On the Visualization of Large-sized Ontologies

Yannis Tzitzikas
University of Crete, and
FORTH-ICS, Greece
tzitzik@ics.forth.gr

Jean-Luc Hainaut
Institut d'Informatique
University of Namur (F.U.N.D.P.), Belgium
jlh@info.fundp.ac.be

ABSTRACT

The visualization of ontologies and metadata is a challenging issue with several applications not only in the Semantic Web but also in Software Engineering, Database Design and Artificial Intelligence. This paper aims at identifying and analyzing the more principal aspects of this problem, surveying some of the work that has been done so far, and at proposing novel ideas that are worth further research and investigation. In particular, it describes the main factors that determine whether an ontology diagram layout is satisfying or not and focuses on the visualization requirements of large-sized ontology diagrams.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

1. INTRODUCTION

The emerging Semantic Web already comprises hundreds of ontologies. To support tasks like ontology selection, specialization and instantiation (i.e. resource annotation), we need methods and tools to support both designers and users in finding the desired ontology, in understanding its structure, and in using it. Hopefully, the Semantic Web trend has rekindled the interest in systems for conceptual modeling and quite a lot of tools have emerged recently that offer visualization for various kinds of conceptual schemas (XML, RDF, OWL), including Spectable (for taxonomy-based sources) Jamabalaya, KAON OI Modeler (for RDF/S), ONTORAMA (RDFS), EROS (RDFS) and also [19] (a short comparative study of these systems can be found in [13]). A common practice not only in the Semantic Web, but also in database engineering and information systems analysis and design, is to adopt *diagrams* in order to present in a natural and readable way the contents and the structure of an ontology, of a database schema (e.g. ER diagrams), or of any kind of system design (e.g. UML). It follows that the visual presentation of these diagrams is essential. The 2D or 3D layout of the diagram must comply with the way users use to perceive the concepts. For instance, an element should be placed close to the major element it depends on or it is a part of, and the hierarchical nature of a set of concepts (e.g. a taxonomic hierarchy) should be suggested by a top-down arrangement of their representation.

Many algorithms have been proposed in the last two decades to produce automatic layout of conceptual diagrams. General algorithms, though powerful, have proved very poor but for very specific schemas, such as those which are purely or nearly hierarchical. It is a widely accepted opinion that the automatic layout facilities offered by current UML-based CASE tools are not satisfactory even for very small diagrams (for more see [5]). Consequently, the vast majority of layouts created today are done "by hand"; a human designer makes most, if not all, of the decisions about the position of the objects to be presented [11]. Concerning the analysis of the problem itself, i.e. the basic and common aesthetic principles for conceptual graph drawing, there are only a few and limited empirical studies on user preferences for conceptual diagrams. For instance, [14] discusses empirically proved user preferences for UML graphs (of small size), while [5] gives a list of aesthetic suggestions for the same kind of graphs. Of course, specialized algorithms for drawing conceptual graphs have also been developed long ago (e.g. see [18]) and recently (e.g. see [4] for UML state charts). The visualization of large conceptual diagrams is even less explored. The classical hierarchical decomposition techniques that are used for visualizing large plain graphs (for a survey see Chapter 3 of [15]), have not been applied or tested on conceptual diagrams. In any case, the labels of the nodes that correspond to clusters (or to clusters of clusters, and so on), which are quite important for understanding a conceptual diagram, seems to be a spiny issue. Consequently, only manual collapsing mechanisms (like those described in [7]) are currently available for decreasing the visual clutter and for aiding the understanding of large conceptual graphs.

Diagram drawing is not a panacea. It has been recognized long ago that the usefulness of conceptual diagrams degrades rapidly as they grow in size. As the Semantic Web is based on ontologies potentially including dozens of thousands of classes, new methods and techniques should be devised in order to tackle the problem of understanding and visualizing large diagrams. In this paper we focus on the visualization of the structural part of ontologies.

AVI '06, May 23–26, 2006, Venezia, Italy.

