Ontology Visualization Tools: A Bibliographic Review and a Proposal

Ezra Gomes Marques
ALGORITMI Research Centre/LASI - DI, University of Minho, Braga, Portugal

Cristiana Araújo
ALGORITMI Research Centre/LASI - DI, University of Minho, Braga, Portugal

Pedro Rangel Henriques
ALGORITMI Research Centre/LASI - DI, University of Minho, Braga, Portugal

Abstract

Information can be graphically presented to users through different visualization methods, which can be complemented by various features meant to facilitate the navigation and interpretation of the presented information. Many of these methods and features are already in use by current ontology visualization tools, while others, along with some concepts from the broader field of information visualization, are merely explored in the literature despite their potential to also be applied in this specific field. This paper aims to analyze key characteristics of the current state of the art of ontology visualization tools, as well as the notable benefits and limitations of each of them, by extensively reviewing this body of work. Additionally, it also aims to demonstrate how the resulting insight has aided the development of PrOnto, a new platform dedicated to the visualization of OntoDL+ ontologies.

2012 ACM Subject Classification Information systems → Ontologies; Human-centered computing → Information visualization

Keywords and phrases Ontology, Information Visualization, Ontology Visualization, Ontology Navigation, OntoDL+

Digital Object Identifier 10.4230/OASIcs.SLATE.2024.3

Funding This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.
The Ph.D. work of Cristiana Araújo is supported by FCT – Fundação para a Ciência e Tecnologia, Research Grant, with reference 2020.09845.BD, `https://doi.org/10.54499/2020.09845.BD`.

1 Introduction

When attempting to describe and correlate various concepts, the development of an ontology may prove an efficient method of synthesizing complex information. By providing a formal explicit description of the concepts and relations among concepts in a given domain of discourse [19], an ontology can aid the understanding and analysis of the information contained within it, while also establishing a common vocabulary to facilitate its discussion [31].

However, parsing the entirety of the information contained within a complex ontology in a purely textual format may quickly become challenging and overwhelming. Alternatively, providing users with a graphical representation of the same information can facilitate its interpretation considerably [40, 14, 8]. These graphical representations of ontologies, or ontology visualizations, can take a variety of forms [13, 24, 22, 38], with the most commonly used one in current ontology visualization tools being the *two-dimensional node-link graph* [13].

Unfortunately, static ontology visualizations often do not scale properly, as they can easily become cluttered and difficult to interpret when representing ontologies with a large number of elements. In order to overcome this limitation, these visualizations are often complemented by various features within ontology visualization tools (such as zooming and panning, search and filtering features, etc.), which allow for finer navigation over an ontology's contents [13].

Implementing a wide variety of these features within a single ontology visualization tool can also make it more flexible and capable of accounting for various user preferences and needs [13, 16, 35]. Similarly, incorporating different visualization methods within a singular platform can also yield the same benefits [13, 16, 34, 24].

Despite this, all of the currently available ontology visualization tools have limitations in regards to what features and visualization methods they can support. Some of these tools are also commercial products, and are not freely available.

As such, this paper aims to analyze the current state of the art of ontology visualization tools, the various visualization methods and techniques they employ, and their respective advantages and disadvantages, by conducting an extensive review of the literature related to current ontology visualization tools and/or concepts which could be applied in the specific context of ontology visualization. This analysis will also inform the development of PrOnto, a platform for visualizing and navigating ontologies described in OntoDL+, which will also be presented in this document.

OntoDL+ is a Domain Specific Language (DSL) which, as of currently, is not supported by any available ontology visualization tool. It has been specifically designed to have a simple syntax meant to facilitate the process of defining and instantiating ontologies [11]. Therefore, the development of an open access platform specifically suited for the visualization of ontologies described in OntoDL+ would prove an essential step in providing better support for the use of this emerging DSL.

This paper is structured in three sections after this one: Section 2 examines the visualization methods, features and information visualization concepts which are relevant for the current state of the art of ontology visualization tools; Section 3 introduces PrOnto, describes its features and illustrates the GUI offered by the present prototype; and Section 4 discusses the conclusions drawn and proposes some directions for future work.

2 State of the Art of Ontology Visualization

The current state of the art of ontology visualization tools employs a variety of visualization methods and techniques, while also borrowing concepts from the broader field of information visualization.

2.1 Information Visualization

The concept of information visualization refers to a cognitive process where, through the perception of a graphical representation of data, referred to as a visualization, a person understands and internalizes the information they are being presented with [28, 40].

In general, visualizations can operate as external memory and cognition aids, which can help users significantly in keeping track of and interpreting information [14]. More specific benefits have also been listed extensively by [8].

Naturally, ontology visualization is encompassed by the broader field of information visualization. As such, some concepts of information visualization are also relevant in the specific context of ontology visualization, as is the case with the following two.

Retinal Properties Retinal properties refer to graphical properties which the retina of the eye is sensitive to independent of their spatial positioning [5, 8], and include color, shape, size, saturation and texture. These can add an extra dimension of expressiveness to an ontology visualization, while encoding a variety of additional information in an easily perceived manner [40, 13]. However, most of these properties are still underutilized in current ontology visualization tools [13].

