UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ISABEL CRISTINA SIQUEIRA DA SILVA

# Visualization of Intensional and Extensional Levels of Ontologies

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Profa. Dra. Carla Maria Dal Sasso Freitas
Advisor

Prof. Dr. Giuseppe Santucci
Coadvisor

Porto Alegre, June 2014

**CIP – CATALOGING-IN-PUBLICATION**

Silva, Isabel Cristina Siqueira da

Visualization of Intensional and Extensional Levels of Ontologies / Isabel Cristina Siqueira da Silva. – Porto Alegre: PPGC da UFRGS, 2014.

101 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2014. Advisor: Carla Maria Dal Sasso Freitas; Coadvisor: Giuseppe Santucci.

1. Information Visualization. 2. Ontology. 3. Interaction. 4. Visual Analytics. I. Freitas, Carla Maria Dal Sasso. II. Santucci, Giuseppe. III. Título.

# ACKNOWLEDGEMENTS

Many people and institutions help me in reaching this point, and I want to thank them all.

Thanks to the Graduate Program, PPGC/UFRGS, for accepting me as a PhD student, and providing me the opportunity to win this challenge in my academic life as a researcher.

Thanks to Capes, for funding my "sandwich PhD" stay in Italy, which allowed me to have a unique experience both in living and studying.

Regarding this, I wish to thank my co-advisor Giuseppe Santucci, who made my stay in Rome filled with new knowledge. He was very friendly and dedicated, and contributed a lot for the evolution of my PhD. thesis.

Thank you, Carla Freitas, my advisor for thirteen years! Thank you very much for sharing your experience and your scientific knowledge with me, as well as your friendship and generosity in guiding me along this long journey.

Thanks also to Sandro Fiorini and Joel Carbonera, which have contributed a lot in my research about creation and manipulation of ontologies.

Special thanks to Leo, my husband, my friend. Love you Leonardo and thank you for all the support you gave me!

Besides Leonardo, another inseparable partner during the final phase of this thesis was our beloved son Henrique, and his little brother or sister that is coming soon... Thanks!

Finally, thanks to my family, friends, colleagues, neighbors: all those who in one way helped me with a word of encouragement.

**Visualização de Níveis Intensional e Extensional de Ontologias**

# RESUMO


Técnicas de visualização de informaçoes têm sido usadas para a representação de ontologias visando permitir a compreensão de conceitos e propriedades em domínios específicos. A visualização de ontologias deve ser baseada em representaccões gráficas efetivas e téquinas de interação que auxiliem tarefas de usuários relacionadas a diferentes entidades e aspectos. Ontologias podem ser complexas devido tanto à grande quantidade de níveis da hierarquia de classes como também aos diferentes atributos.

Neste trabalho, propoẽ-se uma abordagem baseada no uso de múltiplas e coordenadas visualizações para explorar ambos os níceis intensional e extensional de uma ontologia. Para tanto, são empregadas estruturas visuais baseadas em árvores que capturam a característica hierárquiva de partes da ontologia enquanto preservam as diferentes categorias de classes.

Além desta contribuição, propõe-se um inovador emprego do conceito "Degree of Interest"de modo a reduzir a complexidade da representação da ontologia ao mesmo tempo que procura direcionar a atenção do usuádio para os principais conceitos de uma determinada tarefa. Através da análise automáfica dos diferentes aspectos da ontologia, o principal conceito é colocado em foco, distinguindo-o, assim, da informação desnecessária e facilitando a análise e o entendimento de dados correlatos.

De modo a sincronizar as visualizações propostas, que se adaptam facilmente às tarefas de usuários, e implementar esta nova proposta de cćalculo baseado em "Degree of Interest", foi desenvolvida uma ferramenta de visualização de ontologias interativa chamada OntoViewer, cujo desenvolvimento seguiu um ciclo interativo baseado na coleta de requisitos e avaliações junto a usuários em potencial. Por fim, uma última contribuição deste trabalho é a proposta de um conjunto de "guidelines"visando auxiliar no projeto e na avaliação de téncimas de visualização para os níceis intensional e extensional de ontologias.



**Palavras-chave:** Visualização de Informações, Ontologia, Interação, Visualização Analítica.

**Visualization of Intensional and Extensional Levels of Ontologies**

# ABSTRACT

Visualization techniques have been used for the representation of ontologies to allow the comprehension of concepts and properties in specific domains. Techniques for visualizing ontologies should be based on effective graphical representations and interaction techniques that support users tasks related to different entities and aspects. Ontologies can be very large and complex due to many levels of classes' hierarchy as well as diverse attributes.

In this work we propose a multiple, coordinated views approach for exploring the intensional and extensional levels of an ontology. We use linked tree structures that capture the hierarchical feature of parts of the ontology while preserving the different categories of classes.

We also present a novel use of the Degree of Interest notion in order to reduce the complexity of the representation itself while drawing the user attention to the main concepts for a given task. Through an automatic analysis of ontology aspects, we place the main concept in focus, distinguishing it from the unnecessary information and facilitating the analysis and understanding of correlated data.

In order to synchronize the proposed views, which can be easily adapted to different user tasks, and implement this new Degree of Interest calculation, we developed an interactive ontology visualization tool called OntoViewer. OntoViewer was developed following an iterative cycle of refining designs and getting user feedback, and the final version was again evaluated by ten experts. As another contribution, we devised a set of guidelines to help the design and evaluation of visualization techniques for both the intensional and extensional levels of ontologies.

**Keywords:** Information Visualization, Ontology, Interaction, Visual Analytics.

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| DoI | Degree of interest |
| InfoVis | Information Visualization |
| IVTs | Information Visualization Techniques |
| OWL | Web Ontology Language |

# CONTENTS

# 1 INTRODUCTION

Along the last decade, there has been a gradual increase of information made available through several and different media. This increasing volume of information is originated from many sources and represented in different formats, and efficient methods for information retrieval are necessary in order to allow integrating such data from different systems, especially when they are about the same or related entities or phenomena.

Ontologies are representations of categories of things that exist or may exist in different domains, i.e., an ontology is a catalogue of the types of things that are assumed to exist in a given domain of interest D from the perspective of a person P who uses a language L for the purpose of talking about D (SOWA, 2006).

Ontologies can help the discovery of non-explicit information about data and ensure interoperability between systems. They allow sharing the common understanding of the structure of information among people or software agents (NOY; MCGUINNESS, 2001), separating domain knowledge from the operational knowledge, making domain assumptions explicit and enabling reuse. However, the categorization of information in a specific domain only achieves its main goal if presented in an efficient way, allowing user interaction for supporting the desired tasks.

## 1.1 Visualization of Ontologies

Information Visualization (InfoVis), Visual Analytics and Human Computer-Interaction can speed up the comprehension of ontologies.

Visualization techniques provide help for the analysis of ontology concepts streamlining the process of insight about different correlated data. When we analyze an image, we activate our perceptual mechanisms to identify patterns and perform segmentation of elements. The user must perceive the information presented in the display, and the understanding involves cognitive processes. An image can be ambiguous due to lack of relevant information or excess of irrelevant information.

When the searched information can be placed in focus, distinguishing it from the unnecessary information, the understanding of correlated data is facilitated. In order to do that, designers of visualization systems should consider two main issues: the mapping of information for a graphical representation in order to facilitate its interpretation by the users, and means to limit the amount of information that users receive, while keeping them "aware" of the total information space and reducing cognitive effort.

Different InfoVis techniques can be applied, and choosing one technique out of the many available depends on the type of information being handled and tasks that must be performed by the user. Shneiderman (SHNEIDERMAN, 1996) categorizes visualization methods based on two criteria, the data type of the objects to be represented in the interface (linear, planar, volumetric, temporal, multidimensional, tree, network, workspace) and the task typology (overview, zoom, filter, details-on-demand, relate, history, extract).

Graphs (networks in Shneiderman's classification) are the most intuitive form of visualizing the relationships between concepts of ontologies by their both hierarchical and relational characteristics. Relationships can be displayed among expanded nodes but the overlapping edges can be a problem for the efficiency of information display. An interactive graph or tree solves part of the problem, allowing the user to highlight the information in focus through selection, but the overlapping edges are still a problem. Moreover, as the ontology grows, incorporating new concepts (and their relationships) increases the visualization complexity.

The main problems of the current solutions for ontologies visualization are common to any graph visualization: problems of scale versus amount of information that need to be presented. Moreover, there are different needs coming from different users categories, but the two levels (intensional and extensional) should be visualized using a single tool, and this leads to important design and evaluation issues. As for design, a promising alternative way is to use different visualization techniques in multiple and synchronized views.

Katifori et al. (2007) addresses InfoVis techniques that can be used to display two and/or three-dimensional ontologies: indented lists, hierarchies (trees and graphs), zooming, space filling (treemaps, information slices), focus+context and landscapes (see Table 1.1 for more details). Along with the techniques, these authors discuss tools that enable the visualization and interaction with ontologies.

Interactive ontology visualizations need to be efficient and allow rapid comprehension of concepts and their entities, i.e., the intensional level (concepts, relationships, attributes) and the extensional level (instances). Katifori (KATIFORI et al., 2007) confirms that it is not simple to create a visualization that displays effectively all the information, and, at the same time, allows the user to easily perform various operations on the ontology.

Then, the challenge is to define the best way to represent relationships between categorized concepts mainly because each concept can have a number of related attributes and relationships.

## 1.2 Objectives and Expected Contribution

This study aims at investigating the application of InfoVis techniques for aiding the creation, manipulation and analysis of ontologies as well as the results obtained by using them.

We have built a system for visual exploration of ontologies. The system relies on multiple coordinated views (BALDONADO; WOODRUFF; KUCHINSKY, 2000) based on different hierarchical visualization techniques in order to help users to understand complex relationships among different components of an ontology as well as features and aspects that might be semantically important.

Moreover, in order to help the analysis of large ontologies, we employ a suppression technique (FURNAS, 1986) based on the notion of Degree of interest (DoI) that, from the automatic analysis of an ontology's intension (concepts and its relationships and attributes) and extension (instances of concepts), extracts knowledge about the relevance of concepts and relationships according to the user task. That technique allows exploring large ontologies focusing on a main concept and having the view of the most relevant concepts and relationships automatically computed and displayed.

In summary, the contributions of this work are:

Table 1.1: Tasks related to ontologies and information visualization techniques, adapted from (KATIFORI et al., 2007).

| Task | Description | VI Techniques |
|---|---|---|
| Overview | Gain an overview of the entire collection | Trees and graphs, 3D landscapes, treemaps (space filling) |
| Zoom | Zoom in on items of interest | Indented lists, trees and graphs, 3D landscapes |
| Details-on-demand | Select an item or group and get details when needed | Trees and graphs, 3D landscapes |
| Filter | Filter out uninteresting items | Indented lists, trees and graphs |
| Relate | View relationships among items | Indented lists, trees and graphs, zooming, 3D landscapes |
| History | Keep a history of actions to support undo, replay and progressive refinement | - |

- the proposal of an ontology visualization scheme that employs multiple coordinated views to improve the analysis of the ontology hierarchy, classes, attributes, relationships and instances. For that, both focus+context and overview+detail views are provided;

- the proposal of automatic analysis of the ontology's concepts, relationships, and instances, based on a novel definition of DoI that can be easily adapted to different users' tasks. The DoI is used to automatically compute a task oriented view of the ontology;

- the proposal of guidelines to help the design and evaluation of visualization techniques for both the intensional and extensional levels of ontologies.

## 1.3 Overview

We have investigated the main issues in ontology creation and visualization, and have already reported those results (SILVA; NETTO; FREITAS, 2009). In that study we analyzed the InfoVis techniques employed for visualization and interaction with ontologies such as those that could assist the user in the involved processes as well as display the inferred results. Then, we proposed our first approach for ontology visualization, a 2.5D interactive visualization of classes hierarchy and relationships.

The 2.5D visualization was presented to users used to work with tools and methods related to conceptual modeling and ontologies. Aiming at improving our first proposal (SILVA; FREITAS, 2011a), we interviewed the users and gathered requirements. Based on the requirements, we prototyped an ontology visualization tool, with multiple views, and made it available (SILVA; FREITAS, 2011b,c). The visualization underwent evaluation rounds by expert users.

After this qualitative evaluation, we extended the tool with coordinated views, and applied concepts of Visual Analytics in order to automatize the analysis of concepts and properties of the ontology (SILVA; SANTUCCI; FREITAS, 2012). A study involving the investigation and proposition of guidelines to help the design and evaluation of visualization techniques for both the intensional and extensional levels of ontologies was also developed (SILVA; FREITAS; SANTUCCI, 2012).

Lastly, after some new adjustments to cope with more complex ontologies, a final round of evaluation was performed.

Besides this introductory chapter, the text is organized as follows. Chapter 2 presents the background of this work, detailing fundamental concepts about ontologies and forms of representation, and giving a brief overview of InfoVis techniques. Chapter 3 discusses related works which include studies dealing with the visualization of ontologies. Chapter 4 presents the methodology applied in this work addressing the development and design cycle of our ontologies visualization approach. Chapter 5 describes the core of our work: it presents the guidelines proposed for design and evaluation of visualization tools for ontologies, and OntoViewer - our prototype for testing the multiple and coordinated views proposed for ontology visualization, the model for ontology analysis, the prototype validation and associated results. Chapter 6 discusses the results achieved with case studies and evaluation with users, and Chapter 7 contains the conclusions and draws future work.

# 2 BACKGROUND

*In this chapter, we briefly review fundamental concepts of ontologies. We also introduce concepts of information visualization that are important in our work.*

## 2.1 Ontologies

Ontologies are formal and explicit specifications of shared conceptualizations that capture and explain the vocabulary used in semantic applications, which are systems and services that "understand" meanings, and put knowledge to work (BREITMAN, 2005). This fact refers to the way people think about some part of the world one needs to represent for some purpose (GRUBER, 1993).

The explicit specification relates concepts and relationships, which must be supplied in accordance with specific and well-defined terms. In this context, the conceptual modelling represents objects or entities of the real world, i.e., the properties and relationships between them. The intention is to facilitate the understanding of the facts in a reality independent of the technology used to implement the solution.

Conceptual modelling is not a trivial task, because there is a "semantic gap" between the real world and the computer world. A model is an abstraction of the reality, which emphasizes specific characteristics, and represents both the environment vision and part of the whole, allows the gradual approach of complexity, and is useful in organizing information. Thus, we start with an informal representation of the reality followed by an intermediary representation and, finally, we obtain the formal representation, trying to describe the world as a form that the computer "understands" and within existing limitations.

An ontology is proposed from the definition of elements it is intended to describe, these elements referring to classes (concepts) in the domain of interest, relationships (object properties), attributes (data properties) and instances (individuals) of classes. In the example of Figure 2.1, we have:

- Classes: *Person, Car, Country*;
- Relationships between classes: *hasCar, hasSibling and livesIn*;
- Attributes of classes: *Name, Age, Job, Color, Fuel, Continent, Currency*;
- Instances of classes: *John, Mary, Ford, Honda, Fiat, Brazil, Italy*.

Different fields apply ontologies in order to facilitate the analysis of data from a variety of sources. In urban planning, for example, ontologies allow the standardization of the vocabulary used by urban planners and interoperability among databases (see Figure 2.2).

An ontology defines a common vocabulary for researchers who need to share information in a domain, including machine-interpretable definitions of basic concepts in the domain and relations among them (NOY; MCGUINNESS, 2001). Some reasons for

Figure 2.1: Ontology example: classes, relationships, attributes and instances

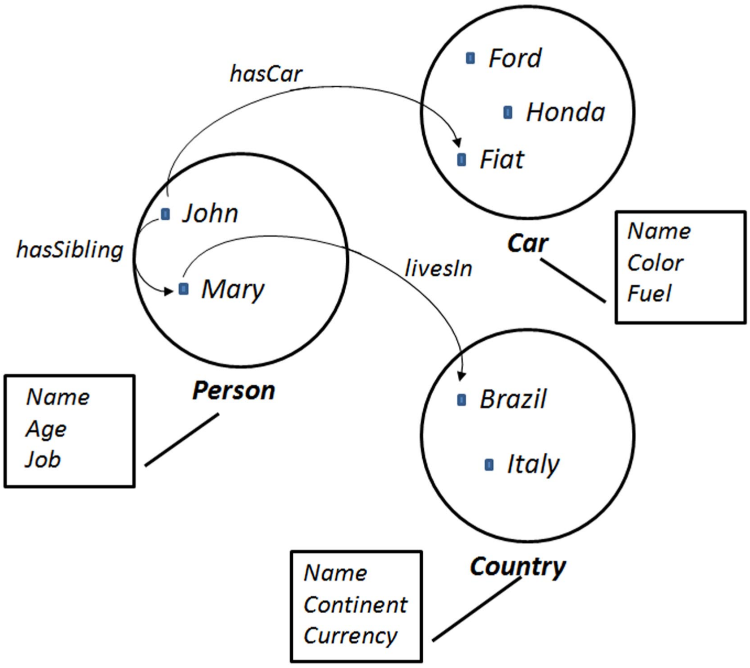
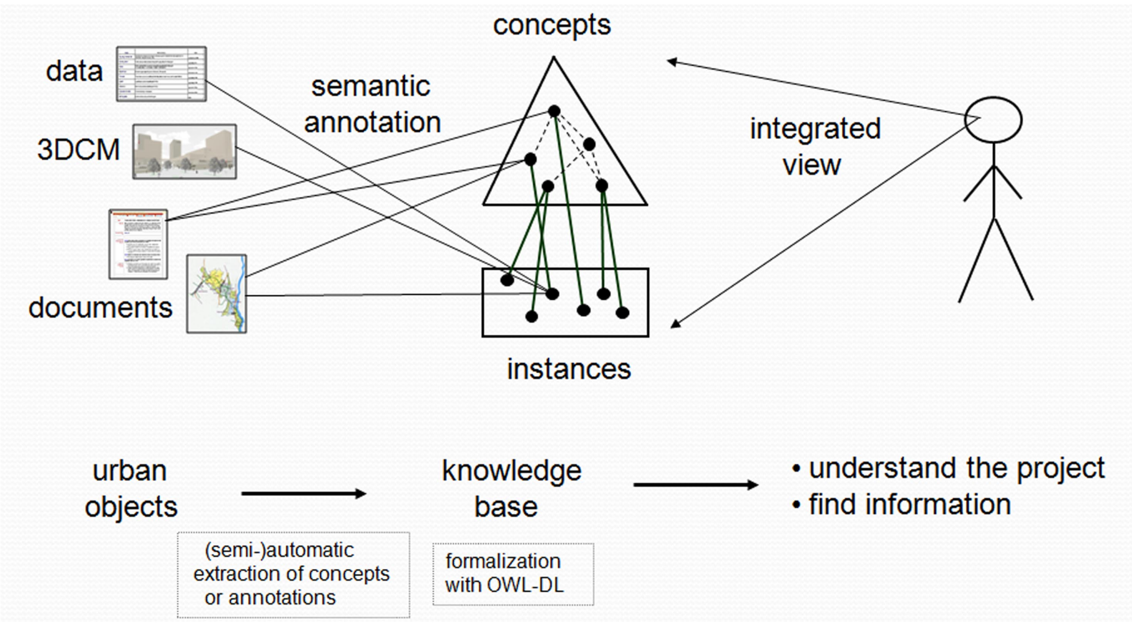



Figure 2.2: Example of the use of an ontology for semantic integration of the information about an urban project represented in different media as databases, documents, 3D city models, etc.




Adapted from (MÉTRAL; FALQUET; VONLANTHEN, 2007)

developing an ontology are:

- Sharing of common understanding of the structure of information among people or software agents;
- Enabling reuse of domain knowledge;
- Making domain assumptions explicit;

- Separation of domain knowledge from operational knowledge;
- Analysis of domain knowledge.

Ontologies have two main levels: intensional and extensional levels. The *intensional* level comprises the classes, relationships and attributes, while the *extensional* level corresponds to the instances of such classes that are created for a specific task or application.

From a knowledge engineer point of view, the intensional level of ontologies is more important. During the process of creating an ontology, users may want to visualize different aspects due to specific demands that arise in certain stages of development, for example, checking the range of an object property. However, the development of ontologies is still a traditional activity, with no defined workflow and directly influenced by the domain.

On the other hand, the extensional level seems to be more interesting from the point of view of professionals that maintain knowledge databases. For these ones, it seems necessary to have views of the instances distribution allowing to see how attribute values are distributed and to perform quick visual queries about instances, observing trends in values, for example.

We must consider that ontologies can encapsulate a large amount of information (hundreds of thousands of classes and relationships, for example). Moreover, this large volume of information can be segmented into several distinct types (classes, attributes with different values, relationships between types and properties). Usually, users do not want to analyze all these types simultaneously, due to the cognitive overload it would arise.

## 2.2 Information Visualization

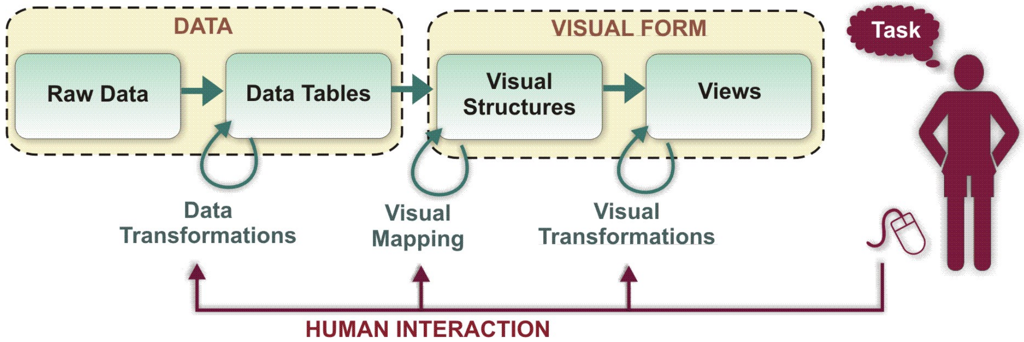
Visualization is a form of communication that transcends application and technological boundaries because it offers a way to see the unseen (DEFANTI; BROWN; MCCORMICK, 1989). According to Ware (WARE, 2008), Visualization used to be mental images that people formed while they thought about something, but now the term is related to a graphical representation of some data or concept.

In this context, Card et al. (CARD; MACKINLAY; SHNEIDERMAN, 1999) divide Visualization in two main areas: Scientific Visualization, where we have data sets associated to some geometry (scalar values, 2D and 3D data flow, etc.), and Information Visualization, related to data without an implicit geometry (for example, tables, trees and graphs).

The first use of the expression "Information Visualization" (InfoVis) was in 1980 by Xerox Parc researchers for a new discipline concerned with the creation of visual artifacts aimed at amplifying the cognition (MAZZA, 2009). Indeed, InfoVis techniques amplify cognition and reduce exploration time of a data set, allowing the recognition of patterns and facilitating inferences about different concepts. Card et al. (CARD; MACKINLAY; SHNEIDERMAN, 1999) propose a InfoVis reference model showed in Figure 2.3 which presents the creation of a visual artifact as a process that can be model through a sequence of successive stages: preprocessing and data transformations, visual mapping and view creation.

In the process of visual mapping, we have two main problems: defining which visual structures should be used to map the data, and defining the location of such visual structures in the display area. These problems involve the correct proposition of visual structures, accuracy, color and location. For solving that, we need to respond the following questions:

Figure 2.3: Visualization reference model proposed by Card et al. (CARD; MACKINLAY; SHNEIDERMAN, 1999)




Adapted from (CARD; MACKINLAY; SHNEIDERMAN, 1999)

- What is the problem?
- What is the nature of the data? (Quantitative, ordinal, nominal)
- How many dimensions are involved? (1D, 2D, 3D, nD)
- What are the data structures? (Linear, network, hierarchical, temporal, spatial)
- What kind of interaction is required?

The next subsections discuss the main techniques related to data structures, visual representations and interaction.

### 2.2.1 Data Structures and Visual Representations

Herein we divide data structures in five main groups: linear, network, hierarchical, temporal and spatial.

Linear data refers to data organized in the form of simple tables, which can be uni-, bi-, tri- or multivariate according to the number of attributes of the elements in the dataset. For bi-, tri- and multivariate data, there is no need of a dependency relationship between the attributes that describe the same element. Figures 2.4, 2.5 and 2.6 present, respectively, examples of linear data.

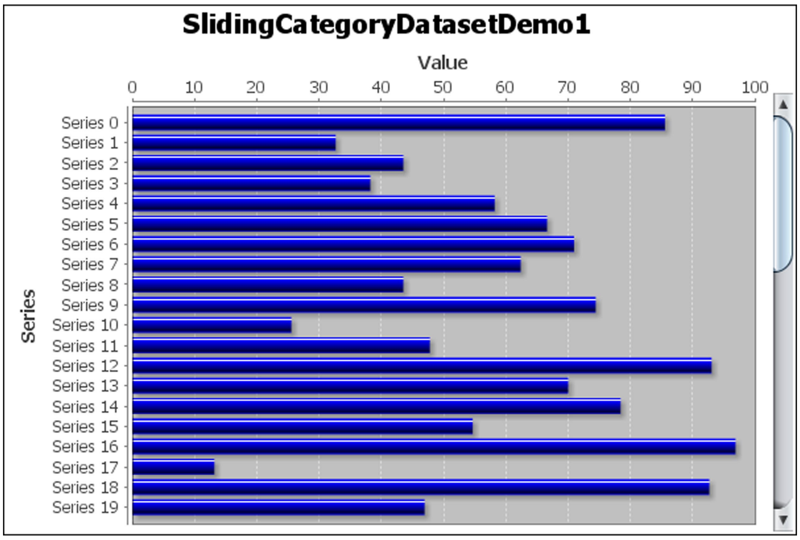
Multivariate linear data have multiple attributes, usually more than four, and because of this fact visualizing such data is one of the major challenges in InfoVis. Different techniques are proposed for visualization of multivariate data and, according to Keim (KEIM, 1997), these techniques can be divided into six classes: geometric, icon-based, pixel-oriented, hierarchical, graph-based and hybrid techniques. Figures 2.7 and 2.8 show two examples of InfoVis techniques for multivariate data.

Data organized as a network is structured according to relationships between the elements, and is represented by a graph. This representation imposed limitations for displaying large datasets. Figures 2.9 and 2.10 show examples of network visualization techniques.

Visualizations for displaying trees are employed to exhibit hierarchical data. However, as a network visualization, hierarchical representation has constraints for the visualization of large datasets. In Figures 2.11 and 2.12, we can see examples of hierarchical visualizations.

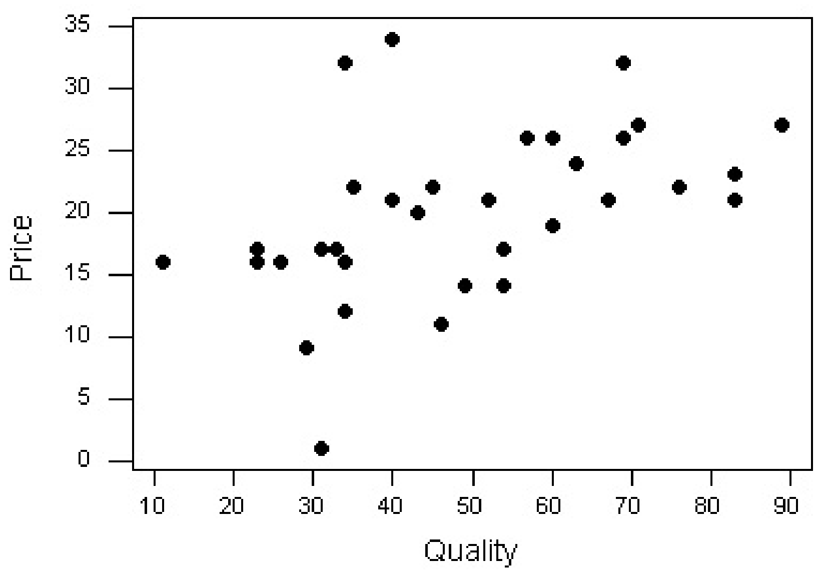
Finally, data carrying temporal and/or spatial/geographical information are used to represent, respectively, data that change in time or data that have a correspondence with geo-referenced information. Figure 2.13 shows an example of such data in a single visualization.

Figure 2.4: Univariate Linear Data in a Bar Chart: the data set is simply a set of Values, each Value associated to an element. Visualization generated with the JFreeChart toolkit




(GILBERT, 2005)

Figure 2.5: Bivariate Linear Data in a Scatterplot: Relation Price x Quality of a Product

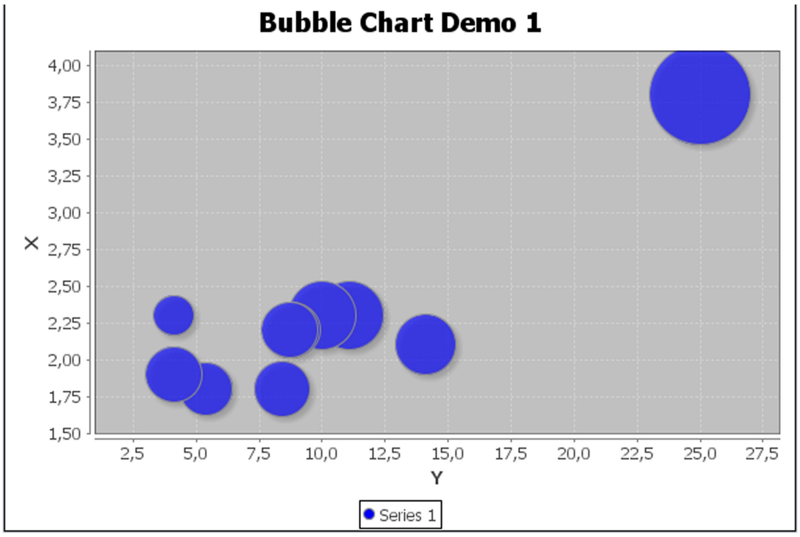



### 2.2.2 Interaction

There are many interactive techniques available to interact with data representations. In this work, we limit ourselves to discuss briefly scrolling/zooming, overview+detail, focus+context and filtering, following the task typology proposed by Shneiderman' (SHNEIDERMAN, 1996) and Katifori (KATIFORI et al., 2007).

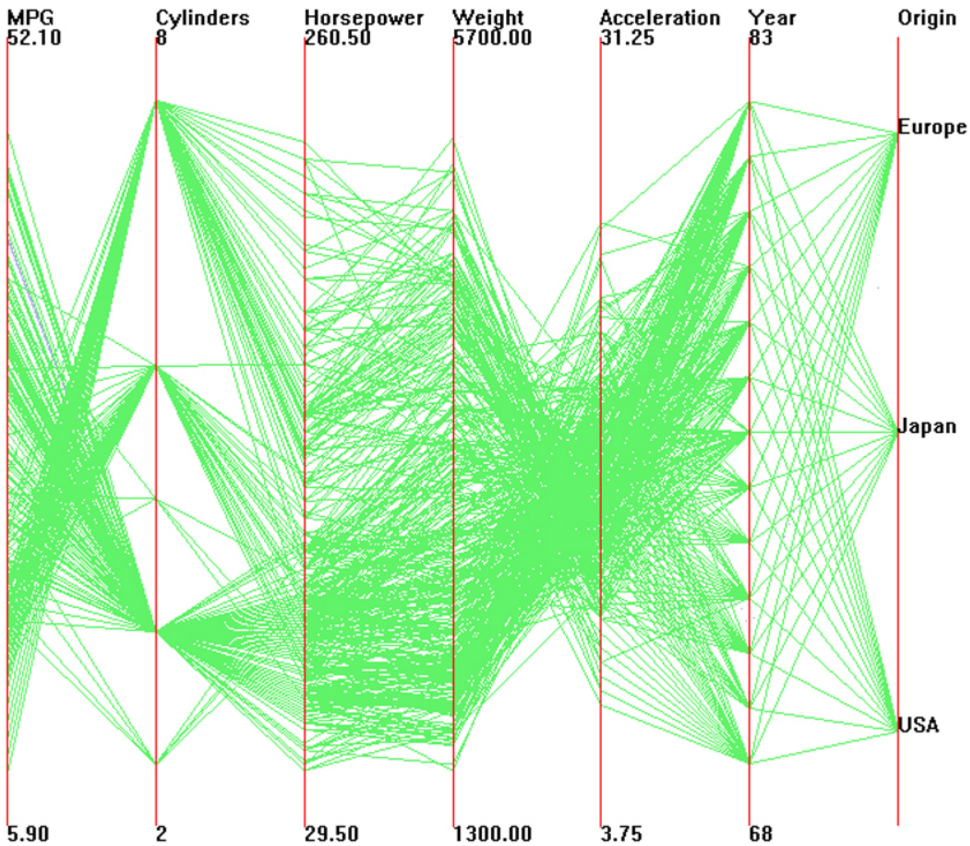
Scrolling and zooming are intuitive interaction techniques although both can hide the

Figure 2.6: Trivariate Linear Data in a Bubble Chart: Y-Values x X-Values x Z-Values (Bubble Size). Visualization generated with the JFreeChart toolkit




(GILBERT, 2005)

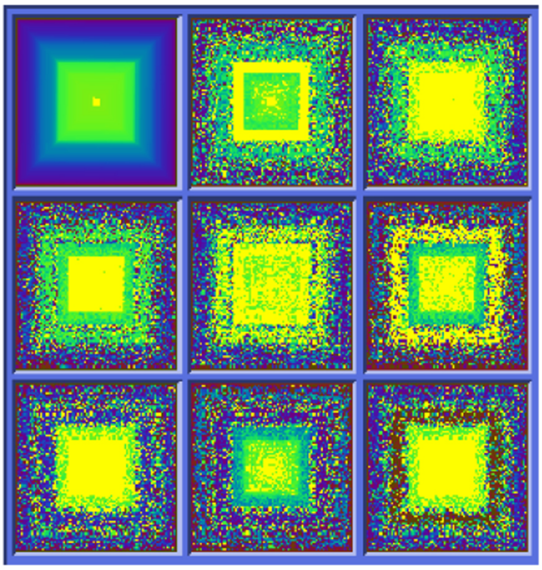
Figure 2.7: Multivariate Linear Data (the classical Cars dataset) exhibited with Parallel Coordinates. Visualization generated with XmdvTool




(WARD, 1994)

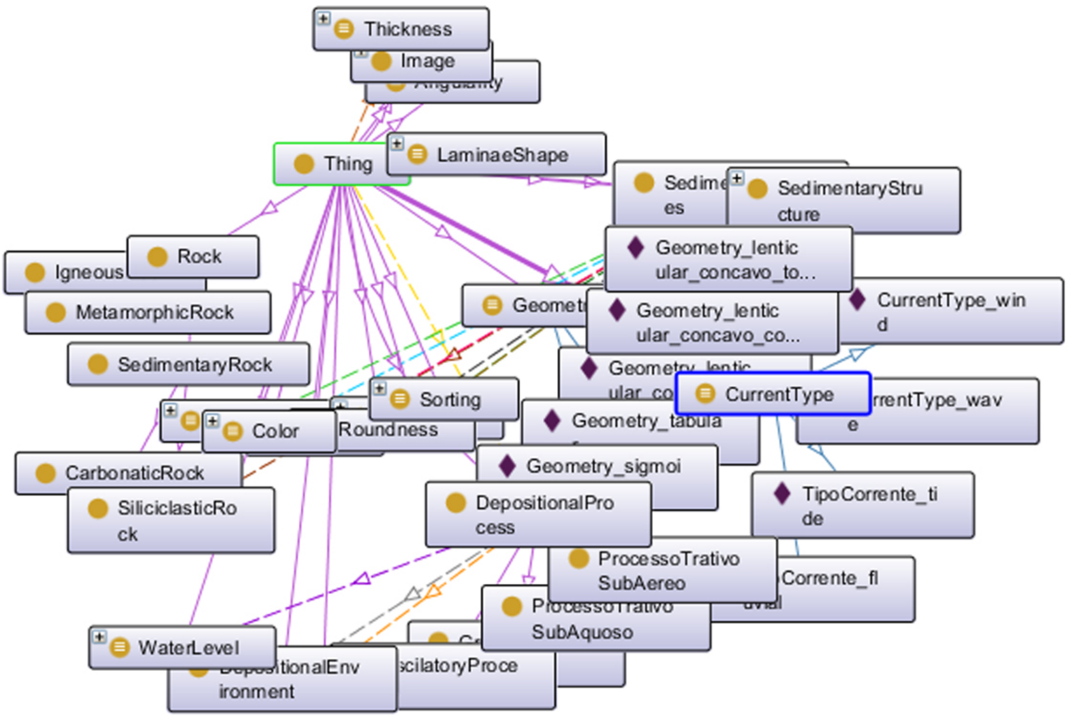
global vision of the entire view. Figure 2.14 brings an example of these techniques.

Figure 2.8: Multivariate Linear Data exhibited with Pixel-Oriented approach. Nine attributes shown in separate windows, as coloured spirals, where pixels in the same relative position pertains to the same tuple and the color represents the value of the attribute.



