

Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Fachstudie Nr. 205

Comparison and Evaluation of Ontology Visualizations

Lukas Balzer, Minh Thu Do, Dennis Maseluk

Course of Study: Software Engineering

Examiner: Prof. Dr. Thomas Ertl

Supervisor: Dr. Steffen Lohmann

Commenced: 2014-07-10

Completed: 2015-01-09

CR-Classification: H.5.2, I.2.4

Abstract

In the last couple of years a large number of software aiming at visualizing ontologies were introduced, causing difficulties in selecting an objectively suitable visualizing tool. This work lists free and commercial tools in four groups named by the visualizing method used by members of each group. Furthermore a detailed examination of each tool in a separate section is presented. Finally the document comes to an end with an overview of all listed tools and a recommendation of the most powerful ones of each group.

Contents

1	Introduction	12
2	Hierarchical Visualization	15
2.1	OntoTrack	15
2.1.1	Installation	15
2.1.2	Implementation details	15
2.1.3	Description	15
2.1.4	Layout options	16
2.1.5	License	16
2.1.6	Screenshot	17
2.1.7	Evaluation	17
2.2	KC-Viz	18
2.2.1	Installation	18
2.2.2	Implementation details	18
2.2.3	Description	18
2.2.4	Layout options	18
2.2.5	License	19
2.2.6	Screenshot	19
2.2.7	Evaluation	20
2.3	CropCircles	21
2.3.1	Installation	21
2.3.2	Implementation details	21
2.3.3	Description	21
2.3.4	Layout options	22
2.3.5	License	22
2.3.6	Screenshot	22
2.3.7	Evaluation	23
2.4	OWLviz	24
2.4.1	Installation	24
2.4.2	Implementation details	24
2.4.3	Description	24
2.4.4	Layout options	24
2.4.5	License	25
2.4.6	Screenshot	25
2.4.7	Evaluation	27
2.5	Knoocks (Knowledge Blocks)	28
2.5.1	Installation	28

2.5.2	Implementation details	28
2.5.3	Description	28
2.5.4	Layout options	28
2.5.5	License	28
2.5.6	Screenshot	29
2.5.7	Evaluation	30
3	Graph Visualization	31
3.1	Ontology Visualizer	31
3.1.1	Installation	31
3.1.2	Implementation details	31
3.1.3	Description	31
3.1.4	Layout options	32
3.1.5	License	32
3.1.6	Screenshot	33
3.1.7	Evaluation	34
3.2	OntoGraf	35
3.2.1	Installation	35
3.2.2	Implementation details	35
3.2.3	Description	35
3.2.4	Layout options	35
3.2.5	License	36
3.2.6	Screenshot	36
3.2.7	Evaluation	38
3.3	OntoSphere 3D	39
3.3.1	Installation	39
3.3.2	Implementation details	39
3.3.3	Description	39
3.3.4	Layout options	39
3.3.5	License	40
3.3.6	Screenshot	40
3.3.7	Evaluation	42
3.4	Visual Ontology Modeler	43
3.4.1	Installation	43
3.4.2	Implementation details	43
3.4.3	Description	43
3.4.4	Layout options	43
3.4.5	License	43
3.4.6	Screenshot	44
3.4.7	Evaluation	44
3.5	OWLPropViz	45
3.5.1	Installation	45
3.5.2	Implementation details	45
3.5.3	Description	45
3.5.4	Layout Options	45

3.5.5	License	46
3.5.6	Screenshot	46
3.5.7	Evaluation	48
3.6	OWLGrEd	49
3.6.1	Installation	49
3.6.2	Implementation Details	49
3.6.3	Description	49
3.6.4	Layout Options	49
3.6.5	License	50
3.6.6	Screenshot	50
3.6.7	Evaluation	52
3.7	TGViz	53
3.7.1	Installation	53
3.7.2	Implementation details	53
3.7.3	Description	53
3.7.4	Layout options	53
3.7.5	License	54
3.7.6	Screenshot	54
3.7.7	Evaluation	55
3.8	OntoRama	56
3.8.1	Installation	56
3.8.2	Implementation details	56
3.8.3	Description	56
3.8.4	Layout options	57
3.8.5	License	57
3.8.6	Screenshot	58
3.8.7	Evaluation	59
3.9	OntoTrix	60
3.9.1	Installation	60
3.9.2	Implementation details	60
3.9.3	Description	60
3.9.4	Layout options	61
3.9.5	License	61
3.9.6	Screenshot	61
3.9.7	Evaluation	62
3.10	VOWL	63
3.10.1	Installation	63
3.10.2	Implementation details	63
3.10.3	Description	63
3.10.4	Layout Options	64
3.10.5	License	64
3.10.6	Screenshot	65
3.10.7	Evaluation	67
3.11	NavigOwl	68
3.11.1	Installation	68

3.11.2	Implementation details	68
3.11.3	Description	68
3.11.4	Layout options	68
3.11.5	License	68
3.11.6	Screenshots	70
3.11.7	Evaluation	71
3.12	GrOWL	72
3.12.1	Installation	72
3.12.2	Implementation details	72
3.12.3	Description	72
3.12.4	Layout options	72
3.12.5	License	72
3.12.6	Screenshots	73
3.12.7	Evaluation	74
3.13	COE (Concept-map Ontology Environment)	75
3.13.1	Installation	75
3.13.2	Implementation details	75
3.13.3	Description	75
3.13.4	Layout options	75
3.13.5	License	76
3.13.6	Screenshot	76
3.13.7	Evaluation	78
4	Universal Visualization	79
4.1	GLOW	79
4.1.1	Installation	79
4.1.2	Implementation details	79
4.1.3	Description	79
4.1.4	Layout options	80
4.1.5	License	80
4.1.6	Screenshot	81
4.1.7	Evaluation	83
4.2	SOVA	84
4.2.1	Installation	84
4.2.2	Implementation details	84
4.2.3	Description	84
4.2.4	Layout options	84
4.2.5	License	85
4.2.6	Screenshot	86
4.2.7	Evaluation	89
4.3	TopBraid Composer	90
4.3.1	Installation	90
4.3.2	Implementation details	90
4.3.3	Description	90
4.3.4	Layout options	90

4.3.5	License	90
4.3.6	Screenshot	91
4.3.7	Evaluation	92
4.4	Jambalaya	93
4.4.1	Installation	93
4.4.2	Implementation details	93
4.4.3	Description	93
4.4.4	Layout options	93
4.4.5	License	94
4.4.6	Screenshot	94
4.4.7	Evaluation	96
5	Related Works	97
5.1	FlexViz	97
5.1.1	Installation	97
5.1.2	Implementation details	97
5.1.3	Description	97
5.1.4	License	97
5.1.5	Screenshot	98
5.1.6	Evaluation	99
5.2	RelFinder	100
5.2.1	Installation	100
5.2.2	Implementation details	100
5.2.3	Description	100
5.2.4	Layout options	100
5.2.5	License	100
5.2.6	Screenshot	101
5.2.7	Evaluation	102
6	Overview	103
6.1	Evaluation	104
6.2	Overview of visualization capability	105
6.3	Tool information	106
6.4	Recommendations	107
	Bibliography	109

List of Figures

2.1	OntoTrack application window [LN03]	17
2.2	Visualization of the Pizza ontology [DHS ⁺ 07] with KC-Viz 1.3.1.2 in the NeOn toolkit [Har14] 2.5.1	19
2.3	Visualization of the Pizza ontology [DHS ⁺ 07] using the CropCircles feature in SWOOP [swo14] 2.3 beta4	22
2.4	Part of the visualization of the Pizza ontology [DHS ⁺ 07] in OWLViz 4.1.4 for <i>Protégé</i> 5.0.0	25
2.5	Visualization of the Pizza ontology [DHS ⁺ 07] with OWLViz 4.1.4 in <i>Protégé</i> 5.0.0	26
2.6	Visualization of the Travel ontology with Knoocks [KW10] [kno15] 4.0	29
3.1	Visualization of the Pizza ontology [DHS ⁺ 07] with Ontolgooy Visualizer in the NeOn toolkit [Har14] 2.5.1	33
3.2	Visualization of the Pizza ontology [DHS ⁺ 07] with OntoGraf 1.0.3 in <i>Protégé</i> 5.0.0 (radial layout)	36
3.3	Visualization of the Pizza ontology [DHS ⁺ 07] with OntoGraf 1.0.3 in <i>Protégé</i> 5.0.0 (tree horizontal layout)	37
3.4	Visualization of the Pizza ontology [DHS ⁺ 07] with OntoSphere 3D 1.0.0 in Eclipse Luna 4.4.0 (main scene)	40
3.5	Visualization of the Pizza ontology [DHS ⁺ 07] with OntoSphere 3D 1.0.0 in Eclipse Luna 4.4.0 (exploring taxonomy branches)	41
3.6	Visualization of the Pizza ontology [DHS ⁺ 07] with OntoSphere 3D 1.0.0 in Eclipse Luna 4.4.0 (detailing on a concept)	41
3.7	Visualization of the Azalea plant with Visual Ontology Modeler [vom13] (Screenshot from the homepage representation video)	44
3.8	Part of the visualization of the Pizza ontology [DHS ⁺ 07] with OWLPropViz 1.0 in <i>Protégé</i> 4.0	46
3.9	Visualization of the Pizza ontology [DHS ⁺ 07] with OWLPropViz 1.0 in <i>Protégé</i> 4.0	47
3.10	Visualization of the Pizza ontology [DHS ⁺ 07] with OWLGrEd [owl14] 1.6.0 (desktop application)	50
3.11	Part of the visualization of the Pizza ontology [DHS ⁺ 07] with OWLGrEd [owl14] 1.6.0 (desktop application)	51
3.12	Visualization of the Pizza ontology [DHS ⁺ 07] with TGViz 1.4 in <i>Protégé</i> 3.0 (spring layout)	54
3.13	Visualization of the Pizza ontology with OntoRamaApp [ERG02]	58
3.14	Visualization of the Pizza ontology [DHS ⁺ 07] with <i>ProtégéVOWL</i> [NLB14] 0.1.3	65

3.15	Visualization of the FOAF (Friend of a Friend) ontology with <i>ProtégéVOWL</i> [NLB14] 0.1.3	66
3.16	Visualization of the FOAF (Friend of a Friend) ontology with WebVOWL (beta 0.3.0) [LLMN14]	66
3.17	Visualization of the Pizza ontology with NavigOwl 1.1.0 in <i>Protégé-OWL</i> 4.1 (random layout)	69
3.18	Visualization of the Pizza ontology with NavigOwl 1.1.0 in <i>Protégé-OWL</i> 4.1 (power layout)	70
3.19	Visualization of the Pizza ontology with GrOWL	73
3.20	Part of the visualization of the Pizza ontology [DHS ⁺ 07] with COE 5.0 [coe15] [HEM ⁺ 05]	76
3.21	Visualization of the Pizza ontology [DHS ⁺ 07] with COE 5.0 [coe15] [HEM ⁺ 05]	77
4.1	Visualization of the Pizza ontology [DHS ⁺ 07] with GLOW [Glo15] [HRFH12] 1.0 in <i>Protégé</i> 4.0.2 (inverted radial tree)	81
4.2	Visualization of the Pizza ontology [DHS ⁺ 07] with GLOW [Glo15] [HRFH12] 1.0 in <i>Protégé</i> 4.0.2 (force-directed graph)	82
4.3	Visualization of the People ontology [peo14] with SOVA 0.8.4 in <i>Protégé</i> 4.3 (force directed layout)	86
4.4	Visualization of the People ontology [peo14] with SOVA 0.8.4 in <i>Protégé</i> 4.3 (radial tree layout)	87
4.5	Visualization of the People ontology [peo14] with SOVA 0.8.4 in <i>Protégé</i> 4.3 (hierarchy tree)	88
4.6	Visualization of the Pizza ontology [DHS ⁺ 07] with TopBraid Composer Maestro Edition [top14] 4.5.0 as graph	91
4.7	Visualization of the Pizza ontology [DHS ⁺ 07] with TopBraid Composer Maestro Edition [top14] 4.5.0 as diagram	91
4.8	Visualization of the Pizza ontology [DHS ⁺ 07] with Jambalaya 2.7.0 in <i>Protégé</i> 3.5 (nested treemap layout)	94
4.9	Visualization of the Pizza ontology [DHS ⁺ 07] with Jambalaya 2.7.0 in <i>Protégé</i> 3.5 (class tree layout)	95
4.10	Visualization of the Pizza ontology [DHS ⁺ 07] with Jambalaya 2.7.0 in <i>Protégé</i> 3.5 (domain/range layout)	95
5.1	The GraphTest application which is contained in the downloadable source of FlexViz [fle15] 2.3.0 (random layout)	98
5.2	Visualization with RelFinder [rel14] 1.3.6	101

List of Tables

2.1	Evaluation of KC-Viz 1.3.1.2	20
2.2	Visualization capabilities of KC-Viz 1.3.1.2	20
2.3	Evaluation of CropCircles	23
2.4	Visualization capabilities of CropCircles	23
2.5	Evaluation of OWLViz 4.1.4	27
2.6	Visualization capabilities of OWLViz 4.1.4	27
2.7	Evaluation of Knoocks 4.0	30
2.8	Visualization capabilities of Knoocks 4.0	30
3.1	Evaluation of Ontology Visualizer	34
3.2	Visualization capabilities of Ontology Visualizer	34
3.3	Evaluation of OntoGraf 1.0.3	38
3.4	Visualization capabilities of OntoGraf 1.0.3	38
3.5	Evaluation of Ontosphere3D 1.0.0	42
3.6	Visualization capabilities of Ontosphere3D 1.0.0	42
3.7	Evaluation of OWLPropViz 1.0	48
3.8	Visualization capabilities of OWLPropViz 1.0	48
3.9	Evaluation of OWLGrEd 1.6.0	52
3.10	Visualization capabilities of OWLGrEd 1.6.0	52
3.11	Evaluation of TGViz 1.4	55
3.12	Visualization capabilities of TGViz 1.4	55
3.13	Evaluation of OntoRama	59
3.14	Visualization capabilities of OntoRama	59
3.15	Evaluation of <i>ProtégéVOWL</i> 0.1.3 and WebVOWL 0.3.0	67
3.16	Visualization capabilities of <i>ProtégéVOWL</i> 0.1.3 and WebVOWL 0.3.0	67
3.17	Evaluation of NavigOwl 1.1.0	71
3.18	Visualization capabilities of NavigOwl 1.1.0	71
3.19	Evaluation of GrOWL	74
3.20	Visualization capabilities of GrOWL	74
3.21	Evaluation of COE 5.0	78
3.22	Visualization capabilities of COE 5.0	78
4.1	Evaluation of GLOW 1.0	83
4.2	Visualization capabilities of GLOW 1.0	83
4.3	Evaluation of SOVA 0.8.4	89
4.4	Visualization capabilities of SOVA 0.8.4	89
4.5	Evaluation of TopBraid Composer Maestro Edition 4.5.0	92

4.6	Visualization capabilities of TopBraid Composer Maestro Edition 4.5.0	92
4.7	Evaluation of Jambalaya 2.7.0	96
4.8	Visualization capabilities of Jambalaya 2.7.0	96
5.1	Evaluation of FlexViz 2.3.0	99
5.2	Visualization capabilities of FlexViz 2.3.0	99
5.3	Evaluation of RelFinder 1.3.6	102
5.4	Visualization capabilities of RelFinder 1.3.6	102
6.1	Evaluation of all tested tools.	104
6.2	Visualization capabilities of all tested tools.	105
6.3	Information about examined tools.	106
6.4	Recommendations for hierarchical visualizations.	107
6.5	Recommendations for graph visualizations.	108
6.6	Recommendations for universal visualizations.	108

1 Introduction

Using ontologies to store information has become a more and more spread practice on the World Wide Web, because ontologies are capable of retaining a large amount of information in a machine-readable form and in a more and more easily accessible way [ReO]. Although many approaches towards the visualization of ontologies were introduced, new representational methods are under development or even released, ontology visualization continues to be a nontrivial issue. Most of the available visualization tools are seen to be incomplete, as they are not able to visualize all the constructs of a large ontology.

However, completeness is only one from the set of the parameters used in visualization tool assessment. As a result, an incomplete tool is not necessarily unhelpful, for example when a quick overview of a specific ontology is required and further details should be ignored.

This paper aims at introducing available visualization tools with an overview on their features and capabilities, which can serve as guidance to choose the most suitable tool depending on specific requirements. The remainder of this paper is organized as follows:

Chapter 2 – Hierarchical Visualization: This chapter examines those tools focusing on the inheritance relations of an ontology and visualizing it as a special graph namely hierarchical tree.

Chapter 3 – Graph Visualization This chapter introduces those tools focusing not only on the inheritance relations but also on the properties and other relationships among the classes of an ontology. Therefore they can visualize a given ontology as a graph and in some cases additionally as a tree.

Chapter 4 – Universal Visualization This chapter examines those tools providing a list of visualization layouts of a given ontology, from which the user can choose one at a time.

Chapter 5 – Related Works This chapter introduces two tools which can visualize different types of data format including OWL/RDF data.

Chapter 6 – Overview This chapter provides a list of tables, each contains all the visualization tools mentioned in chapter 2 to chapter 5 together with their features or capabilities. Recommendations are likewise included.

Following is the structure used to describe each visualization tool:

1. Installation

A short description of installation procedure, e.g. which programs and libraries are required, which steps need to be conducted in order to install the tool.

2. **Implementation details**

A list of details like what programming language or libraries are used to develop the tool.

3. **Description**

A description of the available tool.

4. **Layout options**

A list of notable features available in viewing, navigating or editing an visualized ontology (e.g. multi-layout, zoom, panning etc.).

5. **License**

The license agreements required to use the tool.

6. **Screenshot**

An image of the Pizza ontology [DHS⁺07], an ontology provided by the University of Manchester. Where such an image is not available, a figure of respective application window will be found.

7. **Evaluation**

Comprising of two evaluation tables.

The first table lists seven criteria and assigns each criteria a ranking of green, yellow or red color, which means it is respectively of good, satisfactory or insufficient evaluation. The seven criteria used are listed below.

- **Visualization capability** is the ability to visualize elements of an ontology. If a tool can display up to four constructs, its visualization capability is marked with red color. If it can display five to nine constructs, it has yellow capability. Green color is for the tools that can display between ten and fourteen constructs.
- **Look** presents the optical view of an visualization. A visualization is considered good and marked with green color when different types of constructs have different colors or shapes, and the image looks good. A visualization is of red color if its components are of high density, therefore gives no possibility to understand the visualized ontology. However, it is quite subjective to give a specific color to the look.
- **Usability** represents how easy it is to use a tool. It is again subjective to mark the usability with a specific color.
- **Reliability** tells if a program is robust. A program does not receive green color if it is not always successful in loading a file or suddenly crashes.
- **Documentation** tells if the documentation of a tool is available, accurate and in detail.
- **Installation effort** tells how difficult and time-consuming to install a tool. The color assignment depends additionally on the number of attempts to successfully install a program.

- **Overall impression** of the tool is determined based on six above-mentioned criteria.

The second table displays visualization capabilities of the tools. In order to test the visualization capabilities of each tool, we used test files which are either created by ourselves or downloaded from the W3C website [CDR04]. Each test file contains a single construct of an ontology. It is then imported into a given tool and executed by the tool. A test case is successful when the construct is displayed by the given tool. If a tool fails to load these files, we use other ontologies like the FOAF (Friend of a friend) ontology or the Pizza ontology [DHS⁺07].

2 Hierarchical Visualization

This chapter introduces those tools which mainly focus on the basic hierarchical structure of an ontology and therefore depict merely its "is-a" relations, classes and sometimes its individuals. A quick overview over classes of given ontology is provided, at the same time irrelevant or uninterested details are ignored. As a result, visualization of respective ontology looks simple, users can therefore easily understand its structure.

However, much information of the given ontology would not be displayed, and no complete visualization is available.

2.1 OntoTrack

2.1.1 Installation

The software OntoTrack is no longer available. All information about this tool was collected from the article "OntoTrack: A semantic approach for ontology authoring" [LN03].

2.1.2 Implementation details

Programming language Java, CommonLisp

Libraries Picolo

Design Model-View-Controller

Dependencies RACER reasoner, Jena 2.1 RDF API

2.1.3 Description

OntoTrack is a tool for OWL Lite, a sensible fraction of OWL. It combines graph-based hierarchical layout and instant reasoning feedback within one single view.

An ontology is visualized as directed acyclic graph whose nodes are classes or properties of the ontology. Subsumption relationships between classes and properties are directed edges with top-down, left-right, bottom-up or right-left orientation.

Classes with multiple ancestors summed up in collapsed branches alternatively. Users can expand/collapse sub-branches with a mouse click. Being in collapsed state the sub-branches

are represented either by little triangles or by miniature graphs. The little triangles are of varying length, width and shading. A miniature graph is a topological structure of descendants (number of descendants shown is a user definable threshold). It is possible to view triangle and miniature sub-branches by hovering the mouse over corresponding nodes. Hereby the tool prevents to depict too many classes at the same time, which is extremely helpful since it permits to view the whole ontology without losing the overview. Due to this feature and the usage of tooltips for explanation of subsumption between classes, OntoTrack is suitable for large ontologies.

To enhance the usability of the tool OntoTrack also provides a bird-eye view which appears in upper left corner if ontology exceeds the available rendering area. It is a miniaturized graph of the current ontology expansion and an overlaid movable rectangle representing the current rendering area.