Mental Map The mental map, or mental model, is something users acquire as they navigate and interact with a visualization, and refers to their understanding of its overall structure and the general location and relationships between its parts [15]. Preservation of the mental map is a necessary design consideration for dynamic and/or interactive visualizations, lest users become disoriented or overwhelmed. Some factors that can affect mental map preservation are the visualization's predictability, degree of change, and traceability [15].

2.2 Visualization Methods

In the context of ontology visualization, visualization methods refer to the ways in which the information contained within an ontology, namely its various entities and relations, is graphically displayed to the user.

Of the methods currently in use, the vast majority are two-dimensional (2D), with a few experimenting with three-dimensional (3D) visualizations or incorporating 3D aspects with a mostly 2D visualization. A few methods are also capable of displaying changes that occur in an ontology over time, thus being able to account for a temporal dimension.

2.2.1 Indented Lists

Indented lists, or indented trees, are a simple visualization method which displays a list of entities, with entities lower in the hierarchy shown indented under their parent entity. As such, despite being technically 2D, they depend mainly on one dimension, the horizontal positioning of the displayed elements.

Since indented lists can only display hierarchical relationships, they are often implemented in combination with other visualization methods in ontology visualization tools [24]. Their similarities to file browsers make them intuitive to navigate for users [24, 25, 16, 29], and they are effective at displaying a lot of information within a limited space [29] without suffering from common issues such as cluttered displays or overlapping elements [24]. Studies have also shown indented lists to be more efficient in supporting information searches than node-link visualizations [16, 17], although they are less suitable for overviews [16, 17, 24], and do not display multiple inheritance cases effectively [24, 16].

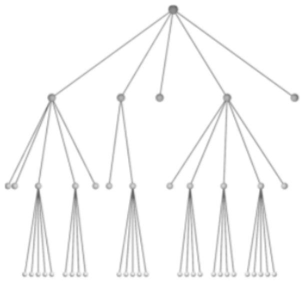
2.2.2 2D Visualizations

2D visualizations are, by far, the most widely used visualizations by the current state of the art. These visualizations can also be further categorized into node-link, space-filling and matrix visualizations.

2.2.2.1 Node-link Visualizations

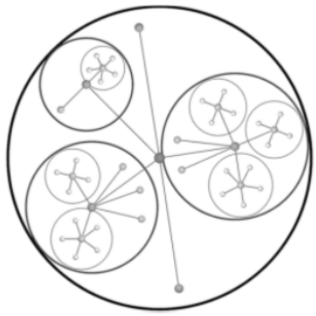
Node-link visualizations utilize labeled nodes and links to represent an ontology's entities and relations respectively. Each node may be labeled either with just the name of the entity it represents, or include other information related to it, such as its properties.

Node-link visualizations can have various layouts, some of which are only suitable for the representation of hierarchical relations, as is the case with tree layouts (used, for example, in *OWLViz* and *KC-Viz*), radial layouts (used in *OntoRama*), and balloon layouts (not in use by any notable ontology visualization tools) (see Figure 1) [13]. Tree layouts in particular have also been noted to make an inefficient use of screen space [24], with radial layouts being more efficient in this regard [13].



(a) Tree layout.

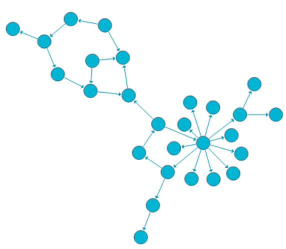
(b) Radial layout.



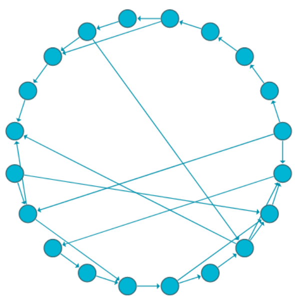
(c) Balloon layout.

Figure 1 Hierarchical node-link layouts (Figures from [23]).

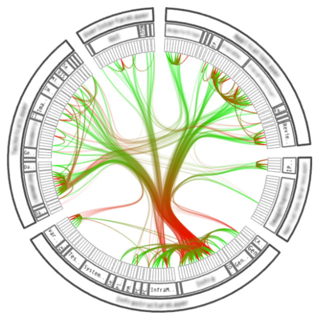
Since ontologies can include both hierarchical and non-hierarchical relations, layouts which are capable of representing both types of relations effectively are more suitable for visualizing ontologies in the majority of cases. Examples of these layouts include force-directed layouts (used, for example, in *WebVOWL* and *NeOn Toolkit's Ontology Visualizer*), circle layouts (used in *OWL-VisMod* and *NavigOWL*), and inverted tree layouts (used in *GLOW* in combination with edge bundling techniques) (see Figure 2) [13].



(a) Force-directed layout.



(b) Circle layout.