(KEIM; KRIEGEL, 1996)

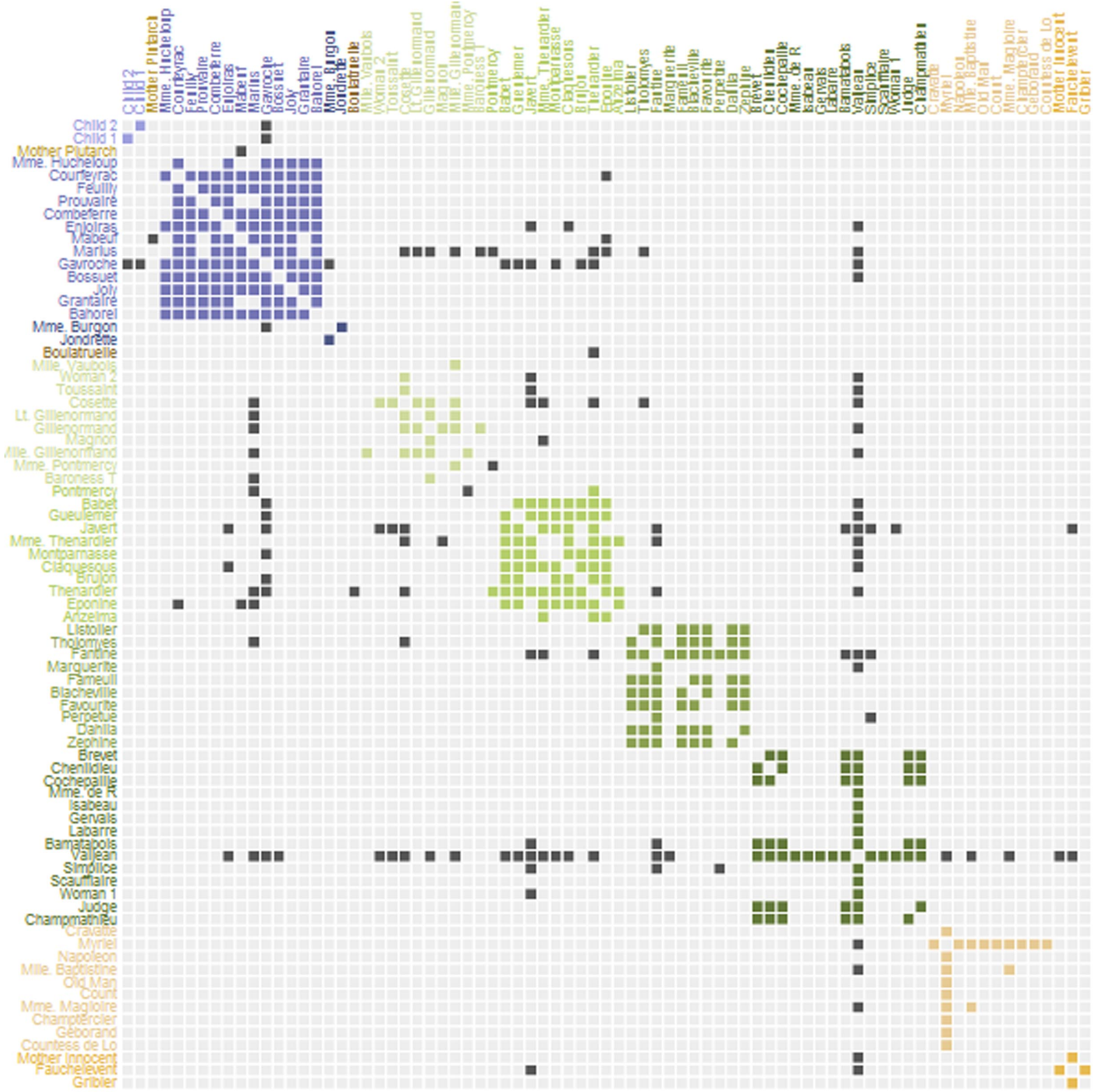
Figure 2.9: Network exhibited as a simple 2D graph. Visualization generated with Ontograf, a plugin for Protègè




(FALCONER, 2010)

An overview+detail technique shows the overview and a detailed part of the dataset in separate views (as can be observed in Figure 2.15) while focus+context techniques

Figure 2.10: Network represented as an adjacency matrix. Visualization generated with Protovis




(BOSTOCK; HEER, 2009)

display the detailed and contextual information in the same view (Figure 2.16) (COCKBURN; KARLSON; BEDERSON, 2009).

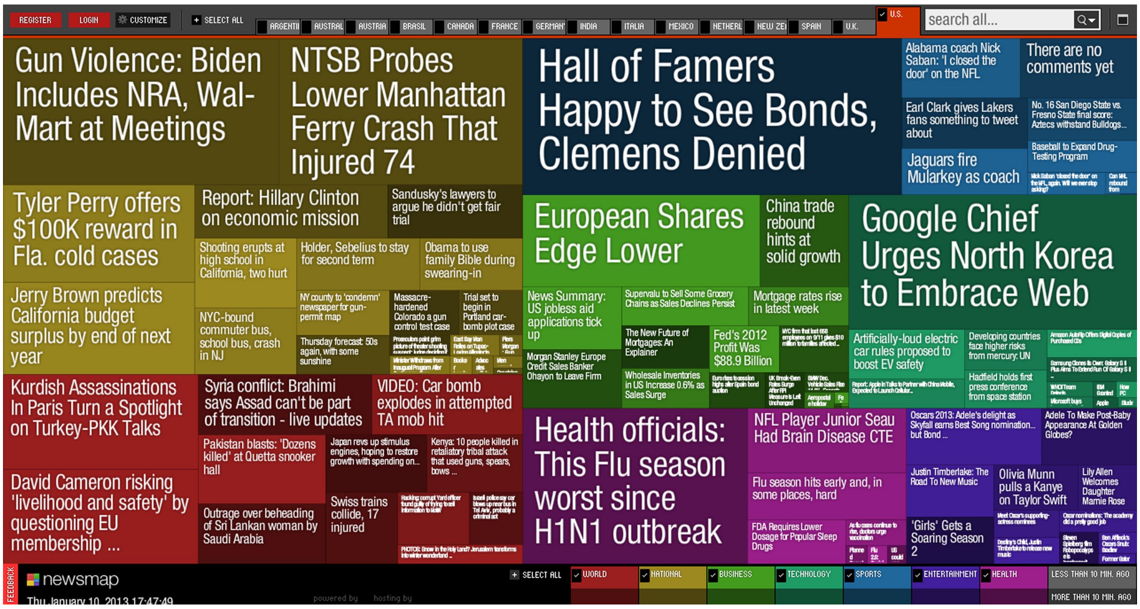
Lastly, Figure 2.17 presents an example of filtering, which allows carrying out analysis focused on parts of the dataset.

### 2.2.3 Evaluation

Although the first visualization techniques were presented without evaluation in the 80's and early 90's, soon it was evident that such techniques needed to be evaluated. Along the years, different authors have been reporting evaluation of information visualization techniques. There has been a specific forum for that in the series of BELIV events (BELIV 2006, 2008, 2010, 2012).

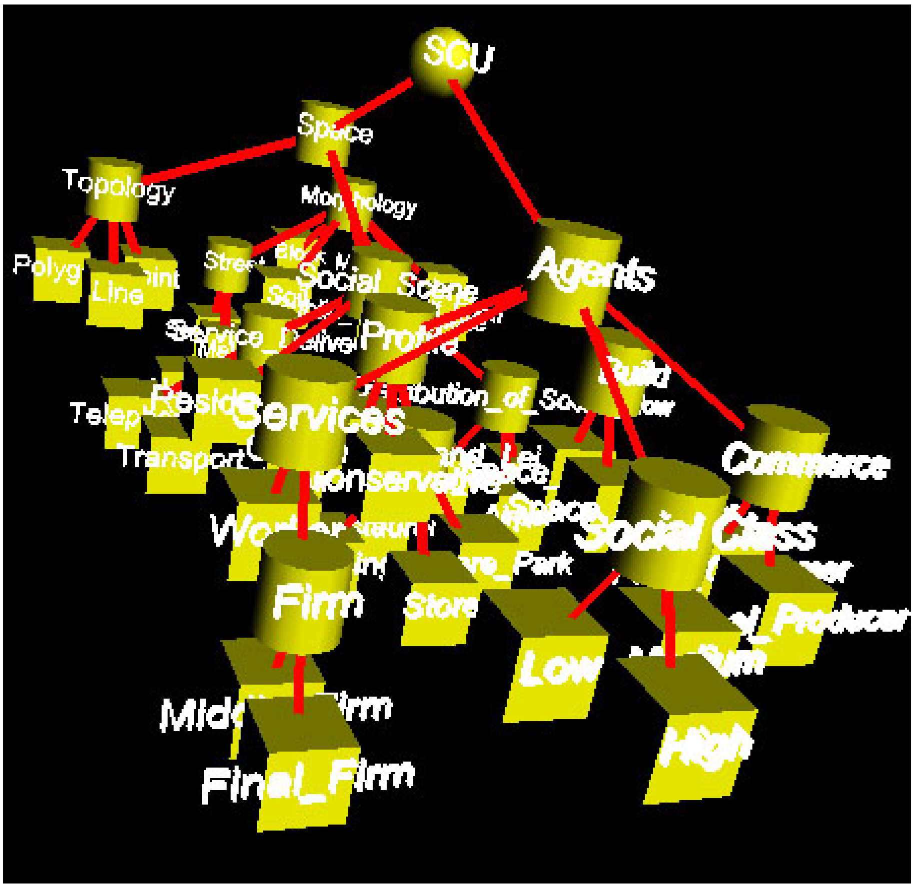
Among some works which try to establish a framework for evaluation of InfoVis techniques, Munzner (MUNZNER, 2006) presents a nested model for visualization design and

Figure 2.11: Hierarchical structure exhibited as a treemap. Visualization generated with Newsmap



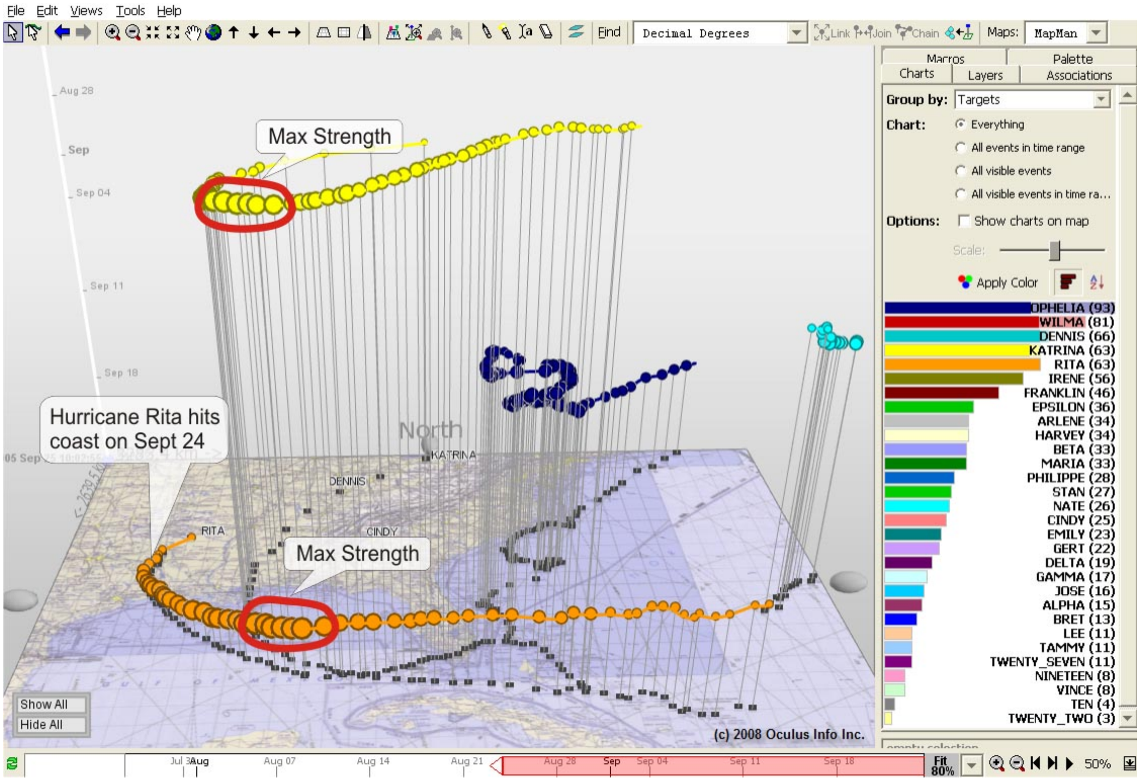

(ONG et al., 2005)

Figure 2.12: Hierarchical structure exhibited as 3D Conetree. Visualization generated with the Java 3D API



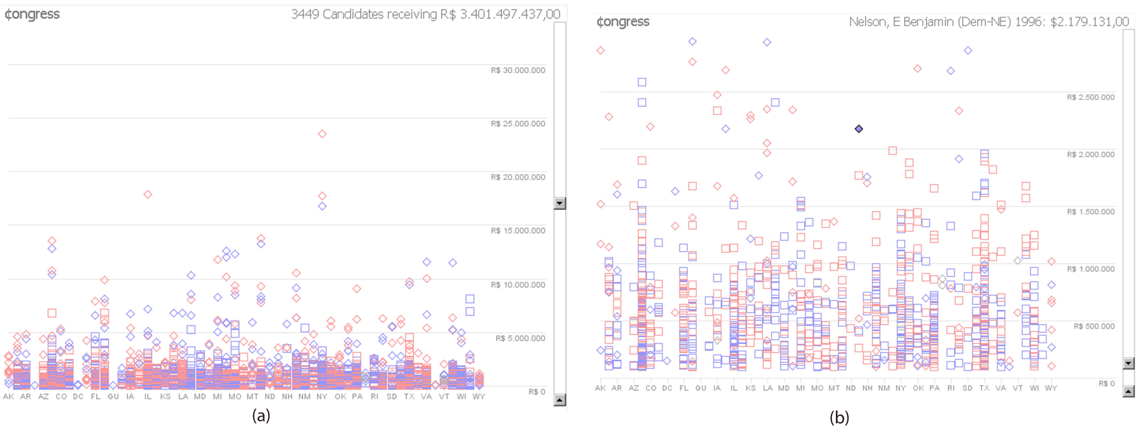

(ORACLE, 2012)

Figure 2.13: Example of temporal, geo-referenced data displayed in an integrated view. Visualization generated with Geotime




(KAPLER; WRIGHT, 2005)

Figure 2.14: Interaction provided by scrolling/zooming in a scatterplot generated with the Prefuse toolkit: (a) and (b) present different zooming levels of the same data visualization with the sliders for vertical scrolling allowing to select the region where to zoom in.




(HEER; CARD; LANDAY, 2005)

validation with four layers:

(1) Domain problem characterization;

(2) Data/Operation abstraction design;

(3) Encoding/interaction technique design;

(4) Algorithm design.

This model provides prescriptive guidance for determining appropriate evaluation ap-

Figure 2.15: Interaction provided by overview+detail used in Street View tool of Google Maps


(GOOGLE, 2007)

proaches by identifying threats to validity unique to each level. Three recommendations are motivated by this model: authors should distinguish between these levels when claiming contributions at more than one of them; authors should explicitly state upstream assumptions at levels above the focus of a paper; and visualization venues should accept more papers on domain characterization.

Another interesting work, valuable for our purposes, is by Isenberg et al. (ISENBERG et al., 2008). The authors suggest grounded evaluation as a process for ensuring that the evaluation of an information visualization tool is situated within the context of its intended use. The work also discusses how qualitative inquiry may be a beneficial approach as part of this process and present case studies in this context. In Figure 2.18, this process starts at (A), using qualitative studies as a form of evaluation that can be carried out before initial design, instead of beginning at (B), which means designing for a specific problem after a thorough investigation of existing literature.

More recently, Forsell and Johansson (FORSELL; JOHANSSON, 2010) proposed general heuristics addressing common and important usability problems in information visualization techniques (see Table 2.1).

A cycle of design and evaluation is the basis of a user-centered perspective (FREITAS; PIMENTA; SCAPIN, 2012), where the authors propose a set of guidelines for usability evaluation of information visualization techniques. The guidelines are:

(1) The context of usage for evaluation must be defined before the beginning of evaluation;

(2) For evaluating information visualization techniques one needs to know who the users are, and to decide which users tasks to support;

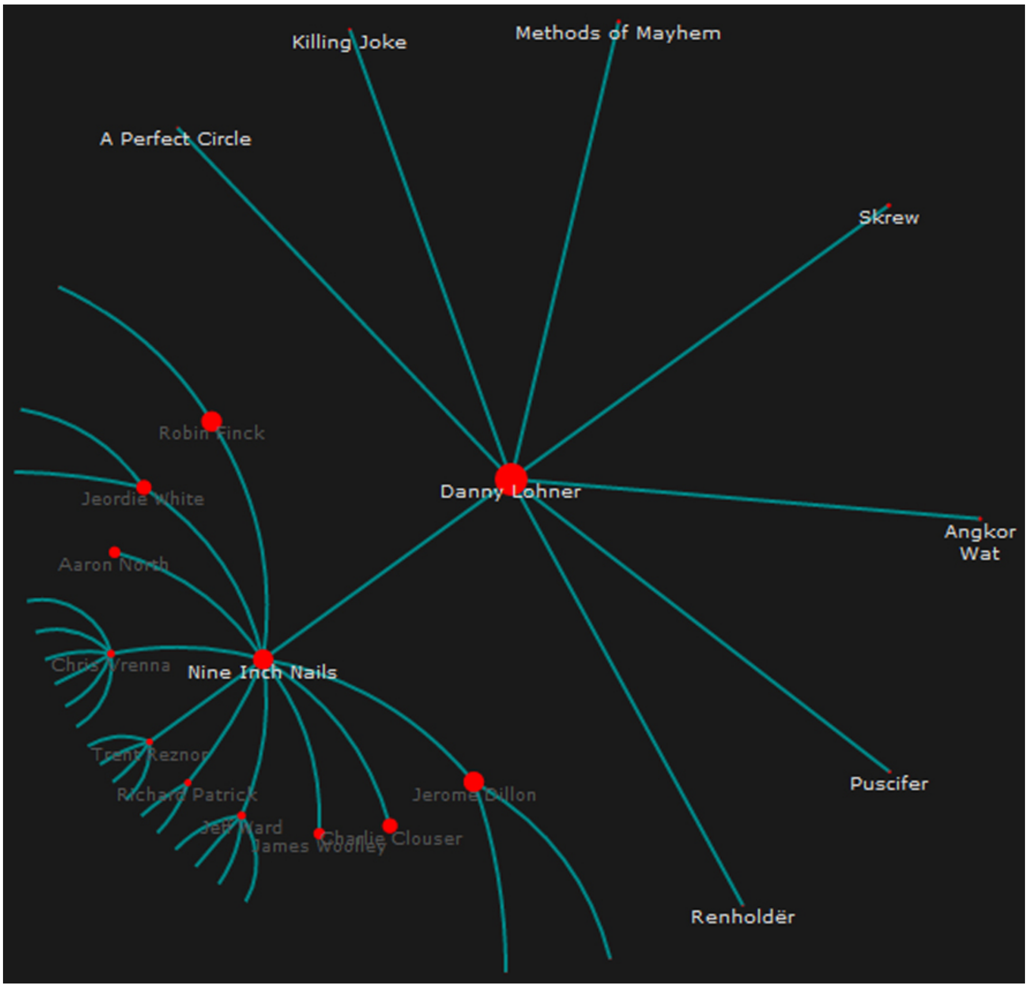
(3) For evaluating information visualization techniques one needs to understand which tasks users have to perform and their characteristics (steps, constraints, and task attributes like frequency, priority, etc.) and to decide which tasks to support, and

(4) Evaluating early.

Table 2.1: Heuristics for evaluating information visualization (FORSELL; JOHANSSON, 2010)

| Heuristics | Description |
|---|---|
| Information coding | Perception of information is directly dependent on the mapping of data elements to visual objects. This should be enhanced by using realistic characteristics/techniques or the use of additional symbols. |
| Minimal actions | Concerns workload with respect to the number of actions necessary to accomplish a goal or a task. |
| Flexibility | Flexibility is reflected in the number of possible ways of achieving a given goal. It refers to the means available to customization in order to take into account working strategies, habits and task requirements. |
| Orientation and help | Functions like support to control levels of details, redo/undo of actions and representing additional information. |
| Spatial organization | Concerns users orientation in the information space, the distribution of elements in the layout, precision and legibility, efficiency in space usage and distortion of visual elements. |
| Consistency | Refers to the way design choices are maintained in similar contexts, and are different when applied to different contexts. |
| Recognition rather than recall | The user should not have to memorize a lot of information to carry out tasks. |
| Prompting | Refers to all means that help to know all alternatives when several actions are possible depending on the contexts. |
| Remove the extraneous | Concerns whether any extra information can be a distraction and take the eye away from seeing the data or making comparisons. |
| Data set reduction | Concerns provided features for reducing a data set, their efficiency and ease of use. |

Figure 2.16: Interaction provided by focus+context in a 2D hyperbolic tree visualization. Image generated with the InfoVis toolkit

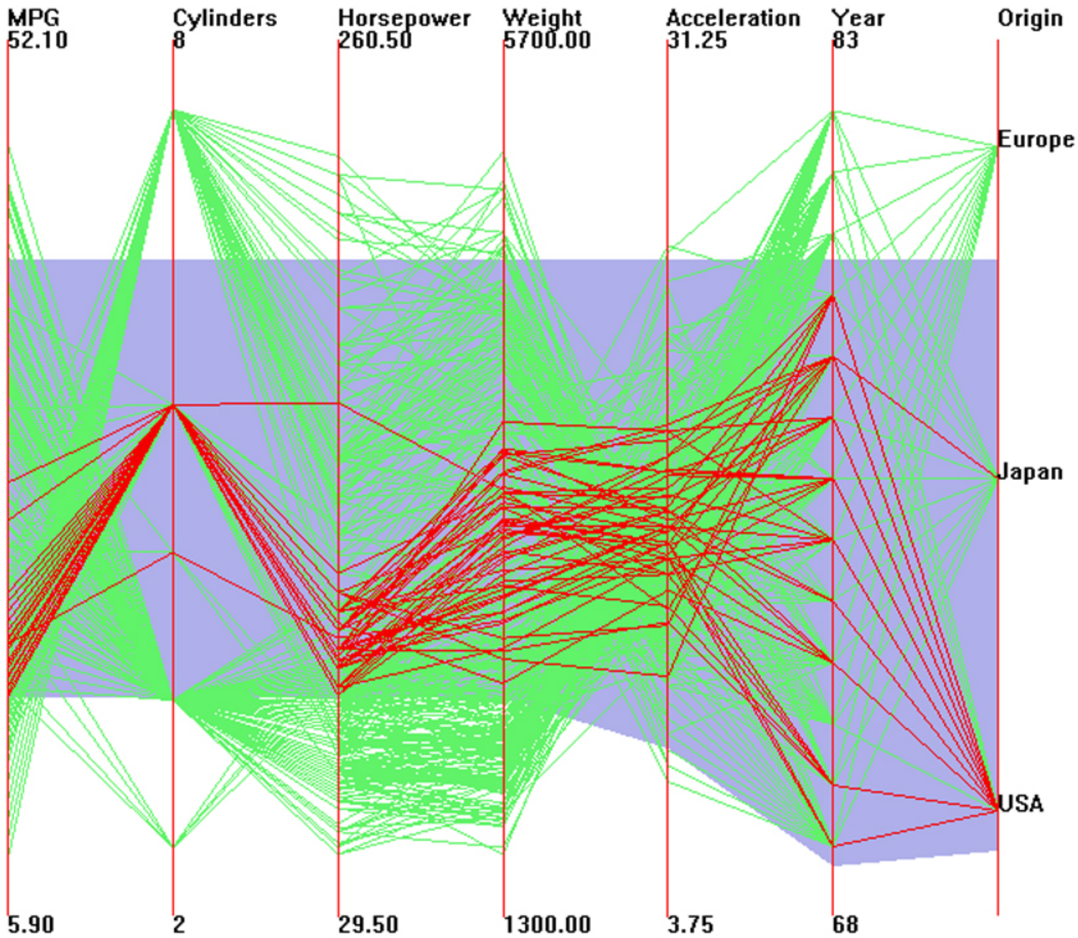



(FEKETE, 2004)

### 2.2.4 Final Comments

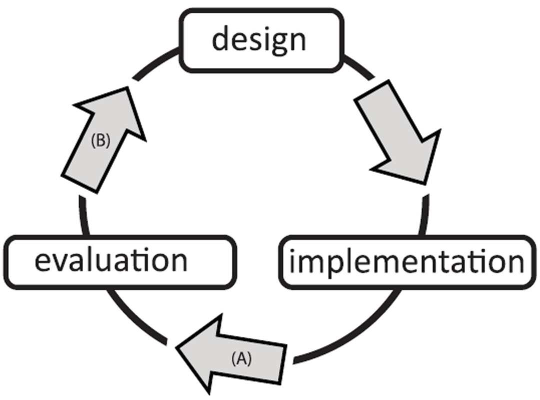
Visualization of ontologies is based on graphical representations, and interaction techniques play an important role because they support users' tasks. In this chapter we reviewed the main concepts regarding ontologies and introduced the visualization and interaction techniques we will adopt for providing interactive visualization of ontologies both at intensional and extensional levels.

Figure 2.17: Filtering in parallel coordinates allows to the decrease the amount of data elements presented in the display. Image generated with XmdvTool




(WARD, 1994)

Figure 2.18: Grounded evaluation of information visualization




(ISENBERG et al., 2008)

# 3 RELATED WORK

*This chapter contains an overview of some works related to the main concepts addressed in the present study. In general, these works do not report user studies or other evaluation techniques to assess the way the intensional and extensional levels of ontologies are displayed.*

## 3.1 Ontology Visualization

According to Katifori et al. (KATIFORI et al., 2007), it is not simple to create a visualization that displays effectively all the information related to an ontology, and, at the same time, allows the user to perform easily various operations on it. Although there are various tools for building ontologies, few of them are devoted to the display of both the intensional and extensional levels of ontologies. In the next paragraphs we briefly review these tools.

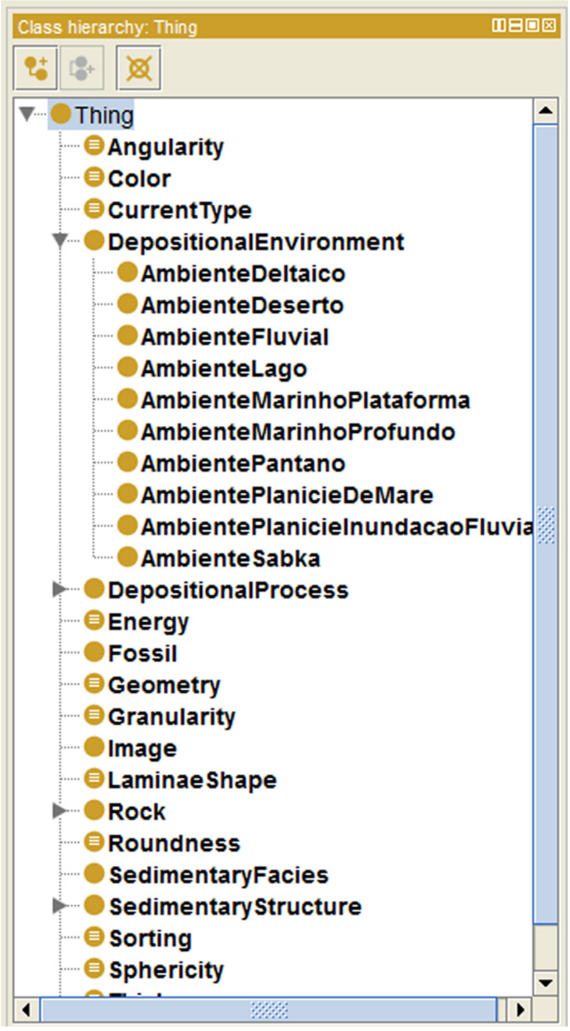
Mostly, researchers use Protègè (NOY; FERGERSON; MUSEN, 2000) for the creation and visualization of ontologies. Although Protègè's main visualization for the ontology hierarchy is a tree view (Class Browser - see Figure 3.1), other visualization techniques have been proposed as OWLViz (Figure 3.2), a plugin for Protègè based on Graphviz (AT&T, 2012), and Ontograf (FALCONER, 2010) (see Figure 3.3). Ontograf presents seven visualization possibilities: alphabetical grid, radial and spring graphs, and four implementations of tree visualization: vertical, horizontal, directed vertical and directed horizontal. Instances are exhibited as leaves of nodes. For user interaction, OntoGraf possibilities search, zoom, pan, expand/retract nodes and filter for the relationships exhibition.

Katifori (KATIFORI et al., 2008) presents a comparative study of four visualization techniques available in past versions of Protègè: Class Browser, Jambalaya (Figure 3.4) (discontinued), TGVizTab (Figure 3.5) (discontinued) and OntoViz (Figure 3.6) (discontinued). The information retrieval provided by these tools were also evaluated and Class Browser showed the best rating because, according to the study, it offers a clear view of the hierarchy without label overlapping combined with the possibility of quick and systematic browsing as well as a "static" node positioning that favours node re-finding.

Baehrecke et al. (BAEHRECKE et al., 2004) proposed the use of treemaps, where color, size, and grouping are used in the visual representation of an ontology (Figure 3.7).

Other proposals, close to our work, combine different information visualization techniques, as in the work based on Concept Modeller by Schevers et al. (SCHEVERS; TRINIDAD; DROGEMULLER, 2006), where the user interacts with the ontology showed with Protègè. Classes representing spatial information (like polygons, points, etc.) are

Figure 3.1: Protègè Class Browser Visualization




(NOY; FERGERSON; MUSEN, 2000)

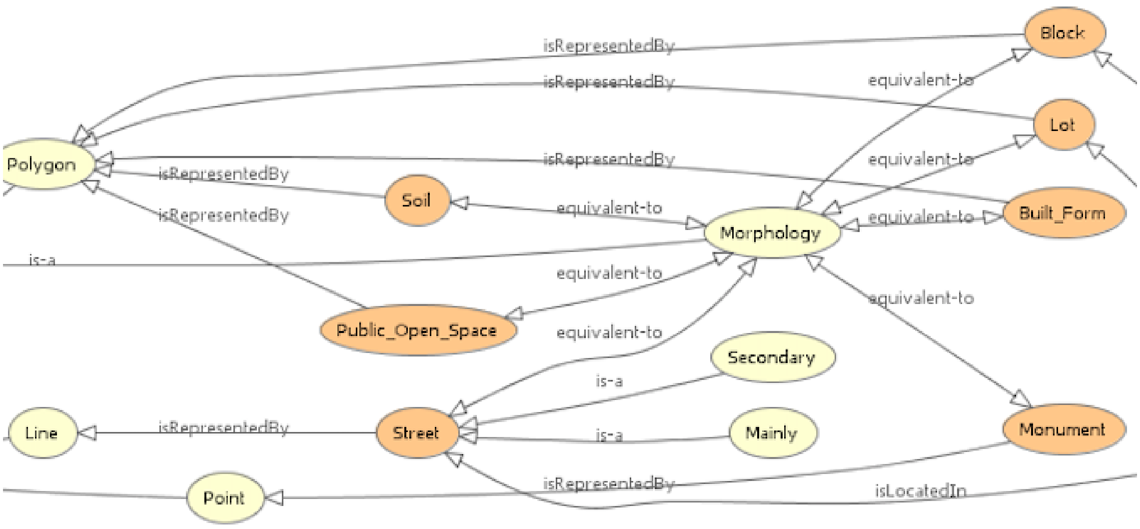
presented in a second graphical interface that is used to mimic the functionality of a GIS (Geographic Information System) (Figure 3.8). Using polygon information from GIS, zones can be defined and contains functions that have several properties that can be changed by the user and each change will have an effect on the zone and after that on the precinct, which is comprised of the zones. The value of the properties of the precinct is displayed in charts.

Catenazzi et al. (CATENAZZI; SOMMARUGA; MAZZA, 2009) present a study about tools for ontologies visualization and propose the OWLeasyViz tool (Figure 3.9). OWLeasyViz combines textual and graphical representations for displaying only the intensional level of ontologies (class hierarchies, relationships and data properties or attributes). Interaction techniques such as zooming, filtering and search are available.

Beyond Falconer (FALCONER, 2010), alternatives for the visualization of both the intensional and extensional levels of ontologies were proposed by other authors (PIZZINATO; RIGO; VIEIRA, 2010), (CATARCI et al., 2010), (KRIGLSTEIN; WALLNER, 2011) and (BACH; PIETRIGA; LICCARDI, 2011).

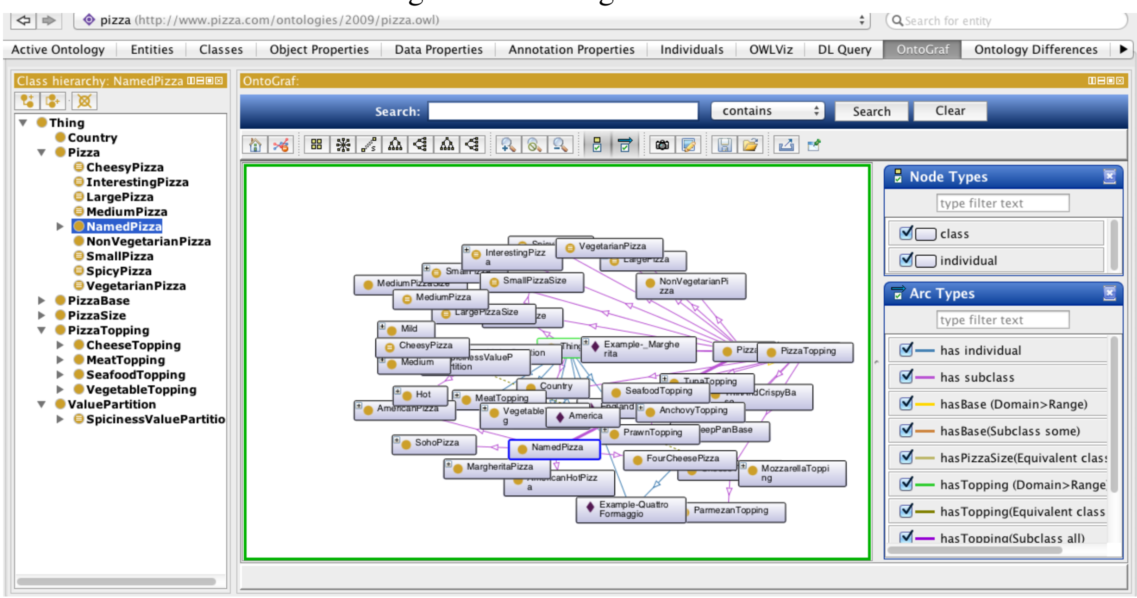
Pizzinato et al. (PIZZINATO; RIGO; VIEIRA, 2010) propose a tool for ontology visualization that shows the classes hierarchy, relationships and attributes as nodes in a

Figure 3.2: Protègè OWLViz




(NOY; FERGERSON; MUSEN, 2000)

Figure 3.3: Protègè OntoGraf



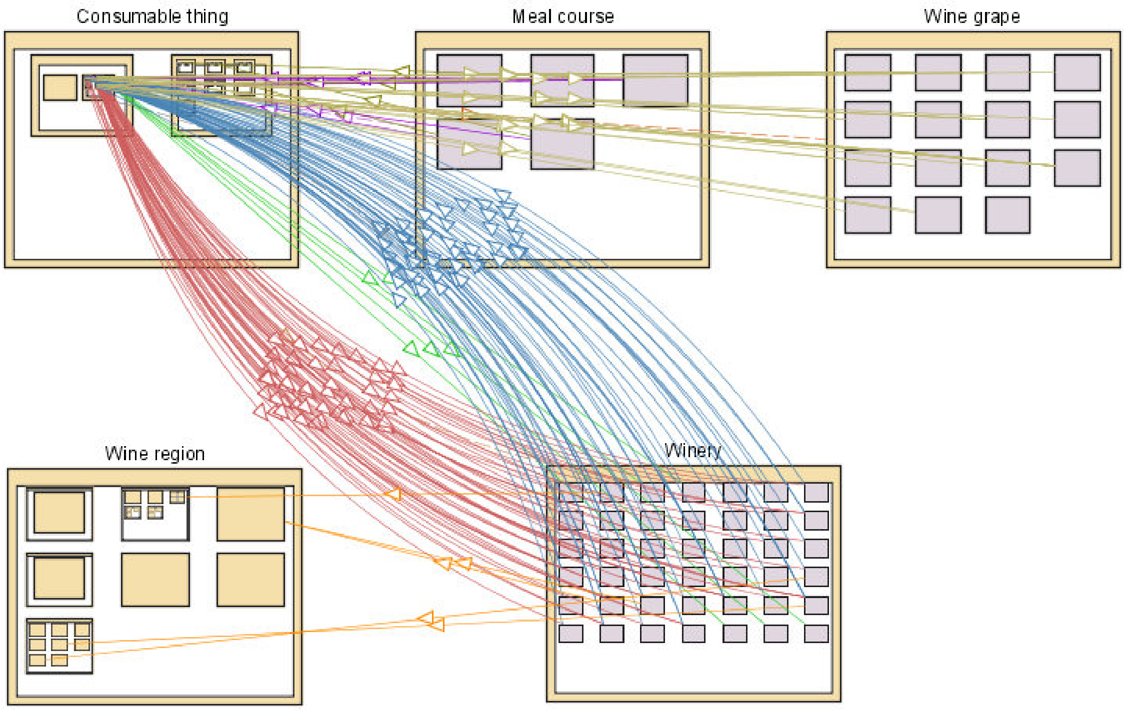

(FALCONER, 2010)

hyperbolic tree (Figure 3.10). Instances are displayed as menu popup related to the node class. The autors inform that multiple inheritance are too displayed but these details are not provided in the work.