OntoTrack is an easy plug-in of reasoner system or non-standard reasoning services. It is possible to edit an ontology with OntoTrack

2.1.4 Layout options

- Directed acyclic graph
- Visualization of collapsed parts with little triangles or miniature graphs
- Zooming and panning
- Expand/collapse of sub-branches by mouse click
- Bird-eye view
- String matching search with graphical highlighting (thumbnails or triangle sub-branches are highlighted)

2.1.5 License

OntoTrack - non-commercial use license
RACER - BSD 3-clause license [bsd99]

2.1.6 Screenshot

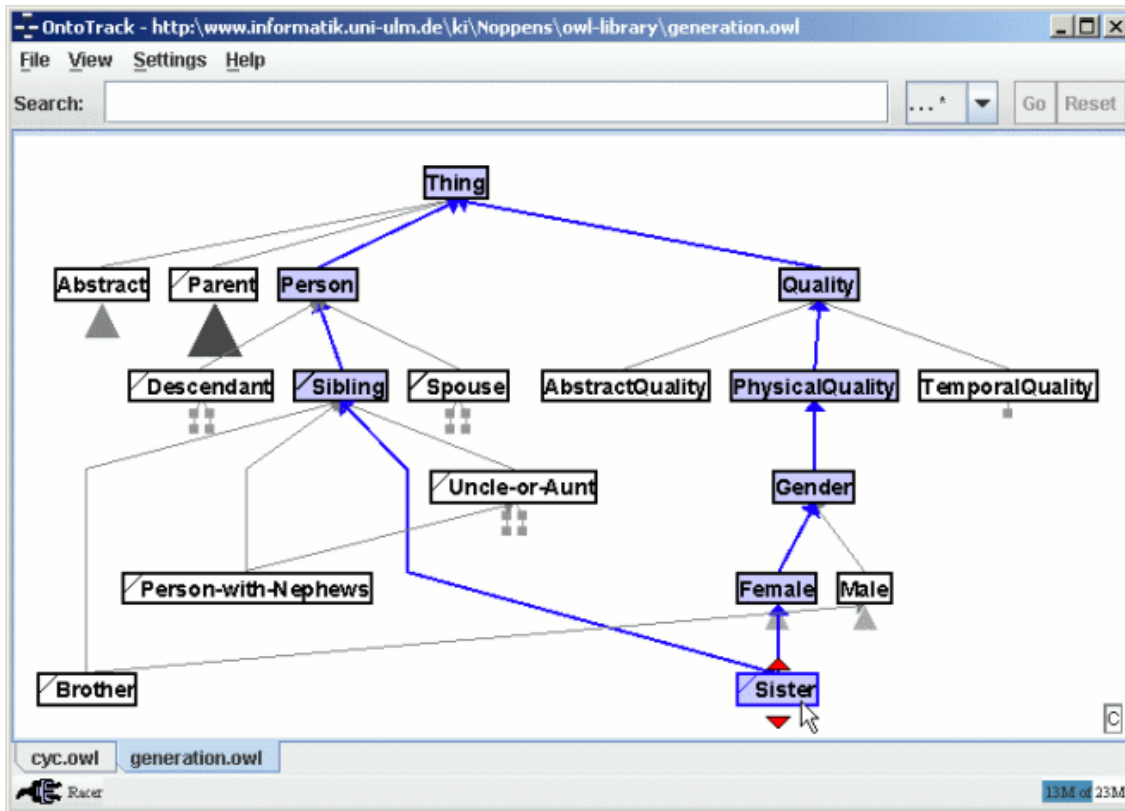


Figure 2.1: OntoTrack application window [LN03]

2.1.7 Evaluation

Since OntoTrack is no longer available, the tool is not tested.

2.2 KC-Viz

2.2.1 Installation

This tool is provided as a plug-in for the eclipse-based NeOn Toolkit [Har14] till version 2.3.*. In version 2.4 of the NeOn Toolkit the KC-Viz [PM14] [MPGP⁺12] plug-in version 1.3.1.2 is integrated in the core installation.

2.2.2 Implementation details

Programming language Java

2.2.3 Description

KC-Viz is a tool for visualizing the key concepts of an ontology. It is therefore used to visualize the classes together with their hierarchical order and domain-range relations, which depict an object property between a domain and a range class.

An ontology is hereby shown as heap of sub- and super-classes which are linked with sub-/superClassOf and domain/range edges, the former are depicted as solid-grey- and the latter as red-dashed-arrows.

Additionally KC-Viz contains a „key concept extraction algorithm“ [MPGP⁺12] which decides depending on configurable preferences how important a class is and whether it is a key concept or not. Due to this rating KC-Viz can then produce a visualization of only the most important key concepts. Each class is represented by a blue circle whose size determines the importance of the class. The class circle owns a label showing its title and two numbers from which the first is telling the number of direct subclasses and the second the size of its sub tree.

It is also possible to hide or expand classes, which helps the user to navigate through the hierarchy with just a few mouse clicks. Each of the expand and hide feature provides a dialog where the user can choose the inflicted classes. In the expand feature the user can additionally choose the depth and proceeding concerning other classes . As result, it is possible to hide long inheritance paths which are shown as green-dashed-arrows. Its content is given as tooltip and is not visible in the visualization.

2.2.4 Layout options

- Left-right and top-down tree layout
- Normal zoom, x-/y-zoom
- Mouse panning
- Supporting undo/redo separated in micro/macro for better distinction of handling drag operations/expanding and hiding of nodes

- Providing a marquee tool
- Supports marquee by double-clicking the view
- „key concept extraction algorithm“

2.2.5 License

KC-Viz - Eclipse Public License - v 1.0 [epf]

2.2.6 Screenshot

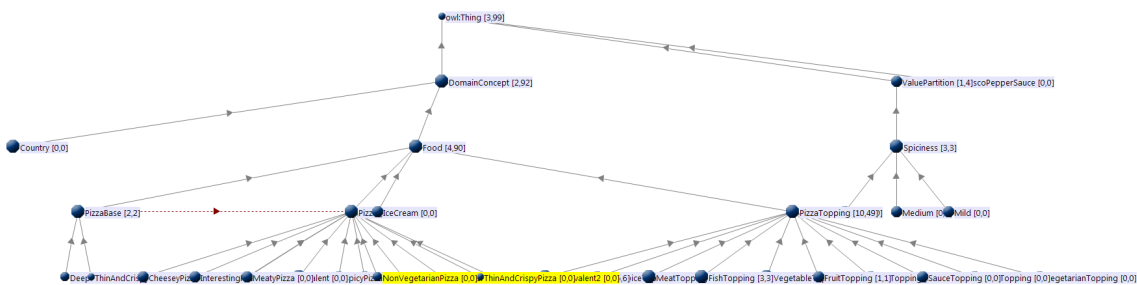


Figure 2.2: Visualization of the Pizza ontology [DHS⁺07] with KC-Viz 1.3.1.2 in the NeOn toolkit [Har14] 2.5.1

2.2.7 Evaluation

Table 2.1: Evaluation of KC-Viz 1.3.1.2

KC-Viz	●	●	●	●	●	●	●
	Visualization capability	Look	Usability	Reliability	Documentation	Installation effort	Overall impression

Table 2.2: Visualization capabilities of KC-Viz 1.3.1.2

KC-Viz	●						●				●			
	Classes	Enumeration	Property restrictions	Cardinality	Intersection	Union	Complement	subClassOf	equivalentClass	disjointWith	Object properties	Datatype properties	Instances	Annotations

2.3 CropCircles

2.3.1 Installation

CropCircles is downloadable as part of the SWOOP Platform [swo14].

How to execute CropCircles:

1. Download the .zip with the Swoop Platform from the download section of the google code resource¹
2. Execute the *runme.bat*
3. Load/Create an OWL/RDF ontology of your choice
4. Go to *Advanced* → *FlytheMotherShip* to open a new window with the CropCircles visualization

2.3.2 Implementation details

Programming language Java

2.3.3 Description

CropCircles [WP06] depicts merely the classes of an ontology. Each class is hereby layouted as a circle inside its super class circle and the whole ontology is limited by a green circle. In that way CropCircles provides a simple and compact representation of a hierarchy. Hierarchical structure of the circles on its turn automatically provides quick information about the depth and structure of the ontology. This also means that one ontology has just one representation, so the circular tree map is initialized once with the whole structure and subsequently can not be modified. The name labels are put in tooltips to get a compact layout and therefore are not visible by default.

CropCircles is a nice approach to quickly get a compact image of an ontology although it does not contain any visualized text.

¹<https://code.google.com/p/swoop/>

2.3.4 Layout options

- Zoom support (buttons and mouse-wheel)
- Focus inner circle by doubleclick
- Circular Tree-map layout
- Panning
- Very compact visualization

2.3.5 License

Swoop - MIT Licence [mit88]

2.3.6 Screenshot

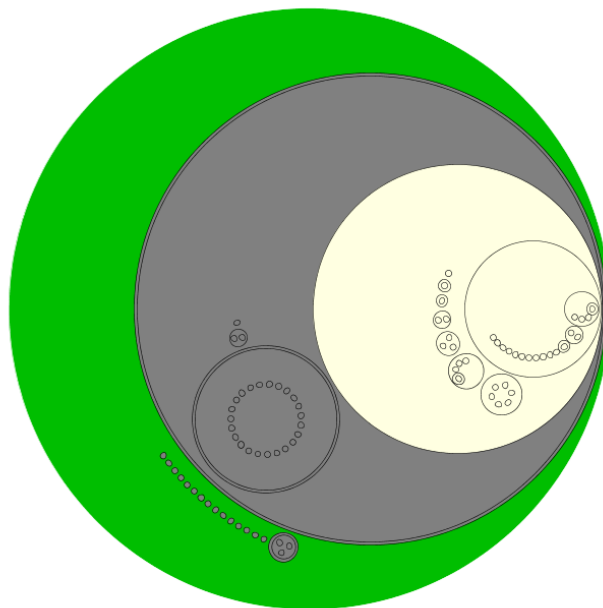


Figure 2.3: Visualization of the Pizza ontology [DHS⁺07] using the CropCircles feature in SWOOP [swo14] 2.3 beta4

2.3.7 Evaluation

Table 2.3: Evaluation of CropCircles

CropCircles	●	●	●	●	●	●	●
	Visualization capability	Look	Usability	Reliability	Documentation	Installation effort	Overall impression

Table 2.4: Visualization capabilities of CropCircles

CropCircles	●						●							
	Classes	Enumeration	Property restrictions	Cardinality	Intersection	Union	Complement	subClassOf	equivalentClass	disjointWith	Object properties	Datatype properties	Instances	Annotations

2.4 OWLViz

2.4.1 Installation

OWLViz² [Hor05] is a plug-in for *Protégé* 3.4.4 - 5.0.0 and already integrated in the *Protégé* core installation. In order to use this plug-in, Graphviz [EN⁺04] a free and open source graph visualization software is additionally required. It should be noted that OWLViz² has not been under active development since 2009.

2.4.2 Implementation details

Programming language Java

Libraries Graphviz [EN⁺04]

2.4.3 Description

OWLViz² limited itself to representing the class hierarchy of ontologies. The OWLViz² visualization appears to be quite simple and modest due to the pale colors.

The tool can display either the whole or a part of an ontology hierarchy, which means that it always depicts all subclasses of one selected top-element. OWLViz² can visualize two different types of classes, the primitive classes (colored in yellow) and the equivalent classes (colored in orange) which are also included as subclasses. The graph is static and can not be modified by the user, for example it is not possible to move the classes.

It is possible to export a graph as .png or .jpeg file. However, it was not possible to export a correct scaled image.

OWLViz² is seen as a simple hierarchy-based visualizer, which can be used without reading any documentation. This is important because the homepage of OWLViz is not available.

2.4.4 Layout options

- Two layout directions (left to right, top to bottom)
- Zoom in/out
- Hide/Show classes

²<http://protegewiki.stanford.edu/wiki/OWLViz>

2.4.5 License

Graphviz - Eclipse Public License - v 1.0 [epl]

Protégé - Mozilla Public License 1.1. [moz]

OWLViz - Lesser GNU Public License [LGP91]

2.4.6 Screenshot

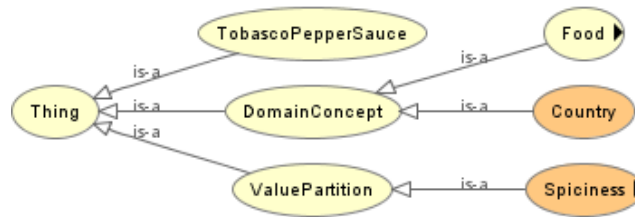


Figure 2.4: Part of the visualization of the Pizza ontology [DHS⁺07] in OWLViz 4.1.4 for Protégé 5.0.0

2 Hierarchical Visualization

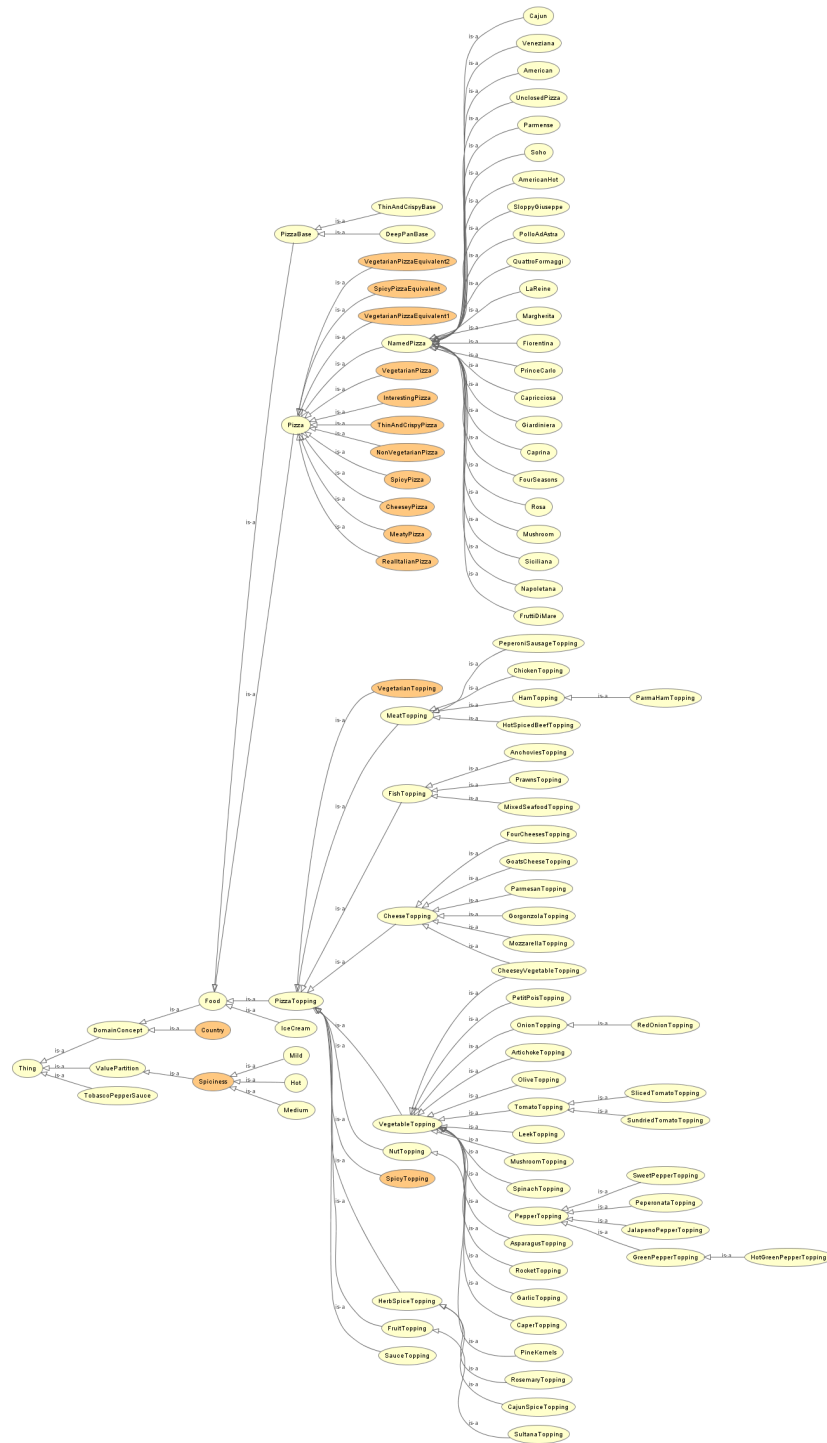


Figure 2.5: Visualization of the Pizza ontology [DHS⁺07] with OWLViz 4.1.4 in Protégé 5.0.0

2.4.7 Evaluation

Table 2.5: Evaluation of OWLViz 4.1.4

	Visualization capability
OWLViz	●
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression

Table 2.6: Visualization capabilities of OWLViz 4.1.4

OWLViz	●
	Classes
	Enumeration
	Property restrictions
	Cardinality
	Intersection
	Union
	Complement
	subClassOf
	equivalentClass
	disjointWith
	Object properties
	Datatype properties
	Instances
	Annotations

2.5 Knoocks (Knowledge Blocks)

2.5.1 Installation

This tool is still under development and not available for download at the time this document is released. It is however possible to request an evaluation copy of the prototype from the Website [kno15]. Knoocks [KW10] visualizes OWL Lite ontologies.

2.5.2 Implementation details

Programming language C#

Libraries OWLDotNetApi ³, OpenGL ⁴

2.5.3 Description

Knoocks visualizes an OWL Lite ontology as nested blocks where each block is depicted as a rectangle containing a subbranch shown as tree map. Knoocks works with the principle of looking on the hierarchical levels separately by providing two visualizations. The first one provides the overview of nested top-level-class blocks, each contains its own subclasses and are connected by their object properties. The second one provides detailed view of one class block, chosen in the overview, containing enumeration of its instances and available for the user to expand/collapse every classes or subclasses.

Object properties can be shown as instance to instance edge and as domain-range edge, which should be helpful in knowing how often a property is really referenced.

2.5.4 Layout options

- Object properties can be depicted by domain-range-edges or instance-instance-edge
- Hide option for each property in particular
- Different colors for different properties
- Layout: arrangements in nested blocks
- Switch between specific view and overview

2.5.5 License

Knoocks - no information

³<http://users.skynet.be/bpellens/OwlDotNetApi/index.html>

⁴<https://www.opengl.org/>

2.5.6 Screenshot

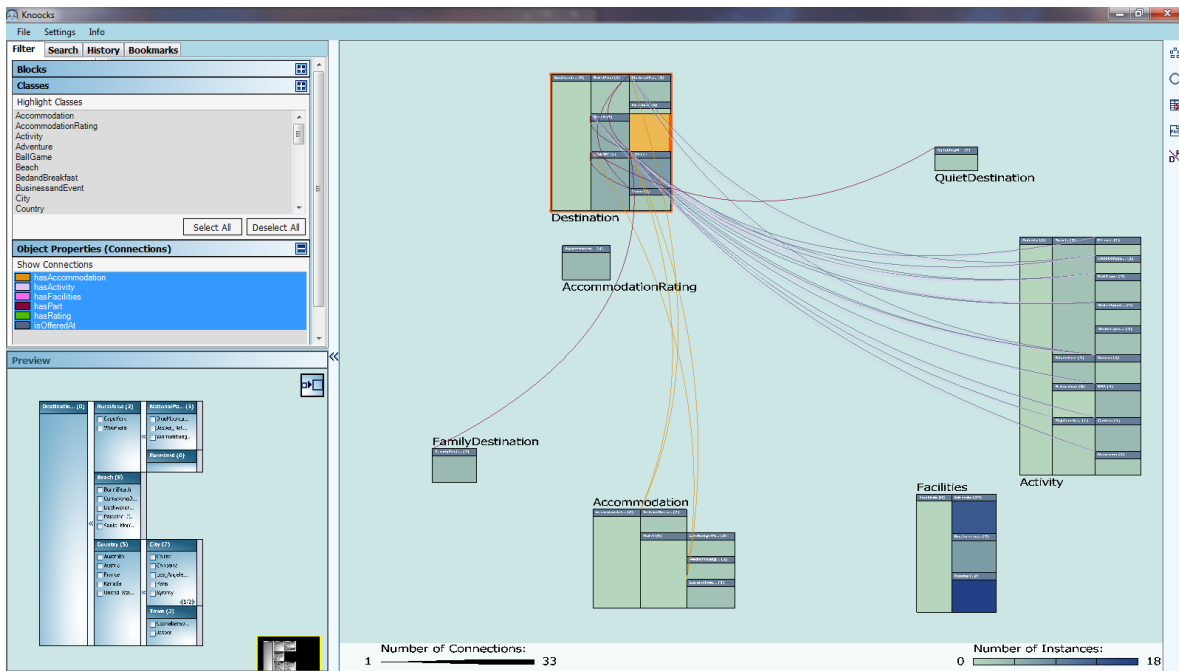


Figure 2.6: Visualization of the Travel ontology with Knoocks [KW10] [kno15] 4.0

2.5.7 Evaluation

Table 2.7: Evaluation of Knoocks 4.0

	Visualization capability
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression
Knoocks	● ● ● ● ● ● ●

Table 2.8: Visualization capabilities of Knoocks 4.0

	Classes
	Enumeration
	Property restrictions
	Cardinality
	Intersection
	Union
	Complement
	subClassOf
	equivalentClass
	disjointWith
	Object properties
	Datatype properties
	Instances
	Annotations
Knoocks	● ● ● ● ● ● ● ● ● ● ● ● ●

3 Graph Visualization

At first glance graph-based and hierarchy-based visualization of an ontology look much alike and seem to bring about the same information. It is actually not the situation. Graph visualization can be much more specific and accurate since there is no lower or upper limit of information amount represented in an ontology. As a result, it is now more difficult to choose an appropriate tool or plug-in that satisfies individual requirements.

3.1 Ontology Visualizer

3.1.1 Installation

The Ontology Visualizer is a plug-in for the NeOn toolkit [Har14] and since version 1.2.3. it is included as basic feature in the default NeOn toolkit installation.

3.1.2 Implementation details

Programming language Java

3.1.3 Description

The Ontology Visualizer displays classes, instances and relations of a given ontology. It is very user-friendly and requires no special skills. In order to use Ontology Visualizer, NeOn Toolkit must be available with an OWL ontology open, otherwise it will not work. The user then right clicks the class to be visualized and select Visualize class.->..in Visualizer. The Visualizer depicts three kinds of relations:

1. Sub/Superclass with grey arrows
2. Property values with green dashed connections
3. Individuals with blue connections

3 Graph Visualization

The structure is first expanded to a level which could be chosen under Window -> Preferences... Further classes can be expanded to the same level with the right mouse button. A left click on the other hand starts a new visualization with the selected class and the initial expansion level. The undo/redo function stores merely classes on which one clicks, so that it's not possible to undo an expand but only possible to go back to the last entry point. The automatic spring layout is not very good for big ontologies regarding degeneration. In other words, the layout is not meant to depict a huge lot of data at the same time.