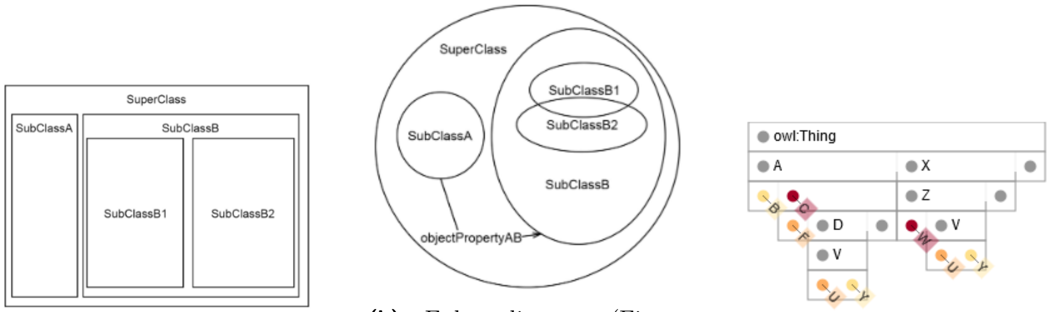
(c) Inverted radial tree layout with edge bundling (Figure from [23]).

Figure 2 Non-hierarchical node-link layouts.

Force-directed layouts are one of the most commonly used layouts for node-link visualizations, since they can produce aesthetically pleasing results devoid of overlapping elements [13]. However, force-directed algorithms are computationally complex and can generate unpredictable layouts [22, 9]. Alternatively, circle and inverted radial tree layouts take on a different approach, where they do not avoid edge crossings but apply edge bundling techniques to facilitate the visualization of large numbers of relations [23].

2.2.2.2 Space-filling Visualizations

Space-filling visualizations represent hierarchical structures, while utilizing the full space available for the display. Relations in these layouts can be represented via enclosure, adjacency or overlap of the shapes depicting the structure's entities [38]. Some examples of space-filling layouts are treemaps (used in *OWL-VisMod* and *Jambalaya*), Euler diagrams (used in *OWLeasyViz*) and icicle trees (used in *OntoPlot* [41]) (see Figure 3) [13].



(a) Treemap (Figure from [13]). **(b)** Euler diagram (Figure from [13]). **(c)** Icicle tree in *OntoPlot*.

Figure 3 Space-filling layouts.

These visualizations are less suitable for visualizing ontologies as it is difficult to represent non-hierarchical relationships within them without producing very cluttered displays [38]. Additionally, while they have been noted to represent entity level information efficiently via size and color encoding, they are also less effective for navigating hierarchies than node-link visualizations, with treemaps performing particularly poorly in this regard [24, 29].

2.2.2.3 Matrix Visualizations

Matrix visualizations represent entities as rows and columns, and the relationships between them as filled cells at the intersection of the connected rows and columns. Color coding of these cells can also be used to denote different types of relationships.

This visualization method is less commonly used for visualizing ontologies, although it has been used for this end in combination with a node-link visualization in *OntoTrix* [38, 3], where the general graph structure is represented by a node-link visualization, while particularly dense areas are displayed through adjacency matrices.

While this visualization method is effective at displaying large amounts of information in a limited space without producing overlapping elements, it is also less familiar to users than node-link visualizations, and often leads them to struggle with path finding tasks when presented with them [38, 3, 18].

2.2.3 2.5D Visualizations

Currently there is only one ontology visualization tool which combines both 2D and 3D elements: *Ontoviewer* [13].

Ontoviewer includes a variety of visualization methods, the most distinctive of them being a 2.5D graph composed of a node-link visualization with a radial layout in a 2D plane representing inheritance relations, and role relations displayed as curved links in the space above that plane [34, 35]. This separation of the two types of relationships effectively facilitates the analysis of the ontology [35].

This method, however, can still present issues with edge crossings, namely within its 2D radial layout when displaying several cases of multiple inheritance [35]. While the curved links on the 3D space can also overlap, this issue is significantly mitigated by *Ontoviewer*'s navigation and interaction features.

2.2.4 3D Visualizations

The most significant examples of ontology visualization tools that employ 3D visualizations are *OntoSphere* and *OntoTrek*, both of which utilize node-link visualizations in a 3D space.

OntoSphere offers two views of the ontology in the form of tree-like 3D layouts, and a third view which, more distinctively, organizes the nodes over the surface of a large translucent sphere [6, 13].

OntoTrek, meanwhile, utilizes a singular tree-like 3D layout that places each node on the z axis according to its hierarchical level, and applies a force-directed algorithm to dictate the positioning of sibling nodes in the xy plane [12]. This layout could be considered a more interesting development of the cone tree layout used in older tools such as *OntoSELF* and *Onto3DViz* [37, 20].

Other relevant 3D layouts which are not present in any current visualization tools are the 3D hyperbolic tree layout of *H3Viewer* [30], and the nested cubes approach proposed by [33], which adapts a treemap approach to a 3D space. Both of these cases are discussed in greater detail on the thesis document.

3D visualization methods are less commonly used than 2D methods partly due to their recency [40], but also due to the ongoing controversy regarding their benefits over 2D approaches [24]. While 3D visualizations are sometimes thought of as more visually appealing [36], they often do not allow users to interpret data as easily as 2D approaches [40, 28]. Additionally, they are often more difficult to navigate [24], and suffer from the loss of depth perception that occurs when translating a 3D structure to a 2D screen [28, 13]. Despite this, such issues can be avoided by the implementation of appropriate navigation features [24, 13], or even through the use of transparency in select parts of the structure [33, 6]. 3D visualizations also hold potential due to their ability to encode data in an unique way, and to make a more efficient use of screen space than 2D visualizations [24].

2.2.5 Temporal Dimension

As there is often an user need to be able to observe the evolution of changes made to an ontology over time [26, 27], a few ontology visualization tools have also been developed with this temporal dimension in mind.

Some tools or plugins can display the history of changes made to an ontology through the use of pie charts, word clouds, lists or other representations, as is the case with *CODEX* [21, 26]. Other tools may incorporate this information into the main ontology visualization method itself, in which case they may take one of two approaches: time-to-time or time-to-space mappings [7, 4]. Hybrid visualizations that combine elements from both categories are also possible, albeit more sparingly researched [4].

2.2.5.1 Time-to-time Mappings

The passage of time and its effects on dynamic structures can be represented through the use of animation. While this approach has not been applied in any currently available ontology visualization tools, it has been studied in reference to animated node-link visualizations [4] which could be used in this context.

This visualization method is suitable for tracing the evolution of specific concepts over time [26], but has many disadvantages. Namely, it tends to cause cognitive overload for the users [7, 4, 26]; suffer from layout stability problems which require advanced algorithms to counter [7, 4, 10]; and make interaction difficult, since the animation must be paused in order to allow for it to happen [7].

2.2.5.2 Time-to-space Mappings

Time-to-space mappings are often referred to as timelines, and can be implemented through various approaches: by overlaying snapshots of the visualization at different points in time or by placing these snapshots next to each other; by integrating timeline information into the static visualization itself; or by combining two or more of these approaches [4].

It should be noted that information can be integrated into a static visualization in various ways, namely by encoding information through color, size and/or shape of the elements (as is the case in *OntoDiffGraph*'s node-link visualization [27], or *PROMPT-VIZ*'s treemap visualization [32]) or by revealing additional information upon interaction with the graph structure (as described by [4] in regard to adjacency matrices). However, it has also been noted that the use of too many visual features at once, which may occur with this approach, can make it more difficult to visually analyze an ontology for specific aspects [7].

Regarding general limitations of time-to-space mappings, their main disadvantage lies in how they must convey a lot of information within a limited space [4], with this issue being particularly prevalent in the case of the first approaches mentioned. However, time-to-space mappings of information are preferable for analyzing visualizations of long periods of time than time-to-time mappings, since they allow for comparison tasks to be performed visually rather than mentally [7, 4].

2.3 Features

In addition to a proper visualization method, there are various visualization and interaction techniques that can be implemented as useful features of an ontology visualization tool's interface in order to provide the user with essential methods to navigate and interact with the ontology.

While a few advanced features are also discussed in the thesis document, this paper will only consider commonly used basic features, namely:

Graphical Zoom Enlarging or decreasing the size of the displayed graphical elements;

Panning Moving the entire visualization by dragging it or the background, as opposed to using scrollbars;

Incremental Exploration Displaying only part of the ontology, and expanding the visualization according to what entities the user selects next;

Entity Focus Centering the view on a selected entity and its surroundings, through panning and/or zooming in, and omitting or obscuring the rest of the ontology;

Pop-up Window Displaying details of a selected entity in a separate window or a tooltip;

Radar View Showing a small "minimap", or bird's-eye view, of the entire ontology. It may also indicate which part of the main visualization is currently zoomed in or selected;

Drag and Drop Moving individual graphical elements around;

Search and Highlight Performing a text-based search and highlighting matching entities;

Filter Parts Hiding parts of the ontology the user is currently not interested in;

Filter Entity Type Hiding all entities of a given type or with a specific property from the visualization. This is a more specific application of the "filtering parts" feature;

History (Undo/Redo) Keeping a history of the navigation steps performed by the user, and allowing them to undo/redo actions;

Fisheye Distortion Enlarging part of the visualization while zooming out the rest of it;

Edge Bundling Grouping edges with similar paths to alleviate clutter and highlight relevant patterns in node-link visualizations;

Textual Editing Editing the name of elements or their properties;

Visual Editing Creating, removing and altering visual elements.

The use of some these features has also been noted to have occasional disadvantages, namely:

- **Zooming** in risks hiding situating context from the visualization. Therefore, its use should be complemented by other navigational cues or features [24];
- **Panning** can disorient the user and potentially damage their mental map of the ontology if this feature is not carefully implemented [24]. This is especially evident in the case of 3D visualizations, where an extra dimension is at play, thus adding more complexity to the movement/rotation interaction;
- **Incremental exploration**, when applied to a node-link force-directed layout, requires old nodes to be re-positioned when new ones are added to the visualization, in order to avoid overlapping with them. However, by changing the layout, this effect risks damaging the user's mental map of the ontology;
- **Entity focus** can also apply transition animations when focusing on the selected entity. These animations can have mixed results, as users can either find them pleasant and helpful in showing the transition path, or disorienting and needlessly time consuming [24, 25]. In 3D visualizations, if the change in focus also changes the viewing angle, disorienting rolling animations can also be produced, as experienced in *OntoTrek* [12];
- **Fisheye distortion** can often significantly distort the structure of the parts of the graph that are out of focus [39]. The distortion can also produce edge crossings when the graph's edges are not curved according to the applied distortion, as this can be more computationally demanding [22]. However, utilizing curved edges has also been noted to make path tracing tasks more difficult [39];
- **Edge bundling** requires edge labels to be either not shown or only shown on demand. As such, it may not be very fitting for the visualization of ontologies with a wide variety of role relations.

