OntoSphere3D: a multidimensional visualization tool for ontologies

Alessio Bosca, Dario Bonino
Politecnico di Torino, Torino, Italy
{alessio.bosca, dario.bonino}@polito.it

Introduction

Visualizing knowledge in two dimensions is a challenging task, since many dimensions are involved: instances, concepts, hierarchical relations just for naming a few. And it is also a well studied paradigm that already produced good solutions such as: Jambalaya [1, 2], GraphViz [3], OntoViz [4], etc. Nevertheless, mapping the many dimensions involved by an ontology, on only two dimensions can sometimes be too restrictive, especially in the case of very large and complex knowledge domains.

In this presentation, we propose a novel approach for inspecting and editing ontologies using a visually enriched 3-dimensional space. Ontology information is represented on a 3D view-port merging structural information (i.e. concepts and relations) with context information (the amplitude of *Is-A* hierarchies rooted in a given ontology concept,…) by means of visual cues. Being the 3-dimensional view quite natural for humans, especially for what concerns navigation, the proposed approach aims at being more effective in browsing ontological data than the currently available 2-dimensional solutions. Improvements are obtained by involving direct manipulation operations such as zooming, rotating, and translating objects, and by introducing more dimensions to convey information on the visualized knowledge model as the color or the size of visualized entities.

The proposed work aims also at tackling space allocation issues for ontology visual models, in fact, in the traditional solutions, big ontologies can easily lead to overcrowded representations that are difficult to browse and that can be more confusing than aiding. Some attempts exist to overcome these problems, as in OntoRama [5,6], where the nodes being inspected are magnified with respect to the other nodes in the ontology. However, even these approaches tend to collapse when visualizing big ontologies such as SUMO [7], counting over than 5'000 concepts. The proposed application, instead adopts a dynamic collapsing mechanism and different views, at different granularities, for granting a constant navigability of the rendered model.

Proposed Approach

A three-dimensional environment is the starting point of the proposed ontology visualization tool, as a 3D space offers one more dimension than traditional 2D approaches to represent ontology data, so simplifying its interpretation. In addition many more dimensions are added to improve completeness and readability of the representation. Two main principles guide the visualization effort: increasing the number of "dimensions" (colors, shapes, transparency, etc.) which represent concepts features and convey additional information without adding the burden of further graphical elements (such as labels) on the scene, and automatically identifying the part of knowledge to be displayed and the detail level to be used in the process, on the base of user interaction with the scene. The latter principle is particularly important for improving the overall system performances since scale factor indeed constitutes a strong issue in visualizing complex graph structures like ontologies. As the cardinality of elements increases, the number of items to be concurrently displayed on the screen worsens the graphical perception of the scene and complicates spotting details. When the amount of visualization space needed to represent all the information within the KB outnumbers the space available on the screen, a few options remain available: to scale down the whole image to the detriment of readability, to present on the screen just a portion of it and allow its navigation or to summarize the information in a condensed graph and provide means for exploration and expansion. As the effectiveness of these options depends on the use case involved (consistency checking, domain comprehension, KB updates) a combined usage of them offers a suitable approach.

To combine seamlessly the options above, taking advantage of their strength points whenever possible, the proposed solution exploits different scenes that present and organize the information on the screen according to differently detailed perspectives. Such scenes interchange in managing the graphical space as user attention shifts from one concept to another, by implicitly inferring the focus from user's interaction with the scene (e.g., a concept selection with a mouse click). In this way, the idea of "focusing" as the application capability of highlighting the elements of interest while leaving out the others, is applied.

The OntoSphere3D [8] user interface is rather minimalist and allows direct manipulation of scenes through rotation, panning and zoom; it allows to browse the ontology as well as to update it and to add new concepts and relations (taking advantage of functionalities provided by the Protégé framework in which is deployed). Every concept within a given scene is clickable with two different results: a single left-click maintains the current perspective and simply navigates through elements, while a double-click leads to a focusing operation, shifting the scene to a more detailed level. Right-clicking on a concept that has direct instances visualizes them, while the same action applied to an instance bring the focus back on the related concept (see section 3.3) .