3.1.4 Layout options

- Zoom with steps of 25 percent
- Spring layout
- Rotation by steps of 90 degrees
- Drag and drop
- Expansion of nodes
- History to go back
- Legend to hide/show individual/object properties

3.1.5 License

Ontology Visualizer - Eclipse Public License v 1.0 [epf]

3.1.6 Screenshot

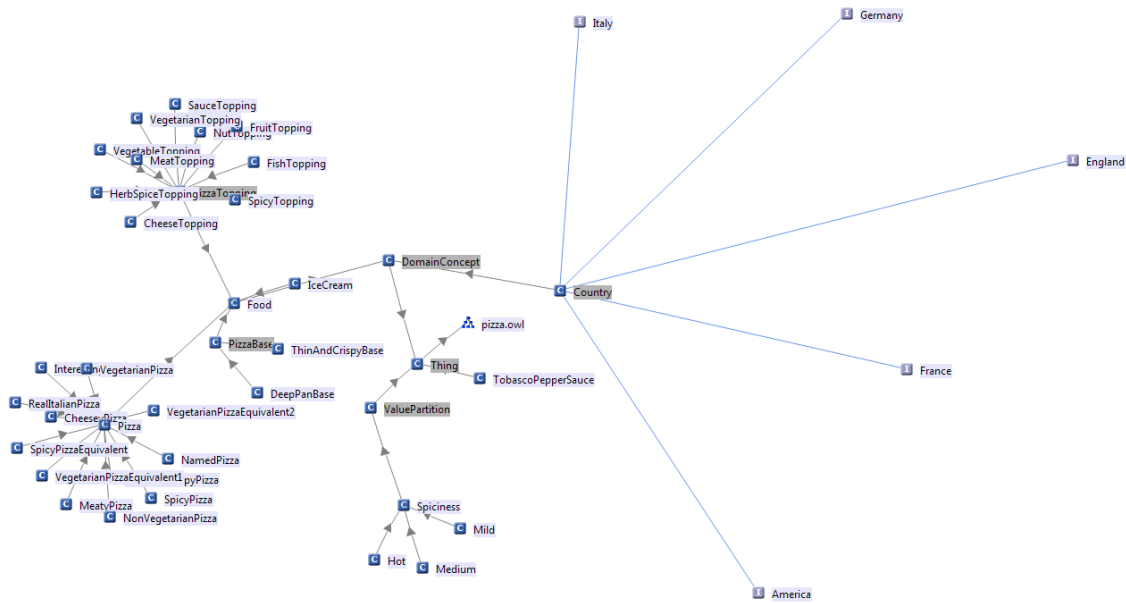


Figure 3.1: Visualization of the Pizza ontology [DHS⁺07] with Ontology Visualizer in the NeOn toolkit [Har14] 2.5.1

3.1.7 Evaluation

Table 3.1: Evaluation of Ontology Visualizer

	Visualization capability
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression
OntologyVisualizer	● ● ● ● ● ● ●

Table 3.2: Visualization capabilities of Ontology Visualizer

	Classes
	Enumeration
	Property restrictions
	Cardinality
	Intersection
	Union
	Complement
	subClassOf
	equivalentClass
	disjointWith
	Object properties
	Datatype properties
	Instances
	Annotations
OntologyVisualizer	●

3.2 OntoGraf

3.2.1 Installation

OntoGraf¹ [Fal10] is a plug-in for *Protégé* 4.1 - 5.0 and already integrated in the *Protégé* core installation.

3.2.2 Implementation details

Programming language Java

Libraries Piccolo²

3.2.3 Description

The plug-in can be intuitively handled. Its graphs look good due to little shading on classes and automatical organization of classes of given ontology. OntoGraf¹ visualizes different language constructs with different colored arrows and different types of arrows. Some language constructs like disjoint or annotations can not be visualized at the first level but only displayed in the tooltips of the classes.

OntoGraf¹ has a search bar with auto-completion function and different search options, e.g. contains, starts with, ends with, exact match, regexp. The user can modify the graph by moving classes via drag & drop or collapse/expand sub-branches with a double click.

The protegewiki entry of OntoGraf¹ is in detail and contains a link to a video demonstration. It is possible to export a visualized graph as a .png, .jpeg, .gif file. Furthermore there is an special feature of exporting a graph as a .dot file.

3.2.4 Layout options

- Zoom in/out
- Hide/Show classes
- Layout: radial, spring, tree vertical, tree horizontal, grid alphabetical
- Connection/Nodetype filter
- Pin node tooltips (shows information of the classes)

¹<http://protegewiki.stanford.edu/wiki/OntoGraf>

²<http://www.cs.umd.edu/hcil/piccolo/>

3 Graph Visualization

3.2.5 License

OntoGraf - no information

Protégé - Mozilla Public License 1.1. [moz]

3.2.6 Screenshot

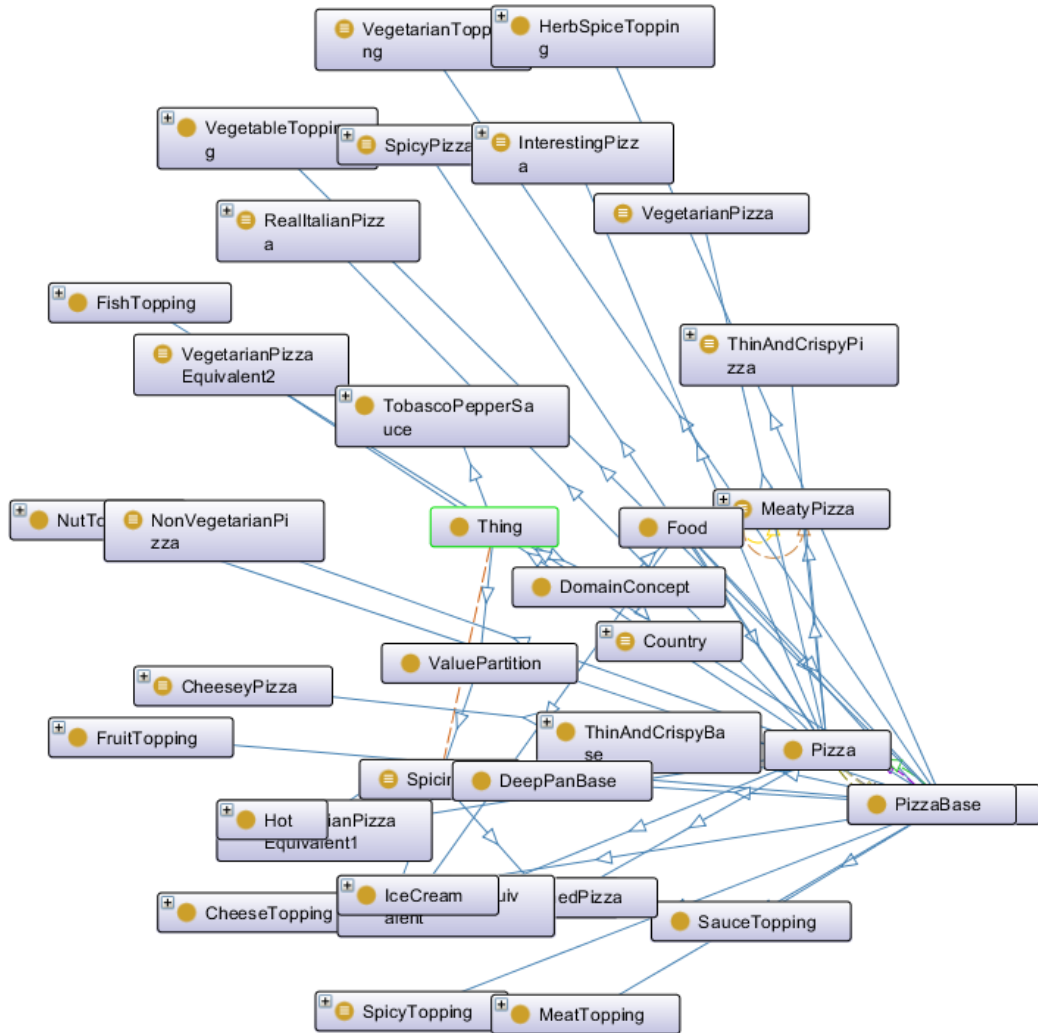


Figure 3.2: Visualization of the Pizza ontology [DHS⁺07] with OntoGraf 1.0.3 in Protégé 5.0.0 (radial layout)

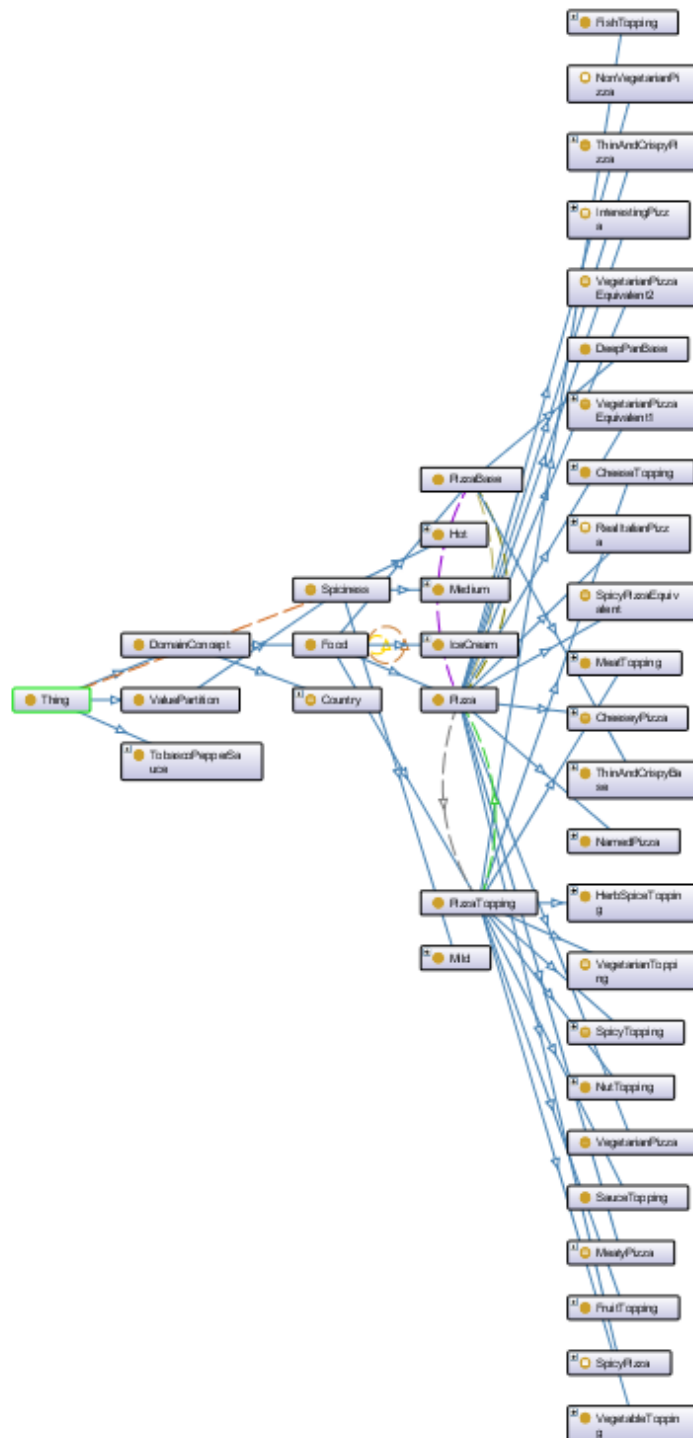


Figure 3.3: Visualization of the Pizza ontology [DHS⁺07] with OntoGraf 1.0.3 in *Protégé* 5.0.0 (tree horizontal layout)

3.2.7 Evaluation

Table 3.3: Evaluation of OntoGraf 1.0.3

	Visualization capability
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression
OntoGraf	● ● ● ● ● ● ●

Table 3.4: Visualization capabilities of OntoGraf 1.0.3

	Classes
	Enumeration
	Property restrictions
	Cardinality
	Intersection
	Union
	Complement
	subClassOf
	equivalentClass
	disjointWith
	Object properties
	Datatype properties
	Instances
	Annotations
OntoGraf	● ● ● ● ● ● ●

3.3 OntoSphere 3D

3.3.1 Installation

OntoSphere 3D³ [BB06] is a plug-in for *Protégé* 3.1.* or newer versions and Eclipse. In order to use OntoSphere 3D³ it is needed to have Java3D available on personal system.

3.3.2 Implementation details

Programming language Java

Libraries Java3D

3.3.3 Description

A graph of OntoSphere 3D³ composed of nodes and arrows. Each class of an ontology is referred to as a concept and OntoSphere 3D³ uses four different perspectives with different levels of detail to display classes.

The first perspective "main scene" shows all entities and their direct semantic relationships. These entities are linked with arrows. The "exploring taxonomy branches" perspective displays the subtree of a concept and some of its relations. The user can get more information about a concept in the "detailing on a concept" perspective. The last perspective "inspecting instances" displays all instances of a concept. It is possible to rotate all perspectives or click them with the mouse.

The visualized graphs have stair-like lines and therefore they look outdated. The animation of hiding and showing subtrees does not work fluently and consequently it is possible to lose the overview of the visualized ontology. Moreover there are some bugs which prohibit a clean layout process. OntoSphere 3D³ has therefore performance and thus usability problems.

The homepage of OntoSphere 3D³ is in detail and provides different information about the plug-in like architecture and installation guidelines. It is however not possible to run the plug-in in *Protégé*, so it is tested in Eclipse.

3.3.4 Layout options

- Zoom in/out with mouse wheel
- Hide/Show children, relations, only outgoing/incoming, datatype
- many options (radiuslimit, transparency, parentcolor, ...)

³<http://ontosphere3d.sourceforge.net/index.html>

3.3.7 Evaluation

Table 3.5: Evaluation of Ontosphere3D 1.0.0

	Visualization capability
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression
Ontosphere3D	● ● ● ● ● ● ●

Table 3.6: Visualization capabilities of Ontosphere3D 1.0.0

	Classes
	Enumeration
	Property restrictions
	Cardinality
	Intersection
	Union
	Complement
	subClassOf
	equivalentClass
	disjointWith
	Object properties
	Datatype properties
	Instances
	Annotations
Ontosphere3D	● ● ● ● ● ● ● ● ● ● ● ●

3.4 Visual Ontology Modeler

3.4.1 Installation

VOM (Visual Ontology Modeler) [vom13] is a commercial plug-in of MagicDraw [mag14] and therefore not tested.

3.4.2 Implementation details

Programming language Java

3.4.3 Description

The visualization of Visual Ontology Modeler is based on UML diagrams, which means each class and connection is visualized as an appropriate UML element. In addition it has a miniature screen of the whole ontology, which ensures a good overview on the ontology.

3.4.4 Layout options

- Drag and drop to create new elements
- Class and connection editing with the mouse in the visualization
- Zoom in/out
- Including hierarchically presentation and manual arrangement

3.4.5 License

MagicDraw - commercial license

VOM - commercial license

3 Graph Visualization

3.4.6 Screenshot

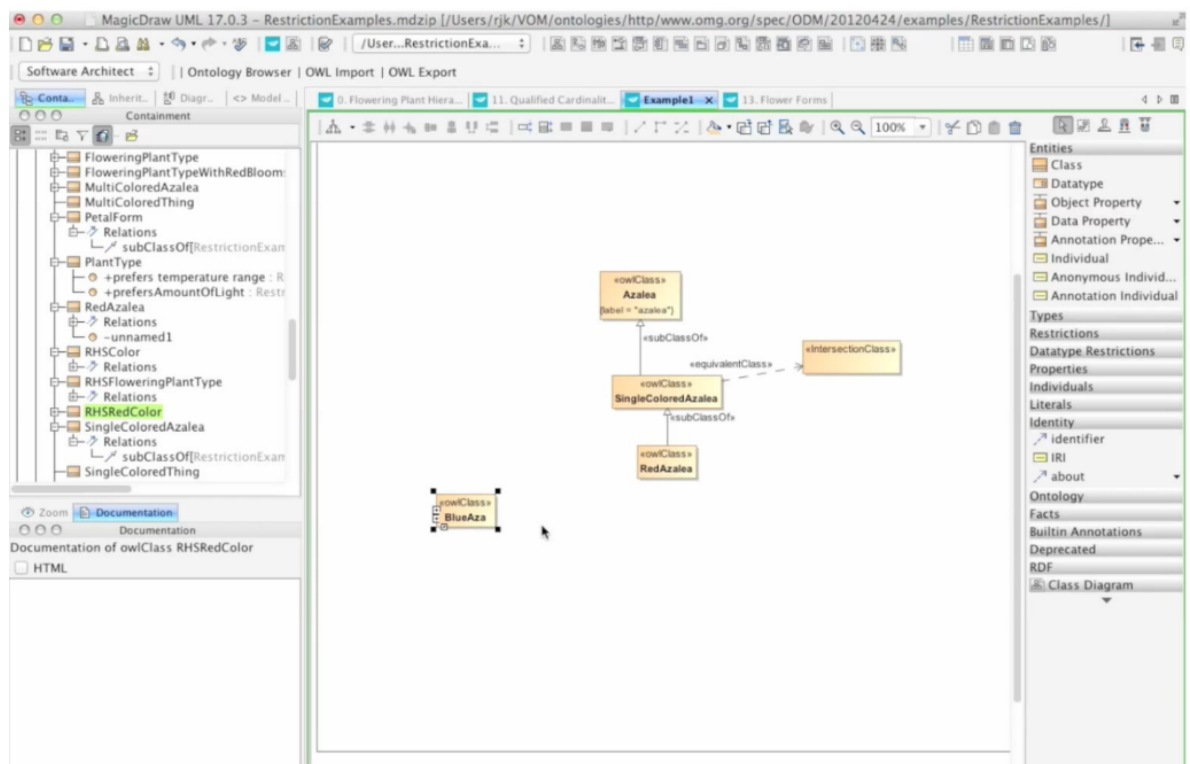


Figure 3.7: Visualization of the Azalea plant with Visual Ontology Modeler [vom13] (Screenshot from the homepage representation video)

3.4.7 Evaluation

Since Visual Ontology Modeler is a commercial plug-in, the plug-in is not tested.

3.5 OWLPropViz

3.5.1 Installation

OWLPropViz⁴ is a plug-in for *Protégé* 4.0. In order to use the plug-in, OWLViz and Graphviz are required to be installed in *Protégé*.

3.5.2 Implementation details

Programming language Java

Libraries Graphviz

3.5.3 Description

OWLPropViz resembles OWLViz a hierarchy-based visualization tool. However it provides more features, it can show the class-axioms "is-a", "disjoint-with" and "equivalent-to". The visualized graph of an ontology contains not only straight but also curly arrows. The tool can display whole or part of an ontology hierarchy and the graph is static. The visualization appears to be quite simple and due to the curly arrows the visualization can appear unclear. The handling is easy and intuitive. It is possible to export a graph as .png file. However, it was not possible to export a correct scaled image.

The homepage is not available and the protegewiki entry provides not many information. During the plug-in was in use, error messages appeared.

3.5.4 Layout Options

- Zoom in/out
- Hide/Show classes
- Hide/Show object properties
- Hide/Show Class Axioms
- Two layout directions (left to right, top to bottom)
- Show classes past radius

⁴<http://protegewiki.stanford.edu/wiki/OWLPropViz>

3 Graph Visualization

3.5.5 License

OWLPropViz - no information

Graphviz - Eclipse Public License - v 1.0 [moz]

Protégé - Mozilla Public License 1.1. [epi]

3.5.6 Screenshot

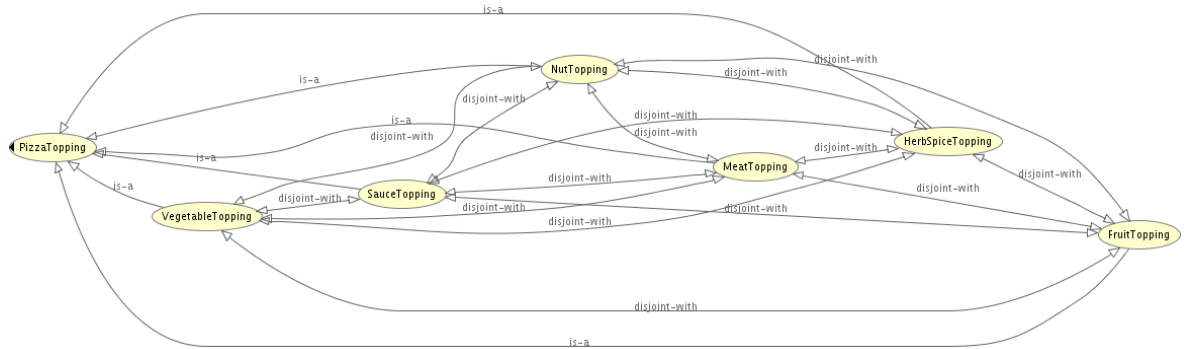


Figure 3.8: Part of the visualization of the Pizza ontology [DHS⁺07] with OWLPropViz 1.0 in Protégé 4.0

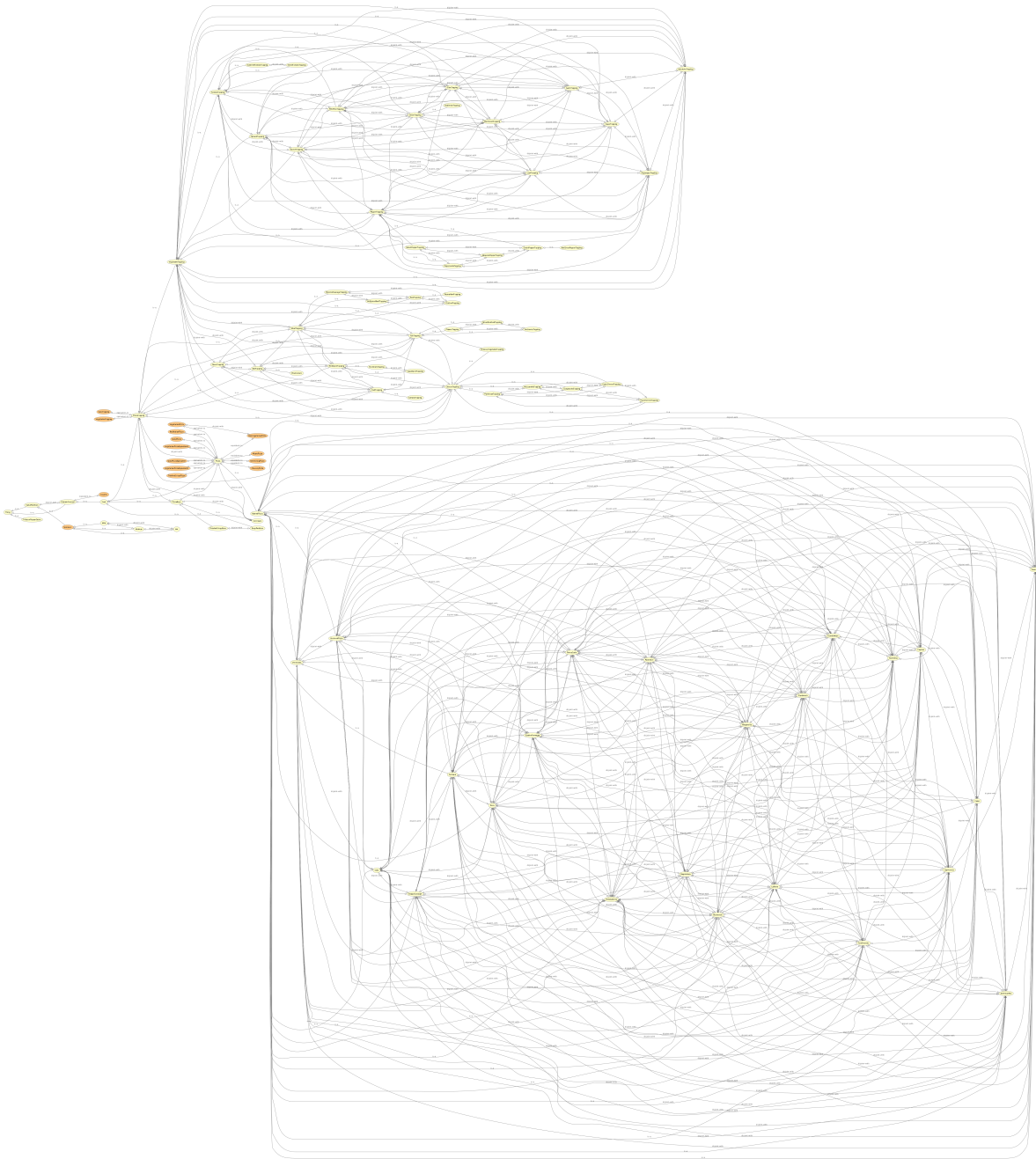


Figure 3.9: Visualization of the Pizza ontology [DHS⁺07] with OWLPropViz 1.0 in *Protégé* 4.0

3.5.7 Evaluation

Table 3.7: Evaluation of OWLPropViz 1.0

	Visualization capability
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression
OWLPropViz	● ● ● ● ● ● ●

Table 3.8: Visualization capabilities of OWLPropViz 1.0

	Classes
	Enumeration
	Property restrictions
	Cardinality
	Intersection
	Union
	Complement
	subClassOf
	equivalentClass
	disjointWith
	Object properties
	Datatype properties
	Instances
	Annotations
OWLPropViz	●

3.6 OWLGrEd

3.6.1 Installation

OWLGrEd [BBČ⁺10] exists as desktop application as well as an online version on the OWLGrEd webpage [owl14].

