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

JOSE MARTIN LOZANO APARICIO

Ontology View: a new sub-ontology extraction method

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

Advisor: Prof. Dr. Mara Abel
Coadvisor: Prof. Dr. Marcelo Pimenta

Porto Alegre
January 2015

CIP – CATALOGING-IN-PUBLICATION

Lozano Aparicio, Jose Martin

Ontology View: a new sub-ontology extraction method / Jose Martin Lozano Aparicio. – Porto Alegre: PPGC da UFRGS, 2015.

169 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2015. Advisor: Mara Abel; Coadvisor: Marcelo Pimenta.

1. Ontology View. 2. Foundational Ontology. 3. Sub-Ontology Extraction. 4. HCI. I. Abel, Mara. II. Pimenta, Marcelo. III. Título.

“If I have seen farther than others,
it is because I stood on the shoulders of giants.”
— Sir Isaac Newton
and me, of course.

ACKNOWLEDGEMENTS

First and foremost, it comes God. Without his assistance, it would be never possible to finish my work.

This thesis grew out of a series of meetings and discussions with my supervisor Dr Mara Abel. All her advices help me to overcome the different problems that aroused during the project, since the planning of the topic to the conclusion of it. I greatly appreciate all the effort she has put into mentoring me. Special thanks also to my co-advisor Dr Marcelo Pimenta, for his support, guidance and helpful suggestions.

I thank my colleagues and friends of BDI Group at UFRGS, who made my master study a wonderful academic experience, specially to Joel Carbonera, who guides me in this process. Thanks are also certainly due to my great Peruvian friends.

I would like to express my immense gratitude to my parents, for their encouragement and advice. Their support throughout the years has been unwavering despite the physical distance that has separated us during my master study. They encouraged my intellectual curiosity from a young age. Throughout my life, they have always ensured that every opportunity is available to me.

ABSTRACT

Nowadays many petroleum companies are adopting different knowledge-based systems aiming to have a better reservoir quality prediction. However, there are obstacles that not allow different background geologists to retrieve information without needing the help of an information technology expert. The main problem is the heterogeneity semantic of end users when doing queries in a visual query system (VQS). This can be worst when there is new terminology in the knowledge-base affecting the user interaction, particularly for novice users.

In this context, we present theoretical and practical contributions that exploit the synergism between ontology and human computer interaction (HCI). On the theory side, we introduce the concept of ontology view for well-founded ontology and provide a formal definition and expressive power characterization. We focus in the ontology view extraction of a well-founded and complete ontology based on ontological meta-properties and propose a language independent algorithm for sub-ontology extraction, which is guided by ontological meta-properties.

On the practical side, based on the principles of HCI and interaction design, we propose a new Visual Query System that uses the ontology view approach to guide the query process. Also, our design includes data visualizations that will help geologists to make sense of the retrieved data. Furthermore, we evaluated our interaction design with five users performing a usability testing through a questionnaire in a controlled experiment. The evaluation was performed over geologists that work in the area of petroleum geology.

The approach proposed is evaluated on the petrography domain taking the communities of Diagenesis and MicroStructural adopting the well known criteria of precision and recall. Experimental results show that relevant terms obtained from the documents of a community varies from 30 to 66 % of precision and 4.6 to 36% of recall depending on the approach selected and the parameters combination. Furthermore, results show that almost for all the parameters combination that recall and f-measure obtained from diagenesis articles using the sub-ontology generated for the diagenesis community is greater than recall and f-measure using the sub-ontology generated for microstructural community. On the other hand, results for all the parameters combination that recall and f-measure obtained from microstructural articles using the sub-ontology generated for the microstructural community is greater than recall and f-measure using the sub-ontology generated for diagenesis community.

Keywords: Ontology View. Foundational Ontology. Sub-Ontology Extraction. HCI.

Vista de Ontologia: um novo metodo para extrair uma sub-ontologia

RESUMO

Hoje em dia, muitas empresas de petróleo estão adotando diferentes sistemas baseados em conhecimento com o objetivo de ter uma melhor predição de qualidade de reservatório. No entanto, existem obstáculos que não permitem geólogos com diferentes formações recuperar as informações sem a necessidade da ajuda de um especialista em tecnologia da informação. O principal problema é a heterogeneidade semântica dos usuários finais quando fazem consultas em um sistema de consulta visual (VQS). Isto pode ser pior quando há uma nova terminologia na base de conhecimentos que afetam a interação do usuário, especialmente para usuários novatos.

Neste contexto, apresentamos contribuições teóricas e práticas que explora o sinergismo entre ontologia e interação homem-computador (HCI). Do lado da teoria, introduzimos o conceito de visão de ontologia bem fundamentada e a sua definição formal. Nós nos concentramos na extração de vista ontologia de uma ontologia bem fundamentada e completa, baseando-nos em meta-propriedades ontológicas e propusemos um algorítmo independente da linguagem para extração de sub-ontologia que é guiada por meta-propriedades ontológicas.

No lado prático, baseado nos princípios de HCI e desenho de interação, propusemos um novo sistema de consulta visual que usa o enfoque de vistas de ontologias para guiar o processo de consulta. Também o nosso desenho inclui visualizações de dados que ajudarão geólogos a entender os dados recuperados. Além disso, avaliamos nosso desenho com um teste de usabilidade a-través de um questionário em experimento controlado. Cinco geólogos que trabalham na área de Geologia do Petróleo foram avaliados.

O enfoque proposto é avaliado no domínio de petrografia tomando as comunidades de Diagênese e Microestrutural adotando o critério de precisão e revocação. Os resultados experimentais mostram que termos relevantes obtidos de documentos de uma comunidade varia entre 30 a 66% de precisão e 4.6 a 36% de revocação, dependendo do enfoque selecionado e da combinação de parâmetros. Além disso, os resultados mostram que, para toda combinação de parâmetros, a revocação obtidos de artigos de diagênese usando a sub-ontologia gerada para a comunidade de diagênese é maior que a revocação e f-measure usando a sub-ontologia gerada para a comunidade de microestrutural. Por outro lado, resultados para toda combinação de parâmetros mostram que a revocação e f-measure obtida de artigos de microestrutural usando a sub-ontologia gerada para a comunidade de microestrutural é maior que a revocação e o f-measure usando a sub-ontologia gerada para a comunidade de diagêneses.

Palavras-chave: Vista de Ontologia, Ontologia fundamental, Extração de uma sub-ontologia, Interação Humano Computador.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

UX	User Experience
HCI	Human Computer Interaction
VQL	Visual Query Language
VQS	Visual Query System
UFO	Unified Foundational Language
OLED	OntoUML Light-Weight Editor
OVUFO	Ontology View for Unified Foundational Ontology
SEL	selection algorithm
OWL	Ontology Web Language
GUI	Graphical User Interface

CONTENTS

1 INTRODUCTION

Many organizations invest huge amount of money capturing, organizing and storing data that will support the business decision making. These organizations need to improve the way that they organize and store such information since not only larger amounts of data are being collected, but new ways to connect and interoperate such data are emerging dynamically. So, the idea of large amount of data (extension) and very flexible data models (intention) evolving over time contribute to making nowadays information management an increasingly challenging task.

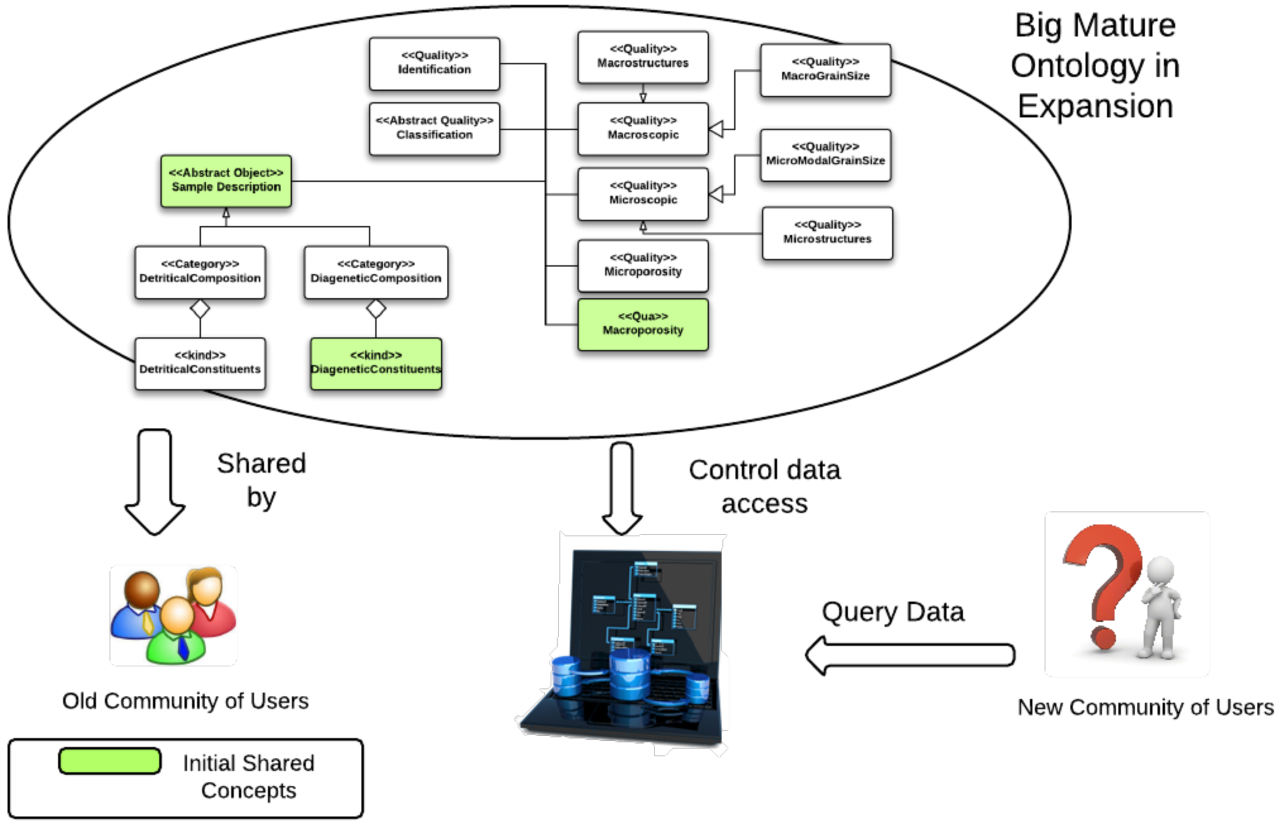
Since semantic technologies are usually considered as the basis for new approaches to deal with this task, there are solutions based on adoption of expressive power of ontology concepts. Indeed, ontology is an adequate way to take into account large and complex schema of related concepts, which can be used at query time in order to allow reasoning about schema structure and to build inferred answers or even intentional queries.

However, the terminology has been grown to a point where information processing and exchange is seriously hampered, as it can no longer be guaranteed that multiple parties interpret the data in the same way and use the same terminology. This implies a system evolution that supports the information process and enhances the data exploration to retrieve the right information. As a real case, there is Petroledge®, an ontology-based system, that is consulted by PetroQuery® [1], which is a visual query system(VQS). PetroQuery® query model uses the ontology to guide the consultation. This ontology includes the concepts required for chemical rock-reservoir evaluation, igneous and metamorphic petrography, chemical analysis of igneous rocks and stratigraphic descriptions, supporting as diverse applications as multidimensional consultation of large set of data, image indexing and several other minor applications (ABEL et al., 2012). As a result, different communities of users have distinct kinds of queries according to their knowledge and the kind of problem the users deal with. Because of the specialized knowledge, the users find problems in formulating queries since they only know part of the terminology that is applied by the interface. A better interaction would be achieved if the interface is customized to each knowledge community. In Figure 1.1, we illustrate our motivation problem: a large mature ontology and the difficulties that a new community of users meet when trying to use the system because of the amount of unknown terminology offered by the system when supporting consultation.

Every ontology-based application will offer some level of difficulty when users build consultations for decision support. This can be reduced by an appropriate interface design. As the application evolves through time, however, this initial difficulty can be increased by the aggregation of new concepts, new functionalities and new community of users. Each community will retain a partial understanding of the terminology of the domain ontology and will find difficult to find out the relevant concepts that support its decision. Thus, we need to reduce the

[1] Petroledge and Petroquery® are trademarks of Endeeper Co. that commercializes the systems.

Figure 1.1: Motivation Problem: evolving mature ontology is shared by old and new community of users.


Big Mature Ontology in Expansion
<<Quality>> Identification
<<Abstract Quality>> Classification
<<Quality>> Macrostructures
<<Quality>> Macroscopic
<<Quality>> MacroGrainSize
<<Quality>> MicroModalGrainSize
<<Quality>> Microscopic
<<Quality>> Microstructures
<<Quality>> Microporosity
<<Qua>> Macroporosity
<<Abstract Object>> Sample Description
<<Category>> DetriticalComposition
<<Category>> DiageneticComposition
<<kind>> DetriticalConstituents
<<kind>> DiageneticConstituents
Shared by
Control data access
Query Data
Old Community of Users
New Community of Users
🟩 Initial Shared Concepts

overload of information charged in user's memory by creating a subset of the whole ontology. This dissertation focuses on ontology view extraction. We propose a sub-ontology extraction algorithm based on the ontological meta-properties. We define conservation principles that the sub-ontology should fulfill. Besides, we developed a visual query system prototype based on the importance of human computer interaction techniques to enhance the data analysis.

1.1 Why Combining Ontology & HCI

Through our research in Ontology and HCI (Human Computer Interaction), we have realized that data analytics is an important aspect in a visual query system. A visual query system can benefit from both disciplines joining forces, and our solution lies in the intersection. Ontology focuses in organizing the concepts and relations orienting the user to formulate the query; HCI focuses on interaction techniques and visualization that leverage the human mind and facilitate the analysis of the retrieved data.

The field of HCI can provide new insights to enhance the interaction in PetroQuery® through a better navigation in the ontology that guide the query process and new visualizations to improve user's sensemaking when doing data analysis.

1.2 Dissertation Overview & Main Ideas

Our approach has proposed an initial experimentation over the visual query system described in appendix A. According to our report, we identified some problems related to the interaction design that was caused by the application evolution. For instance, the principle of Recognition rather than recall from the Nielsen heuristics was not satisfied. This means that there is too much information that not let user to perform the desired query. Thus, the ontology-based system should provide the set of concepts that user requires to formulate the query. This implies a reduction of the whole ontology to obtain a relevant subset of terms. For this purpose, it can be applied ontology module (DORAN, 2009; SEIDENBERG; RECTOR, 2006; D'AQUIN; SABOU; MOTTA, 2006), or ontology view (NOY; MUSEN, 2009; BHATT et al., 2004a). Both of them involve a common step, the sub-ontology extraction. Thus, we designed and developed the sub-ontology extraction algorithm, which uses the Unified Foundational Ontology (UFO) meta-properties to guide the selection of the concepts. Furthermore, we provide three different approaches with flexibility for setting three parameters that will return different well-founded ontology views. We decided to use the term *ontology view* to our generated subset because its definition is focused on user customization and *well-founded* because our algorithm uses UFO meta-properties. We tested over the petrography domain taking the communities of Diagenesis and MicroStructural adopting the well known criteria of precision and recall. The experimental results show that relevant terms obtained from the documents of a community varies from 10 to 70% of precision and 20 to 40% of recall depending on the approach selected and the parameters combination.

On the interface side, we used the theory previously defined to establish that new communities, which have partial acknowledge of the ontology, can benefit of the view visualizing the concepts that are just well understood by them. Thus, instead of showing lists of concepts, attributes and values; we presented a module that visualizes the ontology, but without the annotating meta-data. Finally, we developed a new interaction design implemented in RockQuery prototype as a result of the experimentation.

1.3 Research Contributions

As we know, systems evolve over time, being extended, combined and integrated. Knowledge-based systems and their ontology modeled portion that affects the end-users interaction. Therefore, this thesis bridges ontology and HCI research. We contribute by answering two important, fundamental research questions:

- **How can we improve the comprehension of data to be consulted in a large knowledge base?** Our idea is an Ontology View Approach.

- **How to enhance the interaction in the Visual Query System** PetroQuery$^{®}$ Our idea is the use of HCI techniques and Visualization.

A summary of contributions are listed:

- Algorithms: We design and develop an algorithm that performs the sub-ontology extraction using UFO as a base to obtain the adequate subset of concepts for the view.

- Theories: We present the formalization of the ontology view approach, which enables the segmentation of the ontology and let the extension to the new community knowledge.

- Tools: We develop an application that lets a user import an well- founded ontology modeled with OLED and perform a different subset extraction method.

- Prototype: We develop an interaction design implemented in the RockQuery prototype. We deal with the problems of overhead of terminology with the use of ontology view in the exploration of concepts to formulate the query.

1.4 Structure of this dissertation

This text consists of nine chapters. The core theory is in chapter 6. This chapter has been submitted for publication and accepted in the IEEE International Conference on Tools with Artificial Intelligence (LOZANO et al., 2014).

Chapter 1 describes the research background and the main ideas, and then specifies the research contributions in the research questions and outlines the structure of the dissertation.

Chapter 2 describes the related work done in the area of visual query systems and a literature review of HCI techniques that were used in our development.

Chapter 3 summarizes the ontology issues, foundational ontology theory and ontology visualization.

Chapter 4 describes the different techniques for sub-ontology extraction. Three main groups are identified: query base, network partitioning and traversal approach.

Chapter 5 describes the well-founded ontology developed for purpose of testing the ontology view approach.

Chapter 6 describes the fundamentals of our ontology view approach. It defines the ontology view, presents the different algorithms used in the extraction, defines the approach for obtaining the view. Also, it contains a description of a tool, that lets the ontology engineer to extracts subsets of any ontology modeled with OLED.

Chapter 7 describes the visual query system, the components and the interaction design. A further description of the prototypes done before the final implementation are discussed in the appendix B. In addition, we performed an experimentation of PetroQuery® and it is described in appendix A.

Chapter 8 describes the evaluation of our ontology view approach and the visual query system. We make a comparison in the part of ontology selection with other techniques and the precision and recall are calculated through a control experiment described in the chapter. We also discuss the construction of a well founded ontology of the petrography domain and the acceptance of the developed visual query system prototype with an experimental study.

Finally, Chapter 9 concludes this thesis. It gives a summary (Section 9.1) and a scope to potential future development directions (Section 9.2).

2 VISUAL QUERY SYSTEMS AND ISSUES IN HCI: FUNDAMENTALS

The majority of visual query systems rely mostly on tabular data displays and the process of querying has not a navigation structure to guide users in the formulation task. Thus, the user interface is often not designed for situations at which users can hamper the query interaction. This chapter introduces the most important concepts and methods employed in the field of Visual Query System. We start by reviewing some key concepts that are particularly relevant for our work: interaction design, information visualization, sensemaking. A proper understanding of these aspects is indeed a prerequisite for the design of conversationally competent visual query system. After this overview, we move to a more technical discussion of the software architectures used to implement practical visual query systems.

2.1 Interaction Design

Before explaining interaction design, we introduce briefly basic notions of HCI. In general, the goal of HCI design is to produce an user interface that is easy to use and learn. HCI has really been changing over the years. There are three waves of HCI research. The first wave in 1980 was in studying ergonomic and human factors issues of interaction with computing systems. Then in 1990s was a focus on tasks, efficiency, and completion rates through controlled lab experiments. Now the tendency is a focus on understanding use of new systems in daily life and how to improve the users' experiences.

Design according to Oxford Dictionary is a planning or intention in mind with the purpose to be executed. Design is balance between the utility, the usability and the beauty. The field of interaction design is concerned with the development of products and systems that support the way people think and behave, to provide satisfying interactive experiences. In addition, many academic disciplines contribute to the study and application of interaction design, such as psychology, cognitive science, engineering and computer science. Each of these academic disciplines informs the process of developing interactive products or systems that provide a positive user experience.

The process of interaction design (ROGERS; SHARP; PREECE, 2011) involves many steps in different detailed levels. First, we have to think about the design problem, understand the users' needs, produce possible conceptual models, prototypes, evaluate them according to usability guidelines and objectives of user experience, think about implications of the design from the usability test, do modifications in prototypes and so on. There are three key characteristics of interaction design (a) Focus in the user tasks, (b) Empirical evaluation, (c) Iterative Design. The process of interaction design involves four basic activities:

Establishing Requirements: This stage involves establishing and answering a series of design questions, such as: What does the user need from the design? How easy is it to use the system or product? Does it fit the context? Does it provide the user with sufficient means of completing their device or system-based aims and objectives? Does it have superficial appeal?

Design alternatives: The process seeks input from users themselves to ensure the final design is as free from user-unfriendly elements as possible. The opinions of intended users are sought through questionnaires and interviews, whilst naturalistic observation can be particularly informative. If problems are identified during this phase then alternative designs are, therefore, necessary.

Prototyping: The design process is iterative; at various stages it is important to trial your product or system to ensure any unforeseen problems are brought to light so they can be remedied before the final design is set in stone. Prototypes allow you to see how real users, free from the biases that might influence the interactions of those involved in the design process, interact with your product or system.

Evaluating: This last activity is known as usability engineering. Usability engineering specifies quantitative metrics about a product performance, document and evaluate with respect to those metrics. These four stages are then repeated until problems are eliminated, user needs are satisfied and an enjoyable user experience is provided.

Some guidelines for user interface design are described in Smith and Mosier (1986). Furthermore, there are eight rules for interface design explained in Shneiderman and Plaisant (2004). These rules are:

- Strive for consistency: it implies the same thing in a similar situation. Thus the interface should contain consistent visual layout and identical terminology.

- Enable use shortcuts: The interface should have a set of familiar abbreviations, special keys for most frequently used tasks.

- Offer informative feedback: The interface should show response to reduce uncertainty when performing operations and the status of operation.

- Design dialogs to yield closure: dialogs should have a beginning, middle and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.

- Prevent errors: Limit errors a user can make. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.

- Permit easy reversal of actions: The interface should have buttons that return to previous step.
- Support internal locus of control: Means to design the system to make users the initiators of actions rather than the responders.
- Reduce short term memory load: The basis for design decisions is from the Miller's Magic 7 theory (MILLER, 1956).

There are cognitive processes (POSNER, 1993) underlying the interactive experience, which are as follows:

- **Attention** is the cognitive process of selecting sensory information from our environment, whilst ignoring or filtering out everything else in the sensory stream.
- **Memory** is the cognitive process responsible for the encoding, storage and retrieval of information received by our senses.
- **Language** is a cognitive process that involves learning, understanding, producing and sharing meaning. Almost all tasks require some form of communication, whether it is through written or verbal instruction, so it is essential to use appropriate language in design; otherwise the user will not know what, where, why, when or how they should interacting in order to achieve their device- or system-based aims and objectives.
- **Reasoning** is the cognitive process enabling evaluation and generation of logical arguments, verification of facts and the assimilation, accommodation and rejection of new information on the basis of existing knowledge. Reasoning also allows us to develop new ways of thinking with one idea leading to another. Reasoning underlies the selection of alternate strategies when an existing approach to a problem proves unsuccessful.
- **Problem-Solving** is the cognitive process enabling evaluation and generation of logical arguments, verification of facts and the assimilation, accommodation and rejection of new information on the basis of existing knowledge. Reasoning also allows us to develop new ways of thinking with one idea leading to another. Reasoning underlies the selection of alternate strategies when an existing approach to a problem proves unsuccessful. There are three special characteristics that define problem solving:
  - **Goal directness**: behavior is generated on the basis of a current goal.
  - **Sub-goal decomposition**: if a goal is completed with one simple motion, then this represents the most primitive form of problem-solving. However, higher order problem-solving involves the deconstruction of the overall goal into the necessary component behaviors.

  - **Operator selection**: each sub-goal involves the selection of an appropriate action that fits into the overall sequence. Each of these sub-goal actions is an operator, and in the correct order they solve the overall problem.

- **Decision making** is important to minimize the costs associated with users' actions, so they cannot cause damage and they feel able to move freely through the system or interact with a device confident in the knowledge that their decision making will not prove deleterious.

2.1.1 Research Methods and Techniques

Understanding who users are and what they are doing can and should be a critical component in interaction design. The techniques and methods used to obtain user and task information in our work is described below.

2.1.1.1 Survey Methods

Survey research is one of the most important areas of measurement in applied social research. They provide feedback from the point of view of users. Although , the data collected in the survey can be biased. This means that the answers for some kind of questions may not be reliable. Thus, we have to care when planning the goals of the survey and selecting a representative part of population. Researches use three types of questions in surveys, namely multiple choice, numeric open-end and text open-end (TROCHIM; DONNELLY, 2008). There are two forms of survey research Questionnaire and interviews.

Questionnaire is a method for the elicitation, and recording and collecting information. HCI researchers use questionnaires as tools to capture users' mind. Some well known questionnaires in HCI are listed below:

- Questionnaire for user interface satisfaction(QUIS) (CHIN; DIEHL; NORMAN, 1988) aims to assess users' subjective satisfaction with specific aspects of the human-computer interface. It also contains eleven specific interface factors that are organized hierarchically, namely screen factors, terminology and system feedback, learning factors, system capabilities, technical manuals, on-line tutorials, multimedia, voice recognition, virtual environments, internet access, and software installation. Each factor measures users' satisfaction with the general properties of the interface as well as the specific ones.

- Perceived Usefulness and Ease of use(PUEU) (DAVIS, 1989) refers to the degree to which person believes that using a particular system would be free of effort in the case of ease of use and that particular system would enhance his or her job performance in the case of usefulness.

- Nielsen's attributes for usability(NAU) (NIELSEN, 1993) are five components, which are assessed in a user interface. Those are: learnability, efficiency, memorability, errors, and satisfaction. Based on these attributes are formed questions for testing the user interface usability.
- Nielsen's heuristic evaluation(NHE) (NIELSEN, 1993) is a usability engineering method for finding and assessing usability problems in a user interface design as part of an iterative design process. They are called *heuristics* because they are more in the nature of rules of thumb than specific usability guidelines. Examples of these questionnaires are found Perlman's site[1].

Interviews are flexible because the interviewer has the freedom to change some questions or the asking order of the questions according to the reactions of the users. Finally, interviews are participatory since they require both the interviewer and the participant to join in an interactive conversation. Shaughnessy, Zechmeister and Zechmeister (2006) present the most important types of interviews, which are face-to-face and telephone interviews. In face-to-face interviews, the interviewer works directly with the respondent. Unlike questionnaires, the interviewer has the opportunity to monitor the user and ask follow-up questions. On the other hand, telephone interviews enable a researcher to gather information rapidly, but the interviewed people can feel uncomfortable.

Mainly, there are three methods that are used in designing the interviews in HCI research (ROGERS; SHARP; PREECE, 2011). Unstructured interviewing methods are used during the earlier stages of usability evaluation. The interviewer's objective at this stage is to gather as much information as possible concerning the user's experience and on their expectations of the system. Semi-structured interviews are used when the interviewer has a better understanding of system requirements. Therefore, a more focused interview design can be used to focus on the points of interest. However, there can still be a degree of flexibility to allow the user to expand on an answer. Finally, structured interviewing has a specific, predetermined agenda with specific questions to guide and direct the interview. The interviewer, in this design, has a fully developed product and prepares questions to measure the user's reactions to that product.

2.1.1.2 GQM approach

Goal Question Metric (GQM) approach (BASILI; CALDIERA; ROMBACH, 1994) is a measure for software quality. It is based upon the assumption that for an organization to measure in a purposeful way it must first specify the goals for itself and its projects, then it must trace those goals to the data that are intended to define those goals operationally, and finally provide a framework for interpreting the data with respect to the stated goals. Furthermore, it consists of three major levels: Conceptual level (Goal), Operative level (Question), Quantita-

[1] http://hcibib.org/perlman/question.html

tive level (Metric). In the conceptual level, a goal is defined for an object relative to a particular environment. In the operational level, it is defined a set of questions that will assess the specific goal. In the quantitative level, it is measured the answer into metrics that could be objective or subjective metric.

2.1.1.3 Paper Prototyping

Paper prototyping is a method mainly used to design, test and improve user interfaces. Snyder (2003) defined paper prototyping as one type of usability test of the user interface. Paper-based prototyping is the quickest way to get feedback on your preliminary user interface information architecture, design, and content. Paper prototypes are easy to create and require only paper, scissors and sticky notes.

Snyder (2003) explains in details how it works. The first step is to come up with some scenarios or tasks that you would like the users to perform. Having that on mind, the next step is to make paper-based prototype, which could be a simple drawing on paper or printed-out screen-shots. The real session begins when you present the paper-based prototype design to the potential end users and inform them what task they are required to perform. Users will try to think how they perform the tasks by using this prototype design. In this process the users will feel in real if the interface or solution works for them and have a direct opinion about the design. One of the advantages of using paper prototyping is that we foster design thinking (BUXTON, 2010).

Low-fidelity prototypes are often paper-based and do not allow user interactions. They range from a series of hand-drawn mock-ups to printouts. In theory, low-fidelity sketches are quicker to create. Low-fidelity prototypes are helpful in enabling early visualization of alternative design solutions, which provokes innovation and improvement. An additional advantage is that users may feel more comfortable suggesting changes.

High-fidelity prototypes are computer-based, and usually allow realistic (mouse-keyboard) user interactions. They are assumed to be much more effective in collecting true human performance data (e.g., time to complete a task), and in demonstrating actual products to clients, management, and others.

2.2 Information Visualization

Why should we be interested in visualization (WARE, 2004)? Because the human visual system is a pattern seeker of enormous power and subtlety. The eye and the visual cortex of the brain form a massively parallel processor that provides the highest-bandwidth channel into human cognitive centers. At higher levels of processing, perception and cognition are closely interrelated, which is the reason why the words *understanding* and *seeing* are used as synonymous. However, the visual system has its own rules. We can easily see patterns presented in certain ways, but if they are presented in other ways, they become invisible. The more general point is that when data is presented in certain ways, the patterns can readily be perceived. If we can understand how perception works, our knowledge can be translated rules for displaying information. Following perception-based rules, we can present our data in such way that the important and informative patterns stand out. If we disobey the rules, our data will be incomprehensible or misleading.

Visualization can be a means to let users gain insights into large amounts of information quickly. The information visualization mantra stated by Shneiderman (1996) suggests how tasks can be supported through interactive visualization: *Overview first, zoom and filter, then details-on-demand*. Data visualization builds a bridge from data to knowledge, but only if the tools that we use were built on understanding of visual perception (how we see) and cognition (how we think). Data visualization tools must focus attention and augment memory. Memory plays an important role in human cognition. Because memory suffers from certain limitations, visual analysis tools must be rooted in an understanding of how people think to augment memory. According to Few (2006), good tools can help us increase:

- The amount of information that we can compare (that is, greater quantity)
- The range of information that we can compare (that is, more dimensions)
- The different views of the information that we can compare (that is, multiple perspectives)

There are a diversity of works in the field of information visualization that address visualization variants for different data types and structures, as well as suitable interaction techniques to let users interactively explore and exploit the presented information. Card, Mackinlay and Shneiderman (1999) provide a selection of computer aided approaches in the field and a reference model.