Root Focus Scene

This perspective presents a big "earth-like" sphere bearing on its surface a collection of concepts represented as small spheres (Figure 1). The scene does not visualize any taxonomic information and only shows direct "semantic" relationships between elements of the scene, usually a graph not fully connected. Atomic nodes, the ones without any subclass, are smaller and depicted in blue while the others are white and their size is proportional to the number of elements contained in their own sub-tree.
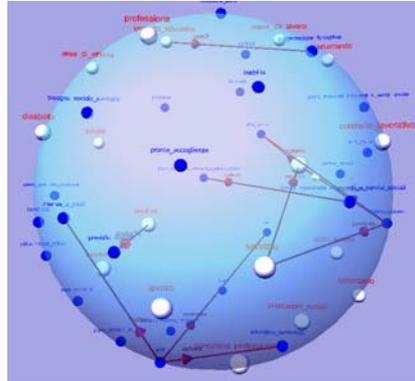
Figure 1. The Root Focus scene.

This view is particularly intended for representing the primitive concepts (i.e., the roots), but can also be used, during the ontology navigation, to visualize direct children of a given node; a pretty useful option in case of heavily sub-classed concepts. Representing primitive concepts within the ontology, and the relations between them, allows to easily identify the conceptual boundaries of the represented domain and provides a very good hint to the question: "what's the ontology about?"

Tree Focus Scene

This scene shows the sub-tree originating from a concept; it displays the Is-A hierarchy as well as other semantic relations between represented classes. Since experimental evidence proves that too many elements on the screen, at the same time, hinder user attention, the scene completely presents only three fully-expanded levels at a time. As the user browses the tree, the system automatically performs expansion and collapse operations in order to maintain a reasonable scene complexity (Figure 2). Collapsed elements are coloured in white and their size is proportional to the number of elements present in their sub-tree; instead concepts located at the same depth level within the tree have the same colour in order to easily spot groups of siblings. *Is-A* relations are displayed with a neutral colour (grey), without labels, whereas other semantic relations involving concepts already in the scene are displayed in red, and are accompanied by the name of the relation. When an element of the scene is related to a node that is not present on the view-port, a small sphere is added for the hidden node in the proximity of the given element, so terminating the end of the arrow; in such cases, incoming relations are represented with a green arrow, while outgoing links with a red one.
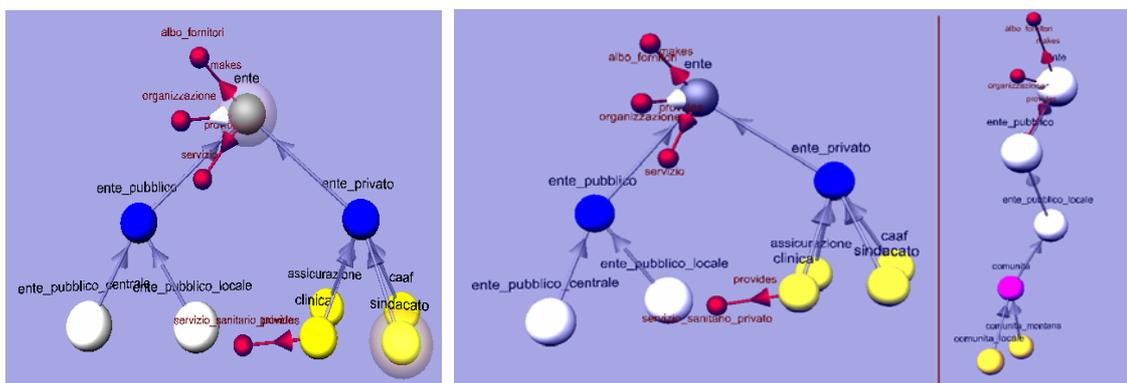
Figure 2. The Tree Focus scene.

Concept Focus Scene

In the concept focus scene, all the available information about a single concept is provided, at the highest possible level of detail. The concept's children and parent(s) are therefore shown as well as its ancestor root(s) and its semantic relations, including the inherited ones. Semantic relations are drawn as arrows terminating in a small sphere: red if the relation is outgoing and green otherwise. Direct relations are drawn close to the concept, with an opaque color, while inherited ones are located a bit farther from the center and depicted with a fairly transparent color.

This scene can be extremely useful during consistency checking operations because it eases the spotting of inconsistent concepts or relations, e.g. whenever a concept inherits from an ancestor a property that "logically" contrasts with other features of its own

Instance Focus Scene

Whenever a concept has direct instances (in the tree focus scene or in the concept focus scene) its sphere is depicted surrounded by a transparent sphere resembling a sort of a shell (Figure 3). By right-clicking the concept, its direct instances are shown, using the same representation paradigm adopted by the concept focus scene. The resulting view represents instances and their properties as inter-connected cubes.
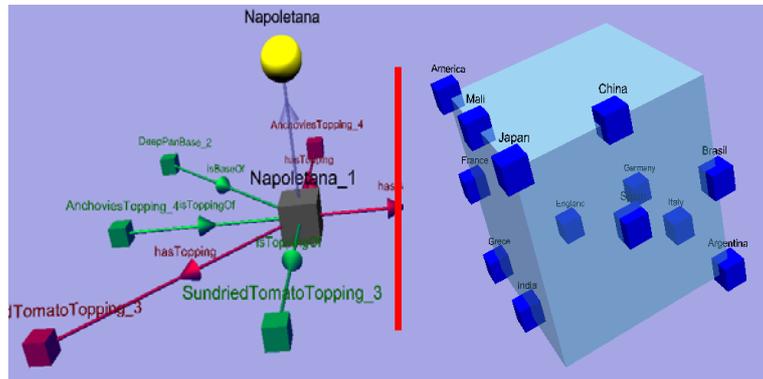
Figure 3 Instance Focus Scene

Results

In order to confirm the initial claim of the proposed work (i.e. improved navigability and inspection capabilities with respect to 2-dimensional approaches) the authors set up an efficiency comparison between 4 different visualization tools: the proposed OntoSphere3D plug-in, the Jambalaya plug-in, the OWLViz plug-in and the TgViz plug-in. These plug-ins have been tested against a predefined set of ontology related operations, namely: visualization of the top concepts, visualization of the relations between the top concepts, visualization of concepts located at level *n* in the *Is-A* hierarchy of the ontology, visualization of the concepts related to a given one, visualization of relations between concepts at the same hierarchy level, navigation of the ontology from one concept to another, search for a given concept.

The required Protégé-related skills have also been taken into account in the evaluation. Each operation has been assigned a predefined difficulty score, as reported in Table 1. Evaluation results are, instead, reported in Table 2.

Table 1 Difficulty scores for several user interactions.

User interaction	Difficulty score
Mouse click	2
Mouse double-click	2
Look at the screen	0
Mouse over	1
Mouse scroll	3
Search (filling a form)	4

Table 2. Results of efficiency evaluation on 4 different visualization plug-ins.

Operation	Jamb.ya	On.3D	OWLViz	TViz
User Experience	1	1	2	2
Top concepts	0	0	3	6
Relations between top concepts	1	0	isA = 0 not-isA = ∞	1
Level–n concepts	n	2n	n.d.	n.d.
Related concepts	4	0	isA = 0 not-isA= 7	0
Relations between concepts at the same level	1	0	isA=0 not-isA=7	1
Concept navigation	2	2	0-3	0-3
Search	10	6	4	6
Jamb.ya = Jambalaya, On.3D = OntoSphere3D, TViz=TgViz				

It is easy to notice that, in most cases the proposed approach outperforms the other applications, except for visualizing concepts at a given level n in the ontology hierarchy, for which n mouse clicks are required, and for searching concepts, where the offered functionality is the one of the Protégé framework as for TgViz. In concept navigation however, data is quite difficult to compare since both Jambalaya, OWLViz and TgViz require scrolling for navigating between ontology concepts. According to the evaluation grid in Table 1, this is not a too heavy task but, when the ontology size grows up from few tens of concepts to several thousands, the required scrolling may become much more cumbersome and thus shall probably be re-weighted. On the contrary, the OntoSphere3D behavior is size-independent, becoming more suitable on really big ontologies such as SUMO.

References

[1] M.A. Storey, M. Musen, J. Silva, C. Best, N. Ernst, R. Fergerson, and N. Noy. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé. In Workshop on Interactive Tools for Knowledge Capture, Victoria, B.C. Canada, October 2001.

[2] Jambalaya. http://www.thechiselgroup.org/chisel/projects/jambalaya/jambalaya.html.

[3] Gansner, E. R. & North, S. C. (1999), An open graph visualization system and its applications to software engineering, Software Practice and Experience 30(11),.

[4] OntoViz Tab: Visualizing Protégé Ontologies. http://protege.stanford.edu/plugins/ontoviz/ontoviz.html.

[5] P.W.Eklund, N.Roberts, S.P.Green, OntoRama: Browsing an RDF Ontology using a Hyperbolic-like Browser, *The First International Symposium on CyberWorlds (CW2002)*, pp.405-411, Theory and Practices, IEEE press, 2002

[6] OntoRama. http://www.ontorama.com/

[7] Niles, I., and Pease, A. 2001. Towards a Standard Upper Ontology. In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Chris Welty and Barry Smith, eds, Ogunquit, Maine, October 17-19, 2001.

[8] http://sourceforge.net/projects/ontosphere3D