humanities domain.

The participants were selected so as to have some basic expertise with computer use, in order to be able to focus on the use of the Protégé visualization without being hindered by more basic problems, for example, the use of scrolling or selecting from pop-up menus. On the other hand, none of the experiment participants were experienced ontology designers. Eight claimed to understand to some extent the notion of an "ontology" and of them 4 stated that they had used the Protégé ontology editor and 1 the Kaon ontology editor [7].

B. *The Evaluation Experiment Ontology*

The choice of the ontology was such that all the users could have at least some familiarity with the concepts it contained, as it was a "University" ontology and all the users were either students or had a university degree. This fact ensured that there would not be significant differences in performance due to substantial diversities in the level of knowledge on the domain.

The ontology used presents the current state of the University of Athens as well as 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 (most instances are placed underneath leaf classes in the ontology hierarchy, while a number of intermediate classes are used for grouping/organization purposes). The remaining classes have 599 instances in total, whose distribution among the classes varies. The maximum is-a relationship path length is 5 class nodes, whereas the mean path length is 2-3 class nodes. Approximately 20 classes are subject to multiple inheritance, having two parents each; no class in the ontology has more than 2 parents. A total of 176 slots have been defined in the ontology, 55% of which describe relations between classes e.g. a Department "belongs to" a Faculty or a Person "authors" an Article.

C. *Pre-configuration of Visualization Methods*

Before the evaluation commenced, some preliminary work was carried out to determine the best set-up for the visualization methods that would be used in the experiment, since three out of the four selected visualization methods offer a multitude of options.

Class Browser was the simplest case, since no pre-configuration was needed. For the remaining methods, it was decided to make available to the users only a part of the available functionality, aiming to keep the visualization method controls as simple as possible. Since the experiment targeted the use of ontology for information retrieval purposes, ontology editing functions were not introduced at all to the users; moreover, it was decided to avoid certain visualization configurations that lead to excessive display cluttering (e.g. showing all relationship links on the screen as directed arcs), since such setups were practically unusable. The pre-configured settings for the three remaining visualizations were set as follows:

Jambalaya: users were introduced to the zoom-in tools, the Back and Forward buttons, the Home button and the Search tool.

TGViz: only inheritance relationships were visible and instances were shown on a different pane (instance browser), and not within the spring layout area. The initial radius of the spring layout was set to 3 (i.e. classes within a range of 3 is-a relationships from the focused node were shown), the users however could change this value.

Ontoviz: again, only inheritance relationships were visible. Users were allowed to select which portions of the ontology would be visible (specific classes or class hierarchies; users could also select whether instances for each class would be displayed or not).

D. *Information Retrieval Tasks*

As already stated, the focus of this evaluation was not overall ontology management and editing, but rather information retrieval and assessment of the suitability of each method for end-user applications where ontologies are used as browsing aids. The query types used in the evaluation are presented in the following paragraphs.

1) Simple Queries

The query types in this section are characterized as simple because, to our view, the can be answered, without significant effort, only by browsing.

T1: Querying for the value of an instance slot or slots of simple type (which is not a class or instance) given the value of another slot or slots, which identified the instance. An example of such a task would be "In which year was the Department of Informatics and Telecommunications founded?"

T2: Locating an instance by giving an indentifying characteristic, traversing some relationship and retrieving a slot value from the reached instance. For example, "What was the year of founding of the Department that Prof. Halatsis serves in?".

T3: Retrieve the names of multiple class, which should be subclasses of a specific class – e.g. "what are the types of publications found in a university?"

T4: Querying for the number of instances of a specific class – e.g. "how many departments are there in the University?".

T5: Querying for the number of instances with a specific common slot value – e.g. "What is the number of departments of the Science Faculty?".

2) Complex Queries

The query types in this section are characterized as complex because they require more effort. The user may have to locate and count instances with specific characteristics or perform computations of time periods.