This model is divided into several stages starting with raw data that is subsequently transformed into data tables. These data tables are enriched to visual structures by mapping them to visual attributes. Finally, the visual data gets rendered into a view that is perceived by a user. In each stage the user can interact in different ways.

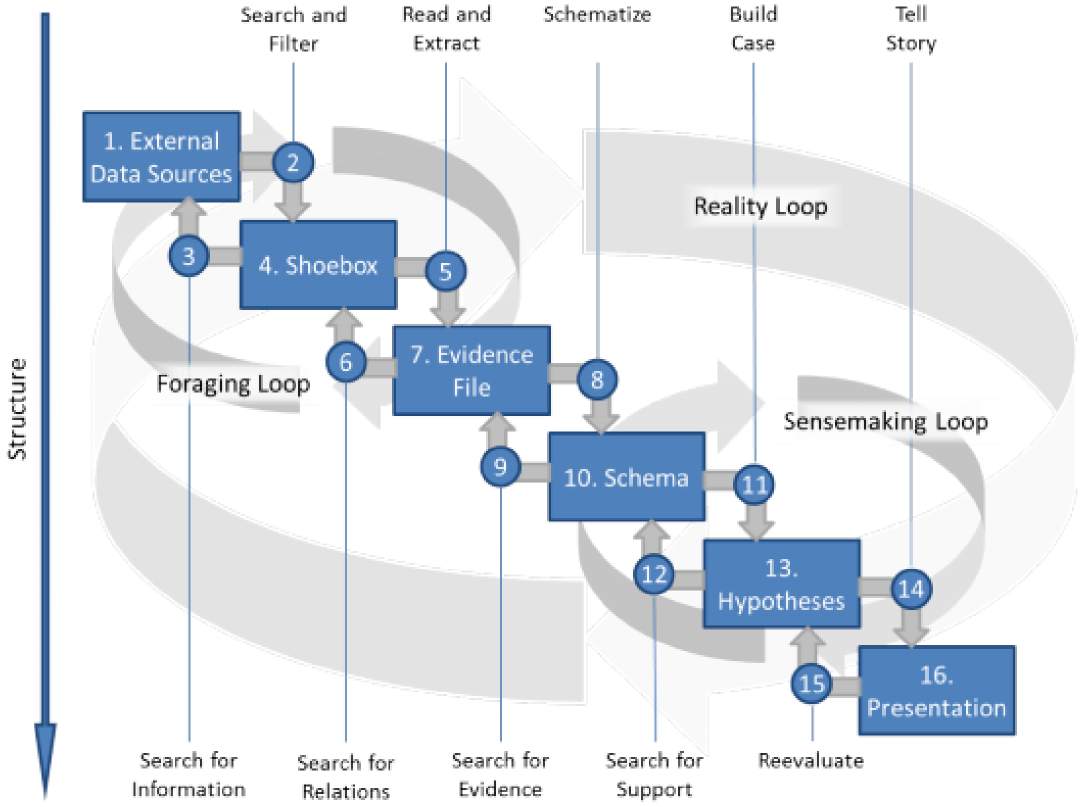
Research into the visualization of information has shown that third dimension can inhibit users and make interfaces more confusing. 3D visualizations have often hindered, rather than supported, participants in their searching activities. Research by Modjeska (2000) has shown that 25 % of the population struggle with 3D visualization displayed on 2D device, such as computer screen. Investigation by Sebrechts et al. (1999) also showed that participants were significantly slower at using a 3D interface, unless they had significant computer skills. Considering these challenges, however, the research described below highlights some of the ideas that have been proposed for 3D visualizations.

2.3 Sensemaking

Sensemaking seems primarily to denote a psychological phenomenon defining how people make sense out of their experience in the world (DERVIN, 1983). On the basis of this definition, Klein, Moon and Hoffman (2006) discuss that sensemaking is not a reinvention of the wheel of the concepts creativity, curiosity, comprehension, mental modeling, and situation awareness, but it is more than that; concluding that sensemaking is a motivated continuous effort to understand connections in order to anticipate their trajectories and act effectively.

In HCI perspective, sensemaking refers to the iterative process of building up a representation of an information space that is useful for achieving the user's goal (RUSSELL et al., 1993). Pirolli and Card (2005) identified a sensemaking model (Figure 2.1) composed of different stages and the different ways to proceed from one stage to another. These stages are grouped in *information foraging loop and sensemaking loop*. In the information foraging, the subtasks of searching, collecting, filtering and preparing are involved. In the sensemaking, the information is analyzed, hypothesis are built and tested based on the previously collected data, conclusions are derived from the information, before it is finally exploited to according action.

Figure 2.1: Sensemaking model



Source: Pirolli and Card (2005)

Good data sensemaking tools support statistical calculations, using the strength of computers to perform those calculations quickly and accurately, and interactive visualizations, making it possible to find and understand the meaningful patterns in our data.

2.4 User Interface for Search

User interface design is a practice whose techniques are encompassed by the field of HCI. On the other hand, searching involves a range of tactics and techniques, rather than simply submitting a query and seeing a list of matching results. As part of special issue on *exploratory search*, Marchionini (2006) identified a series of strategies that users may often need to employ to achieve their goals, such as comparing, synthesizing and evaluating. It is plain to see that a query interface needs to provide more than a simple keyword search form, or query by example to support users in applying such strategies. Thus, the search session is a cycle of query specification, inspection of retrieval results, and query reformulation in the field of information retrieval. In our case the search session will be composed of two components of the visual query system *visual query definition* and *visual result set presentation*.

A set of guidelines for user interface has been identified in section 2.1 within the interaction design process. But, how we put these guidelines into search interfaces. Search in information retrieval is a text search that supports keyword, boolean operators and command-based syntax. Thus, we have to keep user *staying in the flow* while searching in order to improve user interaction. We describe below what are interfaces of flow.

2.4.1 Interfaces of Flow

The theory of flow was developed by the psychologist Csikszentmihalyi, as described in his book, Flow: The Psychology of Optimal Experience (CSIKSZENTMIHALYI, 2009). The essence of Csikszentmihalyi's notion is that a person who experiences flow is completely absorbed by an activity for the pleasure that it provides, and all other stimulation and activities are imperceptible to that person. He describes how he and many other researchers around the world applied the *experience sampling method* to try, understand and characterize this elusive human experience. Csikszentmihalyi's theory of flow is defined by seven characteristics, from which Bederson (2004) focuses on five of them along with interfaces that exemplify those characteristics:

- Challenge and require skill: If users have to re-figure something out, or fight with an unstable feature regularly, they won't be able to get to the point of concentrating on the task. Also, it is important to balance between needs of novice and experts.

- Concentrate and avoid interruption: the interface must be able to focus user's attention at length on the task at hand.

- Maintain control: user must be able to maintain the control over the activity. More, the interface should be adaptable to user needs.
- Speed and feedback: user must receive quick feedback in response to their actions.
- Transformation of time: The user's perception changes when they are in the flow. These changes are due to the difficulty level of the task. Thus, this offers a direction to a possible metric for understanding flow.

A set of properties that a interface with flow should have is described in Hearst (2009). Those properties are:

- Inviting
- Support interrupt-free engagement in the task.
  - No blockages
  - Easy reversal of actions
  - Next steps seem to suggest themselves.

2.5 Visual Query Systems

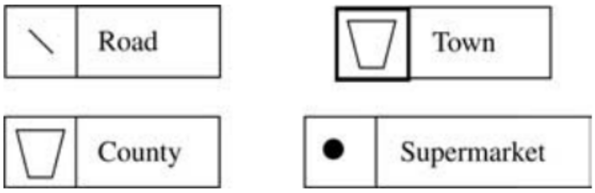
Visual Query Systems (VQS) (CATARCI et al., 1997) use a visual representation that is effective to express different kinds of knowledge in order to depict a domain of interest in expressing related queries. According to Catarci et al. (1997), the goal of people working with VQS is to retrieve the aimed data. Indeed, two main activities are:

- Understanding the reality of interest: consists in the accurate definition of the fragments of schema involved in the query. One technique is the browsing. This browsing may be specialized into an intentional and extensional case. In the intentional case, it is performed on the schema(metadata) of the database. In the extensional one, it is performed on the data itself.
- Formulating the Query: consists in formally express the operands involved in the query, with their related operators. There are different strategies for formulating a query. They are schema navigation, sub-queries, matching and range selection.

The visual representations (CATARCI et al., 1997) are:

- Form-Based: it facilitates non expert users by capitalizing on the natural tendency of people to use regular structures and organize data into tables.

Figure 2.2: Visual Representation of Spatial Objects



Source: Morris et al. (2004)

- Diagram-Based: it uses visual components that have one to one correspondence with specific concept types. It offers the visualization of relationships between concepts. The query is done through selection of visual elements, the traversal on adjacent elements, and the creation of a bridge between disconnected elements.

- Icon-Based: it uses set of icons which denote both the entities of real world and the available functions of the system. The query is expressed by combining icons according to some spatial syntax. This type is affordable for users who are not familiar with the concepts of the data model.

- Hybrid: it uses an arbitrary combination of the above three visual formalisms.

When developing VQS it should be emphasized in the identification of the users and their needs, using it for developing a suitable interface and validated it with usability evaluation. Furthermore, the cycle session of visual query systems is composed of visual query definition and analysis of visual result presentation, described below.

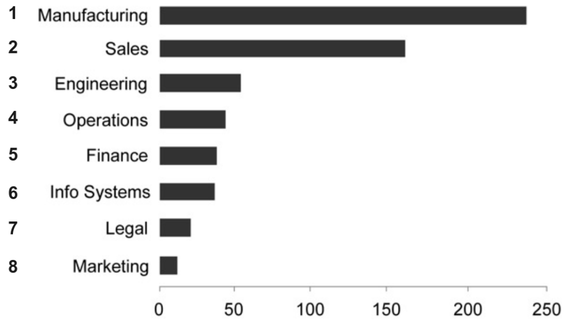
2.5.1 Visual Query Definition

Visual query definition associates query criteria in a display (TUNNING, 2005). Also it is called visual query formulation or specification. By means, the user needs to be able to specify the query visually and interactively when queries tend to get large or complex, and iterative procedures are applied to refine it. Here visual representations of the query structure provide overview and help users to maintain the stay of flow in the interface. For example, visual query definition for a spatial database, (MORRIS et al., 2004) define visual representations for spatial objects (see Figure 2.2), which will be used in the interface to formulate the query.

2.5.2 Visual Result Set Presentation

The visualization of large amounts of data needs an appropriate presentation in order to make sense to the user. Depending on the data type, a broad variety of information visualization techniques is and has been mentioned before in section 2.2. PetroQuery® data result relies

Figure 2.3: Ranking graph



Source: Few (2013)

mostly on tabular data displays (see Figure 2.8). Tables are the best approach if you need to look up individual values, compare a single value to another, or know values precisely, but they do not display patterns or trends. This is a problem, because geologists try to find patterns and trend in the data. Therefore, it is important to provide a visual thinking of the data in order to enable meaningful patterns. One way is through graphs. Properly designed graphs can make user hold much more information in memory. For example, if user needs to remember information in the table, user could hold only about four of the values (that is, four of the monthly sales numbers) in working memory at any one time. But by relying on the graph, twelve values are combined into each of the four lines to form a pattern that user could hold entirely as a single chunk in working memory. Few (2013) presents seven common quantitative relationship graphs, from which we are interested in the following ones:

- A ranking graph shows the sequence of a series of categorical subdivisions, based on the measures associated with them. For example, the sequence of minerals from top to down that occurs in a sample description (see Figure 2.3).

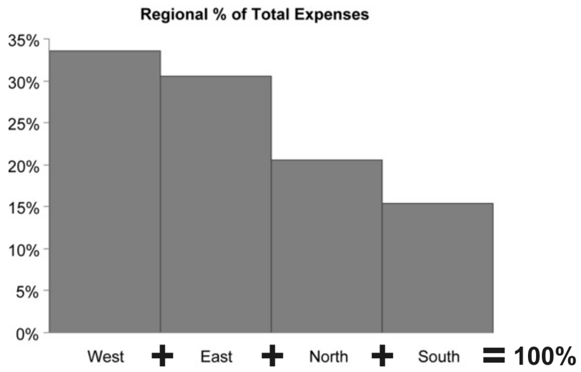
- A part-to-whole graph shows how the measures associated with the individual categorical subdivisions of a full set relate to the whole and to one another. For example, this graph will help user to get sense of the composition of a rock (see Figure 2.4).

- A correlation graph shows whether two paired sets of measures vary in relation to one another, and if so, in which direction (positive or negative) and to what degree (strong or weak). For example, the percentage comparison of two minerals in a well (see Figure 2.5).

- A distribution graph shows the number of times something occurs across consecutive intervals of a larger quantitative range. For example, the distribution of a mineral in certain sample descriptions from a Basin (see Figure 2.6)

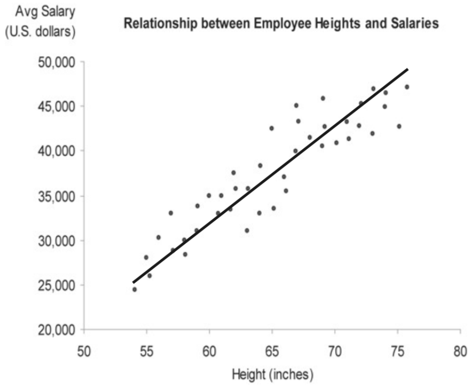
- A geospatial display shows the locations of values, which is useful when geography is relevant to the story that you are telling. It is important this display for geologists because

Figure 2.4: Part-whole graph



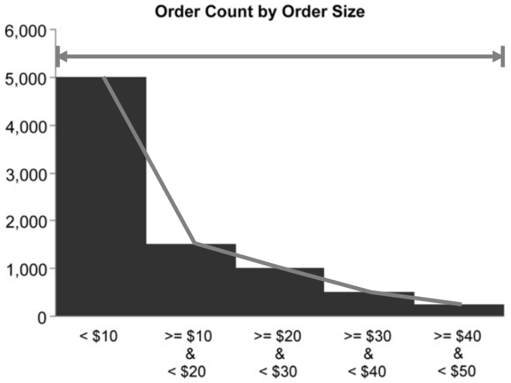
Source: Few (2013)

Figure 2.5: Correlation graph



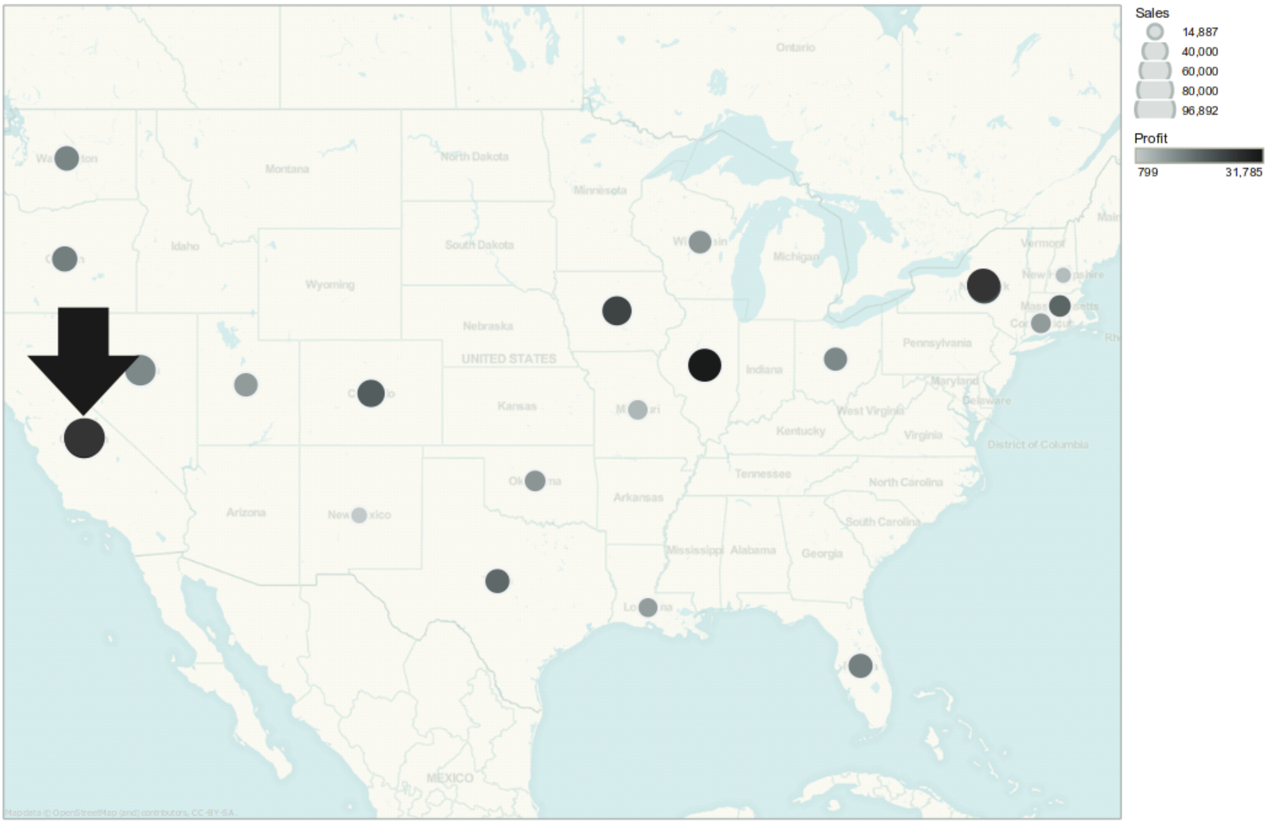
Source: Few (2013)

Figure 2.6: Distribution graph



Source: Few (2013)

Figure 2.7: Geospatial graph



Source: Few (2013)

they recognize immediately map visualizations (see Figure 2.7).

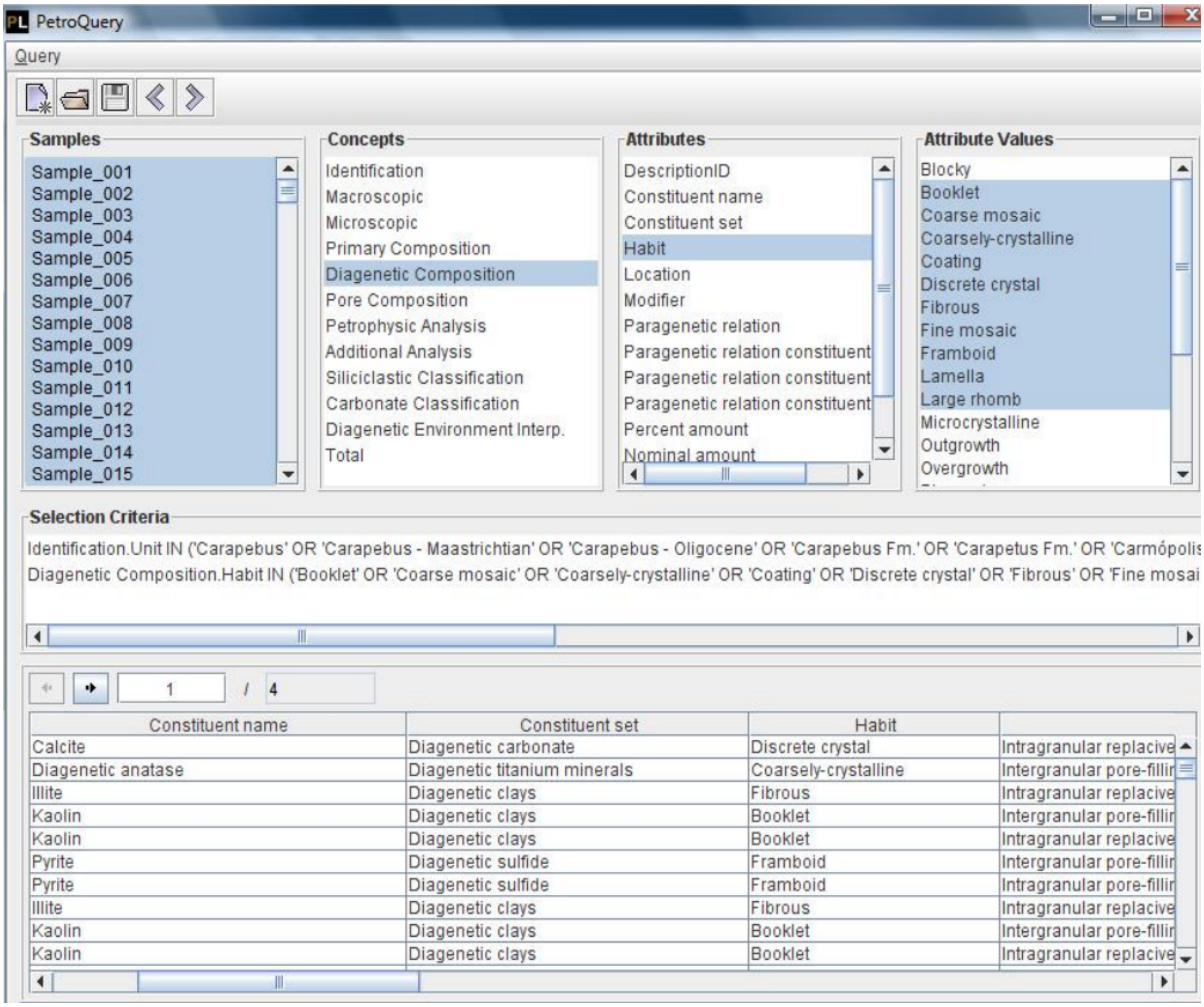
2.6 Four Ontology-based VQS

In the following, we describe four ontology-based visual query systems. The first one is the motivation of our thesis and employs list-views. The second system use a tree view approach, which is used by the majority of systems. The third system use a graph representation to guide the consultation. The fourth system combine textual and graph representation. Our proposed system (described in chapter 7) uses the graph representation, but reducing the amount of information showing in the graph according to the user's community. Also, we focus in the result visualization providing other data visualizations.

2.6.1 PetroQuery®

PetroQuery® (CASTRO et al., 2005) is a commercially mature visual query system that supports multidimensional, user-defined queries over a petrographic data controlled by ontologies. PetroQuery® works on top of Petroledge database. Petroledge database is based in the

Figure 2.8: PetroQuery® Interface



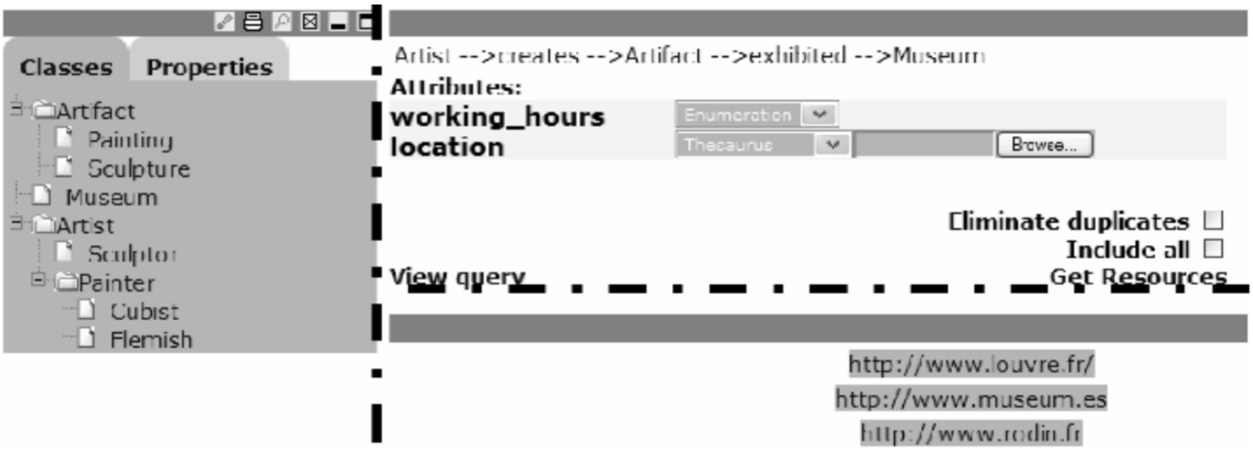
Identification.Unit IN ('Carapebus' OR 'Carapebus - Maastrichtian' OR 'Carapebus - Oligocene' OR 'Carapebus Fm.' OR 'Carapetus Fm.' OR 'Carmópolis
Diagenetic Composition.Habit IN ('Booklet' OR 'Coarse mosaic' OR 'Coarsely-crystalline' OR 'Coating' OR 'Discrete crystal' OR 'Fibrous' OR 'Fine mosai

Constituent name	Constituent set	Habit	
Calcite	Diagenetic carbonate	Discrete crystal	Intragranular replacive
Diagenetic anatase	Diagenetic titanium minerals	Coarsely-crystalline	Intergranular pore-fillir
Illite	Diagenetic clays	Fibrous	Intragranular replacive
Kaolin	Diagenetic clays	Booklet	Intergranular pore-fillir
Kaolin	Diagenetic clays	Booklet	Intragranular replacive
Pyrite	Diagenetic sulfide	Framboid	Intergranular pore-fillir
Pyrite	Diagenetic sulfide	Framboid	Intragranular pore-fillir
Illite	Diagenetic clays	Fibrous	Intragranular replacive
Kaolin	Diagenetic clays	Booklet	Intergranular pore-fillir
Kaolin	Diagenetic clays	Booklet	Intragranular replacive

Source: Castro et al. (2005)

knowledge model of Abel (2001) that consists of a petrographic ontology, which has been implemented originally using a frame representation format, and other abstract concepts that give support to the petrographic description, which is the extensional part. In Silva (2001), it is described part of the structure of the database of the system focusing in the rock interpretation. PetroQuery® GUI consists of three sections (see Figure 2.8). In the first section there are list boxes for the concept, attribute and value selection. In the second section, the query is shown in a structured query language (SQL) like form in a list box. The last section contains the results which are represented in a table. The process of query formulation starts selecting one or multiple samples, and then it is selected the concept, attribute or value that the sample can contain. Through the process of selecting concepts, a textual query appears in the second section. Each line is a SQL like query. The user has the ability to delete a line of the query. At the same time, that query line text is added to the list box, the result appears in the table. Also, the user can visualize in a ternary plot the results in another window.

Figure 2.9: GRQL Interface



Source: Athanasis, Christophides and Kotzinos (2004)

2.6.2 Graphical RQL

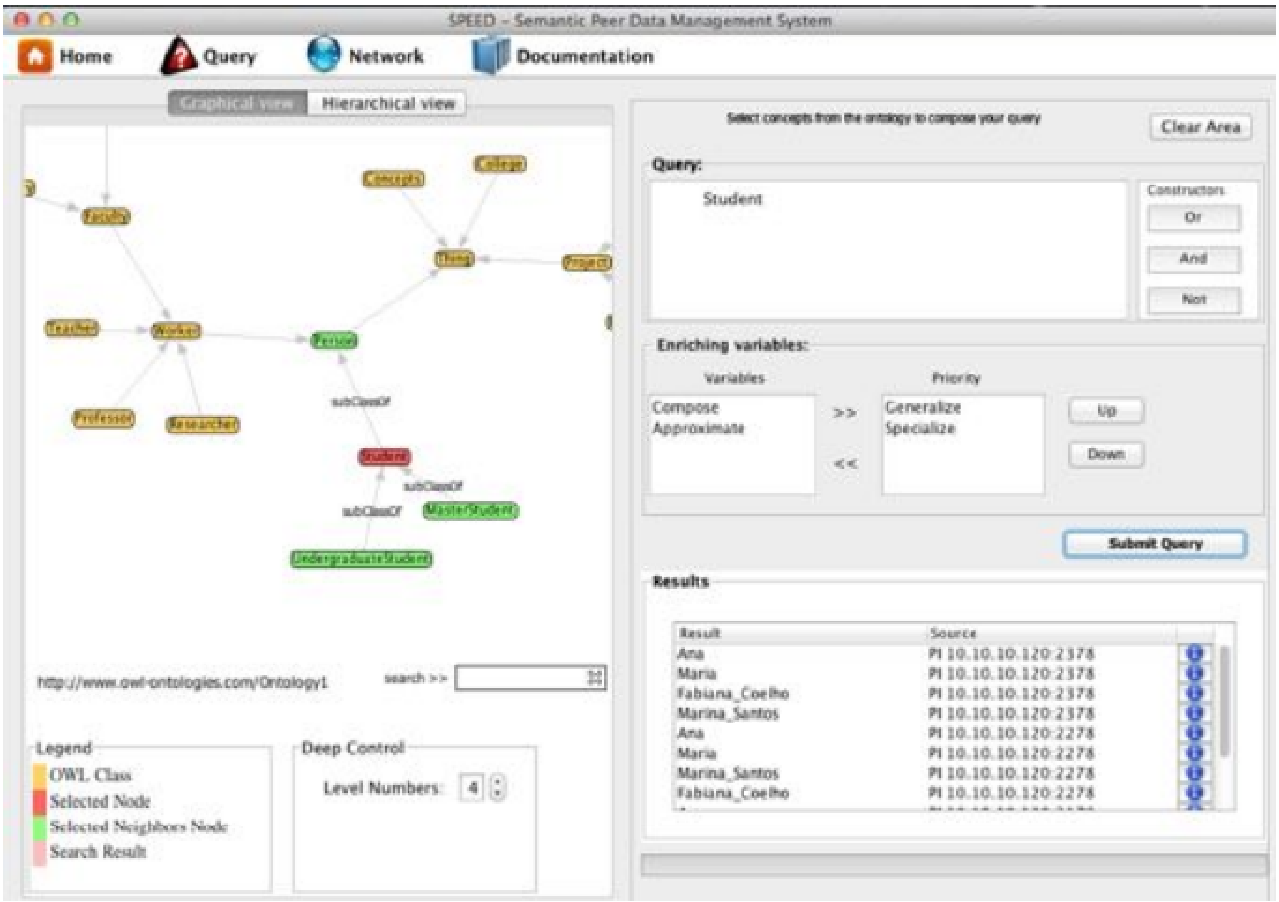
Graphical RQL (ATHANASIS; CHRISTOPHIDES; KOTZINOS, 2004) is a user-friendly GUI for browsing and filtering RDFs description bases. The GRQL GUI consists of three basic interaction areas. The left area provides a tree-shaped display of the subsumption hierarchies of both the classes and properties defined in an RDF schema. The right upper area of the GRQL GUI allows users to explore progressively the individual RDF class and property definitions and generate navigational and/or filtering RQL queries. Finally, the right lower area visualizes the constructed query/view results. A snapshot of the GRQL GUI is illustrated in Figure 2.9. The process of query formulation starts in the left area. Users select a node in the tree display and can access their subclasses and sub-properties by expanding the tree node. After selecting the concept or property, its complete definition is shown in the right upper area and user can perform operations over the instance level.