3.6.2 Implementation Details

Programming language Java

libraries Clojure

3.6.3 Description

OWLGrEd uses UML diagrams to represent the ontologies and does not differentiate between the desktop and online version. The look of the visualizations are simple but adequate. To visualize an ontology online, it is possible to upload your own ontology file and to share the respective link. But the ontology file must not exceed about 1000 elements. The interaction with the visualization graph is limited to zooming and moving. In contrast, the desktop version provides possibility to create ontologies and offers more functions to manipulate the graph (see "Layout options" section). The usability of the desktop application and the online version is both easy and intuitive. The OWLGrEd webpage is in details and provides information like notation and an installation guide. The visualized graph can be exported as .wmf, .emf, .bmp, .jpg or .gif file.

3.6.4 Layout Options

- Zoom in/out
- Undo/Redo
- Set background color
- Diagram overview (small window of the whole visualization)
- Horizontal/Vertical style palette
- Show meta model
- Alignment: Flow, Universal, Symmetrical, Tree, Matrix, Lanes
- Compact diagram mode (hides edge and class information for example object properties and datatypes)
- Possibility to insert commentaries

3 Graph Visualization

3.6.5 License

OWLGrEd - free for personal, academic and evaluation purposes but a commercial license for business

3.6.6 Screenshot

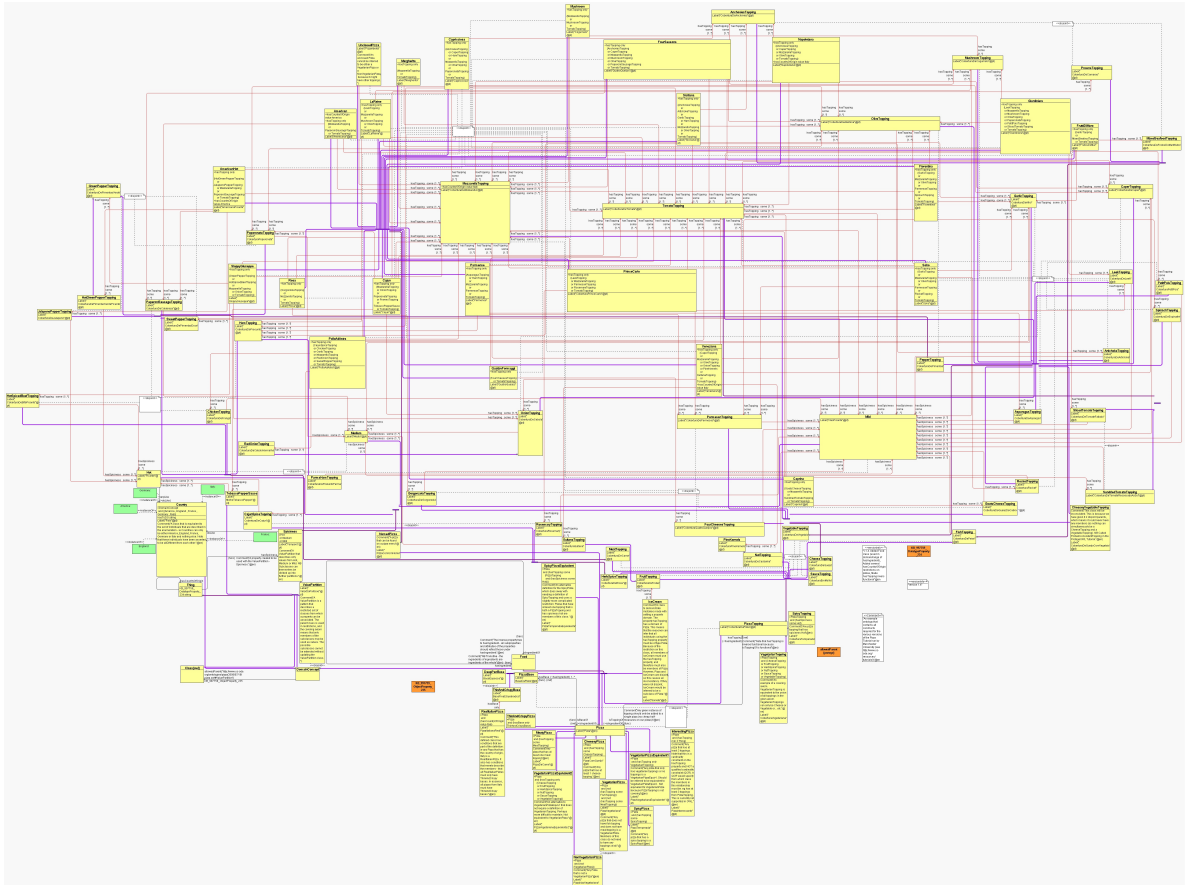


Figure 3.10: Visualization of the Pizza ontology [DHS⁺07] with OWLGrEd [owl14] 1.6.0 (desktop application)

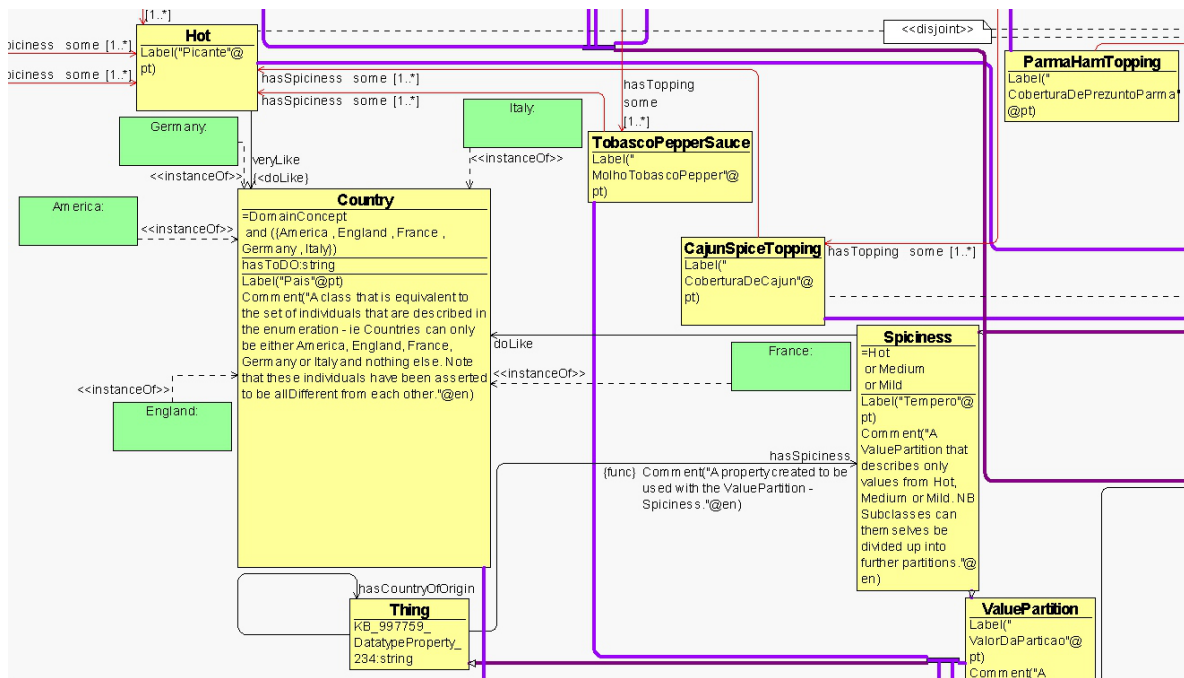


Figure 3.11: Part of the visualization of the Pizza ontology [DHS⁺07] with OWLGrEd [owl14] 1.6.0 (desktop application)

3.6.7 Evaluation

Table 3.9: Evaluation of OWLGrEd 1.6.0

	Visualization capability
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression
OWLGrEd	● ● ● ● ● ● ●

Table 3.10: Visualization capabilities of OWLGrEd 1.6.0

	Classes
	Enumeration
	Property restrictions
	Cardinality
	Intersection
	Union
	Complement
	subClassOf
	equivalentClass
	disjointWith
	Object properties
	Datatype properties
	Instances
	Annotations
OWLGrEd	● ● ● ● ● ● ● ● ● ● ● ● ●

3.7 TGViz

3.7.1 Installation

TGViz [Ala03] is a plug-in for *Protégé* and available in the default package of *Protégé* since version 3.0. For this study TGViz version 1.4, *Protégé* version 3.0 and java version 5.0.22 were installed.

3.7.2 Implementation details

Programming language Java 5

Library TouchGraph⁵

3.7.3 Description

TGViz is an additional tab which can be added as TGVizTab to the *Protégé* window. It provides the possibility to visualize an ontology with a graph whose layout depends on the semantic relationship between classes.

TGVizTab depicts classes as well as instances and object properties, all of which can be expanded/collapsed individually by the user. Both classes and individuals are visualized as borderless labels and only distinguishable by their relationship or by changing their colors in the option dialogue. These relations, like object properties, are visualized either as straight colored links between two components or if they are circular as blue text beneath a hovered component.

An additional feature of the layout is that the user is able to zoom in the structure, rotate or even add a hyperbolic curvature to it, which stretches the inner while compressing the outside parts. The tool has also as a feature a search bar. In conclusion TGViz provides good solutions for visualizing a large amount of information. However it has a few flaws like that a circular relation is only visible or two overlaying relations are only distinguishable when they are hovered.

3.7.4 Layout options

- Zoom in/out
- Hyperbolic bulge
- Rotate
- Hide/Expand classes or properties

⁵<http://www.touchgraph.com>

3 Graph Visualization

- Panning
- Drag and drop of the components
- Spring layout: based on the semantics
- Adaptability of the visibility/color of all relations

3.7.5 License

TGViz - no information

Protégé - Mozilla Public License 1.1. [moz]

3.7.6 Screenshot

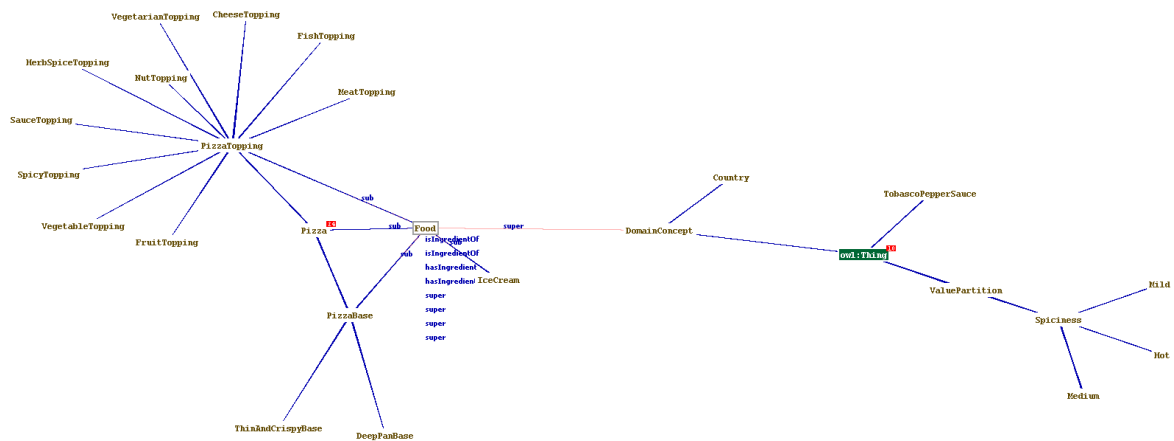


Figure 3.12: Visualization of the Pizza ontology [DHS⁺07] with TGViz 1.4 in Protégé 3.0 (spring layout)

3.7.7 Evaluation

Table 3.11: Evaluation of TGViz 1.4

	Visualization capability
TGViz	●
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression

Table 3.12: Visualization capabilities of TGViz 1.4

TGViz	●
Classes	
Enumeration	
Property restrictions	
Cardinality	
Intersection	
Union	
Complement	
subClassOf	●
equivalentClass	●
disjointWith	
Object properties	●
Datatype properties	
Instances	●
Annotations	

3.8 OntoRama

3.8.1 Installation

Steps to follow to install OntoRama [ERG02]:

1. Checkout OntoRama project code from <https://svn.code.sf.net/p/ontorama/code/trunk/OntoRama>
2. Import the project into Java Developing Environment (e.g Eclipse)
3. Run as Java application (in Select Java Application window, choose OntoRamaApp)
4. Click File -> Import -> loadFile and choose the link of the rdf/owl-File to be displayed

3.8.2 Implementation details

Programming language Java

3.8.3 Description

OntoRama [ERG02] is a Java application that renders RDF-based ontologies in a hyperbolic-style layout. It is required that input items of RDF-based ontologies are interconnected and furthermore connected directly or indirectly to a top ontology item, otherwise the hyperbolic view metaphor would usually breakdown.

OntoRama is capable of displaying concept types (represented by classes in RDF). Relationship types (represented by properties) are ignored. Thus, some information of RDF ontologies are lost. The tool supports multiple visual view over an ontology, including hyperbolic-style view and tree view.

OntoRama uses configuration XML file to configure different properties of object types in an ontology. If we want to display an ontology with different properties for each object types, all we need is to alter the XML file, without rebuilding the OntoRamaApp. An XML file consists of two sections: ontology and rdfMapping.

1. Ontology section: consists all details relevant to loading an ontology.
First of all, ontology section consists of multiple <relation>, each of them defines a relation link. Each relation has an id, which is corresponding to a relation that OntoRama uses. Each relation has also one name or two names in case of reversing relation (e.g subtype/supertype). Each relation consist displaying information (symbol and color, used to create an icon representing the relation link in the ontology).
Secondly, ontology section consists of multiple <conceptProperty>, each has an id as a string describing this property.
2. rdfMappings section: specifies how OntoRama will be able to find items defined in the ontology section in the source RDF file.
<relationLinks> maps relation links defined in ontology section to RDF tag.
<conceptProperties> maps concept types properties to RDF tag.

Advantages of hyperbolic-style view:

1. Large tree structures could be compactly displayed by projecting a tree onto a hyperbolic plane
2. An order of magnitude more nodes of a tree can be rendered in the same display space [ERG02]
3. The focus of attention is maintained on the central vertex and its neighborhood [ERG02]

3.8.4 Layout options

- Graphical zooming of overview by moving the Focal Depth panel (practical and user-friendly) as well as zooming of selected entity
- Quick selection of an entity with the help of hierarchical list on the left side of screen (tree view)
- History is available
- Search is available but does not work

3.8.5 License

OntoRama - BSD License [bsd99]

3 Graph Visualization

3.8.6 Screenshot

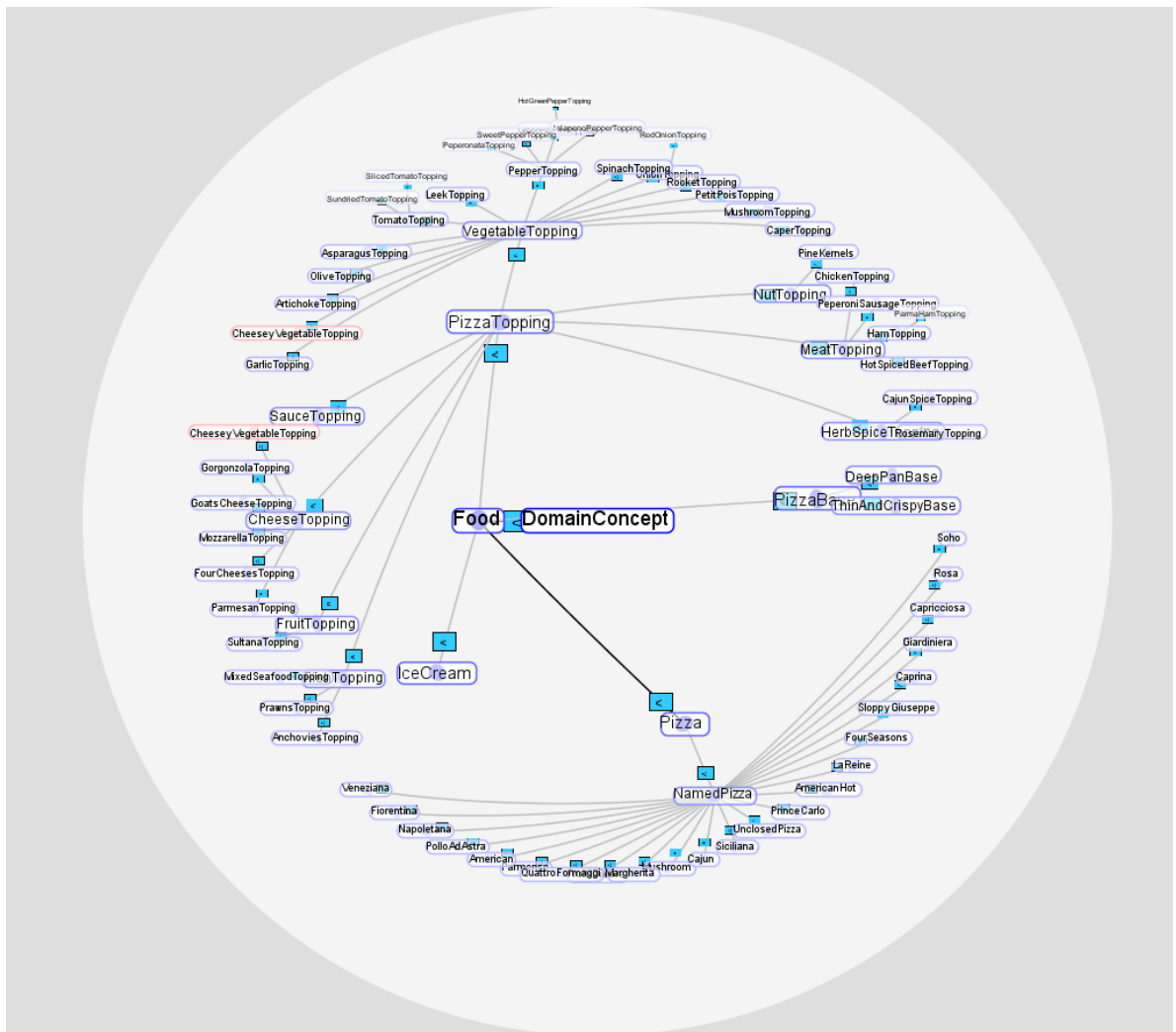


Figure 3.13: Visualization of the Pizza ontology with OntoRamaApp [ERG02]

3.9 OntoTrix

3.9.1 Installation

Steps to follow to install OntoTrix [BPLL11]:

1. Checkout OntoTrix project code from `svn://scm.gforge.inria.fr/svnroot/ontotrix/trunk`
2. Import the project into Java Developing Environment (e.g Eclipse) as an Maven project
3. Build and run
4. Click File -> Open and choose the link of the rdf/owl-File to be displayed

However, step 3 (build and run of OntoTrix code) was not successful because:

1. Used libraries (Jena, Pellet) are not up-to-date, thus dependencies in pom.xml are not resolved
2. Bugs in code (instances used are not declared, exceptions are not resolved)

3.9.2 Implementation details

Programming language Java

Interface based on ZVTM user interface toolkit

Layout and clustering LinLogLayout

Loading ontologies Jena 2

Storing ontologies TDB back end

3.9.3 Description

OntoTrix [BPLL11] is a hybrid ontology visualizer, using node-link diagram to represent the overall structure of an ontology and matrices to represent dense subgraphs. After an ontology was loaded, OntoTrix calculates the layout of the representing graph whose edges are object properties connecting instance nodes that belong to separate matrices.

By calculating the aggregate graph, nodes are matrices and single nodes that do not belong to a matrix (if any). Cubic curves are drawn for each object property connecting two instances that belong to different matrices (not edges of layout graph but actual object properties relating instance nodes). Direction of edges connecting instances belonging to different matrices are visualized by color and shade, that is why no arrow heads are required.

OntoTrix supports four views:

1. Main view is OntoTrix representation of instance set
2. Interactive bird-eye view of main view

3. Class hierarchy as a node-link diagram
4. Property hierarchy as a radial tree layout

Advantages of OntoTrix:

1. OntoTrix represents entities that bear semantics and therefore helps user to identify entities of interest and smoothly navigate between them
2. With proper labeling, OntoTrix can provide semantic zooming. At the moment, by grouping by class membership, each matrix is named by class name. In other cases, matrix has to be labeled manually, currently assigned a random identifier
3. OntoTrix is effective at showing large and dense graphs, it is also very efficient at visualizing locally dense but globally sparse network;

3.9.4 Layout options

- Smooth continuous zooming
- Visual transitions (switching between different grouping methods, elaborate transitions) are smoothly animated
- Filter: properties can be selected and deselected in the object property hierarchy (when grouping matrices by property type, properties not selected are visible but rendered with a lowly-contrasted color)
- Details-on-demand: when clicking inside a node, data properties associated with nodes pop-up
- Coordinated view: by hovering a node in the class hierarchy view, OntoTrix highlights all corresponding instances in the view
- Search: searching for specific instance
- In-place editing is not available

3.9.5 License

OntoTrix - BSD License [bsd]

3.9.6 Screenshot

Since it is not possible to run OntoTrix, it is not possible to create a screenshot.