On-TIME (CATARCI et al., 2010) is a task-centered information management system, where the intensional level is shown together with the extensional level, as a collection of nodes and edges (Figure 3.11). If a class has many instances, edges eventually overlap and this increases the complexity of the analysis process by the user.

Kriglstein and Wallner (KRIGLSTEIN; WALLNER, 2011) presented Knoocks (Figure 3.12), a visualization tool focused on the interconnections within an ontology. It

Figure 3.4: Protègè Jambalaya, a plugin created for Protègè that uses a zoomable method called Shrimp (Simple Hierarchical Multi-Perspective) to visualize the knowledge bases the user has created. Subclasses and instances are drawn as nested rectangles inside their superclass node, and they are distinguished from each other using different colors. Relationships between classes and instances in the knowledge base are represented in the graph using directed arcs.



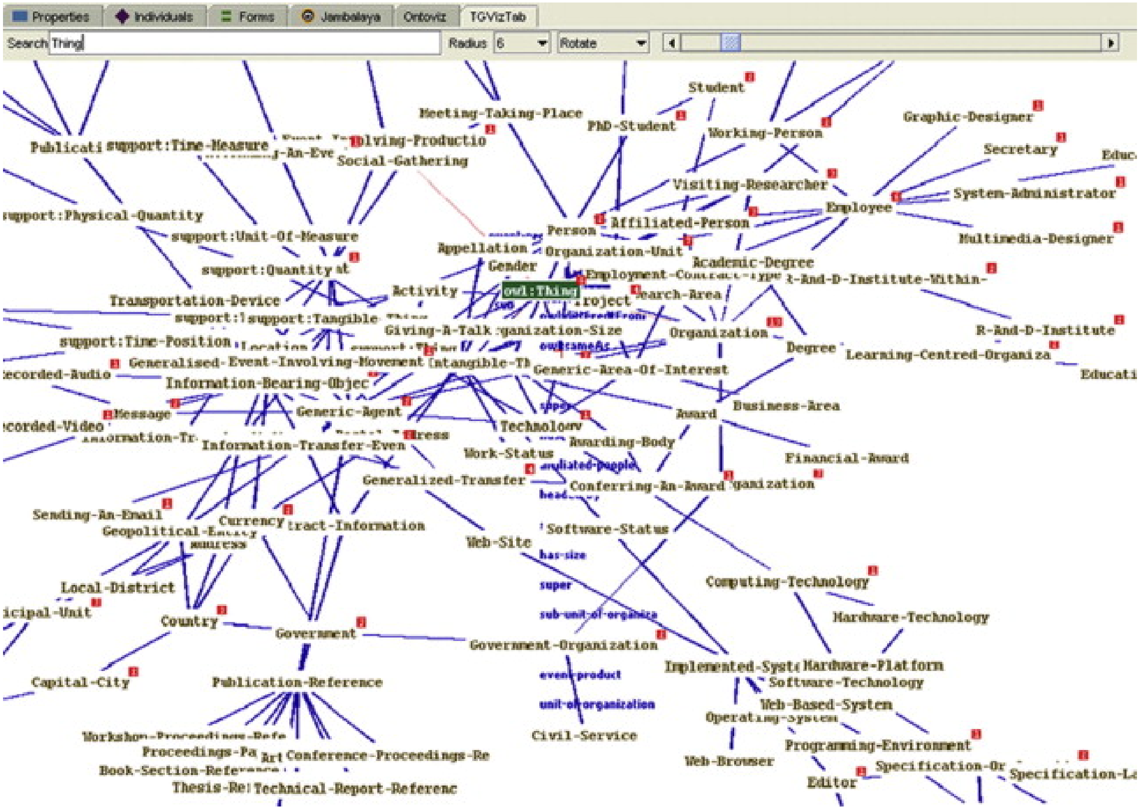

(STOREY, 2001)

also shows the intensional and extensional levels: classes (intensional level) are displayed as blocks and, inside of them, the instances (extensional level) are exhibited as listings. A bar button with arrow is added between each parent class and its children to control folding of classes. So, a similar problem to the one in Catarci et al.'s work (CATARCI et al., 2010) might arise: if a class has several instances, visualization can become hard to understand because the user will need to navigate through an exhaustive sequence of elements.

Bach et al. (BACH; PIETRIGA; LICCARDI, 2011) proposed OntoTrix (Figure 3.13), a visualization technique designed to enable users to visualize large OWL ontology instance sets and the relations that connect them. The technique uses both node-link and adjacency matrix representations of graphs to display ontology intensional and extensional levels. Details about the exhibition of the different views and the switching between them are not presented.

Hop et al. (HOP et al., 2012) presented Glow, a tool for ontology visualization based on hierarchical edge bundles (Figure 3.14) for displaying hierarchical relations, such as concepts' structures formed by "subclass-of" and "type-of" relationships. The hierarchical edge bundle approach represents edges as curves, whose paths are defined by control points generated from the hierarchy: the control points correspond to the hierarchical nodes found along the shortest path from the edge source node to the edge target node.

Figure 3.5: Protègè TGVizTab (TouchGraph Visualization Tab) is a plugin that displays classes as well as instances. These are represented as nodes in the graph, and can be displayed in different colors to be easily distinguishable. Ontology relations (slots) are represented as graph edges (links between the nodes).




(ALANI, 2003)

## 3.2 Automatic Analysis of Ontologies

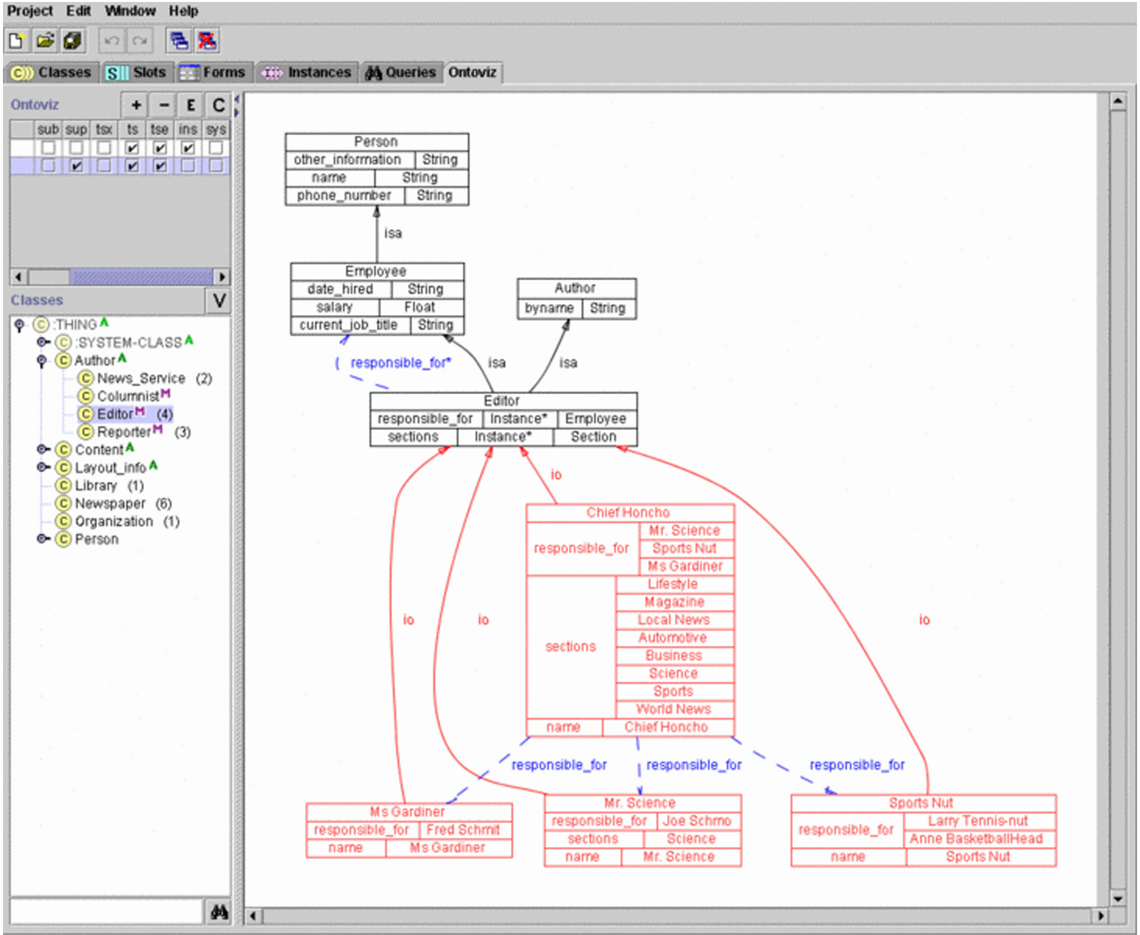
In the works referenced above, there has been little or no concern regarding automating the extraction and display of concepts and properties of ontologies. However, if we use information visualization techniques combined with visual analytics we can amplify cognition and reduce exploration time of a data set, allowing the recognition of patterns and facilitating inferences about different concepts.

Regarding this, Card and Nation (CARD; NATION, 2002) and Spence (SPENCE, 2007) describe the application of the *Degree of Interest* (DoI) concept for tree layouts as a logical filtering of nodes. Husken and Ziegler (HUSKEN; ZIEGLER, 2007) discuss the use of DoI in visualization and exploration of ontologies, where nodes are automatically displayed according to the user's computed DoI.

D'Entremont and Storey (D'ENTREMONT; STOREY, 2009) also apply DoI in their work and present a plug-in for Protègè, called Diamond (Figure 3.15). This tool consists of two components: a mechanism to continuously calculate a user's DoI and a dynamic display of the information that uses the DoI calculation to draw users' attention to interesting elements in order to reduce navigation overhead. The results obtained from DoI are displayed on views existing in Protègè (Class Browser and Jambalaya).

Chan et al. (CHAN; KEETON; MA, 2010) present an interactive visual technique for

Figure 3.6: Protègè OntoViz, a plugin based on the node-link approach, where the ontology is presented as a 2D graph. It has the capability of presenting each class, its name, properties and inheritance and also relations. Its instances are displayed in different colors.




(SINTEK, 2007)

analyzing and understanding hierarchical data, which they have applied for analyzing a corpus of technical reports. The analysis consists of selecting a known entity and then incrementally add other entities to the ontology graph based on known relations (Figure 3.16).

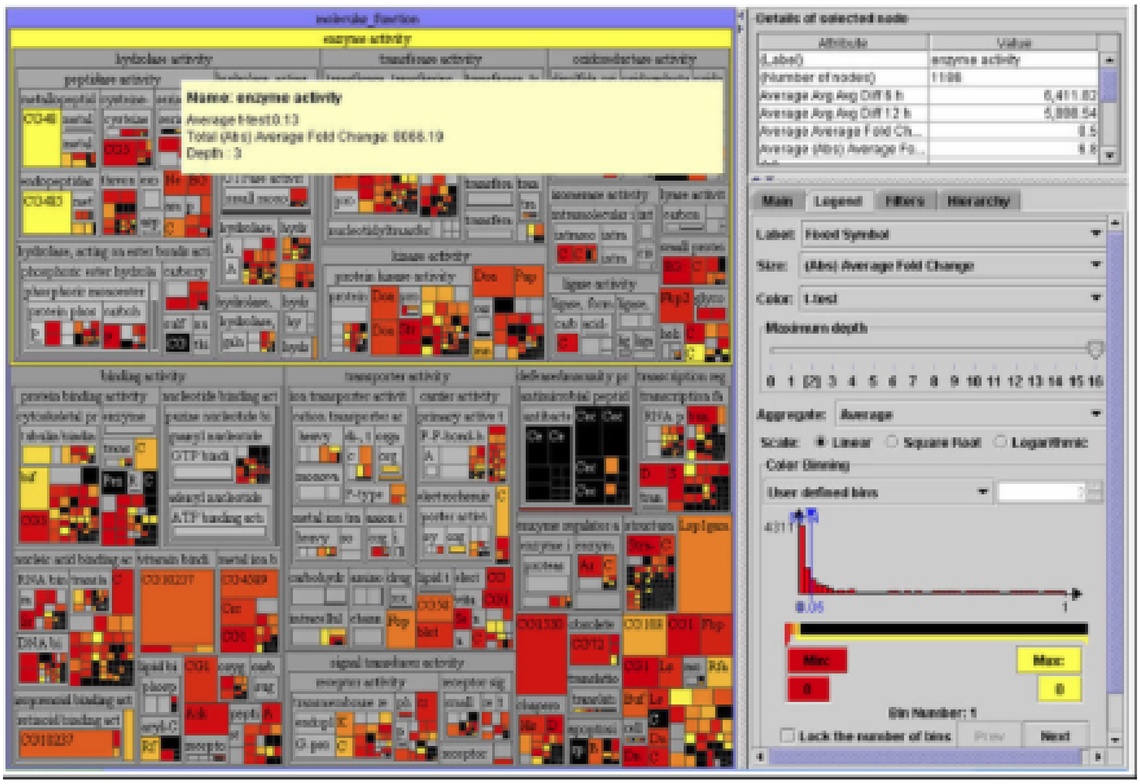
Abello et al. (ABELLO et al., 2013) employs a DoI approach similar to ours but with focus in large dynamic networks (Figure 3.17). By using it to successively refine and investigate the captured details, their technique supports the analysis of dynamic networks from an initial view until pinpointing a user's analysis goal. The approach stands and falls with the ability to specify meaningful DoI functions that exactly reflect a given dynamic pattern a user is interested in.

## 3.3 Discussion and Remarks

In order to identify common requirements for ontology visualizations, we analysed the main features of the visualization solutions for ontologies presented in this chapter.

Class Browser, the Protègè native view for presenting ontology entities, is an common indented list (or treeview) technique employed to present hierarchical structures popular-

Figure 3.7: Visualization employing treemaps for visualization and analysis of microarray and gene ontology. Attributes of genes are represented by size and color-coding.




(BAEHRECKE et al., 2004)

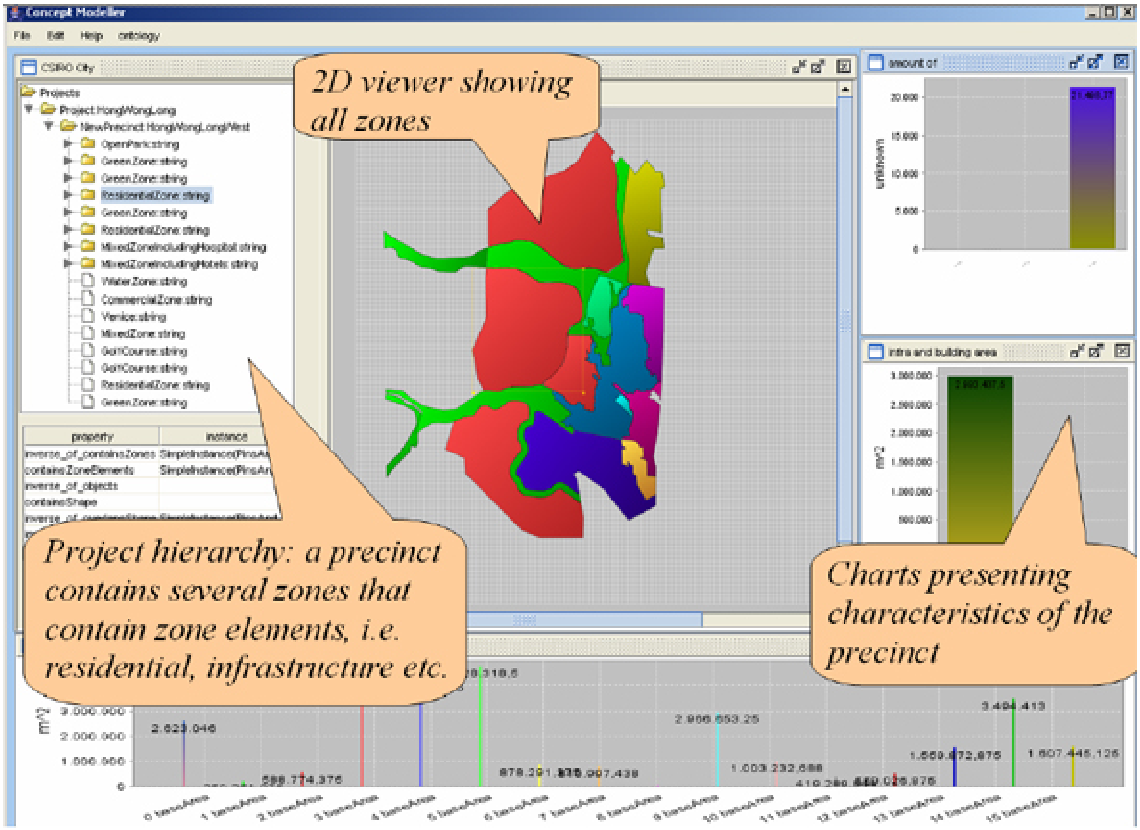
ized by file systems. This technique is also employed in Concept Modeller (SCHEVERS; TRINIDAD; DROGEMULLER, 2006) as well as in On-Time, and complement the visualization plugins proposed for Protègè as OWLViz, Ontograf and Diamond. Although it is an intuitive method for displaying hierarchies, it is a not robust visualization for ontologies because does not allow the exhibition of several entities at the same time with the classes and "is-a" relationships, relationships lists, attributes lists or instances lists.

Most of the tools analysed herein are based on 2D graphs, some of them providing other visualization techniques for the exhibition of different ontologies aspects. Graphs are a very efficient form for representing ontologies, but have the disadvantage of not being adequately scalable. Usually, when the number of nodes and edges increase, the graph becomes too complex. This problem can be observed in OWLViz that shows an ontology as a static 2D graph with a large number of crossing edges and becomes impossible for the user to perceive the graph's general structure. The same problem can be observed in Ontograf, Hyperbolic Tree, On-Time, Knoocks, OntoTrix, Glow, Diamond, Bipartite-Graph-Like and DoI and Large Dynamic Networks, although these are based on non-static graphs.

Knoocks and OntoTrix use hybrid graphs. Knoocks extends the discontinued Jambalaya visualization (Figure 3.4), where the graph nodes are represented by blocks and, inside these classes blocks, we have a list of instances. If a class has several instances, visualization can become hard to understand because the user will need to browse a long sequence of elements.

In OntoTrix, the set of nodes of a given class is represented by an adjacency matrix, and links between nodes are represented as links between matrices elements. The ad-

Figure 3.8: Visualization of urban aspects. For the implementation of an integrated assessment system for urban development, Protègè is used to model an urban ontology and, when a class is selected, a graphical user interface is created automatically using polygon information from GIS.




(SCHEVERS; TRINIDAD; DROGEMULLER, 2006)

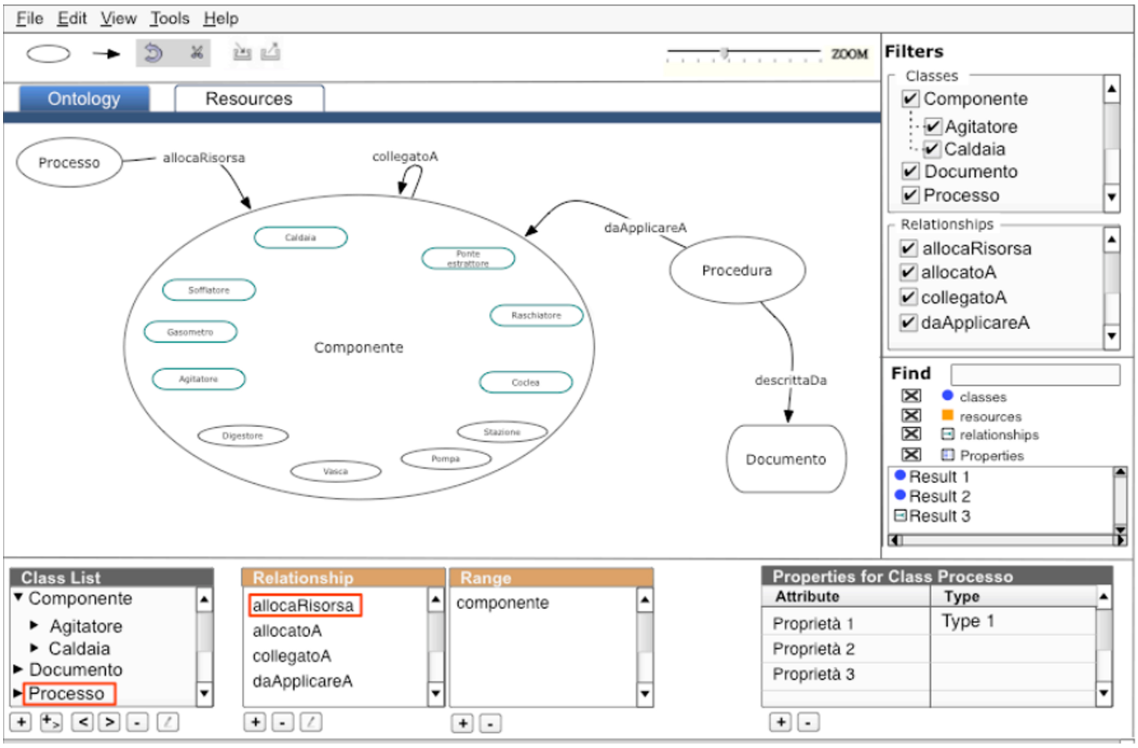
jacency matrix is a technique more compact and free of visual clutter for dense graphs. However, it requires huge efforts for adding/removing a vertex and does not work fine for a fully dynamic structure because it is quite slow for large graphs.

Treemap is an InfoVis technique rarely employed for ontology visualization. According to Mazza (MAZZA, 2009), a treemap is very efficient for representing hierarchical data, where the representation of the nodes through the dimension and color of blocks helps the user to immediately single out and compare nodes, clusterize by patterns, and identify exceptions. But, for ontologies, the main problem is the impossibility of representation of several relationships besides the hierarchical one.

Information at the extensional level of ontologies is not a recurring concern in ontologies visualization proposals. Ontograf, On-Time, Glow and Bipartite-Graph-Like show instances of classes as leaf nodes, while Knoocks and Class Browser show instances as a lists and Hyperbolic Tree employ popup menus over a node. These representations are not good when the ontology has a lot of instances, because they might cause cognitive overload during the analysis process, and do not allow the exhibition of the values of instances' attributes.

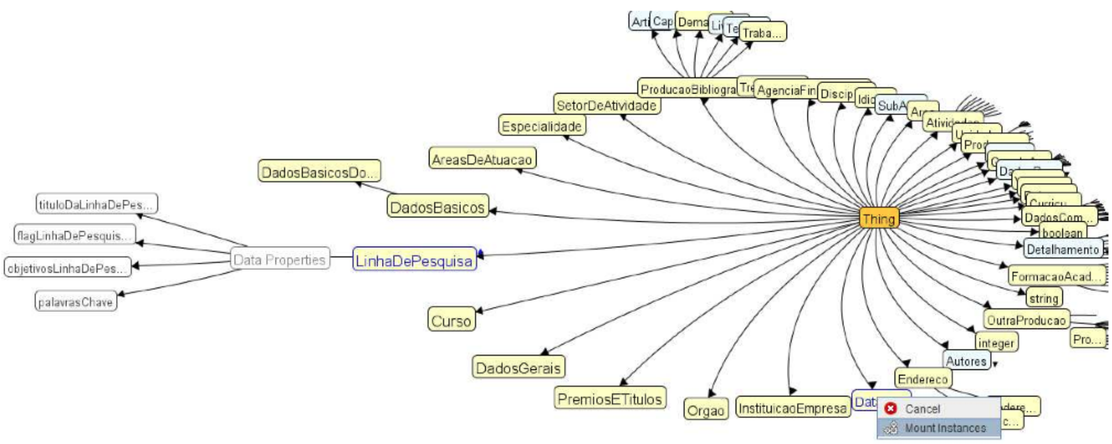
In relation to the use of multiple views, coordinated or not, Concept Modeller and Knoocks provide more than two related views. Other tools (Class Browser, OWLViz, Ontograf, TreeMap, On-Time, OWLeasyViz, OntoTrix, Diamond and DoI and Large Dy-

Figure 3.9: OWLeasyViz tool combining textual and graphical representations for displaying the intensional level of ontologies.




(CATENAZZI; SOMMARUGA; MAZZA, 2009)

Figure 3.10: Ontology visualization where the intensional level is displayed as nodes of a hyperbolic tree and the extensional level in popup menu form.




(PIZZINATO; RIGO; VIEIRA, 2010)

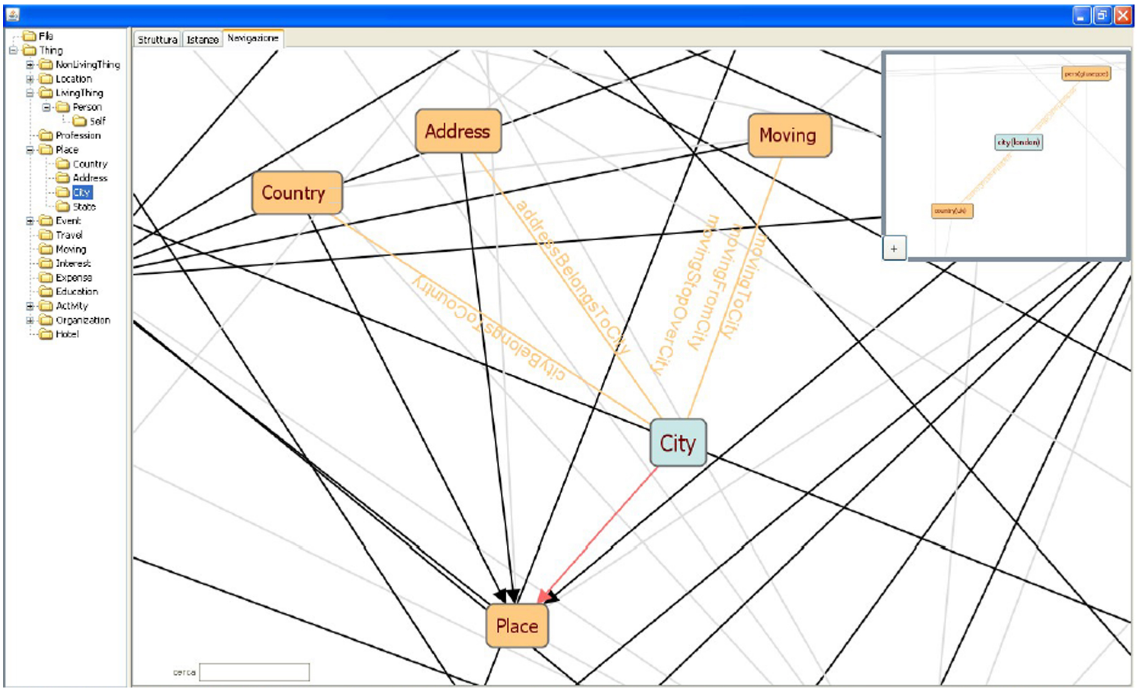
namic Networks) only present some complementary windows with information about the ontology being displayed in the main view. Some of those windows provide interaction in order to allow the user to perform some task on the visualization.

Table 3.1 summarizes the comparison between the main features of the related works presented in this chapter.

Table 3.1: Comparison between features found in the studied techniques and tools.

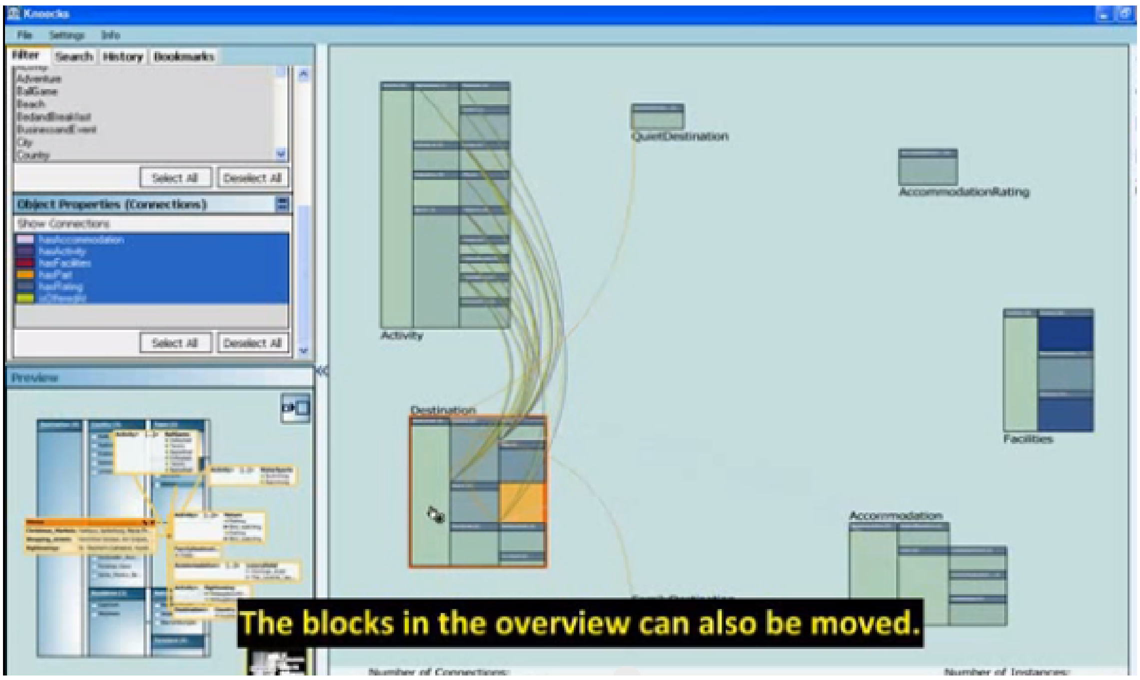
| Visualization Technique | Display of Intensional Level | Display of Extensional Level | Multiple and Synchronized Views | User Task Support | Automatic Analysis |
|---|---|---|---|---|---|
| Protègè Class Browser | Yes | No | No | Yes | No |
| OWLViz (Protègè plugin) | Yes | No | No | Yes | No |
| Ontograf (Protègè plugin) | Yes | Yes | Yes, integrated with Protègè Class Browser | Yes | No |
| TreeMap | Yes | No | No | Yes | Yes |
| Concept Modeller | Yes | Yes | Yes, integrated with Protègè Class Browser | Yes | Yes |
| Hyperbolic Tree | Yes | No | No | Yes | No |
| On-Time | Yes | Yes | Yes | Yes | No |
| OWLeasyViz | Yes | No | No | Yes | No |
| Knoocks | Yes | Yes | Yes | Yes | No |
| OntoTrix | Yes | Yes | No | Yes | No |
| Glow | Yes | Yes | No | Yes | No |
| Diamond (Protègè plugin) | Yes | No | Yes, integrated with Protègè Class Browser | Yes | Yes |
| Bipartite-Graph-Like | Yes | No | No | Yes | No |
| DoI and Large Dynamic Networks | Yes | No | Yes | Yes | Yes |

Figure 3.11: On-TIME system is an overview + detail technique. Classes and instances are displayed as graph nodes and relationships as edges between these. Instances of a selected class are displayed in a second view (class: city; instance: London).



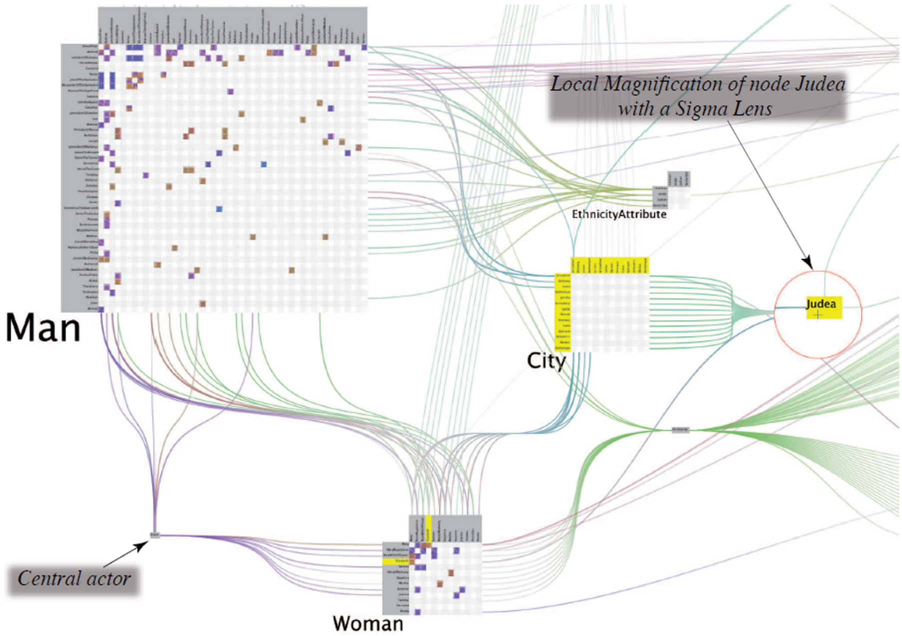

(CATARCI et al., 2010)

Figure 3.12: Knoocks tool. The basic elements are blocks, which are represented by rectangles. Every block stands for a class with its subclasses. The right rectangles are subclasses of the left rectangle, and the instances of a class are listed within their class. Object properties are represented as curved lines and every property type has its own color.



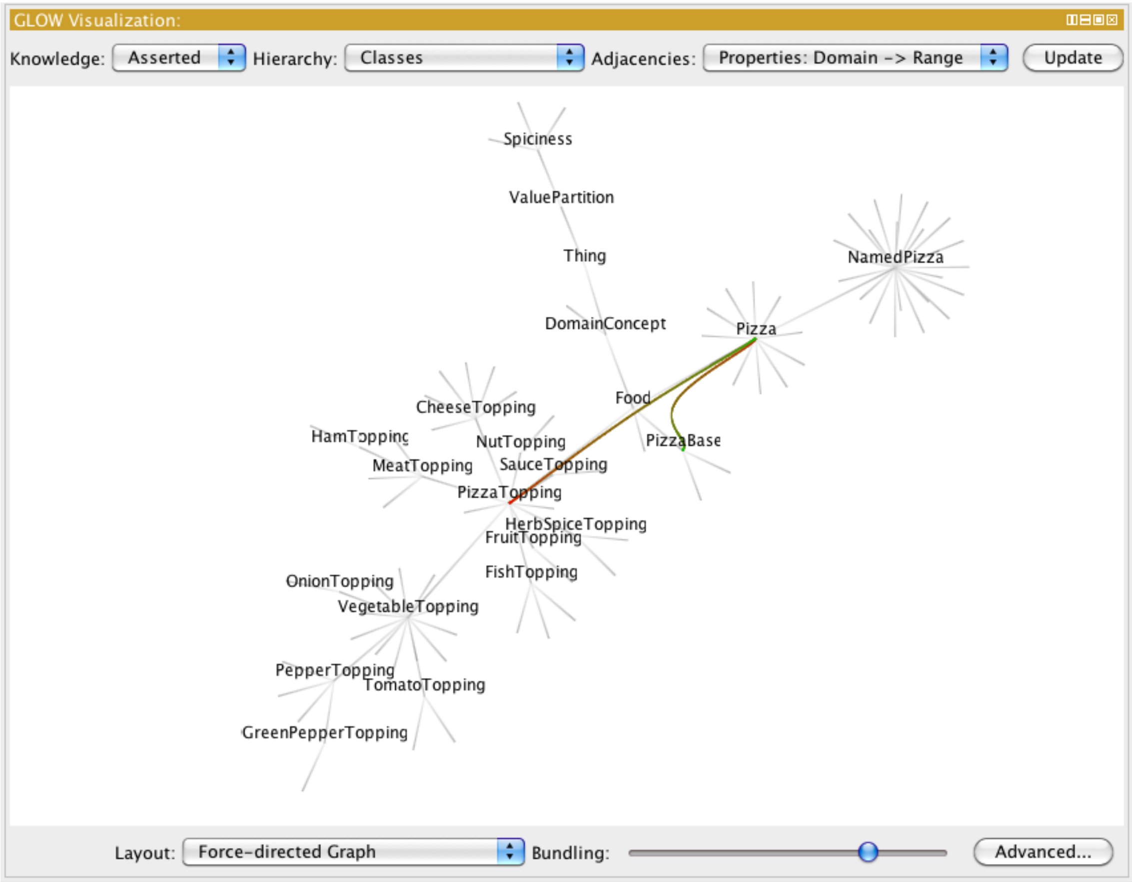

(KRIGLSTEIN; WALLNER, 2011)

Figure 3.13: OntoTrix tool. This technique is based on an hybrid approach for network visualization that uses both node-link and adjacency matrix representations.The example shows part of the ontology corresponding to men and women who live in cities in Judea, and are related to other people.



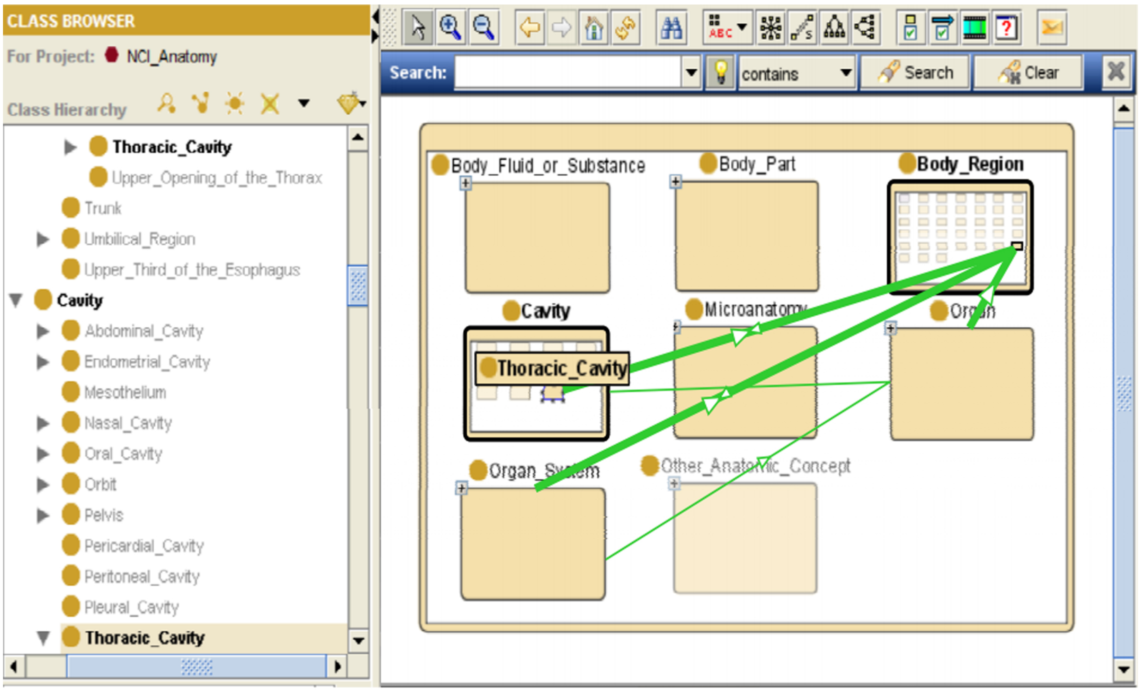

(BACH; PIETRIGA; LICCARDI, 2011)

Figure 3.14: Glow tool: a screen shot of a Glow view inside Protègè, showing the classes and object properties of the Wine ontology using a force-directed graph layout. Colored adjacency edges represent properties, and the endpoints of the adjacency edges correspond to domains and ranges of the properties. The leafs are related to the instances of classes.



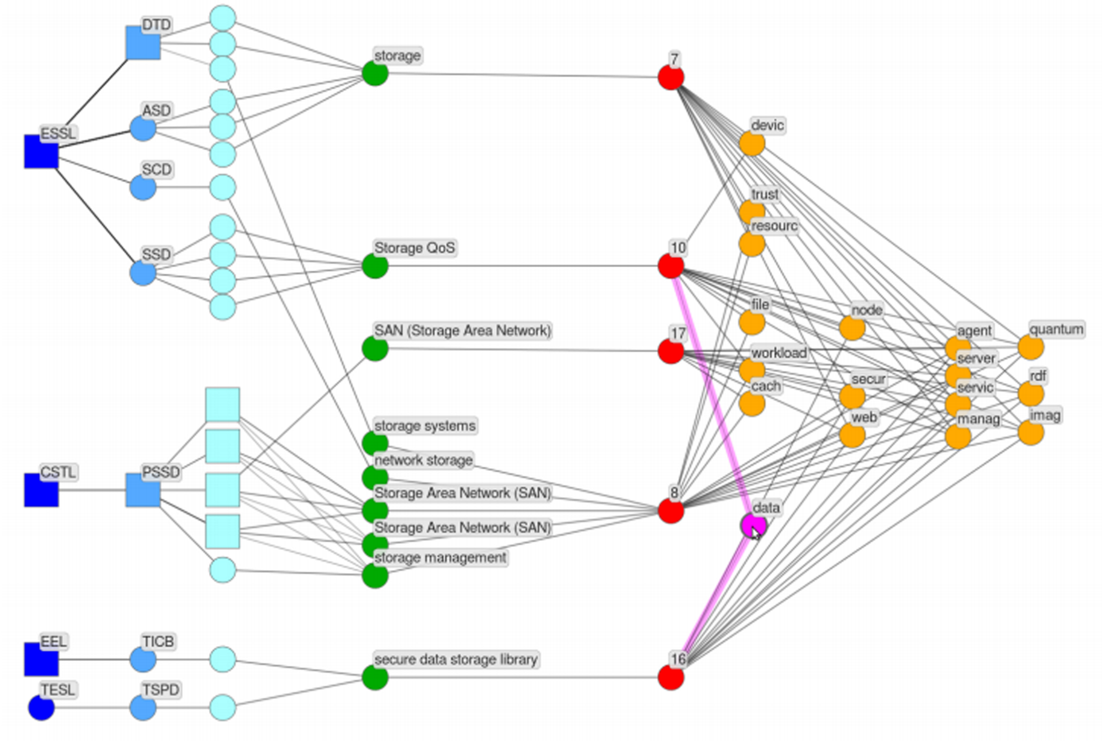

(HOP et al., 2012)

Figure 3.15: Diamond plugin for Protègè: highlighting concepts in Protègè's Class Browser and Jambalaya's nested graph views. Within Class Browser, non-interesting concepts are displayed in gray, interesting concepts are shown in black, and Landmark concepts are highlighted in black. Within Jambalaya's graph views, node labels are displayed using the same font colors and weights as the Class Browser, color intensity and line weight being used to highlight the nodes themselves.



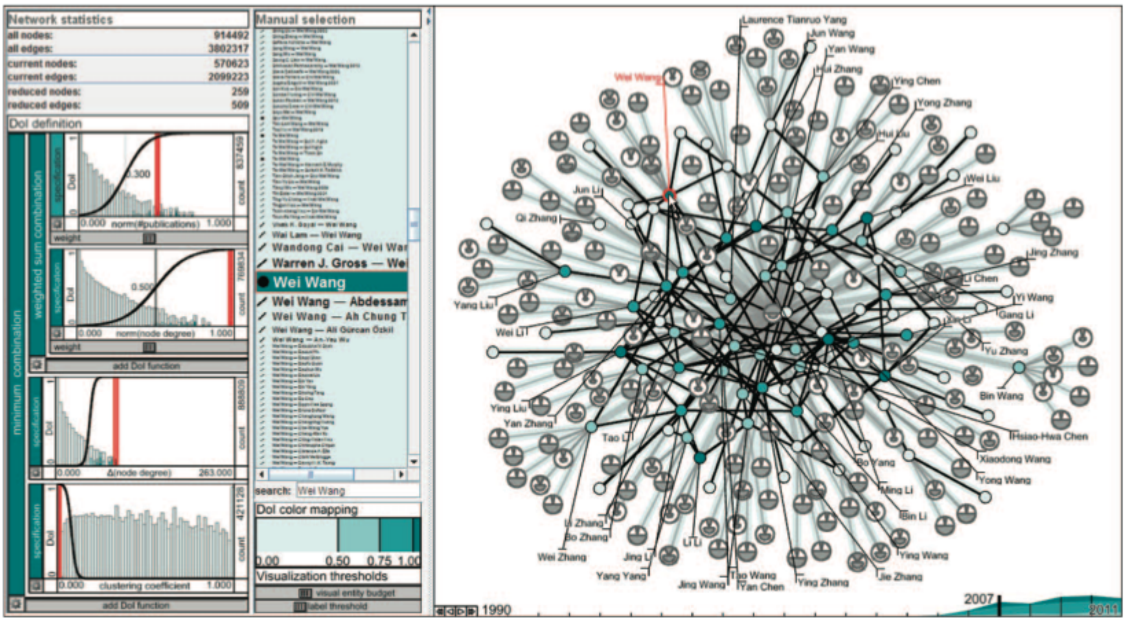

(D'ENTREMONT; STOREY, 2009)

Figure 3.16: Visual technique based on content-oriented analysis of an ontology about technical reports. This corresponds to a bipartite-graph-like view between document nodes (green) related to content "storage" and person nodes (cyan) of their authors and affiliations (departments and laboratories - blue and blue-cyan nodes). From the departments and laboratories nodes at the left, we know what organizational units have worked on this topic. From groups of documents (red nodes) and discriminating terms (orange nodes), users can quickly get a sense about the content of these documents.




(CHAN; KEETON; MA, 2010)

Figure 3.17: Abello et al.'s DoI approach with focus in large dynamic networks proposed with its two main views: (1) the DoI view and (2) the Network view. The Network view shows a snapshot of the DBLP dataset for the year 2007, which has been reduced according to the defined DoI function.




(ABELLO et al., 2013)

# 4 DEVELOPING THE APPROACH FOR ONTOLOGY VISUALIZATION

*This chapter presents the methodology adopted in the development of the present work. We start presenting our first proposal for ontology visualization, which was based on the analysis of InfoVis techniques for graphs and trees, and existing tools (Chapters 2 and 3). Then, we report the requirements elicitation through interviews with experts and, at the end, we discuss the evolution from the first proposal to the present stage of tool development, followed by results from the evaluation performed by ontology experts.*

## 4.1 Analysis of Existing Ontology Visualization Techniques

This study follows the iterative cycle of refining designs and getting user feedback as proposed by Freitas et al. (2012), Forsell and Johansson (2010), Isenberg et al. (2008) and Munzner (2006) (see Section 2.2.3). A summary of these techniques is shown in Table 4.1, illustrating the method we adopted for developing our work.

As mentioned before, many of tools analyzed for ontology visualization in chapter 3 are based on 2D graphs, the most intuitive representations of relationships (edges) between concepts (nodes) of ontologies due to their hierarchical and relational characteristics. However, the overlapping edges can be a problem for the efficiency of information display. Although one can add interaction to solve part of the problem, allowing the user to select the information he/she wants to put into focus, overlapping edges remain a problem (Figure 4.1).

We studied the hypothesis of representing ontologies in a 3D space, allowing the user to navigate through in-depth visual representations, rotating, expanding and selecting the desired items. However, such views require the user immersion and depth perception is crucial (Figure 4.2).

Considering these aspects, in the next section we present our initial propose for a visualization method that fits the tasks listed in Table 4.2 (this is the same as Table 1.1 repeated here for reading convenience) and solves the main problems of 2D and 3D visualizations.

## 4.2 First Approach: 2.5D Visualization of Classes Hierarchy and Relationships

In the first part of our study, we have chosen to focus on visualizing the hierarchy of the ontology and the relationships between concepts. Besides the class hierarchy (relationship "is a"), users of ontologies need to analyze, in an integrated way, the other

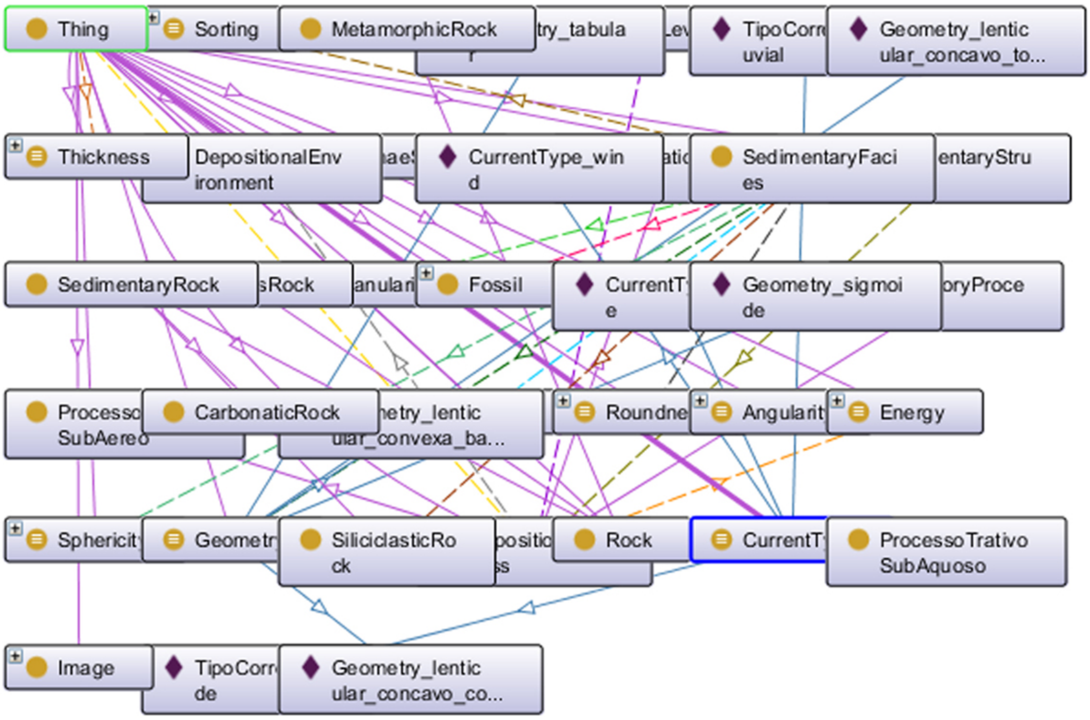
Table 4.1: Evaluation Techniques for InfoVis.

| Authors | Proposal | Methods |
|---|---|---|
| Munzner (2006) | Nested model | Domain problem characterization; data/operation abstraction design; encoding/interaction technique design; algorithm design. |
| Isenberg et al. (2008) | Grounded evaluation | Evaluation; design; implementation cycle. |
| Forsell and Johansson (2010) | Heuristics | Information coding; minimal actions; flexibility; orientation and help; spatial organization; consistency; recognition rather than recall; prompting; remove the extraneous; data set reduction. |
| Freitas et al. (2012) | Guidelines | To define the context of usage before the beginning of evaluation; to know who the users are, and to decide which users tasks to support; to understand which tasks users have to perform and their characteristics (steps, constraints, and task attributes like frequency, priority, etc.) and to decide which tasks to support; evaluating early. |

Table 4.2: Tasks related to ontologies and information visualization techniques, adapted from (KATIFORI et al., 2007).

| Task | Description | VI Techniques |
|---|---|---|
| Overview | Gain an overview of the entire collection | Trees and graphs, 3D landscapes, treemaps (space filling) |
| Zoom | Zoom in on items of interest | Indented lists, trees and graphs, 3D landscapes |
| Details-on-demand | Select an item or group and get details when needed | Trees and graphs, 3D landscapes |
| Filter | Filter out uninteresting items | Indented lists, trees and graphs |
| Relate | View relationships among items | Indented lists, trees and graphs, zooming, 3D landscapes |
| History | Keep a history of actions to support undo, replay and progressive refinement | - |

Figure 4.1: Ontology visualization with Protègè  OntoGraf




(FALCONER, 2010)

ontology relationships. Thus, we actually have a graph along with a tree, but end up with the problem of occlusion of information due to the overlapping edges.

This problem can be solved with the use of a third dimension to display one or more relationships (object properties) selected by the user. To view them, in our first approach, we took the plane where the tree is displayed and performed a 90 degrees rotation around the X-axis (see Figure  4.3). The rotated plane, positioned in 3D as an XZ-plane, displays the hyperbolic tree, and selected relationships are represented as curved lines in space, connecting the related concepts, without interfering with the display of the hierarchical relation.

In addition to rotations around the X-axis, rotations around the axes Y and Z, zoom and pan are also allowed, providing full 3D navigation.  Figure  4.4 shows the proposed 2.5D scheme applied to an ontology hierarchy/graph. We explored color and thickness of edges and line contours in nodes.

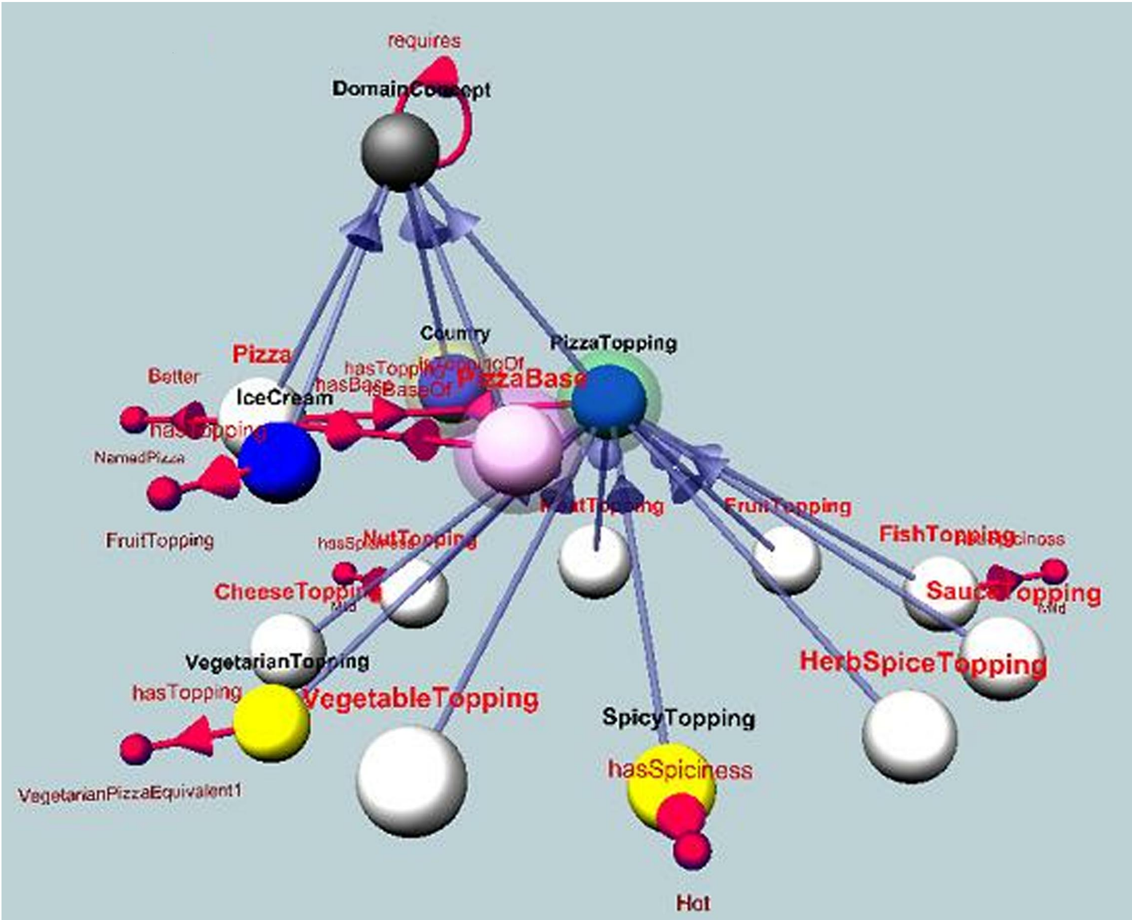
The resulting visualization allows the user to remain "aware" of the ontology hierarchy and to visualize one or more relationships in a separate spatial dimension.

In the next section, we report the interview with experts on creation and manipulation of ontologies in order to elicit relevant requirements for the proposition of a more powerful ontology visualization tool.  These interviews corresponded to the preliminary assessment of our first prototype.

## 4.3 Requirements Elicitation

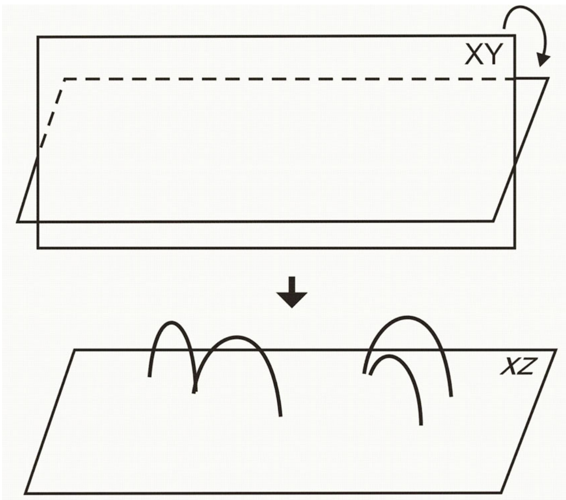
After designing our first idea for the visualization of relationships between classes of ontologies, and before the presentation of the visualization proposed in the previous section, we performed interviews with expert users of ontologies to elicit requirements

Figure 4.2: Ontology visualization with OntoSphere




(BOSCA; BONINO; PELLEGRINO, 2005)

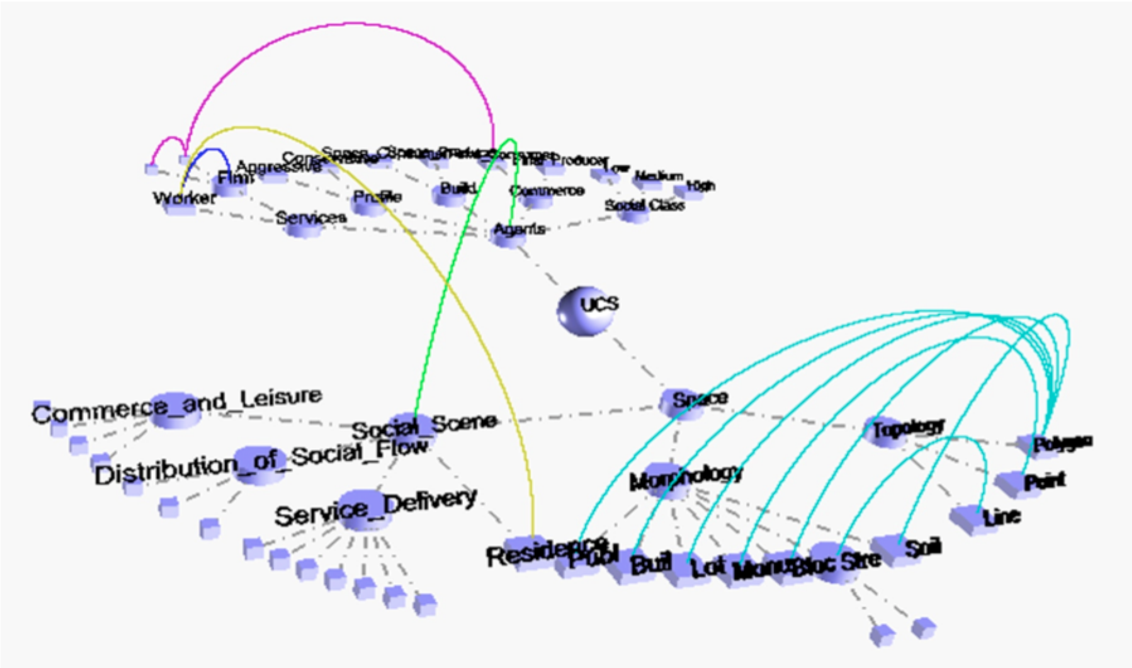
Figure 4.3: 2.5D visualization scheme




and collect impressions related to existing ontologies visualizations propositions.

We interviewed four experts responsible for creating and manipulating ontologies, maintaining intelligent databases and ontologies representation. Due to the low number of participants, quantitative measurements were not taken, but the qualitative notes were

Figure 4.4: 2.5D visualization of classes hierarchy and relationships generated with Java 3D API (ORACLE, 2012)




very interesting, confirming indications by Nielsen (NIELSEN, 1995), and recently, by (ISENBERG et al., 2008). All interviewed people have experience with ontology visualization.

The following questions were posed to the experts

*(1) When an ontology is created, which aspects could be improved with visualization?*

*(2) After the ontology was created, which information is searched more often and how this information could be displayed in order to make understanding more efficient?*

*(3) When and why a visualization is better than another?*

For question (1), users responded that the main focus is on the elements that define the structure of the ontology. These elements refer to the relations between class and subclasses, between classes, and between the instances of classes. An ideal visualization tool should focus on the ontology kernel (question 3).

During the process of ontology development, users want to visualize different aspects of specific demands that arise in certain stages of development. Thus, display features could have privileged access at certain points, for example, checking the range of an object property. However, the development of ontologies is still a traditional activity, with no defined workflow and directly influenced by the domain.

Another important aspect related to question (1) is the visualization of the ontology validation generated by inference processes. Displaying errors can (should) be improved, with the proper indication of correction.

In relation to question (2), we must consider that ontologies can encapsulate a large amount of information (hundreds of thousands of classes and relationships, for example). Moreover, this large volume of information can be segmented into several distinct types (classes, attributes with different values, relationships between types and properties). Usually, users do not want to see these types simultaneously, due to the cognitive overload it would arise.

For example, clicking on a class X, relations with classes Y and Z should be enhanced.

These classes could be highlighted, while other parts of the ontology could loose focus. The highlight could be obtained through visual attributes such as color, transparency, shapes and positioning. This feature would be very useful to get an idea of organizing an ontology (part-to-whole relationship, part-to-part relationship). Clicking on a main class could be easy to identify the classes that represent the parts.

The relationships properties (transitivity, reflexivity, symmetry, if it is functional or not) are an important structural component, because they have impact onto the inference that can be performed with the ontology. Likewise, the attributes of each class (data properties) should be considered in the visualization.

Regarding question (3), the main problems of current tools for ontologies visualization are common to any tool for graph visualization: problems of scale versus amount of information that need to be presented. An alternative would be to use different visualization techniques. According to Gurr (GURR, 1999), visual representations can be constructed in order to express the properties of a concept. The use of tooltip texts can help in the encoding of the displayed information, because they contain high loads of information and are presented selectively as the user explores the visualization of the ontology.

Finally, a simple but important suggestion from the users was that views of ontologies fit on an A4 format, with sufficient level of detail. It would also be interesting to have a tool that allows adding and removing elements of the visualization in a quick and simplified mode.

From such results, we reached the following requirements for ontology visualization:

- Provide overview of hierarchy ontology, with the possibility of detailing some parts;
- Avoid presenting the different aspects of an ontology (classes, description, object properties, data properties, individuals) together in a unique visualization;
- Optimize the results from the ontology validation;
- Explore the use of visual attributes such as color, transparency, and shapes;
- Provide display filters based on different techniques of focus+context and/or overview+detail, zoom, pan and rotation of the image;
- Allow rapid and simple inclusion of visual elements in the visualization, as well as their removal.

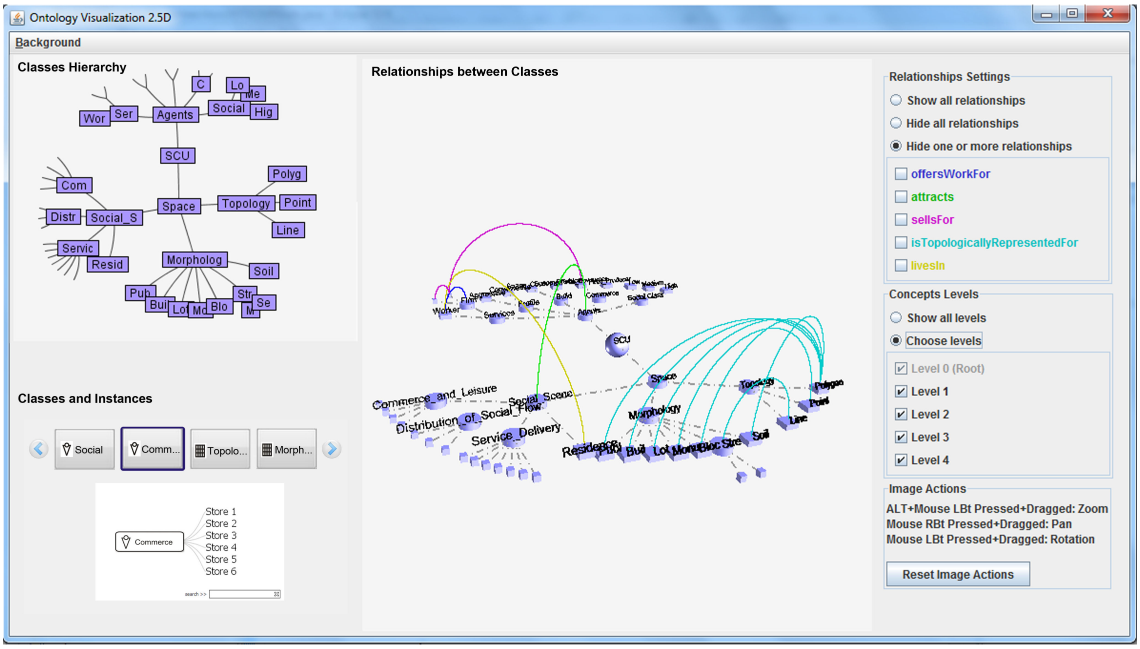
Our 2.5D visualization was presented to the four users after they were interviewed. Informally they approved the new possibilities for displaying and interacting with the ontologies represented in that way. However, we extended the visualization for a multiple view tool as described in the next section and Chapter 5.

## 4.4 Second Approach: Initial Prototype Proposition

We propose a initial visualization tool (Figure 4.5) that fits the requirements pointed out by users as well as the tasks listed by Katifori et al. (KATIFORI et al., 2007) (see Table 4.2). In this step, we have chosen to focus on visualizing the hierarchy of the ontology and the relationships between concepts employing multiple views although some ideas for instances visualization were tested.

For the hierarchy visualization, we employ the 2D hyperbolic tree (upper left corner in Figure 4.5 and Figure 4.6 ), a focus+context technique developed by researchers of

Figure 4.5: Initial proposition for our ontology visualization tool




the Xerox PARC Lab at the beginning of the 1990s (LAMPING; RAO, 1994). In the hyperbolic tree technique, the root is initially placed in the center of a circular area with the child nodes around it, their child nodes placed around them and so forth in a radial arrangement. While moving from the center of the tree to the border of the circumference, the distance between the tree levels is reduced in a way that, as a result of the hyperbolic representation, the whole tree fits in the circular area. The outer nodes are not displayed (MAZZA, 2009).

Some authors, like Souza et al. (SOUZA; SANTOS; EVANGELISTA, 2003) and Katifori et al. (KATIFORI et al., 2007), discuss the advantages of the hyperbolic tree for ontology visualization. This technique provides interactive features that allow representing very large trees. Hyperbolic trees reduce the cognitive overload and the user disorientation that might happen during the interaction with the nodes, expansion and contraction, especially in ontologies with many concepts.

The second view (center of Figure 4.5) displays the proposal described in Section 4.1: a 2.5D radial tree with arcs showing one or more relationships (object properties) selected by the user. We developed a control panel (see lateral tab in Figure 4.5) in order to allow the user to choose which levels of the tree view or hide as well as individual relationships, reducing cognitive overload.

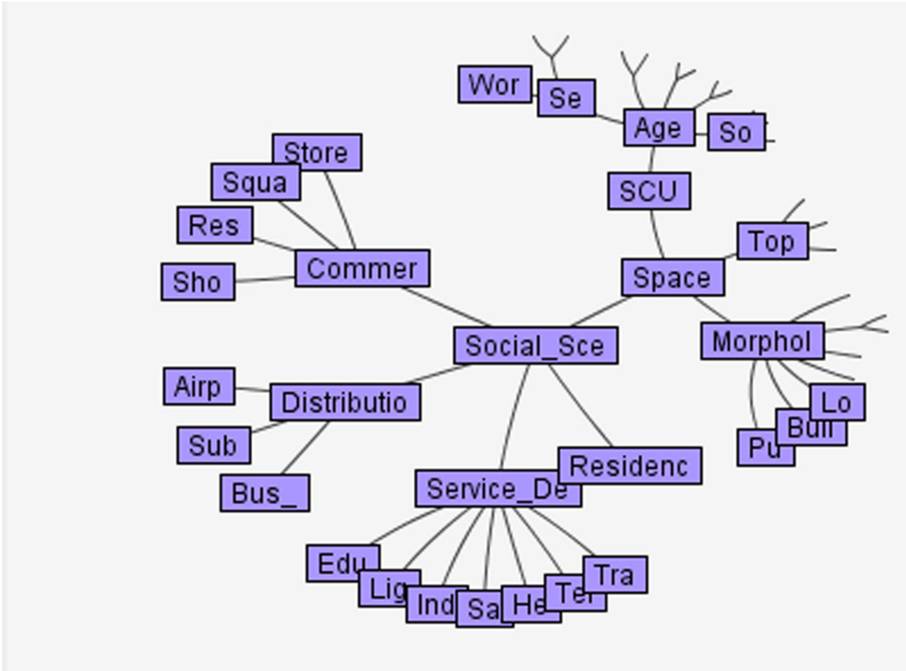
In 2.5D, we prefer to employ the radial tree instead of the hyperbolic representation because the radial layout has straight lines between nodes while maintaing the same rules of interaction of hyperbolic trees. Thus, it is easier for the user to differentiate the relationships "is a" (edges of radial tree) from other relationships represented as curved lines in space.

Summarizing, the main aspects of visualization and interaction of both methods are:

- Nodes are displayed with different geometric forms according to their type (root, subtree and leaf).

- Edges of hierarchy are displayed with solid lines and edges of other relationships

Figure 4.6: 2D hyperbolic tree showing the ontology classes hierarchy generated with HyperTree Java Library (INXIGHT, 2001)




are displayed with dashed curves, the colors being related to relationship type.

- In 2D hyperbolic tree view, the user can choose which nodes will be in focus on the image, hiding the other ones.
- Both 2D and 2.5D views can be displayed together, side by side, so the user remains "aware" of the ontology hierarchy and visualizes one or more relationships in a separate spatial dimension.
- The user can choose to display one or more relationships at the same time or hide them.
- In the 3D representation, the user can choose which levels of the tree view or hide, reducing the cognitive overload.
- In addition to rotations around the X-axis, rotations around the axes Y and Z, zoom and pan are also allowed, providing full 3D navigation.
- Button for reset the 2.5D visualization to the initial appearence.
- The background color can be changed.
- Size of nodes' labels decrease according to the number of nodes sibling and levels of the tree.
- Tooltips are displayed over nodes and edges as additional information.

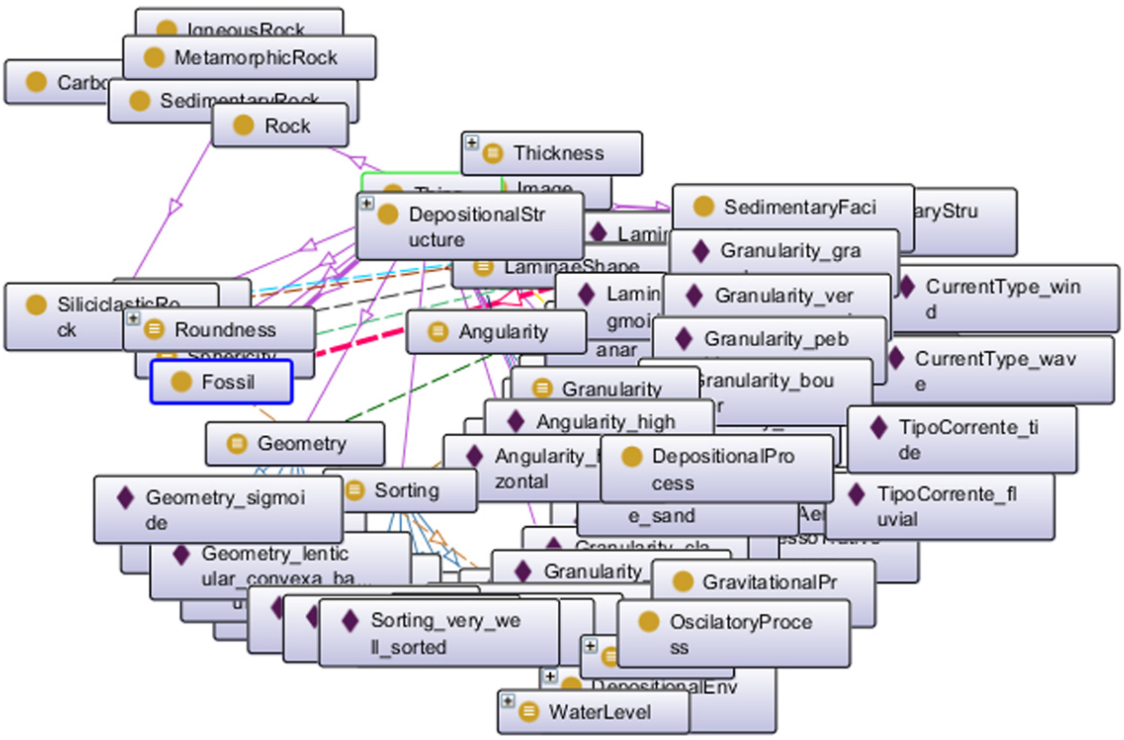
Such usability features aimed at reducing the cognitive effort of the user in analyzing the image and, at the same time, add functionality to the tool.

Finally, on the third view, at the bottom left corner, there is a visualization for showing instances of ontology classes, which employed a schematic treeview. The classes of the ontology are exhibited as miniatures at the topmost part of the visualization (overview area), and the whole set can be navigated using shift buttons as in common pictures displays. The detail area shows the selected class in treeview representation at the center of this view. This approach helps users to understand how the entire collection is organized, keeping both views visible for rapid interaction. The classes can also be found by a search function of a typed keyword.

## 4.5 Qualitative Evaluation of the Proposed Visualization

In order to evaluate our visualization method for both classes hierarchy and relationships, we have chosen to compare it with Ontograf (FALCONER, 2010) that presents seven visualization possibilities: alphabetical grid, radial and spring graphs, and four implementations of tree visualization: vertical, horizontal, directed vertical and directed horizontal. In this step, instances view was not evaluated. Figure 4.7 shows an example of the Ontograf visualization.

Figure 4.7: Ontograf tool visualization example




(FALCONER, 2010)

The four specialists interviewed in the first phase of our study (as described in Section 4.2) were invited again to perform an evaluation of our 2.5D visualization and Ontograf. Moreover, we invited two other specialists in ontology specification to participate (they also have experience with ontology visualization), so we had a sample of six specialists.

For the evaluations we used two ontologies: a large ontology describing Stratigraphy concepts, and a smaller one, representing cities' urban concepts. Before the participants started with the tasks, we shortly introduced them to the important functionalities of the tools related to visualization and manipulation (not creation and edition), and they explored them in many ways using a different training ontology. After the subjects had finished their training, we started the evaluation process.

The tools were presented in different order for users. For each tool, they were asked to perform an analysis based on four questions that were defined in order to obtain the requirements listed in Section 4.2. The questions are listed below:

*(1) Is the initial layout clear?*

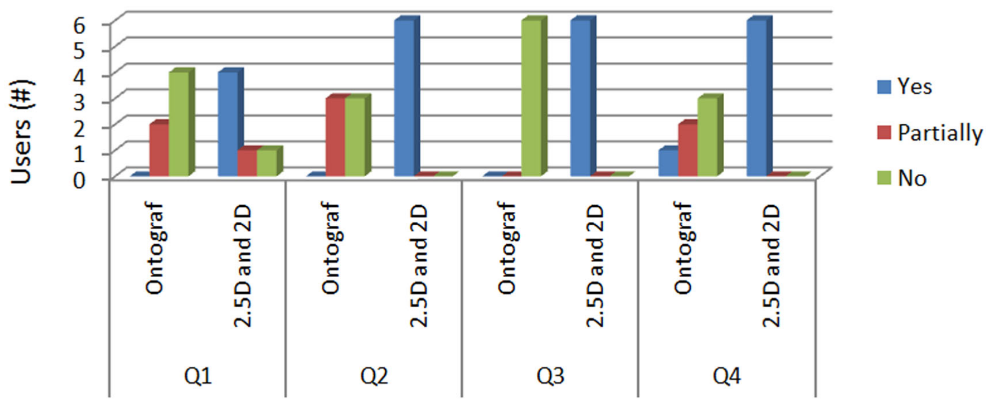
*(2) Is it possible to clearly separate the concepts hierarchy from the other relationships between these concepts?*

*(3) Does the possibility of rotating the ontology representation improve the analysis of relationships?*

*(4) Do the pruning and expansion of the ontology levels help the understanding of hierarchical relationships?*

Three possibilities of answers were defined: Yes; Partially; No. Figure 4.8 summarizes the users answers for these questions.

Figure 4.8: Summary of evaluation results




Regarding question (1), the majority of users (67%) responded that the initial 2.5D layout is clearer when compared with Ontograf. Among the reasons for that, users pointed out the large amount of information displayed at the same time (nodes overlap) in the image of Ontograf. This is a problem of scale versus amount of information, and causes user disorientation. In our 2.5D method, this problem is solved due the nature of the hyperbolic tree.

In relation to question (2), users were divided (50%) between "Partially" and "No" answers for Ontograf, because nodes and edges overlap. Usually, users do not want to see relationships simultaneously, due to the cognitive overload it would arise. Thus, the possibility of analyzing the "is a" (hierarchy) and other relationships in different dimensions helps the user to understand the ontology. Another problem indicated for Ontograf is that the user needs to change the positions of nodes in order to reveal the relationships occluded by them.

An important positive aspect noticed by users in both tools is the presence of tooltips when the mouse is over the nodes or relationships. The use of tooltip texts can help in the encoding of the displayed information, because they contain high loads of information, and are presented selectively as the user explores the ontology.

Users also approved different colors for different types of relationships. Colors are mainly a resource for information categorization, and graphical elements like shapes and location of elements in the space help the user in mapping the concepts (WARE, 2008), and these features are present in our 2.5D method.

Regarding question (3), this functionality is not present in Ontograf, and the users considered it an important interaction mode. In our 2.5D method, rotations around the three axes (X, Y and Z) are possible, and complemented by zoom and pan. Thus, users have more freedom to interact with the visualization, and are able to reset to the original layout at any moment. One of the users reported that when interacting with the 2.5D view, he did not feel claustrophobia, which is common in other tools, including Ontograf.

Finally, in relation to question (4), while 100% of users answered ”Yes”, for the 2.5D view, for Ontograf, most users (83%) answered ”Partially” and ”No”. This result is due to the feature of Ontograf related to the repositioning of nodes when it is pruned or expanded, this fact causing disorientation on users. On the other hand, the 2.5D allows pruning and expansion in two ways: through the hyperbolic tree functionality of repositioning nodes, and through hiding/showing levels of the hierarchy.

These results indicate that the use of 2D and 2.5D visualizations might be a solution to common problems presented by 2D and 3D ontology visualization tools, mainly cognitive overload and user disorientation.

Based on these results, we came up with an approach for ontology visualization, which is described in the next chapter, and implemented it as a visualization tool named OntoViewer.

# 5 ONTOVIEWER - ONTOLOGY VISUALIZATION TOOL

*In this chapter we present OntoViewer, the tool we have developed for ontology visualization and analysis at both the intensional and extensional levels. OntoViewer was designed following a set of guidelines defined for the evaluation of interactive techniques intended for visualization of ontologies.*

## 5.1 Design and Evaluation Guidelines for Ontologies Visualization

Visualization systems are mainly based on a mapping from information to a graphical representation in order to facilitate data interpretation. Usually they should provide ways to limit the amount of information that users receive, while keeping them "aware" of the total information space and reducing cognitive effort. For ontologies visualization, this rule is also true.

Ontologies tend to grow, incorporating new concepts and relationships, therefore increasing the representation complexity, and consequently, the visualization complexity. Static graphs, commonly used for ontology representation, are not the best alternative for such visualizations. One needs efficient visualization and interaction methods tailored for ontologies since the challenge is to define the best way to represent relationships (object properties) between categorized concepts (classes), mainly because each concept can have any number of relationships, attributes (data properties) and instances (individuals).

The representation of concepts, relationships and attributes correspond to the intensional level of the ontology, while the instances of the classes with concrete data values for attributes correspond to the extensional level. Visualization tools must consider the specificities of these two levels, allowing the user to interact with each one individually as well as analyzing them in a combined and efficient form.

Qualitative information collected from interviews with experts is considered very representative (NIELSEN, 1995) because it aims at achieving the understanding of the interplay between factors that influence visualizations, their development and their use. The interview with experts, reported in Chapter 4, and the analysis of the InfoVis techniques for ontology visualizations, presented in Chapter 3, allowed us to obtain the adequate insight for proposing an ontology visualization scheme.

Moreover, from such qualitative data based on the accumulated experience of the experts and researchers, we identified some basic guidelines for the design and evaluation of visualizations applied to ontologies, focusing on different aspects of the intensional and extensional levels:

a) *The visualization must combine dimensions of quality, as classes, relationships, attributes, instances.*

This feature helps the engineer to have an overall view of the global status of the ontology in terms of the dimensions of quality, thus giving an idea of its complexity. An ontology is something more than a hierarchy of concepts since it is enriched with role relations among concepts and each concept has various attributes related to it.

According to Mazza (MAZZA, 2009), in these cases, complex data with different quality dimensions are best viewed if their attributes are mapped to proper graphical elements and properties for creating effective visual representations of data.

b) *The user must be able to focus, separately, both on the intensional and extensional levels of the ontology, and to switch easily from one to the other.*

If the focus of the user is on a concept C at the intensional level, what does it mean to switch to the extensional level? The most intuitive behavior is to switch the visualization, keeping the focus on the selected concept.

Baldonado and co-authors (BALDONADO; WOODRUFF; KUCHINSKY, 2000) point that the presence of diversity is one of the foremost reasons for designing a multiple view system because it can help users to understand complex relationships when coupling two or more views which can show hidden relations.

c) *If there are different views related to intensional and extensional levels, they should be synchronized and complement each other.*

In fact, Roberts (ROBERTS, 2007) discusses that the overall premise for the employment of multiple and coordinated views is that users understand their data better if they interact with it and view the information through different representations. Textual and graphical views can be used to optimize analyses.

d) *The user should be able to customize the interface in order to focus at only a subset of the information, reducing the cognitive overload.*

Common approaches for ontologies visualization tools (KATIFORI et al., 2007; KHAN; KHAN, 2011) as overview+detail and focus+context, and interactive tasks as zooming, panning, selecting, linking, filtering, rearranging and/or remapping can assist this process.

e) *Parallel to the graphical visualization of intensional and extensional levels of the ontology, textual information can help the analysis of different aspects, especially when the ontology has a complex structure.*

Tooltips and indented lists (or treeview) are examples of textual information that complements, in a intuitive manner, more complex visualizations. This approach is employed by most ontology editors as Protègè (NOY; FERGERSON; MUSEN, 2000).

These guidelines are specific to the problem of interactive visualization of intensional and extensional levels of ontologies. However, we consider that it is also important to combine such guidelines with general information visualization heuristics as proposed by Forsell and Johansson (2010)(see Section 2.2.3): information coding, minimal actions, flexibility, orientation and help, spatial organization, consistency, recognition rather than recall, prompting, remove the extraneous, data set reduction.

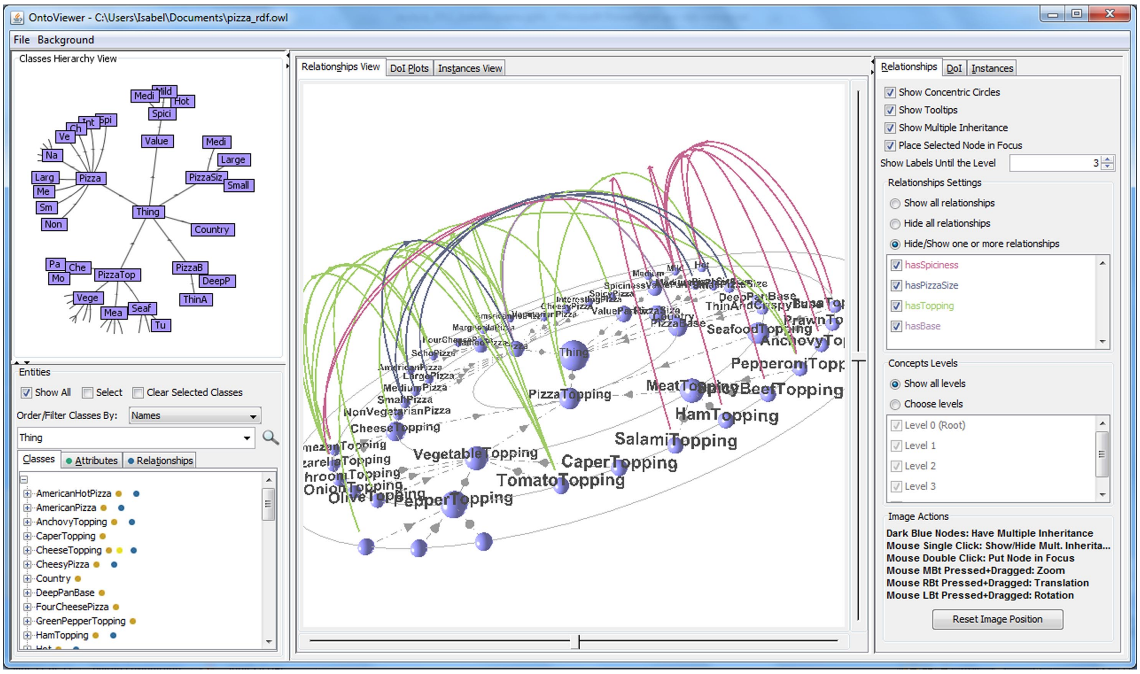
## 5.2 OntoViewer Description

In order to explore the visualizations proposed in multiple and coordinated views, we have built a tool called OntoViewer. The design of this tool was based on the requirements elicited from experts and reported in Section 4.2. Besides that the analysis presented in Section 3.3 is also a motivation for the visualization scheme proposed in OntoViewer as will be discussed here and shown in Table 5.1. We followed a user-centered perspective aiming at a cycle of design and evaluation as presented in (FREITAS; PIMENTA; SCAPIN, 2012) and (ISENBERG et al., 2008). We adopted the guidelines described above (Section 5.1) for helping design and evaluation.

The tool was developed in Java 2D, Java 3D and OWL API, and supports the OWL formalism.

Considering Shneiderman's visual information seeking "mantra" (SHNEIDERMAN, 1996): "First, **overview**, then, **zoom** and **filtering**, and finally, **details on demand.**", we employ multiple and coordinated views and provide automatic analysis of the ontology's concepts, relationships, attributes (intensional level) and instances (extensional level). Figure 5.1 presents an overview of OntoViewer. We can observe well-defined areas that combine information visualization techniques and interaction modes (zoom, pan, selecting, linking, filtering and rearranging).

Figure 5.1: Ontoviewer user interface




### 5.2.1 Visualization of Intensional Level

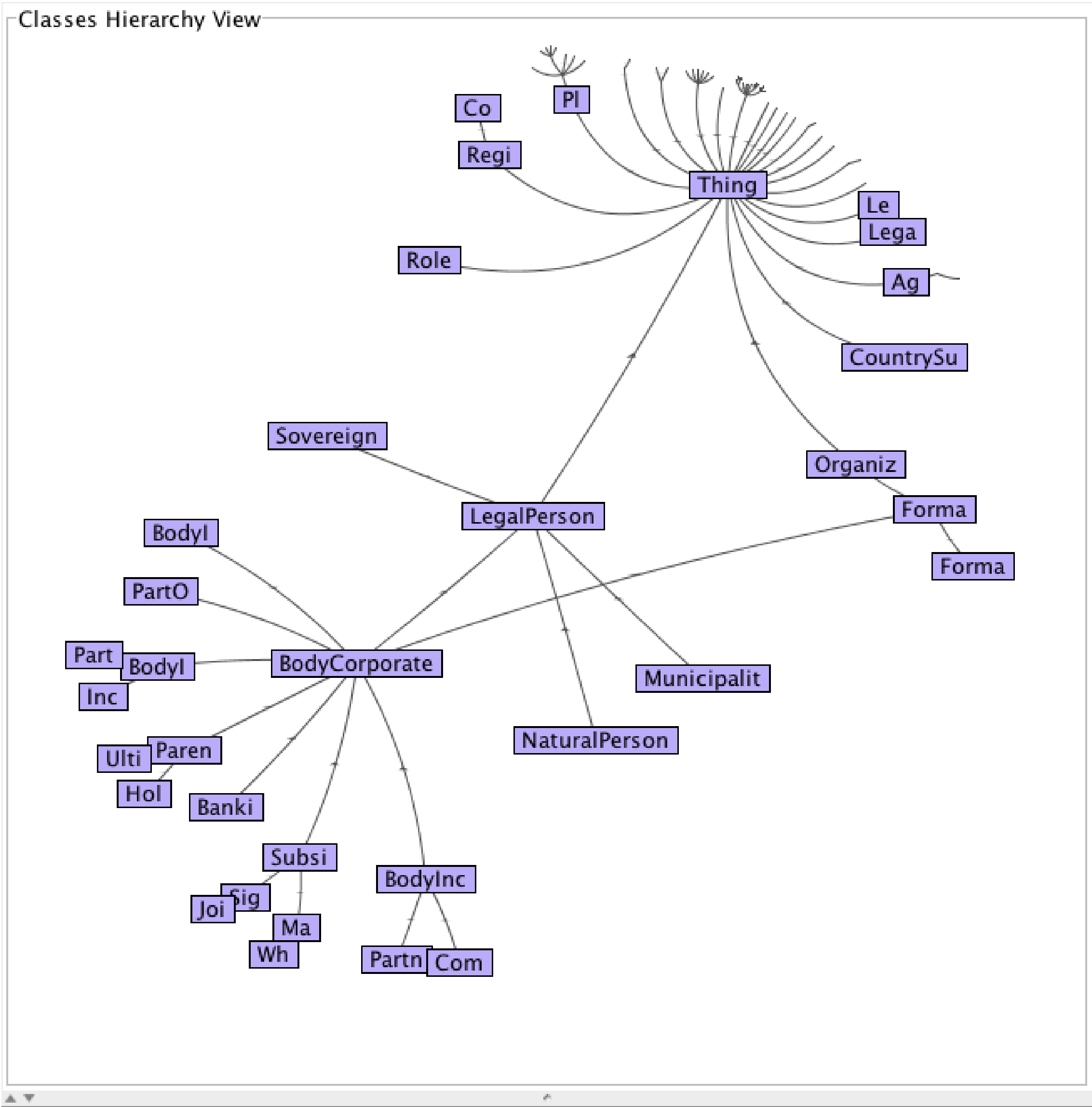
Figure 5.2 shows the view for representing the classes' hierarchy. For this one, we employed a 2D hyperbolic tree, a focus+context technique. This technique allows dragging and dropping classes, selection, and changing the displayed hierarchy dynamically (see more details in Section 4.3).

Although the hierarchy is represented as a tree, sometimes we need to represent classes with multiple inheritance. So, if a class has more than one superclass, we draw

Table 5.1: Characteristics in related works compared to OntoViewer.

| raggedright Visualization Technique | Display of Intensional Level | Display of Extensional Level | Multiple and Syn-chro-nized Views | User Task Support | Automatic Analysis |
|---|---|---|---|---|---|
| Protègè Class Browser | Yes | No | No | Yes | No |
| OWLViz (Protègè plugin) | Yes | No | No | Yes | No |
| Ontograf (Protègè plugin) | Yes | Yes | Yes, in-tegrated with Protègè Class Browser | Yes | No |
| TreeMap | Yes | No | No | Yes | Yes |
| Concept Modeller | Yes | Yes | Yes, in-tegrated with Protègè Class Browser | Yes | Yes |
| Hyperbolic Tree | Yes | No | No | Yes | No |
| On-Time | Yes | Yes | Yes | Yes | No |
| OWLeasyViz | Yes | No | No | Yes | No |
| Knoocks | Yes | Yes | Yes | Yes | No |
| OntoTrix | Yes | Yes | No | Yes | No |
| Glow | Yes | Yes | No | Yes | No |
| Diamond (Protègè plugin) | Yes | No | Yes, in-tegrated with Protègè Class Browser | Yes | Yes |
| Bipartite-Graph-Like | Yes | No | No | Yes | No |
| DoI and Large Dynamic Networks | Yes | No | Yes | Yes | Yes |
| **OntoViewer** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** |

Figure 5.2: Hierarchy of classes: 2D Hyperbolic tree view

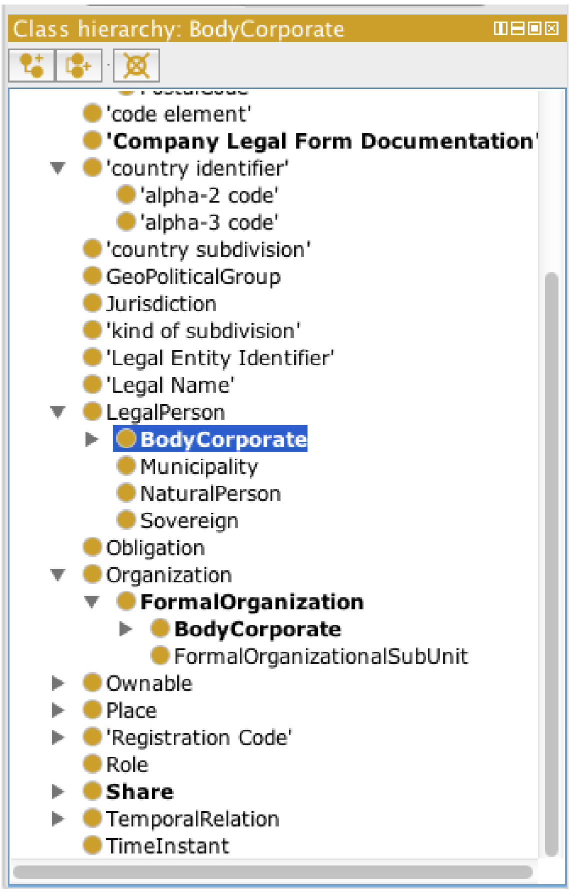



one more relation "is a" for this class, as can be seen in Figure 5.2 - the class "BodyCorporate" has two superclasses, "LegalPerson" and "FormalOrganization". We consider this representation more effective than that provided by Protègè Class Browser (NOY; MCGUINNESS, 2001), which is based on replicating the class "BodyCorporate", as can observed in Figure 5.3.

This solution for multiple inheritance was also implemented in the 2.5D radial tree, which is presented in Sections 4.1 and 4.3, another focus+context visualization (Figure 5.4). This technique displays the classes hierarchy in a XZ plane (edges with arrows indicating their orientation), and the relationships between these classes as curves in the 3D space, avoiding the overlapping of lines. The multiple inheritance was represented with the inclusion of oriented edges between the involved classes and color darkening of classes' spheres (Figure 5.5).

Interaction is performed through zooming, panning, rotation and selection. Tooltips inform the names of classes. The user can enables/disables the exhibition of tooltips, concentric circles, labels in nodes according the level and multiple inheritance, and the

Figure 5.3: Visualization of multiple inheritance in Protègè Class Borwser visualization




(NOY; MCGUINNESS, 2001)

node selection.

Borrowing the patterns of colors and information presentation from Protègè, OntoViewer provides a treeview (Figure 5.6) for listing the main aspects of classes, relationships and attributes of the ontology. In this view, the user can filter searches for classes, attributes and relationships (Figure 5.7). Classes can be search through the editable combo box with auto complete feature. When a class is selected (combo box or treeview), the other views change the visualization in a synchronized way.

### 5.2.2 Visualization of the Extensional Level

Commonly, instances of classes are modelled as branches of a node representing a class. Another common representation is lists. In OntoViewer, we aimed at providing a more intuitive visualization technique, based on users' requirements. The visualization of instances of classes, which compose the extensional level, is accomplished through an overview+detail method that uses a 2D icicle tree combined with a pixel-oriented approach, as presented in Figure 5.8.

The Icicle tree was employed for showing only the classes that have instances, reducing the information overload, and an overview of the instances number for each selected class. Icicle tree is a hierarchical technique that follows the familiar top-to-bottom or left-to-right tree structure (BARLOW; NEVILLE, 2001). When the user selects a specific class (Figure 5.9), in icicle tree or in the auxiliary combo box, the values of the instances' attributes are mapped to colors and actualized in icicle tree view, that is complemented

Figure 5.4: Relationships between classes: 2.5D radial tree view.

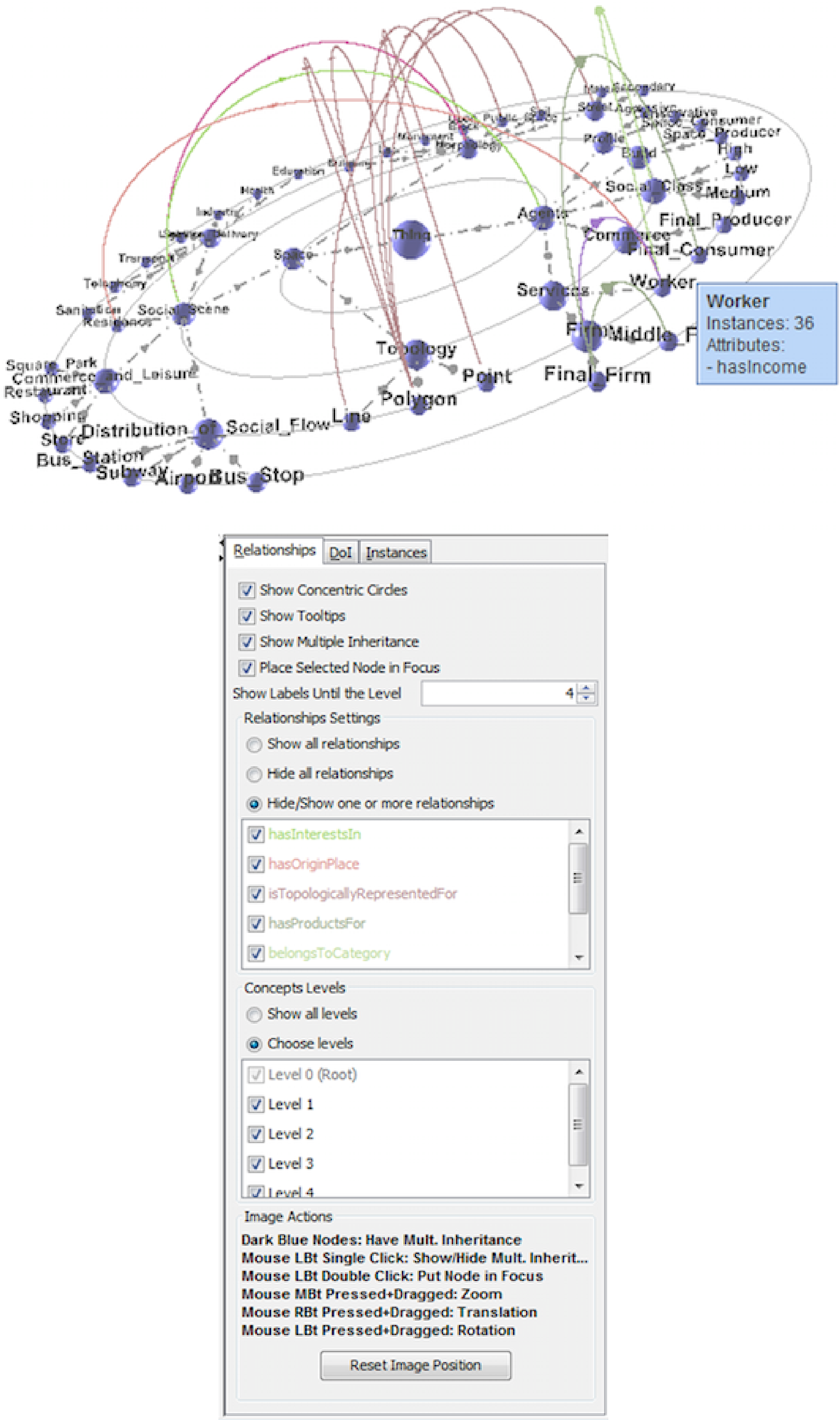
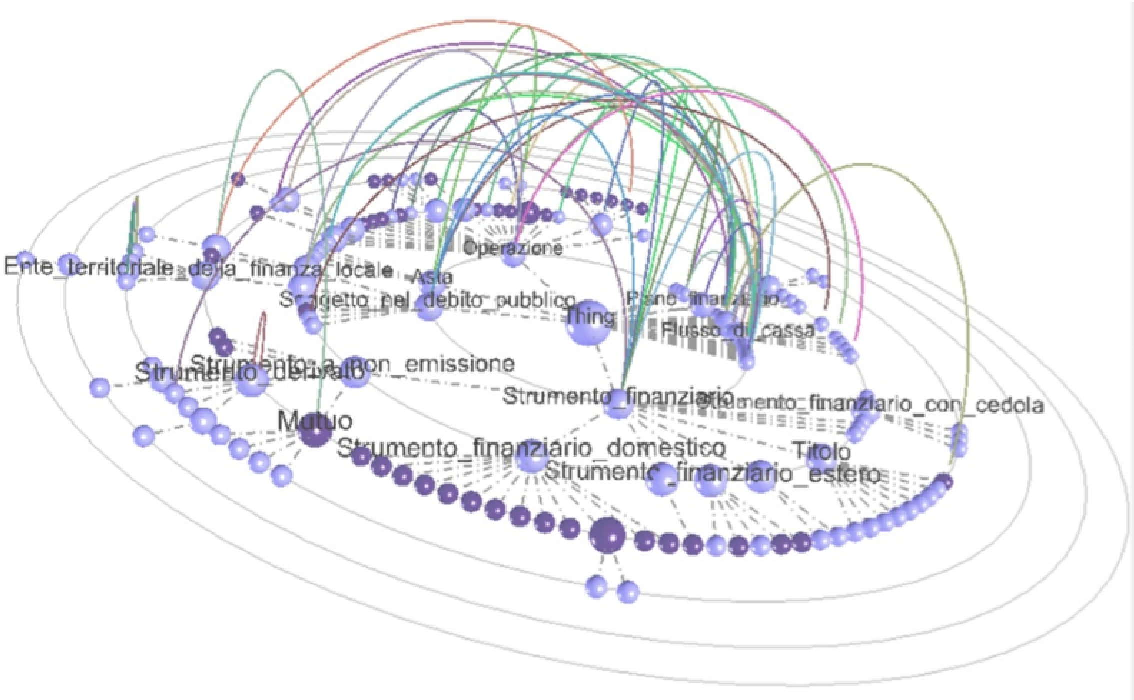


Figure 5.5: Visualization of multiple inheritance in 2.5D radial view




by a histogram and a second view, using a pixel-oriented technique (KEIM; KRIEGEL, 1996). Then, the user can observe trends in these values.

Figure 5.10 presents a combined view of both intensional and extensional levels. We can see the 2D pixel-oriented view of attributes values of classes instances along with the 2.5D view of classes hierarchy and relationships. These two views allow the visualization and analysis of trends in attributes of instances of classes that share relationships.

All these views and tab controls can be hidden or shown according to users' preferences and/or needs.

### 5.2.3 Ontology Analysis

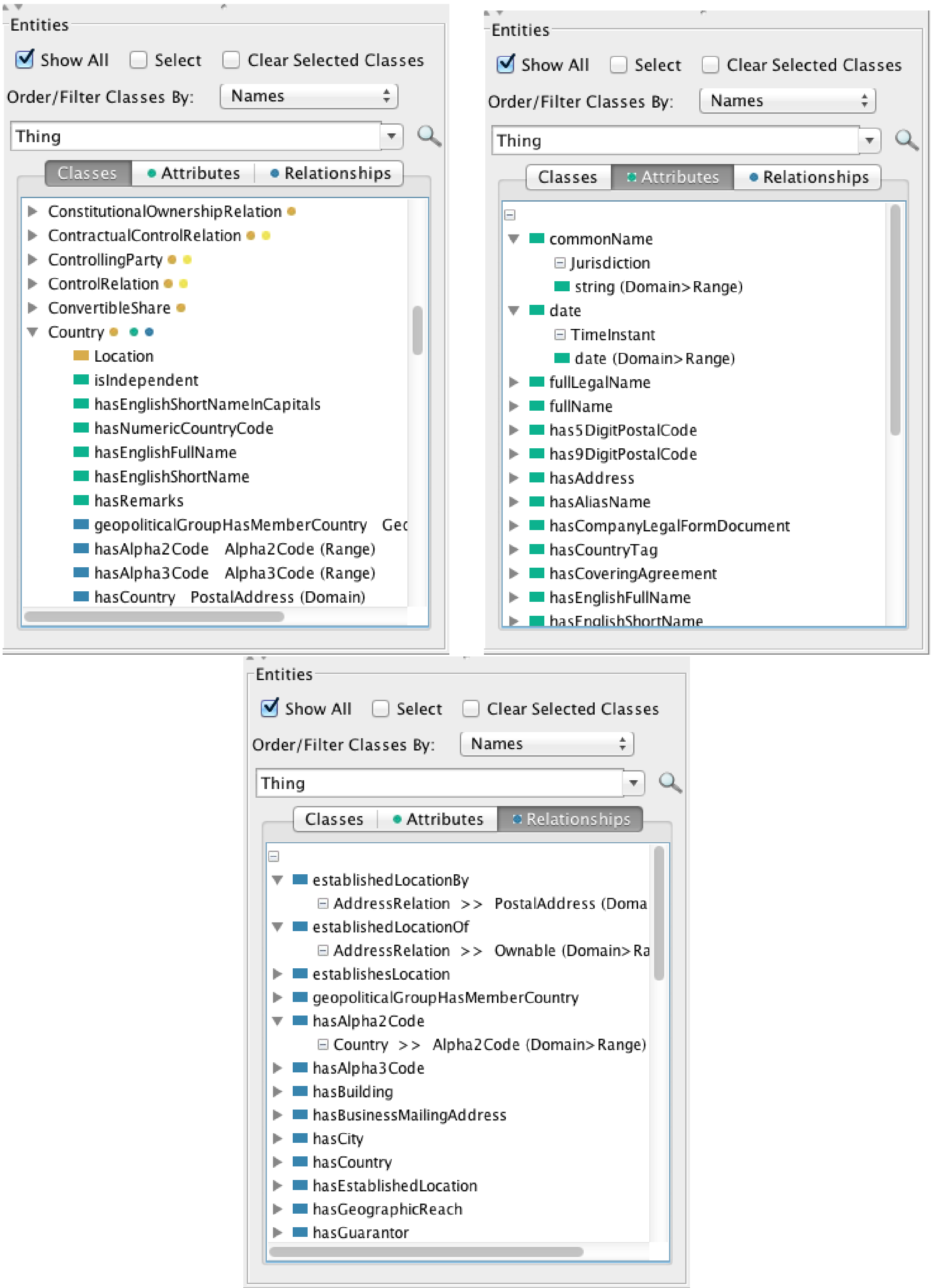
Visual Analytics can improve both quality and efficiency of ontology visualization systems, providing semi-automatic means for driving the visual exploration.

In order to cope with very large ontologies, we have chosen a suppression technique (FURNAS, 1986) based on the notion of *Degree of Interest* (DoI) that, from the automatic analysis of an ontology's intension and extension, extracts knowledge about the relevance of concepts and relationships according to the user task. That technique allows exploring ontologies entities focusing on a main concept and having the view of the most relevant concepts and relationships automatically computed and displayed.

Developing an ontology (see (NOY; MCGUINNESS, 2001)), includes four main aspects: defining classes, arranging such classes in a hierarchy, defining relationships among classes, and defining instances of classes and relationships.

According to this, we model an ontology as a tuple $O = (C, H, R, I_C, I_R, A)$ (adapted from (EHRIG; SURE, 2004)). Concepts $C$, which are classes of real-world objects, are organized in a hierarchy $H$; relationships $R$ exist between pairs of concepts, describing properties of classes and instances. $I_C$ is the set of the instances of all concepts and $A$ are the concepts' attributes (also referred as classes' properties); $I_R$ are the instances of the relationships.

