

# Ontology Visualization: Approaches and Software Tools for Visual Representation of Large Ontologies in Learning

Anton Anikin<sup>1</sup>, Dmitry Litovkin<sup>1</sup>, Marina Kultsova<sup>1</sup>(✉), Elena Sarkisova<sup>1</sup>, and Tatyana Petrova<sup>2</sup>

<sup>1</sup> Volgograd State Technical University, Volgograd, Russia

`Anton@Anikin.name, marina.kultsova@mail.ru`

<sup>2</sup> Volgograd State Socio-Pedagogical University, Volgograd, Russia

`poas@vstu.ru`

**Abstract.** In this paper, we address the issue of large ontologies visualization for learning. The ontologies can be used to improve the efficiency of the learning when the learner explores a new subject domain and needs its conceptual model. The ontologies can help to overview this new subject domain, to better understand current knowledge, the knowledge that should be got, the subject domain structure, main concepts, relationships between them and relevant information resources. However with increasing the complexity of domain structure, the number of concepts and relationships, it becomes more difficult to overview and understand conceptual model represented with the ontology. We review the approaches to ontology visualization and their implementation in the software tools that can help to resolve this issue.

**Keywords:** Semantic web · Ontology · Ontology visualization

## 1 Introduction

An ontology is a convenient tool of domain modelling for wide range of tasks, such as [2, 3, 18, 19]:

- Exploring a new subject domain by studying its ontology-based model.
- Studying a new subject domain via ontology-based domain modeling.
- Linking the ontology concepts with relevant information resources to provide automated information retrieval during the studying a new subject domain.
- Support of inference rules allows using ontologies not only for automated information retrieval but also for learning management.

Thus, the use of ontologies in learning is well grounded and reasonable, both for studying a new subject domain by individual users, as well as within the educational process in universities to manage single courses and learning process as a whole.

Within ontological modeling, the developer faces a number of problems concerning the creation, edit, distribution and sharing ontologies, and their adaptive visual representation (visualization).

The last task is topical and widely applicable due to the large scale and complexity of the domain ontologies structure. Ontologies can contain hundreds and thousands of concepts associated with different types of relationships, as well as links to the ontologies of the related domains. These makes it difficult for the user to perceive information, and so the domain ontology visualization becomes the very important task that includes adaptive visual representation of ontologies in learning, ontology navigation and search of elements within the ontology.

## 2 Background and Related Work

An information support plays a key role in the learning process and determines its quality and efficiency. Competency-based learning is directed to the acquirement of appropriate competencies, knowledge, and skills. Achievement of this goal is impossible without an appropriate information support of learning that implies using the relevant information and educational resources.

Information support of learning processes involves thematic information retrieval and creation of personal learning collections of information resources for some topic as a result of their retrieval and integration. The creation of such collections can be performed on the basis of user cognitive space and information space of subject domain.

A cognitive space is the set of concepts and relations among them held by a human. The cognitive space can be individual as well as shared by a group of people. Using the modern software tools the cognitive space can be mapped into conceptual model represented as a mind map, topic map, concept map (conceptual diagram) or ontology. An information space is the set of objects and relations among them held by an information system. The components of the information space for the information retrieval task include concepts, documents, words, relations among words and documents. So the information space should be consistent with the cognitive space of particular humans or groups.

An essential and actual problem is creation of the domain information space which is relevant to the personal cognitive space of subject of information process [3]. The use of ontologies for the cognitive and information spaces representation is an appropriate and promising approach to solve this problem.

Within ongoing research project, the authors proposed an ontology-based approach to the collaborative construction of the domain information space on the basis of the cognitive spaces of individuals or groups and the existing information spaces [2, 3, 18, 19]. It allows to decrease the time and increase the efficiency of retrieval and reuse of the information resources which are relevant to the subject domain and the user cognitive space. One of the important issues in the framework of this approach is representation and visualization of the domain ontology for the user.

The ontology visualization affects the efficiency of perception of domain knowledge by the learner that includes the concepts and concept hierarchy,

different types of relations between the concepts, concepts properties (in the case of the domain information space - the list of information resources, corresponding to the concepts). The requirements to ontology visualization depend on tasks defined in the ontology and user competence in the subject domain. E.g., the subject domain can be totally new for the learner, the learner can know some of the concepts (and need to find it and discover the related concepts step-by-step), or he can know most of the concepts and need to find and fill the knowledge gaps in this domain.

Generally, the requirements which the visualization method and tool should be met are formulated as follows:

- One should be able to visualize all the components of the ontology, such as classes, instances, and relations.
- The tool should allow visualizing the ontology stored independently of the visualizer as well as sharing and collaborative editing the ontology by the group of users at the same time. SPARQL-endpoint can be considered as a data source to meet this requirement.
- A simple and obvious model should be used for visualization, like a graph for example.
- Some special techniques should be used for visualization of a subset of ontology elements (fragments of ontology) for navigation during the ontology studying and transformation of these fragments within the full ontology.
- The search of ontology elements (classes, individuals, relations) and visualization of search results in the full ontology and ontology fragment should be provided by the visualizer.
- As the ontology supports reasoning, the results of the reasoning should be visualized as well.