3.9.7 Evaluation

Since it is not possible to run OntoTrix, the tool is not tested.

3.10 VOWL

3.10.1 Installation

VOWL [NLHE14] is available as plug-in for *Protégé* [NLB14] and as web application [LLMN14]. This study refers to the plug-in *ProtégéVOWL*⁶ version 0.1.3 and *WebVOWL*⁷ version beta 0.3.0. The plug-in was tested using *Protégé* version 4.3 and beta 5.0 with java version 1.7.0.. In difference to *ProtégéVOWL* which understands OWL directly one needs to translate the file into the VOWL-JSON format using *OWL2VOWL*⁸ in order to use the ontology with *WebVOWL*.

3.10.2 Implementation details

Programming language Java(*ProtégéVOWL* and *OWL2VOWL*), JavaScript (*WebVOWL*)

Libraries Prefuse [HCL05]

3.10.3 Description

VOWL visualizes the whole ontology as graph with almost all components defined in OWL [BHH⁺14]. VOWL distinguishes different components by representing classes as circles while everything else is inserted as rectangles, all displayed in a color related to their type or whether they are deprecated or external.

In consequence an ontology which contains only regular classes is depicted as structure of light blue bubbles (Figure 3.14). Furthermore it also depicts the existence of instances by increasing the size of the class they belong to. The layout implements a gravity-like behavior which in *ProtégéVOWL* lets all classes with no direct binding constantly move away from the center. This animation is possibly a bit disturbing since many structures are independent without any binding, but it can optionally be paused.

In summary, VOWL is a very powerful visualizer which can depict more structures than most of other visualizers even if one must be aware that currently *WebVOWL* is the far more matured implementation. It appears to be the most reasonable tool for big well-structured ontologies especially because *WebVOWL* also allows to filter the ontology and hide some structures. On the other hand, VOWL graphs look over-crowded when visualizing big hierarchical structures like 3.14.

⁶<http://vowl.visualdataweb.org/protegevowl/org.visualdataweb.vowl.jar>

⁷http://vowl.visualdataweb.org/webvowl/WebVOWL_0.3.0.zip

⁸<http://vowl.visualdataweb.org/webvowl/owl2vowl.zip>

3.10.4 Layout Options

- Force-directed graph layout
- Different graphical primitives and colors
- Zoom
- Manipulating of the class offset
- Panning

3.10.5 License

ProtégéVOWL - MIT License [mit88]

WebVOWL - MIT License [mit88]

3 Graph Visualization

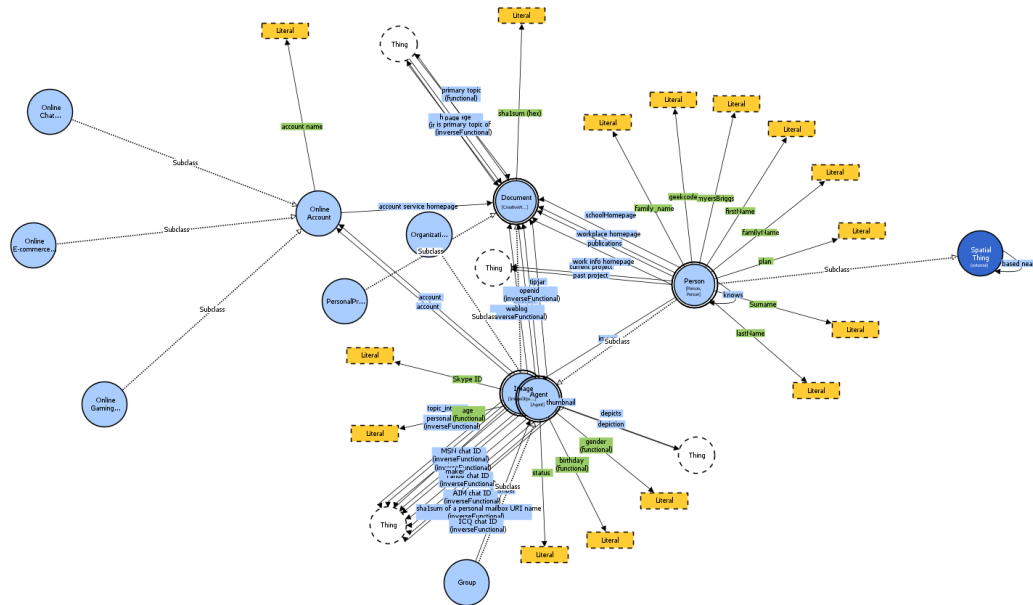


Figure 3.15: Visualization of the FOAF (Friend of a Friend) ontology with *ProtégéVOWL* [NLB14] 0.1.3

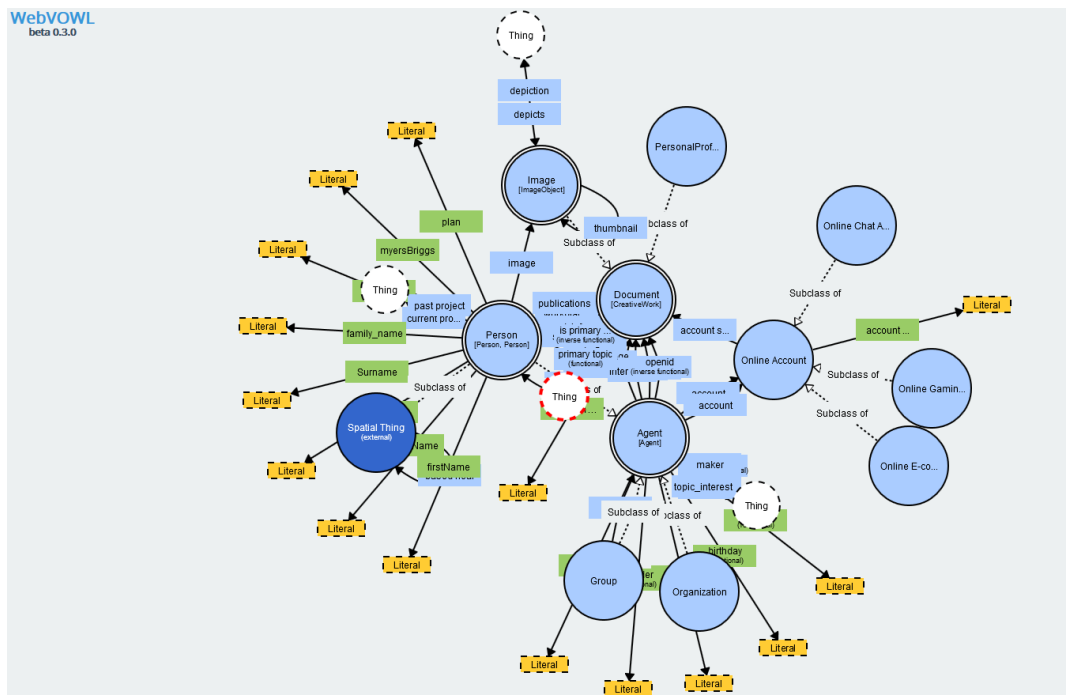


Figure 3.16: Visualization of the FOAF (Friend of a Friend) ontology with WebVOWL (beta 0.3.0) [LLMN14]

3.10.7 Evaluation

Since VOWL is provided in two totally different versions the evaluation is also given for WebVOWL and *ProtégéVOWL*.

Table 3.15: Evaluation of *ProtégéVOWL* 0.1.3 and WebVOWL 0.3.0

	Visualization capability	Look	Usability	Reliability	Documentation	Installation effort	Overall impression
<i>ProtégéVOWL</i>	●	●	●	●	●	●	●
WebVOWL	●	●	●	●	●	●	●

Table 3.16: Visualization capabilities of *ProtégéVOWL* 0.1.3 and WebVOWL 0.3.0

	Classes	Enumeration	Property restrictions	Cardinality	Intersection	Union	Complement	subClassOf	equivalentClass	disjointWith	Object properties	Datatype properties	Instances	Annotations
<i>ProtégéVOWL</i>	●	●	●	●	●	●	●	●	●	●	●	●	●	●
WebVOWL	●	●	●	●	●	●	●	●	●	●	●	●	●	●

3.11 NavigOwl

3.11.1 Installation

NavigOWL [HLTR14] is a plug-in for *Protégé*-OWL 4.1 [Pro15]. The installation of NavigOWL is performed by copying the latest version of NavigOwl into *Protégé* 4.1 plug-ins directory.

3.11.2 Implementation details

Programming language Java

Libraries Piccolo 2D, Jena 2.0

3.11.3 Description

NavigOwl is capable of loading an existing ontology (in form of *.rdf or *.owl file) and visualizing it as a graph of nodes and edges. The way these nodes are distributed is calculated upon power law, whose core concept is to partition the node set of a graph into power and non-power nodes and to establish local neighbourhood clusters among power nodes. This method of distribution is considered a strong point of NavigOwl, because it facilitates large-scale ontologies.

NavigOwl visualizes the whole role relation hierarchy defined in ontology with node tool-tips. It also implements specific arrow shapes and strokes for different types of relations between entities. The tool provides different layouts of the same graph, including circle layout, random layout, force layout, spring layout and power layout.

NavigOwl supports graph coloring. The colour-scheme is inspired from *Protégé* in order to remain the consistence. NavigOwl minimizes node cluttering and edges-overlapping, which are not avoidable in large ontologies.

3.11.4 Layout options

- Showing or hiding of node labels
- Zooming out the graph overview
- Handling mouse events like pan, drag, mouse-over for nodes of a graph
- Searching node service with highlights of searched result in the graph

3.11.5 License

NavigOwl - no information

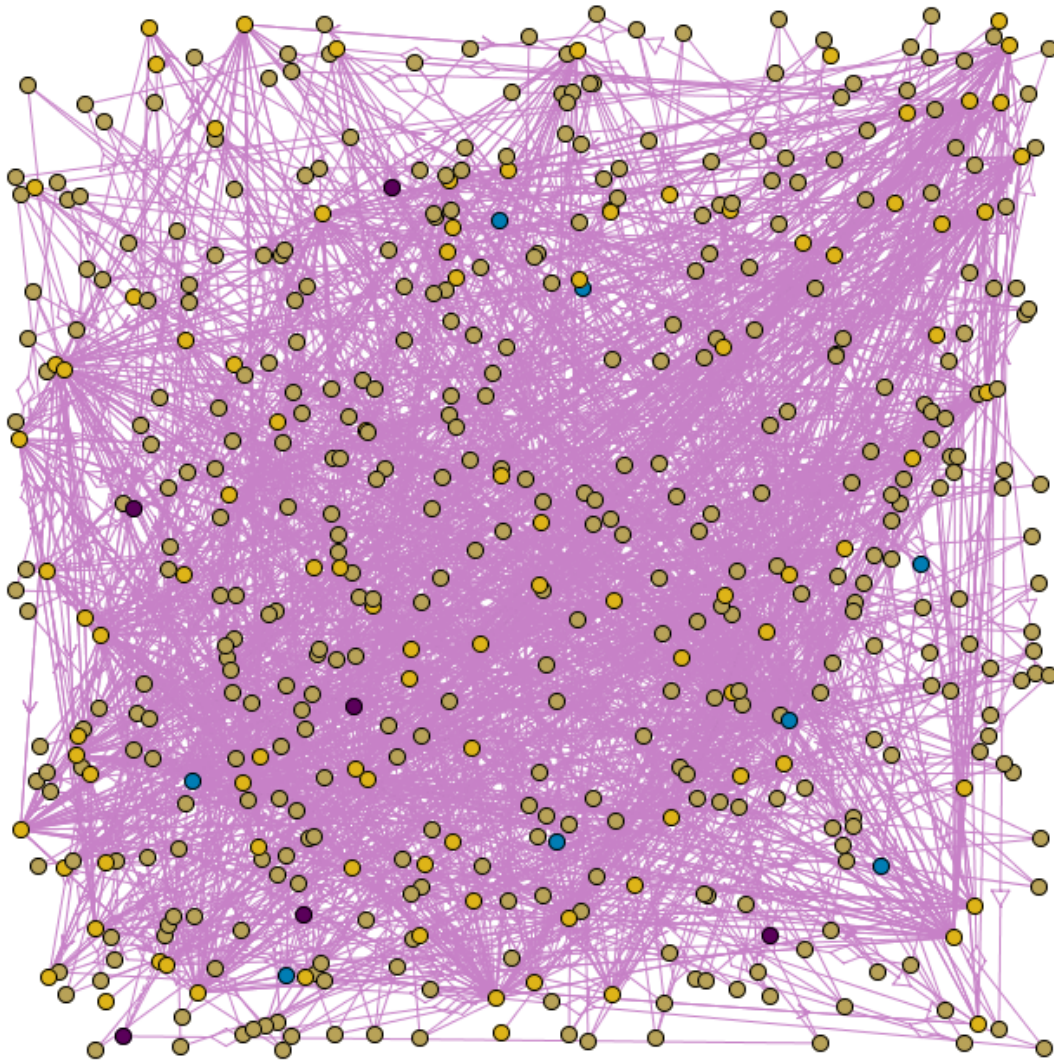


Figure 3.17: Visualization of the Pizza ontology with NavigOwl 1.1.0 in *Protégé*-OWL 4.1 (random layout)

3.11.6 Screenshots

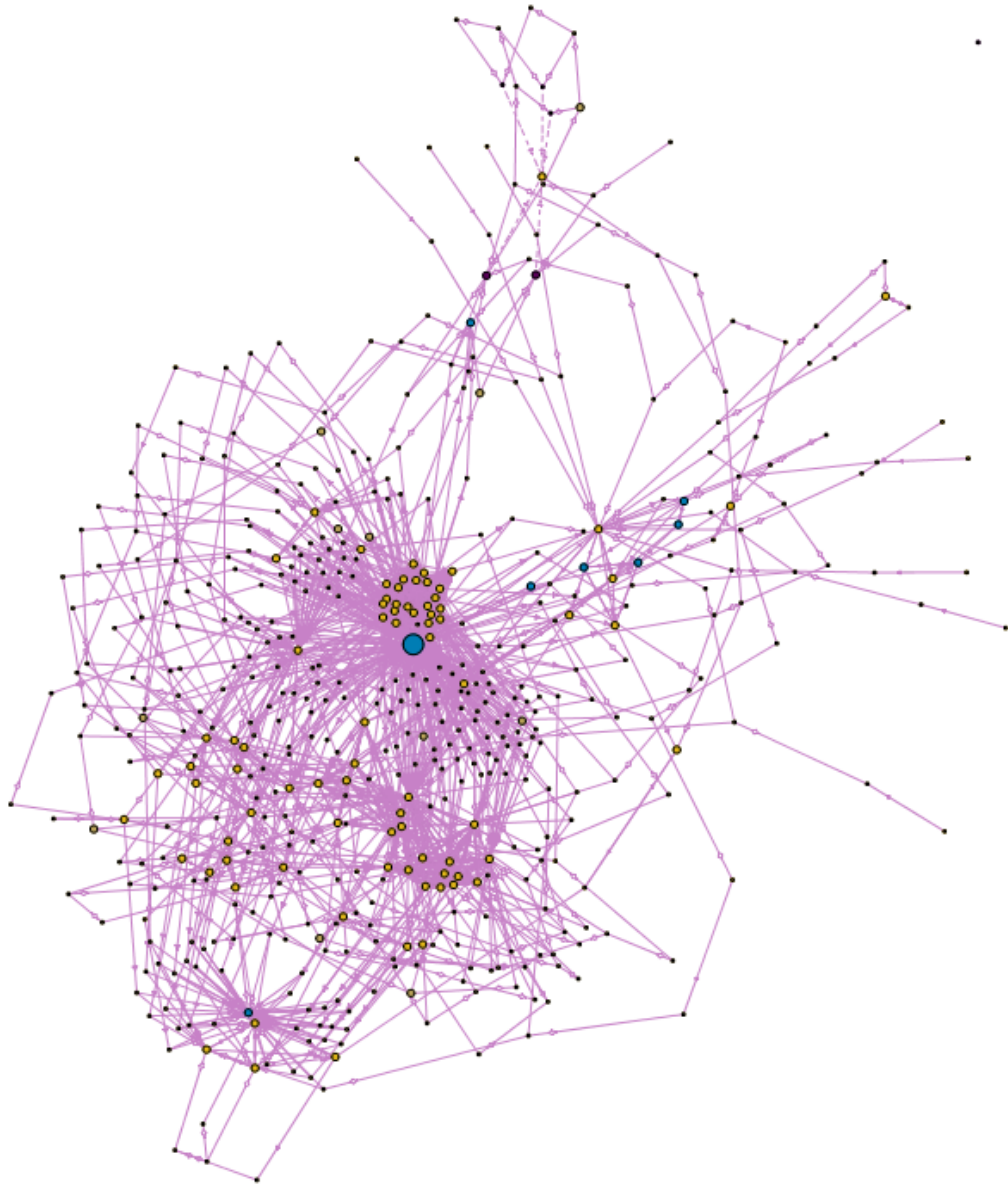


Figure 3.18: Visualization of the Pizza ontology with NavigOwl 1.1.0 in *Protégé*-OWL 4.1 (power layout)

3.11.7 Evaluation

Table 3.17: Evaluation of NavigOwl 1.1.0

	Visualization capability
NavigOWL	●
	Look
	●
	Usability
	●
	Reliability
	●
	Documentation
	●
	Installation effort
	●
	Overall impression
	●

Table 3.18: Visualization capabilities of NavigOwl 1.1.0

NavigOWL	●
Classes	●
Enumeration	●
Property restrictions	●
Cardinality	●
Intersection	●
Union	●
Complement	●
subClassOf	●
equivalentClass	●
disjointWith	●
Object properties	●
Datatype properties	●
Instances	●
Annotations	●

3.12 GrOWL

3.12.1 Installation

GrOWL [KWV07] is built both as stand alone application and as an applet. The applet version was used in Ecosystem Services Database, which is not publicly available. For the purpose of this study, file GrOWLEditor.jar is provided by the supervisor.

3.12.2 Implementation details

Libraries Prefuse [HCL05], Jena API, WonderWeb OWLAPI, TouchGraph⁹

Design Policy-based design

3.12.3 Description

GrOWL is a visualization and editing tool that based on OWL-DL (Web Ontology Language - Description Logic). It visualizes the underlying DL semantics of OWL ontologies, without exposing the complex OWL syntax.

GrOWL displays an ontology automatically in tree layout and graphical layout. Furthermore, class hierarchy of ontology is presented in two forms, one as a navigation tree in the left hand side of the screen, the other as a graph of classes with or without instances. It also displays ABox, TBox or RBox separately without choosing any nodes.

3.12.4 Layout options

- Zoom in and out by clicking corresponding icons
- Filter, which means displaying only class definition, the subclasses, the superclasses and instances of a selected node (The selection can be done either in hierarchical tree or in graph)
- Prefix search: When user types the name of a node (class or property), a list of names will be displayed for him to choose. Once an item is selected, the locality network will be showed using the chosen filter

3.12.5 License

GrOWL - No information

⁹<http://www.touchgraph.com>

3.12.6 Screenshots

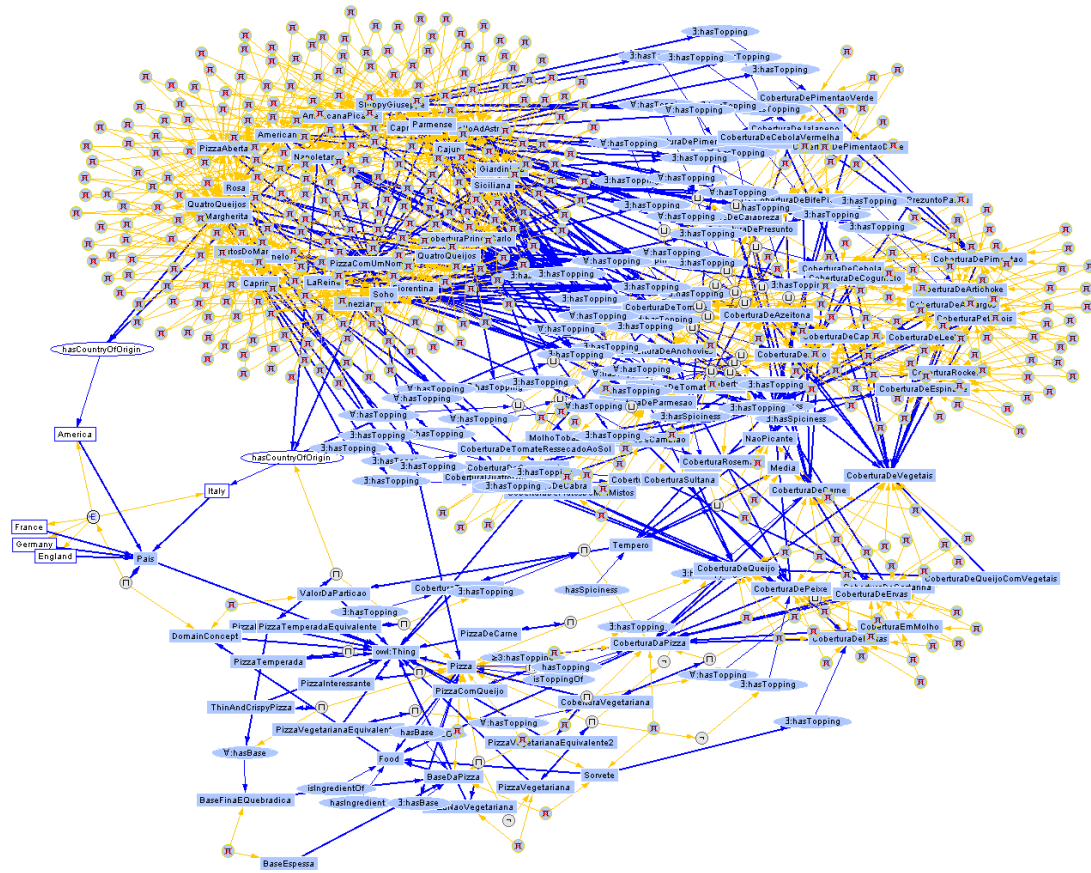


Figure 3.19: Visualization of the Pizza ontology with GrOWL

3.12.7 Evaluation

Table 3.19: Evaluation of GrOWL

	Visualization capability
GrOWL	●
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression

Table 3.20: Visualization capabilities of GrOWL

GrOWL	●
Classes	●
Enumeration	●
Property restrictions	●
Cardinality	●
Intersection	●
Union	●
Complement	●
subClassOf	●
equivalentClass	●
disjointWith	●
Object properties	●
Datatype properties	●
Instances	●
Annotations	●

3.13 COE (Concept-map Ontology Environment)

3.13.1 Installation

The installer for COE (Concept-map Ontology Environment) version 5.0 [HEM⁺05] [coe15] is available for Windows, Mac OS X, Solaris and Linux. There is no need to install any additional software.