Figure 5.6: Intensional Level Entities: Treeview




We represent an ontology as a graph $G = (V, E \cup OE)$, where vertices $V$ are the concepts $C$, edges $E \subseteq V \times V$ are the relationships $R$ and the oriented edges $OE \subseteq V \times V$ are the classes' hierarchy $H$ $(E \cap OE = \emptyset)$. Moreover, to model the intensional part of the ontology and the *A Priori Importance* (API) of classes and relationships we introduce the following functions (where $v \in V$ and $e \in E$ ):

- $att(v)$, $att : V \rightarrow 2^A$, where $A$ is the set of all concepts' attributes. Such a function returns the attributes of $v$;

Figure 5.7: Intensional Level Entities: Treeview entities filter

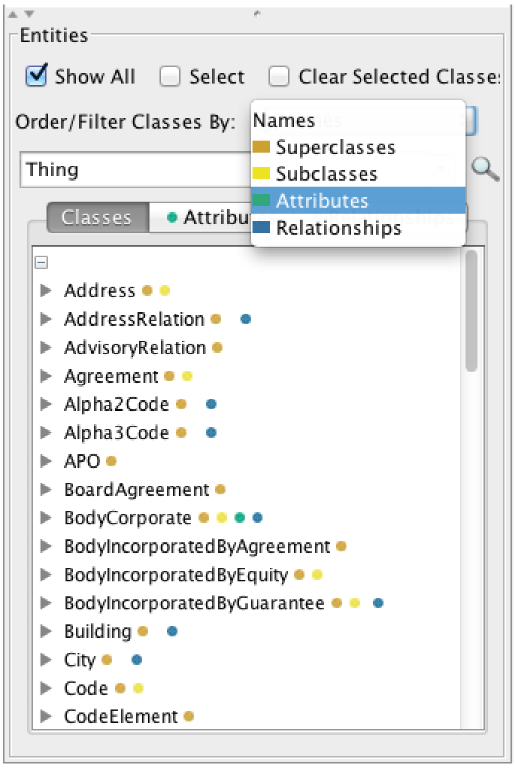
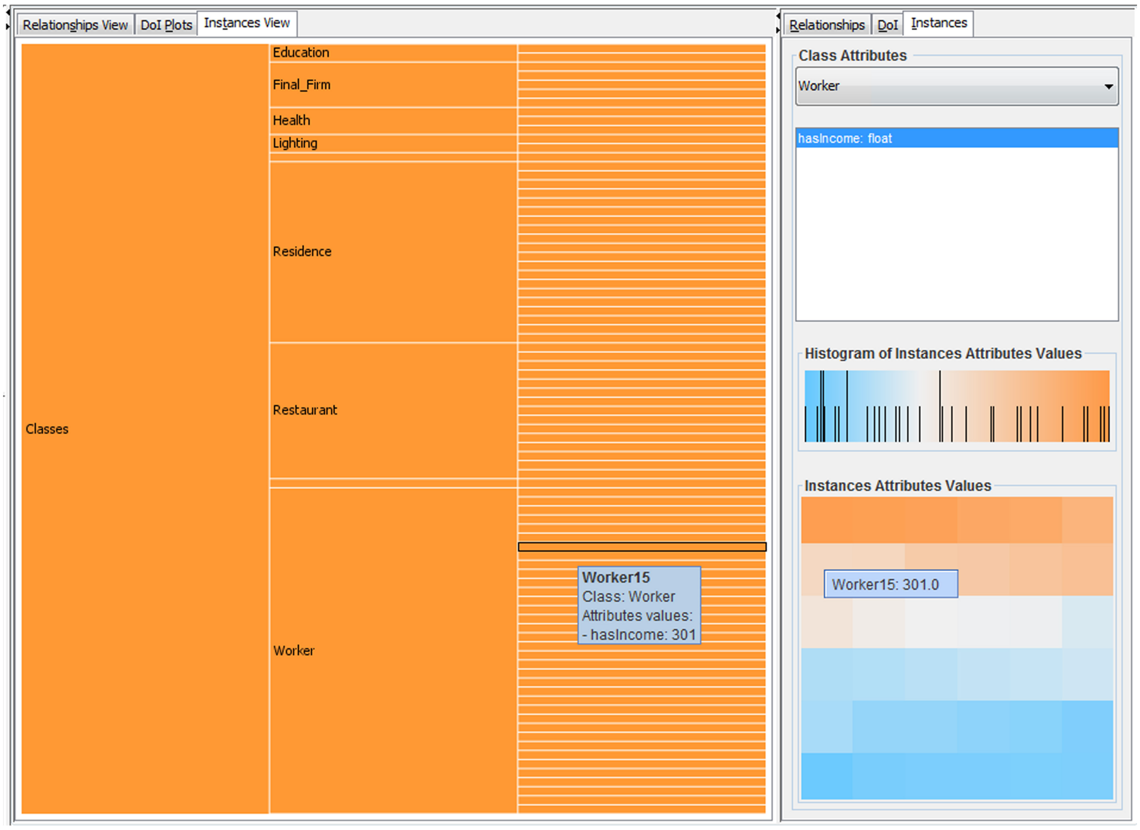



Figure 5.8: Instances of Classes: 2D Icicle Tree and 2D Pixel-Oriented Visualization




- $inst(v)$, $inst : V \to 2^{I_C}$ where $I_C$ is the set of all concepts' instances. Such a

Figure 5.9: Instances of a Selected Class

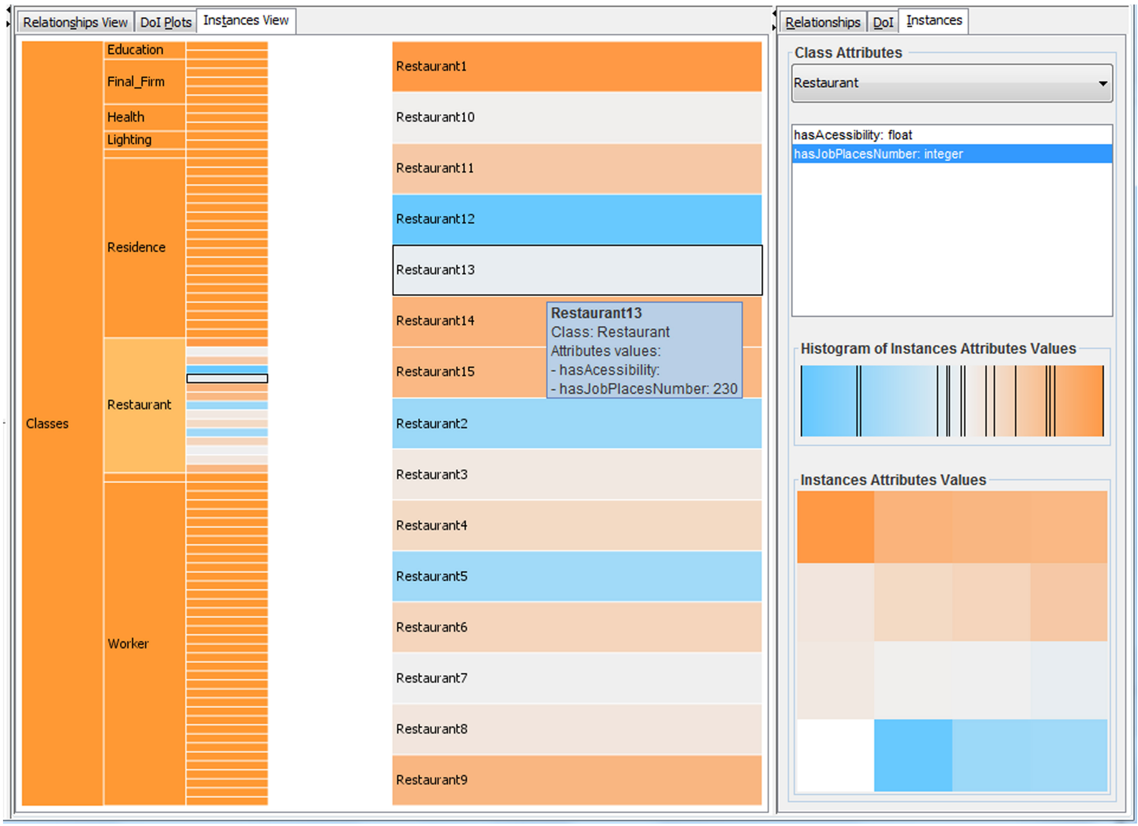
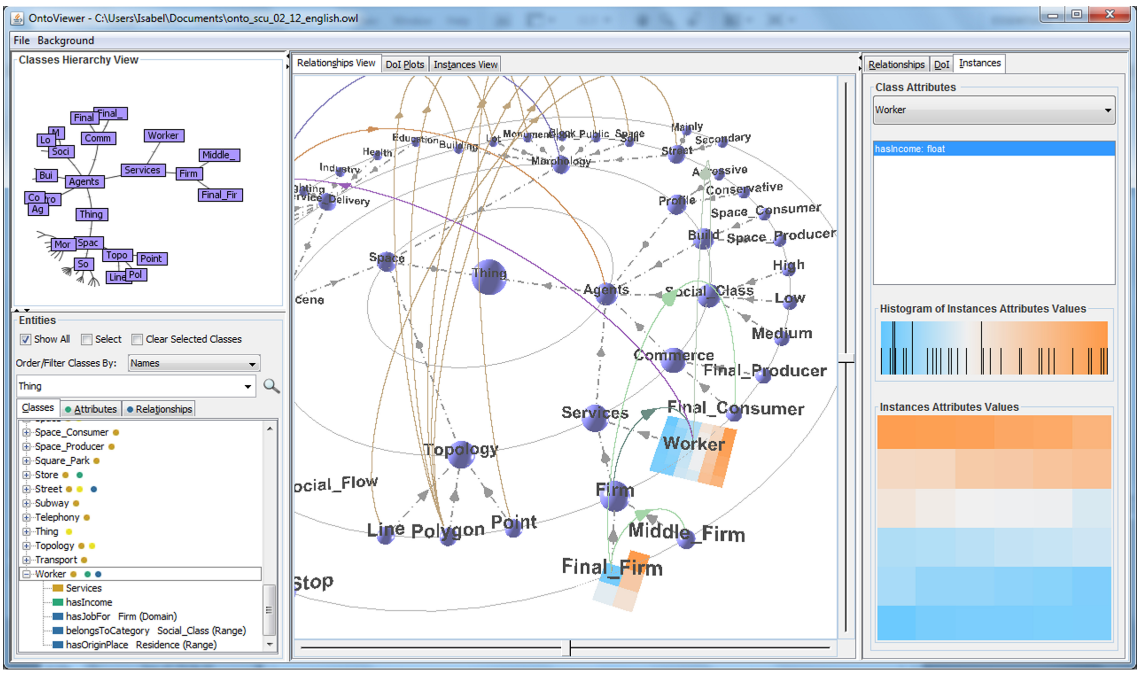



Figure 5.10: Intensional and Extensional Levels of Ontology




function returns the instances of the class $v$;

- $inst(e)$, $inst : E \rightarrow 2^{I_R}$ where $I_R$ is the set of all relationships' instances. Such a function returns the instances of the relationship $e$;

- $rel(v)$, $rel : V \to 2^E$. Such a function returns all edges in $E$ that involve $v$;
- $dep(v)$, $dep : V \to N^+$. Such a function returns the depth of $v$ in the ontology hierarchy.

The cardinalities $|att(v)|$, $|inst(v)|$, and $|rel(v)|$, $|inst(e)|$, and $\frac{1}{dep(v)}$ are linearly combined to compute the API of concepts and relationships. Using such APIs and the distance of a concept from the user selected *main concept* (MC) it is possible to compute the DoI of classes (Equation 5.1).

$$DoI = f(API, D) \tag{5.1}$$

The $DoI$ is used to automatically compute an ontology view containing the most relevant vertices and edges with respect to the MC. More precisely, to compute the DoI we follow four steps:

1. we assign an $API$ value to each vertex in the ontology independently of the intended focus, i.e., the main concept MC selected by the user. In particular the $API$ is computed using the Equation 5.2:

$$API(v) = c_1|att(v)| + c_2|rel(v)| + c_3\frac{1}{dep(v)} + c_4|inst(v)| \tag{5.2}$$

2. we assign an $API$ value to each edges $e \in E$ using the Equation 5.3:

$$API(e) = c_5\frac{|\{x| < x, y >\in inst(e)\}| + |\{y| < x, y >\in inst(e)\}|}{|inst(a)| + |inst(b)|} \tag{5.3}$$

where $a$ and $b$ are the vertices connected by $e$ and assuming $c_5 = 1$ it holds that $API(e) \in [0, 1]$. Roughly speaking, we can say that $API(e)$ corresponds to the percentage of instances of $a$ and $b$ that are involved in the relationship $e$. Moreover we label $e$ with $1 - API(e)$: such a label represents the *semantic distance* between $a$ and $b$: if most of the instances of $a$ and $b$ are involved in the relationship, the classes are very related each other, the $API(e)$ is very close to 1 and the label is very close to 0;

3. we calculate the distances $D(v, MC)$ between the MC and each concept in $V$ analyzing the different paths that exist between them (see example in Figure 5.11). In the most general case, we have $n_{oe}$ paths composed by $OE$ edges and $n_e$ paths composed by $E$. We label $oe \in OE$ edges with 1 and $e \in E$ edges with $1 - c_5API(e)$; the length of a path $l(p_i)$ is just the sum of its labels. In order to compute the overall distance we use a parallel resistor-like Equation 5.4 (the more the parallel paths the closer the two classes are):

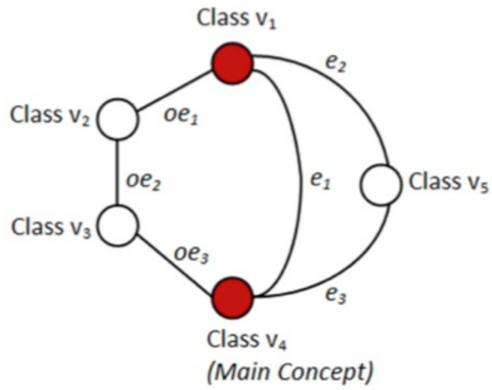
$$D(v, MC) = \frac{\prod_{i=1}^{n_{oe}+n_e} l(p_i)}{\sum_{i=1}^{n_{oe}+n_e} l(p_i)} \tag{5.4}$$

4. we normalize $D$ and API and we compute the DoI as showed in Equation 5.5:

$$DoI(v, MC) = API(v) - c_6D(v, MC) \tag{5.5}$$

and we normalize it.

Figure 5.11: Example of distance calculation between the class v1 and the class v4 (*main concept*)




Coefficients $c_1 \ldots c_6$ are set according to the user task: high $c_1$ and $c_2$ values are suitable when the user is interested in classes with high structural complexity (great number of attributes) and highly connected; high $c_3$ values are suitable when the user is looking for very abstract classes (close to the root); high $c_4$ and $c_5$ values allow for focusing on highly populated classes and relationships; and high $c_6$ values allow for exploring concepts that are far from the main concept. Initial values for these coefficients have been set during an informal user study involving expert ontology designers and undergraduate students; the actual version of the system allows changing such defaults and exploring the impact the changes have on API and DoIs (see Section 4).

Once DoI has been computed, it is sufficient to select a suitable threshold $k$ and show on the view only the vertices where $DoI(v, mc) \geq k$. That results in a subgraph $G'$ of the ontology induced by $mc$ and $k$, where $G' = (V', OE' \cup E')$ is a subgraph of $G = (V, OE \cup E)$ with $V' \subseteq V$, $OE' \subseteq V' \times V'$ and $E' \subseteq V' \times V'$.

The DoI tab (see right side of Figure 5.12) shows the API values of classes and relationships, and the DoI calculation parameters. These values can be changed through sliders that control the coefficients $c_1 \ldots c_6$. The interaction with the sliders generates new results that can be analyzed in the lists of this tab and in a 2D plot as shown in Figure 5.12. The slider DoI threshold allows filtering classes and relationships in the 2.5D view (Figure 5.13), in order to reduce the complexity of the visualization according to the user task.

## 5.3 Final Comments

In this chapter, we presented OntoViewer, a tool for visualization and analysis of ontologies at their intensional and extensional levels. We also briefly described the guidelines that were used for its development. The evaluation of this tool is presented in the next chapter and was performed according to three methods: case studies involving an expert, inspection using the design, and a remote usability evaluation.

Figure 5.12: Automatic Analysis (DoI): 2D Plot

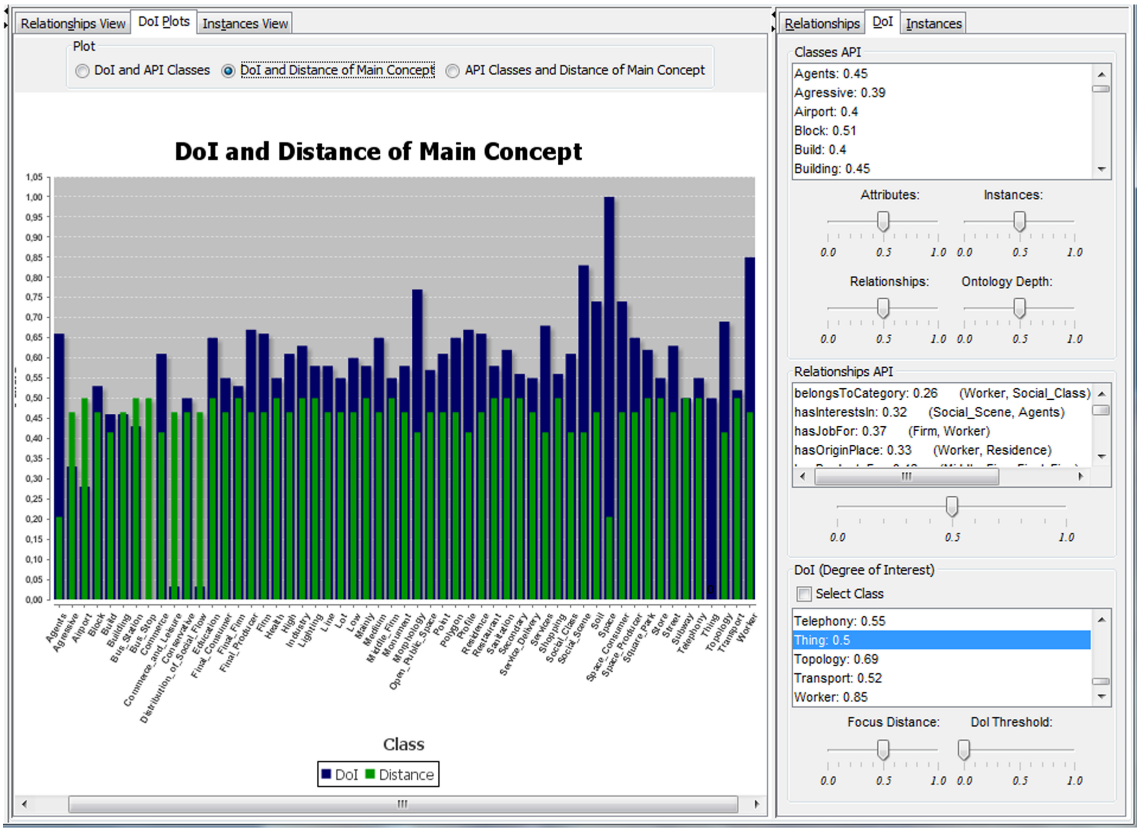
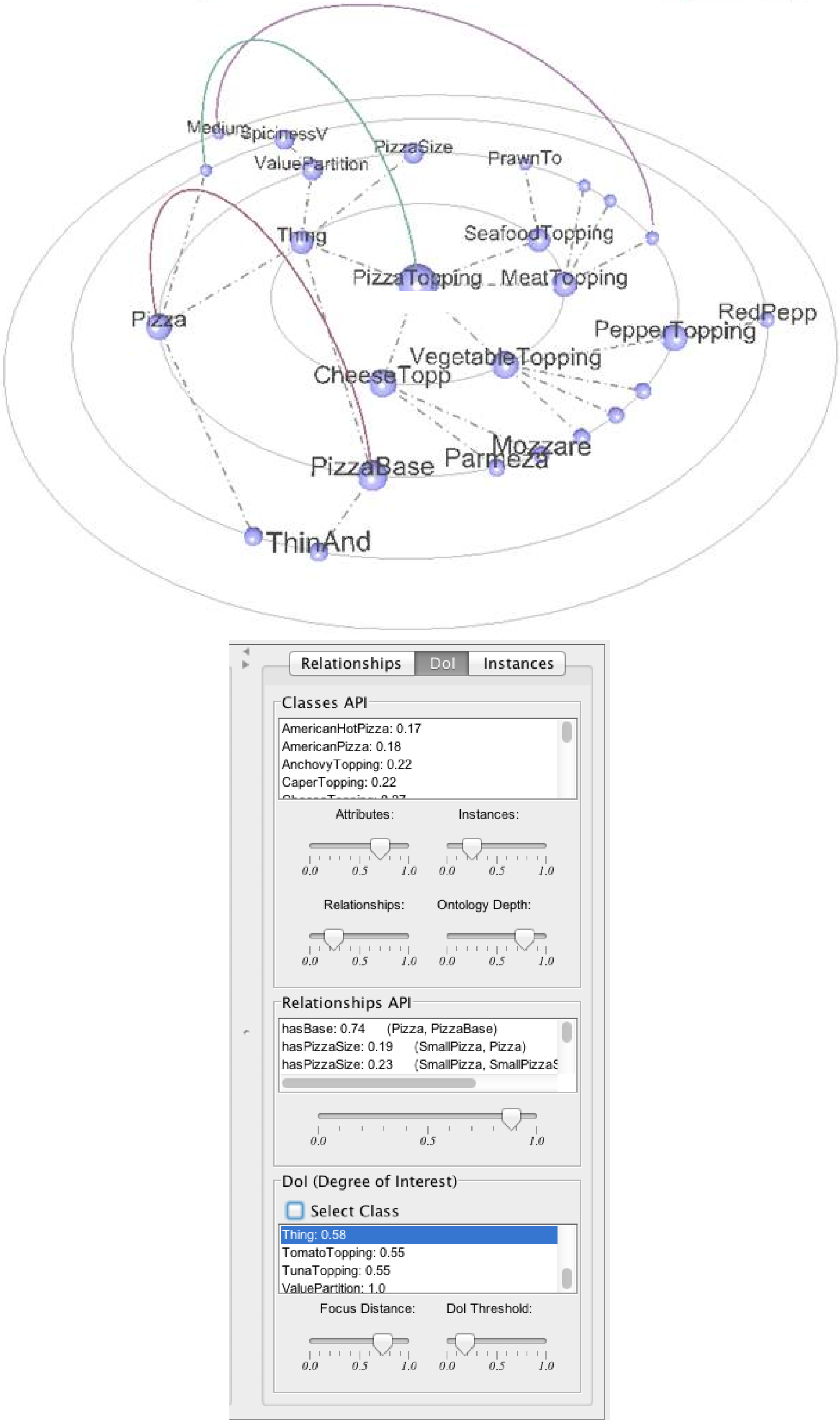


Figure 5.13: Results of DoI threshold calculation: some classes and their relationships was occluded of the 2.5D visualization by DoI filter



# 6 RESULTS AND DISCUSSION

*Although OntoViewer has been developed in a cycle of design-prototyping-evaluation as described in Chapter 4, in this chapter we present its use in different situations and report the results of a final evaluation round. We present an inspection using the guidelines we proposed in Section 5.1, and describe a remote usability evaluation with experts.*

## 6.1 Visualizing Ontologies with OntoViewer

Along the overall development, different ontologies were used with the successive prototypes. These ontologies are from specific domains and differ in terms of size and complexity. In this section, we describe six of these ontologies among all used. We give only a simple query example for each one, but for all the ontologies we have used, some specific queries were improved by the use of multiple views and the degree of interest filtering technique.

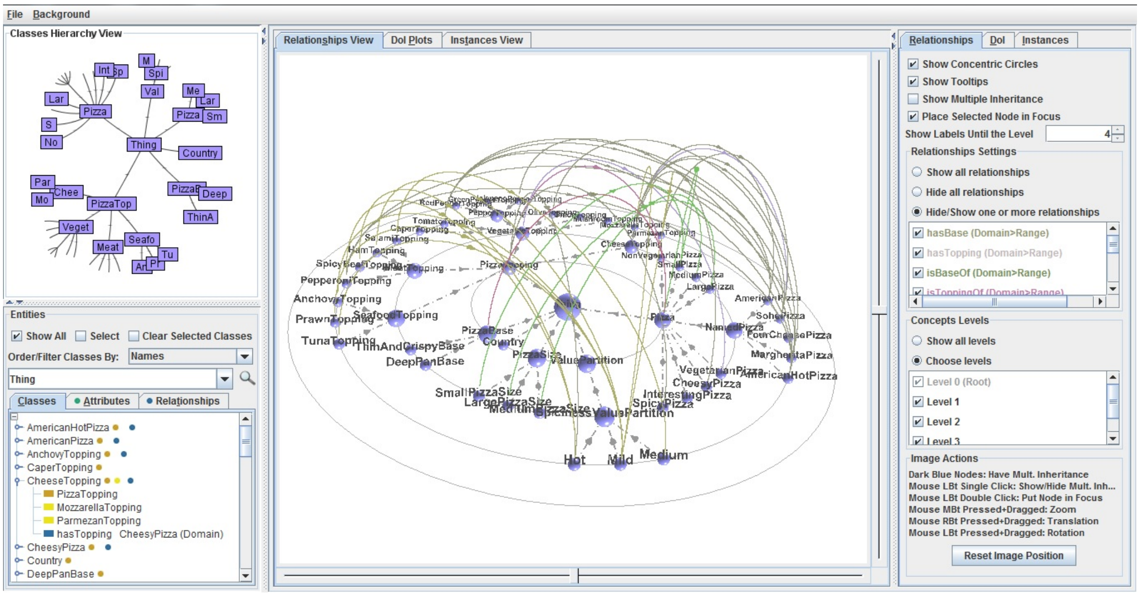
The pizza ontology (see Figure 6.1) contains a familiar set of concepts and is also used in the Protègè guide for building OWL ontologies (HORRIDGE, 2011). The queries over this ontology involve common concepts as, for example, *"Get all pizzas with a spinach topping"*, which is a query based on a relationship "hasTopping" and a class "SpinachTopping", or *"Get all pizzas from Brazil"*, considering a relationship "hasCountryOfOrigin" and some instance of the class "Country".

The urban ontology (see Figure 6.2) was developed to represent a model for urban spatial dynamics simulation, based on the concept of urban space and its successive transformations, and the agents related to such space (people, commerce, factories, etc.) (KRAFTA, 2004). Example of a query over this ontology: *"Get the higher income of some worker"*, based on the entities class "Worker", relationship "hasIncome", attribute "income (double)" and some instance of "Worker".

Apart from these two simple ontologies, we have tested other four ontologies: MSG (master course in management systems) (DIAS, 2012), FIBO (Financial Industry Business Ontology) (NEWMAN; BENNETT, 2012), MEF (Treasury of the Italian Ministry of Economy and Finance) (ANTONIOLI et al., 2013) and sedimentary stratigraphy (LORENZATTI et al., 2009).

The MSG ontology (see Figure 6.3) was proposed for representing the research lines of a Master Course in Management Systems: Total Quality, Environment, Social Responsibility and Work Safety. Examples of queries over this ontology are: *"What would be the set of keywords that represent the knowledge generated in the different lines of research at MSG?"*, *"What are the types of relationships presented in this set of keywords under an interdisciplinary perspective?"* and *"What would be appropriate to represent the knowledge produced in MSG instruments?"*.

Figure 6.1: Visualization of pizza ontology




The FIBO ontology (see Figure 6.4) is an industry initiative to define financial industry terms, definitions and synonyms using semantic web principles such as RDF/OWL, in the Canadian financial system. FIBO also aid in regulatory reporting by providing clear and unambiguous meaning of data from authoritative sources. According to the authors of this ontology, the primary practical use is descriptive of various kinds of financial instruments. Regulators and financial market participants have a common language to talk about things. FIBO integrates around 60 ontologies which complement one each other.

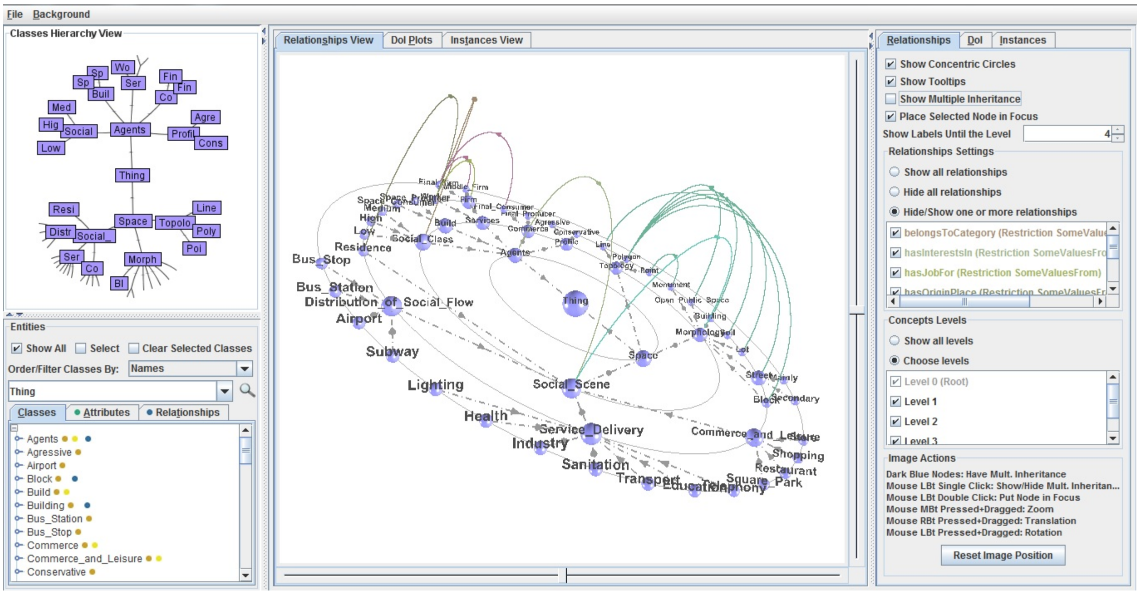
The MEF ontology (see Figure 6.5) represents a new paradigm for accessing and integrating financial data, whose key idea is to resort to a three-level architecture, constituted by the ontology, the data sources, and the mapping between the two. MEF ontology was designed considering a scenario related to: issuance and management of the public debt, liquidity management, management of the government securities amortization fund, analysis of the problems inherent to the management of the public debt at both national and international level and to the functioning of the financial markets, coordination and supervision of the access to the financial markets by public entities.

Finally, the sedimentary stratigraphy ontology (see Figure 6.6) proposes new primitives based on the combination of the conceptual and pictorial primitives to model Geology domain knowledge. The structure of this ontology offers support for the process of interpretation of depositional processes and this fact generates necessity for well-defined queries over the ontology. For example, over the class "Sedimentary Facies", the grain size can be used with the relationship "hasGrainSize" in order to help the experts to analize and describe a core and an outcrop.

For the stratigraphy ontology, we reproduce a case study in the creation and manipulation of this ontology with OntoViewer. The expert user is a knowledge engineer from the Intelligent Data Bases group at UFRGS. In this case study, he did not use the features of analysis and of visualization of instances, only the visualization of intensional level. But, he expressed some impressions about the visualization of extensional levels and automatic analysis over the ontology.

The objectives of the case study are:

Figure 6.2: Visualization of urban ontology




1. *Make a survey about how the ontology of sedimentary stratigraphy is used in practice by users with a knowledge engineer profile;*

2. *Perform an interview with a specialist in creation and manipulation of ontologies using OntoViewer;*

3. *Detail how the ontology visualization and the features available in the tool supports the task of developing an ontology.*

The fist step was related to the understanding of a common scenario of ontologies use. Then, three questions were posed to the expert:

*(1) How a knowledge engineer works with ontologies?*

*(2) What types of tasks are performed in the ontology engineering process?*

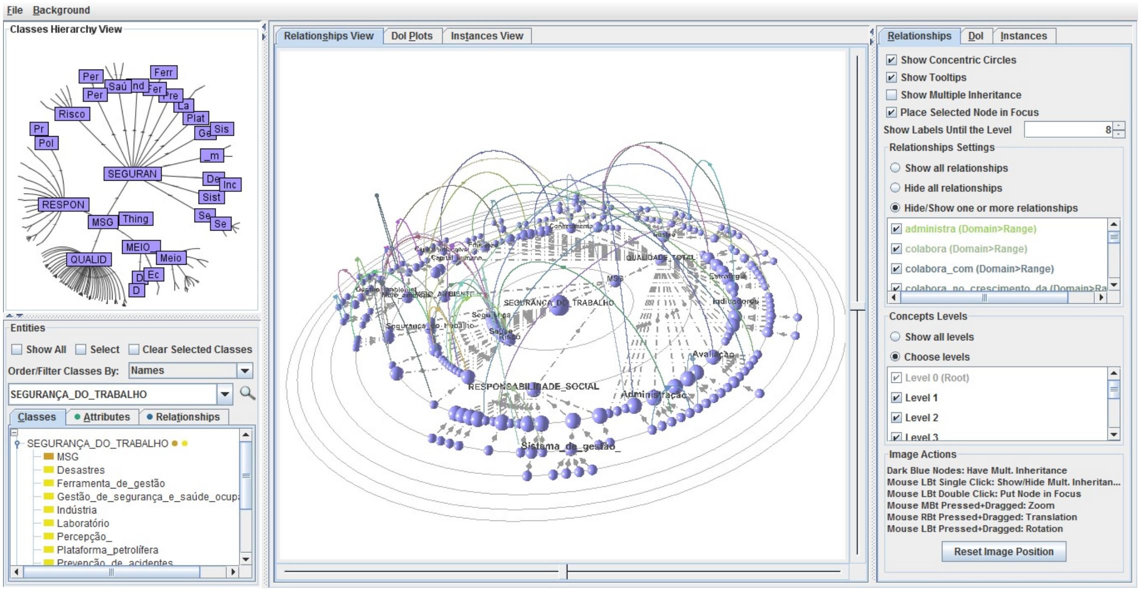
*(3) For the tasks listed in (2), a tool for visualization of ontologies would be helpful? Why?*

Regarding question (1), a knowledge engineer can work with ontologies in two different ways: in a process of ontology engineering (building and editing the ontology), and when the ontology is being incorporated in systems development; in this last way, the ontology becomes "a piece of software" and the user role is that of a software engineer.

For the question (2), seven tasks were enumerated: creation and removal of concepts (classes); change of the taxonomy; creation and removal of attributes; changing attributes range (data type); creation, modification and removal of relationships; editing properties of relations (transitivity, reflexivity, etc); alteration of taxonomy relations. These tasks need a tool that allows creating and editing ontologies apart of OntoViewer, but a visualization tool helps the process because improves the understudying the user has about the main aspects of the ontology structure.

Finally, in relation to question (3) , a tool for visualizing ontologies is helpful for four main tasks: to find concepts (classes) in the graph of ontology hierarchy; to check the attributes related to the concepts (classes); to check the relationships between concepts (classes); to have an overview of the ontology.

Figure 6.3: Visualization of MSG ontology




The second step consisted of a usage section. The expert was told to freely explore the functionalities, modifying the layout as he saw fit. After, he performed an analysis task based on the seven tasks previously identified (creation and removal of classes; change of the taxonomy; creation and removal of attributes; changing attributes range or data type; creation, modification and removal of relationships; editing properties of relations; alteration of taxonomy relations). For each task, the expert explained how the tool can assist it, detailing the steps and features used, pointing what is more usable and indicating future improvements. Furthermore, the expert indicated if the tool helped, helped partially or not helped in the task, explaining the reason for his response.

Three questions, related to this second step, were posed to the expert:

*(1) How OntoViewer was used in tasks execution? Has the tool helped?*

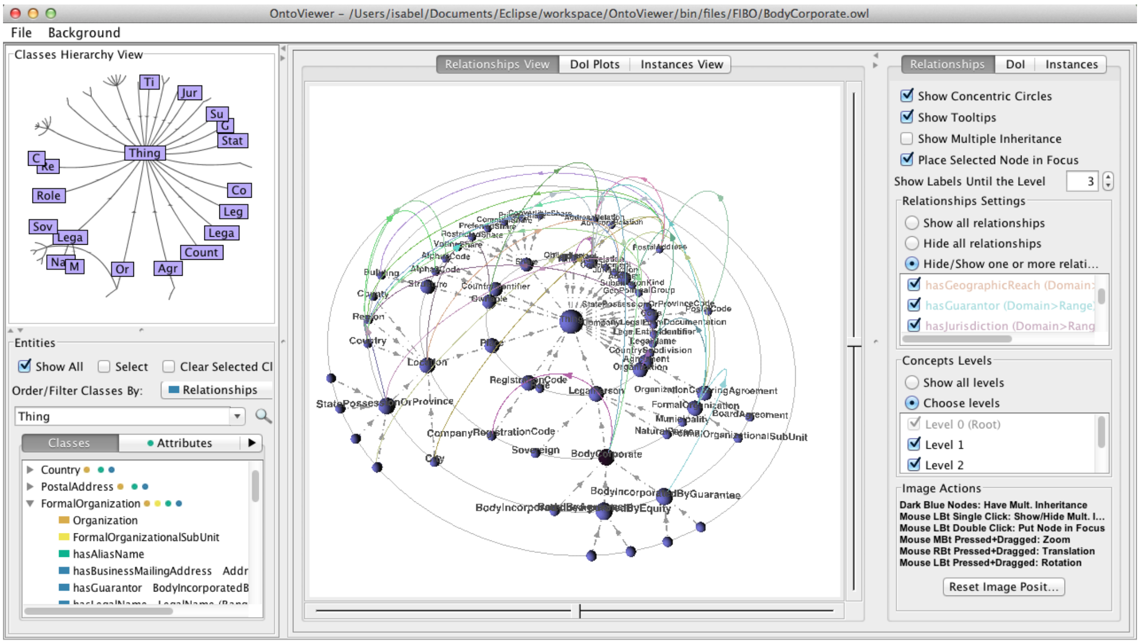
*(2) During the task execution, which features were used?*

*(3) Do you have comments or suggestions about the tool? If yes, what?*

For task 1 (*creation and removal of concepts (classes)*), it was simulated the need to add a new concept/class in a given taxonomy, for example, "SedimentaryStructure". For that, the expert searched for the class name on the entity view combo box and identified the desired level in the hierarchy of the ontology exhibited in the hyperbolic tree in order to insert the concept/class. After doing that, the expert commented that during the class search, it would be interesting to expand other classes of the taxonomy in treeview. He also commented that the possibility of putting the searched class in focus in all the views in a coordinated form helped the understanding of the ontology hierarchy as well as the task execution. We noticed that OntoViewer helped the task execution through the use of two functionalities: hierarchy visualization (hyperbolic tree) and entities visualization (treeview).

For task 2 (*change of taxonomy*), it was simulated the need of changing the taxonomy hierarchy of a concept/class as, for example, the integration of a branch of the taxonomy with other existing one (in "SedimentaryStructure", to integrate the chemical structures with branch biogenetic structures). For that, the expert also searched for the class name in the entity combo box and in the hyperbolic tree visualization. The expert commented that although the tool does not allow drag an entire branch of the taxonomy into another,

Figure 6.4: Visualization of FIBO ontology - *"BodyCorporate.owl"*




the fact that it possible to view the hierarchy helped in the identification of such necessary changes. As in task 1, OntoViewer helped the user in performing the task through the use of the two functionalities provided by hyperbolic tree and treeview visualizations.

The task 3 (*creation and removal of attributes*) consisted in creating the attribute "lithology" within the class "SedimentaryFacies". Thus, the expert searched for the class name in the entity combo box, and in both hyperbolic tree and radial tree visualizations to better understand where that attribute fits. The expert commented that it was useful that OntoViewer (a) enables filtering classes which have attributes and (b) has a tab to view them (tree view) combined with hyperbolic and radial tree views. But, for attributes removal it would be useful to allow the search for the attributes name.

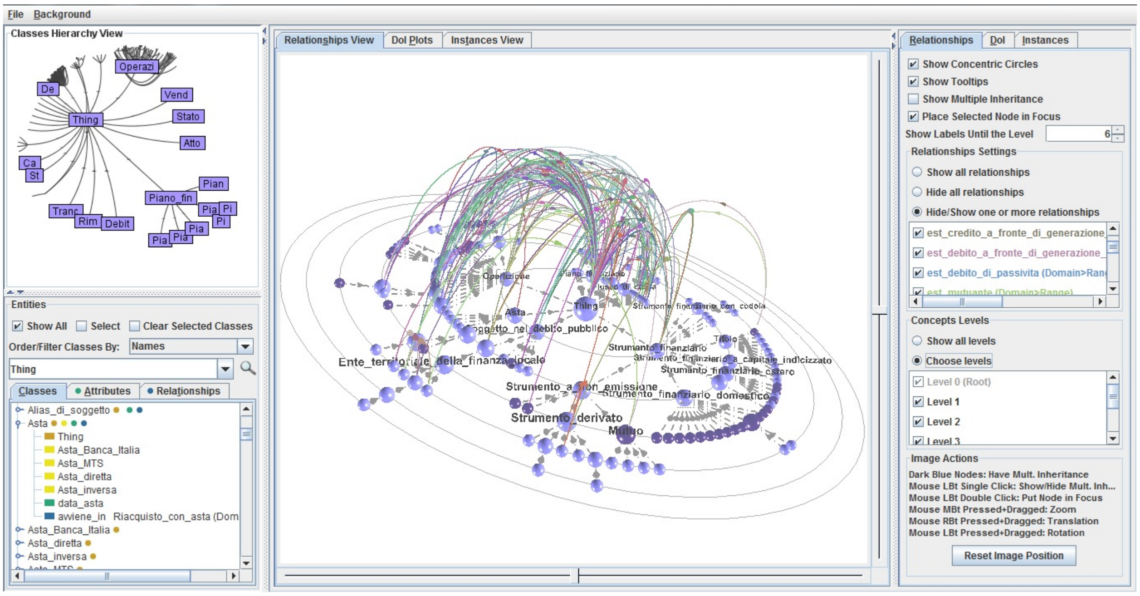
The task 4 (*changing attributes range*) became trivial once the task 3 is possible: the fact that the tool provides a tab showing attributes, and one can specify the type and class it belongs to, assists greatly in this task according to the expert. Then, the functionality provided by treeview is important.