Despite this, all of these features have considerable utility. In some cases, their specific benefits have also been highlighted in the literature, as is the case with the following:

- **Incremental exploration** avoids the limitations of a top-down approach to ontology visualization (where the entire ontology is first shown to the user), which would take longer to display and might exhibit occlusion issues which this approach avoids [38];
- **Drag and drop** can add a level of flexibility that keeps users engaged, and allows them to better keep track of which entities they have already visited by allowing them to rearrange them accordingly [16];
- **Search and highlight** is a crucial feature for effective ontology navigation as browsing does not suffice in helping the user locate specific entities, especially in the case of large ontologies [24].

Finally, in order to better illustrate the current state of the art of ontology visualization tools in regards to their implementation of these features, 11 tools[1] were manually tested, with the results listed in Figure 4.

---

[1] While [13] surveyed 33 ontology visualization tools for their use of these features, most of these tools are currently unavailable or unsupported. As such, they were filtered down to 9 tools which could be installed, run and used to visualize at least one OWL ontology file or URL (therefore also excluding tools which did not allow for ontology upload), and added to 2 other more recent ontology visualization tools, *OntoPlot* [41] and *OntoTrek* [12], which were not previously included.

	Graphical Zoom	Panning	Incremental Exploration	Entity Focus	Pop-up Window
KC-VIZ	x	x	x	x	x
Neon Toolkit Ontology Navigator			x		x
Neon Toolkit Ontology Visualizer	x		x	x	
OntoGraf	x	x	x	x	x
OntoPlot			x	x	x
OntoTrek	x	x			x
OWLGrEd	x				x
OWLViz	x		x	x	
Protégé Class Browser			x		x
SOVA	x	x	x	x	x
WebVOWL	x	x			x
PrOnto	x	x	x	x	x

	Radar View	Drag and Drop	Search and Highlight	Filter Parts	Filter Types
KC-VIZ		x		x	
Neon Toolkit Ontology Navigator			x	x	
Neon Toolkit Ontology Visualizer		x	x		
OntoGraf		x	x	x	
OntoPlot			x	x	x
OntoTrek			x	x	
OWLGrEd		x	x		
OWLViz				x	
Protégé Class Browser			x	x	
SOVA		x	x	x	x
WebVOWL		x	x	x	x
PrOnto	x	x	x	x	x

	History	Fisheye Distortion	Edge Bundling	Textual Editing	Visual Editing	Total Features
KC-VIZ	x					8
Neon Toolkit Ontology Navigator				x		5
Neon Toolkit Ontology Visualizer	x					6
OntoGraf						8
OntoPlot						6
OntoTrek						5
OWLGrEd	x			x	x	7
OWLViz						4
Protégé Class Browser				x		5
SOVA						9
WebVOWL						7
PrOnto	x					11

Figure 4 Features present in currently available ontology visualization tools and in PrOnto.

3 PrOnto: A platform for visualizing and navigating ontologies

PrOnto, while still in development, already has a prototype version available at `https://pronto-computationalthinking4all.epl.di.uminho.pt/`. It is important to note that the design and development of this platform benefited greatly from the conducted analysis of the current state of the art of ontology visualization tools, which informed many decisions regarding key aspects of PrOnto, namely, its visualization method and feature requirements, which are also described in greater detail on the thesis document.

PrOnto employs a 2D node-link force-directed layout with name-only labeling as its main visualization method (see Figure 5). This approach aims to produce graphs that are intuitive and familiar to users given node-link structures' prevalence in other similar applications, while also presenting a limited number of overlapping elements. There is also a focus on incremental exploration, so that there is less of a need to compute large layouts for entire ontologies at once, thus limiting load times, and so that the users' mental map can be more easily maintained by the gradual reveal of information.

As it has been noted in Figure 4, the majority of the current ontology visualization tools do not include all the common features which may benefit such an application. Therefore, in order to give this platform a good level of flexibility that may accommodate varying user needs, PrOnto aims to include all basic features listed in Section 2.3, with the exception of those referring to visual effects (fisheye distortion and edge bundling), as their use cases are overly specific, and those referring to editing, which will have to be relegated to future work due to time constraints. The implementation of this large variety of features should also help account for the disadvantages introduced by some of them.