T6: Query producing sequences of values for a specific slot. For example, "Who became Full Professor after X years from the time they s/he was elected as Associate Professor?"

T7: Querying for a person Entity Timeline, i.e. to retrieve all information relevant to a specific person that may be located in the ontology. In this case, the user has to locate all the instances that are related to a specific person – for example a specific person may have been a student, later a PhD candicate and subsequently a lecturer.

T8/9: Querying for an institution Entity Timeline, i.e. to retrieve all information relevant to a specific institution, a faculty, museum, etc, that may be located in the ontology. For example, "What are the data present in the ontology related to a university department with a specific name?" In this case, the user has to locate all the instances that may be relevant to a specific institution and record the related information. T8 is somewhat more difficult as the institution evolution presents changes of Class between the instances of the entity – e.g. two Chairs were merged to form a Department which later became a Faculty. T9 was added to the second series of experiments in order to further evaluate the second ontology version. It is easier than T8 in the sense that the entity instances belong to the same class. For example, a museum was split to 3 new museums.

E. Description of User Session

Before each user was asked to carry out the IR tasks, a training session took place which lasted for about an hour. During this training session, users were introduced to the concept of an ontology and its features as well as to the ontology visualization methods to be employed during the experiment.

After the training period, users were asked to complete the set of the 9 IR tasks using each of the visualization methods. The order of the visualizations alternated for different users and, for each visualization, a different set of tasks was given, so as to prevent users from giving answers they merely remembered from the previous visualizations they used. Finally, users were asked to fill in a questionnaire consisting of two parts. In the first part users gave their opinion on various characteristics, the perceived ease of use and usefulness of each visualization. In the second part, users were asked to rate the four visualizations comparatively (1^{st} to 4^{th}).

The experiment conductor monitored the user throughout the experiment and recorded the time taken to complete each task. Failures to complete certain tasks as well as cases that users gave incomplete answers were also recorded; difficulties that users had were logged as well. Users were asked to think aloud, in order to record any comments on the visualizations as well as the search strategies employed for finding the answers.

V. Evaluation Results

For the statistical analysis of the results, Mann-Whitney tests were applied between method pairs. This section presents selected results from the analysis of correct answer percentages, questionnaires, measured times per task and user actions and comments.

A. Correct Answer Percentages

For the simple tasks, the mean correct answer percentage across all visualizations is approximately 87%, with Class Browser at 95%, Jambalaya and TGViz at about 88% and OntoViz last with 78.4%. For complex tasks, Class Browser, Jambalaya and TGViz are at about 40% whereas OntoViz at 34%.

No statistical difference was noted on all tasks for the correct answer percentages of the visualizations. However, observing the percentages, there are comments to be made. Firstly, whereas in simple tasks, the correct answer percentages are high for all visualizations, for the complex tasks they are very low. This suggests that the visualizations were effective to support browsing for locating simple pieces of information but they performed badly when the participant had to combine information in order to produce the answer. For T7, in particular, which included comparing dates between two different groups of "Person" instances, the mean percentage across all visualizations is about 21%, the lowest of all tasks. Detailed statistical figures regarding correct answers are shown in table 2.

TABLE 2
STATISTICS FOR CORRECT ANSWERS FOR INFORMATION RETRIEVAL TASKS

Visualization method	Mean correct answers			StdDev		
	All	Simple	Complex	All	Simple	Complex
Class Browser	70.66	95.0	40.25	30.29	3.67	14.41
Jambalaya	66.78	87.2	41.25	27.61	7.95	19.62
TGViz	64.15	88.0	39.75	35.52	7.84	30.92
OntoViz	53.63	78.4	34.25	37.39	18.28	44.84

B. *Comparative Measured Times*

As the analysis has shown, the overall "winner" of the evaluation is Class Browser, the mean successful completion time of which (74 sec) was found significantly better that that of the other 3 visualizations. The second place is shared by Jambalaya (94) and TGViz (97), which have no significant difference between them, and the last is held by OntoViz (190), which performed significantly worse than the other three. A note to be made here is the considerably greater standard deviation of TGViz (103) as opposed to that of Jambalaya (83), even though their mean is almost the same. This suggests that the users performed more uniformly with Jambalaya, an idea that is re-enforced by the user comments and our observations.

OntoViz had a very poor performance in the measured completion times. However, we should not take these results into account as indicative for all node-link/tree ontology visualizations, as they are mostly due to interaction issues (Ontoviz converts the ontology to a bitmap graphic and displays the latter, providing no means for presenting node details, traversing relationships, expanding/collapsing subhierarchies etc; only scrolling and zooming are available to the user), but rather use them as insight to possible shortcomings of this type of visualization.

For T1, a simple node location task, we did not record a significant difference between Class Browser, Jambalaya and TGViz. On the other hand, the performance of OntoViz was significantly worse, with a mean of 225 seconds against the global mean of 91 seconds. This particularly bad result indicates that node-link/tree visualizations for ontologies of this size (approximately 800 nodes) are not particularly effective for browsing to locate a certain node. This is to be expected, as this visualization consumes more space than any of the rest in order to present the ontology, taking into account that the vertical tree layout used leaves a lot of unused space near the root. As a result, the user has to scroll the whole graph in order to locate a specific node and this may become frustrating, particularly in the case of OntoViz where scrollbars and zoom in and out do not function as the user would expect.

T2 is again a node location task, but a bit more complex, as it involves firstly locating a specific instance and then from it going to a related one to find the value of a specific attribute. In this case, again for the aforementioned reason, OntoViz performed significantly worse. However, Jambalaya was also found to be significantly worse than Class Browser and TGViz significantly better than OntoViz. These results suggest that Jambalaya is not that effective for locating specific nodes. Its zoomable interface involves a constant zoom in and out browsing method, as the class hierarchy is organized in boxes within boxes. This makes systematic browsing a bit difficult, as the user becomes easily disoriented as to what s/he has already visited and what has not. TGViz on the other hand had every class available on screen. Although this produces a set of problems related to node overlap and clutter, the user may browse quickly the whole ontology just by looking at it. Class Browser showed again in this case the best performance. It has the advantage of allowing quick and systematic browsing, being also able to avoid the label overlap problems of TGViz.

For T3 (the hierarchy related task) the user is asked for the sub-classes of a particular node. Class Browser has again the smaller mean completion time (42), but the analysis showed that it is significantly better only than TGViz (86). This is to be expected due to the node overlap and circular node positioning of TGViz.

Task T4 involved locating a class and then finding how many instances it has. The two main issues here were the class location and counting the instances in Jambalaya, TGViz and OntoViz or directly seeing their number, available in Class Browser beside the class name. Class Browser, TGViz and Jambalaya performed significantly better than Ontoviz. OntoViz, as already stated is not very effective for locating a specific class and, again, as it is spread very widely, it requires a lot of scrolling to count the instances. Class Browser had again the best mean completion time as it is both effective for quick browsing and it offers an indication of the number of instances that a class contains.

For T5, locating the instances with a specific characteristic, OntoViz was not included in the statistical analysis, as it had only 2 correct answers. Class Browser was found to be significantly better than the other two. In this case, as the question was of the form "How many departments are there in the Faculty of Philosophy?", most of the participants focused on looking for the Faculty and then finding the appropriate slot, or in some cases looked in the department instances to try and find what Faculty they belong to. When they adopted the second option, browsing became difficult as they would have to look to every department to see the faculty. In this task, the user's choice regarding the search strategy had a serious impact on the completion time.

T6 was a task of the complex tasks group that involved comparison of certain slots of instances belonging to 2 different classes. Users had to first locate the classes and then find the appropriate way to extract the requested information. For example, they had to find out who became Lecturer 5 years after the completion of his/her PhD. The mean time for all the visualizations for this particular task is significantly greater than those of all the other tasks. This was also the most frustrating task for the users. The basic reason for this is that they lost a lot of time on realizing that the information they were looking for would be found by doing the particular comparison. In this case, Class Browser has proven to be the most successful, as it allowed a systematic and organized browsing of the ontology and allowed the users to concentrate more on the way to perform the task and less on locating specific classes and moving from one class to the other. OntoViz had only two successful task completions and was consequently excluded from the analysis.

For tasks T7, T8 and T9, which were all related to entity evolution, there is no significant difference in user performance for the three visualizations. (OntoViz was again excluded for the same reasons as for T6.)

The analysis of the total means for the simple and complex task groups revealed that in the simple tasks group the same order as before was noted, as to the performance. On the other hand, in the complex tasks group, significant difference has been found only between Class Browser, TGViz and Jambalaya, on one hand, and OntoViz, on the other, verifying one more time the low performance of OntoViz in this experiment. As already stated, complex tasks are not as visualization-related as simple ones. In order to answer them by browsing alone (i.e. without the support of a supplementary search tool) it is necessary to spend a considerable amount of time in finding the way to extract the information from the visualization. In the case of complex tasks, the successful and quick completion is not relevant so much to the visualization itself as to the ontology model and also what the user expects to see. Especially concerning evolution-related tasks, the four evaluated visualizations were not found sufficient to support successful and rapid completion.

Detailed statistical figures regarding task completion times are shown in table 3. Mean times are measured in seconds.

TABLE 3
STATISTICS FOR INFORMATION RETRIEVAL TASK COMPLETION TIME

Visualization method	Mean completion time			StdDev		
	All	Simple	Complex	All	Simple	Complex
Class Browser	74	56	133	72	84	59
Jambalaya	94	80	144	83	70	103
TGViz	97	77	181	103	77	160
OntoViz	190	177	451	163	147	151

C. User Comments

The **Class Browser** in general received neutral to positive reactions. The vast majority of the users grasped its functionality easily and had no problems browsing the ontology with it. It was characterized as "simple", "concise", "easy to learn" and "intuitive". Some commented on their familiarity with the visualization due to the use of the Microsoft Windows Explorer. Most of the users believed that it is useful for quick information finding and easy and efficient to navigate. Some users commented positively on its presentation, characterizing it as "nice", "pleasant" and "serious". The implicit comparison with the other three visualization methods lead them to positive comments on the fact that it shows the number of instances of a class, it allows two instance windows to be simultaneously open and it is "static", not changing the class and instance structure and order and, as a result, facilitates easy information re-finding.

The same characteristics, however, were perceived as drawbacks by some users, leading to negative comments. Class Browser was thus characterised as "more boring" and "too serious-looking". Some users commented on the classes not being presented in alphabetical order, which made browsing more difficult. Furthermore, some suggested the addition of "Expand All" and "Retract All" buttons.

TGVizTab received intense but contrasting reactions. The main advantage commented by most of the participants was that the visualization offered a good overview of the ontology, as most of the classes are visible on screen. Some participants found it "easy" to learn and to use as well as "interesting" and "different". Furthermore, some commented positively on its interactivity and the nodes' "spontaneous" movement.

On the whole, the basic problem of TGViz commented by the users was the fact that it was too "alive", it seemed to have a will of its own and move like a living organism. Although this was found exciting by some users, most found it disorienting. They were not very content having to "chase the concept which is moving by itself" or found the effect "dizzying" and "frustrating". A serious issue was that when re-drawing the ontology, the only functionality offered close to a "home" one, the nodes were placed in different positions, resulting in the participant having to re-locate previously found classes. There was overlap between the nodes, with some labels almost completely occluded. The visualization did not offer a clear view of the hierarchy to the participants and some of them commented that the information is too concentrated on the screen, making the visualization "chaotic" and node location very difficult.

Jambalaya in general got positive reactions. This is probably due to the fact that its presentation and animation makes it "initially impressive" as one user stated. Many users found it easy to learn and use, intuitive and pleasant, as well as aesthetically pleasing. They liked the organization of the ontology in “boxes” and “diving into them” to locate information. The animated transition received positive comments by some of the users; when double clicking on an instance or class, they liked “flying together with the visualization to locate the information”. Some suggested that the animated transition is in fact useful for learning the ontology structure.

However, there were certain features of Jambalaya that were commented negatively. The prolonged use of animated transition proved to be dizzying and frustrating. Many users commented on that, and especially towards the last tasks many of them showed the tendency to avert their gaze from the screen during the animated transition. There were also comments on the animation speed. Most users would like the animation to be faster (“I lose time waiting”) but also some would like it to be slower (“not enough time to understand the transition”). 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 the 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 received very negative comments and it was a source of distress for the participants. As one of the users described, “it is like looking for your car in a 10 acre parking lot looking through a microscope”.

Very few of the participants had positive comments for OntoViz: They believed that it presented a nice view of the hierarchy and that it was easy to understand and effective for simple queries. Some commented that the visualization could be usable for smaller ontologies or if the user is familiar with the ontology as it seemed to them effective for the presentation of hierarchies.

As already stated, the main problem of OntoViz was its lack of interactivity. The participants felt that it was “huge” and “overwhelming”, a huge image they had to explore only visually. It gave the impression of untidiness and little information to offer. If many entities are visualized at the same time, links are not easily discernible. In many cases labels were occluded and the participant had no way of reading the label as it was not possible to move or expand it. Another main problem of this implementation of a tree visualization was the fact that it lost the node on focus very easily.

D. Questionnaire Results

In all questions except the ones on how appealing and interesting the visualization was, Class Browser was found significantly better than the rest of the visualizations with a mean score of 7.9/9. In these two, its mean score was the same with Jambalaya. The standard deviation of Class Browser is in general smaller than that of the other methods. This suggests a more uniform reaction of the users towards it. On the other hand, TGViz has in most cases a slightly higher standard deviation than the rest, which again confirms the users’ contrasting reactions towards it.

The total mean score of OntoViz was significantly lower than that of the other three visualizations (3.4). The mean scores of Jambalaya were in general greater than those of TGViz, with a total mean of 6.84 and 6.03 respectively, suggesting a significant difference of $p<0.008$ for the total means.

VI. Conclusions and Future Work

In this paper we presented selected results from a comparative evaluation of four visualization methods. The indented list visualization Class Browser was the overall “winner” in both measured times, correct answer percentages and questionnaires. The increased familiarity of the participants with this visualization does not seem to be the major reason of Class Browser’s success, as it is a method that offers a clear view of the hierarchy without label overlap combined with the possibility for quick and systematic browsing as well as a “static” node positioning that favours node re-finding.

OntoViz on the other hand performed really poorly. This performance may not be considered characteristic of node-link and tree visualizations, as this particular implementation had serious interactivity problems. However, for this range of ontology sizes, such a visualization without sophisticated mechanisms for node positioning and retraction/expansion has little chance of gaining user acceptance.

Jambalaya and TGViz, despite the fact that had almost the same mean performance and correct answer percentages, seem to differ in the questionnaires and user comments. TGViz was rated lower than Jambalaya in the questionnaires and commented negatively in general. The animated movement of the nodes seemed to produce a feeling of uneasiness and lack of control. Jambalaya on the other hand, although it seemed more appealing initially, it was not very effective for systematic browsing and the animation proved to be dizzying.

Future work includes the more focused study of individual visualization features, as well as the creation of a visualization for entity evolution.

References

[1] Noy, N. F., McGuiness 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

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

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

[10] Katifori , A., Torou, E., Halatsis, C., Vassilakis, C., Lepouras G. A Comparative Study of Four Ontology Visualization Techniques in Protégé: Experiment Setup and Preliminary Results, Proceedings of the 10th International Conference on Information Visualisation (IV06).