Regarding task 5 (*creation, modification and removal of relationships*), for the process of relationships creation, it was assumed that the related classes were previously known. Then, the expert searched for the class name, selected the class and placed it in focus, and used the feature to display only the desired relationship. After that, for removing a relationship, the radial visualization helped the identification of the classes that share that relationship. The expert pointed out as advantageous the possibility of searching for the relationship by name and showing only it in the radial visualization. Treeview and radial visualizations helped in performing this task.

Task 6 (*editing properties of relations*) was performed for testing the editing of the transitivity or reflexivity properties. The expert suggested that the radial tree view could present nodes (or other visual objects) on the relationships arcs in order to indicate the relationship properties. As task 5, treeview and radial visualizations helped in performing this task.

Task 7 is related to *alteration of taxonomy relations* and the expert pointed out that it would be interesting to show the relationships hierarchy. This task was not completed.

Figure 6.5: Visualization of MEF ontology




Although the extensional level features were not surveyed in this case study, the expert expressed that the tool seems very interesting from the point of view of those who have to maintain knowledge bases, since it offers synthetic views of the distribution of class instances in a kind of "space of instances", which allows viewing how the attribute values are distributed. He also commented that it is useful to make quick visual queries about instances, noticing trends in values, for example.

In relation to the automatic analysis of ontology, according to the expert, the DOI also assists the engineer to get a sense of what will be the impact of the changes he intends to make in the ontology, for example, when it decides to adopt a design pattern rather than another. For this, the engineer would occasionally check the degree of interest of the classes that one wants to change and check, in advance, how much this modification will impact specific aspects of the ontology.

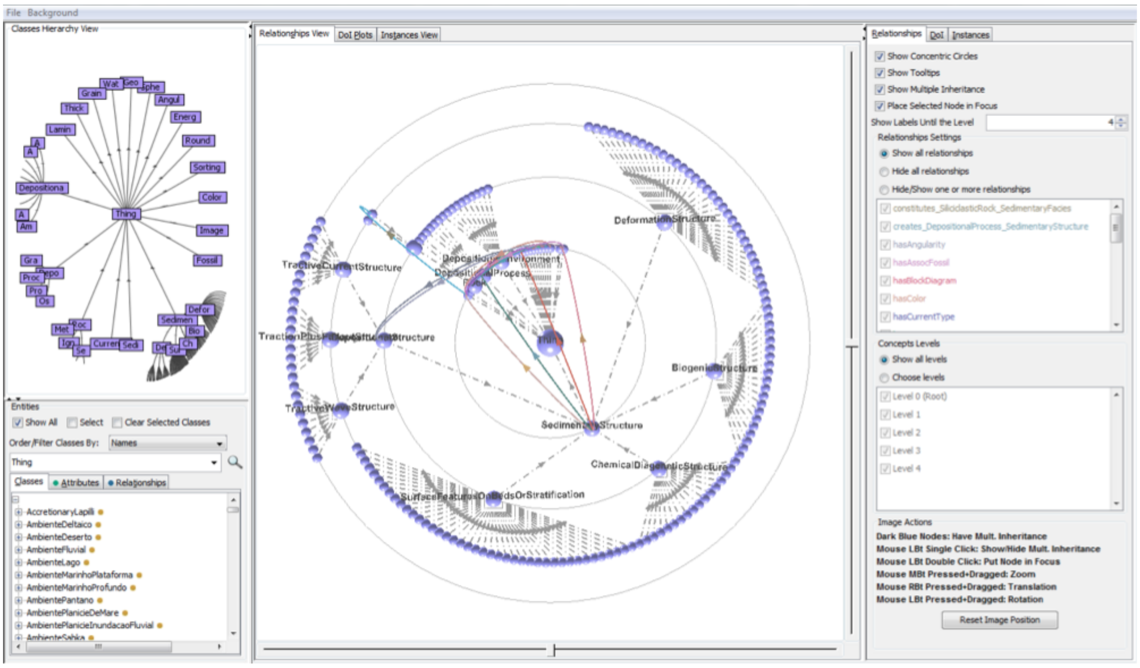
### 6.1.1 Inspecting OntoViewer based on Guidelines

Guidelines for ontologies visualization were elicited (see Section 5.1) and followed in the development of OntoViewer. We then used that set of guidelines to inspect the final version of OntoViewer before applying a broader evaluation method. We also assessed Ontoviewer based on the heuristics proposed by Forsell and Johansson (FORSELL; JOHANSSON, 2010), which also serves as guidelines.

The guideline *"The visualization tool must combine dimensions of quality, as classes, relationships, attributes, instances"* was considered in the general design of OntoViewer, as can be observed in Figures 5.1 and 5.8. We decided not to show all aspects of the ontology in the same view in order to avoid cognitive overload. Visualizations presented in Figures 5.2, 5.4, 5.6 and 5.7 are related to the intensional levels (classes, relationships and attributes) while those in Figures 5.8 and 5.9 show only the intensional aspects. Complementing these, the views in Figures 5.10 and 5.13 allow combining different dimensions qualities (intensional and extensional levels), helping the user to compare different states of the same ontology and a common knowledge base (ontology + instances).

As for the guideline *"The user must be able to focus, separately, both on the intensional and extensional level of the ontology, and to switch easily from one to the other"*,

Figure 6.6: Visualization of sedimentary stratigraphy ontology



all the views of OntoViewer, and their functionality, are easily accessible by manipulating a limited number of tabs and controls.

In relation to the guideline *"The views related to intensional and extensional levels should be synchronized and complementary"*, all the OntoViewer views follow this recommendation. If the user selects a class in a view, the complementary information about that class is displayed, in a synchronized manner, in another views, allowing the user to analyze different aspects of the intensional and extensional levels of the ontology.

Since the visualization techniques implemented in OntoViewer employ both focus+ context and overview+detail approaches, the guideline *"The user should be able to customize the interface in order to focus in only a subset of the information, reducing the cognitive overload"* is also followed. The user remains "aware" of the ontology hierarchy while focusing on one or more details.

Finally, related to the guideline *"Parallel to the graphical visualization of intensional and extensional levels of the ontology, textual information can complement the analysis of different aspects, especially when the ontology has a complex structure"*, we designed the tooltips in all views for giving extra information without overloading the user. The treeview (Figure 5.6) also helps in simplifying the visualization of data related to the intensional level.

Regarding the heuristics proposed by Forsell and Johansson (FORSELL; JOHANSSON, 2010), the inspection results are described below:

- *Information coding:* user perception is dependent on the mapping of data elements to visual objects and the use of hierarchical visualization techniques helps the exhibition of classes hierarchy and aspects of relationships, attributes and instances in a intuitive way according to the characteristics of the ontology structure. Besides that, pixel oriented views and 2D plots allow for the visualization of statistical data and/or visual analytics results.

- *Minimal actions, flexibility, orientation and help:* as more simple the interaction is, more intuitive to the user will be the actions towards the goals. So, we choose simple interaction with common actions for allowing levels of detail control (selecting, filtering and remapping), reset of actions and the presentation of additional information (tooltips, treeview) in order to assist the user. In OntoViewer, we have multiple and synchronized views, with complementary information, thus helping the user in accomplishing tasks.

- *Spatial organization, consistency, recognition rather than recall and prompting:* we have carefully designed the interface, by keeping the minimal amount of graphical items, with efficient use of space. Both intensional and extensional levels contexts keep interaction consistent, and the user does not need to memorize a lot of information to carry out his/her tasks, with the tool enabling different alternatives depending on the context.

- *Remove the extraneous and data set reduction:* due to filtering, only relevant information is displayed in the views of Ontoviewer helping the user to keep the focus during the task.

All these characteristics, obtained through the use of the guidelines and heuristics, were also highlighted by the experts that have participated in the remote usability evaluation we describe in the next section.

## 6.2 Remote Usability Evaluation

The usability testing was designed to prove the hypothesis that *"the multiple and coordinated views of the intensional and extensional levels of ontologies aid their creation, manipulation and visual analysis, helping the users to understand complex relationships among different features and aspects of the ontology"*.

This evaluation was developed considering two general evaluation criteria:

1. *Effectiveness:* Does OntoViewer allow users to have information on the different aspects of an ontology in order to do the right task, complete activities and achieve goals?

2. *Usability:* Is the interaction with the OntoViewer graphical interface simple and intuitive enough for the users?

In the following sections we describe participants and procedure, and discuss the results.

### 6.2.1 Participants and Scenario

Ten experts volunteered to this evaluation. They are between 26 and 59 years old, three women and seven men of different research fields as intelligent databases, ontologies representation and specification, technology information management and software engineering. All subjects graduated in Computer Science or related areas, and have a M.Sc. or Ph.D. degree or are M.Sc or Ph.D. students. Moreover, all have some previous knowledge of ontology visualization tools.

Although the participants were invited to try OntoViewer with their own ontologies, they received access to two ontologies of different domains and sizes: pizza and urban ontologies (described in the first section of this chapter).

### 6.2.2 Procedure

Participants were invited by e-mail, with an explanation of the evaluation purpose and procedure. Once they have agreed on taking part of the survey, they received a .zip file containing OntoViewer code ready for running, the evaluation form and the two test ontologies. Participants should freely use the tool either with the ontologies provided with OntoViewer evaluation package or with their own. At the end, they should answer the questionnaire composed by 15 statements along with a five point Likert scale of agreement ("Fully Agree", "Partially Agree", "Undecided", "Partially Disagree" and "Fully Disagree"). Users will answer them by marking their level of agreement to the statement. If the expert does not have experience related to the question, we offered the option "Not Used".

Statements 1 to 5 are based on the five proposed guidelines described in Section 5.1, while statements 6 to 15 are related to the heuristics proposed by Forsell and Johansson (FORSELL; JOHANSSON, 2010):

(1) *OntoViewer visualization tool helps the ontology exploration because it combines dimensions like classes, relationships, attributes, and instances.*

(2) *OntoViewer allows the user to focus, separately, both on the intensional and extensional level of the ontology and to switch easily from one to the other.*

(3) *Different views of OntoViewer related to intensional and extensional levels maximize the analysis of the ontology because they are synchronized and complement each other.*

(4) *In OntoViewer, the user is able to customize the interface in order to focus at only a subset of the information.*

(5) *Parallel to the graphical visualization of intensional and extensional levels of the ontology, textual information provided by OntoViewer (Entities window) can help the analysis of different aspects, especially when the ontology has a complex structure.*

(6) *OntoViewer implements information coding by mapping data elements to visual objects that enhances the perception of information.*

(7) *OntoViewer is based on minimal actions over ontology entities requiring few actions to accomplish a goal or a specific task.*

(8) *OntoViewer is flexible considering the means available for the user to customize the views depending on working strategies and task requirements involving ontologies.*

(9) *OntoViewer offers to the user ways of controlling the levels of details, redo/undo of actions and representation of additional information.*

(10) *The spatial organization in OntoViewer allows user to be oriented in the information space through the distribution of elements in the layout, precision and legibility, efficiency in space usage and distortion of visual elements.*

(11) *OntoViewer is consistent related to the way design choices are maintained in similar contexts, and are different when applied to different contexts.*

(12) *Regarding control functions in OntoViewer, the user does not need to memorize a lot of information to carry out tasks.*

(13) *The different views of OntoViewer help the user to know all the alternatives, when several actions are possible depending on the contexts.*

(14) *The visual elements of OntoViewer are simple and intuitive, and prevent the user from being distracted.*

(15) *In OntoViewer, the visualizations based on focus+context and overview+detail techniques reduce the ontology complexity, and increases the efficiency and ease of use.*

### 6.2.3 Results

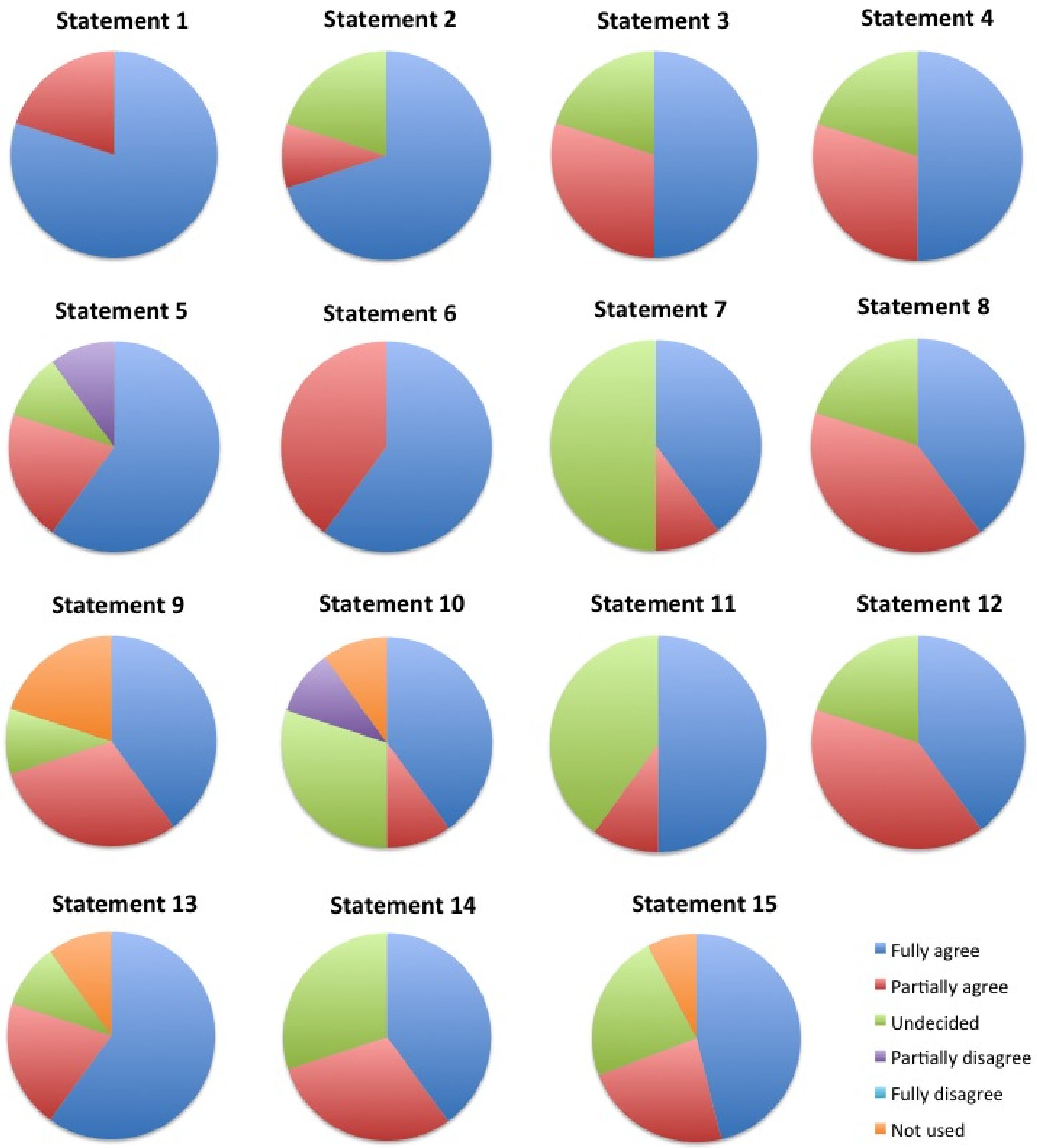
Figure 6.7 shows pie charts for the distribution of subject answers for each statement while Figure 6.8 is a bar chart where the categories "Fully Agree" and "Partially Agree" were aggregated in a single category, as well as "Fully Disagree" and "Partially Disagree". The mode and the interquartile range (IQR) results can be observed in Figures 6.9 and 6.10. As to the mode chart, answers were mapped to numbers, with 1 standing for "Fully Disagree", and 5 being "Fully Agree". For the IQR calculation, answers were ordered and the difference between the first quartile (25th percentile) and the third quartile (75th percentile) was calculated in order to obtain the middle 50 percent of the distribution unaffected by extreme values.

As can be seen in Figures 6.7, 6.8 and 6.9, the subjects agree ("Fully Agree" + "Partially Agree") with most statements (87%). Only two statements had 50% of "Agree" responses: statement 7 (*OntoViewer is based on minimal actions over ontology entities requiring few actions to accomplish a goal or a specific task.*), while the other 50% marked "Undecided", and statement 10 (*The spatial organization in OntoViewer allows user to be oriented in the information space through the distribution of elements in the layout, precision and legibility, efficiency in space usage and distortion of visual elements.* ), with the remaining 50% divided among "Undecided" (30%), "Partially Disagree" (10%) and " Not Used" (10%).

Moreover, in relation to statement 7, the mode was "Undecided" (50%) since the categories "Fully Agree" and "Partially Agree" are not grouped into a single category for the mode calculation.The option "Not Used" was answered for four statements: 9 (*OntoViewer offers to the user ways of controlling the levels of details, redo/undo of actions and representation of additional information.*), 20%; 10 (*The spatial organization in OntoViewer allows user to be oriented in the information space through the distribution of elements in the layout, precision and legibility, efficiency in space usage and distortion of visual elements.* ), 10%, 13 (*The different views of OntoViewer help the user to know all the alternatives, when several actions are possible depending on the contexts.*), 10%, and 15 (*In OntoViewer, the visualizations based on focus+context and overview+detail techniques reduce the ontology complexity, and increases the efficiency and ease of use.*), 10%.

There was a non-negligible number of subjects that marked "Undecided", but very few disagreed - only in statements 5 (*Parallel to the graphical visualization of intensional and extensional levels of the ontology, textual information provided by OntoViewer (Entities window) can help the analysis of different aspects, especially when the ontology has a complex structure.*) and 10 we observe 10% reported as "Partially Disagree" and none statement received "Fully Disagree" as answer. This is supported by the mode and IQR charts, which show that the mode was most often 5 ("Fully Agree" - 87%), rarely 4 ("Partially Agree" - 6,5%) and 3 ("Undecided" - 6,5%), never less. Variability was overall low, with the IQR being minor or equal to 1 in 66% of the statements, equal to 1,75 in statements 9 and 14, and higher than 1,75 in only three cases (2,0 for statements 7, 10 and 11).

Figure 6.7: Subject answers for each statement




For the statements 7, 10 and 11 (*OntoViewer is consistent related to the way design choices are maintained in similar contexts, and are different when applied to different contexts.*), where the variability was 2.0, the agree responses correspond to 50%, 50% and 60%, respectively, and the remainder is "Undecided". Probably, the subjects need to get more experience with the interaction provided by the multiples and coordinated views in order to take full advantage of these features.

Statements 1 and 6 (respectively, *OntoViewer visualization tool helps the ontology exploration because it combines dimensions like classes, relationships, attributes, and instances* and *OntoViewer implements information coding by mapping data elements to visual objects that enhances the perception of information*) have 100% of agree proving that the proposed visualization helps the ontology exploration since it maps data elements to visual objects that enhance the perception of information related to different entities of the ontologies.

Figure 6.8: Agreggated answers for each statement

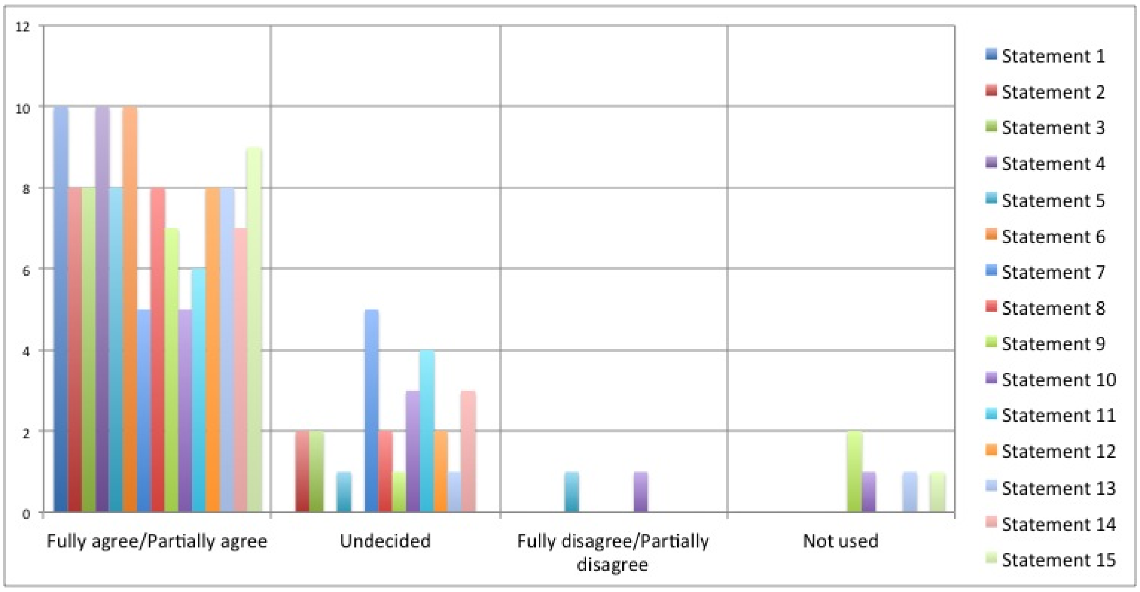
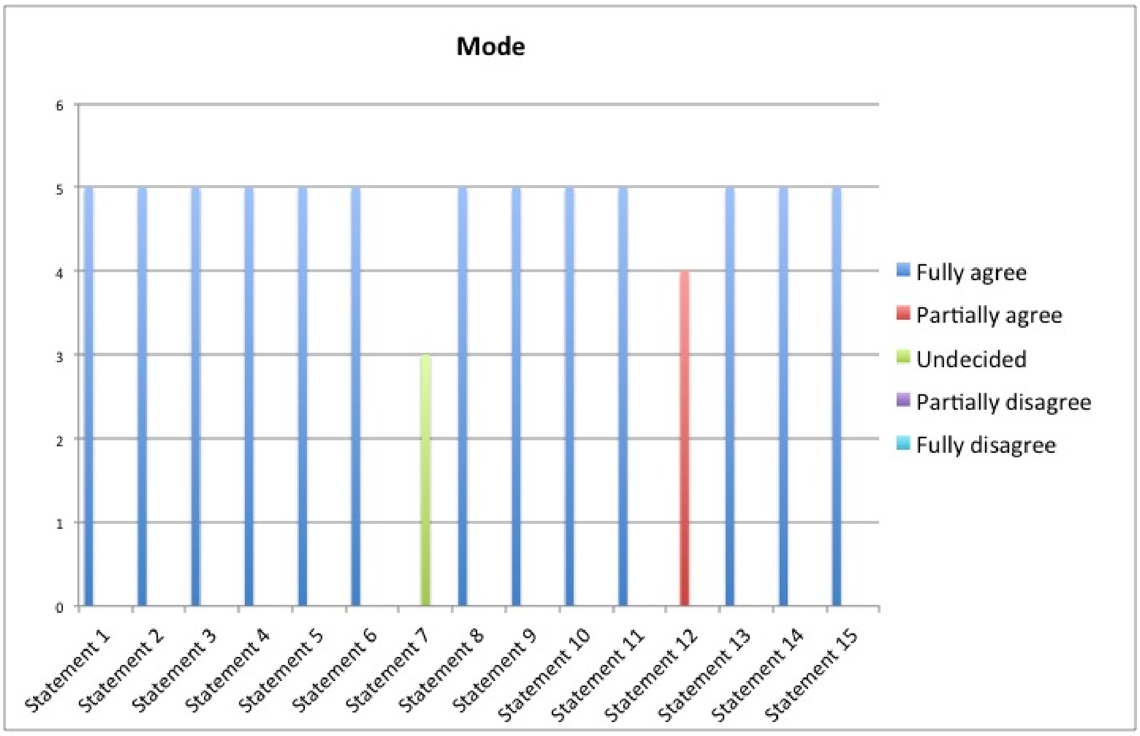
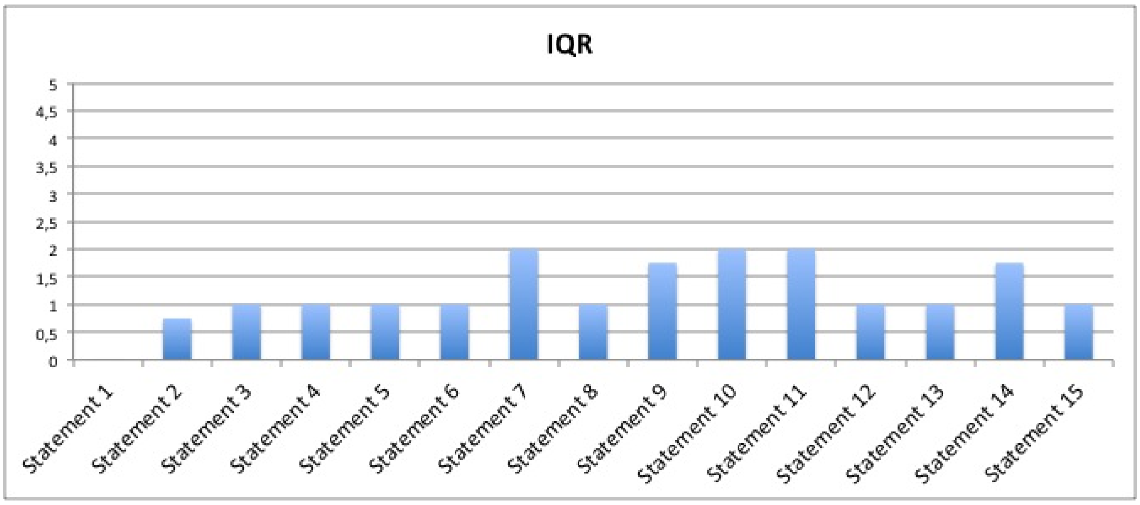



Figure 6.9: Mode for each statement




For the statements 2 (*OntoViewer allows the user to focus, separately, both on the intensional and extensional level of the ontology and to switch easily from one to the other*), 3 (*Different views of OntoViewer related to intensional and extensional levels maximize the analysis of the ontology because they are synchronized and complement each other*), 4 (*n OntoViewer, the user is able to customize the interface in order to focus at only a subset of the information*), 8 (*OntoViewer is flexible considering the means available for the user to customize the views depending on working strategies and task requirements involving ontologies.*), 12 (*Regarding control functions in OntoViewer, the*

Figure 6.10: Interquartile Range for each statement




*user does not need to memorize a lot of information to carry out tasks*) and 13 (*The different views of OntoViewer help the user to know all the alternatives, when several actions are possible depending on the contexts*), we have 80% of agree responses; the remaining 20% were marked as "Undecided" or "Not Used". Then, we can conclude that most subjects recognize that the multiple and coordinated views allow the user to focus, separately, both on the intensional and extensional levels of the ontology, but in a complementary way. This fact helps the analysis of the ontology depending on working strategies and task requirements in specific contexts.

Statement 5 (*Parallel to the graphical visualization of intensional and extensional levels of the ontology, textual information provided by OntoViewer (Entities window) can help the analysis of different aspects, especially when the ontology has a complex structure.*) also had 80% of agree responses, but the other 20% are divided between "Undecided" and "Partially Disagree". This fact confirms the adequacy of using textual information for complementing the graph visualization, although one subject considered confusing the way how the class hierarchy (superclasses and subclasses) is shown in the *Intensional Level Entities* window (see first image in Figure 5.6).

The other statement where we have "Partially Disagree" response is 10 (*The spatial organization in OntoViewer allows user to be oriented in the information space through the distribution of elements in the layout, precision and legibility, efficiency in space usage and distortion of visual elements.*). The remaining 90% are divided as 10% for "Not Used", 30% for "Undecided" and 50% for agree responses. The subjects that answered "Partially Disagree" reported they had difficulty in manipulating the visualization 2.5D because they don't have familiarity with visualization and interaction in 3D.

In relation to statements 9 (*OntoViewer offers to the user ways of controlling the levels of details, redo/undo of actions and representation of additional information.*), 14 (*The visual elements of OntoViewer are simple and intuitive, and prevent the user from being distracted.*) and 15 (*In OntoViewer, the visualizations based on focus+context and overview+detail techniques reduce the ontology complexity, and increases the efficiency and ease of use*), we have 70% of subjects that agree but in 9 and 14 the IQR variability was 1,75, 30% being divided between "Undecided" and "Not Used". These three statements are related to the fact that OntoViewer provides an overview of the ontology, so that users gain understanding of the entire scope and can intuitively focus on a specific part of the data of particular interest.

Regarding statements 6 and 14, one subject suggested the use of different colors for classes, aiming the clustering of these in groups and, thus, the possibility of visualizing only a specific class group. Another subject pointed that the visualization proposed in OntoViewer encourages the user in the quest for discovery about the knowledge represented in the ontology through the complete view of this in contrast to, for example, the simplified one-dimensional format of Protègè Class Browser and its current visualization plugins.

From these results, we can suggest that the users approved the multiple and coordinated visualization of ontologies, which effectiveness was our hypothesis. None of the subjects pointed out problems with the interface, so we can suggest that there is no severe usability problems in the current version of OntoViewer.

### 6.2.4 Discussion and Remarks

Based on the analysis of current solutions for ontology visualization (Chapter 3) associated to the requirements elicited in beginning of this study (Section 4.3), we developed OntoViewer, trying to cover the different aspects of visualizing the intensional and extensional levels of ontology.

The use of OntoViewer with different ontologies and the results from the evaluations presented in this chapter allows further discussion regarding the features of our visualization and analsys approach.

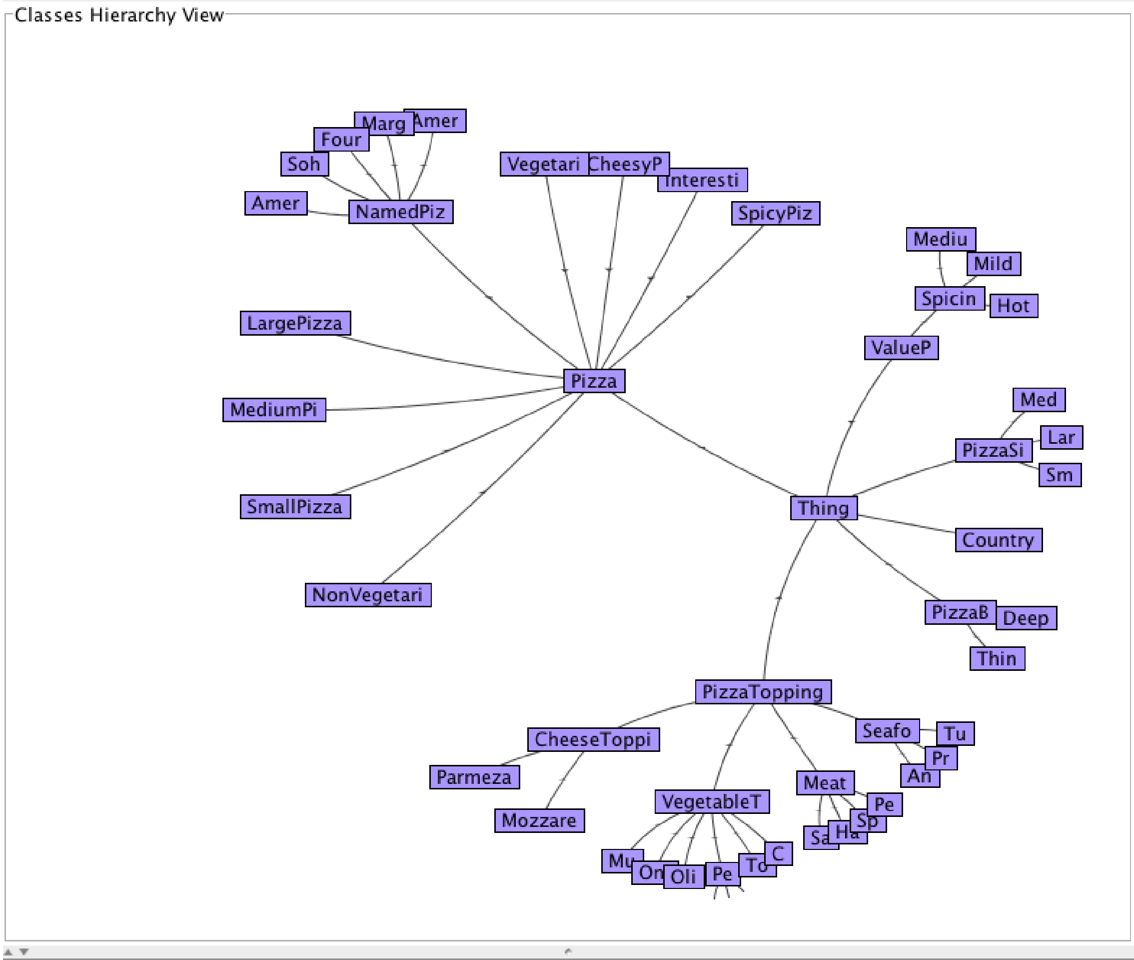
For the classes hierarchy (relationships *is-a*), we consider most appropriate the use of the hyperbolic tree, which represents a node-link style design with focus+context interaction technique. This is because the node on focus is usually the central one, and the rest of the nodes are presented around it, reduced in size until they reach a point that they are no longer visible, without using two separate views for the visualization of details of the classes hierarchy. In the usability evaluation, most of the subjects agreed with the appropriateness of this visualization in their comments after the test.

However, some users are not familiar with the hyperbolic tree visualization. Due to it does not have the usual top-down approach for the exhibition of the classes hierarchy, users can have difficulty in understanding the inheritance relations between classes. In the remote usability evaluation, one subject made this observation, and we optimize the exhibition of nodes offspring by inserting unobtrusive arrows in the edges as can be seen in the Figure 6.11. These arrows are dynamically exhibited when the user puts nodes into focus.

Besides this, another important aspect in the visualization of ontology hierarchy is the exhibition of multiple inheritance. In our approach, we draw edges between classes with multiple inheritance and the hyperbolic tree stay with graph characteristics. However, in the case of an ontology having several classes with multiple inheritance, problems as crossing edges occur (see Figure 6.12). Nevertheless, subjects prefer this proposal over the Protègè Class Browser (NOY; MCGUINNESS, 2001), which is based on replicating classes with multiple inheritance (Figure 5.3).

Mazza (MAZZA, 2009) states that 2D representations should be preferred over 3D representations because the latter ones increase cognitive overload or the user's mental effort to correctly interpret and interact with the visualization. 3D representations should only be used in limited and particular cases as, for example, when the information to be represented has a 3D spatial component. To avoid the problems of 3D but take advantage of its benefits, we propose a 2.5D radial tree for relationships exhibition, in order to

Figure 6.11: Hyperbolic tree with arrows in the edges indicating the selected node's offspring



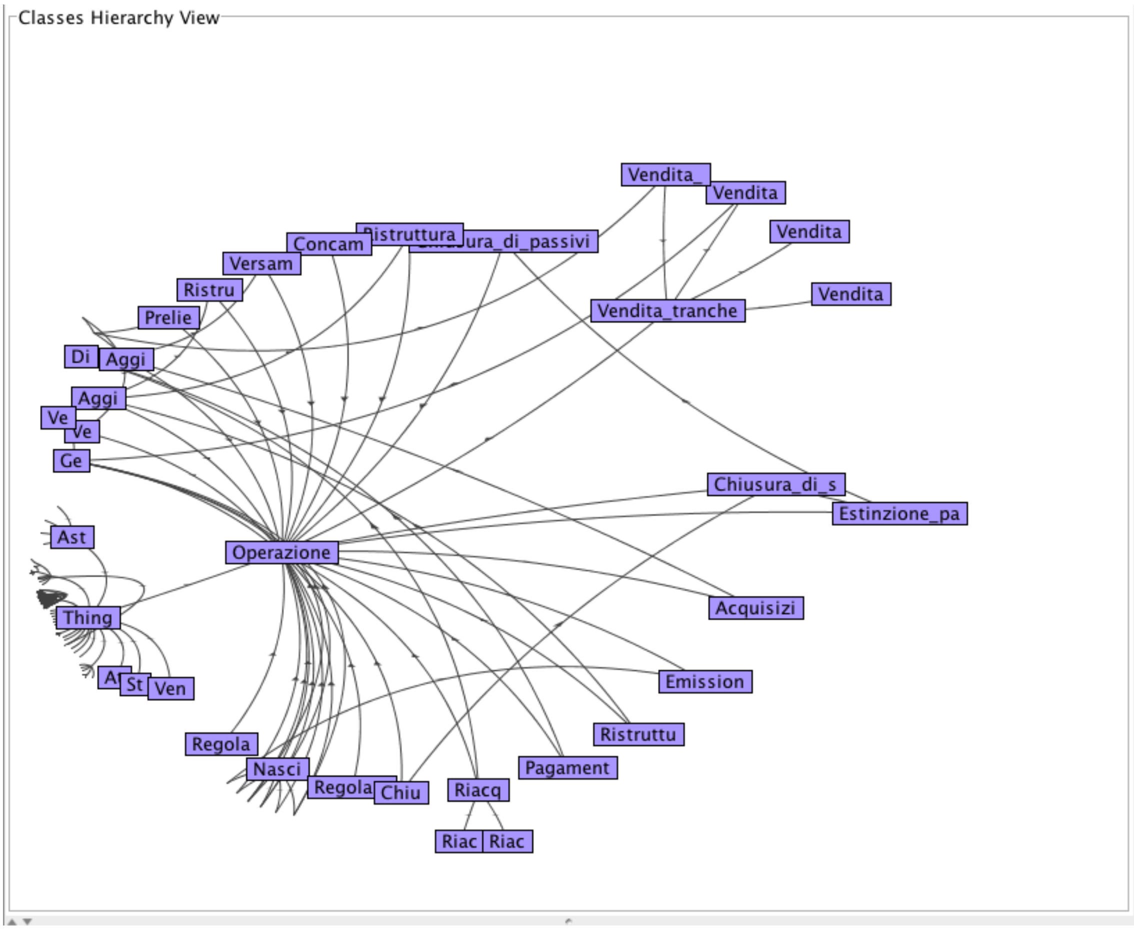

complement the information shown through the hyperbolic tree.

Katifori et. al (KATIFORI et al., 2007) assert that an ontology is something more than a hierarchy of classes; it is enriched with role relationships among concepts and each concept has various attributes related to it. Thus, considering the specificities of relationships, our 2.5D method enables the visualization of the relationships *is-a* in a plane while the other relationships are exhibited as curved lines in space avoiding the crossing of edges between these two types of relationships. According most subjects during the evaluation, this visualization allows for efficient analysis of the ontology intensional aspects besides offering easy interaction (zoom, pan, rotation, relationships filter, hierarchy filter, tooltips). Only one subject had difficulties with this interaction because he has problems with the depth perception.

Beyond the visualizations devoted to classes hierarchy and relationships, we also proposed a treeview for helping the user in the insight of different ontology's aspects combining visualization of the entities at the intensional level (superclasses, subclasses, relationships, attributes). Thus, the information presented in the treeview complements another views in a intuitive way. The subjects agreed with this proposal, because it facilitates the control of levels of details with additional information about the visualizations.

As for the extensional level, we explore the icicle tree and pixel oriented visualization techniques. The icicle tree is another focus+context technique, where the rectangles posi-

Figure 6.12: Hyperbolic tree showing several classes multiple inheritance




tions represent the classes and related instances. Classes are positioned on the right while their instances are implied by adjacent spatial relationship. A higher number of instances implies in smaller rectangles representing them. This technique allows the representation of the actual data in the data base. When a class of icicle tree is selected, another view is generated based on a pixel oriented technique in order to exhibit the values of instances attributes (see Figures 5.8 and 5.9). The subjects pointed out that this method is more effective for instances visualization than the common solution adopted by several tools based on showing instances as leaves as in Protègè TGVizTab (ALANI, 2003), Protègè OntoGraf (FALCONER, 2010) and On-TIME (CATARCI et al., 2010) and Glow (HOP et al., 2012) as discussed in Chapter 3.

Lastly, we add to the visualization of intensional and extensional levels of ontologies the feature of automatic analyses of aspects such as number of attributes, relationships, instances and depth of a class in the ontology hierarchy. The results are shown in two dynamic views. During the evaluation, mainly the subjects that work with intelligent databases expressed interest in the use of this novel model of automatic analyses provided by OntoViewer.

# 7 CONCLUSIONS

*This chapter contains final comments as well as perspectives on future work for improving the features of OntoViewer as an ontology visualization tool.*

## 7.1 Overview

Visualization techniques are important for ontology representation at the intensional and extensional levels in order to allow the comprehension of aspects of specific domains. Due to the complexity and size of ontologies such techniques need to be efficient in showing all the structural elements in an intuitive visualization.

We have designed a visual and interactive way to explore an ontology, supporting the understanding of such data by using multiple coordinated views and automatic analysis. Our visualization method combines aspects of both 2D and 3D interactive visualization techniques based on hierarchical views and focus+context concepts. For the data analysis, we calculate the degree of interest (DoI) and show the results in two dynamic views to help in the understanding of the analyzed data according to users' tasks.

In order to develop the visualisation approach, we performed an iterative cycle of refining designs and getting user feedback as proposed by Freitas et al. (2012), Forsell and Johansson (2010), Isenberg et al. (2008) and Munzner (2006) (see Section 2.2.3). Our study started with the analysis of existing ontology visualization solutions (FREITAS; PIMENTA; SCAPIN, 2012). After, we performed the evaluation of these existing ontology visualization tools (ISENBERG et al., 2008), we propose our first method based on a 2.5D visualization of the hierarchy of ontology classes classes and their relationships (SILVA; FREITAS, 2011a).

Following the guidelines of Freitas et al. (FREITAS; PIMENTA; SCAPIN, 2012), we interviewed experts on ontology development to know who the users are, and to understand the tasks they perform, in order to decide which users tasks to support. We elicited requirements for ontology visualization and collected impressions related to our initial proposition from the same users. We then took these requirements and mapped them into a visualization (MUNZNER, 2006). The visualization was extended to multiple coordinated views in order to help users to understand different aspects of data sets particularly when coupling two or more views showing different parts of the ontology. This approach was implemented as a tool called OntoViewer.

OntoViewer was firstly evaluated by six volunteers: the four experts interviewed in the initial phase of our study and two others specially invited for the evaluation. The results indicated that the use of 2D and 2.5D visualizations would be a solution to common problems presented by 2D and 3D ontology visualization tools, mainly cognitive overload and user disorientation.

From such qualitative feedback based on the accumulated experience of the experts and researchers, we have proposed a novel set of guidelines focusing on both intensional and extensional levels of the ontology to drive the design and evaluation of visualization tools. At the end, we improved the design of OntoViewer with different visualizations considering concepts related to both intensional and extensional levels of ontologies, which resulted in a more robust solution than those analyzed in Chapter 3. We also formalized a visual analytics method, aiming at amplifying the understanding of the data according to different user tasks. This OntoViewer version was submitted to users' evaluation one more time.

Based on the proposed guidelines and on a set of heuristics for information visualization systems (FORSELL; JOHANSSON, 2010), we performed an inspection of OntoViewer, simulating different analysis tasks on the ontologies with our visualization tool. Besides that, we have performed case study involving one ontology expert. All features recommended by the guidelines and heuristics were confirmed by the expert, that agreed with the fact that the visualizations implemented in OntoViewer combine different quality dimensions, which assist the users to get a sense of the impact of the changes they would intend to make in the ontology.

After these evaluations, we performed a remote usability test with ten experts in ontology creation and manipulation. A five-point Likert scale questionnaire (based on our guidelines and on the heuristics by Forsell and Johansson (FORSELL; JOHANSSON, 2010)) was proposed to these experts after they freely used OntoViewer in exploring a set of ontologies. From this usability evaluation, we obtained good results in agreement rate regarding the power of OntoViewer as an ontology visual analytics, interactive tool.

In this study we discuss different aspects related to the employment of multiple and coordinated views as a solution to common problems presented by ontology visualization tools as graphs scalability, cognitive overload and user disorientation. We also confirm that InfoVis aspects combined with visual analytics techniques can help the understanding of the analyzed data according during different tasks related to the ontology.

Finally, we conclude that ontology visualizations should present different interaction possibilities once browsing is not enough for actions related to locating a specific class or instance, especially for complex ontologies. Most experts tend to prefer visualizations that offer the possibility of an orderly and clear interaction of the presented information, even if in some cases it requires focusing on a specific entity or feature of the ontology. This fact implies that visualizations should also be robust, in order to guide and support the ontology exploration for different user profiles.

## 7.2 Publications

Along the work, we have produced the following publications:

- SILVA, I. C. S. ; FREITAS, C. M. D. S. ; NETTO, V.M. . Ontologia para Sistemas Configuracionais Urbanos. In: II Seminario em Ontologia no Brasil, 2009, Rio de Janeiro.

- SILVA, I. C. S.; NETTO, V. M.; FREITAS, C. M. D. S. Novos caminhos para simulacao urbana: integrando metodos de visualizacao de informacoes e modelagem de agentes e redes espaciais. In: XIII CONGRESO IBEROAMERICANO DE GRAFICA DIGITAL-SIGRADI, Sao Paulo, Brazil, 2009. p.153-155.

- SILVA, I. C. S.; FREITAS, C. M. D. S. Requirements for Interactive Ontology Visualization - Using Hypertree +2.5D Visualization for Exploring Relationships between Concepts. In: IVAPP - International Conference on Information Visualization - Theory and Applications, Vilamoura, Algarve, Portugal, 2011. p.242-248.
- SILVA, I. C. S.; FREITAS, C. M. D. S. Using Visualization for Exploring Relationships between Concepts in Ontologies. In: IV 2011 - 15th International Conference on Information Visualization, Londres, England, 2011. IEEE Computer Society, p.317-322.
- SILVA, I. C. S.; FREITAS, C. M. D. S. Using Multiple Views for Visual Exploration of Ontologies. In: IV Seminar on Ontology Research in Brazil (ONTOBRAS) and VI International Workshop on Metamodels, Ontologies and Semantica Technologies (MOST), Gramado, RS, Brazil, 2011. CEUR-WS. p.25-36.
- SILVA, I. C. S. ; FREITAS, C. M. D. S. . Hierarchical Visualizations and Ontologies. In: XXIV SIBGRAPI Conference on Graphics, Patterns and Images: Workshop of Theses and Dissertations (WTD), 2011, Maceio, AL, Brazil, 2011. p. 31-36.
- SILVA, I.; SANTUCCI, G.; FREITAS, C. Ontology Visualization: one size does not fit all. In: EUROVA 2012: International Workshop on Visual Analytics - Workshop of EUROVIS 2012, Vienna, Austria, 2012. Eurographics Association, p.91-95.
- SILVA, I.; FREITAS, C; SANTUCCI, G. An Integrated Approach for Evaluating the Visualization of Intensional and Extensional Levels of Ontologies. In: BELIV 2012 - Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization - Workshop of VisWeek 2012, Seattle, WA, USA, 2012. ACM DL, 2012.
- SILVA, I.; FREITAS, C; SANTUCCI, G. Improving the ontologies visualization insight using multiple and coordinated views and automatic analysis. Submitted to Expert Systems With Applications, 2014.

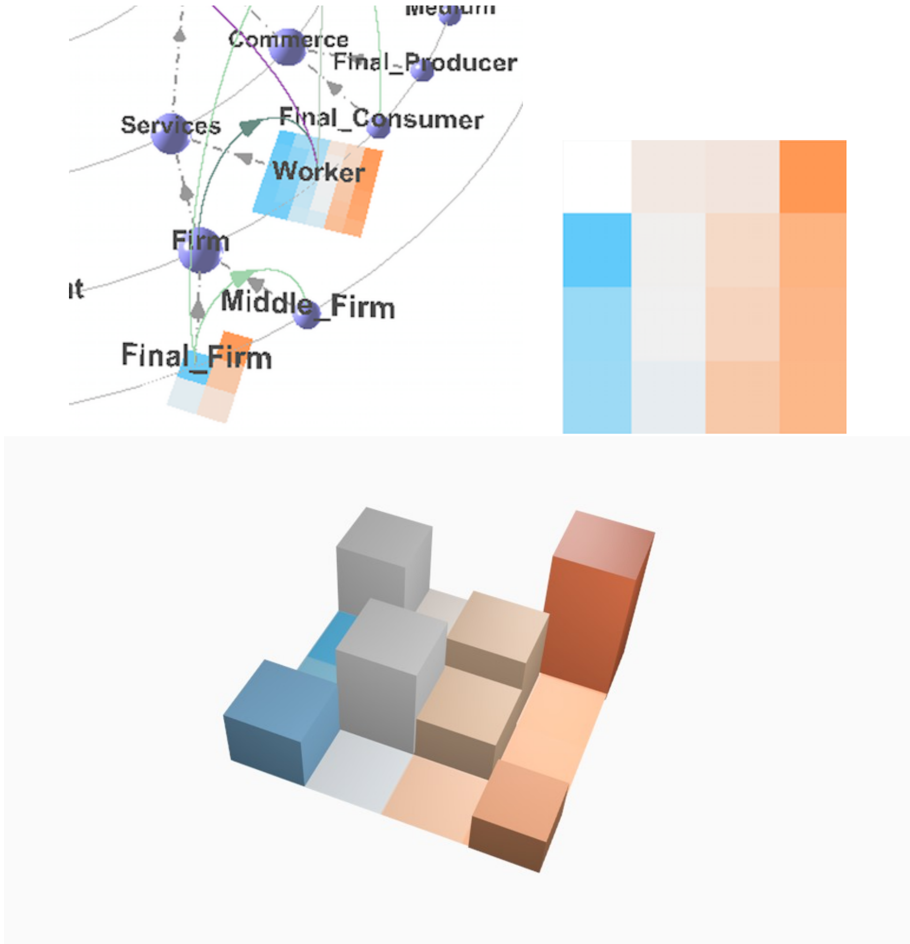
## 7.3 Future Work

In this study, our objectives were achieved with the investigation of information visualization techniques to aid the creation, manipulation and visual analysis of ontologies, both at their intensional and extensional levels.

However, new features can be added to OntoViewer to integrate other resources in order to improve the visual analytics power. One possibility is implementing the analysis of class instances, showing those that share relationships through a mapping of values to graphical and textual representations. This information is relevant because not always an instance will share the relationships of its defining class with another instances. For example, considering an ontology that has two classes - "worker" and "company" - which share the relationship "worksIn", if an instance of the class "worker" is unemployed, there is no relationship "worksIn" between this and any instance of the class "company". Then, not all instances of a given class will share all relationships with instances of another (related) class, and this is an information that can be graphically encoded in the visualization of intensional and extensional levels.

A future improvement for Ontoviewer is to employ the third spatial dimension to explore the graphical structures already implemented for instances visualization in the 2.5D visualization (Figure 6.7). For example, Figure 7.1 simulates an extrusion of the pixel oriented visualization. In this example, the size of the blocks encodes the number of relationships shared with other instances.

Figure 7.1: Extrusion of pixel oriented representations for mapping the number of relationships shared by an instance




This solution avoids the crossing of several edges which would represent the relationships of instances in one class. This problem happens in Jambalaya (STOREY, 2001) (Figure 3.4) and Knoocks (KRIGLSTEIN; WALLNER, 2011) (Figure 3.12). Even in relation to crossing edges, we need to improve the exhibition of a large number of multiple inheritance in OntoViewer because it is likely to generate perception problems and cognitive overload for the user (see Figures 7.2 and 7.3).

Exploring our proposition of multiple and coordinated views, it is possible to extend the visualization for ontology alignment as proposed by Lanzenberger and Sampson (LANZENBERGER; SAMPSON, 2006).

Apart from this, we can optimize the 2.5D visualization (which implies the drawing of several geometric objects) through the use of some graphics API more robust than Java 3D. During the development of this work, we perceived some disadvantages of Java 3D as slow performance, high rate of memory consuming and the need of a Java runtime and operation systems specific binaries in order for it to run. This fact makes it harder to develop interactive visualizations of large ontologies.

Figure 7.2: Multiple inheritance of a complex ontology shown on top of the hyperbolic tree

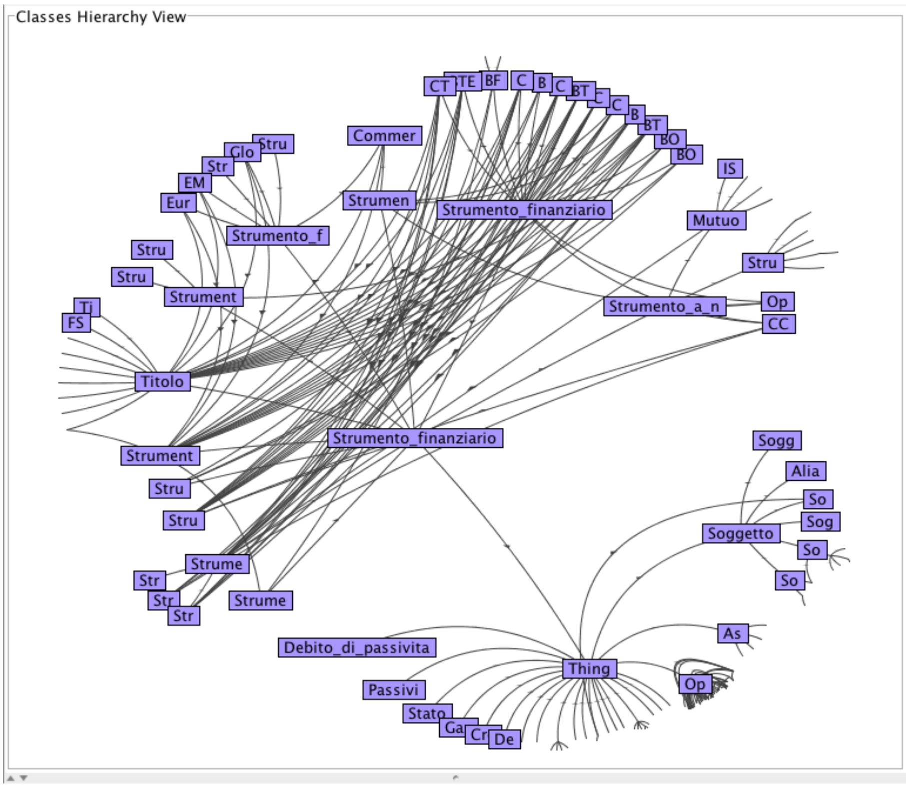
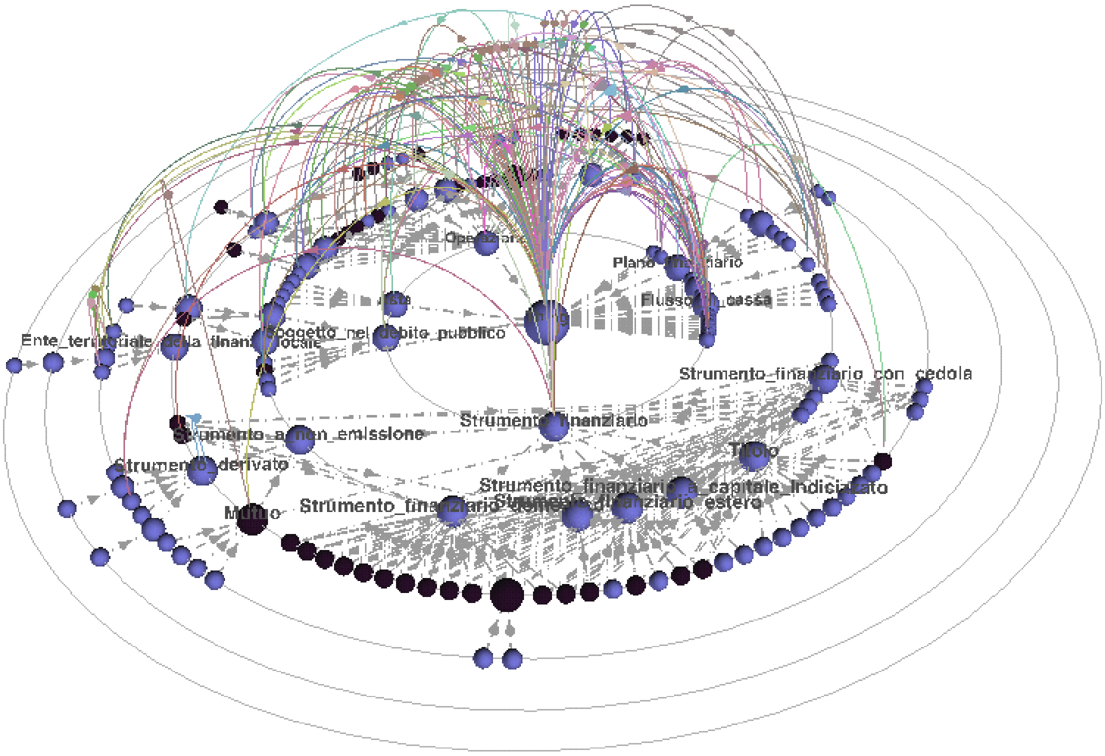



Figure 7.3: Multiple inheritance of a complex ontology shown inn top of the 2.5D radial tree



With a new version of OntoViewer it will be possible to migrate it to a web platform as suggested by some of our experts. Web applications present less conflicts and do not depend on the environmental settings, thus being compatible with most operating systems. However, a desktop version of OntoViewer is relevant in order to integrate it to Protègè, the common tool for creation and editing of ontologies.

# REFERENCES


ABELLO, J. et al. A Modular Degree-of-Interest Specification for the Visual Analysis of Large Dynamic Networks. **IEEE Transactions on Visualization and Computer Graphics**, Los Alamitos, CA, USA, v.99, n.PrePrints, p.1, 2013.

ALANI, H. TGVizTab: an ontology visualisation extension for protégé. In: KNOWLEDGE CAPTURE (K-CAP'03), WORKSHOP ON VISUALIZATION INFORMATION IN KNOWLEDGE ENGINEERING, Sanibel Island, Florida, USA. **Proceedings...** [S.l.: s.n.], 2003. p.23–26.

ANTONIOLI, N. et al. Ontology-Based Data Access: the experience at the italian department of treasury. In: CAISE INDUSTRIAL TRACK. **Proceedings...** CEUR-WS.org, 2013. p.9–16. (CEUR Workshop Proceedings, v.1017).

AT&T, C. **Graphviz - Graph Visualization Software**. Accessed: 2012.

BACH, B.; PIETRIGA, E.; LICCARDI, I. OntoTrix: a hybrid visualization for populated ontologies. In: WWW '11: 19TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, Hyderabad, Inde. **Proceedings...** ACM, 2011.

BAEHRECKE, E. et al. Visualization and analysis of microarray and gene ontology data with treemaps. **BMC Bioinformatics**, [S.l.], v.5, p.84, 2004.

BALDONADO, M.; WOODRUFF, A.; KUCHINSKY, A. Guidel ines for Using Multiple Views in Information Visualization. In: WORKING CONFERENCE ON ADVANCED VISUAL INTERFACES (AVI '00). **Proceedings...** ACM New York, 2000.

BARLOW, T.; NEVILLE, P. A Comparison of 2-D Visualizations of Hierarchies. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION 2001 (INFOVIS'01), Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2001. p.131–. (INFOVIS'01).

BOSCA, A.; BONINO, D.; PELLEGRINO, P. OntoSphere: more than a 3d ontology visualization tool. In: SWAP 2005, THE 2ND ITALIAN SEMANTIC WEB WORKSHOP. **Proceedings...** CEUR-WS.org, 2005. (CEUR Workshop Proceedings, v.166).

BOSTOCK, M.; HEER, J. Protovis: a graphical toolkit for visualization. **IEEE Transactions on Visualization and Computer Graphics**, Piscataway, NJ, USA, v.15, n.6, Nov. 2009.

BREITMAN, K. K. (Ed.). **Web semântica**: a internet do futuro. Rio de Janeiro: LTC, 2005. 199p. v.1.

CARD, S. K.; MACKINLAY, J. D.; SHNEIDERMAN, B. **Readings in information visualization**: using vision to think. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

CARD, S. K.; NATION, D. Degree-of-interest trees: a component of an attention-reactive user interface. In: WORKING CONFERENCE ON ADVANCED VISUAL INTERFACES (AVI '02). **Proceedings...** ACM New York, 2002. p.231–245.

CATARCI, T. et al. **The On-TIME User Interface**. [S.l.]: Physica-Verlag HD, 2010.

CATENAZZI, N.; SOMMARUGA, L.; MAZZA, R. User-Friendly Ontology Editing and Visualization Tools: the owleasyviz approach. In: INTERNATIONAL CONFERENCE INFORMATION VISUALISATION, 13., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2009. p.283–288. (IV '09).

CHAN, Y.; KEETON, K.; MA, K. Interactive Visual Analysis of Hierarchical Enterprise Data. In: IEEE 12TH CONFERENCE ON COMMERCE AND ENTERPRISE COMPUTING (CEC). **Proceedings...** [S.l.: s.n.], 2010. p.180 – 187.

COCKBURN, A.; KARLSON, A.; BEDERSON, B. A review of overview+detail, zooming, and focus+context interfaces. **ACM Comput. Surv.**, New York, NY, USA, v.41, n.1, p.2:1–2:31, Jan. 2009.

DEFANTI, T. A.; BROWN, M. D.; MCCORMICK, B. H. Visualization: expanding scientific and engineering research opportunities. **Computer**, Los Alamitos, CA, USA, v.22, n.8, p.12–25, 1989.

D'ENTREMONT, T.; STOREY, M. Using a degree of interest model to facilitate ontology navigation. In: IEEE SYMPOSIUM ON VISUAL LANGUAGES AND HUMAN-CENTRIC COMPUTING, VL/HCC 2009. **Proceedings...** [S.l.: s.n.], 2009. p.127–131.

DIAS, E. **MAPEAMENTO TERMINOLOGICO**: uma experiencia baseada nas dissertacoes de um mestrado profissional. 2012. Master Thesis — Fluminense Federal University.

EHRIG, M.; SURE, Y. Ontology Mapping - An Integrated Approach. In: FIRST EUROPEAN SEMANTIC WEB SYMPOSIUM, Heraklion, Greece. **Proceedings...** Springer Verlag, 2004. p.76–91. (Lecture Notes in Computer Science, v.3053).

FALCONER, S. **OntoGraf**. Available in: <http://protegewiki.stanford.edu/wiki/OntoGraf>. Accessed in: 2012.

FEKETE, J. The InfoVis Toolkit. In: IEEE SYMPOSIUM ON INFORMATION VISUALIZATION, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2004. p.167–174. (INFOVIS '04).

FORSELL, C.; JOHANSSON, J. An heuristic set for evaluation in information visualization. In: INTERNATIONAL CONFERENCE ON ADVANCED VISUAL INTERFACES, New York, NY, USA. **Proceedings...** ACM, 2010. p.199–206. (AVI '10).

FREITAS, C.; PIMENTA, M.; SCAPIN, D. **User-Centered Evaluation of Information Visualization Techniques**: making the hci-infovis connection explicit. [S.l.]: Springer-Verlag, 2012.

FURNAS, G. Generalized Fisheye Views. In: HUMAN FACTORS IN COMPUTING SYSTEMS CHI'86 CONFERENCE. **Proceedings...** [S.l.: s.n.], 1986. p.16–23.

GILBERT, D. **JFreeChart**. Available in: <http://www.jfree.org/index.html>. Accessed in: 2012.

GOOGLE, I. **Google Maps with Street View**. Available in: <www.google.com/streetview>. Accessed in: 2012.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowl. Acquis.**, London, UK, UK, v.5, n.2, p.199–220, 1993.

GURR, C. Effective Diagrammatic Communication: syntactic, semantic and pragmatic issues. **Journal of Visual Languages and Computing**, [S.l.], v.10, n.4, p.317–342, 1999.

HEER, J.; CARD, S.; LANDAY, J. Prefuse: a toolkit for interactive information visualization. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, New York, NY, USA. **Proceedings...** ACM, 2005. p.421–430. (CHI '05).

HOP, W. et al. Using Hierarchical Edge Bundles to visualize complex ontologies in GLOW. In: ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING, 27., New York, NY, USA. **Proceedings...** ACM, 2012. p.304–311. (SAC '12).

HORRIDGE, M. **A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools**. [S.l.]: The University Of Manchester, 2011.

HUSKEN, P.; ZIEGLER, J. Degree-of-Interest Visualization for Ontology Exploration. In: IFIP TC 13 INTERNATIONAL CONFERENCE ON HUMAN COMPUTER INTERACTION (INTERACT'07), 11., Heidelberg. **Proceedings...** Springer-Verlag Berlin, 2007. p.116–119.

INXIGHT, I. **HyperTree Java Library**. Available in: <http://hypertree.sourceforge.net/>. Accessed in: 2012.

ISENBERG, P. et al. Grounded evaluation of information visualizations. In: WORKSHOP ON BEYOND TIME AND ERRORS: NOVEL EVALUATION METHODS FOR INFORMATION VISUALIZATION, 2008., New York, NY, USA. **Proceedings...** ACM, 2008. p.6:1–6:8. (BELIV '08).

KAPLER, T.; WRIGHT, W. Geo time information visualization. **Information Visualization**, [S.l.], v.4, n.2, p.136–146, July 2005.

KATIFORI, A. et al. Ontology visualization methods: a survey. **ACM Computing Surveys (CSUR)**, [S.l.], v.39, n.4, 2007.

KATIFORI, A. et al. Selected results of a comparative study of four ontology visualization methods for information retrieval tasks. In: CONFERENCE ON RESEARCH CHALLENGES IN INFORMATION SCIENCE (RCIS 2008). **Proceedings...** IEEE, 2008. p.133–140.

KEIM, D. A. Visual techniques for exploring databases. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY IN DATABASES (KDD'97). **Proceedings...** [S.l.: s.n.], 1997.

KEIM, D. A.; KRIEGEL, H. Visualization Techniques for Mining Large Databases: a comparison. **IEEE Trans. on Knowl. and Data Eng.**, Piscataway, NJ, USA, v.8, n.6, p.923–938, 1996.

KHAN, M.; KHAN, S. Data and Information Visualization Methods, and Interactive Mechanisms: a survey. **International Journal of Computer Applications**, New York, NY, USA, v.34, n.1, p.1–14, 2011.

KRAFTA, R. Space is the Machine, with a Ghost Inside. In: DEVELOPMENTS IN DESIGN & DECISION SUPPORT SYSTEMS IN ARCHITECTURE AND URBAN PLANNING. **Proceedings...** Eindhoven University of Technology, 2004. p.157–173.

KRIGLSTEIN, S.; WALLNER, G. Development Process and Evaluation of the Ontology Visualization Tool Knoocks - A Case Study. In: INTERNATIONAL JOINT CONFERENCE ON COMPUTER VISION, IMAGING AND COMPUTER GRAPHICS THEORY AND APPLICATIONS (VISIGRAPP/IVAPP), Portugal. **Proceedings...** SciTePress, 2011.

LAMPING, J.; RAO, R. Laying out and visualizing large trees using a hyperbolic space. In: ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 7., New York, NY, USA. **Proceedings...** ACM, 1994. p.13–14. (UIST '94).

LANZENBERGER, M.; SAMPSON, J. Alviz - a tool for visual ontology alignment. In: INFORMATION VISUALIZATION (IV '06). **Proceedings...** IEEE Computer Society, 2006. p.430–440.

LORENZATTI, A. et al. Uma ontologia de dominio para estratigrafia sedimentar baseada em primitivas conceituais e pictoricas. In: WORKSHOP DE ONTOLOGIAS NO STIL, 2009. **Proceedings...** [S.l.: s.n.], 2009.

MAZZA, R. **Introduction to Information Visualization**. 1.ed. [S.l.]: Springer Publishing Company, Incorporated, 2009.

MÉTRAL, C.; FALQUET, G.; VONLANTHEN, M. An Ontology-based Model for Urban Planning Communication. In: **Ontologies for Urban Development**. [S.l.: s.n.], 2007. p.61–72.

MUNZNER, T. A Nested Model for Visualization Design and Validation. **IEEE Transactions on Visualization and Computer Graphics**, Piscataway, NJ, USA, v.15, n.6, p.921–928, 2006.

NEWMAN, D.; BENNETT, M. Semantic Solutions for Financial Industry Systemic Risk Management. In: SEMANTIC TECHNOLOGY & BUSINESS CONFERENCE. **Proceedings...** [S.l.: s.n.], 2012.

NIELSEN, J. Usability inspection methods. In: CONFERENCE COMPANION ON HUMAN FACTORS IN COMPUTING SYSTEMS, New York, NY, USA. **Proceedings...** ACM, 1995. p.377–378. (CHI '95).

NOY, N. F.; MCGUINNESS, D. L. **Ontology Development 101**: a guide to creating your first ontology. [S.l.]: Standford University, 2001.

NOY, N.; FERGERSON, R.; MUSEN, M. The Knowledge Model of ProtProtègè: combining interoperability and flexibility. In: EUROPEAN WORKSHOP ON KNOWLEDGE ACQUISITION, MODELING AND MANAGEMENT, 12., London, UK, UK. **Proceedings...** Springer-Verlag, 2000. p.17–32. (EKAW '00).

ONG, T. et al. Newsmap: a knowledge map for online news. **Decis. Support Syst.**, Amsterdam, The Netherlands, The Netherlands, v.39, n.4, p.583–597, June 2005.

ORACLE, C. **Java 3D API**. Available in: <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138252.html>. Accessed in: 2012.

PIZZINATO, P.; RIGO, S.; VIEIRA, R. Um Visualizador Online de Ontologias Utilizando Arvores Hiperbolicas. In: III SEMINARIO DE PESQUISA EM ONTOLOGIA NO BRASIL (ONTOBRAS). **Proceedings...** [S.l.: s.n.], 2010.

ROBERTS, J. C. State of the Art: coordinated & multiple views in exploratory visualization. In: FIFTH INTERNATIONAL CONFERENCE ON COORDINATED AND MULTIPLE VIEWS IN EXPLORATORY VISUALIZATION, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2007. p.61–71. (CMV '07).

SCHEVERS, H.; TRINIDAD, G.; DROGEMULLER, R. Towards Integrated Assessments for Urban Development. **Journal of Information Technology in Construction (ITcon): Special Issue Decision Support Systems for Infrastructure Management**, [S.l.], v.11, p.225–236, 2006.

SHNEIDERMAN, B. The Eyes Have It: A task by data type taxonomy for information visualizations. In: IEEE VISUAL LANGUAGES, College Park, Maryland 20742, U.S.A. **Proceedings...** [S.l.: s.n.], 1996. n.UMCP-CSD CS-TR-3665, p.336–343.

SILVA, I. C. S.; FREITAS, C. M. D. S. Requirements for Interactive Ontology Visualization - Using Hypertree +2.5D Visualization for Exploring Relationships between Concepts. In: INTERNATIONAL JOINT CONFERENCE ON COMPUTER VISION, IMAGING AND COMPUTER GRAPHICS THEORY AND APPLICATIONS (VISIGRAPP/IVAPP). **Proceedings...** [S.l.: s.n.], 2011. p.242–248.

SILVA, I. C. S.; FREITAS, C. M. D. S. Using Multiple Views for Visual Exploration of Ontologies. In: IV SEMINAR ON ONTOLOGY RESEARCH IN BRAZIL (ONTOBRAS) AND VI INTERNATIONAL WORKSHOP ON METAMODELS, ONTOLOGIES AND SEMANTIC TECHNOLOGIES (MOST). **Proceedings...** CEUR-WS, 2011. p.25–36.

SILVA, I. C. S.; NETTO, V. M.; FREITAS, C. M. D. S. Novos caminhos para simulação urbana: integrando métodos de visualização de informações e modelagem de agentes e redes espaciais. In: XIII CONGRESO IBEROAMERICANO DE GRÁFICA DIGITAL (SIGRADI). **Proceedings...** [S.l.: s.n.], 2009. p.153–155.

SILVA, I. da; FREITAS, C. Using Visualization for Exploring Relationships between Concepts in Ontologies. In: INTERNATIONAL CONFERENCE ON INFORMATION VISUALISATION, 17, 2011, Los Alamitos, CA, USA. **Proceedings...** IEEE Computer Society, 2011. v.0, p.317–322.

SILVA, I. da; FREITAS, C.; SANTUCCI, G. An Integrated Approach for Evaluating the Visualization of Intensional and Extensional Levels of Ontologies. In: WORKSHOP: BEYOND TIME AND ERRORS - NOVEL EVALUATION METHODS FOR VISUALIZATION (BELIV), New York, NY, USA. **Proceedings...** ACM, 2012. p.2:1–2:7. (BELIV ’12).

SILVA, I.; SANTUCCI, G.; FREITAS, C. Ontology Visualization: one size does not fit all. In: EUROVA 2012: INTERNATIONAL WORKSHOP ON VISUAL ANALYTICS, Vienna, Austria. **Proceedings...** Eurographics Association, 2012. p.91–95.

SINTEK, M. **OntoViz**. http://protegewiki.stanford.edu/wiki/OntoViz: German Research Center for Artificial Intelligence, 2007. Available in: <http://protegewiki.stanford.edu/wiki/OntoViz>. Accessed in: 2012.

SOUZA, K.; SANTOS, A. dos; EVANGELISTA, S. Visualization of ontologies through hypertrees. In: LATIN AMERICAN CONFERENCE ON HUMAN-COMPUTER INTERACTION, New York, NY, USA. **Proceedings...** ACM, 2003. p.251–255.

SOWA, J. F. **Guided Tour of Ontology**. http://www.jfsowa.com/ontology/guided.htm: Knowledge Creation Diffusion Utilization, 2006. Available in: <ttp://www.jfsowa.com/ontology/guided.htm>. Accessed in: 2012.

SPENCE, R. **Information Visualization**: design for interaction. 2.ed. England: Pearson, Prentice Hall, 2007.

STOREY, M. e. a. Jambalaya: interactive visualization to enhance ontology authoring and knowledge acquisition in protègè. In: WORKSHOP ON INTERACTIVE TOOLS FOR KNOWLEDGE CAPTURE. **Proceedings...** [S.l.: s.n.], 2001.

WARD, M. XmdvTool: integrating multiple methods for visualizing multivariate data. In: VISUALIZATION ’94, Los Alamitos, CA, USA. **Proceedings...** IEEE Computer Society Press, 1994. p.326–333. (VIS ’94).

WARE, C. **Visual Thinking**: for design (morgan kaufmann series in interactive technologies). First Edition.ed. [S.l.]: Morgan Kaufmann, 2008.