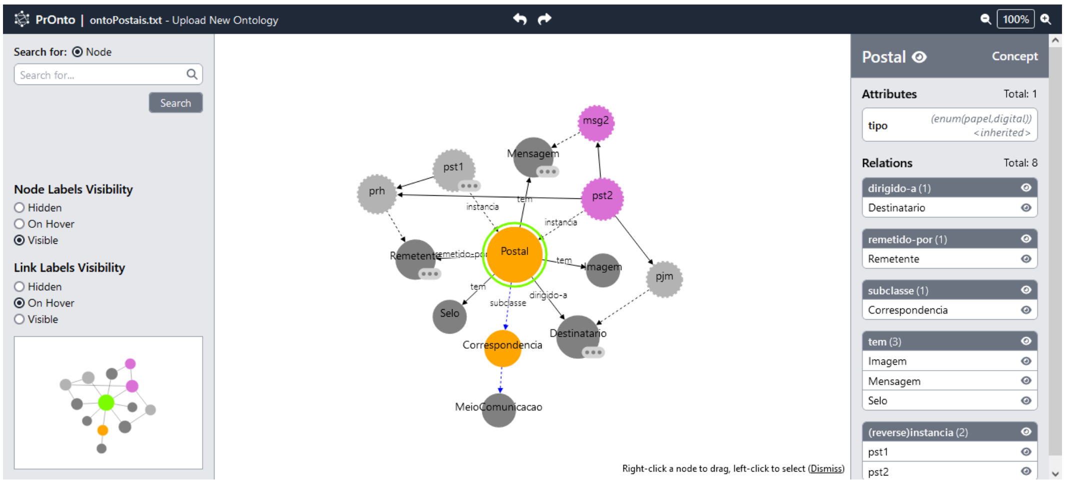


Figure 5 PrOnto's Prototype.

PrOnto also makes use of a few retinal properties in order to encode information. Color, for example, is used to distinguish concept and instance nodes, as well as to highlight selected and hovered nodes. However, as this property may not be a reliable indicator for colorblind users, it is also complemented by texture: instance nodes have squiggly borders, contrasting the smooth borders of concept nodes; and the blue aura that indicates a hovered node is dotted, while the green one for selected nodes is continuous. Finally, size is also used to highlight nodes with a higher number of relations.

Regarding PrOnto's architecture, it is composed of a front-end and a back-end, as depicted in Figure 6.

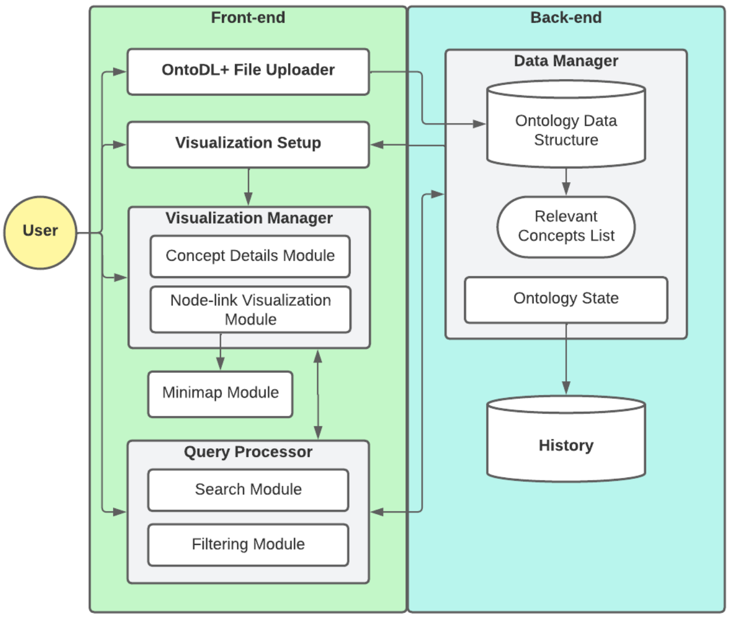


Figure 6 Architecture of PrOnto.

The front-end refers to the GUI through which the user is able to upload an OntoDL+ text file describing an ontology, setup the visualization (by choosing a starting concept or opting to visualize the entire ontology at once), and navigate and interact with it. Interactions are handled by the *Query Processor*, which communicates with the back-end to register any changes done to the visualization. It is also responsible for handling the searching and filtering functionalities. Meanwhile, the *Visualization Manager* handles the visual aspect of the GUI regarding both the node-link visualization and the information contained in the details window. It should also be noted that since the user can only observe the minimap but not interact with it, this module is not considered part of the *Visualization Manager*.

Meanwhile, all information related to the ontology's data, state, and history of user interactions is stored in the back-end. When an OntoDL+ text file describing an ontology is uploaded, a data structure containing all of its information is generated and stored within the *Data Manager*. This method, which has the back-end rely on a specified data structure instead of a stored version of the uploaded file, simplifies the process of adapting the platform to support other languages or file types in the future, as only this conversion step would need to be adjusted rather than multiple parts of the application.

Additionally, as the ontology data is processed and stored, the back-end also calculates which are its most relevant concepts, based on their number and depth of relations. These are then listed back to the user in the visualization setup window as suggested starter concepts for incremental exploration (see Figure 7), before any visualization is produced. If the user selects one of these concepts, or writes in their own valid choice, the visualization will focus on it and only show its directly related concepts upon starting.

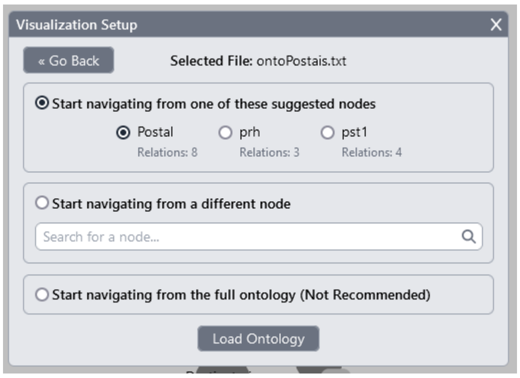


Figure 7 PrOnto's Prototype - Visualization Setup Window.