2.6.3 VisualSPEED

VisualSPEED (see Figure 2.10) is a visual query interface that provides a natural visual query interface, and supports automatic query generation (ALENCAR; SALGADO, 2013). The system is structured mainly by the user interaction layer and the management layer. The user interaction layer consists of four modules:

- View Ontology: responsible for ontology visualization.
- Form Query: responsible for formulating queries that are sent to the query module.
- View Results: responsible for organizing and displaying the results of queries.
- View network: responsible for displaying network topology.

Figure 2.10: VisualSPEED Interface



http://www.owl-ontologies.com/Ontology1

Level Numbers: 4

Query: Student

Variables	Priority
Compose	Generalize
Approximate	Specialize

Result	Source
Ana	PI 10.10.10.120:2378
Maria	PI 10.10.10.120:2378
Fabiana_Coelho	PI 10.10.10.120:2378
Marina_Santos	PI 10.10.10.120:2378
Ana	PI 10.10.10.120:2278
Maria	PI 10.10.10.120:2278
Marina_Santos	PI 10.10.10.120:2278
Fabiana_Coelho	PI 10.10.10.120:2278

Source: Alencar and Salgado (2013)

The management layer is composed by two modules responsible for the communication between the User Interaction and the SPEED's core layers:

- Query Manager: it performs the integration of query results translating them into a format comprehensible by the View Results module.

- Communication Manager: responsible for communicating the User Interaction Layer with SPEED core.

The process of query formulation is done through the selection of concepts in the graphical representation of the ontology. The selected concepts will be showed on the query composition field, which is in the form query module. In this area, constructors as OR, AND or NOT can be used to compose the query. The query submitted by the user is interpreted by the system and translated to a SPARQL command to be executed in data. Users can customize the query by selecting and prioritizing enriching variables (approximation, subconcept, superconcept and aggregation). These variables represent semantic relationships between the concepts of the query represented by the generated semantic correspondences. The enrichment of the query is shown in the bottom right side of Figure 2.10. The system displays the results organized in a table shown in the bottom right side of Figure 2.10. Users can also see a visualization of the data in another window. This VQS combines a graph representation for performing their queries.

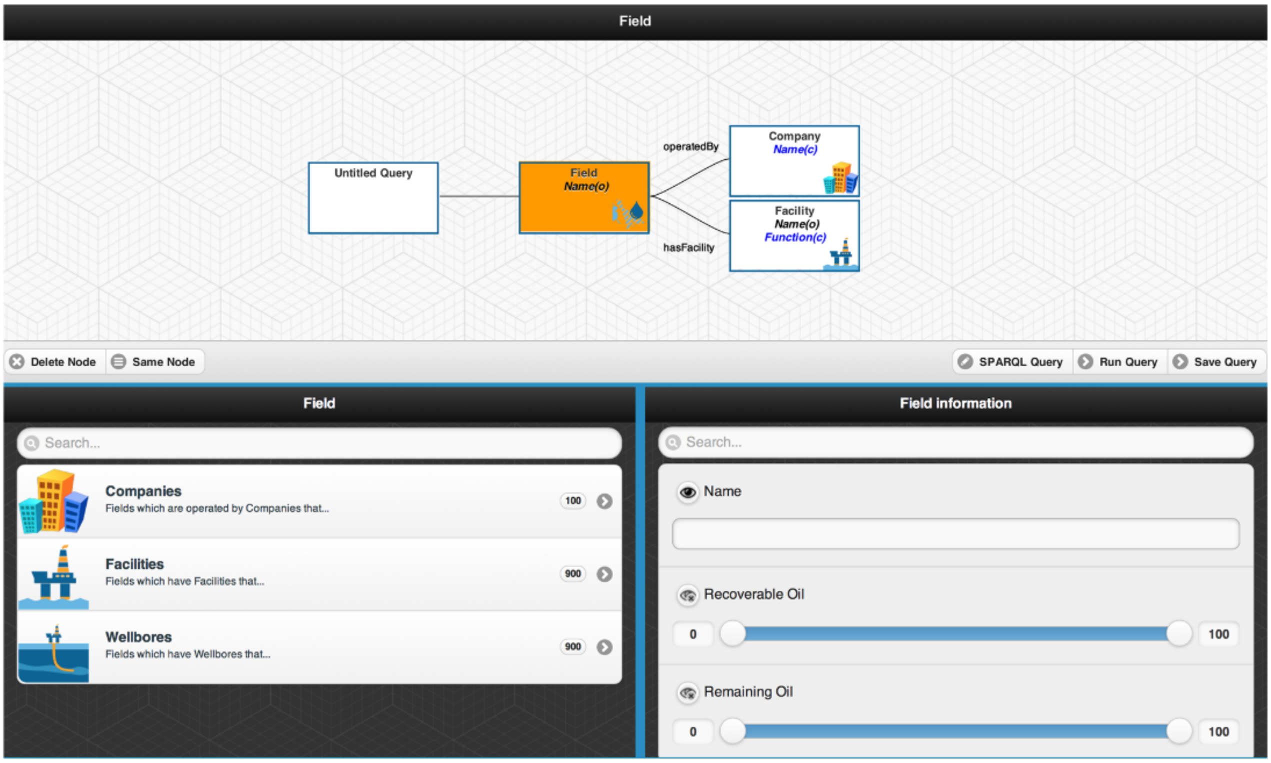
2.6.4 OptiqueVQS

OptiqueVQS (SOYLU et al., 2013) is an ontology-based visual query system for the Optique Scalable End-user Access to Big Data project[2]. It relies on an ontology-based data access (OBDA) framework (KOGALOVSKY, 2012), which is not part of the scope of our thesis, that allows access to relational data over ontologies. OptiqueVQS is designed as a user-interface (UI) mashup built on widgets. According to authors, widgets are the building blocks of their VQS and refer to portable, self-contained, full-edged, and mostly client side applications with limited functionality and complexity. They have three widgets depicted in Figure 2.11. The first widget (W1 - see the bottom-left part of Figure 2.11) is a menu-based query by navigation widget and allows users to navigate concepts through pursuing relationships between them, hence joining relations in a database. The second widget (W2 - see the bottom-right part of Figure 3) is a form-based widget, which presents the attributes of a selected concept for selection and projection operations. The third widget (W3 - see the top part of Figure 2.11) is a diagram-based widget and provides an overview of the constructed query and affordances for manipulation.

The process of query formulation is described as follows: a user first selects a kernel concept, i.e., the starting concept, from W1, which initially lists all domain concepts accompanied

[2] http://www.optique-project.eu/

Figure 2.11: Optique Query Interface



Companies
Fields which are operated by Companies that... 100

Facilities
Fields which have Facilities that... 900

Wellbores
Fields which have Wellbores that... 900

Name

Recoverable Oil 0 100

Remaining Oil 0 100

Source:Soylu et al. (2013)

with icons, descriptions, and the potential/approximate number of results. The selected concept becomes the focus/pivot concept (i.e., the node coloured in orange or highlighted), appears on the graph (i.e., W3) as a variable-node , W2 displays its attributes, and W1 displays all concept-relationship pairs pertaining to this concept. The user can select attributes to be included in the result list (i.e., using the *eye* button) and/or impose constraints on them through form elements (i.e., W2). Currently, the attributes selected for output appear on the corresponding variable-node in black with a letter *o*, while constrained attributes appear in blue with letter *c*. The user can select any available option from the list, which results in a join between two variable-nodes over the specified relationship and moves focus to the selected concept (i.e., pivot). The user has to follow the same steps to involve new concepts in the query and can always jump to a specific part of the query by clicking on the corresponding variable-node. The arcs that connect variable-nodes do not have any direction, since for each active node only outgoing relationships, including inverse relationships, are presented for selection in W1; this allows queries to be always read from left to right. The user can also switch to SPARQL mode and see the textual from of the query by clicking on *SPARQL Query* button at the bottom-right part of the W3 as depicted in Figure 2.11. The user can keep interacting with the system in textual form and continue to formulation process by interacting with the widgets. For this purpose, pivot/focus node is highlighted and every variable-node is made clickable to allow users to change focus. Currently, the textual SPARQL query is non-editable and is for didactical purposes, so that advanced end-users, who are eager to learn the textual query language, could switch between two modes and see the new query fragments added after each interaction.

OptiqueVQS presents in its interaction design the importance of using search filters because it lets the user find easily the desired term. In our interaction design, we also considered the search filters for searching concepts or values of the concepts.

2.7 Summary

In this introductory chapter, we provided the background used in the practical side of our thesis. Also, we described four ontology-based visual query systems. Furthermore, we follow the interaction design process in the implementation of RockQuery system prototype. We used the research methods and techniques in the preliminary PetroQuery® study described in appendix A, which gave us the basis for *understanding and establishing the requirements*. Then, the *design alternatives* are presented in appendix B. Finally, *prototyping and evaluating* are discussed in chapter 7 and in subsection 8.2.

3 ONTOLOGY, UFO AND ONTOLOGY VISUALIZATION

This chapter explains the basic notions of ontology, foundational ontology and ontology visualization. First, we start defining what is an ontology. Then, we describe the Unified Foundational Ontology (UFO), which is used in our ontology view extraction approach. Finally, we present some ontology visualization works.

3.1 Ontology

An ontology is a collection of concepts and relationships among them organized in a special structure. Some authors misused the term ontology as a taxonomy of concepts or names without specification of formal relations between classes. In this section, we describe the definition of ontology and types of ontologies and provide the notion of foundational ontology.

3.1.1 Definition

Ontology begins with the field of philosophy with the classical study of being which remotes to the Greek philosopher Aristotle BC. Aristotle's description *the study of being qua being* involves three things: (1) a study, (2) a subject matter (being), and (3) a manner in which the subject matter is studied (qua being); which introduce the base to the science of metaphysics of first philosophy. So in a philosophical discipline, ontology is characterized by being singular, perspective- and domain independent- and oriented towards making strong claims about the world (ORSTROM; ANDERSEN; SCHARFE, 2005). However, the term *ontology* was itself coined in 1613 by two philosophers, Rudolf Gockel (Goclenius), in his Lexicon philosophicum and Jacob Lorhard (Lorhardus), in his Theatrum philosophicumm (SMITH; WELTY, 2001).

The importance of ontology in computer science has grown in the last decade gaining a specific role in Artificial Intelligence, Computational Linguistics, and Database theory. Thus, ontology in computer science has begun with the definition of Neches et al. (1991), who stated that ontology establishes the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary. Later on, Gruber (1993) defined as an explicit specification of a conceptualization. Based on Gruber's definition, Borst (1997) defined as a formal specification of a shared conceptualization. After that, Studer's definition merges Borst and Gruber defining ontology as a formal and explicit specification of a shared conceptualization (STUDER; BENJAMINS; FENSEL, 1998). However, we adopt Guarino's definition (GUARINO, 1998) who considers ontology as *a logical theory accounting for the intending meaning of a formal vocabulary. The intended models of a logical language using such vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models.*

There exist many classifications of ontology that Gomez-Perez, Fernandez-Lopez and Corcho (2007) present, but in general they can be categorized based on the formalness of the knowledge captured. According to Baader et al. (2003) there are top-level ontology, domain ontology and application ontology which are similar to Guarino's classification (1997), excepted for the task ontology that is not included. Thus, Guarino (1997) distinguishes four types of ontologies which are:

- *Top Level Ontologies*- describe general concepts that are independent of a domain.
- *Domain Ontologies*- describe vocabulary related to a generic domain.
- *Task Ontologies*- describe vocabulary related to a generic task.
- *Application Ontologies*- describe concepts depending of a particular domain or task. These concepts often correspond to roles played by domain entities while performing certain activity.

Furthermore, domain specific ontology includes among other terminologies, glossaries, thesauri and nomenclatures which are associated to specific domains.

It is important to mention that for the semantic web community some definitions are different. A *Concept* can be defined in a variety of ways, potentially providing a lot of additional information about itself, and its relationships (and topological proximity) to other elements. Through an inheritance structure concepts can be made specializations and generalizations of other concepts. The conventions used for a *concept* are exactly the same as what is labeled a *class* in the ontology web language (OWL)[1]. Also, both *attributes* and *relationships* are regarded as properties that belong to a concept.

3.2 Foundational Ontologies and UFO

Foundational Ontologies are theoretically well-founded domain independent systems of categories that can be used to develop models of specific domains (GUIZZARDI, 2005). Being domain independent well-grounded formal theories, they can serve as a foundation for analyzing domain specific concepts, providing guides to make modeling decisions in the conceptual modeling process, clarifying and justifying the meaning of the models, expliciting the ontological commitments that underlie the ontologies, improving the understandability and reusability. In this work, we adopt a foundational ontology called *Unified Foundational Ontology* (UFO). This ontology provides a set of categories that account for the ontological distinctions underlying language and cognition, and that are empirically supported by investigations in cognitive sciences.

UFO is an ontology of *particulars* and *universals*. Roughly speaking, the distinction between particular (or individual) and universal is analogous to the distinction between *types*

[1] http://www.w3.org/TR/2004/REC-owl-guide-20040210/

(*class* or *classifier*) and their instances, in conceptual modeling (GUIZZARDI, 2005). Thus, UFO provides a set of categories of particulars and a set of categories of universals. The categories of universals can be viewed as meta-types, since they can be understood as *types of types*. These meta-types are characterized according to meta-properties and classify concepts in specific (domain) ontologies. In this sense, we can view the concepts in specific domain ontologies as instances of the meta-types provided by UFO. Following, we will summarize the main UFO features that we will apply in this work. A full description of UFO can be found in Guizzardi (2005).

The meta-types of UFO are organized in a taxonomy according to some ontological meta-properties, such as *identity*, *rigidity*, *existential dependency*, *relational dependency* and so on. Thus, UFO includes the principles of the the well-known OntoClean methodology (GUARINO; WELTY, 2004), which allows (a) the analysis of the concepts in domain ontologies according to philosophically well founded meta-properties and (b) the subsequent meta-classification of these concepts. The meta-properties and meta-types of UFO are very similar to the ones provided by OntoClean. Indeed, UFO can be viewed as an integration of several aspects of Ontoclean (GUARINO; WELTY, 2004), DOLCE(GANGEMI et al., 2002), GFO(HERRE, 2010) and GOL(DEGEN et al., 2001), covering some problematic issues that were not covered in a satisfactory manner by existing foundational ontologies.

In this work we propose using meta-properties (and meta-types as well) for guiding the sub-ontology extraction algorithm. Due to this, an *ontology of universals* (that provides meta-types) is needed instead of an *ontology of particulars* (such as DOLCE). Thus, we have adopted UFO because it extends the framework of OntoClean and provides a strongly formalization for its meta-types and meta-properties.

As summarized in Carbonera (2012), the most generic UFO concept is Thing, which is specialized in two fundamental entities: Urelement and Set. Urelement is an entity that is not a set. The first distinction that is made between the specializations of Urelement is the fundamental distinction between the categories of Individuals and Universals. Individuals are entities that exist in reality, such as a person, an apple, etc. Universals, in turn, are standard features that can be instantiated in a number of different individuals; it can be understood as high-level abstractions that characterize different classes of individuals. In general, for each of the specializations for Universals, UFO also provides a corresponding specialization for Individuals.

Initially, UFO makes a distinction between *Endurant Universal* and *Perdurant Universal* (or Event Universal) as shown in Figure 3.2. Instances of an Endurant Universal (such as Dog, Person, Country, etc) are individuals wholly present whenever they are present. On the other hand, instances of a Perdurant Universal (or Events), such as Game, War, etc, are individuals composed by temporal parts, that is, they happen in time, accumulating temporal parts.

Within the Endurant Universals, UFO defines *Substantial Universals*, whose instances are individuals that, in general, are *existentially independent* from all other individuals. Some of their instances can be *existentially dependent* when they are considered *inseparable parts* of their hosts. Some Substantial Universals are *Sortal Universals*, which have some *principle of identity* (PI). In this context, a PI supports the judgment whether two instances of the universal are the same.

At this point, it is important to introduce the notion of *rigidity*. A certain universal is rigid when its extension is the same in all possible worlds. That is, an instance of a rigid universal cannot cease to be an instance of it without ceasing to exist. This notion can be clarified considering the distinction between *Person* and *Student*. Person can be viewed as a *rigid universal*, since any person cannot cease to be a person without ceasing to exist; meanwhile all instances of Student (which is an *anti-rigid universal*) can still exist (as persons) if they cease to be students.

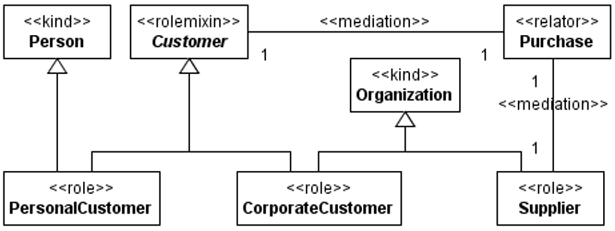
Within the sortal universals, UFO includes three distinct types of *substance sortals*, which are *rigid sortals that provide their own principle of identity*: *Kind*, which represents complexes integral wholes (Person, Dog, Chair, etc); *Collective* (Swarm, Forest, etc), which represents collectives; and *Quantity*, which represents objectified portions of matter (Wine, Water, Gold, etc). Besides that, *Subkind* is a *rigid sortal* that does not provide its own PI, but *carries* a principle of identity which is supplied by a given substance sortal.

UFO also defines two *anti-rigid sortals*: *Roles* and *Phases*. Phases are universals that constitute possible stages in the history of a substance sortal. Phases are *relationally independent*, since they depend solely on intrinsic properties. For example, *Caterpillar* and *Butterfly* are considered phases of *Lepidopteran*; as well as *Baby*, *Toddler*, *Kid*, *Teenager* and *Adult* are considered phases of *Human*. On the other hand, Roles are *relationally dependent*, since they depend on extrinsic (relational) properties. This is the case, for example, when we say that for an instance of person to be considered a student, she must be enrolled at an educational institution.

Other substantial universals do not have the properties of sortals; they are *dispersive universals*. This is the case, for example, of *Category*, which is a rigid universal that does not have a PI. Categories represent essential properties that are common to all instances of many disjoint universals that provide distinct PIs. Rational agent is an example of Category, since it abstracts an essential property (namely, the rationality) of instances of Person and Artificial Agent, which are disjoint universals, with distinct PIs. *Role Mixin*, on the other hand, is an anti-rigid universal that does not have a PI. It can be viewed as a generalization of roles of different substance sortals. For example, Customer is a role mixin that generalizes Personal Customer, which is a role of Person; and Corporate Customer, which is a role of Organization. Finally, *Mixin* is a universal that does not have a PI and that is *semi-rigid*; that is, it has some instances that are *necessarily* its instances, but it also has some instances that are only *contingently* its instances. It usually generalizes rigid and anti-rigid universals. For example, *Seatable Object* is a mixin that generalizes *Chair*, which is a rigid universal; and *Solid Crate*, which is an anti-rigid universal (actually, it is a phase of a *Crate*, which can also be a *Broken Crate*).

On the other hand, *Moment Universals* are Endurant Universals whose instances are *existentially dependent* individuals that *inheres* in other individuals. Some moment universals depend

Figure 3.1: Diagram representing a modeling scenario using kinds, roles, relators and role mixins.



Source: The authors

existentially on a single entity. This is the case of *Quality Universals* and *Modes*. *Quality Universals* represent the *properties* in the conceptual models. A Quality Universal characterizes other Universals and is related to *Quality Structures*, that is, a structure that represents a set of all values that a quality can assume. Thus, considering the property color as a Quality Universal, a given instance of *Car* could be characterized by an instance of quality *Color*, which is associated with a value (called quale) in the *ColorStructure*, which represents all the possible values that the property color can assume. On the other hand, *Modes* are universals whose instances are *existentially dependent* individuals, and that are not associated to *Quality Structures*. Examples of modes are *Skill*, *Belief*, *Headache*, etc. Both *Quality universals* and *Modes* are related to the entities that they characterize through a relation of *characterization*. Besides that, *Relators* are moments that depend existentially on two or more entities. Examples of relators are *enrollment*, *contract*, etc. Relators are related to entities that it relates through a relation of *mediation*. The relators also represent the relational dependency of roles and role mixins. Due to this, roles and role mixins must be related to some relator, through a relation of *mediation*. Figure 3.1 represents a modeling case using relators, roles and role mixins.

UFO proposes four types of parthood relations, clarifying its semantics: *componentOf*, *memberOf*, *subCollectionOf* and *subQuantityOf*. Each parthood relation can only be established between individuals of specific UFO meta-types, respecting some ontological constraints embedded in UFO. These relations can be characterized by five meronymic meta-properties that indicate: *essential part*, *inseparable part*, *immutable part*, *immutable whole* and *shareable part*.

As important as the characterization of the meta-properties and meta-types, UFO also provides some postulates that a model must follows:

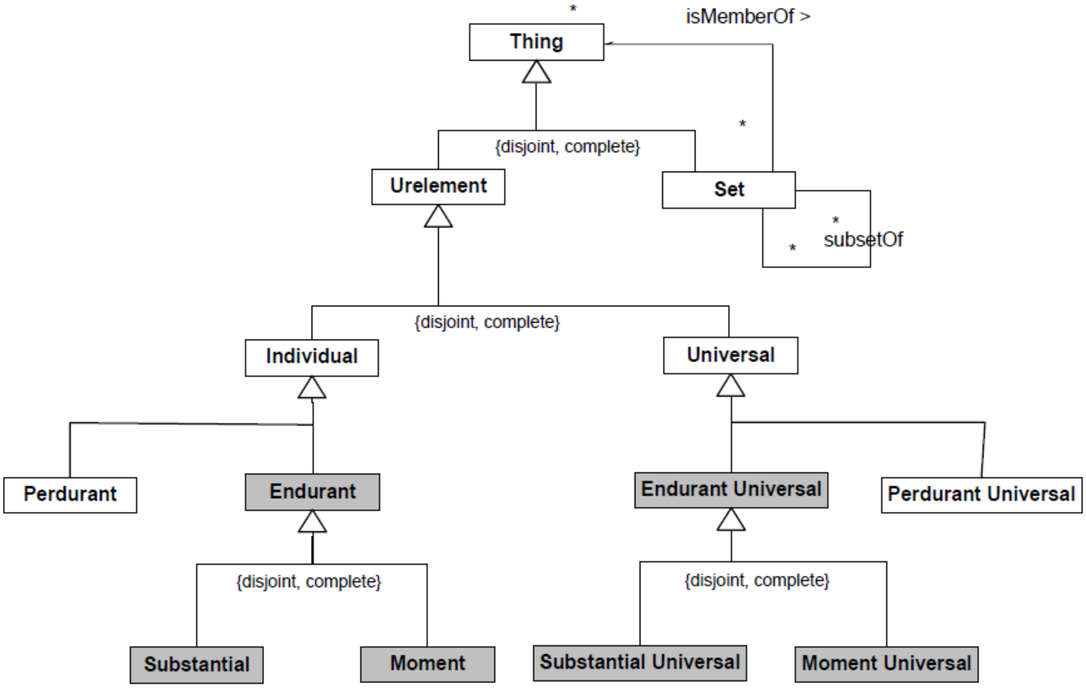
- **Postulate 1:** Every individual in a conceptual model of the domain must be an instance of a *sortal*.
- **Postulate 2:** An individual represented in a conceptual model of the domain must instantiate *exactly one* ultimate *Substance Sortal* (*kind*, *quantity* or *collective*).
- **Postulate 3:** A rigid universal cannot specialize (restrict) an anti-rigid one.

- **Postulate 4:** A dispersive universal cannot specialize a Sortal.

Furthermore, it is important to notice that every sortal that does not provide its own principle of identity (Role, Phase and SubKind) must be subsumed by exactly one concept that provides its own identity (one of the Substance Sortals). We use the described meta-types to orient the selection of concepts in the ontology in order to guarantee that the concepts that are inserted in the view preserve their integral meaning.

Besides the UFO itself, in Guizzardi (2005) it is proposed a conceptual modeling and ontology representation language called OntoUML. OntoUML is a redesign of the Unified Modeling Language (UML) meta-model, assuming the ontological distinctions and formal constraints prescribed by UFO. Thus, it can be used for representing reference ontologies according to UFO distinctions. According to Benevides et al. (2009), OntoUML is a modeling conceptual language because it offers meta constructs that represent the ontological distinctions described in UFO, and facilitates the ontology engineer the model conceptualization. In addition, the language was created adapting the basics of UML 2.0. and increasing the formal constraints that guarantee that the language will just accept models that satisfy the axiomatization established by the foundational ontology UFO. In our work, we developed a tool for ontology view extraction that uses as input ontologies represented as OntoUML models.

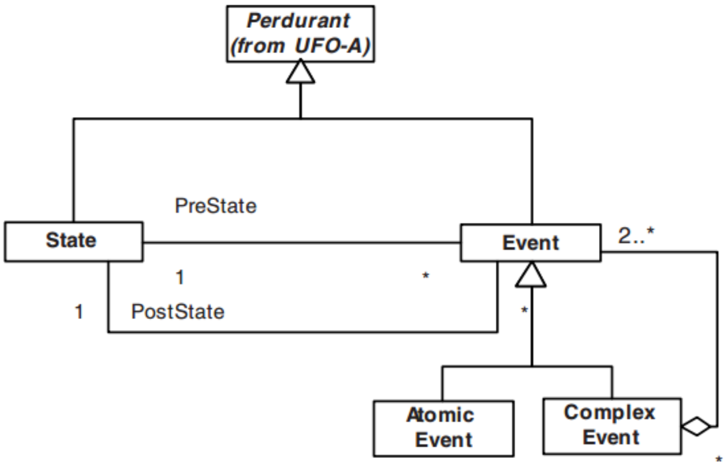
Figure 3.2: UFO Structure



Source:Guizzardi (2005)

In UFO-B, described in Guizzardi and Wagner (2008), the main focus is Event (Perdurant or Occurrent) which are possible changes from a portion of reality to another, i.e., they may trans-form reality by changing the state of affairs from one pre-state situation to a post-state situation. Events are existentially dependent on their participants in order to exist. Each participation is itself an event that can be atomic (with no improper parts) or complex (composed of at least two events that can themselves be atomic or complex), but that existentially depends on a single substantial. UFO-B is appreciated in the Figure 3.3.

Figure 3.3: UFO B



Source:Guizzardi (2005)

3.3 Ontology Visualization

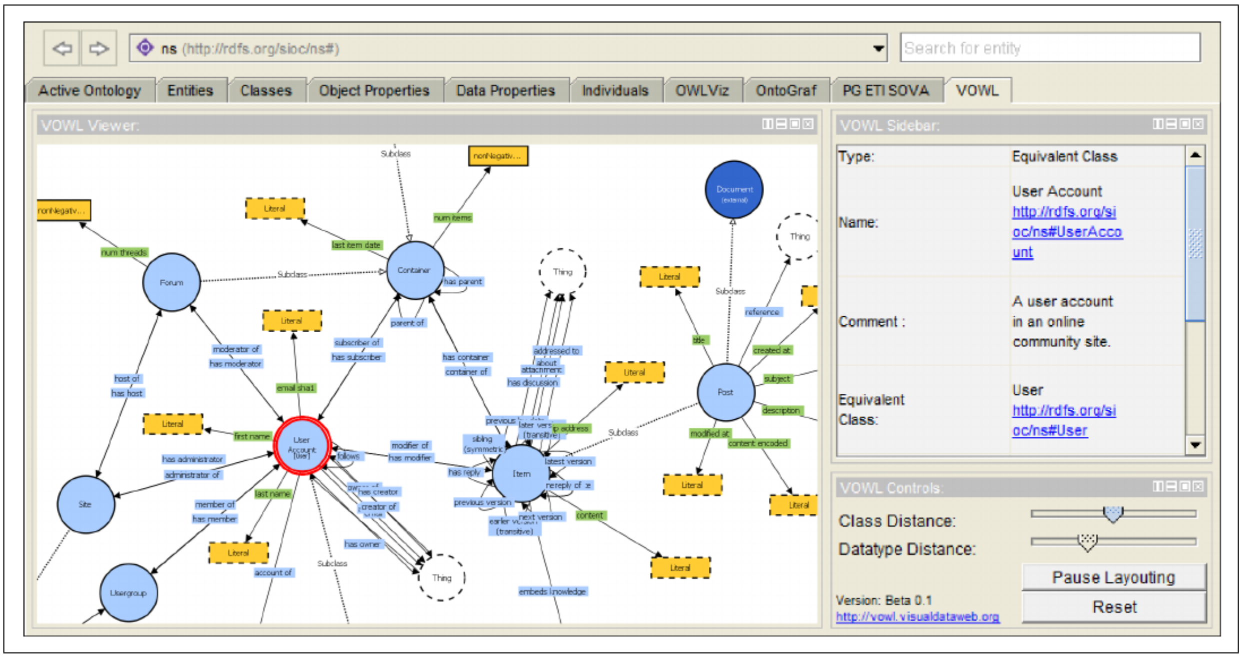
Visualizations are commonly used as a cognitive aid for presenting large ontologies and instance data. While several visualizations for ontologies have been developed in the last couple of years, they either focus on specific ontology aspects or are hard to read for non-expert users. The silver bullet would be an ontology visualization that is equally comprehensive and comprehensible. It must be printable, but also provides intuitive ways to interactively explore ontologies. Some of ontology visualizations are described below.

The ontology visualization should concatenate visualization techniques with customization operators such as pruning (BERCOVICI, 2008). These operators may help to show only the relevant, more focused parts of ontologies, rather than showing the entire graphs with potential thousands of nodes.

Protege VOWL, presented by Lohmann, Negru and Bol (2014), is an OWL plugin visualization module. In their work, it was presented a visual notation for OWL ontologies and were defined for many elements of OWL graphical depictions. These visual elements are based on only a handful graphical primitives forming the alphabet of the visual language: Classes are depicted as circles that are connected by lines and arrows representing the property relations, while property labels and datatypes are shown in rectangles. The visual elements are combined to a graph visualization representing the ontology and being arranged in a force-directed layout.

Some OWL elements are treated in a special way to increase the readability of the visualization. For instance, the predefined classes *owl:Thing* and *rdfs:Resource* usually do not carry domain information. They are multiplied and depicted in smaller size in order to give them less prominence in the visualization. Similarly, *rdfs:datatype* and *rdfs:literal* are shown multiple times so that datatype properties are arranged radially around the classes they are connected with. In addition, VOWL defines a color scheme for a better distinction of the different elements. The colors are defined in an abstract way leaving room for customization, but concrete colors and color codes are recommended by VOWL. VOWL viewer uses Prefuse for graph diagramming. Prefuse uses a physics simulation to generate the force-directed graph layout, consisting of three different forces: edges act as springs, while nodes repel each other and drag forces ensure that nodes settle (HEER; CARD; LANDAY, 2005). The forces are iteratively applied resulting in an animation that dynamically positions the nodes. The user can smoothly zoom in to analyze certain ontology parts in detail or zoom out to explore the global structure of the ontology. They can pan the background and move elements around, which results in a repositioning of the nodes by an animated adaptation of the force-directed layout.