3.13.2 Implementation details

Programming language Java

3.13.3 Description

The COE tool is built on CmapTools, which is a tool for concept mapping. Therefore COE works with concept mapping and in addition with OWL ontologies. It is possible for user to create every construct of an ontology (class, link, etc.) and edit it with the mouse.

The tool visualizes various aspects of an ontology in different way. For example, sub-links are shown with blue and definitions with red lines. Links are connections representing relations between classes (complement, subclassOf, disjointWith) and being labeled with simplified expressions.

COE is also able to depict individuals shown as rectangles and connected with their parent class by plain arrows labeled with "is a". Among is-a relations CEO depicts object- and datatype properties with restrictions as arrows and domain-range properties as dashed arrows. To avoid clutter by representing too much information, CEO included redundant classes, which means a complex structure can be simply splitted in two structures.

The homepage of the tool provides many information like a manual or startup instructions. COE can import .owl, .rdfs and .rdf files from XML files or from URIs. It is possible to export the ontologies as an .owl file, but not as an image. The handling is designed for concept maps and therefore it was inconvenient to visualize ontologies. For this study, an ontology should be imported and then displayed using COE, but it did not work well. The program must be repeatedly restarted.

3.13.4 Layout options

- Zoom in/out
- Editing the visualization with the mouse
- Hierarchical/Force-Directed Layout
- Node distance options
- Line options (straight, with corners, curved)

3 Graph Visualization

- Navigation tool (little map of the whole ontology)
- Add background

3.13.5 License

COE - individual license

3.13.6 Screenshot

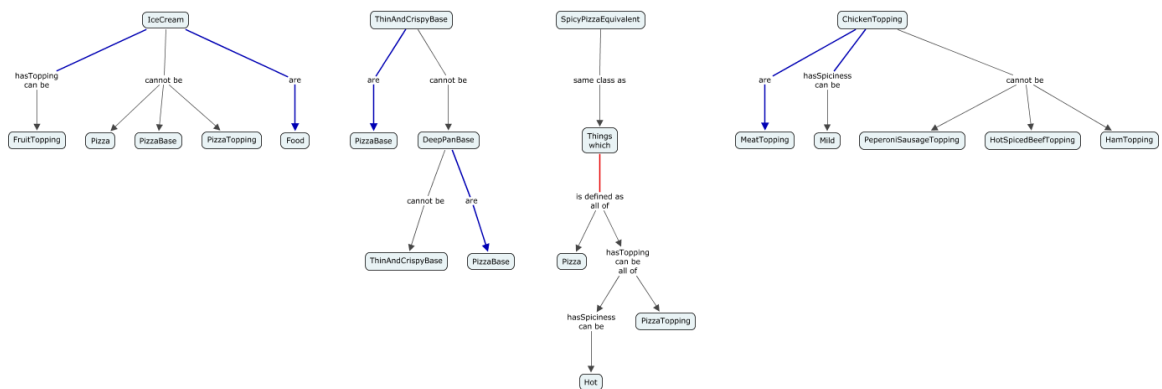


Figure 3.20: Part of the visualization of the Pizza ontology [DHS⁺07] with COE 5.0 [coe15] [HEM⁺05]

3.13 COE (Concept-map Ontology Environment)

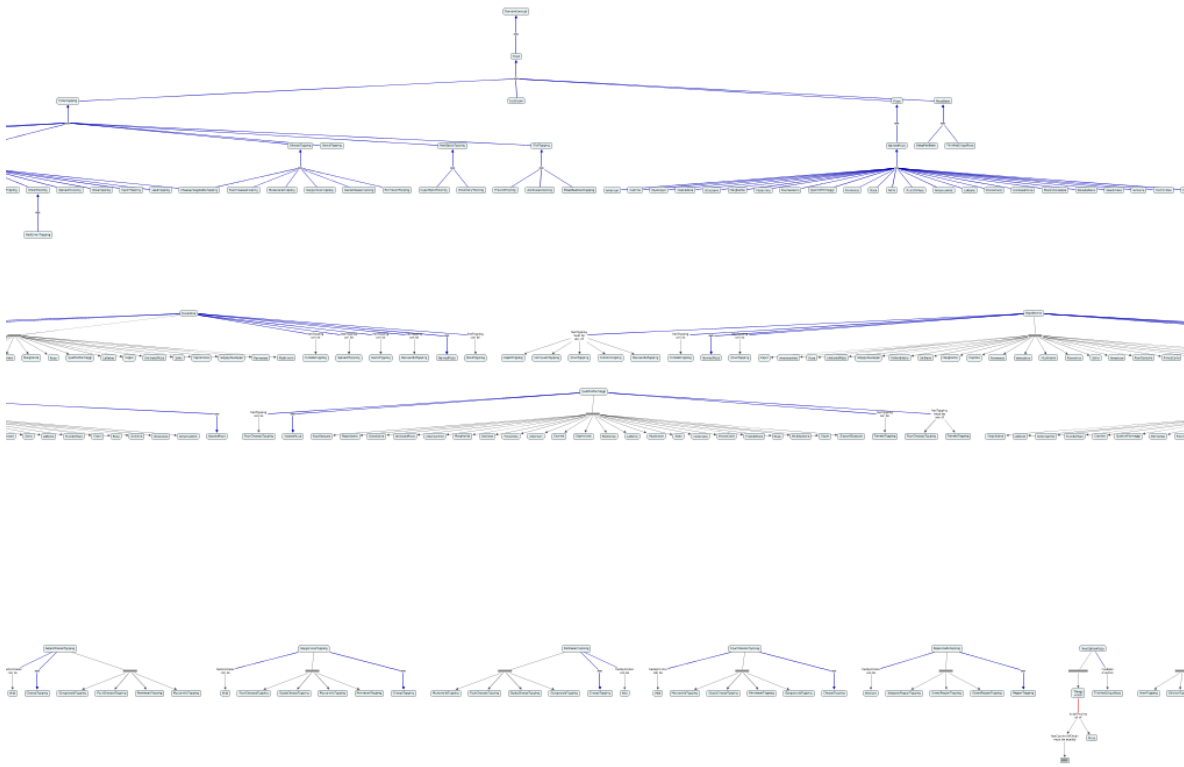


Figure 3.21: Visualization of the Pizza ontology [DHS⁺07] with COE 5.0 [coe15] [HEM⁺05]

3.13.7 Evaluation

Table 3.21: Evaluation of COE 5.0

COE	●	●	●	●	●	●	●
	Visualization capability	Look	Usability	Reliability	Documentation	Installation effort	Overall impression

Table 3.22: Visualization capabilities of COE 5.0

COE	●	●	●	●	●	●	●	●	●	●	●	●	●	●
	Classes	Enumeration	Property restrictions	Cardinality	Intersection	Union	Complement	subClassOf	equivalentClass	disjointWith	Object properties	Datatype properties	Instances	Annotations

4 Universal Visualization

Chapter 2 and chapter 3 gave an introduction of those tools focusing on a single visualization approach and then provide a specific visualization of an ontology. This chapter aims at the tools with different approaches of visualization.

4.1 GLOW

4.1.1 Installation

GLOW [Glo15] [HRFH12] version 1.0.0 is a plug-in for *Protégé* since version 4.0. In order to run it, JOGL 2.0 and Java SE6 are required.

4.1.2 Implementation details

Programming language Java

Libraries JOGL 2.0

4.1.3 Description

GLOW provides four layout options, only two of which worked in the test environment, depicting classes and object properties of an ontology. Each layout is computed automatically and does not provide any interaction apart from regulating the bundling of domain/range edges with a slider on the bottom of the window.

By default GLOW depicts the ontology as force-directed graph like figure 4.2. Alternatively the user can choose an inverted radial tree layout which also depicts inheritance by printing subclasses as sub-radials of their parents like figure 4.1.

Node-link tree and a squarified tree layout are also promised to be available. However they did not work and therefore no screenshots would be provided.

Furthermore the user can only choose between classes and classes+individuals. In the first case the user has an opportunity to see domain-range edges. When he chooses the second alternative a list of all available adjacency relations is provided. Domain-range edges are visualized by two-colored lines (e.g. from green to red), which can be changed in the advanced menu on the bottom right of the window.

4.1.4 Layout options

- Force-directed graph
- Inverted radial tree
- Node link tree (not working)
- Squarified map(not working)
- Semantic zoom
- Panning
- Nodes/Connections can not be moved/modified or selected by the mouse
- Full automatic layouting

4.1.5 License

Protégé - Mozilla Public License 1.1. [moz]

JOGL 2.0 - New BSD 2-Clause License [bsd]

4.1.6 Screenshot

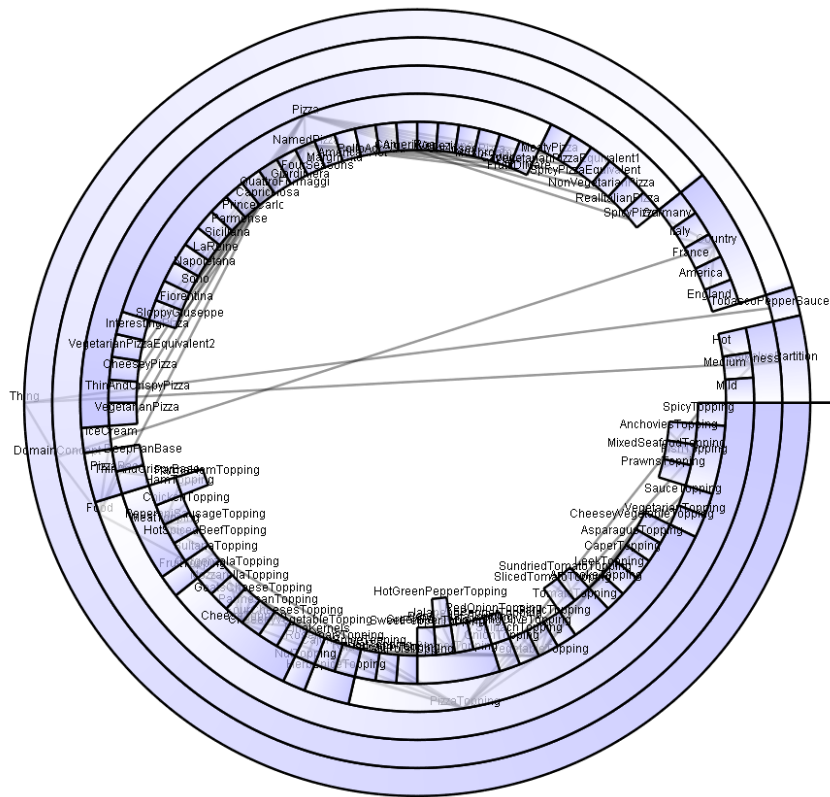


Figure 4.1: Visualization of the Pizza ontology [DHS⁺07] with GLOW [Glo15] [HRFH12] 1.0 in *Protégé* 4.0.2 (inverted radial tree)

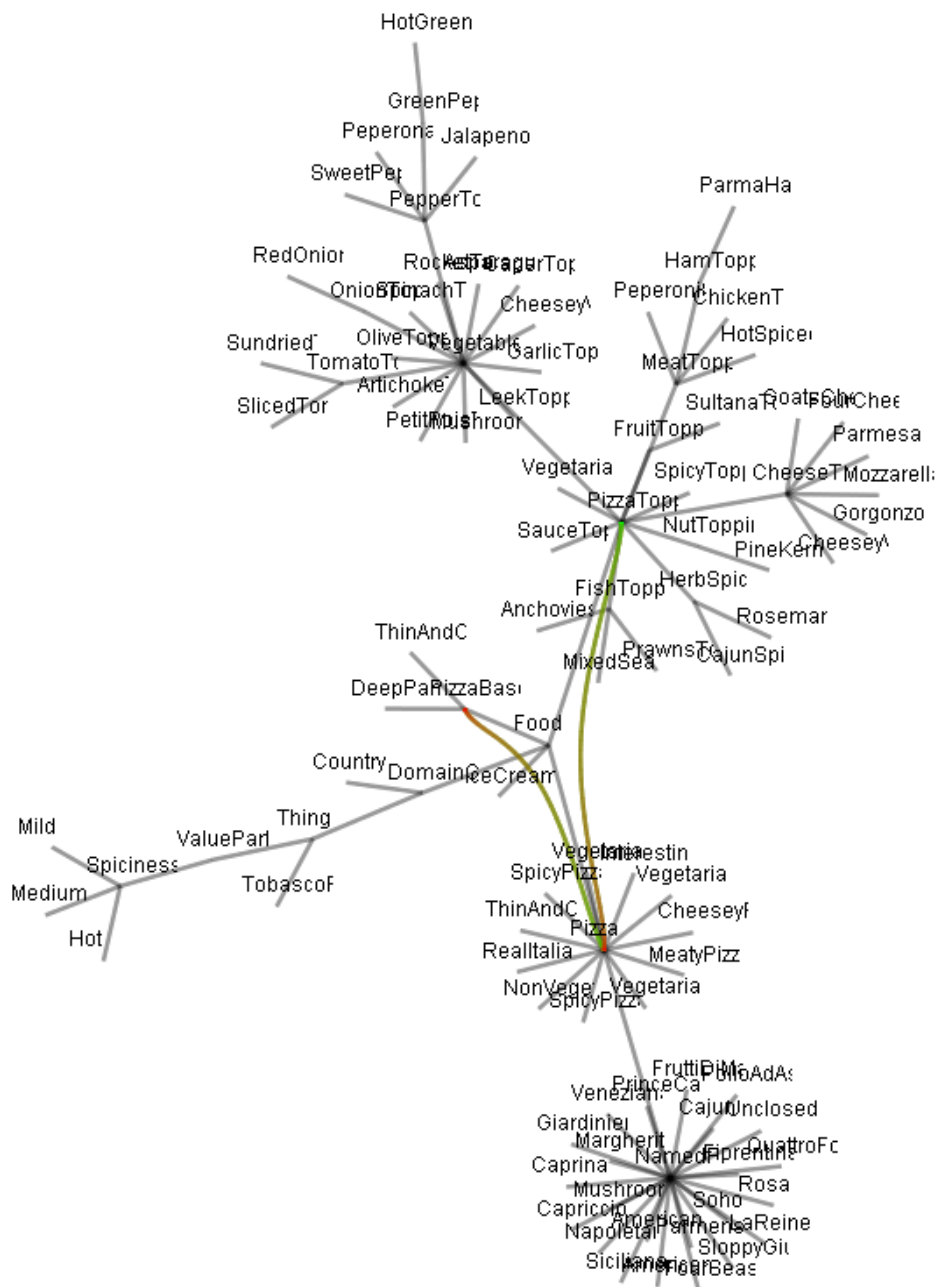


Figure 4.2: Visualization of the Pizza ontology [DHS⁺07] with GLOW [Glo15] [HRFH12] 1.0 in *Protégé* 4.0.2 (force-directed graph)

4.1.7 Evaluation

Table 4.1: Evaluation of GLOW 1.0

	Visualization capability
GLOW	●
	●
	●
	●
	●
	●
	●
	●

Table 4.2: Visualization capabilities of GLOW 1.0

	Classes	Enumeration	Property restrictions	Cardinality	Intersection	Union	Complement	subClassOf	equivalentClass	disjointWith	Object properties	Datatype properties	Instances	Annotations
GLOW	●				●			●			●	●	●	

4.2 SOVA

4.2.1 Installation

SOVA¹ (Simple Ontology Visualization API) is a plug-in for *Protégé* 4.0 - 4.3. In order to use the plug-in, Hermit must be installed within *Protégé*.

4.2.2 Implementation details

Programming language Java

Libraries Hermit

4.2.3 Description

SOVA¹ offers graphical and hierarchical visualization of ontologies. The graphical visualization has two layouts, namely force directed and radial tree layout.

The plug-in can be easily and transparently handled. There are only a few control elements like searchbar and play/stop button to turn on/turn off the auto-arrange of the graph.

If the mouse moves over a node, all linked nodes are highlighted.

The plug-in can visualize almost every language constructs (except annotations) through several symbols and colors. The hierarchical visualization is limited to classes and individuals. In the graph layout it is not possible to hide parts of the ontology but the hierarchy tree has the option to switch between displaying the full tree and displaying a tree with a depth of three levels.

The homepage of SOVA¹ provides a link to the protegewiki entry where the user can find all necessary information about the plug-in. During the plug-in was in use, error messages appeared with some ontology files, therefore the screenshots were not created from the pizza [DHS⁺07] ontology but from the people ontology [peo14]. It is possible to save the image as .png file.

4.2.4 Layout options

- Zoom in/out (with Scroll wheel)
- Auto-arrange off/on
- Movable graph
- Force directed layout/radial tree layout
- Hierarchy Tree

¹<http://protegewiki.stanford.edu/wiki/SOVA>

- Filters (e.g. class, cardinality, domain, ...)
- Connectivity filter (configurable distance)

4.2.5 License

SOVA - GNU LESSER GENERAL PUBLIC LICENSE Version 3 [LGP07]

HermiT - GNU LESSER GENERAL PUBLIC LICENSE [LGP91]

Protégé - Mozilla Public License 1.1. [moz]

4.2.6 Screenshot

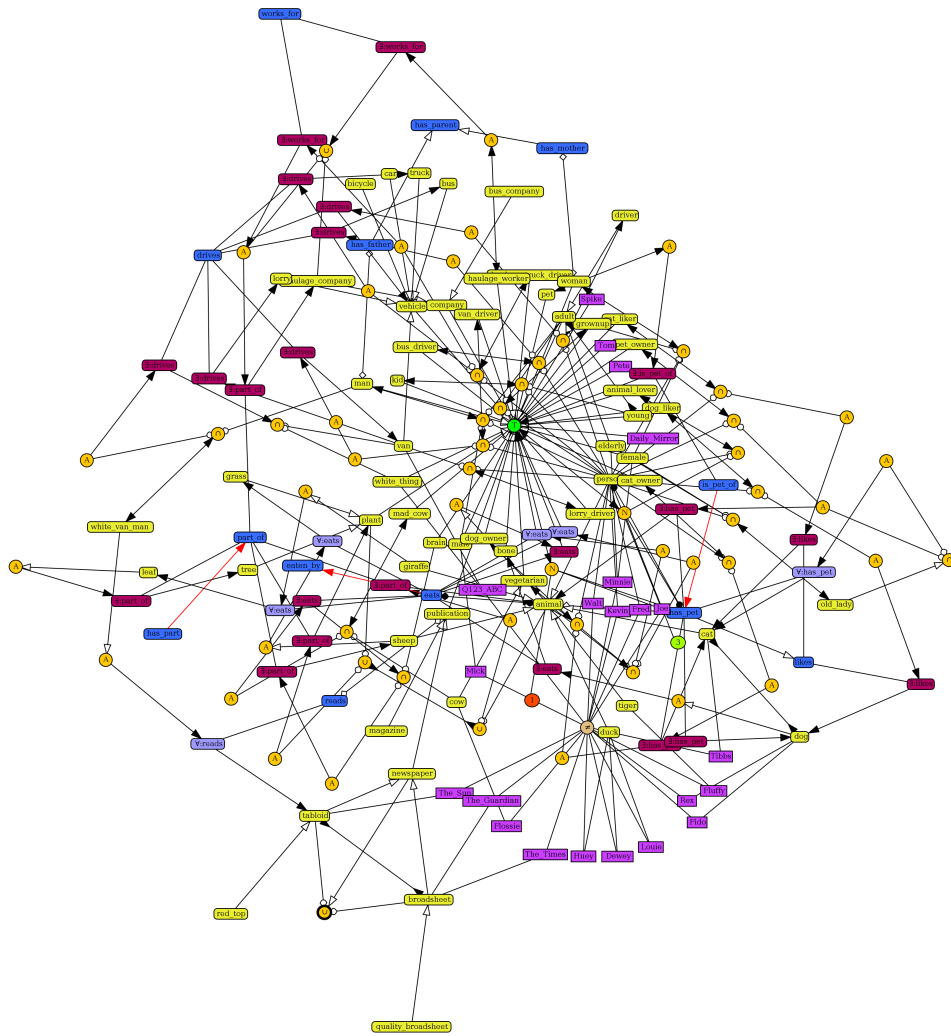


Figure 4.3: Visualization of the People ontology [peo14] with SOVA 0.8.4 in *Protégé* 4.3 (force directed layout)

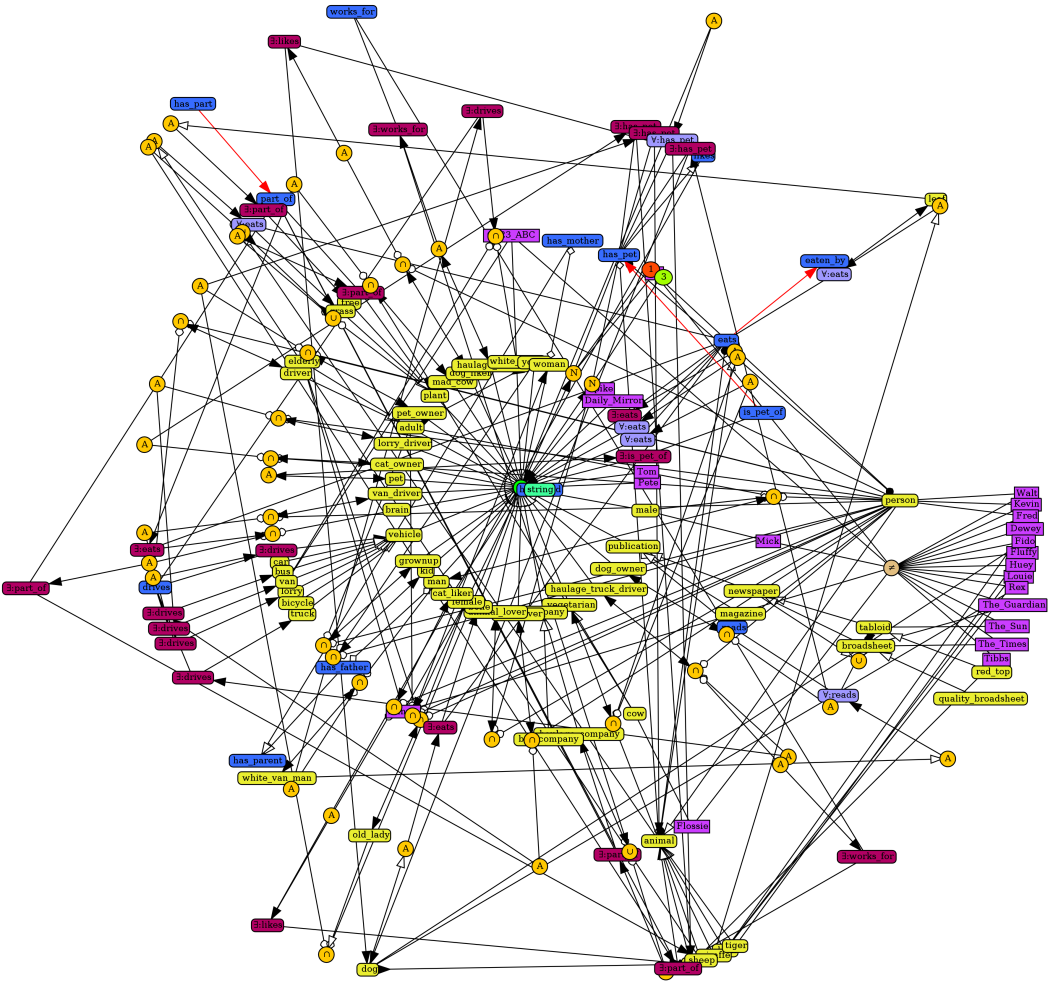


Figure 4.4: Visualization of the People ontology [peo14] with SOVA 0.8.4 in Protégé 4.3 (radial tree layout)