4 Conclusion

Reviewing the available literature regarding the current state of the art of ontology visualization tools allowed for a comprehensive study of the benefits and limitations of various visualization methods, as well as the identification of essential features and other aspects that should be taken into account while designing and developing such a platform. This insight was also extremely helpful in planning and developing PrOnto, a new platform dedicated to the visualization of OntoDL+ ontologies.

At the moment being, PrOnto is accessible at `https://pronto-computationalthinking4all.epl.di.uminho.pt`, and is available for testing. It has been included in the *Computational Thinking 4 All* platform so it may be used to visualize and navigate the OntoCnE, the ontology which first motivated this project. The OntoCnE is a large OntoDL+ ontology which defines and correlates the knowledge domains of Computational Thinking and Computer Programming so that it may be used as an educational tool in teaching the latter through the former, [1, 2], which is the focus of the *Computational Thinking 4 All* project. However, in order for the OntoCnE to be usable for its intended purpose, it first required a suitable platform for its visualization and navigation, such as PrOnto.

It should be noted that there is still room for future work on PrOnto. Namely, at this moment, some navigational features still need to be included. In the future, the platform would also greatly benefit from the implementation of ontology editing features and, possibly, of alternative visualization methods which may better represent some specific use cases that the current visualization method, while very versatile, may not be as suitable for. Additionally, experiments with real users are yet to be designed and performed, which will allow for a crucial assessment of the overall performance of the tool, its usability, and the user satisfaction.

In conclusion, the analysis described in this paper could serve as an useful reference point for the design and development of new ontology visualization tools, so that they may take into account the mistakes and innovations of previous works, while also making use of the most suitable techniques for their intended purposes.

References

1 Cristiana Araújo, Pedro Rangel Henriques, and João José Cerqueira. Training computational thinking to leverage citizens of next generation. In Alvaro Rocha, Hojjat Adeli, Gintautas Dzemyda, Fernando Moreira, and Valentina Colla, editors, *Information Systems and Technologies*, pages 481–490, Cham, 2024. Springer Nature Switzerland.

2 Cristiana Esteves Araújo. *Training Computational Thinking: exploring approaches supported by Neuroscience*. PhD thesis, University of Minho, March 2022. PDInf - prethesis.

3 Benjamin Bach, Emmanuel Pietriga, Ilaria Liccardi, and Gennady Legostaev. OntoTrix: A Hybrid Visualization for Populated Ontologies. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 177–180, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1963192.1963283.

4 Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. The State of the Art in Visualizing Dynamic Graphs. *EuroVis (STARs)*, 2014.

5 Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.

6 Alessio Bosca, Dario Bonino, and Paolo Pellegrino. OntoSphere: more than a 3D ontology visualization tool. In *Swap*. Citeseer, 2005.

7 Michael Burch and Steffen Lohmann. Visualizing the evolution of ontologies: A dynamic graph perspective. In *VOILA@ ISWC*, pages 69–76, 2015.

8 Stuart Card. *Information Visualization*, pages 509–543. Lawrence Erlbaum Associates, January 2008.

9 Se-Hang Cheong and Yain-Whar Si. Force-directed algorithms for schematic drawings and placement: A survey. *Information Visualization*, 19(1):65–91, 2020. doi:10.1177/1473871618821740.

10 Se-Hang Cheong, Yain-Whar Si, and Raymond K. Wong. Online force-directed algorithms for visualization of dynamic graphs. *Information Sciences*, 556:223–255, 2021. doi:10.1016/j.ins.2020.12.069.

11 Alexandre Miguel Costa Dias. ONTODL+, An ontology description language and its compiler. Master's thesis, Minho University, Braga, Portugal, September 2021. MSc dissertation.

12 Damion M Dooley and William WL Hsiao. 3D Visualization of Application Ontology Class Hierarchies. In *JOWO*, 2019.

13 Marek Dudáš, Steffen Lohmann, Vojtěch Svátek, and Dmitry Pavlov. Ontology visualization methods and tools: a survey of the state of the art. *The Knowledge Engineering Review*, 33:e10, 2018. doi:10.1017/S0269888918000073.

14 Jean-Daniel Fekete, Jarke J Van Wijk, John T Stasko, and Chris North. The Value of Information Visualization. *Information Visualization: Human-Centered Issues and Perspectives*, pages 1–18, 2008. doi:10.1007/978-3-540-70956-5_1.

15 Manuel Freire and Pilar Rodríguez. Preserving the mental map in interactive graph interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, pages 270–273, 2006. doi:10.1145/1133265.1133319.

16 Bo Fu, Natalya F Noy, and Margaret-Anne Storey. Indented tree or graph? A usability study of ontology visualization techniques in the context of class mapping evaluation. In *The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I 12*, pages 117–134. Springer, 2013. doi:10.1007/978-3-642-41335-3_8.

17 Bo Fu, Natalya F Noy, and Margaret-Anne Storey. Eye tracking the user experience–An evaluation of ontology visualization techniques. *Semantic Web*, 8(1):23–41, 2017. doi:10.3233/SW-140163.

18 M. Ghoniem, J.-D. Fekete, and P. Castagliola. A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. In *IEEE Symposium on Information Visualization*, pages 17–24, 2004. doi:10.1109/INFVIS.2004.1.