Figure 3.4: Protege VOWL



Type: Equivalent Class

Name: User Account http://rdfs.org/sioc/ns#UserAccount

Comment : A user account in an online community site.

Equivalent Class: User http://rdfs.org/sioc/ns#User

Version: Beta 0.1
http://vowl.visualdataweb.org

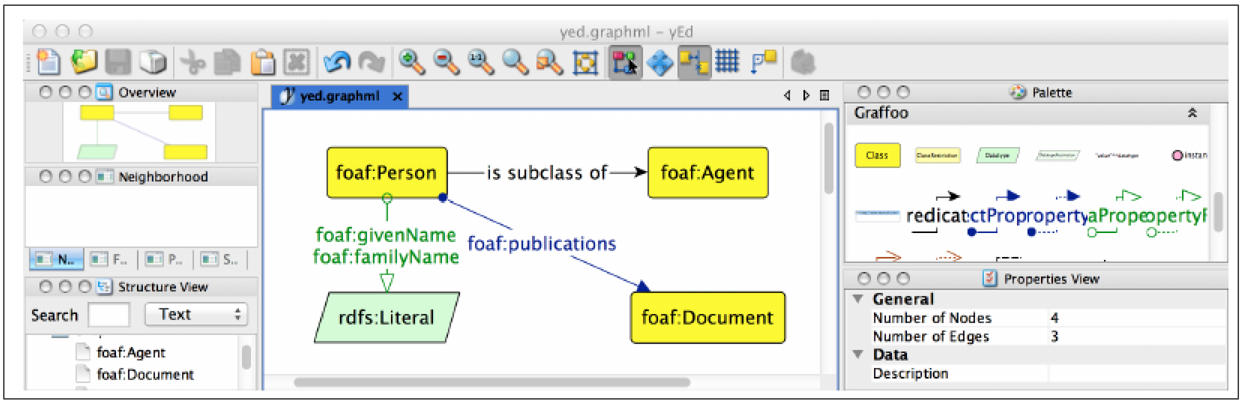
Source: Adapted from Lohmann, Negru and Bol (2014)

Grafoo (FALCO et al., 2014) is a graphical notation for OWL ontologies that uses the standard library yEd[2]. In Graffoo, there are two different kinds of graphical elements, blocks (or nodes) and arcs. Blocks are used to define classes and class restrictions (yellow rectangles

[2] Available at http://www.yworks.com/en/products yed about.html.

with solid and dotted borders respectively), datatypes and datatype restrictions (green rhomboids with solid and dotted borders respectively), individuals (pink circles with solid black border), ontologies (boxes with light-blue heading and dotted black border), additional axioms in Manchester Syntax for all those constructs that are not directly supported by a particular graphical element (light-blue and folded boxes), and rules (boxes with light-grey heading and black dashed border). Arcs are used to define assertions (black lines ending with a solid arrow), annotation properties (orange lines beginning with backslash and ending with a dashed arrow), data properties (green lines beginning with an empty circle and ending with an empty arrow), and object properties (blue lines beginning with a solid circle and ending with a solid arrow). In addition to these graphical elements, there is a particular kind of graphical element (named property facilities, i.e., arcs having dotted border and referring to data, object and annotation properties), that were studied to decrease the cognitive effort of users when understanding an ontology. For instance, they allow one to say explicitly that a certain property can be used in the context of two classes without declaring them as domain and range. They evaluate their tool based in learnability and usability.

Figure 3.5: Grafoo Tool



Number of Nodes 4
Number of Edges 3

Source:Falco et al. (2014)

GrOWL (KRIVOV et al., 2007) is another visualization model. It uses color, shading, and shape of nodes to encode properties of the basic language constructs. The objective of GrOWL was to make browsing ontologies more intuitive for non-technical users, limiting exposure to the complexities of DL.

Ontology navigation can be treated as a cognitive task. It is a complex process that involves the cognitive abilities that allow us to understand our environment, to plan actions and then to execute those actions (JUL, 2004). These cognitive abilities are: (a)information gathering, which refers to collect information about the environment such as where things are and how they are related spatially, (b) spatial knowledge preservation, which refers to encode and store spatial knowledge as well as recall and decode such information, (c) wayfinding, which refers to solve a spatial problem that involves determining where to go and how to go there, (d) locomotion, which refers to direct and control environment. A set of principles for providing cognitive support for navigating ontologies is described in d'Entremont and Storey (2009) and those principles are:

Provide Overviews: An overview of the whole ontology supports the information gathering process by enabling users to understand the scope of the ontology.

Provide Context: It allows users to see where they are within the structure and what other part exists. It is related with the cognitive ability spatial knowledge preservation.

Reduce complexity: It allows user to focus on smaller chunks of the ontology. It is related with the cognitive ability wayfinding process.

Indicate point of interest: It supports users in determining which areas within the ontology are worthy of further exploration. It is related with the cognitive ability locomotion.

Allow incremental exploration: It allows users to travel a route and enables browsing of the ontology with the current term as the focal point. It is related with the cognitive abilities information gathering and locomotion.

As a result, Diamond (D'ENTREMONT; STOREY, 2009), a visualization plugin for protege, is based on these principles. One of the advantages is that Diamond uses the fisheye strategy, which defines a Degree of interest (DOI) function. This function assigns a value to each item in the information structure. The authors introduce the notion attention reactive interfaces, which basically employs the DOI to reduce navigation overhead. Diamond uses two threshold values to define three interest levels: non-interesting, interesting and landmark. For each level, it allows users to specify a color, which help user to identify the level. The focus of these plugin was mostly for experienced users.

We review the ontology visualizations works because in the ontology exploration the visualization should help user to understand the ontology. We implement our ontology visualization in a very simple manner with the basic functionalities of zooming, panning, lenses and color nodes.

In the next chapter, we explain the sub-ontology extraction techniques used in the literature to obtain a subset of a larger ontology.

4 SUB ONTOLOGY EXTRACTION

In this work, we assume that both ontology module extraction and ontology view extraction involve a common step of *sub-ontology extraction*, whose goal is to select a subset of semantically related elements of a given ontology. The idea of extracting a subset of a larger ontology is referred to many different names by different authors. According to (SEIDENBERG; RECTOR, 2006), subset extraction techniques can be broken down into three main categories:

- Query-based methods
- Network Partitioning
- Traversal Approach

There is also the logical approach (VESCOVO et al., 2013; TSARKOV; PALMISANO, 2012) that is receiving a lot of attention due to its applicability to web semantic. The logical approach is considered as a modularization technique (PARENT; SPACCAPIETRA, 2009). The network partitioning approach can be considered as included in the logical approach classification, because it uses the advantages of a logical language. In the logical approach, one seeks what logical language offers the conditions for being a modular ontology language. The most well-known logical languages are Distribution Description logics and its syntax C-OWL, $\mathcal{E}$-connection and Packaged-extended Description logics(P-DL). Two broad classes of approaches are adopted to asserting and using semantic relations between multiple ontology modules: DDL and E-connections adopt the *linking* approach that assumes that the modules are nonoverlapping or disjoint, while P-DL adopts the *importing* approach that allows direct use of foreign terms in an ontology module. As we can see, one of the drawbacks of using logical approaches is the *language dependency*. A detailed description can be found in Bao, Caragea and Honavar (2006). In the next section we present the use of network partitioning applied with logical approach.

In this chapter, we will describe above the categories in the following order: query-based methods, network partitioning and traversal approach.

The sub-ontology extraction step is performed differently in module extraction and view extraction, since the requirements that the sub-ontology should meet in both approaches can be different. We roughly define what is ontology module and ontology view, before presenting each technique.

The notion of a module is well-understood in the software engineering community, but on ontology modularization it can be understood in rather different ways. Doran (2009) in his PhD thesis presents the principles of ontology modularization and the following definition: *An ontology module is a reusable component of a larger or more complex ontology, which is self-contained but bears a definite association to other ontology modules, including the original ontology.*

The notion of view is well-defined in the database community, but in the ontology view community it can also be understood in rather different ways. According to Noy and Musen (2009), Bhatt et al. (2004a), ontology view is a portion of an ontology, which is extracted according to the *user requirements* and that can *overlap* other ontology views.

4.1 Query-based Methods

Query-based methods provide a view mechanism similar to those existing in SQL. This mechanism makes them intuitively familiar to computer scientists with a background in databases. The shortcomings of these approaches are that they provide only very low-level access to the semantics of the ontology being queried. Query-based views are good for getting very small, controlled, single-use extracts, which are tightly focused around a few concepts of interest. By contrast, the methods presented herein create self-standing, persistent, multi-use ontology subset. The examples provided by the different works consider ontology not only the classes and relations but also the instances within the ontology definition. In the following, we describe some characteristics of some query-based methods, namely vSPARQL, KAON views, RVL and SAIQL.

4.1.1 vSPARQL

The view definition language vSPARQL (SHAW et al., 2011) allows to define views for semantic web content. vSPARQL is used to specify both the selection of the information that can be accessed through the view and how the selected information is reorganized and modified through the view. vSPARQL might be a good low-level tool for extracting views, but it is not a solution itself.

They define the following requirements for their language:

- The input and output should be RDF graphs.
- View definitions should be able to include arbitrary facts. By means, it must explicitly indicate the triples to include and exclude. vSPARQL supports basic edge selection, specification of paths of arbitrary length.
- The view definition language should allow the combination of content from different graphs.
- Views should be able to restructure, modify or augment selected facts. Indeed, they add new information to the original subset.

Their view definition language does not use formal logics to guarantee that a derived subset has the same properties as the original ontology. Instead, it allows the application developer to specify exactly which facts are relevant and how those facts should be arranged or augmented. One of the drawbacks is that vSPARQL can be difficult to write for users who are not computer scientists.

4.1.2 KAON views

An ontology view mechanism (VOLZ; OBERLE; STUDER, 2003) is defined based upon the RQL query language(ALEXAKI et al., 2000) that allows easy selection, customization and integration on the semantic web. It defines views for classes and properties. Authors implemented their view mechanism in the KAON Server (BOZSAK et al., 2002). Views definitions are stored in RDF syntax. As soon as the view is created by the user, consistency checks are performed through a set of axioms. The use of this query language as the previous one is focused on users that are computer scientists.

4.1.3 RVL

The view definition language RVL (MAGKANARAKI et al., 2004) is a conceptually simple language that enables both humans and applications to understand view specifications as normal RDF/S schemas. An RVL view specifies a virtual description schema graph (or virtual schema for brevity). Its extension corresponds to a virtual description base graph (or virtual base for brevity), which is a valid instance of the virtual view schema.

RVL allows queries to reorganize the RDFS hierarchy when creating a view. This allows views to be customized on-the-fly for specific applications' requirements. It can be used to implement advanced user aids, such as personalized navigation and knowledge maps. Their views are merely a collection of pointers to the actual concepts, and are discarded after they have served their purpose.

4.1.4 SAIQL

OWL-SAIQL (Schema And Instance Query Language) (BERCOVICI, 2008) is a query language that combines T-Box and A-Box in an integrated manner. SAIQL defines a query mechanism with template patterns. The query template pattern extracts all the concepts, their individuals and the description concepts related to them. The benefit provided by SAIQL in this specific case is twofold: first, Bercovici is not only pruning an ontology but, at the same time, he extracts a part of this ontology. This means the part comprising the relation, the concepts, their description and the associated individuals. Whereas the previous pruning methods only provide the concepts and their subsumed concepts. The second benefit of SAIQL is that the query patterns allow very fine-grained statements to be crafted and organized into families or groups that can be eventually invoked by the user through a graphical user interface. Hence, query-based customization operators can be numerous and can include both generic and highly domain-specialized templates, which lead to substantially more numerous ways for also customizing ontologies.

4.2 Network Partitioning

In this case, the ontology is treated as a network of nodes connected by links. The class hierarchy can be interpreted as a directed acyclic graph and any relations between classes can be represented as links between the nodes. They use the idea of network partitioning in their algorithms. But, the result is not always an ontology that guarantees semantic understandability because the algorithms are focused in decomposing the ontology creating clusters, which is different from obtaining a well sub-ontology. Some works done with the idea of partitioning are presented below, but we do not describe detailing the algorithms because it focused in a logical approach employing description logic syntax, which is out of our scope.

4.2.1 Structure based partitioning

This method, explained in Stuckenschmidt and Schlicht (2009), partitions large ontologies into smaller modules. Its focus is on obtaining a set of modules or partitions, where modules have not overlapping portions. Modules are disjoint and consist of a set of concepts that are connected semantically to each other and do not have strong semantical dependency with information from outside the module. Authors introduce the notion of dependency, which is used throughout their technique. Their method consists of five independent steps:

- Creating a dependency graph: a dependency graph is extracted from a source file.
- Determining the strength of dependencies: Based in the structure of the dependency graph the weight of the dependencies is determined. They use the social network theory.
- Determining the modules: the sets of strongly related concepts are detected.
- Assigning single concepts during the partition. Then, those nodes were isolated using the degree of proximity with other nodes.
- Module Merging: two adjacent modules can be merged if the height is 1.0 or 0.5. This height is a measure used for checking the strength of the internal dependency.

As part of the algorithm, users have to define the size of the module, which can be difficult to assign when the objective is to obtain consistent modules. They prove their algorithm over SUMO and NCI cancer ontology. One of the disadvantages is the criteria for assigning heights. It is not well explained and does not offer a consistent ontological principles for doing that.

4.2.2 Automated Partitioning using E-connections

This method, presented in Grau et al. (2005), partitions a knowledge base (KB) represented in OWL formalism. For that purpose, $\mathcal{E}$-connection, which is a language for combining $\mathcal{SHOIN}$ KB that is a family of a description logic, provides modularity benefits. The modules produced by their algorithm are formally proven to contain the minimal set of atomic axioms necessary in order to maintain crucial entailments. One disadvantage is that this method is just applied to one description logic family.

4.3 Traversal Approach

On the other hand, traversal approaches represent the extraction as a graph traversal. Traversal methods start from one or several concepts of the ontology, and include in the module the concepts and relations that are linked to these elements. Some traversal-based approaches are language-dependent as well, since they assume that the ontology is represented in a specific ontology representation language, such as OWL. However, some approaches represent the ontology as an abstract graph, in a *language-independent* way. Due to this feature, here we focus in traversal-based approaches for subset extraction.

d'Aquin, Sabou and Motta (2006) extract modules including all elements that are either directly or indirectly related with the target entities. Their approach assumes the OWL formalism, taking advantage of inferences, which are used during the extraction. The advantage is that they include elements that are implicitly related by the mean of inferences. The input of their algorithm is the ontology O and the sub-vocabulary SV of O that the extracted module should cover. SV is described by a set C(SV) $\supseteq$C(O) of concept names, a set P(SV) $\supseteq$ P(O) of property names and a set I(SV) $\supseteq$ C(O) of individual names.

The modularization algorithm consists in computing C(M), P(M), I(M) and A(M) recursively, in a fix-point like algorithm. All of them composed the subset extracted. They defined the following rules for selecting concepts, properties, individuals and assertions.

Concept: The algorithm takes a concept C if C is the super concept of two concepts already in the subset C(M); if C is the most specific concept of an individual already in the subset I(M); if C is a concept expression such that a concept D, which D $\in$ C(M), D $\supseteq$ C or if C is in a D expression.

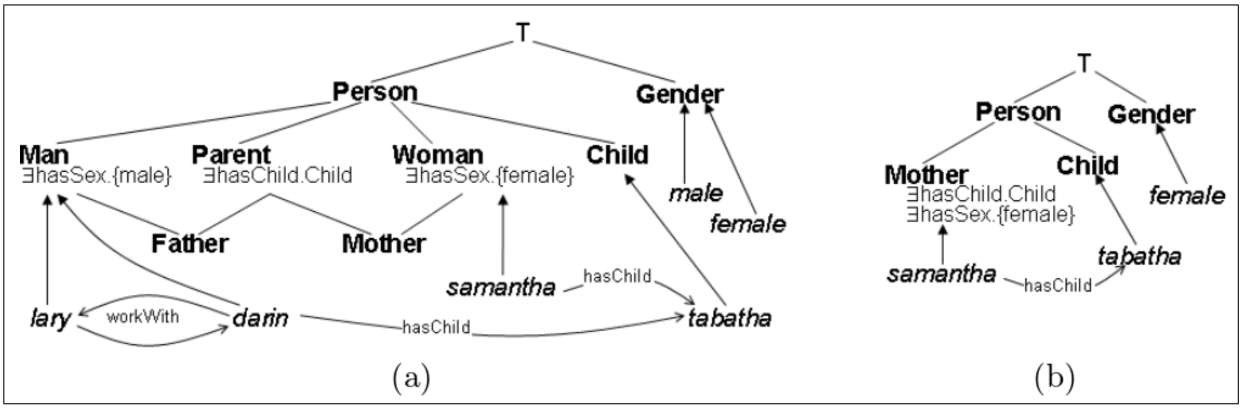
Property: The algorithm takes a property p if p is the super-property of two properties in P(M), if p relates an individual such p(a,b) and a$\in$ I(M), or if P is in a concept expression that belongs to C(M)

Instance: The algorithm takes an individual a if p(b,a) and b in I(M). Also, if the individual is an instance of a concept c in C(M).

Assertion: The algorithm takes an assertion A if A relates elements of C(M), P(M) or I(M).

For example in Figure 4.1, we have the original ontology in the part *a* and the result in the part *b*. We are interested in extracting from this ontology the knowledge concerning Samantha and the concept of mother. Therefore, the sub-vocabulary SV used as an input for the algorithm is: C(SV) =Mother,P(SV) =female, and I(SV)=samantha. We start the algorithm with *Mother* concept and it is in C(M). It is added the properties *hasChild* and *hasSex* because those properties relate the *Mother* concept, which is in C(M). Then, it is included *tabatha* because it is related with *Samantha* which is in I(M) and *female* because it is used in an included concept expression. Also, *Gender* and *Child* are selected because they are the most specific concepts of the individuals *female* and *Tabatha*, respectively. Then it is taken *Person* because is the superconcept of two concepts that are already in the subset which are *Mother* and *Child*.

Figure 4.1: The original ontology (a) and the resulting module (b)



Source: d'Aquin, Sabou and Motta (2006))

The main criticism of this approach is that it is tightly focused on knowledge selection. Due to that it is also considered the selection of individuals. Using individuals involves to select the concepts from which individuals are instantiated. These concepts can be disjoint.

Doran, Tamma and Iannone (2007) tackle the problem of ontology module extraction from the perspective of an Ontology Engineer wishing to reuse part of an existing ontology. The approach extracts an ontology module corresponding to a single user-supplied concept that is self-contained, concept centered and consistent. They proposed an abstract graph model for the extraction process. The model is an edge-labeled directed graph G, given an alphabet $\sum_E$, is an ordered pair G=(V,E) where:

- V is a finite set of vertices

- $E \supseteq V \times \sum_E \times V$ is a ternary relation describing the edges(including label). Needles to say, E is not symmetric.

The ontology model is defined as $G_M = (V_M, E_M)$. Their technique do not traverse *disjoint* labeled edges. Their algorithm is called recursively. It is used a container of booleans for E not to be followed (*Excluded*); V that have been visited (*Visited*) and V to be visited (*ToVisit*). The algorithm input is a vertex. If the vertex is not visited, the algorithm inserts into *Visited* and create a container of the relations where the given vertex is the source. Then, the algorithm loops through that container and takes the first element of the container and if it is not in the prohibited relations, it is added to E_M and inserts the range vertex in the container of *ToVisit*. If *ToVisit* is not empty we extract the first element and called again the algorithm. The pseudo-code is presented in algorithm 1.

Algorithm 1 Module Extraction (DORAN; TAMMA; IANNONE, 2007)

```
procedure EXTRACTMODULE($Vertex\ s$)
    if $s \notin$ Visited then
        insert $s$ into Visited
        create container$X = e \in E|s \times \sum_E \times v$ Visited
        while $X$ is not empty do
            $y =$ first element of $X$
            if $y \notin$ Excluded then
                $y \cup E_M$
                insert $r$such that $y = s \times \sum_E \times r$ intoToVisit
            end if
            if ToVisit is not empty then
                $t =$ first element of ToVisiti
                remove $t$ from ToVisit
                extractModule($t$)
            else
                output $G_M$
            end if
        end while
    end if
end procedure
```

As result, they implement their algorithm in ModTool. Their tool makes use of JENA[1] for storing the generated ontology and Pellet [2] for checking consistency. They take an ontology about the University domain illustrated in Figure 4.2. The input of the algorithm is the *Academic Staff*. In the first iteration, *Academic Staff* is added to *Visited*. Due to *Admin Staff* is disjoint with *Academic Staff*, *Admin Staff* is not added to *ToVisit*. *Admin Staff* has no more edges to traverse and is removed from *ToVisit*; the extraction now continues with *Lecturer* as the concept of focus.

In the second iteration, *Lecturer* is added to Visited. *Lecturer* is the domain of the object property *supervises*; the range of this property is *PhD Student*, thus *PhD Student* is added to *ToVisit*. *Lecturer* has no more edges to traverse and is removed from *ToVisit*; the extraction now continues with *Research Staff* as the concept of focus.

In the third iteration, *Research Staff* is added to Visited. *Research Staff* has one subclass

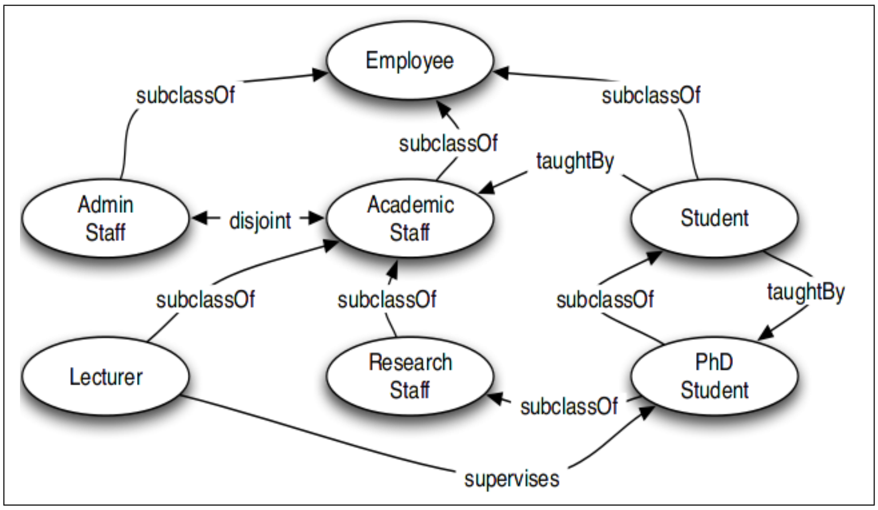
[1] http://jena.sourceforge.net/

[2] http://clarkparsia.com/pellet/

PhD Student. *PhD Student* has already been added to *ToVisit*, which does not permit duplicate elements, however the edge that describes the subclass relation will be included in the module. *Research Staff* has no more edges to traverse and is removed from *ToVisit*; the extraction now continues with *PhD Student* as the concept of focus.

Finally, *PhD Student* is added to Visited. *PhD Student* has no valid edges to traverse. Even though *PhD Student* is the range of an object property, this edge is not traversed. *PhD Student* is removed from *ToVisit*. *ToVisit* is now empty; the extraction process ends and the ontology module is outputted.

Figure 4.2: Ontology of the University domain



Source: Doran, Tamma and Iannone (2007)

The advantage of this approach is that it is easy to understand and adapt to any kind of formalism. But, a container is created for excluded relations, which means that the ontology engineer should know what relations are not going to be taken. This seems to be a manual operation performed by the ontology engineer.

The approach proposed by Seidenberg and Rector (2006) takes advantage of the detailed semantics captured within an OWL format. It was exemplified with the Galen Ontology. They called the subset of an ontology *segments*. But the idea is the same. Thus, their algorithm traverses upwards of the hierarchy until the top concept is reached. The algorithm goes down the class hierarchy. The property hierarchy is, however, never traversed downwards. Properties are not of interest unless they are used in the class hierarchy. So, if they are used, they and their super-properties and no other properties, are included. Sibling classes are not included in the extract.

Having selected the classes up and down the hierarchy from the target class, their restrictions, intersection, union and equivalent classes now need to be considered: intersection and union classes can be broken apart into other types of classes and processed accordingly. Equivalent classes (defined classes, which have another class or restriction as both their subclass and their superclass) can be included like any other superclass or restriction, respectively. Restrictions generally have both a type (property) and a filler (class), both of which need to be included in the subset.

Additionally, the superproperties and superclasses of these newly included properties and classes also need to be recursively included, otherwise these concepts would just float in OWL hyperspace. The authors do not include the subclasses of those classes included via links because there is the risk that the result could be the entire ontology.

After that the subset is constrained by property filtering and depth limiting using boundary classes. In property filtering, they did an analysis of GALEN property hierarchy and organize them in upper level meta-properties which are: modifierAttribute, constructiveAttribute, locativeAttribute, structuralAttribute, partitiveAttribute and functionalAttribute. Depth limiting is a number, which limits the traversal depth. It occurs when a set of links or properties are added when doing the extraction. Each classes' restrictions has links to other classes, which are included to produce a semantically correct extract. However, if, upon reaching a certain recursion depth, calculated from the target concept, all the links on a class are removed, this class becomes a boundary class.

In the example below, one might remove the axiom stating that the Pericardium (the membrane that surrounds the heart) is a component of the CardiovasuclarSystem (line three of the Figure), since one may not be interested in including the CardiovascularSystem and everything related to it in a segment of the Heart. This creates a boundary class.

$Heart \subseteq \exists hasStructuralComponent.Pericardium$

$Pericardium \subseteq SerousMembrane$

$Pericardium \subseteq \exists isStructuralComponentOf.CardiovascularSystem$

The approach is more automated, aiming to produce a heuristic algorithm that creates a useful segment without much user intervention. Another advantage of their technique is that they constrain the module size. Even though they mention that they use meta properties for properties, it is just relation attributes that are not ontological meta-properties.

4.3.1 PROMPT

Noy and Musen (2003) present an extension to the PROMPT suite of ontology maintenance tools, which is a plugin to the Protege ontology editor. Noy and Musen (2009) introduce the notion of traversal view, which is a view where a user specifies the central concept or concepts of interest, the relationships to traverse to find other concepts to include in the view, and the depth of the traversal. For example, given a large ontology of anatomy, a user may use a traversal view to extract a concept of Lung and organ parts that surround the lung or are contained in the lung. They consider ontologies expressed in RDF Schema, and also adapted to OWL formalism. They present a formal definition for traversal view, which consists of two parts: the view specification and the view computation. As part of view specification, it is defined traversal directive and a traversal view specification, where this last one is defined as a set of traversal directives. *Traversal directive* is defined as pair of (C_st, PT) where C_st is a class or an instance in the ontology (the starter concept of the traversal); PT is a set of property directives. Each property directive is a pair (P, n), where P is a property in the ontology and n is a non negative integer or infinity (inf), which specifies the depth of the traversal along the property P. In the view computation, it is presented *traversal directive result* , which is the result of applying the directive; and *traversal view*, which is the union of traversal directive results for each traverse directive.

The algorithm starts from one class of the ontology being considered. Relations from this class are recursively traversed to include the related entities. These relations are selected by the user, and for each relation selected, a depth of traversal (or traversal directive) is assigned. The traversal directive is used to halt the traversal of the corresponding relation when the specified depth is reached.

For example, in their work was taken the Foundational Model of Anatomy (FMA) ontology was used, and extracted a subset through the traversal directive: $C_st = \text{Lung}; PT = (\text{hasPart}, 2), (\text{containedIn}, 1)$.

- Lung is included in the view.

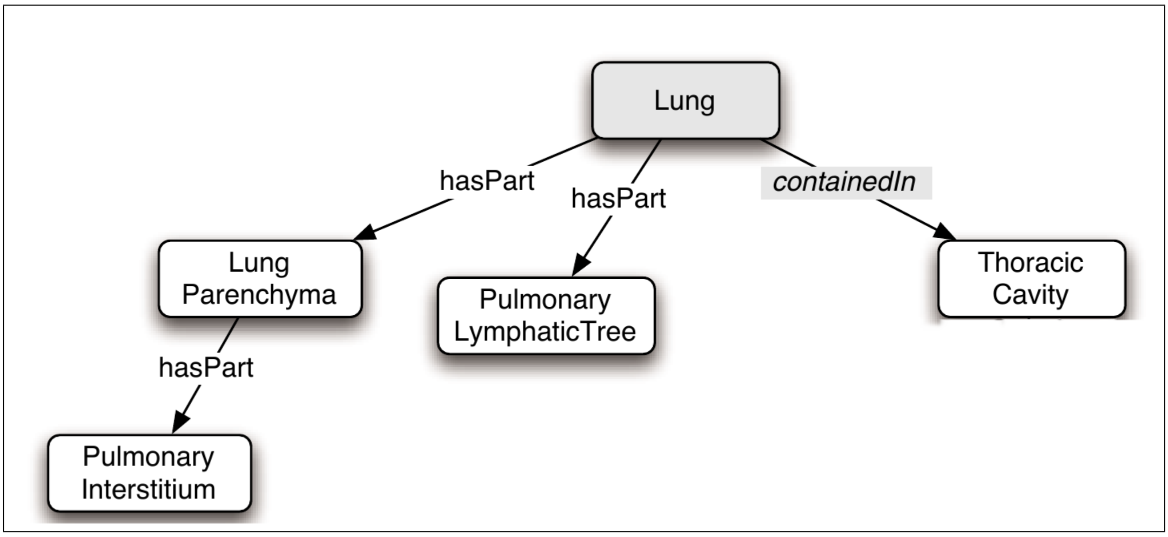
- If Lung is in the domain of the property *hasPart*, then all classes in the range of this property are also included in the view. We will denote the set of these classes C_{hasPart} . From the classes in Figure 4.3, at this step, it will be added LungParenchyma and PulmonaryLymphaticTree to the view and to $C_{hasPart}$.

- If Lung is also an instance (RDF Schema does not prevent classes from being instances as well) and has a value for the property *hasPart*, those values are also included in the view. Then, it is added these values to the set $C_{hasPart}$. The view now contains the traversal along the *hasPart* property of depth 1.

- It is repeated steps 2 and 3 once for each concept in the set $C_hasPart$ to add values for the traversal along the hasPart property of depth 2. In the example, in Figure 4.3, this step will add the class PulmonaryInterstitium to the view.