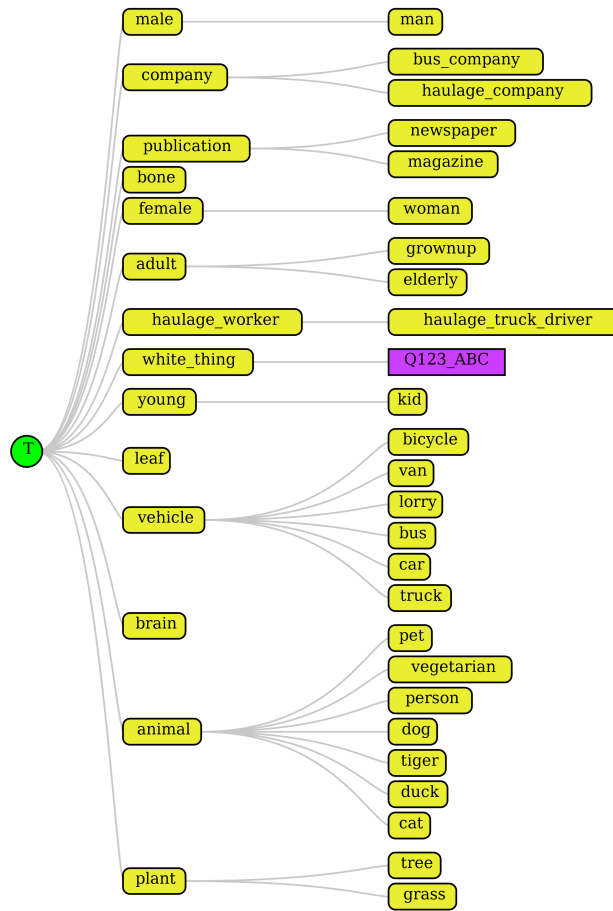


Figure 4.5: Visualization of the People ontology [peo14] with SOVA 0.8.4 in *Protégé* 4.3 (hierarchy tree)

4.2.7 Evaluation

Table 4.3: Evaluation of SOVA 0.8.4

SOVA	Visualization capability	●
	Look	●
	Usability	●
	Reliability	●
	Documentation	●
	Installation effort	●
	Overall impression	●

Table 4.4: Visualization capabilities of SOVA 0.8.4

SOVA	Classes	●
	Enumeration	●
	Property restrictions	●
	Cardinality	●
	Intersection	●
	Union	●
	Complement	●
	subClassOf	●
	equivalentClass	●
	disjointWith	●
	Object properties	●
	Datatype properties	●
	Instances	●
	Annotations	

4.3 TopBraid Composer

4.3.1 Installation

TopQuadrant [top14] offers two commercial versions and a free version of its TopBraid Composer, but only the two commercial ones provide visualization of RDF files. The latest version 4.5.0 can be downloaded on the TopBraid download page² and run on Java version 7.

4.3.2 Implementation details

Programming language Java

Libraries Jena

4.3.3 Description

Strictly spoken the TopBraid Composer Maestro Edition offers two visualization approaches. The first approach is a class diagram that displays inheritance, properties and (dis-)join links as colored connections between class rectangles. The second one is a class graph where users can browse through classes or properties of a navigation tree.

4.3.4 Layout options

- Tree layout as topdown, bottomup or leftright
- Diagramm layout
- Zoom in/out
- Overview window
- Drag and drap for nodes
- Manual/Automatic layouting
- Show/Hide comments and labels

4.3.5 License

TopBraid - EULA³

Eclipse - Eclipse Public License - v 1.0 [ep1]

Apache Jena - Apache License, Version 2.0 [apa04]

²<http://www.topquadrant.com/downloads/topbraid-composer-install/>

³[//www.topquadrant.com/docs/legal/EULA.pdf](http://www.topquadrant.com/docs/legal/EULA.pdf)

4.3.6 Screenshot

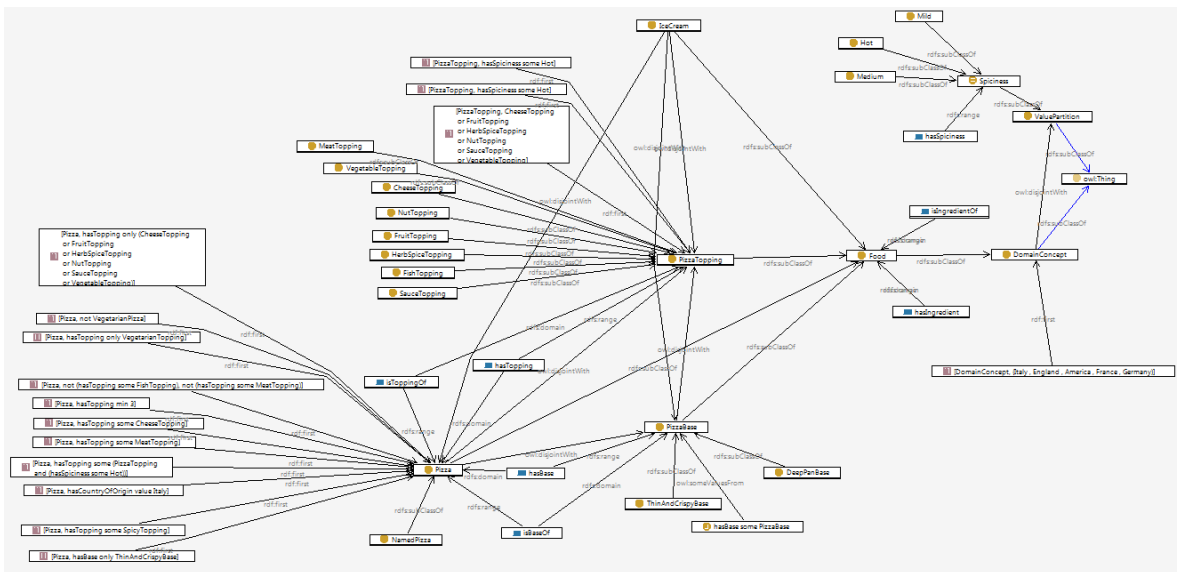


Figure 4.6: Visualization of the Pizza ontology [DHS⁺07] with TopBraid Composer Maestro Edition [top14] 4.5.0 as graph

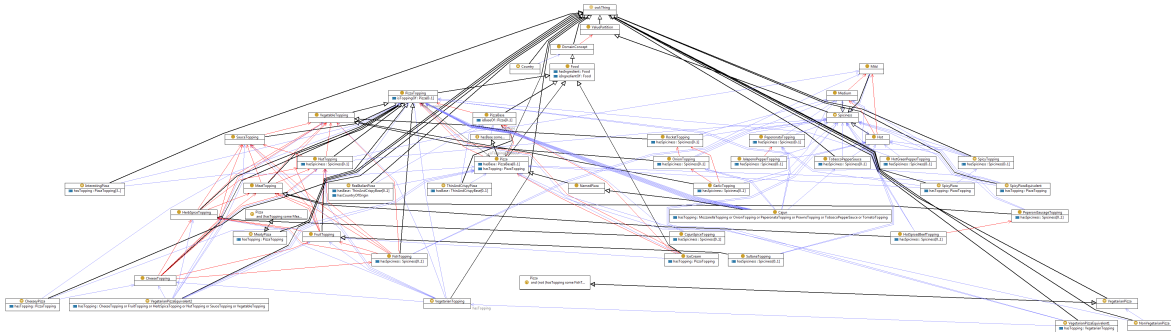


Figure 4.7: Visualization of the Pizza ontology [DHS⁺07] with TopBraid Composer Maestro Edition [top14] 4.5.0 as diagram

4.3.7 Evaluation

Table 4.5: Evaluation of TopBraid Composer Maestro Edition 4.5.0

	Visualization capability	Look	Usability	Reliability	Documentation	Installation effort	Overall impression
TopBraid Composer	●	●	●	●	●	●	●

Table 4.6: Visualization capabilities of TopBraid Composer Maestro Edition 4.5.0

	Classes	Enumeration	Property restrictions	Cardinality	Intersection	Union	Complement	subClassOf	equivalentClass	disjointWith	Object properties	Datatype properties	Instances	Annotations
TopBraid Composer	●	●	●	●	●	●	●	●	●	●	●	●	●	●

4.4 Jambalaya

4.4.1 Installation

Jambalaya⁴ [SMS⁺01] is a plug-in for *Protégé* 3.2.1 - 3.4 and is already contained in the installation of *Protégé*.

4.4.2 Implementation details

Programming language Java

4.4.3 Description

The plug-in has many different visualizing alternatives and functionalities. The visualized ontologies look a bit outdated, even if jambalaya uses simple structures.

Jambalaya⁴ has many elements to improve the overview of the visualization, for example arc and node filters or the thumbnail view. There are six quick views, which can also be adjusted with the radial, spring, tree or grid layout.

It is possible to arrange, hide and increase/decrease the classes with the mouse. With a double click on a class in the class browser or with the zoom tool you can edit the class in the visualization.

Although these features increase the usability of Jambalaya⁴ there are so many functions need a settling-in period. The search bar has auto-completion function and different search options, including contain, start with, end with, exact match and regexp.

Beside redo/undo buttons the plug-in has a "Filmstrip" feature which allows the user to take snapshots of the current visualization and switch between them.

There is a built-in manual with a short overview of Jambalaya⁴ but the webpage with the user manual is unavailable. It is possible to export the image of the visualization as a .png, .jpg or .gif file.

4.4.4 Layout options

- Radial layout, spring layout, tree layout vertical/horizontal
- Quick views (nested view, nested treemap, nested composite view, class and individual tree, class tree, domain/range)
- Create own custom views
- Grid layout (by number of children, by alphabetical order, by number of relationships, by type, by attribute)

⁴<http://protegewiki.stanford.edu/wiki/Jambalaya>

4 Universal Visualization

- Zoom in/out
- Hide entities
- Arc labels on/off
- Arc filter
- Node labels (on node, fit to node, wrap to node, above node (fixed), above node (level))
- Node filter
- Thumbnail view
- Query view

4.4.5 License

Protégé - Mozilla Public License 1.1.

Jambalaya - individual license (located in the “About Plugins” window)

4.4.6 Screenshot

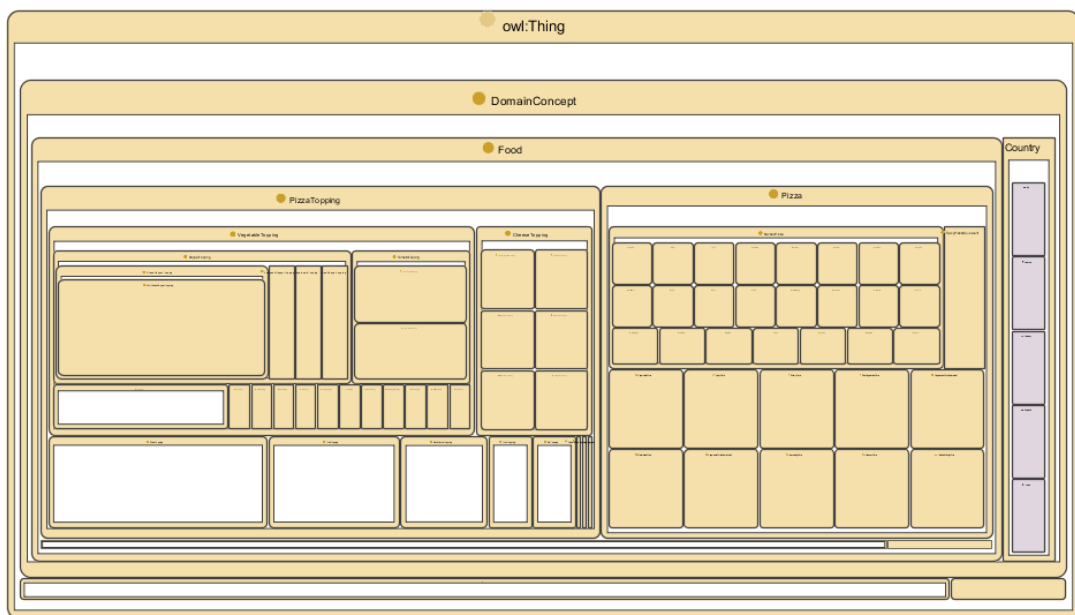


Figure 4.8: Visualization of the Pizza ontology [DHS⁺07] with Jambalaya 2.7.0 in *Protégé* 3.5 (nested treemap layout)

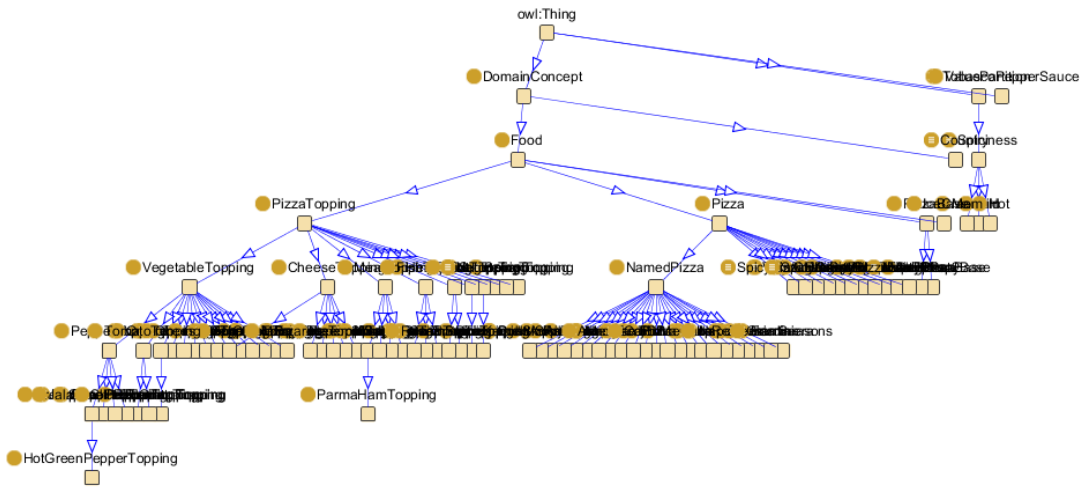


Figure 4.9: Visualization of the Pizza ontology [DHS⁺07] with Jambalaya 2.7.0 in *Protégé* 3.5 (class tree layout)

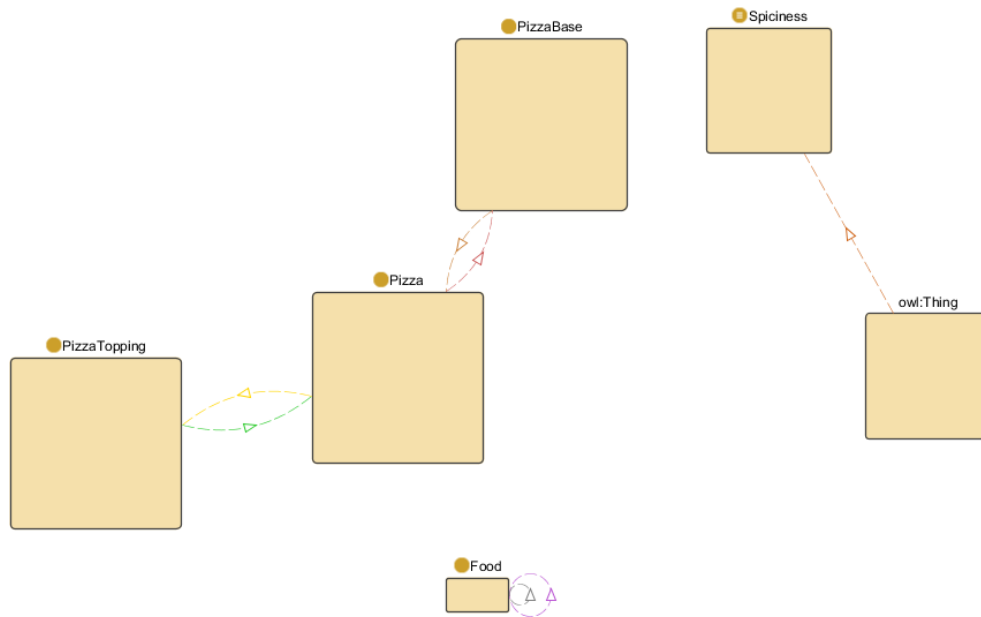


Figure 4.10: Visualization of the Pizza ontology [DHS⁺07] with Jambalaya 2.7.0 in *Protégé* 3.5 (domain/range layout)

4.4.7 Evaluation

Table 4.7: Evaluation of Jambalaya 2.7.0

	Visualization capability
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression
Jambalaya	● ● ● ● ● ● ●

Table 4.8: Visualization capabilities of Jambalaya 2.7.0

	Classes
	Enumeration
	Property restrictions
	Cardinality
	Intersection
	Union
	Complement
	subClassOf
	equivalentClass
	disjointWith
	Object properties
	Datatype properties
	Instances
	Annotations
Jambalaya	● ● ● ● ● ● ● ● ● ● ● ●

5 Related Works

Apart from the "pure" OWL ontology visualization tools, there are others providing knowledge representation with SPARQL [spa14], other OWL layers [DVR07] or alternative approaches. This chapter introduces a selection of such tools.

5.1 FlexViz

5.1.1 Installation

FlexViz is a project for the Adobe Flash Builder [fla15], which can be tested for a limited period and after that should be purchased. The trial code can be downloaded from [fle15]. FlexViz is written in ActionScript and Adobe Flex. The user can individually adapt the source and then export the program as .swc (Shockwave Flash), which can be deployed on a webpage.

5.1.2 Implementation details

Programming language Adobe Flex, Action Script 3.0

5.1.3 Description

Implementation of a personal FlexViz instance requires professional skills and time. Luckily an example implementation can be found on [bio15].

FlexViz enables visualization of an ontology with classes and relationships between them. It provides different visualization layouts such as vertical layout, radial layout, ect... Since it does not natively support any input files, it can be fully integrated and customized according to personal needs. The downloaded source included some example projects, one of which is displayed in figure 5.1. This is a good example to show how FlexViz project can be used.

5.1.4 License

- FlexViz - Mozilla Public License 1.1. [moz]

5 Related Works

5.1.5 Screenshot

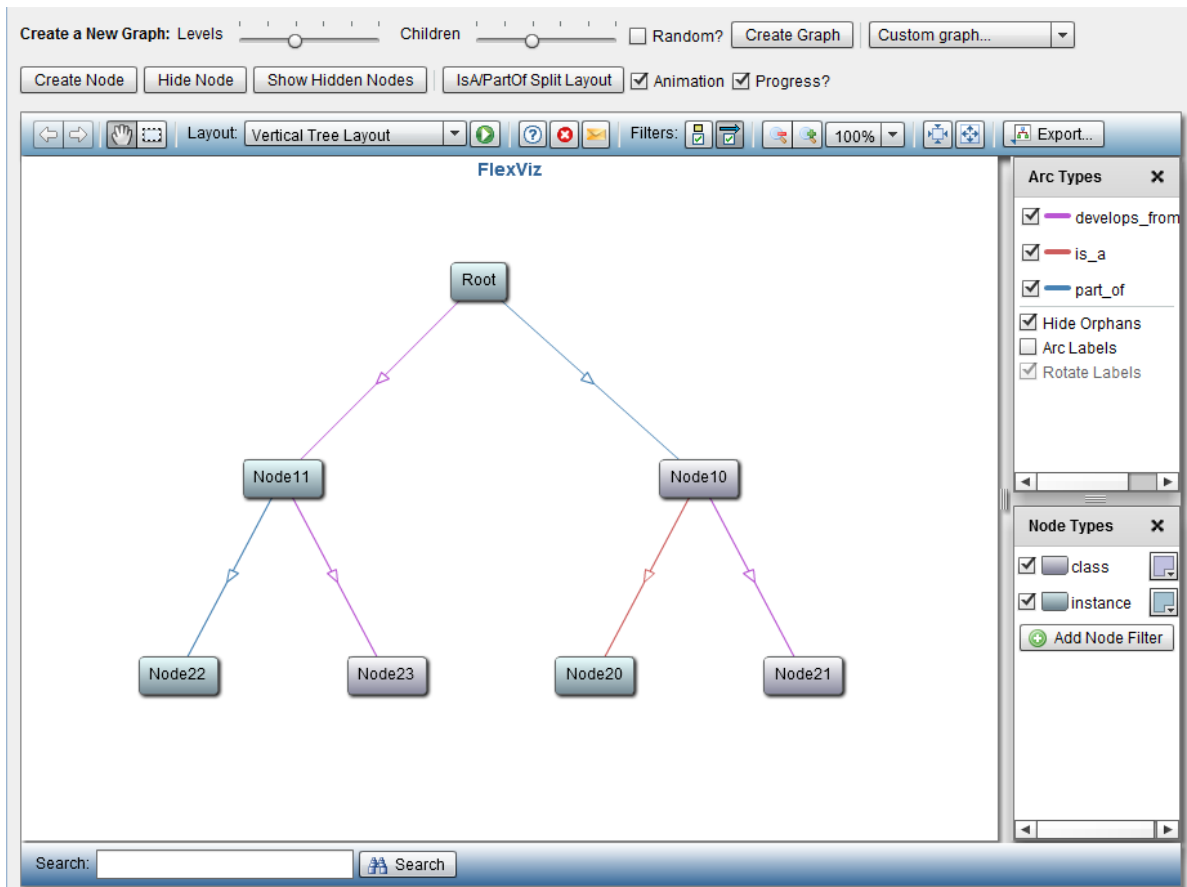


Figure 5.1: The GraphTest application which is contained in the downloadable source of FlexViz [fle15] 2.3.0 (random layout)

5.1.6 Evaluation

Table 5.1: Evaluation of FlexViz 2.3.0

	Visualization capability
	Look
	Usability
	Reliability
	Documentation
	Installation effort
	Overall impression
FlexViz	● ● ● ● ● ● ●

Table 5.2: Visualization capabilities of FlexViz 2.3.0

	Classes
	Enumeration
	Property restrictions
	Cardinality
	Intersection
	Union
	Complement
	subClassOf
	equivalentClass
	disjointWith
	Object properties
	Datatype properties
	Instances
	Annotations
FlexViz	● ● ● ● ● ● ● ● ● ● ●

5.2 RelFinder

5.2.1 Installation

RelFinder is used for linked data queries and mentioned in this paper for reason of completeness!