2. WHAT MAKES A DIAGRAM DRAWING A GOOD DRAWING?

Experience shows that the best layout of a diagram rests on three sets of rules:
(I) *Graph-based* drawing rules
These rules rely on general *graph theory* and apply on any kind of graphs (e.g. see [2]). The objective of these algorithms is to make a graph as readable as possible. This includes minimizing the edge crossings, reducing the space occupied by the graph, placing the nodes in an orthogonal grid, displaying symmetries if there is any of them, and satisfying other (in many cases subjective) aesthetic criteria.
(II) *Semantic-based* drawing rules
These rules take into account the semantics of the specific constructs of the data model, like *ISA* relations, *part-of* associations and *one-to-many* associations (e.g. see [18]). They lead to *heuristics* (such as those mentioned in the introductory section) and yield better results for small to medium size diagrams. These rules have triggered much lower research interest. Another rising issue concerns the visualization of the inferred knowledge (e.g. see [10]).
(III) *Domain-specific* drawing rules
This set of rules are *domain specific* and cannot translate easily into formal rules. For instance, though each `OrderLine` entity equally depends on an `Order` entity and a `Product` entity, experience has shown that users perceive it as being closer to `Order` than to `Product`. This domain still is largely unexplored.

In general, the above rules are in many cases conflicting and no single category, by itself, can provide satisfying layout: for large diagrams the first two lead to unnatural and most often unreadable layouts, while the third one, being manual, is impractical. One important remark is that we could view all kinds of rules as (a) *constraints* that the desired layout should satisfy, and (b) *objectives* that the desired layout should maximize. Consequently, graph drawing problems could be formalized as constraint satisfaction problems and/or multi-objective optimization problems. Unfortunately, most of these optimization problems are computationally hard and constraint satisfaction is not a light task either. In addition, finding a language appropriate for expressing this kind of constraints is by itself a challenging and not trivial task.

2.1 Requirements for Understanding and Visualizing Large Diagrams

Suppose that one gives us a very large diagram (illustrating an ontology or the database schema of an existing information system) and asks us to understand this diagram without telling us anything about the application domain of the information system. Two basic observations are in order:
(α) The *labels* of the nodes and edges is the basic means for understanding the graph. Even a small conceptual diagram but with no labels or with labels in Chinese is useless (at least for the authors of this paper).
(β) The main problem for understanding the entire diagram is the *magnitude* of the graph and not its layout. A graph of 100.000 nodes even if it is perfectly positioned, cannot be understood easily (or at all).

The first observation (α) suggests that we must exploit as much as we can the *labels* of the conceptual diagram, so methods originated or inspired by the area of *Information Retrieval*, or *Web searching* [3, 9], could be applied and could prove useful in practice. The second observation (β) suggests that we need methods for generating *focused*, *summarized* and *abstract* views of the graph. Below we discuss three kinds of such methods, namely *context-based browsing*, *filtering*, and *clustering*.

Context-based browsing
Instead of displaying the entire diagram the user could start browsing the diagram starting from any node of the diagram. At each point in time, the neighborhood of the *focal* (selected) node is displayed, e.g. as a *star* graph. The user is then able to click on any other displayed node in order to change the focus. In this way, the visualization, and thus the understanding of the conceptual diagram, is done gradually. As the displayed subgraph is not too big we can easily generate an aesthetically good layout (of course, the bigger the radius of the neighborhood is, the more difficult this problem becomes). The only limitation of this approach is that the understanding of the conceptual diagram is done with the assistance of a computer, i.e. we cannot do it on paper. However, we could also print them on paper using a technique analogous to the way that big roadway maps are printed on books: at every boundary edge of each page we place the number of the page that shows the next part of the map in the corresponding direction (north, south, east, west). As in the case of conceptual diagrams there is no reference system of coordinates, we could use more than 4 pointers at each page and with an appropriate algorithm we could probably print the entire map without having to print one page for every node, but with much less number of pages. Figure 1 sketches graphically this idea. The graph on the left has six nodes and of course we could print it by printing six pages each one displaying the star-view of one node. However, we could also use only three pages as it is illustrated at the right part of Figure 1. This figure shows a set of three star-views of radius 1 that satisfy the following properties: (a) for each node of the graph there is one page that displays all of its adjacent edges, and (b) all edges between any pair of nodes that appear in the same page are displayed.