- Then it is repeated steps 2 and 3 for the class Lung and property containedIn once, adding values for the traversal along the property *containedIn* of depth 1. This step will add the class ThoracicCavity from Figure 4.3 to the view.

Figure 4.3: A subset of FMA, having the class *Lung* as input



Source: Adapted from Noy and Musen (2009)

This flexible approach allows an Ontology Engineer to iteratively construct the ontology module that they require by extending the current view. However, the ontology engineer needs to have a deep understanding of the ontology that is being used. This technique is not automatic and takes into account the user involvement in selecting the relations to be traversed and associating to each of them a level of recursion, at which the algorithm should stop traversing relations. The focus of this method is on query answering.

4.3.2 MOVE

Bhatt et al. (2004a) present a distributed approach to sub-ontology extraction. They called the result of this process a sub-ontology or materialized ontology view.

The process begins with the import of an ontology (i.e. constructing an internal memory representation of the ontology), which is represented using an ontology standard. Also the user (or application) requirements and specifications are imported. By means, it allows a user to provide subjective information, pertaining to what must/must not be included in the target sub-ontology, on which the extraction process is based on. The input can be a concept, attribute, or relationship.

Every ontological element may have a labeling of selected, must be present in the sub-ontology; deselected, must be excluded from the sub-ontology or void, the extraction algorithm is free to decide the respective elements inclusion/exclusion in the sub-ontology.

In the extraction process optimization schemes are defined, which handle several issues pertaining to it, such as ensuring the consistency of initial requirements. They do not use a graph representation for their algorithm. Instead, they propose an object-oriented design creating partitions and applying the parallelism paradigm. This is followed by the execution of the optimization algorithms that finally produce the extracted sub-ontology. The first one is Requirements Consistency Optimization Scheme (RCOS), which ensures that the requirements as expressed by the user are consistent and correct. These rules were defined in Wouters et al. (2002) and they are described below.

- RCOS1: This rule stipulates that if a binary relationship is selected, the two concepts must be in the target ontology.

- RCOS2: This rule enforces the condition that if an attribute mapping has a selected labelling, the associated attribute and the concept that it is mapped onto must be *selected* to be present in the target ontology.

- RCOS3: This rule stipulates that if an attribute mapping has a deselected labelling, its associated attribute must be disqualified from the target ontology.

- RCOS4: It uses the notion of path. If an attribute is selected, but the concept it *belongs* (mapped) to is deselected, there must exist a path from the attribute to another concept that is not deselected.

The second optimization scheme is called Semantic Completeness (SCOS) (BHATT et al., 2004b) that states the following conditions:

- SCOS1: If a concept is selected, all its super-concepts, and the inheritance relationships between the concepts and its super-concepts have to be selected.

- SCOS2: If a concept is selected, all the aggregate part-of concepts of this concept, together with the aggregation relationship have to be selected as well.

- SCOS3: If a concept is selected, all the attributes it possesses with a minimum cardinality other than zero and their attribute mappings should be selected as well.

The third optimization scheme is called Well Formedness (WFOS). This optimization scheme contains the proper rules to check that the new sub-ontology is a valid ontology. By means, if a concept was deselected its attributes must be deselected, if an attribute is deselected the mapping between the attribute and the concept must be deselected, if relationship is deselected, the concepts that are in the relationship must be deselected. As a result, it should not exist islands in the ontology. An island is defined as a group of 1 or more ontological concepts that cannot reach every other ontological concept in the total group.

The last optimization scheme is called Total Simplicity (TSOS). The result is a smallest subset that is still a valid ontology. It is applied a modified version of kruskal algorithm for minimal spanning trees. For that purpose the elements that are with *void* label are changed with accepted and rejected label following certain rules defined in Wouters et al. (2009). A detailed description of their algorithms is in Wouters et al. (2009)

Materialized ontology view (WOUTERS et al., 2009) is a (valid) ontology that consists solely of projections, copies, compressions, and/or combinations of elements of the base ontology, presenting a varying and/or restricting perception of the base ontology, without introducing new semantic data. Also for a minimum quality in materialized ontology view should include these two optimization schemes RCOS and WFOS.

In the import process they annotate a concept with a label. The advantage of this process are these optimization schemes, which provide a way to introduce quality in the extraction process; and being user centered, language independent. The focus of their algorithm is to improve the efficiency of information retrieval.

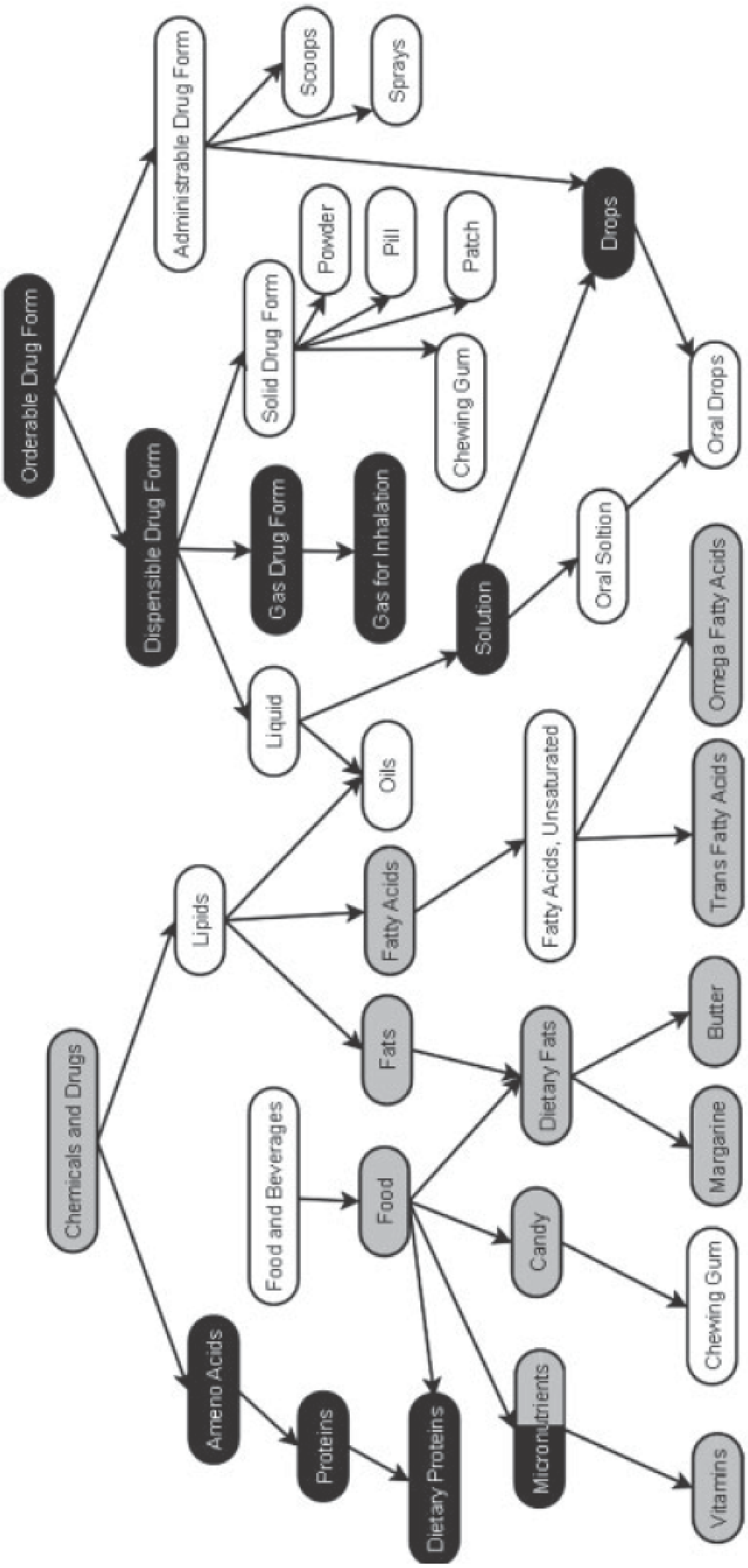
Although, the extraction mechanism is user centered, this process implies many interactions because user has to define what optimization schemes to follow. They stated that the less relationships, the better is the sub-ontology extracted. Thus, their method achieves the statement above by their optimization schemes. However, when reducing relationships and clustering concepts, there are lost of information. Unless the user is an expert in the domain, this will be helpful, otherwise not.

The algorithm is exemplified in Flahive et al. (2011) with a portion of the Unified Medical Language (UMLS) meta-thesaurus ontology, which is the pharmacy ontology depicted in Figure 4.5. The scenario is when a pharmacist selects the main pieces of information required from the UMLS ontology and passes the list to the ontology engineer. The ontology engineer uses this list as a *labeling* for the UMLS ontology. The concepts are labeled with *selected*, *deselected* and *void* depicted in Figure 4.4 with nodes in gray, dark gray, and white color, respectively. After that, the extraction of the *selected* elements is done following RCOS and WFOS. The next step is to ensure that each element in the sub-set is connected by some path, which is done according to SCOS. So for example, *Fatty Acids* concept has the label *selected*, and for SCOS1, *Lipids* concept must be selected as well. The extracted sub-set is depicted in Figure 4.5.

4.4 Discussion

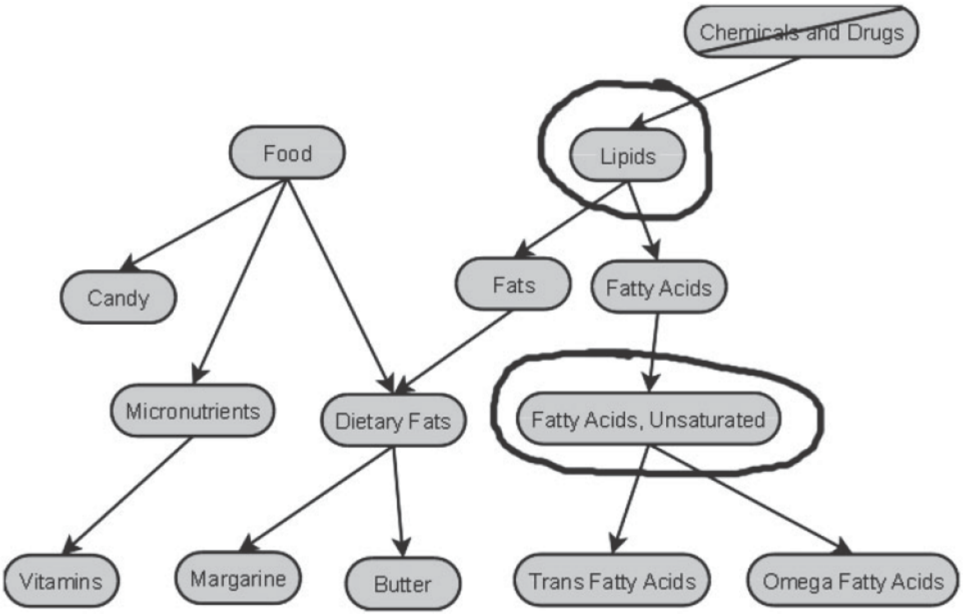
In the approaches mentioned above, two main limitations are noticed. First, the existing approaches of ontology modularization rely on static ontologies that can be inconsistent to cover basic user's tasks. Second, modularization algorithms consider mainly the structure of the input ontology, instead of semantics. Consequently, we need semantics-based criteria to determine the border of ontology modules. Moreover, the contextuality of the ontology module or ontology view will considerably depend on the semantic covertness of the original input ontologies. Our proposal presented in the next chapter considers the use of meta-properties in the sub-extraction algorithm to cover this gap.

Figure 4.4: A Labeled Portion of the UMLS Meta-thesaurus Ontology as a Connected Graph



Source: Flahive et al. (2011)

Figure 4.5: Extracted sub-set



Source: Flahive et al. (2011)

5 DOMAIN ONTOLOGY FOR DIAGENESIS AND MICROSTRUCTURAL

In this chapter, we describe the well-founded domain ontology developed for testing our approach covering the communities of sedimentary diagenesis and microstructural characterization. In this case, a knowledge community is the group of people whose job requires shared domain knowledge to support problem solving. Firstly, we describe the domains of diagenesis and microstructural analysis. Then, we explain the most important terms of the domain ontology and finally, we discuss about the developed ontology.

This well-founded domain ontology was the result of an ontological analysis over the Petroledge® ontology, restructuring the ontology based on foundational ontology principles preserving the original conceptualization that supports the knowledge models. Furthermore, this ontology was extended with microstructural terms and validated by geologists.

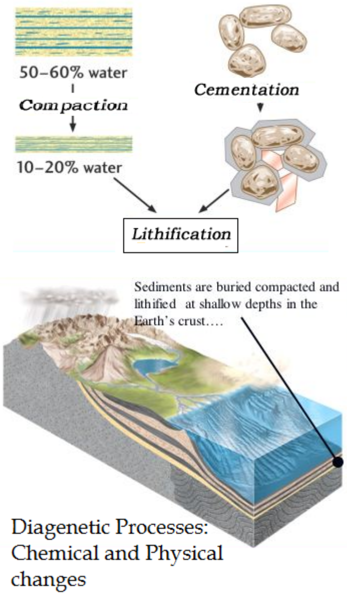
5.1 Diagenesis and Microstructural Analysis

In this section, we introduce the basic notions of diagenesis and microstructural characterization. For that purpose, the larger areas that involve those domains are those of Petrology and Microtectonics, respectively. Petrology is a subfield of geology that involves the study of rocks, their composition, textures and the process that formed them. Petrology is closely related to Geochronology, involving techniques for the determination of the ages of rock, and to Geochemistry, which deals with the amount, distribution, and migration of chemical elements (and their isotopes) contained in minerals, rocks and soils in materials from the Earth and other planetary bodies. Microtectonics (PASSCHIER; TROUW, 2005) concerns the interpretation of geometries of solid-state deformation in rock thin sections, in order to reconstruct their tectonic evolution. It is also related to Structural Geology. Structural Geology is the study of the three-dimensional distribution of rock units with respect to their deformational histories.

Diagenesis comprises a broad spectrum of physical, chemical and biological post-depositional processes, by which original sedimentary assemblages and their pore waters react at low temperatures (below 200 degrees Celsius) by attempting to reach textural and geochemical equilibrium with their environment. These processes are continually active as the environment evolves in terms of temperature pressure and chemistry during the deposition burial and uplift cycles of sedimentary basin.

The terms defining texture, composition of minerals and fluids, their paragenetic sequence, porosity, and diagenetic processes are fundamental for the diagenesis community. The composition of rocks corresponds to a set of detrital constituents, diagenetic constituents and pores. Detrital and Diagenetic constituent are composed of minerals. A mineral (KLEIN, 2002) is a naturally occurring substance with a highly ordered atomic arrangement and a definite chemical composition. Most minerals are naturally formed by inorganic process. Minerals are most commonly classified on the basis of the presence of their major chemical component into oxides,

Figure 5.1: Diagenesis Processes

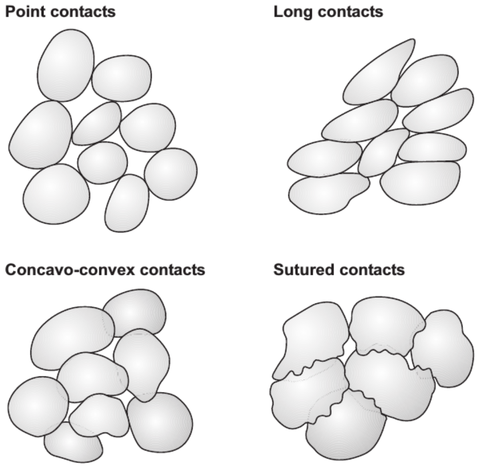

50–60% water
Compaction
10–20% water
Cementation
Lithification
Sediments are buried compacted and lithified at shallow depths in the Earth's crust....
Diagenetic Processes: Chemical and Physical changes

Source: Adapted from Press et al. (2006)

sulfides, silicates, carbonates, phosphates, and so forth. Grains (PASSCHIER; TROUW, 2005) are volumes of crystalline material separated from other grains of the same or other minerals by a definite boundary. Grains can be subdivided into Monomineralic grains, *Rock Fragments* and *Intrabasinal constituents*, such as *Bioclasts*. *Bioclasts* are fragmented or full skeletal remnants(fossils) of an organism preserved since some time in the geologic past. *Rock Fragments* consist of polymineralic or polygranular grains that are particles eroded from igneous, sedimentary, or metamorphic rocks. Pores (NICHOLS, 2009) correspond to the volume between or within grains that is void. Pores may be connected through pore throats forming a pore system. The porosity of a rock is the proportion of its volume that is not occupied by solid material but is instead filled with a gas or liquid. A paragenetic sequence (WORDEN; BURLEY, 2009) is the interpreted order in which diagenetic processes occurred during rock formation.

The diagenetic processes are the physical and chemical changes that alter the characteristics of sediments after deposition. Those processes are compaction, cementation, dissolution, recrystallization and replacement. The effect of compaction in a clastic rock is determined by looking at the nature of the grain contacts (see Figure 5.2). Contacts can be point, long, concavo-convex and sutured. In the cementation process, it occurs the nucleation and growth of crystals within pore spaces in sediments. Chemical compaction involves the dissolution of grains by pressure dissolution along grain contacts. Recrystallization is the formation in situ of new crystal while retaining the same mineral composition. Replacement refers to the process whereby one mineral dissolves and another is precipitated in its place essentially simultaneously. These process are important to be modeled, however, in this work we aim to deal only with concepts of endurants(UFO-A) and then not with events.

Figure 5.2: Types of grain contacts



Source: Press et al. (2006)

Microstructures (PRIOR; RUTTER; TATHAM, 2011) play a key role in studies of the deformations caused by differential stresses that affected rocks. Microstructures are characterized mainly based on careful observation of fabrics in order to understand the sequence of events that affected the rock. The main objective of microstructural analysis is therefore to unravel the relation between deformation and diagenesis events that have affected the texture and structures of a sedimentary rock, and their effects on porosity and permeability.

A structure (SNOKE; TULLIS; TODD, 1998) is understood as the arrangement of the parts of a rock mass irrespective of scale, including spatial relationships between the parts, their relative size and shape and the internal features of the parts. A deformational structure is a disorder in the arrangement of the parts. The effects of the deformation process over sedimentary rocks are deformational structures and deformation zones. The terms deformation band, fault, joint, vein, stylolite, breccia, deformation zone, fault rock are fundamental for the microstructure community.

Deformational Processes are classified as *brittle deformation* and *ductile deformation*. *Brittle deformation* occurs when a rock breaks. *Ductile deformation* occurs when rocks bend or flow. As a result of *Brittle deformation*, *fractures* and *fault zones* occur. A *fracture* is a planar discontinuity usually involving some dilation, including cracks, joints (large cracks) and faults. Fractures are easy to recognize by their sharp, narrow and usually straight nature and by the displacement of markers. A joint is a plane surface of fracture or parting in a rock, without displacement. A fault is a shear fracture. *Normal Fault*, *reverse fault*, *strike-slip fault* (FOSSEN, 2010) are types of faults.

Deformation zones are portions of rock bodies that have suffered deformation. The deformation zones can be further divided into *fracture zones* and *shear zones*. A *fault zone* is a type of *fracture zone*. *Deformation bands*(PASSCHIER; TROUW, 2005) are brittle *fault zones* that develop very close to the Earth's surface in poorly or even unconsolidated porous sediment. They can be divided into disaggregation band, phyllosilicate band, dissolution band, cataclastic band, and dilatation band. A fault zone has three components: *fault core*, *damage zone* and *protolith*. In coherent sandstones, *fault cores* (LAUBACH et al., 2014) are usually narrow (less than 1 meter), consisting of low-porosity highly deformed materials. *Damage zone* (FOSSEN, 2010) is the volume of deformed rocks that results from initial process zone development and subsequent slip surface initiation, propagation, and linkage or interaction in the fault zone. *Protolith* corresponds to the original undeformed rock.

During dilatation, structures such as veins, strain shadows, fringes and microboudins can be formed. ((PASSCHIER; TROUW, 2005)). *Veins* are subplanar concentrations of minerals that have precipitated from solution along fracture. When a vein lies at a high angle relative to their opening direction it is called *an extension vein or tension gash*, and when it lies at small angle, it is called a *shear vein*. A *Strain shadow* is when the dilatation site happens flanking rigid objects. A *Strain fringe* is a type of strain shadow containing fibrous material precipitated adjacent to a stiff or rigid object. A *jog* is a step in a planar structure such as a fault.

Folds occurs as a result of *ductile deformation*. Folds (HUDLESTON; TREAGUS, 2010) are geological structures that are seen in layered rocks in many different scales. Folds are classified as anticlinal and synclinal. Fault rocks are volumes of rocks delimited by shear zones (SIBSON, 1977).

Brittle fault rocks can be subdivided into incohesive and cohesive types. Incohesive brittle fault rocks can be subdivided into incohesive breccia, incohesive cataclasite and fault gouge. *Clay smear* is clay-rich fault gouge formed in sedimentary sequences containing clay-rich layers which are strongly deformed and sheared into the fault gouge. Incohesive cataclasite can be subdivided into *ultraclasite*,*mesocataclasite* and *protocataclasite*. Cohesive fault rocks can be subdivided into cohesive breccia, cohesive cataclasite and pseudotachylyte. Mylonite is a fault rock which is cohesive and characterized by a well developed schistosity resulting from tectonic reduction of grain size. Mylonites may be subdivided according to the relative proportion of finer-grained matrix into protomylonite, mesomylonite and ultramylonite.

Microfractures (BLENKINSOP, 2000) can be sub-divided into *microfaults* and *microcracks*. Microcracks are planar discontinuities at the grain scale or smaller, commonly with some dilation but with negligible displacement. Microcracks can be classified as intragranular (within single grains), transgranular (across two or more grains) and circumgranular or along grains boundaries. *Microfaults* are shear microfractures that contain grain fragments formed by cataclasis. Microfractures are called intragranular if they affect only single grains, and intergranular if they affect two or more grains.

Within those other types of structure deformation we found *microkinks*, *deformation lamellae*, and *deformation twin*. *Microkinks* occur as small isolated structures in quartz and feldspars. *Deformation lamellae* are particularly common in quartz, where they usually have a sub-basal orientation. *Deformation twin* occurs common in deformed carbonates and plagioclase feldspar.

In the next section, we describe the domain ontology of our case study developed for enhancing and improving the PetroQuery® system.

5.2 Domain Ontology of Case Study

The PetroQuery® (described in section 2.5) system was initially designed for the purpose of consultation of petrographic features of siliciclastic rocks. But the software evolution has expanded the scope of the knowledge-domain and increased the number of terms accessed by the interface. In our work, we started with the well-established ontology of Petroledge® for sedimentary rocks (described in appendix A) and then we introduced the concepts of microstructural domain. After interviews, ontological analysis of ontology of Petroledge® and literature review of the concepts that are found in the petrography description, we obtain a well founded ontology validated with OLED [1].

For testing purpose, we studied the terminology of *diagenesis* and *microstructure* communities. The ontology presented in our work was done based in UFO-A. Therefore, we used OntoUML(Ontological Unified Modeling Language) as language modeling. It is important to mention that OntoUML does no contain the primitives to model UFO-B(perdurants), thus, we do not model the events involved in the area of diagenesis neither microstructural.

The current work started with the model established in (ABEL, 2001) and the set of terms in the Petroledge database. Based on these structure, we modified and expanded with other terms extracted from different geology bibliographic sources ((TEIXEIRA et al., 2008), (NICHOLS, 2009), (WORDEN; BURLEY, 2009), (LAUBACH et al., 2010), (FETTER; ROS; BRUHN, 2009), (TORABI; FOSSEN, 2009), (FISHER; KNIPE, 1998), (DEHLER et al., 2009)), wikipedia[2]. Mostly of terms were verified with an expert through interviews to verify the consistency of each term metatype. Thus, we present the definitions and ontological analysis of the main terms and the general taxonomy of each of them.

The diagenesis ontology (see Figure 5.3) contains the following main concepts *basin, bioconstructor, matrix, grain, mineral, pore, fluid, cement, rock sample, paragenesis*.

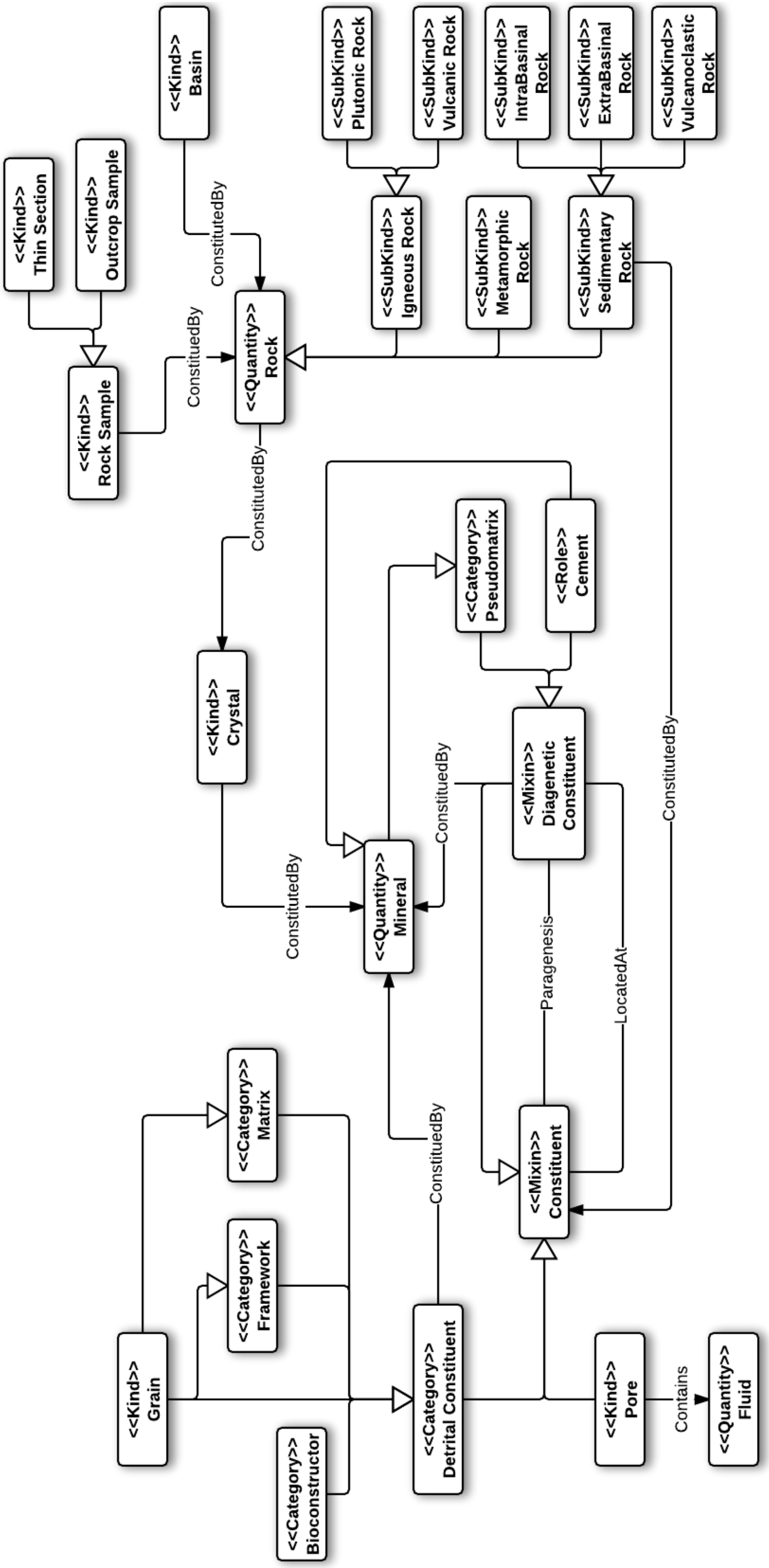
Basin is an area where sediments have been deposited. Ontologically, it has its own identity and it is rigid for all the worlds. The basin is localized in a *Country*, which its meta-type is a *Kind* because *country* implies a region and it obeys the counting principle and it is rigid.