### 3 Approaches and Software Tools for Visual Representation of Large Ontologies

The subject domain might have a rather complex structure and cover a huge set of concepts and relations between them, that complicates extremely the study of appropriate information resources by the person who is not familiar with the domain. In this case, it is advisable to create the own domain model or use and extend the domain models created by other persons. But due to the complexity, it is difficult to overview and understand conceptual model represented with the ontology.

The usage of various approaches and tools for ontology visualization and navigation depends on the user category who work with the ontology and the use cases. The user categories considering below can be defined depending on their relation to the subject domain and role within ontology engineering process:

- **Ontology Engineer:** a specialist in Ontology Engineering - a research area which studies the methods and methodologies for building ontologies: formal representations of a set of domain concepts and the relationships between

them. So, this type of user has high competence in the ontology engineering but does not always have enough competence in the applied subject domain.

- Domain expert: a specialist in applied subject domain who does not always have enough competence in the ontology engineering.
- Learner: ontology user that is not a specialist in the applied subject domain and needs to learn it, and who also has not enough competence in the ontology engineering.

For the task of domain information space studying the user category usually can vary from learner to an expert in this classification, but stays closer to the first one.

The following use cases are distinguished when working with ontologies [7]:

- Editing: ranges from developing a new ontology to changing a value of some property. Usually, the purpose of visualization, in that case, is to find the entity to be changed or to become linked with a new entity.
- Inspection: needs a detailed view of the ontology to see errors or deficiencies. It often implies Editing, as the user needs to see the state of the ontology before and after editing.
- Learning: means gaining knowledge about the domain the ontology covers, or learning about the ontology itself so as to use it. The view should not be so detailed as for errors discovering (for Inspection). The user often only needs to see the available classes and properties, their hierarchy and their domain/range relationships; it is actually especially for non-technical users who only want to learn the subject domain and unlikely to understand complex axioms anyway.
- Sharing: is similar to Learning, except one more actor for whom the ontology should be explained and shared with. The visualization should support displaying a part of the ontology; e.g., a picture of it can be made for an article describing the ontology.

In the case of using ontologies in the learning process by the learner, it usually implies learning and sharing use case (where the user is the second actor).

The following approaches to the ontology visualization can be distinguished [7]:

1. Zoom-Out Overview: zooming out to get a summary view of the ontology.
2. Radar View: displaying a small “minimap” of the displayed ontology.
3. Graphical Zoom: enlarging the displayed graphical elements.
4. Focus on Selected Entity: centering the view on a selected entity and its surroundings and hiding other parts of the ontology.
5. History: keeping the history of navigation steps performed by the user, thus allowing for undo/redo actions.
6. Pop-Up Window: displaying details on a chosen entity in a separate window.
7. Incremental Exploration: starting by a small part of the ontology and gradually expanding the nodes selected by the user (as detailed in [11]).
8. Search: text-based search leading to highlighting the matched entities.

9. Hide Selected Entity: hiding parts of the ontology the user is currently not interested in, thus avoiding a cluttered view.
10. Filter Specific Entity Type: e.g., hiding all object properties at once.
11. Fisheye Distortion: zooming in for a part of the graph and zooming out for the rest; focuses on a detail but keeps the context, see, e.g., [32].
12. Edge Bundles: grouping edges with similar paths, thus alleviating the clutter; implemented, e.g., in GLOW [12].
13. Drag&Drop Navigation: moving a graph that is bigger than the screen around by dragging it with the mouse.
14. Drag&Drop User Layout: allowing to move the individual nodes around.
15. Clustering: intelligent grouping of nodes or displaying a subset of important nodes, as in KC-Viz [23].
16. Integration with Editing: the user can select a visualized node or edge and edit its properties in, e.g., a pop-up window.
17. Graphical Editing: the tool supports creating new entities by, e.g., drawing edges between the displayed nodes.

Referring to [10], approaches 1–5 are associated with a criteria class called “Help and User Orientation”, 6–8 with “Navigation and Browsing”, 9–10 with “Dataset Reduction”, 11–14 with “Spatial Organization”, and 15 with “Information Coding”. 16–17 have no corresponding class in [10]. In each use case category most of the 17 features should ideally be supported. However, their importance can vary.

Evidently, more low-level issues and approaches can be distinguished within each of the items above. Also, some approaches can extend the set of criteria classes.

Most of the visualization approaches and software tools are oriented to the specialists in ontological modeling and are used to solve the problem of understanding ontology (“ontology sensemaking”) [23]. In this case, the relevant tasks are a search of existing ontology for reuse, optimization of large ontologies structure, when the specialist should understand the ontology structure. For this task, the tools for the high-level overviews creation can be used (1–2 in the list above), zooming (3) and element filtering (10) [23, 30, 33, 39].

Within the tasks of learning and sharing the ontology, for efficiency knowledge sharing included into the ontology, the cognitive properties of the ontology are important that allows to user easy understanding and interpreting them. In that case the visualization of the ontology implies the visualization of some fragment of this ontology including some concepts with the context (so called “viewpoint”) [1]. In [22] this problem resolved e.g., with the cognitive frame based representation of the ontology and patterns.

Also the ways of visualization below can be distinguished [21]:

1. Graph Visualizations of Ontologies [23, 25, 28, 29, 38] - a simple, understandable and convenient way for ontology presentation that almost does not require any special user knowledge and can be extended to represent special ontology properties.
2. Specific Diagram Types [15, 16].

3. UML-based Ontology Visualizations [4,20,35,37] - requires special user knowledge in UML.
4. Other Diagrammatic Ontology Visualizations [5,27,36].
5. RDF Visualizations of Ontologies [14,31] - represents the ontology as RDF graph with all concepts and relations that are not useful with the large ontologies.

Also, the software tool should support the ability to extend its functionality (with plugins or source code modification) and should be open source in the last case.

So, browsing ontologies to understand their structure and content is an important activity in ontology engineering [3,22,34]. Moreover, ontologies nowadays are developed and used not only by experts but also by a wide range of non-expert users in a lot of fields, from research, health-care to daily business. The question about how to support ontology visualization and navigation in a better way has attracted a lot of attentions from researchers. A good ontology visualization system by convention [16] should support some functionalities like a high-level overview of ontology, effective scalability by concrete part of ontology, filtering options and so on.

Usually, in the early stage of the process of making sense of ontology, what user tries to do is to understand the general organization of the ontology rather than paying attention to its specific representation details. Ontology visualizer with high overview level should support user to quickly understand what main areas the current ontology is about, how it is structured. As can be seen, the issue of ontology high-level overview has never been easy, because once ontology is large enough, it is impossible to visualize the whole ontology structure on limited screen of a computer. Different approaches have been proposed to solve this issue, including:

1. Treemaps [16], which focuses to maximizing the visualized information, using space-fill solutions.
2. OwlViz visualizer [28] - a Protege plugin, which allows user to navigate through ontology by selecting, expanding, hiding nodes.
3. Ontology Navigator [23] in Neon Toolkit [8,9,25], which uses file system navigation approach to show class hierarchy of ontology.
4. KC-Viz [23,29] in Neon Toolkit [25], proposes the approach using key concepts extraction to extract most common concepts from ontology for visualization
5. Ontology Exploration Tool [17], a tool from Hozo Ontology Viewer [13], which allows user to view ontologies structure from different viewpoints in a user-friendly visualization form called Conceptual Map.
6. Ontodia [26] - online OWL and RDF diagramming tool.

### 3.1 Treemaps

Treemaps are solution intended for visualization of ontology with complex tree hierarchies, where tree structures are visualized as 2-D space-filling representation and 100% of visualization space is used. This effective usage of visualization

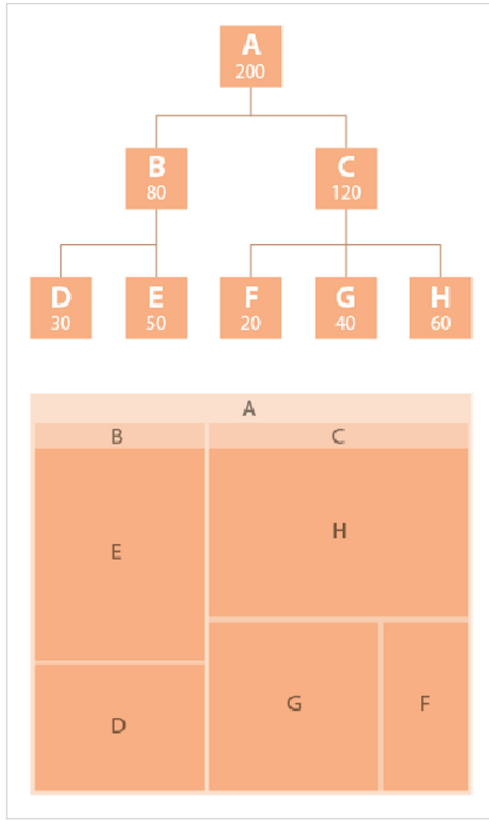
space allows displaying completely large hierarchies and facilitates representation of semantic information. In treemaps, visualization space is split into bounding rectangles that representing a tree structure. The distribution of nodes into their bounding rectangles depends on the content of the nodes and can be interactively controlled. Interactive control allows user to configure the representation method in structural level (depth, boundary, etc.) as well as in information level (display of properties such as color mapping, etc.). This is in contrasts to traditional static methods of visualizing hierarchically structured information, where there are usually low effective usages of visualization space or hiding a large amount of information from users. With Treemaps, parts of hierarchy, which contain more important information can be allocated more space than parts, which contain less important information. A sample of ontology visualization by treemaps is shown in Fig. 1.