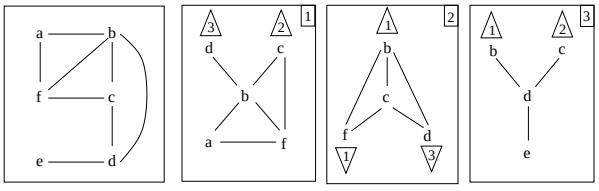

Figure 1: Partitioning a graph of 6 nodes into three star-view like graphs of radius 1

Filtering
Another method to reduce the number of displayed elements is to filter out some of them according to a number of criteria. Some of them are discussed below.
Property-based criteria
Here the filtering condition is evaluated upon a property (e.g. the label) of the graph elements. For example, hide/show only those nodes or edges whose label contains (or not) the

substring "company".

Local Connectivity criteria

Here the filtering condition is evaluated upon a connectivity-based property of the graph elements. For example, show/hide those nodes whose fan-out/fan-in is lower/greater than a given number (e.g. 2).

Global Connectivity criteria

Here the appearance/occultation of some elements is based on connectivity properties whose evaluation requires considering several nodes and edges of the graph. For example, show/hide all elements which are connected (directly or directly) with a selected set of graph elements (or show the subgraph of all nodes that can be reached within at most k edges from the focal element). Or, show only those nodes and edges that are related with *isA* relationships with the focus node.

Complex global connectivity criteria

The ontology query language (e.g. RQL[8] or OWL Lite[1]) can be used in order to specify the desired selection or filtration. However this is not possible if the ontology is unknown to the designer and the user. This means that queries and views (e.g. RVL[12]) could be used only by designers for providing users with various views of an ontology. The existence of a metamodel (this will be discussed later) would allow exploiting the query language in a more generic manner. For example, we could have condition/projection rules that are expressed in terms of the metamodel and thus applicable to any ontology (under that metamodel).

Sophisticated connectivity criteria (for large diagrams)

It is worthy to investigate whether the techniques that are applied for Web-searching could be of useful for the problem at hand. As the Web is very big, the *link structure* is exploited in order to deduce the pages that contain valuable information. Actually this is the key difference between Web-retrieval and the classical Information Retrieval. The number of hyperlinks that point to a page provides a measure of its popularity and quality. Also, many links in common between pages or pages referenced by the same page, often indicate a relationship between those pages. This kind of information is exploited in order to identify the pages that have valuable information and thus should be ranked high. A PageRank-like scoring scheme for "taming" (abstracting) very large ER diagrams is presented in [21].

Formal ontological criteria (or metamodel-based criteria)

Although there isn't any broadly accepted formal ontology (or philosophical theory in general), it seems that there are some notions that show up in one form or another in every application domain, including: *Categories* (e.g. `apples, tomatoes`), *Measures* (e.g. `mass, age, price`), *Composite objects*, (e.g. `cars` have `engines`, `soccer` teams have 11 `players`), *Time, Space* and *Change*, *Evens* and *Processes* (e.g. `raining`), *Physical objects*, *Substances* (e.g. `tomato juice, water`), *Mental objects* and *Beliefs*. If ontologies were based (or at least annotated) with respect to a formal ontology, e.g. like the one shown in Figure 2 (reproduced from [16]), then we could exploit this ontological information in several ways. For example, we could show/hide some parts of the graph (e.g. show only events, or hide all measures, etc). Moreover, special viewing techniques could aid comprehensibility (e.g. every ontological type could be associated with a different color or symbol). Unfortunately, current conceptual schemas and ontologies are not based on such an ontological analysis. Moreover, from an existing schema we cannot infer automatically the conceptualization (the intellectual lens) through which the designers of the schema viewed the domain and designed the schema. However, the increasing interest that we observe the latest years around formal ontology (e.g. see OntoClean[2]) may affect the form of the conceptual graphs of the next decade. Another associated remark is that formal ontological information could be expressed as meta-schema information, i.e. the schema could be annotated according to a formal ontological meta-model. For instance, as the semantic model adopted by the SIS system[3], namely SIS-Telos, supports unlimited number of instantiation levels (token level, class level, meta-class level, meta-meta-class level, and so on), the viewer of SIS exploits the meta-schema in order to offer readable drawings of big parts of the schema. Specifically, the appearance of nodes and edges in these drawings is based on criteria whose evaluation is based on the classification of the schema to the meta-schema. Apart from this, the browsing the meta-schema itself would aid understanding the domain of a conceptual graph, e.g. whether the schema models the *static*, *dynamic*, *intentional*, or *social* aspects of a domain. Although UML has a metamodel (see MOF of OMG), this is not useful for our purposes (it does not carry any ontological information).

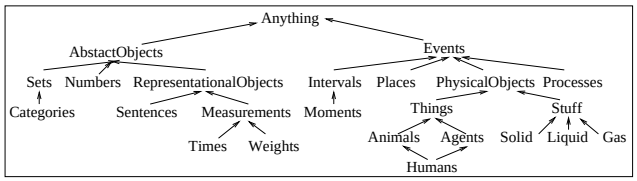

Figure 2: An example top-level ontology of the world

External criteria

For instance, if the ontology (or schema) is populated with data (e.g. annotations or tuples in case we have a relational schema), then we can exploit the data layer in multiple ways. For instance, we can filter out or emphasize the nodes of the graph on the basis of the number of tuples that are associated with them. User feedback obtained during the interaction with the system could also be exploited, in the context of a CAOE (Computer Aided Ontology Engineering) tool.

Clustering

Another method to reduce the number of displayed elements is to form *groups* (or *clusters*) of graph elements, to *collapse* each group into one node, and finally, to display only these composite nodes. This not only reduces the number of displayed nodes but also (and more importantly) the number of displayed edges. Notice that in many cases it is the multitude of edges that makes a graph look messy and unreadable. Clustering could be performed either manually, semiautomatically or automatically. ER clustering is an example. However, most of the proposals for clustering either

[1]http://www.w3.org/TR/owl-features

[2]http://ontolog.cim3.net/cgi-bin/wiki.pl?OntoClean

[3]http://www.ics.forth.gr/isl/r-d-activities/sis.html

require human input [20], or they haven't been tested on large conceptual schemas [1]. Recent work on ontologies includes [17] and [6]. In the context of a CAOE tool, the user could use a rectangle to encapsulate the desired graph elements and then ask from the viewer to collapse them into one node. Instead of specifying explicitly the elements of the group the user could specify the desired subgraph by using a number of declaratively (or even procedurally) specified criteria. Roughly, every criterion that could be used for filtering (e.g. any label or connectivity criterion) could be also used for clustering.

3. CONCLUDING REMARKS

Conceptual diagram drawing is not a one-shot task. Therefore CAOE tools should support a semiautomatic methodology for ontology visualization and drawing. Force-directed placement algorithms are well suited for this scenario and they also allow incorporating external knowledge, e.g. domain specific rules expressed in the form of a force. Global complexity reduction operations are very important for understanding big ontologies and for this reason ranking, clustering and collapsing should be supported. We believe that future CAOE should support all the above functions. Also note that currently ontologies are exchanged in a layout-missing format (e.g. RDF). So there is an extra need for standards for the exchange of visualization information across different CAOE tools. These standards should not only allow exchanging information (e.g. coordinates) but also force models (with their configuration parameters) and other external information that could be exploited for visualization.

4. REFERENCES

[1] J. Akoka and I. Comyn-Wattiau. "Entity-Relationship and Object-Oriented Model Automatic Clustering". *Data and Knowledge Engineering*, 20(2):87–117, 1996.

[2] G. D. Battista, P. Eades, R. Tamassia, and I. Tollis. *"Graph drawing: algorithms for the visualization of graphs"*. Prentice Hall Englewood Cliffs (N.J.), 1999. ISBN/ISSN : 0-13-301615-3.