Mineral is a crystalline natural substance represented by a chemical formula. Chemical composition and crystalline structure give the identity criteria to each mineral. It is not countable, which means that lack of unity. Thus, its meta-type is a quantity. In Figure 5.4, we observe the principle taxonomy of minerals following their identity criteria. The main mineral families are carbonates, oxides, sulfides, phosphates, vanadates, halides, sulfates, hydroxides, and the most abundant, silicates, which contains five main sub families: tectosilicate, sheet silicate

[1] https://code.google.com/p/ontouml-lightweight-editor/

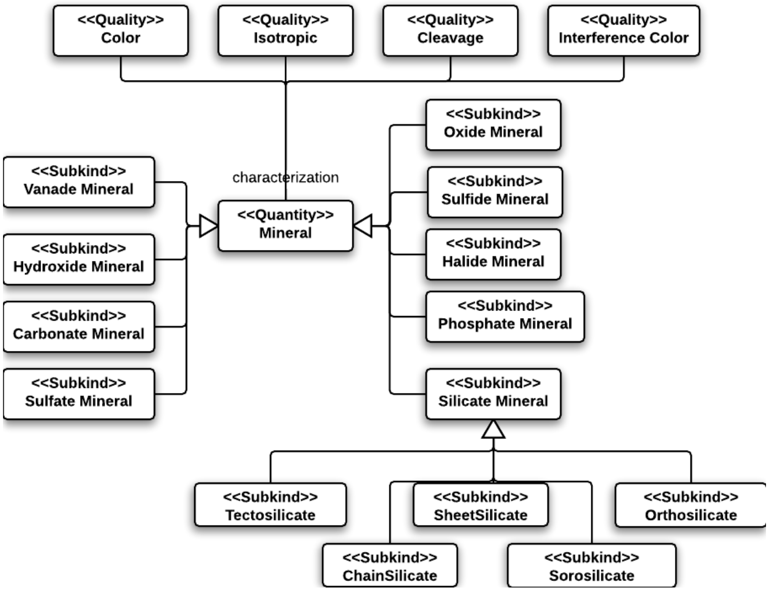
[2] http://www.wikipedia.org/

Figure 5.3: Diagenesis Ontology



Source: The authors

Figure 5.4: Mineral taxonomy


<<Quality>> Color
<<Quality>> Isotropic
<<Quality>> Cleavage
<<Quality>> Interference Color
characterization
<<Quantity>> Mineral
<<Subkind>> Vanade Mineral
<<Subkind>> Hydroxide Mineral
<<Subkind>> Carbonate Mineral
<<Subkind>> Sulfate Mineral
<<Subkind>> Oxide Mineral
<<Subkind>> Sulfide Mineral
<<Subkind>> Halide Mineral
<<Subkind>> Phosphate Mineral
<<Subkind>> Silicate Mineral
<<Subkind>> Tectosilicate
<<Subkind>> SheetSilicate
<<Subkind>> Orthosilicate
<<Subkind>> ChainSilicate
<<Subkind>> Sorosilicate

Source: The authors

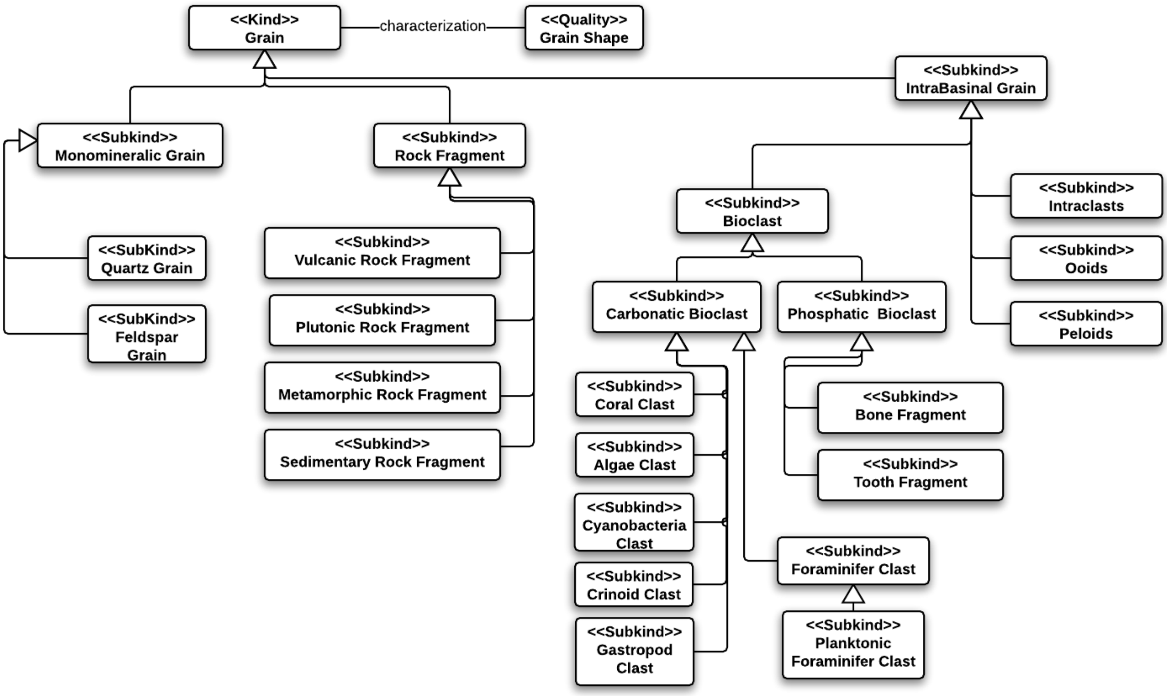
(also called phyllosilicate), chainsilicate, sorosilicate and orthosilicate.

Grain is a 3D spatial region that limits a specific mineral. In the same way that a cattail limits the wood in the space , a *Grain* is related with *Mineral* with the relation of constitution and they are collocated in the space. Thus, *Grain*'s meta-type is kind. Grains are further divided into *Monominerallic grains* (constituted by a single mineral), *Rock Fragments* and *Intrabasinal grains* (see Figure 5.5), which include *Bioclasts*, *intraclasts*, *ooids*, *peloids*. A *Bioclast* is a grain that contains whole or broken pieces of the hard parts of organisms. Most organisms use calcium carbonate minerals to construct their hard parts. Its meta-type is a subkind. *Carbonatic bioclast*, *phosphatic bioclast* are subtypes of *Bioclast*. They inherit its ontological properties. *Foraminifer bioclast*, *Algae bioclast*,*Coral bioclast* are subtypes of *carbonatic bioclast*. They inherit its ontological properties. *Rock Fragment* is made up of multiple grains that are connected on the grain scale. It is a subtype of *grain* and inherits its ontological meta-properties. In the literature, its synonym is lithic fragment. *Sedimentary Rock Fragment*, *Metamorphic Rock Fragment*, *Plutonic Rock Fragment* and *Volcanic Rock Fragment* are subtypes of *Rock Fragment*. They inherit its ontological properties. Grains have the attribute of grain shape.

The *Framework* of sedimentary rocks is formed by grains and minerals that support the rock. A framework can stop being considered a framework because the rock is broken, but the concrete entities that were before a framework (grain and mineral) is still there. Thus, framework will subsume grain and mineral, which are rigids, we suggest modeling as *category*.

Bioconstructor are marine organisms, such as encrusting calcareous algae, that form part of

Figure 5.5: Grain taxonomy


<<Kind>> Grain
characterization
<<Quality>> Grain Shape
<<Subkind>> IntraBasinal Grain
<<Subkind>> Monomineralic Grain
<<Subkind>> Rock Fragment
<<SubKind>> Quartz Grain
<<SubKind>> Feldspar Grain
<<Subkind>> Vulcanic Rock Fragment
<<Subkind>> Plutonic Rock Fragment
<<Subkind>> Metamorphic Rock Fragment
<<Subkind>> Sedimentary Rock Fragment
<<Subkind>> Bioclast
<<Subkind>> Intraclasts
<<Subkind>> Ooids
<<Subkind>> Peloids
<<Subkind>> Carbonatic Bioclast
<<Subkind>> Phosphatic Bioclast
<<Subkind>> Coral Clast
<<Subkind>> Algae Clast
<<Subkind>> Cyanobacteria Clast
<<Subkind>> Crinoid Clast
<<Subkind>> Gastropod Clast
<<Subkind>> Bone Fragment
<<Subkind>> Tooth Fragment
<<Subkind>> Foraminifer Clast
<<Subkind>> Planktonic Foraminifer Clast

Source: The authors

biogenic rocks. Thus, *bioconstructor* represents an essential property of marine organisms and it is rigid because it is broken it continue being a *bioconstructor*. Therefore, its metatype is a *category*.

Matrix is the finer grained mass of material in which larger grains, crystals or clasts are embedded. In the same way, *framework* can cease to exist when a rock is broken, a *matrix* suffers the same phenomenon. Thus, its metatype is *category*.

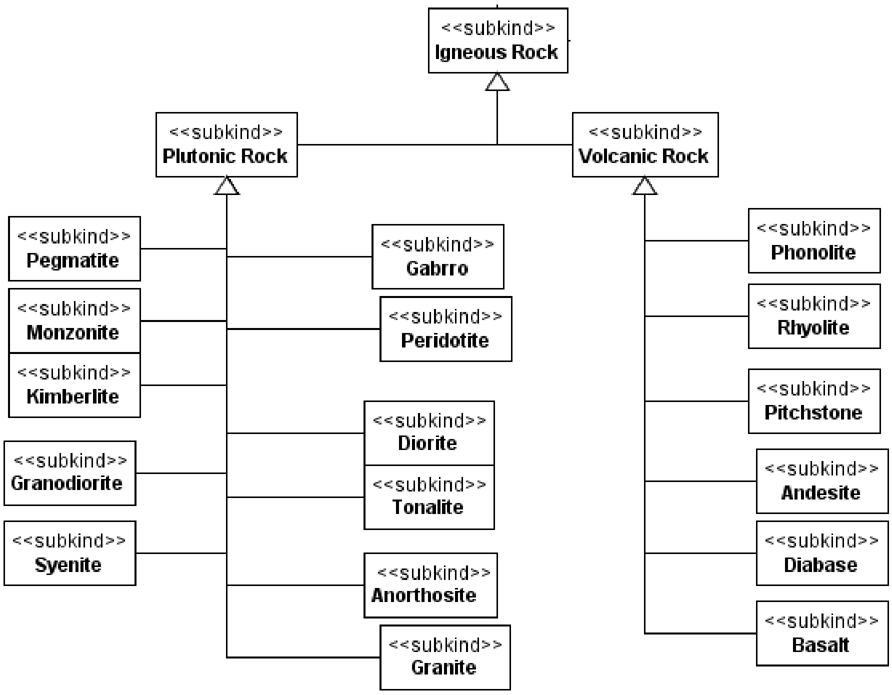
Cement is a mineral originated filling a pore. The term *cement* is anti-rigid. Thus, we suggest modeling as a *role* of mineral and with *Filling* as a *relator* that mediates between *cement* and *pore*. *PseudoMatrix* is a fine-grained material placed among the grains as result of the extreme deformation of some grains through compaction. In the same way that a *matrix* can cease to exist, a *pseudo matrix* can also cease to exist when a rock is broken. We suggest modeling *pseudomatrix* as *category*.

Constituent defines the instances of pore, mineral, diagenetic and detrital constituent that constitute the rock. It subsumes *detrital constituent*, *diagenetic constituent* and *pore*. It has as an attribute the *modifier*, which represents the modifications that the constituent suffered. The modifier is inherent to the constituent, its metatype is *quality*. Constituent's taxonomy can be seen in Figure 5.3. *Detrital constituent*'s meta-type is a *Category* because it groups the detrital characteristic of *grain*, *bioconstructor*,*framework* and *matrix*, which are rigids. *Diagenetic constituent*'s metatype is a *Mixin* because subsumes concepts whose meta-types are *role* and *category*. It is constituted by *mineral*. It subsumes *pseudomatrix* and *cement*. Diagenetic constituents have their *habits* as attribute. *Habits* refers to how crystals are organized in diagenetic constituent. Thus it is a characteristic inherent to the diagenetic constituent having as metatype *quality*. The quality domain of habit is composed of: acicular, blocky, booklet, botryoidal, fibrous, cubic, drusiform, felted, hopper, lamellar, fascicular, vermicule, meniscus, ingrowth, outgrowth, poikilotopic and pendular.

Rock is defined by its internal petrological properties: chemical and mineral composition, texture, porosity, density, permeability. It can not be individualized and does not obey the counting principle inferring that rock's meta-type is a *quantity*. Rock has the following attributes texture, fabric, structure, porosity, density and permeability. The quality domain of texture is composed of grain size, crystal size, crystallinity, sphericity, roundness, sorting. The quality domain of fabric is composed of orientation, support, packing. *Roundness* is a characteristic inherent of the grains. Its metatype is a quality. The quality domain is composed of *well grounded*, *rounded*, *subrounded*, *subangular*, *angular*, *very angular*.

Sedimentary Rock is a type of *rock*. As a consequence *Sedimentary Rock* is an specialization of quantity, called subquantity. In the OLED tool this subquantity is modeled through a subkind. We should not confuse that subkind is only use for specifying a subtype of kind. We can model an object with subkind metatype where the entity that gives identity is a substance sortal. The rock hierarchy is composed of *sedimentary rock*,*igneous rock* (see Figure 5.6) and *metamorphic rock* (see Figure 5.7). *Sedimentary Rock* can be subdivided into *extrabasinal rock*, *intrabasinal*

Figure 5.6: Igneous Rock taxonomy



Source: The authors

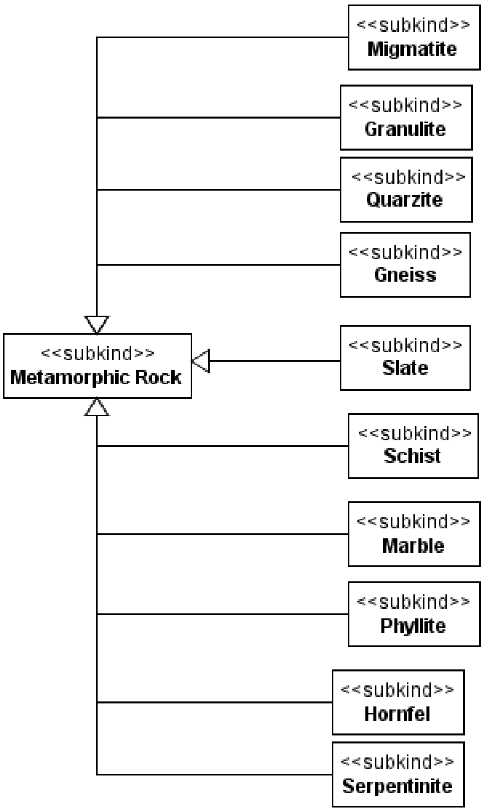
rock and *volcanoclastic rock* . *Intrabasinal rock* can be further divided into *phosphatic rock*, *carbonate rock* , *evaporitic rock*, *ferriferous rock*, *siliceous rock*. *Intrabasinal rocks* are also called *organic chemical rocks*. *Limestone* and *dolstone* are subtypes of *Carbonate rock*(see Figure 5.9).

Igneous rock can be subdivided into *volcanic rock,plutonic rock*. *Schist,marble*, *gneiss*, *slate*, *quarzite,phyllite* are subtypes of *metamorphic rock*. They inherit *metamorphic rock* ontological meta-properties.

Pore system is composed of pores and pore throats. It has its own identity but it can not be countable. Thus, its meta-type is *quantity*. *Pore* is a discrete space within the rock fabric. Although pores may be considered as discrete entities, pore systems are continuous spaces. Due to the definition of discrete, we can identify a pore. Also, it has not relational dependency, so we can assume that its meta-type is a kind. *Pore throat* are the connections between pores. A *Pore throat* is countable and it has its own identity. *Intergranular pore*, *interparticle pore*, *cavern pore*, *vug pore*, *moldic pore,fracture pore* are subtypes of *Pore* and inherit its ontological meta-properties. Figure 5.8 illustrates some subtypes of Pore. The pores contain *fluid* such as liquid and gaseous, hydrocarbons, water and air. Its meta-type is *quantity*. We will only adopt the term Hydrocarbon Fluid in the model. *Oil*, *Gas* and *Condensate* are subtypes of *Hydrocarbon fluid* and inherit its ontological properties, such as being uncountable.

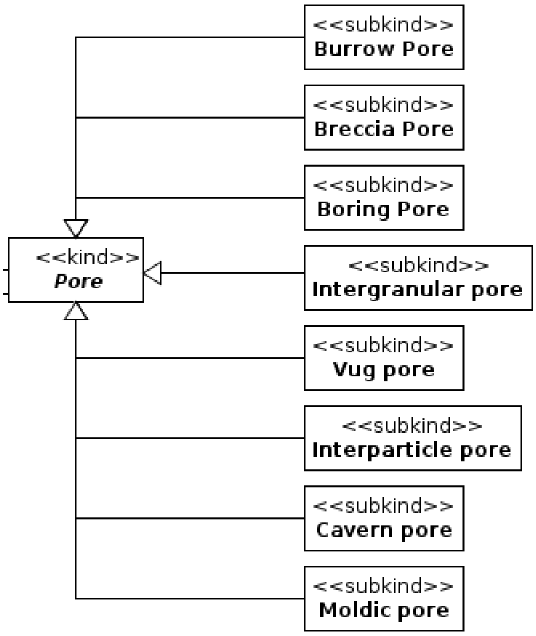
Rock Sample is the central concept that involves the petrographic study. The identity criteria

Figure 5.7: Metamorphic Rock taxonomy



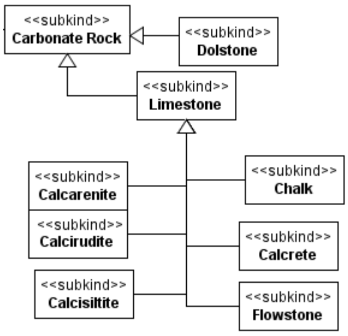
Source: The authors

Figure 5.8: Partial Pore taxonomy



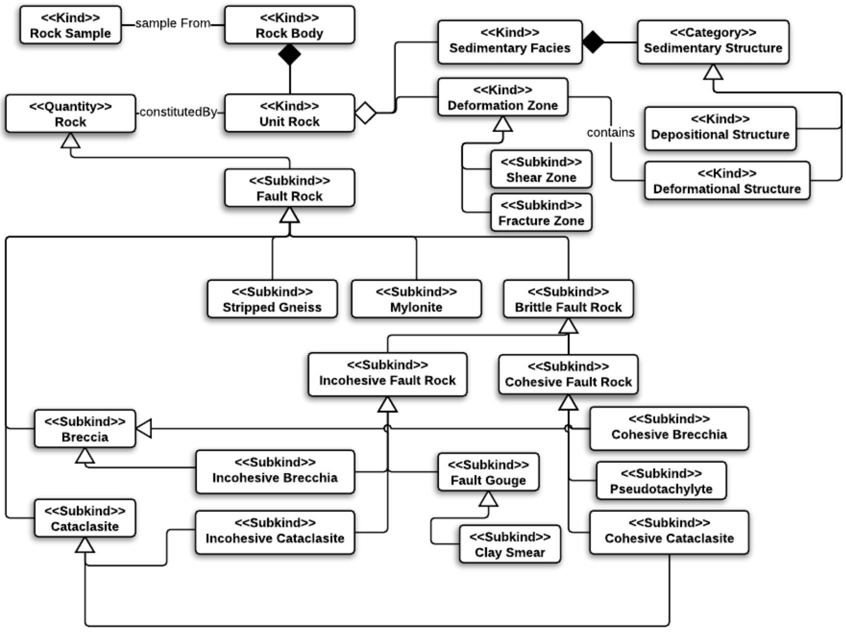
Source: The authors

Figure 5.9: Carbonate Rock taxonomy



Source: The authors

Figure 5.10: Rock taxonomy for the Microstructural Community


<<Kind>> Rock Sample
sample From
<<Kind>> Rock Body
<<Kind>> Sedimentary Facies
<<Category>> Sedimentary Structure
<<Quantity>> Rock
constitutedBy
<<Kind>> Unit Rock
<<Kind>> Deformation Zone
contains
<<Kind>> Depositional Structure
<<Subkind>> Shear Zone
<<Kind>> Deformational Structure
<<Subkind>> Fault Rock
<<Subkind>> Fracture Zone
<<Subkind>> Stripped Gneiss
<<Subkind>> Mylonite
<<Subkind>> Brittle Fault Rock
<<Subkind>> Incohesive Fault Rock
<<Subkind>> Cohesive Fault Rock
<<Subkind>> Breccia
<<Subkind>> Cohesive Brecchia
<<Subkind>> Incohesive Brecchia
<<Subkind>> Fault Gouge
<<Subkind>> Pseudotachylyte
<<Subkind>> Cataclasite
<<Subkind>> Incohesive Cataclasite
<<Subkind>> Clay Smear
<<Subkind>> Cohesive Cataclasite

Source: The authors

of a rock sample includes its topological whole and physical body. The rock sample can be further divided in *rock thin section*, *well core sample* and *outcrop sample*. *Rock Sample* is rigid in all the possible worlds. Its meta-type is kind and the meta-type of their sub types are subkind.

The term *Crystal*, similar to grain, delimits in the space the mineral because the arrangement of the mineral form a crystalline structure. It can be individually visualized. This concept is the constituent of igneous, metamorphic, and carbonate rocks. Crystal can be further divided in *Phenocryst* and *Xenocryst*. *Phenocrysts* usually formed earlier in the crystallization sequence of a magma, however, they can also form by later hydrothermal growth or diagenesis process. *Crystal size* is a property of crystalline rocks, such as some carbonatic (e.g dolostone), evaporitic and siliceous rocks where it is used as quality of the texture domain.

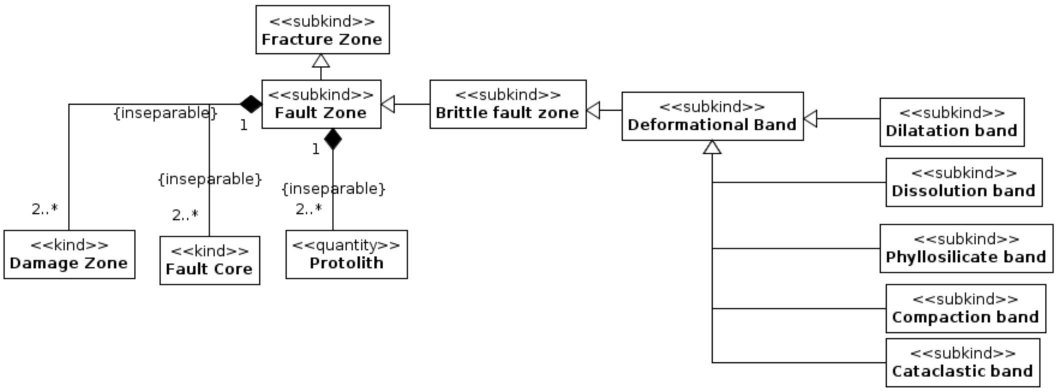
Paragenetic sequence is the result spatial relationship among the grains that reflects the temporal order in which diagenetic process occurred. Thus, we modeled it as the paragenetic relation between constituents, which the endurant results of the perdurant diagenetic processes.

The core microstructural ontology is composed of three main taxonomies, which are the unit rock taxonomy, deformation zone taxonomy and intracrystalline deformation structure.

Rock Body is a term, which meta-type is a kind, because it represents a whole delimited by our mind, it has its own identity and is constituted by *rock*. *Rock body* is composed of *Rock Unit*, which meta-type is kind. Its taxonomy is illustrated in Figure 5.10.

Fault Rock is a specialization of *Rock* and inherits its ontological metaproperties. *Fault rock* has as subtypes *mylonite*, *stripped gneiss* and *brittle fault rock*. These subtypes inherit

Figure 5.11: Fracture Zone taxonomy



<<subkind>> Fracture Zone
<<subkind>> Fault Zone
<<subkind>> Brittle fault zone
<<subkind>> Deformational Band
<<subkind>> Dilatation band
<<subkind>> Dissolution band
<<subkind>> Phyllosilicate band
<<subkind>> Compaction band
<<subkind>> Cataclastic band
{inseparable}
1
{inseparable}
1
{inseparable}
2..*
2..*
2..*
<<kind>> Damage Zone
<<kind>> Fault Core
<<quantity>> Protolith

Source: The authors

the ontological meta-properties of *Fault rock*. *Mylonite* can be further subdivided into *ultramylonite*, *mesomylonite*, *protomylonite*. Those concepts inherit the ontological meta-properties of *Mylonite*. According to the literature, *brittle fault rock* can be further subdivided in *incohesive fault rock* and *cohesive fault rock*, but also there is another classification that groups some of the subtypes of *incohesive fault rock* and *cohesive fault rock*, being *breccia* and *cataclasite*. These terms' meta-types are subkinds. *Incohesive breccia*,*incohesive cataclasite* and *fault gouge* are subtypes of *incohesive fault rock* and inherits its ontological meta-properties. *Cohesive breccia*, *cohesive cataclasite* and *pseudotachylyte* are subtypes of *cohesive fault rock* and they inherit its ontological meta-properties.

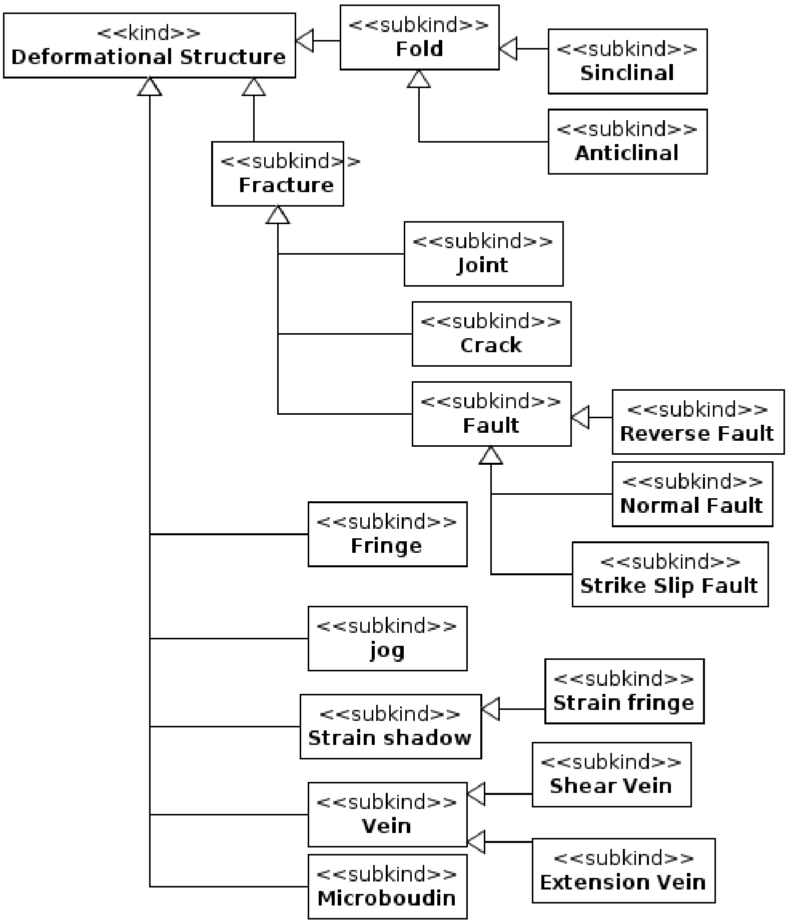
Incohesive cataclasite can be divided in *ultraclasite*, *mesocataclasite*, *protoclasite*. These concepts inherit the ontological metaproperties of *Incohesive cataclasite*.

Clay smear is clay-rich fault gouge. It is a subtype of *Fault Gouge* and inherits its ontological meta-properties. The same analysis is done for *shale smear*.

Deformation zone is considered as kind because is a portion of *rock body* and this portion offers its own identity and counting principles. The relation between these two concepts is *component-Of*. *Deformation zone* is further divided in *fracture zone* and *shear zone*. Its taxonomy is illustrated in Figure 5.11. *Fault zone* is a subtype of *fracture zone*. *Deformational band* is a subtype of brittle fault zone, and inherits its ontological properties. Its meta-type is subkind. *Fault zone* is composed of *protolith*, *fault core* and *damage zone*. *Damage zone* is a volume of deformed rock and it can be delimited by its boundaries. It has its own identity. Thus, its meta-type is kind. *Protolith*'s meta-type is a quantity because of its definition of being unmetamorphosed rock. In the literature, it is not considered as a type of rock, however it has the behavior of substance.

Sedimentary Facies constitute a functional complex that is composed of other functional complexes. (CARBONERA, 2012) modeled an ontology detailing the taxonomy of *deposi-*

Figure 5.12: Deformation Structure taxonomy

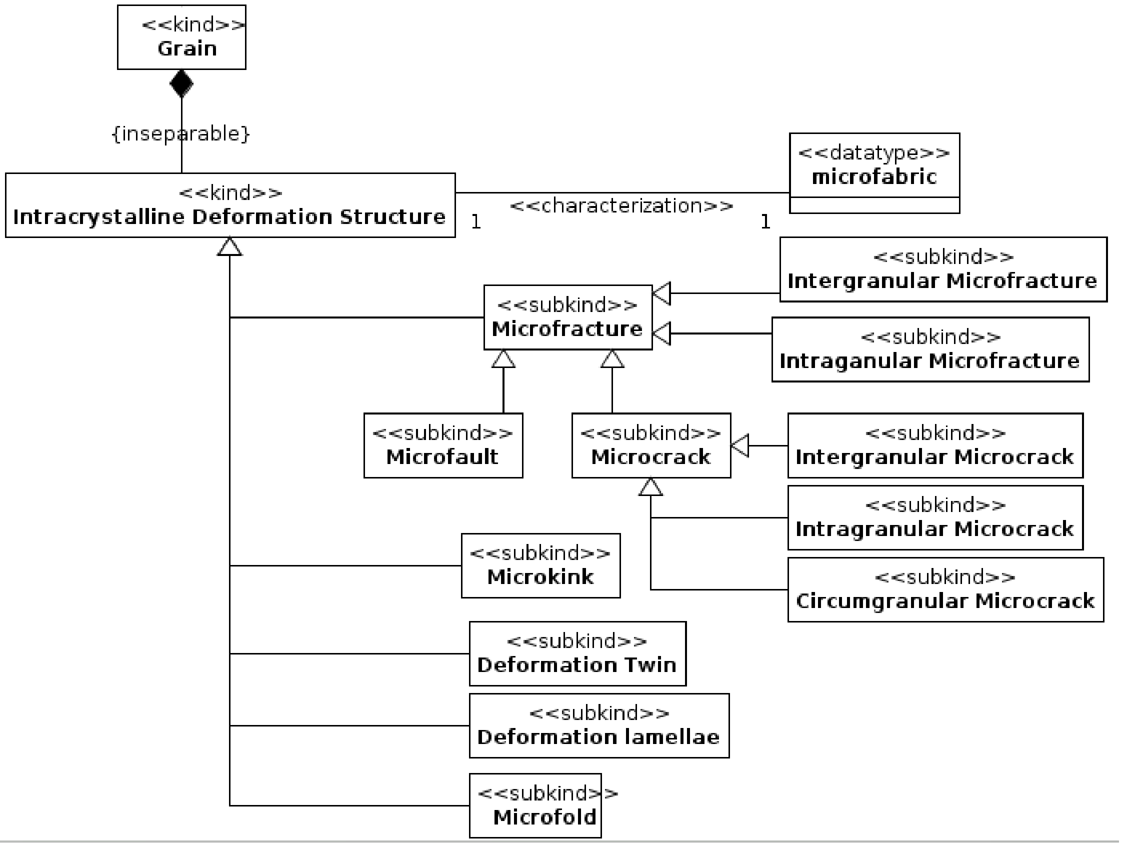


Source: The authors

tional structure and established the relation of *component-Of* with *Sedimentary Facies* having the characteristic that the relation was *inseparable-part*. He gave the metatype of *Category* to *depositional structure* because it contains all types of specific depositional structures. The same analysis can be done for *deformation structure*. *Deformation structure* is the category of *fold*, *fracture*, *vein*, *fringe*, *strain shadow*, which are kinds because each of them represent a structure generated by the effect of deformation process. Its taxonomy is illustrated in Figure 5.12. *Fault*,*joint* and *crack* are subtypes of *Fracture* and inherit its ontological meta-properties. *Shear vein* and *extension vein* are subtypes of *vein* and inherit its ontological meta-properties. *Synclinal* and *anticlinal* are subtypes of *Fold* and inherit its ontological meta-properties.

Finally, *Intracrystalline Deformation Structure* is related to *grains* with a *component-Of* relation. Its taxonomy is seen in Figure 5.13. We applied the same analysis used for deformation structure. Thus, *intracrystalline deformation structure*'s meta-type is a kind. It is further subdivided into *microfracture*, *microfold*, *microkink*, *deformation twin*, *deformation lamellae*, which are subkind.

Figure 5.13: Intracrystalline Deformation Structure taxonomy



Source: The authors

5.2.1 Discussion

Rock, mineral and fluid do not represent topology wholes. They need to be individuated by a second object that delimits their existence. A rock sample has instances, but these are instances of the sample concept and not instances of the rock concept.

In the literature, there are different classifications of sedimentary rocks. Initially, we had used the term *clastic rock* as a subtype of extrabasinal rock. However, the usage's term also consider vulcanoclastic rock. Thus, it will be better to change for siliciclastic rock which will be the best term in this classification.

6 ONTOLOGY VIEW: A PROPOSAL

In this chapter, we describe our approach for *sub-ontology extraction* that is agnostic with respect to the language the ontology is represented in. Firstly, we shall provide some basic definitions in order to allow the understanding of our approach, including the formal characterization of an ontology. Then, we describe the minimal requirements that an ontologically well-founded ontology view must meet and we describe the *sub-ontology extraction algorithm*.

6.1 Well Founded Ontology View

Initially, there is a necessity of downsizing the knowledge presented in a complete ontology. Therefore, an ontology subset should be extracted. But, this subset should be consistent and fulfill ontological requirements. Thus, for doing the extraction the input ontology should be well founded. In the following section, our approach is described.

6.1.1 Basic Definitions

Before describing our approach, we provide a formal description of a well-founded ontology, which will be used in the context of this approach. Let $O_b = (C, R)$ be a base ontology, where $C = \{c_1, \ldots, c_n\}$ is a set of *concepts* and $R = \{r_1, \ldots, r_n\}$ is a set of *semantic relationships* (between concepts). Each $c_i \in C$ is a 3-tuple, such that, $c_i = (cn, def, cmt)$, where cn is the *concept name*; def is the *concept definition* and cmt is the *concept metatype*. On the other hand, each $r_i \in R$ is a 5-tuple, such that $r_i = (rn, rmt, rmp, c_s, c_t)$, where rn is the relation name; rmt is the meta-type of the relation; rmp is the set of meta-properties of relation; $c_s \in C$, is the concept that belongs to the domain of the relation (the source); and $c_t \in C$, is the concept that belongs to the range of the relation (the target).

6.1.2 View

For the purpose of this dissertation, the ontology must be well-founded and complete. Moreover, we define the minimal requirements that the ontology view should have and define the ontology view approach.

Definition 1. *A View is a tuple*

$$V_0 = (id, D, L, O_b, C_v) \tag{6.1}$$

where id corresponds to the identifier of the view, D is the description of the view, L is the ontology language used to implement, O_b is the initial ontology, C_v is the set of concepts that were the starting points for building the view.

6.1.3 Conservation Principles

Our notion of ontologically well-founded ontology view is defined considering certain *principles of conservation* that we have proposed having in mind the meta-properties and postulates defined in Guizzardi (2005). These principles of conservation define which properties must be preserved in a view for it being considered an ontologically well-founded ontology view. Thus, an ontologically well-founded ontology view, in our perspective, is a view that complies with all the principles of conservation that are described below.

Conservation of identity: The view v must conserve the principle of identity of every concept c that it includes. This means that if v includes a given concept c that does not provide its own principle of identity, then the view must include also all the supertypes of c from which c inhered its principle of identity, as well as, all the subsumption relations that are held between these concepts. It is important to notice that, ultimately, the *substance sortal* (*kind*, *quantity* or *collective*) that *provides* the principle of identity to all its subclasses, including c, must be included in the view. For instance, if the target concept is *zeolite* in the example illustrated in Figure 6.1. The conservation of identity will search to the concept that offers identity. In this case, the algorithm will traverse in a bottom-up way until *mineral* concept that is a *quantity*.

Conservation of the existential dependence: If a given concept c_1 is included in the view v, and instances of c_1 are *existentially dependent* on instances of c_2, then must also be included in v, the concept c_2 and the relation held between c_1 and c_2 that is necessary for the conservation of the existential dependence. This involves the analysis of the *part-of* relation that are essential or inseparable because they imply an existential dependence. For instance, if the target concept is *Unit Rock*, the concept *Rock Body* should be included because *Unit Rock* existentially depends of it. Another case, if the target concept is *Porosity*, the concept *Rock* must be included because *porosity* is existentially dependent of *Rock* (see Figure 6.2).

Conservation of relational dependence: If a given concept c_1 is included in the view v, and c_1 is relationally dependent on a relation (materialized through a given *relator*) with the concepts in $\{c_2, ..., c_n\}$, then must also be included in v: the relator r, all the concepts in $\{c_2, ..., c_n\}$ and all relations that are held between the concepts in $\{c_2, ..., c_n\}$, r and c_1 that are necessary for the conservation of the relational dependence. For instance, if the target concept is *Cement*, the concepts *pore* and *filling* should be included in the view because a *mineral* is considered *cement* when *mineral* is filling *pore*. Thus, cement is relational dependent of *filling* and *pore* (see Figure 6.3).

Besides these conservations principles, we also adopt strategies that were adopted in other approaches, including the conservation of the taxonomy and the conservation of attributes.

Figure 6.1: Conservation of identity example

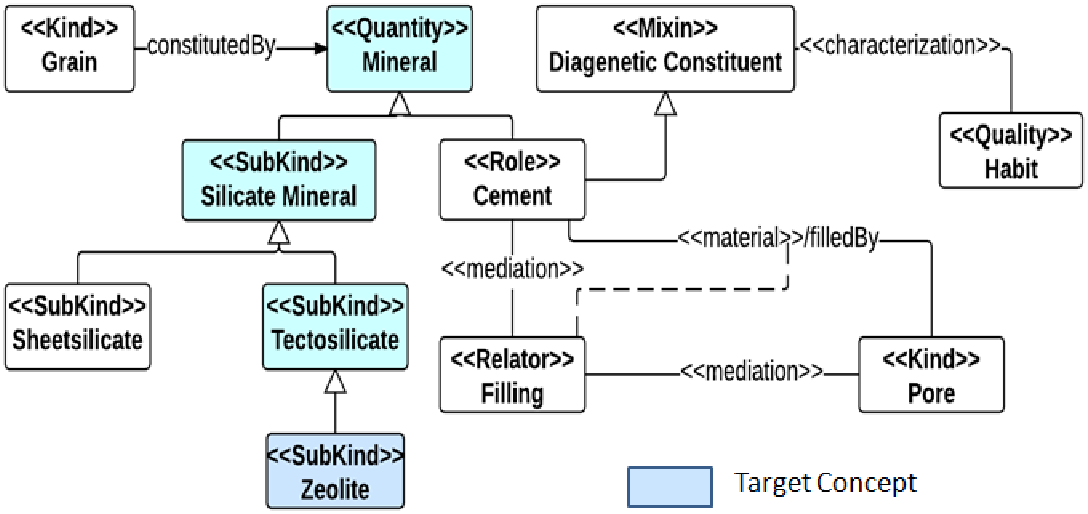


Figure 6.2: Conservation of the existential dependence example

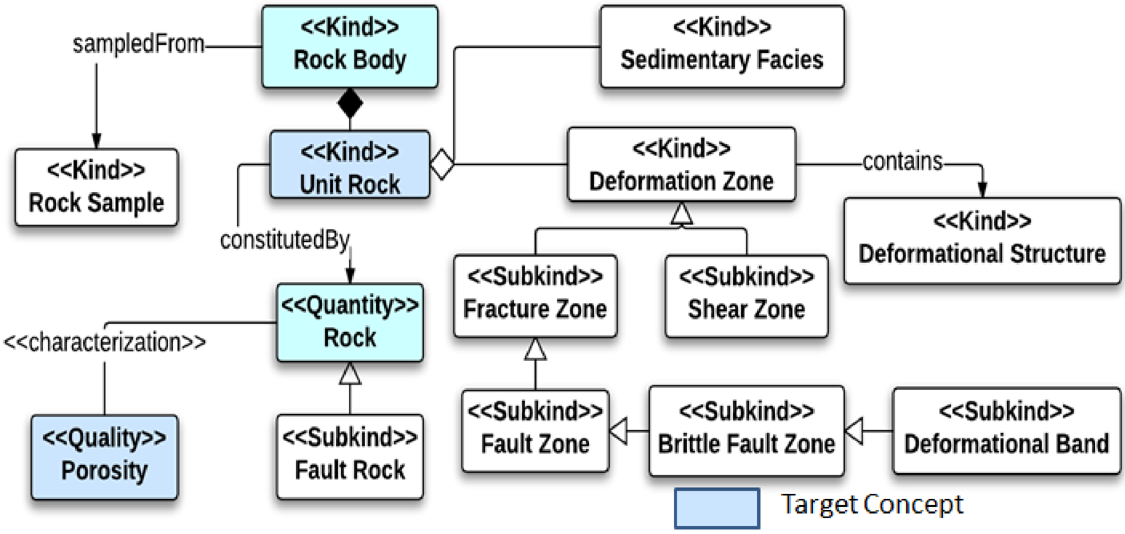


Figure 6.3: Conservation of relational dependence example

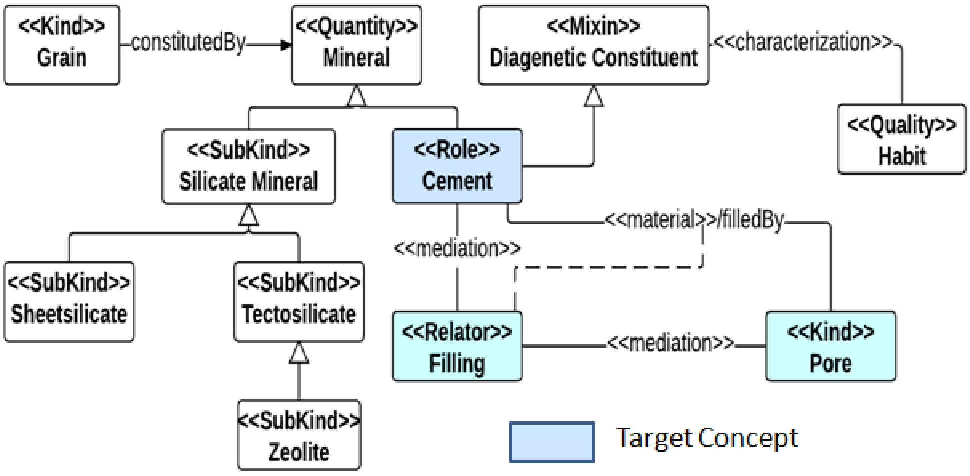
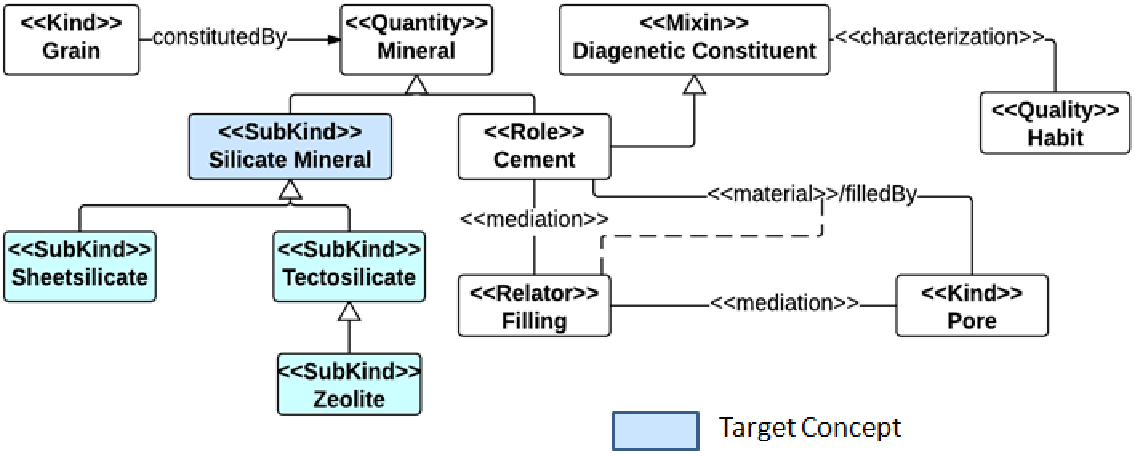


Figure 6.4: Conservation of taxonomy



Conservation of taxonomy: If a view v includes the concept c_1, it must also include all the concepts that are subsumed by c_1. For instance, if the target concept is *Silicate Mineral*, all the taxonomy concepts that it subsumes are included in the view (see Figure 6.4).

Conservation of attributes: If a view includes a given concept c_1, every attribute[1] of c_1 also must be included in v. For instance, if the target concept is *Diagenetic Constituent*, the concept *habit* is included in the view because it is a quality of *Diagenetic Constituent* (see Figure 6.5).

Conservation of formally related concepts: If a view includes a given concept c_1, every concept that is related with c_1 in a formal relation is added. This principle was adopted in Noy and Musen (2009). If the target concept is *Unit Rock*, the concept *Rock* is included

[1] Adopting the UFO, attributes are considered *Quality Universals*

Figure 6.5: Conservation of attributes

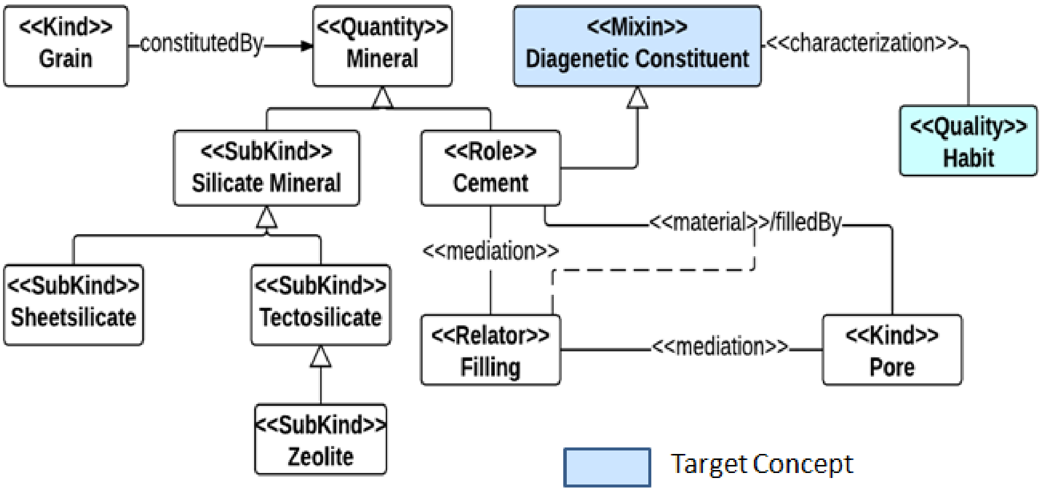
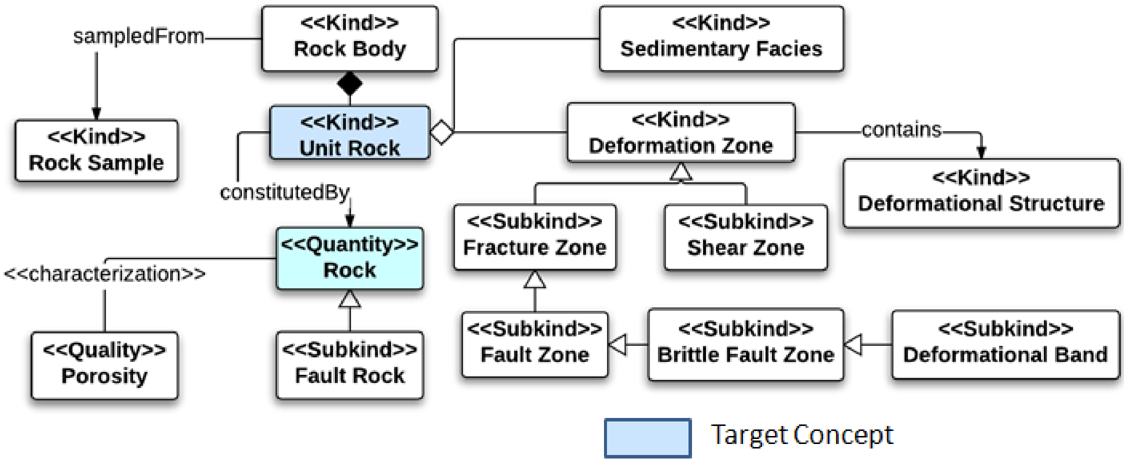


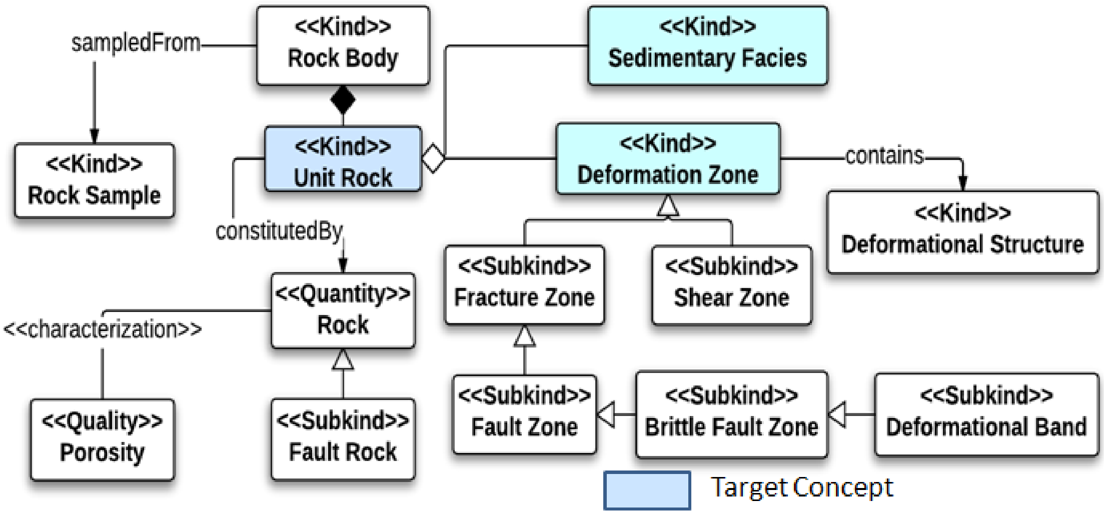
Figure 6.6: Conservation of formally related concepts example



in the view because there is the formal relation *constitute by* between *Unit Rock* and *Rock* (see Figure 6.6).

Conservation of parts: If a view includes a given concept c_1, all the concepts whose instances are parts of instances of c_1 should be included. The part-of relations are also conserved in Bhatt et al. (2004a). However, if the instance of the concept c_1 is a part of another instance. Then, in this case is applied the conservation of the existential dependence. For instance, if the target concept is *Unit Rock*, the concepts *deformation zone* and *sedimentary facies* are included in the view (see Figure 6.7).

Figure 6.7: Conservation of parts example



6.2 Sub-Ontology Extraction Algorithm

We have presented some sub-extraction methods and their details earlier (Chapter 4). Now we describe our algorithm for sub-ontology extraction. Before introducing the algorithm, we define some functions that will be used in the algorithm. The function

$$Rel : rMT \times C \times C \rightarrow R \tag{6.2}$$

maps a relation metatype $rmt \in rMT$, the source concept $c_s \in C$ of a relation, the target concept $c_t \in C$ of a relation to a given relation $r \in R$. For example,

$$Rel(subsumption, c, v)$$

maps to a relation $r \in R$ where c is subsumed by concept v. On the other hand, the function

$$metaType : C \rightarrow cMT \tag{6.3}$$

. maps a given concept $c_1 \in C$ to its metatype $cmt \in cMT$, where cMT is a set of concept metatypes $cMT = \{mt_1, mt_2, \ldots, mt_n\}$. For our approach, we are using the metatypes defined in UFO. Thus, $cMT = \{ROLE, MIXIN, \ldots\}$, which were explained in Section 3. Finally, the relation

$$relMP : rMT \times rMP \times C \times C \rightarrow R \tag{6.4}$$

maps a relation metatype $rmt \in rMT$, a meta-property $rmp \in rMP$, the source concept $c_s \in C$ of a relation, the target concept $c_t \in C$ of a relation to a given relation $r \in R$. For example,

$$relMP(componentOf, essential, c, v)$$

maps to a relation $r \in R$ where c is component of v. Notice that in addition to the relation metatype it is also provided a relation meta-property. In this example, c is a essential component of v.

The algorithm that performs the sub-ontology extraction is denoted as SEL. The SEL algorithm (presented in Algorithm 2) is a recursive algorithm that receives as input the following parameters: the ontology base (O_b), a set of user required concepts ($targets$), a set of relations ($relations$), and the resulting extracted sub-ontology (S_o). At the beginning, $relations$ and S_o are empty. The algorithm analyses each concept in $targets$. For each concept, the conservation principles are applied for ensuring that the result will be an ontologically well- founded ontology view. The conservation principles are applied through the following functions: *conservesTAX*, for conservation of taxonomy; *conservesQUA* for conservation of attributes; *conservesIP*, for conservation of identity principle; *conservesED* for conservation of existential dependence; and *conservesRD*, for conservation of relational dependence. In the main loop, these functions accumulate concepts (in $newC$) and relations (in $newR$) that are necessary for ensuring the defined principles for a given concept c in $targetConcepts$.

Algorithm 2 Sub-Ontology Extraction

```
Require: Well-Founded Ontology
procedure SEL($O_b, targetConcepts, targetRelations, S_o$)
    $S_o.C \leftarrow S_o.C \cup targetConcepts$
    $S_o.R \leftarrow S_o.R \cup targetRelations$
    $newC \leftarrow \emptyset$
    $newR \leftarrow \emptyset$
    for all $c \in targetConcepts$ do // Apply the conservation principles
        $conservesTAX(O_b, c, newC, newR)$ // Conservation of taxonomy
        $conservesQUA(O_b, c, newC, newR)$ // Conservation of attributes
        $conservesIP(O_b, c, newC, newR)$ // Conservation of identity
        $conservesED(O_b, c, newC, newR)$ // Conservation of existential dependence
        $conservesRD(O_b, c, newC, newR)$ // Conservation of relational dependence
        $conservesFR(O_b, c, newC, newR)$ // Conservation of formally related concepts
        $conservesPR(O_b, c, newC, newR)$ // Conservation of parts
        $newC \leftarrow newC - S_o.C$
        $newR \leftarrow newR - S_o.R$
    end for
    if $newC \neq \emptyset$ then
        SEL($O_b, newC, newR, S_o$) // Call recursively
    else
        if $newR \neq \emptyset$ then
            $S_o.R \leftarrow S_o.R \cup newR$
        end if
    end if
end procedure
```

The function *conservesTAX*(algorithm 3) selects the concept taxonomies of a given concept. It takes as parameters the ontology base O_b, a given concept c, and the set of new targets

($newC$) and relations ($newR$) that will store all the concepts and relations that must be included. Intuitively, this function searches all the concepts that are subsumed by c and includes them in $newC$. The subsumption relations that are held between these concepts are included in $newR$. The same process is applied for *conservesPR*, which selects all the parts of a given concept. The difference is that the algorithm analyses parthood relations instead of subsumption relations.

Algorithm 3 Conserve Taxonomy

```
procedure CONSERVESTAX($O_b, c, newC, newR$)
    for all $v \in O_b.C | \exists r = Rel(subsumption, c, v)$ do
        $newR \leftarrow newR \cup r$
        $newC \leftarrow newC \cup v$
        $conservesTAX(O_b, v, newC, newR)$
    end for
end procedure
```

The function *conservesIP* (algorithm 4) analyzes those concepts that do not provide their own principle of identity and searches for concepts that provide the principle of identity. It is used a path variable that is used by the function *findIdentityProvider* (algorithm 5). Essentially, the algorithm traverses the taxonomy in a bottom-up way, trying to find the substance sortals (kind, quantity or collective) that provide the identity to a given concept c. When the algorithm finds such concept, it is stored in the path and the algorithm halts returning true. If the algorithm finds a dispersive universal (mixin, role mixin or category), this means that in this path there is no substance sortal. In this case, the algorithm halts, returning false. The algorithm includes all the concepts in the path between the concept c and its identity provider. The subsumption relations are also included in the path. Finally, the concepts and relations in the path are included in the view.

Algorithm 4 Conserves Identity

```
procedure CONSERVESIP($O_b, c, newC, newR$)
    for all $v \in O_b.C \ | \exists r =$
$Rel(subsumption, c, v)$ do
        $path.C \leftarrow \emptyset$
        $path.R \leftarrow \emptyset$
        $result \leftarrow findIdentityProvider(O_b, c, path)$
        if $result = True$ then
            $newC \leftarrow newC \cup path.C$
            $newR \leftarrow newR \cup path.R$
        end if
    end for
end procedure
```

The function *conservesQUA* (algorithm 6), which conserves the concept attributes, iterates through the *characterization* relation with the concept c, including the respective *quality universals* that characterizes c.

The function *conservesED* (algorithm 7) includes all the concepts from whose instances the instances of *quality universals*, *modes* and *relators* are existentially dependent on. It is

Algorithm 5 Find Identity Provider

```
procedure findIdentityProvider($O_b, c, path$)
    if $(metaType(c) \in \{\mathrm{Subkind, Phase, Role}\})$ then
        for all $v \in O_b.C\ |\exists r = Rel(subsumption, v, c)$ do
            $path.C \leftarrow path.C \cup v$
            $path.R \leftarrow path.R \cup r$
            if $metaType(v) \in \{\mathrm{Kind, Quantity, Collective}\}$ then
                $return\ true$
            else
                $return\ findIdentityProvider(O_b, v, path)$
            end if
        end for
    else
        if $metaType(c) \in \{\mathrm{Relator, Mode, Quality}\}$ then
            if $v \in O_b.C|\exists r = Rel(subsumption, c, v)$ then
                for all $v \in O_b.C\ |\exists r = Rel(subsumption, c, v)$ do
                    $path.C \leftarrow path.C \cup v$
                    $path.R \leftarrow path.R \cup r$
                    $return\ findIdentityProvider(O_b, v, path)$
                end for
            else
                $return\ true$
            end if
        end if
    end if
    $return\ false$
end procedure
```

Algorithm 6 Conserves Qualities

```
procedure conservesQUA($O_b, c, newC, newR$)
    for all $v \in O_b.C|\exists r = Rel(characterization, c, v)$ do
        $newC \leftarrow newC \cup v$
        $newR \leftarrow newR \cup r$
    end for
end procedure
```

also defined the function *findEssentialWhole* (algorithm 8), which traverses the taxonomy in a bottom-up way, trying to find a substance sortal from which a concept is an inseparable part in a *parthood* relation.

Algorithm 7 Conserve Existential Dependency

```
procedure CONSERVESED($O_b, c, newC, newR$)
    if $metaType(c) \in \{\text{Relator, Mode, Quality}\}$ then
        if $metaType(c) \in \{\text{Mode, Quality}\}$ then
            for all $v \in O_b.C | \exists r = Rel(characterization, c, v)$ do
                $newC \leftarrow newC \cup v$
                $newR \leftarrow newR \cup r$
            end for
        else
            for all $v \in O_b.C | \exists r = Rel(mediation, c, v)$ do
                $newC \leftarrow newC \cup v$
                $newR \leftarrow newR \cup r$
            end for
        end if
    else
        if $v \in O_b.C \; | \exists r = RelMP(parthood, inseparablepart, c, v)$ then
            for all $v \in O_b.C | \exists r = RelMP(parthood, inseparablepart, c, v)$ do
                $path.C \leftarrow path.C \cup v$
                $path.R \leftarrow path.R \cup r$
            end for
        else
            for all $v \in O_b.C \; | \exists r = Rel(subsumption, c, v)$ do
                $path.C \leftarrow \emptyset$
                $path.R \leftarrow \emptyset$
                if $findEssentialWhole(O_b, v, path)$ then
                    $newC \leftarrow newC \cup path.C$
                    $newR \leftarrow newR \cup path.R$
                end if
            end for
        end if
    end if
end procedure
```

The function *conservesRD* (algorithm 9) includes all the concepts from which the *roles* and *role mixins* are relational dependent on. Moreover, it includes the respective relations between the concepts.

The function *conservesFR* (algorithm 10) includes the concepts that are related with formal relations from which the target concept is the source of the relation.

Furthermore, we present two variations of the basic approach 1. One variation consists in taking just the taxonomy of the original target concepts. The function *conserves the taxonomy* is called once in order to extract only the target concepts taxonomy. We called this approach the *approach 2* (see algorithm 11). This algorithm calls the sub-ontology extraction target (see algorithm 12). For instance in the example illustrated in figure 6.8 (a), if the target concept is *grain*, the basic approach will include all those concepts. Thus, the second approach will analyze only one time with the conservation of taxonomy and apply to those concepts surrounded by a circle in figure 6.8 (b) the other conservation principles. As result, the sub-ontology will

Algorithm 8 Find Essential Whole

```
procedure FINDESSENTIALWHOLE($O_b, c, path$)
    if $v \in O_b.C \; |\exists r =$
$RelMP(parthood, inseparablepart, c, v)$ then
        for all $v \in O_b.C|\exists r =$
$RelMP(parthood, inseparablepart, c, v)$ do
            $path.C \leftarrow path.C \cup v$
            $path.R \leftarrow path.R \cup r$
        end for
        $return: \; true$
    else
        for all $v \in O_b.C \; |\exists r = Rel(subsumption, c, v)$ do
            $path.C \leftarrow path.C \cup v$
            $path.R \leftarrow path.R \cup r$
            $return: \; findEssentialWhole(O_b, v, path)$
        end for
    end if
    $return \, false$
end procedure
```

Algorithm 9 Conserve Relational Dependency

```
procedure CONSERVESRD($O_b, c, newC, newR$)
    if $metatype(c) \in \{\text{Role}, \text{RoleMixin}\}$ then
        for all $v \in O_b.C|\exists r = Rel(mediation, v, c)$ do
            $newC \leftarrow newC \cup v$
            $newR \leftarrow newR \cup r$
        end for
        for all $v \in O_b.C|\exists r = Rel(material, c, v)$ do
            $newC \leftarrow newC \cup v$
            $newR \leftarrow newR \cup r$
        end for
        for all $v \in O_b.C|\exists r = Rel(subsumption, c, v)$ do
            if $metaType(v) = \text{RoleMixin}$ then
                $newC \leftarrow newC \cup v$
                $newR \leftarrow newR \cup r$
                $conservesRD(O_b, v, newC, newR)$
            end if
        end for
    end if
end procedure
```

Algorithm 10 Conserve Formally related concepts

```
procedure CONSERVESFR($O_b, c, newC, newR$)
    for all $v \in O_b.C|\exists r = Rel(formalAssociation, c, v)$ do
        $newC \leftarrow newC \cup v$
        $newR \leftarrow newR \cup r$
    end for
end procedure
```

not include the taxonomy of *Mineral* and *Intracrystalline deformational structure* depicted in figure 6.8 (b).