The RelFinder [LHSZ10] [rel14] is a web application which is either accessible on visualdataweb¹ or can be integrated on a webpage as flash by downloading the binary file²

5.2.2 Implementation details

Programming language ActionScript, XML

Libraries Adobe Flex

5.2.3 Description

The RelFinder reads from web resources like DBPedia³. It is capable of finding relationships between two objects (instances) based on a SPARQL [spa14] queries.

It displays objects as corner-rounded rectangles which are connected by directed arrows. These arrows are labeled with a property or a common class name.

5.2.4 Layout options

- Graph layout
- Pin nodes to a position
- Panning
- Hide/show parts of the visualization

5.2.5 License

RelFinder Sources - GNU General Public License [GPL89]

¹<http://www.visualdataweb.org/refinder/refinder.php>

²<http://www.visualdataweb.org/integrating.php>

³de.dbpedia.org/

5.2.6 Screenshot

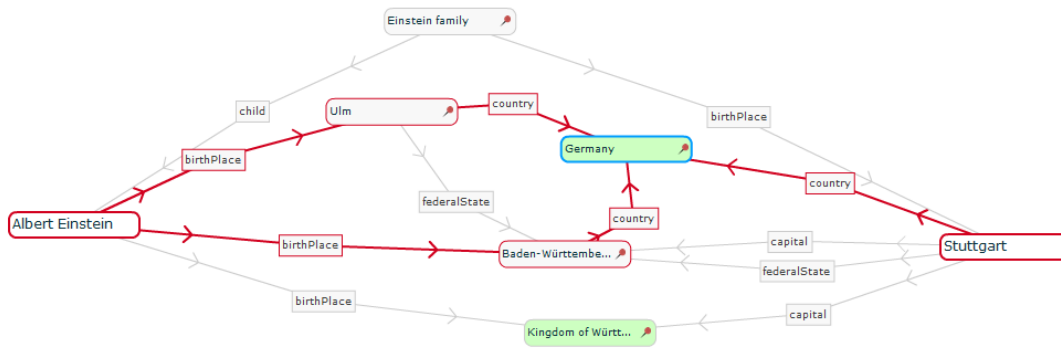


Figure 5.2: Visualization with RelFinder [rel14] 1.3.6

5.2.7 Evaluation

Table 5.3: Evaluation of RelFinder 1.3.6

RelFinder	●	●	●	●	●	●	●
	Visualization capability	Look	Usability	Reliability	Documentation	Installation effort	Overall impression

Table 5.4: Visualization capabilities of RelFinder 1.3.6

RelFinder	●																	
	Classes	Enumeration	Property restrictions	Cardinality	Intersection	Union	Complement	subClassOf	equivalentClass	disjointWith	Object properties	Datatype properties	Instances	Annotations				

6 Overview

As seen in previous chapters, the selected tools are not only tested against functional capabilities but also evaluated rationally with seven criteria. The evaluation is displayed in table 6.1 and the test result is summarized in sub-category 6.2. Further information about each tool is provided in table 6.3. Finally, table 6.4 lists three first ranking tools in each category, together with brief rational argumentation.

6.1 Evaluation

Table 6.1: Evaluation of all tested tools.

	Visualization cap.	Look	Usability	Reliability	Documentation	Installation effort	Overall impression
COE	●	●	●	●	●	●	●
CropCircles	●	●	●	●	●	●	●
FlexViz	●	●	●	●	●	●	●
GLOW	●	●	●	●	●	●	●
GrOWL	●	●	●	●	●	●	●
Jambalaya	●	●	●	●	●	●	●
KC-Viz	●	●	●	●	●	●	●
Knoocks	●	●	●	●	●	●	●
NavigOWL	●	●	●	●	●	●	●
Ontograf	●	●	●	●	●	●	●
OntologyVisualizer	●	●	●	●	●	●	●
OntoRama	●	●	●	●	●	●	●
Ontosphere3D	●	●	●	●	●	●	●
OWLGrEd	●	●	●	●	●	●	●
OWLPropViz	●	●	●	●	●	●	●
OWLviz	●	●	●	●	●	●	●
RelFinder	●	●	●	●	●	●	●
SOVA	●	●	●	●	●	●	●
TGViz	●	●	●	●	●	●	●
TopBraid Composer	●	●	●	●	●	●	●
<i>ProtégéVOWL</i>	●	●	●	●	●	●	●
WebVOWL	●	●	●	●	●	●	●

6.2 Overview of visualization capability

Table 6.2: Visualization capabilities of all tested tools.

	Classes	Enumeration	Property restrictions	Cardinality	Intersection	Union	Complement	subClassOf	equivalentClass	disjointWith	Object properties	Datatype properties	Instances	Annotations
COE	•	•	•	•	•		•	•	•	•	•	•	•	
CropCircles	•							•						
FlexViz	•							•					•	
GLOW	•				•			•			•	•	•	
GrOWL	•	•	•	•	•	•	•	•	•	•	•	•	•	•
Jambalaya	•		•					•			•		•	
KC-Viz	•							•			•			
Knoocks	•							•			•	•	•	
NavigOWL	•	•	•	•	•	•	•	•	•	•	•	•	•	•
OntoGraf	•							•			•		•	
OntologyVisualizer	•							•			•	•	•	
OntoRama	•	•		•	•			•	•		•	•	•	•
OntoSphere3D	•				•	•		•		•	•	•	•	
OWLGrEd	•	•	•	•	•	•	•	•	•	•	•	•	•	•
OWLPropViz	•							•	•	•	•			
OWLViz	•							•						
RelFinder	•										•		•	
SOVA	•	•	•	•	•	•	•	•	•	•	•	•	•	•
TGViz	•							•	•		•		•	
TopBraid Composer	•	•	•	•	•	•	•	•	•	•	•	•	•	•
<i>ProtégéVOWL</i>	•		•		•			•	•		•	•		
WebVOWL	•		•	•	•	•	•	•	•	•	•	•	•	

6.3 Tool information

In the below table 6.3, general information about each tool is provided. If the tool is a plug-in for a program, the "plug-in for" column is filled with this program. The "supports" column displays which types of file can be loaded into the tool. The "editor" column displays if the tool has a built-in editor. The "commercial" column is filled with YES if it is a commercial tool. Finally the "available" column tells if the tool is available or not.

Table 6.3: Information about examined tools.

	Plug-in for	Supports	Editor	Commercial	Available
COE	-	OWL/RDF	YES	NO	YES
CropCircles	SWOOP	OWL	-	NO	YES
FlexViz	-	-	-	YES	YES
GLOW	<i>Protégé 4.X</i>	OWL	NO	NO	YES
GrOWL	-	OWL-DL	YES	NO	NO
Jambalaya	<i>Protégé 3.2.1 - 3.4</i>	OWL	NO	NO	YES
KC-Viz	NeOn Toolkit	OWL	NO	NO	YES
Knooks	-	OWL Lite	YES	NO	YES
NavigOWL	<i>Protégé 4.1</i>	OWL	NO	NO	YES
OntoGraf	<i>Protégé 4.1 - 5.0</i>	OWL	NO	NO	YES
Ontology Visual.	NeOn Toolkit	OWL	NO	NO	YES
OntoRama	-	OWL	NO	NO	YES
OntoSphere3D	<i>Protégé 3.1.* / Eclipse</i>	OWL	NO	NO	YES
OntoTrack	-	OWL Lite	YES	NO	NO
OntoTrix	-	-	-	NO	YES
OWLGrEd	-	OWL	YES	NO (YES)	YES
OWLPropViz	<i>Protégé 4.0</i>	OWL	NO	NO	YES
OWLViz	<i>Protégé 3.4.4 - 5.0.0</i>	OWL	NO	NO	YES
RelFinder	-	RDF	NO	NO	YES
SOVA	<i>Protégé 4.0 - 4.3</i>	OWL	NO	NO	YES
TGViz	Protege 3	OWL	NO	NO	YES
VOM	<i>Protégé 2.X</i>	OWL-DL	-	YES	YES
TopBraid Comp.	-	OWL	NO	YES	YES
<i>ProtégéVOWL</i>	<i>Protégé 4</i>	OWL/RDF	NO	NO	YES
WebVOWL	-	VOWL-JSON	NO	NO	YES

6.4 Recommendations

Table 6.4: Recommendations for hierarchical visualizations.

Hierarchical Visualization		
Rank	Tool	Explanation
1	KC-Viz	Among the five mentioned hierarchical visualizers, KC-Viz has proven to be the most useful and reasonable one. In direct comparison with OWLViz it proved to be much better in editing the visualization, because it not only allows manual dragging of nodes but is also able to visualize hidden inheritance paths. Definitely a highlight is it's „key concept extraction algorithm“with which one can easily reduce the displayed classes to just the most important key concepts.
2	OWLViz	Although not our first choice, OWLViz definitely provides the most intuitive view what makes it very nice to look at. OWLViz provides a static class tree with which one can only interact via a context menu or by triggering a new layouting of a subbranch by clicking on an node.
3	CropCircles	CropCircles provides a beautiful and at the moment unique approach. It has several advantages like discussed in the article, but is in the third ranking, because it is not suitable for large ontologies and provides only one static visualization.

Table 6.5: Recommendations for graph visualizations.

Graph Visualization		
Rank	Tool	Explanation
1	OWLGrEd	In order to visualize a large amount of information, a clear and structured layout is as important as a logic placement of information. Both are well provided in OWLGrEds UML-based layout which positively distinguishes it from its opponents. OWLGrEd also provides multiple options to edit and to navigate the displayed graph.
2	WebVOWL	In comparison with OWLGrEd WebVOWL is not able to visualize that much information and is more likely to be overcrowded with big structures. On the other hand WebVOWL is certainly a better solution for the visualization of a complex structure with only a few classes combined with many literals, datatypes and properties. Another advantage of WebVOWL is that components can be identified very easily due to a very clear visual distinction between intersections, unions or disjoints.
3	TGViz	TGViz is simpler than WebVOWL. It provides a hyperbolic view on the ontology to focus on parts of the structure. It is also possible to show parts of the ontology, which is a helpful feature when working with large structures. It is also possible to filter the whole ontology so that only the necessary components and edges are shown.

Table 6.6: Recommendations for universal visualizations.

Universal Visualization		
Rank	Tool	Explanation
1	Jambalaya	Among the three candidates, Jambalaya is the most convincing one. It was the only tool that came up with six different views, from which all worked in the test.
2	SOVA	SOVA combines a high visualization capability with multiple visual possibilities. It would have even been top-rated but unluckily it failed to load the Pizza ontology. If the loading function would have worked better, SOVA would be a very robust visualization.

Bibliography

- [Ala03] H. Alani. TGVizTab: An ontology visualisation extension for Protégé. 2003. (Cited on page 53)
- [apa04] Apache License, Version 2.0. <http://www.apache.org/licenses/LICENSE-2.0>, 2004. (Cited on page 90)
- [BB06] A. Bosca, D. Bonino. OntoSphere3D: A Multidimensional Visualization Tool for Ontologies. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications, DEXA '06*, S. 339–343. IEEE Computer Society, Washington, DC, USA, 2006. doi:10.1109/DEXA.2006.106. URL <http://dx.doi.org/10.1109/DEXA.2006.106>. (Cited on page 39)
- [BBČ⁺10] J. Bārzdīņš, G. Bārzdīņš, K. Čerāns, R. Liepiņš, A. Sproģis. OWLGrEd: a UML Style Graphical Notation and Editor for OWL 2. In *Proceedings 7th International Workshop “OWL: Experience and Directions”(OWLED-2010)*, http://www.webont.org/owled/2010/papers/owled2010_submission_5.pdf. Citeseer, 2010. (Cited on page 49)
- [BHH⁺14] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, L. A. S. Stein. OWL Specification, 2014. URL <http://www.w3.org/TR/owl-ref/>. (Cited on page 63)
- [bio15] FlexoViz, 2015. URL <http://keg.cs.uvic.ca/ncbo/flexviz/FlexoViz.html>. (Cited on page 97)
- [BPLL11] B. Bach, E. Pietriga, I. Liccardi, G. Legostaev. OntoTrix: A Hybrid Visualization for Populated Ontologies. In *Proceedings of the 20th International Conference Companion on World Wide Web, WWW '11*, S. 177–180. ACM, New York, NY, USA, 2011. doi:10.1145/1963192.1963283. URL <http://doi.acm.org/10.1145/1963192.1963283>. (Cited on page 60)
- [bsd] The BSD 2-Clause License. <http://opensource.org/licenses/BSD-2-Clause>. (Cited on pages 61 and 80)
- [bsd99] The BSD 3-Clause License. <http://opensource.org/licenses/BSD-3-Clause>, 1999. (Cited on pages 16 and 57)
- [CDR04] J. J. Carroll, J. De Roo. OWL web ontology language test cases. *W3C recommendation*, 10, 2004. URL <http://www.w3.org/TR/owl-test/>. (Cited on page 14)

- [coe15] Concept-map Ontology Environment Homepage, 2015. URL <http://www.ihmc.us/groups/coe/>. (Cited on pages 9, 75, 76 and 77)
- [DHS⁺07] N. Drummond, M. Horridge, R. Stevens, C. Wroe, S. Sampaio. Pizza ontology, 2007. URL <http://130.88.198.11/co-ode-files/ontologies/pizza.owl>. (Cited on pages 8, 9, 13, 14, 19, 22, 25, 26, 33, 36, 37, 40, 41, 46, 47, 50, 51, 54, 65, 76, 77, 81, 82, 84, 91, 94 and 95)
- [DVR07] M. Dumontier, N. Villanueva-Rosales. Three-Layer OWL Ontology Design. In *WoMO*. Citeseer, 2007. (Cited on page 97)
- [EN⁺04] J. Ellson, S. North, et al. Graphviz-graph visualization software, 2004. URL <http://www.graphviz.org>. (Cited on page 24)
- [ep1] Eclipse Public License. <https://www.eclipse.org/legal/epl-v10.html>. (Cited on pages 19, 25, 32, 40, 46 and 90)
- [ERG02] P. Eklund, N. Roberts, S. Green. OntoRama: Browsing RDF Ontologies Using a Hyperbolic-Style Browser. In *Proceedings of the First International Symposium on Cyber Worlds (CW'02)*, CW '02, S. 0405–. IEEE Computer Society, Washington, DC, USA, 2002. URL <http://dl.acm.org/citation.cfm?id=794192.794831>. (Cited on pages 8, 56, 57 and 58)
- [Fal10] S. Falconer. OntoGraf. *Protégé Wiki*, 2010. URL <http://protegewiki.stanford.edu/wiki/OntoGraf>. (Cited on page 35)
- [fla15] Adobe Flash Builder, 2015. URL <http://www.adobe.com/de/products/flash-builder-family.html>. (Cited on page 97)
- [fle15] FlexViz Homepage, 2015. URL <http://thechiselgroup.org/2010/07/06/flexviz/>. (Cited on pages 9, 97 and 98)
- [Glo15] GLOW Visualization, 2015. URL <http://www.glowvis.org>. (Cited on pages 9, 79, 81 and 82)
- [GPL89] GNU General Public License Version 1.0. <http://www.gnu.org/licenses/old-licenses/gpl-1.0>, 1989. (Cited on page 100)
- [Har14] A. Harth. NeOn-Toolkit, 2014. URL <http://www.neon-foundation.org/>. (Cited on pages 8, 18, 19, 31 and 33)
- [HCL05] J. Heer, S. Card, J. Landay. Prefuse: a toolkit for interactive information visualization. *Proceedings of the SIGCHI Conference on Human Factors in Computing System CHI's 05*, S. 421–430, 2005. (Cited on pages 63 and 72)
- [HEM⁺05] P. Hayes, T. C. Eskridge, M. Mehrotra, D. Bobrovnikoff, T. Reichherzer, R. Saavedra. COE: Tools for collaborative ontology development and reuse. In *Knowledge Capture Conference (K-CAP)*, Band 2005. 2005. (Cited on pages 9, 75, 76 and 77)
- [HLTR14] A. Hussain, K. Latif, A. Tariq Rextin. NavigOwl, 2014. URL <http://protegewiki.stanford.edu/wiki/NavigOWL>. (Cited on page 68)

- [Hor05] M. Horridge. Owlviz-a visualisation plugin for the protege owl plugin, 2005. URL <http://www.co-ode.org/downloads/owlviz1>. (Cited on page 24)
- [HRFH12] W. Hop, S. de Ridder, F. Frasincar, F. Hogenboom. Using Hierarchical Edge Bundles to Visualize Complex Ontologies in GLOW. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC '12*, S. 304–311. ACM, New York, NY, USA, 2012. doi:10.1145/2245276.2245338. URL <http://doi.acm.org/10.1145/2245276.2245338>. (Cited on pages 9, 79, 81 and 82)
- [kno15] Knoocks (Knowledge Blocks), 2015. URL <http://enomisk.net/knoocks/>. (Cited on pages 8, 28 and 29)
- [KW10] S. Kriglstein, G. Wallner. Knoocks-a visualization approach for owl lite ontologies. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*, S. 950–955. IEEE, 2010. (Cited on pages 8, 28 and 29)
- [KWV07] S. Krivov, R. Williams, F. Villa. GrOWL: A Tool for Visualization and Editing of OWL Ontologies. *Web Semant.*, 5(2):54–57, 2007. doi:10.1016/j.websem.2007.03.005. URL <http://dx.doi.org/10.1016/j.websem.2007.03.005>. (Cited on page 72)
- [LGP91] GNU Lesser General Public License Version 2. <https://www.gnu.org/licenses/old-licenses/lgpl-2.0>, 1991. (Cited on pages 25, 40 and 85)
- [LGP07] GNU Lesser General Public License Version 3. <https://www.gnu.org/licenses/lgpl.html>, 2007. (Cited on page 85)
- [LHSZ10] S. Lohmann, P. Heim, T. Stegemann, J. Ziegler. The RelFinder User Interface: Interactive Exploration of Relationships between Objects of Interest. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI 2010)*, S. 421–422. ACM, New York, NY, USA, 2010. URL <http://doi.acm.org/10.1145/1719970.1720052>. (Cited on page 100)
- [LLMN14] V. Link, S. Lohmann, E. Marbach, S. Negru. WebVOWL, 2014. URL <http://vowl.visualdataweb.org/webvowl.html>. (Cited on pages 9, 63 and 66)
- [LN03] T. Liebig, O. Noppens. OntoTrack: Fast browsing and easy editing of large ontologies. In *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003)*. Florida, USA, 2003. (Cited on pages 8, 15 and 17)
- [mag14] No Magic, 2014. URL <http://www.nomagic.com/>. (Cited on page 43)
- [mit88] Mit-License. <http://opensource.org/licenses/mit-license.php>, 1988. (Cited on pages 22 and 64)
- [moz] Mozilla Public License Version 1.1. <https://www.mozilla.org/MPL/1.1/>. (Cited on pages 25, 36, 40, 46, 54, 80, 85 and 97)

- [MPGP⁺12] E. Motta, S. Peroni, J. M. Gómez-Pérez, M. d’Aquin, N. Li. Visualizing and Navigating Ontologies with KC-Viz. In *Ontology Engineering in a Networked World*, S. 343–362. Springer, 2012. (Cited on page 18)
- [NLB14] S. Negru, S. Lohmann, D. Bold. ProtégéVOWL: VOWL Plugin for Protégé, 2014. URL <http://vowl.visualdataweb.org/protegevowl.html>. (Cited on pages 8, 9, 63, 65 and 66)
- [NLHE14] S. Negru, S. Lohmann, F. Haag, T. Ertl. VOWL 2: User-Oriented Visualization of Ontologies. In *Proceedings of 19th Conference on Knowledge Acquisition, Modeling and Management, EKAW ’14*, S. 266–281. Springer, 2014. (Cited on page 63)
- [owl14] Graphical Ontology Editor - OWLGrEd, 2014. URL <http://owlgred.lumii.lv/>. (Cited on pages 8, 49, 50 and 51)
- [peo14] The People Ontology, 2014. URL <http://owl.man.ac.uk/2006/07/sssw/people.owl>. (Cited on pages 9, 84, 86, 87 and 88)
- [PM14] S. Peroni, E. Motta. KC-Viz, 2014. URL <http://neon-toolkit.org/wiki/KC-Viz>. (Cited on page 18)
- [Pro15] Protege-OWL, 2015. URL <http://protegewiki.stanford.edu/wiki/Protege-OWL>. (Cited on page 68)
- [rel14] RelFinder - Visual Data Web, 2014. URL <http://www.visualdataweb.org/relfinder.php>. (Cited on pages 9, 100 and 101)
- [ReO] *International Journal of Research in Computer Science*, Kapitel A Comparative Study of Recent Ontology Visualization Tools With a Case of Diabetes Data. White Globe Publications. (Cited on page 12)
- [SMS⁺01] M.-A. Storey, M. Musen, J. Silva, C. Best, N. Ernst, R. Fergerson, N. Noy. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé. In *Workshop on Interactive Tools for Knowledge Capture (K-CAP-2001)*. Citeseer, 2001. (Cited on page 93)
- [spa14] SPARQL 1.1 Entailment Regimes, 2014. URL <http://www.w3.org/TR/rdf-sparql-query/>. (Cited on pages 97 and 100)
- [swo14] Swoop, 2014. URL <https://code.google.com/p/swoop/>. (Cited on pages 8, 21 and 22)
- [top14] TopQuadrant, 2014. URL <http://www.topquadrant.com/>. (Cited on pages 9, 90 and 91)
- [vom13] Visual Ontology Modeler - Thematix Partners, 2013. URL <http://thematix.com/tools/vom/>. (Cited on pages 8, 43 and 44)

- [WP06] T. D. Wang, B. Parsia. CropCircles: Topology Sensitive Visualization of OWL Class Hierarchies. In *Proceedings of the 5th International Conference on The Semantic Web, ISWC'06*, S. 695–708. Springer-Verlag, Berlin, Heidelberg, 2006. doi:10.1007/11926078_50. URL http://dx.doi.org/10.1007/11926078_50. (Cited on page 21)

All links were last checked on the 06.01.2015.

Declaraton

We hereby declare that the work presented in this thesis is entirely our own and that we did not use any other sources and references than the listed ones. We have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. We have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature