Visualizing the Evolution of Ontologies: A Dynamic Graph Perspective

Michael Burch[1] and Steffen Lohmann[2]

[1] VISUS, University of Stuttgart, Germany
[2] VIS, University of Stuttgart, Germany

Abstract. Ontologies can be represented as graphs, since they essentially comprise a set of interconnected concepts describing a certain field of knowledge. Consequently, ontologies are often visualized as graphs, using different visual notations and common graph drawing techniques. The visualization of static graphs has been researched a lot, but when it comes to time-varying graphs, researchers face much more challenges in order to design useful, readable, and intuitive visualizations. If we have to deal with dynamic, i.e., evolving and time-dependent ontologies, we have to adapt existing visualization techniques to this challenging problem or develop new ones. In this position paper, we take a look at the visual representation of time-varying ontologies, and provide a discussion from a dynamic graph visualization perspective.

Keywords: time-varying ontologies, graph-based ontology visualization

1 Introduction

Visualization can be a powerful tool in the exploration and analysis of ontologies. Advanced visual designs are demanded to effectively and efficiently manage, browse, and navigate ontologies, and to finally gain insights and conclusions. The growing number and size of ontologies, on the other hand, require sophisticated visualization techniques that are capable to handle algorithmic, perceptual, and visual scalability problems. Consequently, developers of ontology visualizations need to enhance their visual designs by developing faster and better layouts, by providing more visual features, and by improving existing approaches based on user studies and other evaluation methods.

Although various techniques for ontology visualization have flourished over the past years [10,26,28,31], we are still facing the challenging problem of visually handling time-varying ontologies, i.e., ontologies that change over time. Ontologies are often not static but have an inherent dynamic behavior, making them an evolving data structure that is worth researching.

Since time-varying ontologies can be regarded as some kind of dynamic graph, we discuss the applicability of existing visualization techniques for dynamic graphs surveyed by Beck et al. [6], while we basically distinguish between time-to-time mappings (animations) and time-to-space mappings (static displays).

2 Related Work

Numerous approaches for ontology visualization have been presented in the last couple of years [10,26,28,31]. Most of them represent ontologies as graphs, while the graphs are typically rendered as node-link diagrams in a force-directed, hierarchical, or radial layout [31].

Examples for force-directed graph visualizations of ontologies are provided by TGViz [1], NavigOWL [25], and VOWL [31]. Hierarchical graph layouts depicting the inheritance tree of ontologies are used in OWLViz [23] and OntoTrack [29], among others. There are also approaches that represent the inheritance tree with treemaps [35], nested circles [40], or other visualization techniques for hierarchical tree structures, such as hyperbolic trees [11].

Fu et al. [14] conducted a user study where they compared graph visualizations of ontologies with indented tree representations. They found that the graph visualizations are perceived as "more controllable and intuitive without visual redundancy, especially for ontologies with multiple inheritance". They are considered "more suitable for overviews" and "held [the] attention" of the study participants better than trees [14].

Some approaches combine different techniques to visualize ontologies, such as node-link diagrams and adjacency matrices [4], or provide various graph layouts that the users can choose from depending on their task [13,22]. Others apply techniques such as hierarchical edge bundling [21] to increase the readability of the graph visualization, while yet others propose 3D graph visualizations for ontologies [7,15].

Furthermore, there are diagrammatic approaches that use UML [5] or similar graph-based notations [12,38] to visualize ontologies. For instance, COE [18] adopts the popular idea of Concept Maps [32] and applies it to the visualization of OWL ontologies, while a similar attempt has been made with Concept Diagrams [24] that particularly consider the logic of OWL.

However, the visualization of time-varying ontologies has not received any attention in all these approaches. Although the evolution of ontologies has been subject to research [20,33,36], we are not aware of any approach that visualizes evolving ontologies over time. There are methods to compute and analyze differences between two or more versions of an ontology [16,17,34], and tools that display such differences using rudimentary visual properties [17,27,35]. However, we do not know of any sophisticated visualization that supports the detailed analysis of ontologies at different points in time and assists users in the detection of dynamic patterns and trends in time-varying ontologies.

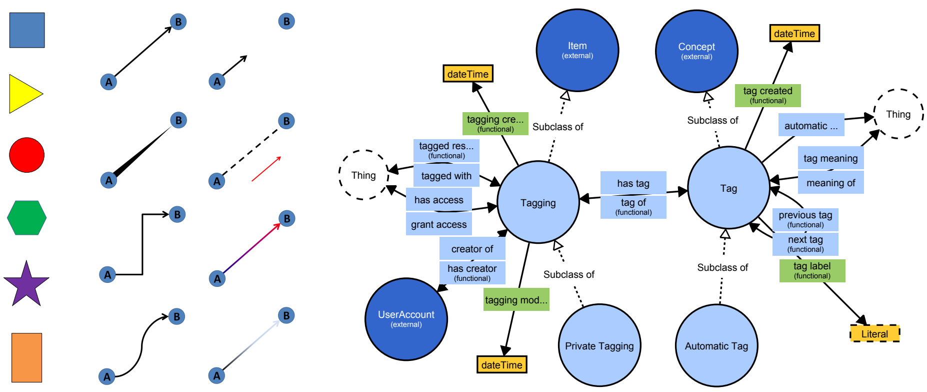
3 Visualization of Time-Varying Ontologies

Visualizing time-varying data is challenging due to the fact that users need to obtain an overview of a longer subsequence of individual time steps. This aspect must also be considered for dynamic, i.e., evolving ontologies. Usually, the users have to solve comparison tasks in order to reliably derive trends and countertrends or to find outliers and anomalies. In general, two paradigms for time-oriented visualizations are distinguished: 1) time-to-time mappings (animated diagrams), and 2) time-to-space mappings (static displays, often enhanced by interaction techniques). In the following, we discuss these two alternatives and address some of their benefits and drawbacks.

3.1 Representation of Vertices and Edges

If ontologies are visualized as graphs in the form of node-link diagrams, the vertices and edges can have various visual appearances. The vertices usually carry different semantic information as well as additional attributes. If the attributes are of a rather categorical nature, color coding and shape can be used as visual features to support the viewer to efficiently distinguish vertices of different types. Quantities may be visually indicated by varying the sizes of the graphical primitives (circles, triangles, rectangles and the like, see Figure 1a).

Using too many visual features at once, however, can make it troublesome to visually analyze the ontology for certain aspects. This can be seen as a conjunction search that does not allow for preattentive processing [19], i.e., elements have to be explicitly searched, which is more time-consuming. Consequently, our suggestion is to use as little visual features as possible—less is more in this case.



(a) (b) (c)

Fig. 1: Different representations for a) vertices and b) directed edges; c) node-link visualization of a small ontology.

Edges can be directed, weighted, attributed, and they can occur multiple times between the same pair of vertices, turning the ontology representation into a multigraph. Directions are typically indicated by arrowheads, color gradients, or with tapered representation styles. Weights can be expressed by color coding or by using differently thick link representations, which produce additional visual clutter on the negative side. Figure 1b illustrates several link representations for directed edges (namely, arrows, partial, tapered, animated, orthogonal, colored, curved, and dark-to-light links).

Figure 1c shows a node-link representation of the small MUTO ontology [30], using the VOWL notation [31]. Classes are represented by circles and property labels by rectangles. Datatypes are also depicted as rectangles but with a border and in a different color. Dashed and dotted lines indicate special types of classes and properties. The visual graph elements are not weighted in this example.

3.2 Topology, Structure, and Hierarchy

The topology of the graph representation is important, since it conveys useful information on the structure of subgraphs, clusters, or cliques. This aspect is also crucial in ontology visualization, as it can provide useful insights into the ontology structure.

In graph visualization (which is mainly applying node-link visual metaphors), layout algorithms are typically following aesthetic graph drawing criteria that produce diagrams with reduced visual clutter, which is "the state in which excess items, or their representation or organization, lead to a degradation of performance at some task" [37]. In the example of Figure 1c, a force-directed algorithm has been used to generate the initial layout, which was then manually optimized.

In many graph datasets, a hierarchical organization among the vertices is of special interest. Either it is inherently present in the data or it can be generated, for example, by a hierarchical clustering algorithm. Ontologies often contain concept hierarchies that have to be visualized in order to visually explore the ontology on different levels of hierarchical granularity or to use the hierarchy as a means to interact, filter, aggregate, and navigate in the ontology.

In Figure 1c, there are only two relatively flat hierarchies, each consisting of three classes, which can be easily spotted due to the small number of well arranged vertices. For ontologies with large inheritance trees, other graph layouts, or even tree visualizations, might be more appropriate to clearly depict the different hierarchy levels. However, since ontologies allow for multiple inheritance, simple tree visualizations can be confusing [14]. Also, most approaches have their limitations when it comes to the visualization of very large ontologies, at least when a node-link representation of the graph is used.

3.3 Visual Encoding of Time

The representation of time-varying ontologies demands for sophisticated visualization techniques that consider all the aforementioned features in order to sufficiently support the visual exploration of ontologies for dynamic patterns. Consequently, vertices, edges, the topology and structure, as well as any existing hierarchical organization among the vertices are of special interest.

One major type of tasks that users typically want to answer when inspecting time-varying data are comparison tasks. Several time steps are visually compared to derive insights by detecting changes or stabilities over time. To answer such tasks, the human visual system has to rely on its short term memory allowing to briefly remember visual patterns which are then compared at different spatial positions. Only by this internal cognitive process we are able to come up with the detection of trends, countertrends, or anomaly patterns over time [8].

If animated diagrams were used for the exploration of dynamic ontologies, we soon reach a point where the cognitive load becomes high. The human viewer may have difficulties to visually analyze the time-varying ontology for dynamic patterns. Advanced and time-complex layout algorithms have to be used to guarantee a high degree of dynamic stability [9], with the goal to preserve a viewer's mental map while inspecting the graph [3]. However, in the end, the detection of trends can be challenging anyway, even if the animation is replayed several times. Moreover, interaction techniques cannot be integrated in a traditional way, since the graphical elements are moving around in the worst case, which demands to stop the animation to meaningfully interact with the dynamic ontology.

Another option for displaying dynamic ontologies is by means of static diagrams [2] that map the time dimension to display space, which are known as time-to-space mappings. Such diagrams can, for example, use a vertex-aligned representation that allows to attach a hierarchical organization in a static way. Other than in ontology animation, the users can decide where to look at in the display in order to search for static or dynamic visual patterns. They can perform comparison tasks visually, not mentally as in animated diagrams that demand for higher cognitive efforts and are usually performing worse for time-oriented tasks.

To illustrate this second approach, we created different versions of the MUTO ontology [30] that was already shown in Figure 1c. We visualized these ontology versions with VOWL [31] in small display regions next to each other, which is known as a *small multiples* visualization [39]. If the vertices are roughly aligned and the visual features are not encoded differently over time (apart from changing variables), this approach leads to a good means to derive time-dependent patterns.

In Figure 2, we see the VOWL representations of six of the MUTO versions, starting with version 0.1 in the upper left and ending with version 1.0 in the lower right. We can observe how the ontology has changed over time. At first, the key concepts and links are defined, which are gradually extended by further concepts and links. At some point, alignments to existing ontologies are added, followed by the definition of datatype properties describing attributes for the key concepts. Finally, subclasses are introduced providing specializations of the key concepts.

3.4 Interaction Techniques

It must be noted that small multiples visualizations usually serve as an overview representation for the time dimension. If users detect a dynamic pattern of interest, they can apply interaction techniques to zoom and filter, and finally get details on demand also for individual ontology versions. The individual ontology

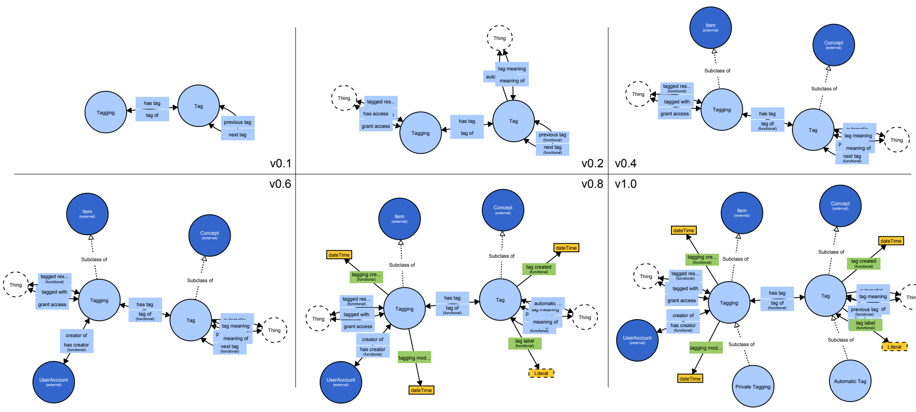


Fig. 2: Different versions of an ontology visualized as small multiples.

versions might be represented with existing ontology visualization techniques, as described in Section 2 and applied in Figure 2.

However, interacting with visual representations of time-varying data can be challenging, in particular, when the visualization is animated, i.e., when the single snapshots are changing over time. Users have to stop the animation when they like to interact with the visualization, otherwise a detected pattern might have already been disappeared until the users realize, for example, that they would like to select it for a more detailed exploration. Consequently, time-to-space mappings usually provide better means to visualize time-varying ontologies from the perspective of applying interaction techniques, since the ontology sequence is shown in a static fashion. Different ontology versions can be individually explored and visually connected by linking and brushing, even if this form of graphical representation usually does not scale to very long time sequences.

4 Conclusion and Future Work

In this position paper, we discussed the challenge of visualizing time-varying ontologies. We described benefits and drawbacks of common visualization techniques for dynamic graphs, since ontologies can be considered as a special type of graphs with additional attributes attached to the vertices and edges.

The visualization of ontologies has mainly been researched from the perspective of static graphs so far, which is – at least in our opinion – only half of the truth. Ontologies – as many other data structures – are typically not staying static but evolve over time. Consequently, the visualization of ontology evolution is a topic of its own and worth discussing.

For future work, we plan to apply dynamic graph visualization techniques to time-varying ontologies. Typically, we first need an overview representation which shows dynamic patterns in a static view, allowing the user to effectively and efficiently dig deeper and analyze the time-varying ontology.

References

1. H. Alani. TGVizTab: An ontology visualisation extension for Protégé. In *2nd Workshop on Visualizing Information in Knowledge Engineering*, 2003.
2. D. Archambault and H. C. Purchase. The mental map and memorability in dynamic graphs. In *IEEE Pacific Visualization Symposium*, pages 89–96, 2012.
3. D. Archambault and H. C. Purchase. The "map" in the mental map: Experimental results in dynamic graph drawing. *International Journal on Human-Computer Studies*, 71(11):1044–1055, 2013.
4. B. Bach, E. Pietriga, I. Liccardi, and G. Legostaev. OntoTrix: A hybrid visualization for populated ontologies. In *20th International Conference on World Wide Web, Companion Volume*, pages 177–180. ACM, 2011.
5. J. Bārzdiņš, G. Bārzdiņš, K. Čerāns, R. Liepiņš, and A. Sproģis. OWLGrEd: A UML style graphical notation and editor for OWL 2. In *International Workshop on OWL: Experiences and Directions*, CEUR-WS, vol. 614, 2010.
6. F. Beck, M. Burch, S. Diehl, and D. Weiskopf. The state of the art in visualizing dynamic graphs. In *EuroVis – STARs*, pages 83–103. EA, 2014.
7. A. Bosca, D. Bonino, and P. Pellegrino. OntoSphere: More than a 3D ontology visualization tool. In *2nd Italian Semantic Web Workshop*, CEUR-WS, vol. 166.
8. M. Burch and D. Weiskopf. Visualizing dynamic quantitative data in hierarchies – TimeEdgeTrees: Attaching dynamic weights to tree edges. In *International Conference on Information Visualization Theory and Applications*, pages 177–186, 2011.
9. S. Diehl and C. Görg. Graphs, they are changing. In *International Symposium on Graph Drawing*, pages 23–30, 2002.
10. M. Dudáš, O. Zamazal, and V. Svátek. Roadmapping and navigating in the ontology visualization landscape. In *19th International Conference on Knowledge Engineering and Knowledge Management*, pages 137–152. Springer, 2014.
11. P. Eklund, N. Roberts, and S. Green. OntoRama: Browsing RDF ontologies using a hyperbolic-style browser. In *1st International Symposium on Cyber Worlds*, pages 405–411. IEEE, 2002.
12. R. Falco, A. Gangemi, S. Peroni, D. Shotton, and F. Vitali. Modelling OWL ontologies with graffoo. In *ESWC 2014 Satellite Events*, pages 320–325. Springer, 2014.
13. S. Falconer. OntoGraf. http://protegewiki.stanford.edu/wiki/OntoGraf, 2010.
14. B. Fu, N. F. Noy, and M.-A. Storey. Indented tree or graph? A usability study of ontology visualization techniques in the context of class mapping evaluation. In *12th International Semantic Web Conference*, pages 117–134. Springer, 2013.
15. S. S. Guo and C. W. Chan. A tool for ontology visualization in 3D graphics: Onto3DViz. In *23rd Canadian Conference on Electrical and Computer Engineering*, pages 1–4. IEEE, 2010.
16. M. Hartung, A. Groí, and E. Rahm. COnto-Diff: Generation of complex evolution mappings for life science ontologies. *Journal of Biomedical Informatics*, 46(1):15–32, 2013.
17. M. Hartung, T. Kirsten, A. Groß, and E. Rahm. OnEX: Exploring changes in life science ontologies. *BMC Bioinformatics*, 10(250), 2009.
18. P. Hayes, T. C. Eskridge, R. Saavedra, T. Reichherzer, M. Mehrotra, and D. Bobrovnikoff. Collaborative knowledge capture in ontologies. In *International Conference on Knowledge Capture*, pages 99–106. ACM, 2005.
19. C. Healey and J. Enns. Attention and visual memory in visualization and computer graphics. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1170–1188, 2012.

20. J. Heflin and J. A. Hendler. Dynamic ontologies on the web. In *17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, pages 443–449. AAAI, 2000.
21. D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.
22. W. Hop, S. de Ridder, F. Frasincar, and F. Hogenboom. Using hierarchical edge bundles to visualize complex ontologies in GLOW. In *27th Annual ACM Symposium on Applied Computing*, pages 304–311. ACM, 2012.
23. M. Horridge. OWLViz. http://protegewiki.stanford.edu/wiki/OWLViz, 2010.
24. J. Howse, G. Stapleton, K. Taylor, and P. Chapman. Visualizing ontologies: A case study. In *10th International Semantic Web Conference*, pages 257–272. Springer, 2011.
25. A. Hussain, K. Latif, A. Rextin, A. Hayat, and M. Alam. Scalable visualization of semantic nets using power-law graphs. *Applied Mathematics & Information Sciences*, 8(1):355–367, 2014.
26. A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology visualization methods – a survey. *ACM Computer Surveys*, 39(4), 2007.
27. M. C. A. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov. Ontology versioning and change detection on the web. In *13th International Conference on Knowledge Engineering and Knowledge Management*, pages 197–212. Springer, 2002.
28. M. Lanzenberger, J. Sampson, and M. Rester. Visualization in ontology tools. In *International Conference on Complex, Intelligent and Software Intensive Systems*, pages 705–711. IEEE, 2009.
29. T. Liebig and O. Noppens. OntoTrack: A semantic approach for ontology authoring. *Web Semantics*, 3(2-3):116–131, 2005.
30. S. Lohmann, P. Díaz, and I. Aedo. MUTO: The modular unified tagging ontology. In *7th International Conference on Semantic Systems (I-SEMANTICS '11)*, pages 95–104. ACM, 2011.
31. S. Lohmann, S. Negru, F. Haag, and T. Ertl. Visualizing ontologies with VOWL. *Semantic Web Journal*, to appear.
32. J. D. Novak and A. J. Cañas. The theory underlying concept maps and how to construct them. Technical Report IHMC CmapTools 2006-01, Florida Institute for Human and Machine Cognition, 2006.
33. N. F. Noy and M. Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems*, 6(4):428–440, 2004.
34. N. F. Noy and M. A. Musen. PromptDiff: A fixed-point algorithm for comparing ontology versions. In *18th National Conference on Artificial Intelligence*, pages 744–750. AAAI, 2002.
35. D. Perrin. PROMPT-Viz: ontology version comparison visualizations with treemaps. Master thesis, University of Victoria, 2004.
36. P. Plessers, O. De Troyer, and S. Casteleyn. Understanding ontology evolution: A change detection approach. *Web Semantics*, 5(1):39–49, 2007.
37. R. Rosenholtz, Y. Li, J. Mansfield, and Z. Jin. Feature congestion: A measure of display clutter. In *Conference on Human Factors in Computing Systems*, pages 761–770, 2005.
38. M. Sintek. OntoViz. http://protegewiki.stanford.edu/wiki/OntoViz, 2007.
39. E. R. Tufte. *The visual display of quantitative information.* Graphics Press, 1992.
40. T. D. Wang and B. Parsia. CropCircles: Topology sensitive visualization of OWL class hierarchies. In *5th International Semantic Web Conference*, pages 695–708. Springer, 2006.