### 3.2 Ontology Navigator (in Neon Toolkit)

Ontology Navigator in Neon Toolkit [8,9,25] follows the classic navigation metaphor of file system, where clicking into a directory will open up its sub-directories. It allows user to navigate through hierarchy of ontology by simple clicks into classes to view subclasses. Although at first look, it is a pretty basic solution, compare to other complex visualization metaphors, this approach can be surprisingly effective thanks to two reason: (1) the navigation method is very familiar to users; (2) it allows to displaying a lot of information in a small space, in contrast to node-link approach, which can take up a lot of space. One of problems in Ontology Navigator is when clicking into a class, all of its subclasses are always shown up. It is impossible to control the number of subclasses that will be displayed or open more than one level with a single click. Also, it not allows to visualize the individuals. As can be seen, the visualization approach used in Ontology Navigator is still not enough to support making sense of the general mental model of ontology. The latest version of NeOn Toolkit is dated 2011 year. A sample of ontology visualization by Ontology Navigator is shown in Fig. 2.

### 3.3 KC-Viz

KC-Viz uses an innovative method based on idea of key concepts extraction to support the task of high level overview and understanding large ontologies. It provides possibility of navigation through ontology, beginning from most information-rich nodes. These nodes are extracted from ontology using algorithm of Key Concepts Extraction, which allows to create a summary of ontology. The visualization and navigation approach used in KC-Viz is analogous to visualization and navigation in maps from Geographical Information System, where large cities are displayed more noticeably than others, depending on the current level of detail. Visualization and navigation of ontology with KC-Viz begin with generation of initial summary of ontology, so that user can have general overview about ontology. A summary of ontology CodeOntology [6] (contains 65 classes, 17 individuals, 94 object properties and 13 data properties) is shown



Data types in C

■ Scalar data types ■ Structured data types ■ Pointer ■ Void type

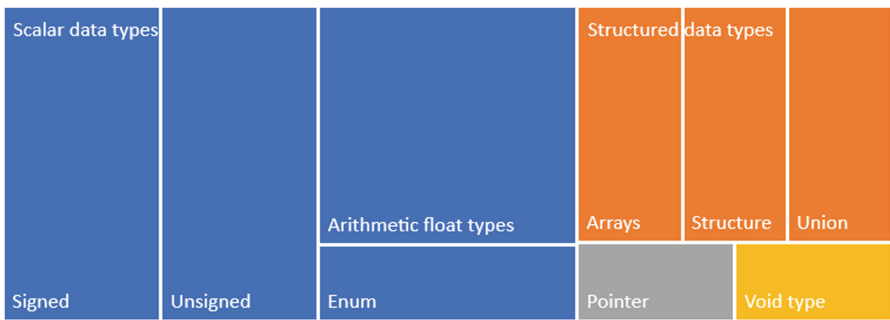
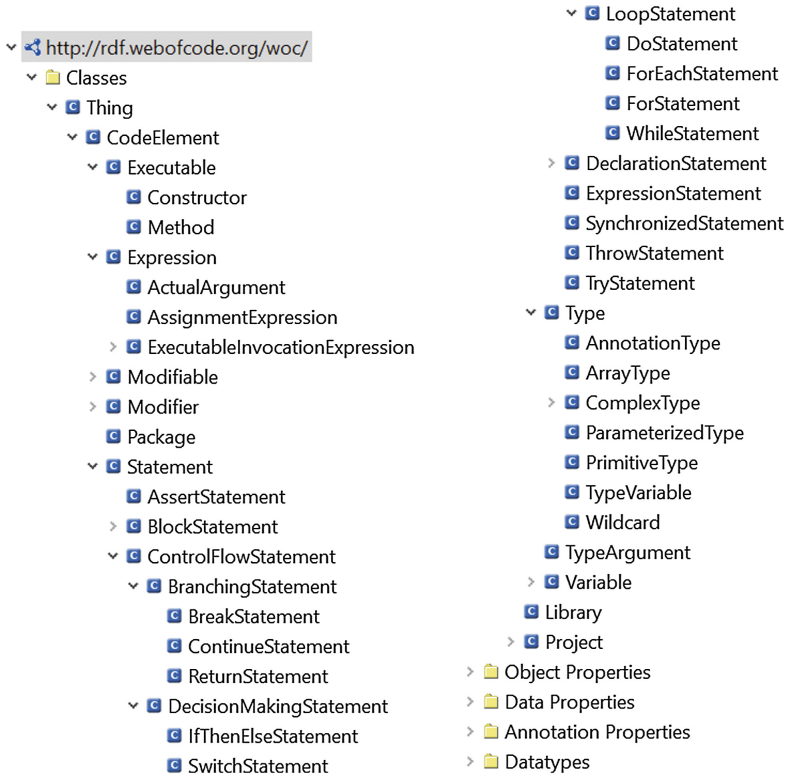


Fig. 1. Visualization of a simple tree structure with treemaps approach





**Fig. 2.** Visualization of the CodeOntology [6] with Ontology Navigator in NeOn Toolkit

in Fig. 3. This summary includes 16 most common classes (or may be a different amount, depending on the configuration of user) from different level, 15 of which are extracted by Key Concepts Extraction algorithm and the last concept is owl:Thing in order to create a connected graph. Grey arrows in the picture denote the direct rdfs:subClassOf relationships, and the dash green arrows denote the indirect rdfs:subClassOf relations. By hovering mouse to the indirect links, user can see the chain of direct rdfs:subClassOf relations, which summarized by indirect links. For user to capture an image of how large the current ontology is, KC-Viz displays two number inside each visualized node, indicates the number of direct and indirect subclasses of current class. Detail information about each class can be seen by hovering mouse to selected node.

So this approach is efficiency enough to visualize large ontologies regardless of the number of concepts included. It does not allow to display the instances, but the algorithm can be extended to do this.

The last version of NeOn Toolkit with KC-Viz algorithm integrated is dated 2011 year.

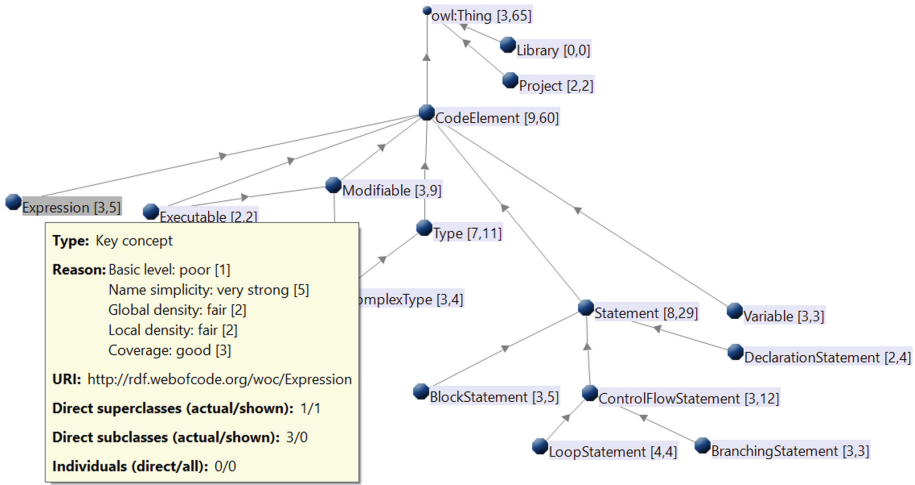


Fig. 3. Summary of CodeOntology [6] with KC-Viz

### 3.4 Ontology Exploration Tool (in Hozo Ontology Viewer)

Ontology Exploration Tool allows to visualize the conceptual structure of ontology as generally as possible and allows on-the-fly reorganizing some conceptual structures according to users perspective. It provides a possibility of exploring the ontology from several viewpoints to obtains an understanding of the ontology from multi-perspectives. The viewpoint is defined as a combination of a focal point and an aspect. The focal point indicates a concept to which the user pays attention as a starting point for exploration. The aspect is the manner in which the user explores the ontology. It can be the represented by a set of methods for extracting concepts according to its relations because an ontology consists of concepts and relations among them [17]. An example of ontology exploration with Ontology Exploration Tool is shown in (Fig. 4). A user sets Any (the root of ontology) as a focal point and sets Sub concepts to be the aspect of exploration. The result is that nine concepts: representation, substance, measure, Entity, attribute, location, Proposition, time, representation form are extracted, following is-a relations (rdfs:subClassOf in OWL). Next, the user focuses on concept location and selects Concepts to be referred by some relationship as an aspect, the following concepts: animal, action, basic judgment, pass, make fly, ... are extracted, following attribute-of relations (rdfs:properties in owl). As a result of this concept extraction, conceptual chains that match users interest are generated and visualized as a conceptual map. In the conceptual map, extracted concepts and followed relationships are represented as nodes and links respectively, and the nodes are located on concentric circles in which the focal point is located at the center [17].



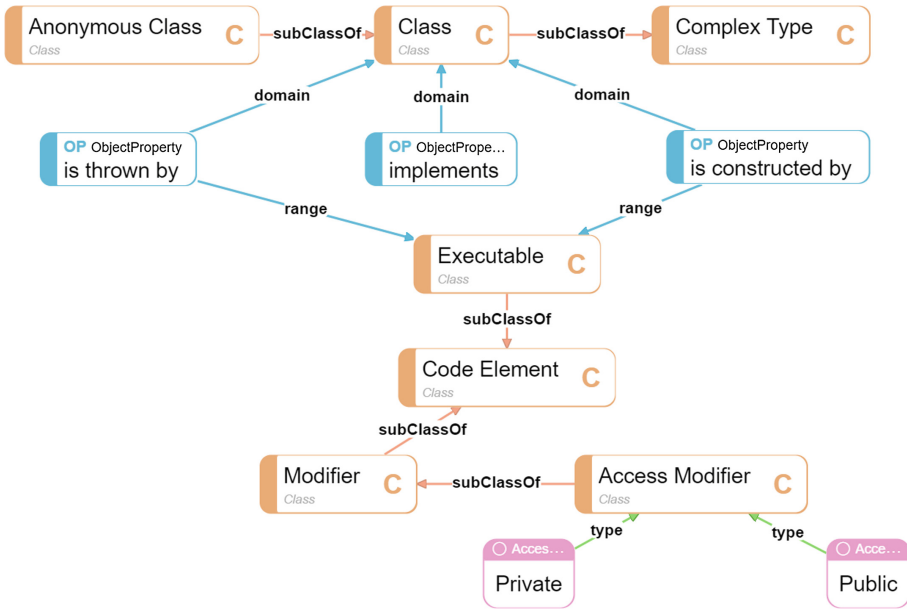


Fig. 5. Ontodia: visualization of CodeOntology fragment

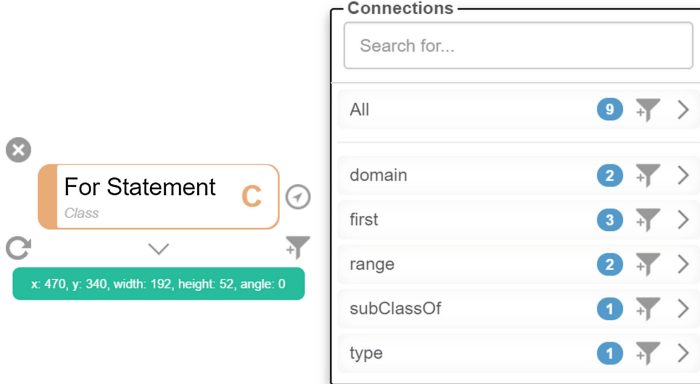


Fig. 6. Ontodia: sequential disclosure of ontology elements through the selection of interested classes, individuals and relationships

Also, this tool can work with RDF files or any SPARQL endpoint as a data source but does not work as a standalone application and requires to include the data source in the javascript code.

As it is an open source project, the functionality can be extended using different approaches including mentioned above.

## 4 Comparative Analysis of Software Tools for Graph-Based Ontology Visualization

We tried to compare the features of software tools mentioned above in the context of the task of graph-based visualization of large ontology. We defined the criteria for comparing that seem to us most important and informative for the considered problem. The results of the comparative analysis are presented in the Table 1.

**Table 1.** Comparison of software tools for graph-based ontology visualization

Software tool	Protege with OntoGraf plugin	WebVOWL	Ontology Exploration Tool (in Hozo)	KC-Viz in Neon Toolkit	Ontodia
SPARQL-endpoint as a data source	Not available	Not available	Not available	Not available	Available (endpoint is hardcoded)
Visualization of classes and instances	Yes (OntoGraf)	Classes only	Yes	Classes only	Yes
Visualization of the ontology after transformation (using visualization patterns)	No	No	No	No	No
Creation of ontology by specifying graph	No (only manual graph creation using ontology concepts)	No	No	No (only manual graph creation using ontology concepts)	No (only manual graph creation using ontology concepts)
Visualization of subset of elements/ fragment of ontology	Manual adding the ontology elements to the workspace with graph and nodes expansion (OntoGraf)	Only graph scaling and scrolling	No (only full ontology visualization)	Yes (visualization of key concepts and manual expansion of these concepts)	Manual adding the ontology elements to the workspace with graph

*(continued)*

**Table 1.** (*continued*)

Software tool	Protege with OntoGraf plugin	WebVOWL	Ontology Exploration Tool (in Hozo)	KC-Viz in Neon Toolkit	Ontodia
Search of ontology elements	Search of classes and individuals, graph visualization with graph rebuilding	Search of classes and relations with search string autocomplete, results highlighting on the graph	Search of classes	No	Sequential search of classes, individuals and relations
Visualization of the ontology with reasoning results	No	No	No	No	Yes (if supported by backend connected with SPARQL-endpoint)

## 5 Conclusion and Future Work

In this paper, we have compared some visualization approaches as well as software tools for the purpose of their application for visualization of large ontologies in learning. We faced this problem during our work on the project devoted to the collaborative construction of the domain information space based on the cognitive spaces of individuals or groups in learning and scientific research. The considered task defines a number of specific requirements for visual representation of the ontology, and none of the existing software tools provide the possibilities to satisfy these requirements. We chose the open-source javascript tool Ontodia as a promising visualization environment due to the following advantages. It allows connecting to the SPARQL-endpoint as a data source to separate the ontology storage and viewing/editing functions and provides collaborative developing, sharing and exploring the ontology. Also, this tool allows visualizing the ontology derived knowledge, as well as originally defined knowledge because the reasoning engine works separately from the ontology storage. Ontodia provides the possibilities for visualization of all the types of ontology elements, search and visualization of the search results, for the creation of ontology graph by different ways.

As Ontodia is open source project, it can be improved and extended with additional approaches, algorithms, and functions. The integration the KC-Viz algorithm into the Ontodia library seems to be a promising way to fix the main

shortcomings and improve the usability of this tool and allows the user to start the domain ontology studying from the most important concepts.

Further evolution of the project implies extending the Ontodia diagramming tool with the functionality based on the approaches reviewed above (particularly, KC-Viz approach) that allows improving an efficiency of ontological modeling.

**Acknowledgement.** This paper presents the results of research carried out under the RFBR grant 15-07-03541 Intelligent support of decision making in management of large scale systems on the base of integration of different types of reasoning on ontological knowledge.

## References

1. Acker, L., Porter, B.: Extracting viewpoints from knowledge bases. In: Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI 1994, vol. 1, pp. 547–552. American Association for Artificial Intelligence, Menlo Park (1994). <http://dl.acm.org/citation.cfm?id=199288.199322>
2. Anikin, A., Kultsova, M., Zhukova, I., Sadovnikova, N., Litovkin, D.: Knowledge based models and software tools for learning management in open learning network. In: Kravets, A., Shcherbakov, M., Kultsova, M., Iijima, T. (eds.) JCKBSE 2014. CCIS, vol. 466, pp. 156–171. Springer, Cham (2014). doi:[10.1007/978-3-319-11854-3\\_15](https://doi.org/10.1007/978-3-319-11854-3_15)
3. Anikin, A., Litovkin, D., Kultsova, M., Sarkisova, E.: Ontology-based collaborative development of domain information space for learning and scientific research. In: Ngonga Ngomo, A.-C., Křemen, P. (eds.) KESW 2016. CCIS, vol. 649, pp. 301–315. Springer, Cham (2016). doi:[10.1007/978-3-319-45880-9\\_23](https://doi.org/10.1007/978-3-319-45880-9_23)
4. Barzdins, J., Barzdins, G., Cerans, K., Liepins, R., Sprogis, A.: OWLGrED: a UML style graphical notation and editor for OWL 2. In: Sirin, E., Clark, K. (eds.) Proceedings of the 7th International Workshop on OWL: Experiences and Directions (OWLED 2010), San Francisco, California, USA, 21–22 June 2010. CEUR Workshop Proceedings, vol. 614. CEUR-WS.org (2010). [http://ceur-ws.org/Vol-614/owled2010\\_submission\\_5.pdf](http://ceur-ws.org/Vol-614/owled2010_submission_5.pdf)
5. Catenazzi, N., Sommaruga, L., Mazza, R.: User-friendly ontology editing and visualization tools: the OWLeasyVIZ approach. In: 2009 13th International Conference Information Visualisation. pp. 283–288, July 2009
6. CodeOntology. <https://doi.org/10.5281/zenodo.577939>
7. Dudáš, M., Zamazal, O., Svátek, V.: Roadmapping and navigating in the ontology visualization landscape. In: Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E. (eds.) EKAW 2014. LNCS (LNAI), vol. 8876, pp. 137–152. Springer, Cham (2014). doi:[10.1007/978-3-319-13704-9\\_11](https://doi.org/10.1007/978-3-319-13704-9_11)
8. Erdmann, M., Waterfeld, W.: Overview of the NeOn toolkit. In: Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A. (eds.) Ontology Engineering in a Networked World, pp. 281–301. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-24794-1\\_13](https://doi.org/10.1007/978-3-642-24794-1_13)
9. Espinoza, M., Gómez-Pérez, A., Mena, E.: Enriching an ontology with multilingual information. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 333–347. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-68234-9\\_26](https://doi.org/10.1007/978-3-540-68234-9_26)

10. Freitas, C., Luzzardi, P.R.G., Cava, R.A., Winckler, M., Pimenta, M.S., Nedel, L.P.: On evaluating information visualization techniques. In: Proceedings of the Working Conference on Advanced Visual Interfaces, AVI 2002, pp. 373–374. ACM, New York (2002). <http://doi.acm.org/10.1145/1556262.1556326>
11. Herman, I., Melancon, G., Marshall, M.S.: Graph visualization and navigation in information visualization: a survey. *IEEE Trans. Vis. Comput. Graph.* **6**(1), 24–43 (2000)
12. Hop, W., de Ridder, S., Frasinca, F., Hogenboom, F.: Using hierarchical edge bundles to visualize complex ontologies in glow. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. SAC 2012, pp. 304–311. ACM, New York (2012). <http://doi.acm.org/10.1145/2245276.2245338>
13. Hozo: Ontology editor. <http://www.hozo.jp>
14. IsaViz: A Visual Authoring Tool for RDF. <https://www.w3.org/2001/11/IsaViz/>
15. Jambalaya - Protege plug-in. <https://protegewiki.stanford.edu/wiki/Jambalaya>
16. Johnson, B., Shneiderman, B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In: Proceedings of the IEEE Conference on Visualization, Visualization 1991, pp. 284–291, October 1991
17. Kozaki, K., Hirota, T., Mizoguchi, R.: Understanding an ontology through divergent exploration. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., Leenheer, P., Pan, J. (eds.) *ESWC 2011*. LNCS, vol. 6643, pp. 305–320. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21034-1\_21
18. Kultsova, M., Anikin, A., Litovkin, D.: An ontology-based approach to collaborative development of domain information space. In: Proceedings of the 12th International Conference on Educational Technologies (EDUTE 2016), Proceedings of the 10th International Conference on Business Administration (ICBA 2016), Barcelona, Spain, 13–15 February 2016, pp. 13–19 (2016). <http://www.wseas.us/e-library/conferences/2016/barcelona/EDBA/EDBA-01.pdf>
19. Kultsova, M., Anikin, A., Zhukova, I., Dvoryankin, A.: Ontology-based learning content management system in programming languages domain. In: Kravets, A., Shcherbakov, M., Kultsova, M., Shabalina, O. (eds.) *CIT&DS 2015*. CCIS, pp. 767–777. Springer, Cham (2015). doi:10.1007/978-3-319-23766-4\_61
20. Liepins, R., Grasmanis, M., Bojars, U.: OWLGrED ontology visualizer. In: Verborgh, R., Mannens, E. (eds.) *Proceedings of the ISWC Developers Workshop, Co-located with the 13th International Semantic Web Conference (ISWC 2014)*, Riva del Garda, Italy, 19 October 2014, *CEUR Workshop Proceedings*, vol. 1268, pp. 37–42. CEUR-WS.org (2014). <http://ceur-ws.org/Vol-1268/paper7.pdf>
21. Lohmann, S., Negru, S., Haag, F., Ertl, T.: Visualizing ontologies with VOWL. *Semant. Web* **7**(4), 399–419 (2016). <http://dx.doi.org/10.3233/SW-150200>
22. Lomov, P.: Primenenie patternov ontologicheskogo proektirovaniya dlya sozdaniya i ispolzovaniya ontologiy v ramkah integrirovannogo prostranstva znaniy. *Ontologiya proektirovaniya* **5**, 233–245 (2015)
23. Motta, E., Mulholland, P., Peroni, S., d’Aquin, M., Gomez-Perez, J.M., Mendez, V., Zablith, F.: A novel approach to visualizing and navigating ontologies. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) *ISWC 2011*. LNCS, vol. 7031, pp. 470–486. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25073-6\_30
24. Mouromtsev, D., Pavlov, D., Emelyanov, Y., Morozov, A., Razdyakonov, D., Galkin, M.: The simple, web-based tool for visualization and sharing of semantic data and ontologies. In: *CEUR Workshop Proceedings*, p. 77 (2015)
25. NeOn toolkit. <http://neon-toolkit.org>



26. OntoDia. <http://www.ontodia.org>
27. OntoViz. <http://protegewiki.stanford.edu/wiki/OntoViz>
28. OWLViz. <https://protegewiki.stanford.edu/wiki/OWLViz>
29. Peroni, S., Motta, E., d'Aquin, M.: Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. In: Domingue, J., Anutariya, C. (eds.) ASWC 2008. LNCS, vol. 5367, pp. 242–256. Springer, Heidelberg (2008). doi:10.1007/978-3-540-89704-0\_17
30. Plaisant, C., Grosjean, J., Bederson, B.B.: Spacetree: supporting exploration in large node link tree, design evolution and empirical evaluation. In: IEEE Symposium on Information Visualization, INFOVIS 2002, pp. 57–64 (2002)
31. RDF Gravity. <https://www.w3.org/2001/sw/wiki/RDFGravity>
32. Sarkar, M., Brown, M.H.: Graphical fisheye views of graphs. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1992, pp. 83–91. ACM, New York (1992). <http://doi.acm.org/10.1145/142750.142763>
33. Shneiderman, B.: Tree visualization with tree-maps: 2-D space-filling approach. *ACM Trans. Graph.* **11**(1), 92–99 (1992). <http://doi.acm.org/10.1145/102377.115768>
34. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proceedings of the 1996 IEEE Symposium on Visual Languages, VL 1996, pp. 336–343 (1996). <http://dl.acm.org/citation.cfm?id=832277.834354>
35. TopBraid Composer. <http://www.topquadrant.com/tools/modeling-topbraid-composer-standard-edition/>
36. VisioOWL. <https://sites.google.com/site/semanticsimulations2/visioowl>
37. Visual Ontology Modeler. <http://thematix.com/tools/vom/>
38. VOWL: Visual Notation for OWL Ontologies. <http://vowl.visualdataweb.org/v2/>
39. Wang, T.D., Parsia, B.: CropCircles: topology sensitive visualization of OWL class hierarchies. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 695–708. Springer, Heidelberg (2006). doi:10.1007/11926078\_50