19 Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993. doi:10.1006/knac.1993.1008.

20 Simon Suigen Guo and Christine W Chan. A tool for ontology visualizaiton in 3D graphics: Onto3DViz. In *CCECE 2010*, pages 1–4. IEEE, 2010. doi:10.1109/CCECE.2010.5575219.

21 Michael Hartung, Anika Gross, and Erhard Rahm. CODEX: exploration of semantic changes between ontology versions. *Bioinformatics*, 28(6):895–896, 2012. doi:10.1093/BIOINFORMATICS/BTS029.

22 Ivan Herman, Guy Melançon, and M Scott Marshall. Graph Visualization and Navigation in Information Visualization: a Survey. *IEEE Transactions on visualization and computer graphics*, 6(1):24–43, 2000. doi:10.1109/2945.841119.

23 Danny Holten. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006. doi:10.1109/TVCG.2006.147.

24 Akrivi Katifori, Constantin Halatsis, George Lepouras, Costas Vassilakis, and Eugenia Giannopoulou. Ontology visualization methods—a survey. *ACM Computing Surveys (CSUR)*, 39(4):10–es, 2007.

25 Akrivi Katifori, Elena Torou, Constantin Halatsis, Georgios Lepouras, and Costas Vassilakis. A Comparative Study of Four Ontology Visualization Techniques in Protege: Experiment Setup and Preliminary Results. In *Tenth International Conference on Information Visualisation (IV'06)*, pages 417–423, 2006. doi:10.1109/IV.2006.3.

26 Patrick Lambrix, Zlatan Dragisic, Valentina Ivanova, and Craig Anslow. Visualization for Ontology Evolution. In *2nd International Workshop on Visualization and Interaction for Ontologies and Linked Data, Kobe, Japan, October 17, 2016*, pages 54–67. Rheinisch-Westfaelische Technische Hochschule Aachen University, 2016. URL: https://ceur-ws.org/Vol-1704/paper5.pdf.

27 André Filipe Amorim Lara. Visualization of Ontology Evolution using OntoDiffGraph. Master's thesis, Minho University, Braga, Portugal, September 2017.

28 Riccardo Mazza. *Introduction to Information Visualization*, chapter 1,2. Springer Science & Business Media, 2009.

29 Enrico Motta, Paul Mulholland, Silvio Peroni, Mathieu d'Aquin, Jose Manuel Gomez-Perez, Victor Mendez, and Fouad Zablith. A Novel Approach to Visualizing and Navigating Ontologies. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal,

Natasha Noy, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2011*, pages 470–486, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

30 T. Munzner. Exploring large graphs in 3D hyperbolic space. *IEEE Computer Graphics and Applications*, 18(4):18–23, 1998. doi:10.1109/38.689657.

31 Natalya F Noy, Deborah L McGuinness, et al. Ontology Development 101: A Guide to Creating Your First Ontology, 2001.

32 David Stephen John Perrin. PROMPT-Viz: Ontology Version Comparison Visualizations with Treemaps. Master's thesis, University of Victoria, 2004.

33 Jun Rekimoto and Mark Green. The Information Cube: Using Transparency in 3D Information Visualization. In *Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, volume 13, pages 125–132, 1993.

34 Isabel Cristina Siqueira Silva, Carla Maria Dal Sasso Freitas, and Giuseppe Santucci. An Integrated Approach for Evaluating the Visualization of Intensional and Extensional Levels of Ontologies. In *Proceedings of the 2012 BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, BELIV '12, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2442576.2442578.

35 Isabel Cristina Siqueira Silva, Giuseppe Santucci, and Carla Maria Dal Sasso Freitas. Visualization and analysis of schema and instances of ontologies for improving user tasks and knowledge discovery. *Journal of Computer Languages*, 51:28–47, 2019. doi:10.1016/j.cola.2019.01.004.

36 H.S. Smallman, M. St. John, H.M. Oonk, and M.B. Cowen. Information availability in 2D and 3D displays. *IEEE Computer Graphics and Applications*, 21(5):51–57, 2001. doi:10.1109/38.946631.

37 Ramanathan Somasundaram. *OntoSELF: A 3D ontology visualization tool.* PhD thesis, Miami University, 2007.

38 Tatiana Von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J van Wijk, J-D Fekete, and Dieter W Fellner. Visual analysis of large graphs: state-of-the-art and future research challenges. In *Computer graphics forum*, volume 30, pages 1719–1749. Wiley Online Library, 2011. doi:10.1111/J.1467-8659.2011.01898.X.

39 Yunhai Wang, Yanyan Wang, Haifeng Zhang, Yinqi Sun, Chi-Wing Fu, Michael Sedlmair, Baoquan Chen, and Oliver Deussen. Structure-aware Fisheye Views for Efficient Large Graph Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):566–575, 2019. doi:10.1109/TVCG.2018.2864911.

40 Colin Ware. *Information Visualization: Perception for Design*, chapter 1,7. Morgan Kaufmann, 2013.

41 Ying Yang, Michael Wybrow, Yuan-Fang Li, Tobias Czauderna, and Yongqun He. OntoPlot: A Novel Visualisation for Non-hierarchical Associations in Large Ontologies. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1140–1150, 2019. doi:10.1109/TVCG.2019.2934557.