[3] S. Brin and L. Page. "The Anatomy of a Large-scale Hypertextual Web Search Engine". In *Procs of the 7th Intern. WWW Conf.*, Brisbane, Australia, April 1998.

[4] R. Castello, R. Mili, and I. Tollis. "A Framework for the Static and Interactive Visualization of Statecharts". *Journal of Graph Algorithms and Applications*, 6(3):313–351, 2002.

[5] H. Eichelberger and J. W. von Gudenberg. "UML Class Diagrams - State of the Art in Layout Techniques". In *Procs of Vissoft 2003, Intern. Workshop on Visualizing Software for Understanding and Analysis*, pages 30–34, 2003.

[6] B. C. Grau, B. Parsia, E. Sirin, and A. Kalyanpur. "Automatic Partitioning of OWL Ontologies Using E-Connections". In *Procs of the 2005 Intern. Workshop on Description Logics, DL'05*, 2005.

[7] J. Huotari, K. Lyytinen, and M. Niemela. "Improving Graphical Information System Model Use with Elision and Connecting Lines". *ACM Transactions on Computer-Human Interaction*, 10(4), 2003.

[8] G. Karvounarakis, V. Christophides, and D. Plexousakis. "RQL: A Declarative Query Language for RDF. In *Eleventh International World Wide Web Conference (WWW)*, Hawaii, USA, May 2002.

[9] J. Kleinberg. "Authoritative Sources in a Hyperlinked Environment". In *Procs of 9th ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, USA, 1998.

[10] T. Liebig and O. Noppens. "OntoTrack: A Semantic Approach for Ontology Authoring". *Journal of Web Semantics*, 3(2-3):116–131, 2005.

[11] S. Lok and S. Feiner. "A Survey of Automated Layout Techniques for Information Presentations". In *Procs of the 1st. Int. Symp. on Smart Graphics*, Hawthorne, NY, 2001.

[12] A. Magkanaraki, V. Tannen, V. Christophides, and D. Plexousakis. "Viewing the semantic web through RVL lenses". *Journal on Web Semantics*, 1(4):359–375, 2004.

[13] A. Ouwerkerk and H. Stuckenschmidt. "Visualizing RDF Data for P2P Information Sharing". In *Procs of the workshop on Visualizing Information in Knowledge Engineering, VIKE'03*, Sanibel Island, FL, 2003.

[14] H. C. Purchase, J.-A. Allder, and D. Carrington. "Graph Layout Aesthetics in UML Diagrams: User Preferences". *Journal of Graph Algorithms and Applications*, 6(3):255–279, 2002.

[15] A. J. Quigley. *"Large Scale Relational Information Visualization, Clustering, and Abstraction"*. PhD thesis, University of Newcastle, Australia, August 2001. (http://www.it.usyd.edu.au/∼aquigley/thesis/aquigley-thesis-mar-02.pdf).

[16] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995.

[17] H. Stuckenschmidt and M. Klein. "Structure-Based Partitioning of Large Concept Hierarchies". In *Procs of the 3rd Intern. Semantic Web Conference ISWC'2004*, Hiroshima, Japan, 2004.

[18] R. Tamassia, C. Batini, and M. Talamo. "An algorithm for automatic layout of entity-relationship diagrams". In *Procs of the 3rd Intern. Conf. on Entity-relationship approach to software engineering*, pages 421–439. Elsevier North-Holland, Inc., 1983.

[19] A. C. Telea. "Visualizing RDF(S)-based Information". In *Procs of IV'2003*, Hawthorne, NY, 2003.

[20] T. J. Teory, W. Guangping, D. L. Bolton, and J. A. Koenig. "ER Model Clustering as an Aid for User Communication and Documentation in Database Design". *Communications of the ACM*, 32(8):975–987, 1989.

[21] Y. Tzitzikas and J.-L. Hainaut. "How to Tame a Very Large ER Diagram (using Link Analysis and Force-Directed Placement Algorithms)". In *Proceedings of 24th Int. Conf. on Conceptual Modeling, ER'2005*, Klagenfurt, Austria, October 2005.