Algorithm 11 Sub Ontology Extraction selecting just the taxonomy of the original target concepts

```
Require: Well-Founded Ontology
procedure SELATARGET($O_b, targetConcepts, targetRelations, S_o$)
    $S_o.C \leftarrow S_o.C \cup targetConcepts$
    $newC \leftarrow \emptyset$
    $newR \leftarrow \emptyset$
    for all $c \in targetConcepts$ do
        $conservesTAX(O_b, c, newC, newR)$
    end for
    $newC \leftarrow newC \cup targetConcepts$
    $newR \leftarrow newR \cup targetRelations$
    $selversion2(O_b, newC, newR, S_o)$ // Call a variation of the sub-ontology extraction algorithm that not takes conservation of taxonomy
end procedure
```

Algorithm 12 Sub-Ontology Extraction Target

```
Require: Well-Founded Ontology
// It is not called the function of conservation of taxonomy
procedure SELVERSION2($O_b, targetConcepts, targetRelations, S_o$)
    $S_o.C \leftarrow S_o.C \cup targetConcepts$
    $S_o.R \leftarrow S_o.R \cup targetRelations$
    $newC \leftarrow \emptyset$
    $newR \leftarrow \emptyset$
    for all $c \in targetConcepts$ do
        $conservesQUA(O_b, c, newC, newR)$
        $conservesIP(O_b, c, newC, newR)$
        $conservesED(O_b, c, newC, newR)$
        $conservesRD(O_b, c, newC, newR)$
        $conservesFR(O_b, c, newC, newR)$
        $conservesPR(O_b, c, newC, newR)$
        $newC \leftarrow newC - S_o.C$
        $newR \leftarrow newR - S_o.R$
    end for
    if $newC \neq \emptyset$ then
        $selversion2(O_b, newC, newR, S_o)$
    else
        if $newR \neq \emptyset$ then
            $S_o.R \leftarrow S_o.R \cup newR$
        end if
    end if
end procedure
```

The second variation consists in not analyzing the elements obtained when it is performed the Identity Provider function (algorithm 5). This means that the path obtained to find the identity provider concept is not analyzed by the other conservation algorithms. We called this approach the *approach 3* (see algorithm 13). For this purpose, we define two variables called $newCup$ and $newRup$, which will store the concepts and relations obtained from algorithm 5.

Figure 6.8: Approach 2 example

(a) If it were applied the approach 1 in the target concept, the concepts colored in light blue are included in the view. (b) If were applied the approach 2.

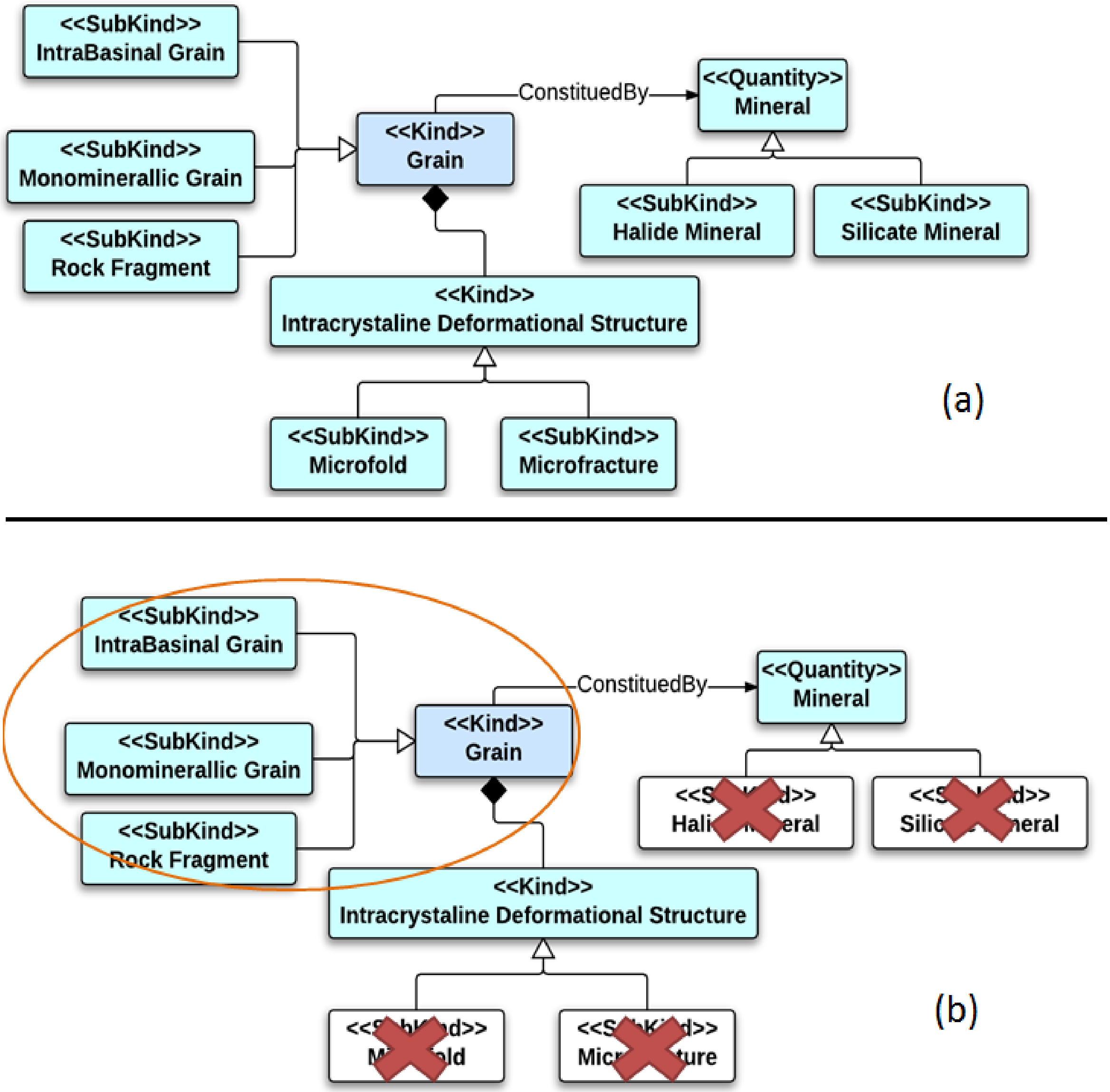
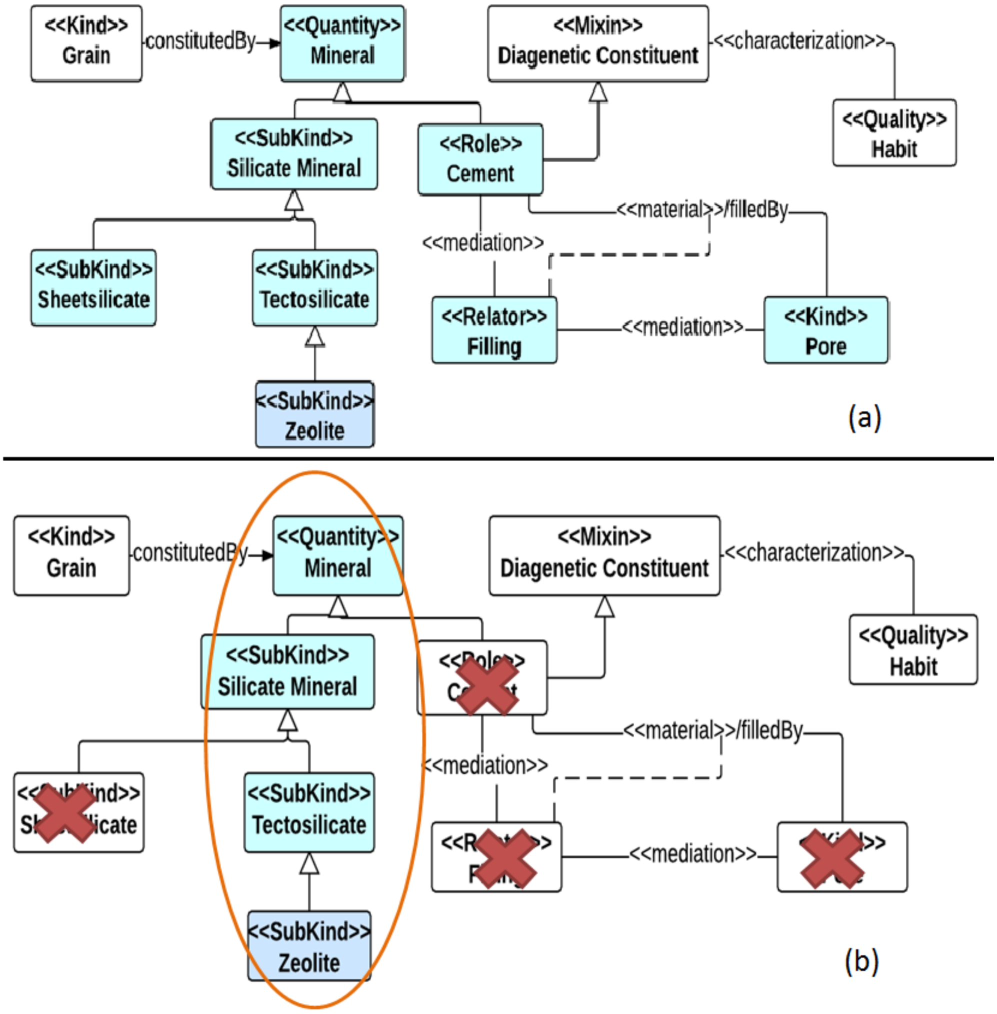

<<SubKind>> IntraBasinal Grain
<<SubKind>> Monominerallic Grain
<<SubKind>> Rock Fragment
<<Kind>> Grain
ConstituedBy
<<Quantity>> Mineral
<<SubKind>> Halide Mineral
<<SubKind>> Silicate Mineral
<<Kind>> Intracrystaline Deformational Structure
<<SubKind>> Microfold
<<SubKind>> Microfracture
(a)
<<SubKind>> IntraBasinal Grain
<<SubKind>> Monominerallic Grain
<<SubKind>> Rock Fragment
<<Kind>> Grain
ConstituedBy
<<Quantity>> Mineral
<<Kind>> Intracrystaline Deformational Structure
(b)

Figure 6.9: Approach 3 example

(a) If it were applied the approach 1 in the target concept, the concepts colored in light blue are included in the view. (b) If were applied the approach 3.



For instance, in the example illustrated in figure 6.9 (a), if the target concept is *zeolite*, the *approach 1* will recover all the concepts colored. But, in *approach 3*, the concepts *sheet silicate*, *cement*, *filling* and *pore* will not be included in the view because those concepts are included as a result of analyzing the path recovered by the conservation of identity. The result is depicted in figure 6.9 (b).

The application offers the flexibility of setting three parameters through the variables *withP* (with Partonomy), *onlyR*(only Rigid Taxonomy) and *withFR* (with formal relation). The ontology engineer can specify if the desired subset should bring the partonomies or not, should include or not the concepts that are connected through the part-of relationship, the rigid taxonomies or non rigid objects, and all formal relations. For instance, we illustrate the modification in the original approach in algorithm 14.

In order to provide the option of rigid taxonomy, we implement a variation of the conserva-

Algorithm 13 Sub-Ontology Extraction Identity

```
Require: Well-Founded Ontology
procedure SELAIDENTITY($O_b, targetConcepts, targetRelations, S_o$)
    $S_o.C \leftarrow S_o.C \cup targetConcepts$
    $S_o.R \leftarrow S_o.R \cup targetRelations$
    $newC \leftarrow \emptyset$
    $newR \leftarrow \emptyset$
    $newCup \leftarrow \emptyset$ // Store concepts of conservesIP
    $newRup \leftarrow \emptyset$ // Store relations of conservesIP
    for all $c \in targetConcepts$ do
        $conservesTAX(O_b, c, newC, newR)$
        $conservesQUA(O_b, c, newC, newR)$
        $conservesIP(O_b, c, newCup, newRup)$
        $conservesED(O_b, c, newC, newR)$
        $conservesRD(O_b, c, newC, newR)$
        $conservesFR(O_b, c, newC, newR)$
        $conservesPR(O_b, c, newC, newR)$
        $newC \leftarrow newC - S_o.C$
        $newR \leftarrow newR - S_o.R$
        $newCup \leftarrow newCup - S_o.C$
        $newRup \leftarrow newRup - S_o.R$
    end for
    if $newCup \neq \emptyset$ then
        $S_o.C \leftarrow S_o.C \cup newCup$
        $S_o.R \leftarrow S_o.R \cup newRup$
    end if
    if $newC \neq \emptyset$ then
        $selAIdentity(O_b, newC, newR, S_o)$
    else
        if $newR \neq \emptyset$ then
            $S_o.R \leftarrow S_o.R \cup newR$
        end if
    end if
end procedure
```

Algorithm 14 Sub-Ontology Extraction

```
Require: Well-Founded Ontology
procedure SELPARAMETERIZED($O_b, targetConcepts, targetRelations, S_o, withP, onlyR, withFR$)
    $S_o.C \leftarrow S_o.C \cup targetConcepts$
    $S_o.R \leftarrow S_o.R \cup targetRelations$
    $newC \leftarrow \emptyset$
    $newR \leftarrow \emptyset$
    for all $c \in targetConcepts$ do
        if $onlyR$ then
            $conservesTAXR(O_b, c, newC, newR)$
        else
            $conservesTAX(O_b, c, newC, newR)$
        end if
        $conservesQUA(O_b, c, newC, newR)$
        $conservesIP(O_b, c, newC, newR)$
        $conservesED(O_b, c, newC, newR)$
        $conservesRD(O_b, c, newC, newR)$
        if $withFR$ then
            $conservesFR(O_b, c, newC, newR)$
        end if
        if $withP$ then
            $conservesPR(O_b, c, newC, newR)$
        end if
        $newC \leftarrow newC - S_o.C$
        $newR \leftarrow newR - S_o.R$
    end for
    if $newC \neq \emptyset$ then
        selParameterized($O_b, newC, newR, S_o, withP, onlyR, withFR$)
    else
        if $newR \neq \emptyset$ then
            $S_o.R \leftarrow S_o.R \cup newR$
        end if
    end if
end procedure
```

tion taxonomy principle taking just the substance sortals, subkind and category (see algorithm 15).

Algorithm 15 Conserve Taxonomy only Rigid

```
procedure CONSERVESTAXR($O_b, c, newC, newR$)
    for all $v \in O_b.C | \exists r = Rel(subsumption, c, v)$ do
        if $metaType(v) \in \{\text{SubKind}, \text{Collective}, \text{Kind}, \text{Quantity}, \text{Category}\}$ then
            $newR \leftarrow newR \cup r$
            $newC \leftarrow newC \cup v$
            $conservesTAXR(O_b, v, newC, newR)$
        end if
    end for
end procedure
```

6.3 Summary

In this chapter, we present the notion of well-founded ontology view, its formalization, the conservation principles that the ontology view should preserve and our sub-ontology extraction algorithm along with its two variants. Also, we provided an approach for obtaining an ontology view. Finally, we finalize the *part-of* relations under the conservation of existential dependence and conservation of parts, but from different perspectives. The first one is applied when the given concept is a part and the relation is analyzed if it contains the essential or inseparable meta-properties. The other conservation principle is applied when the concept is a whole and all concept parts are recovered independently of the meta-properties of the relation.

7 ONTOLOGY VIEW BASED QUERY SYSTEM FOR RESERVOIR PETROGRAPHY

In this chapter, we describe our proposal of RockQuery, a system that uses sub-ontology extraction method to guide the consultation and lets users to analyze data combining visualization and rich user interaction. The ontology applied is described in section 5.2. In order to understand the requirements, we performed a previous study over PetroQuery® System (described in section 2.5). Based on this study and following the interaction design process, different prototypes were proposed and tested with the user. The final prototype was evaluated by 5 participants in a controlled experimental study in order to find if the new interaction design was relevant. Subjective feedback of RockQuery was very positive. In the following, OVUFO visualizer is described.

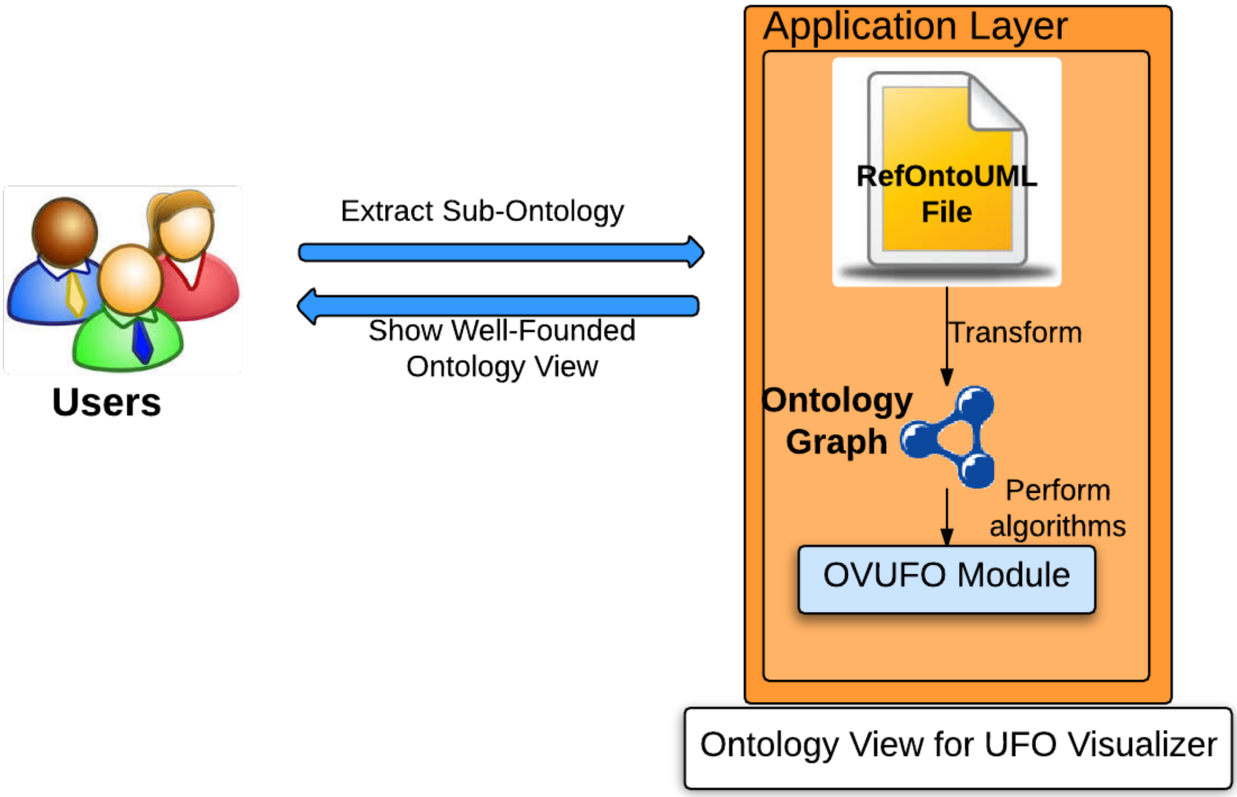
7.1 OVUFO Visualizer

In this section, we describe OVUFO (Ontology View for Unified Foundational Ontology) the interface for visualizing the results of performing the sub-ontology extraction algorithm. OVUFO visualizer (see Figure 7.2) is a tool that allows an ontology engineer to extract well-founded views of a big founded ontology modeled with UFO. The motivation of building this tool was to help the ontology engineer in the establishment of the initial ontology graph that will appear in the interface when the user signs in. However, it can be used independently of domain for modeling purpose.

OVUFO module implements all the proposed theory incorporating the three approaches of the sub-ontology extraction algorithm and offering more flexibility to the ontology engineer to parametrize the algorithm with the options of conserving partonomy, conserving only rigid taxonomy and conserving formal relation. In this way, the ontology engineer will perceive what is the best subset generated for a posterior use.

The interface is designed to be simple to use. It consists of two panels. The visualization panel where is shown the ontology and the operation panel. User starts by opening a file that contains the ontology. Since we work with foundational ontologies, we filter files exported by OLED, which are in the RefOntoUML format. After that, the user enters the target term, select the approach and click in the *Extract Sub-Ontology* button. The tool converts the file into an ontology graph, where nodes are concepts and edges are relations. Each of them has metadata that are used in the sub-ontology extraction algorithms. The architecture of the tool is illustrated in Figure 7.1.

Figure 7.1: Architecture OVUFO Visualizer



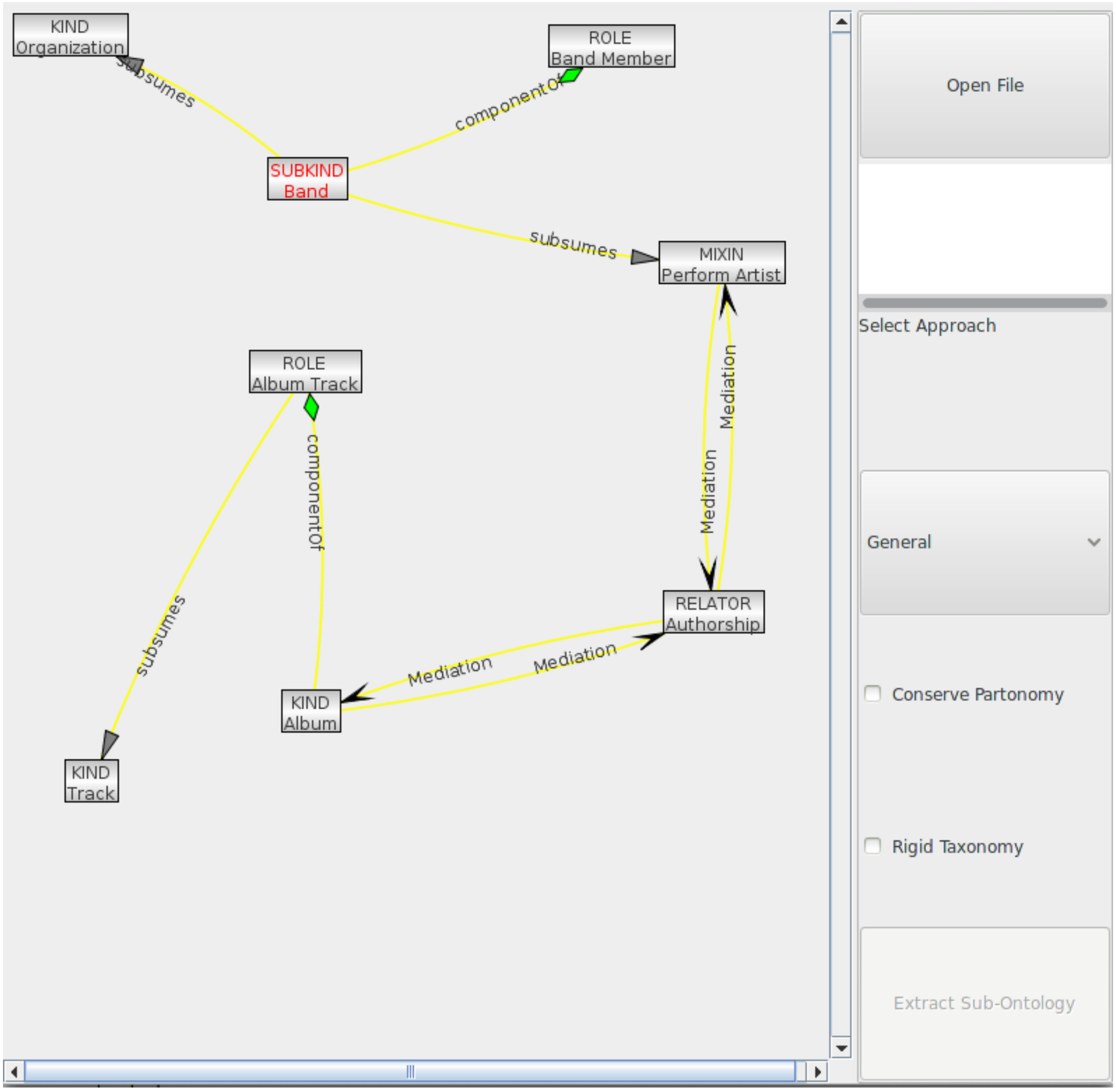
7.2 RockQuery Architecture

The visual query system RockQuery architecture (see Figure 7.3) is composed of a knowledge base, a relational database and Petroledge® system. The knowledge base is materialized in the relational database. Petroledge® system saves rock descriptions in the database. RockQuery uses *OVUFO module* to obtain a sub-ontology having as input a term entered by a user and the knowledge base, which contains the well founded ontology. Then, RockQuery queries to the database through the ontology graph, retrieving the data entered by Petroledge®. Thus, the ontology controls the query definition.

The sequence of activities that user performs to query data are depicted in Figure 7.4. User starts entering a term. Then, RockQuery calls OVUFO module that will return the sub-ontology and the system will show it as graph. Moreover, user selects a node of the graph and RockQuery queries the database passing as parameter the node selected. The query result is shown in a list box of RockQuery. User refines its query selecting the instances that he wants and RockQuery shows the data visualization.

It is important to mention that OVUFO module is configured to run over the second approach, which applies only one time the conservation of taxonomy, parametrized with the option *with Partonomy* assigned with value true and the other two parameters in false. The reason of using second approach with this parameter combination is because in the sub-ontology evalu-

Figure 7.2: OVUFO Visualizer



Select Approach

General

□ Conserve Partonomy

□ Rigid Taxonomy

Figure 7.3: RockQuery Architecture

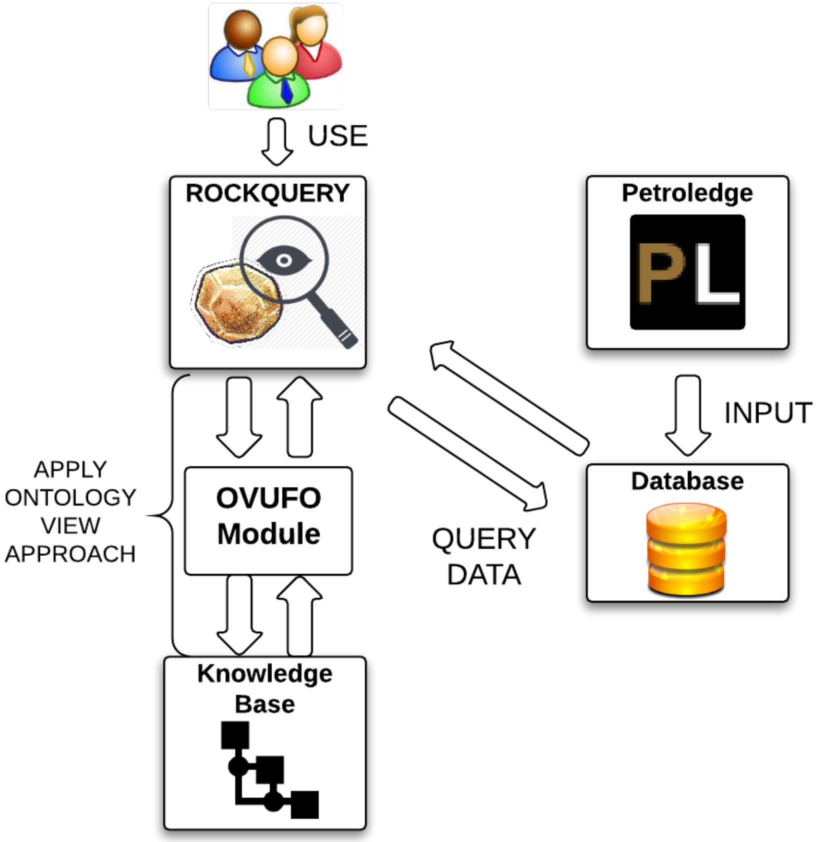
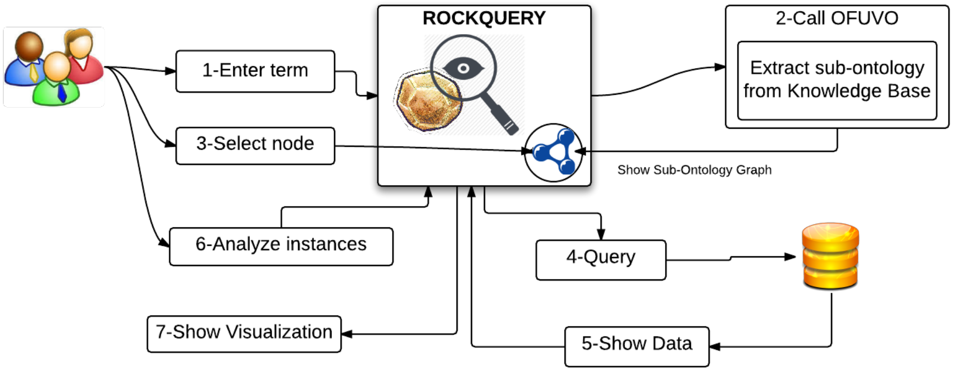


Figure 7.4: RockQuery Activity Diagram



ation method performed over the literature of diagenesis community, explained in section 8.1.1, the recall was the highest.

7.3 Functional Requirements

The principal functional requirement identified was to get the right information by reducing the quantity of terms and showing the principal concepts that users employ in their daily tasks. Also, the importance of filtering support was raised from the empirical observation.

Another functional requirement was to integrate the data analysis area and the query area in a single interface. Also, the necessity of other data visualizations rather than ternary and scatter plot was pointed out by users.

7.4 RockQuery Functionalities Description

The interface consists of three main areas.

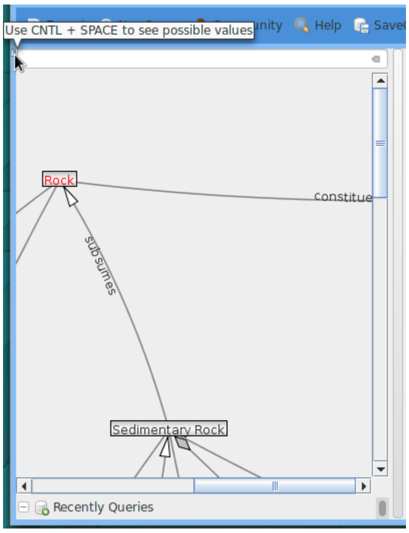
- The Exploration panel (see Figure 7.5): is located on the left side of the interface. It provides several means for the user to get acknowledge of the ontology.

- The Processing panel (see Figure 7.6): provides the selection of the desired instances and if instances were numeric values, the user can perform operations over that.

- The Analysis panel (see Figure 7.7): lets the user visualize the results using different kinds of graphs.

We describe these with more detail in the following subsections.

Figure 7.5: Exploration Panel



7.4.1 Exploration Panel

The exploration panel(see Figure 7.5) consists of three widgets, which are a text box, an *ontology visualization* widget and a *recently Queries* widget . A widget is any object in a graphical user interface that displays information and/or allows the user to interact with an application. The first one is a text box where the user can filter the term that he/she is searching. While the user is writing, there will be an auto-complete function that will suggest possible related terms. Then, the ontology visualization widget shows the ontology according to the input terms enter in the text box offering the capabilities of zooming and panning. Finally, the *Recently Queries* widget shows queries saved by the user.

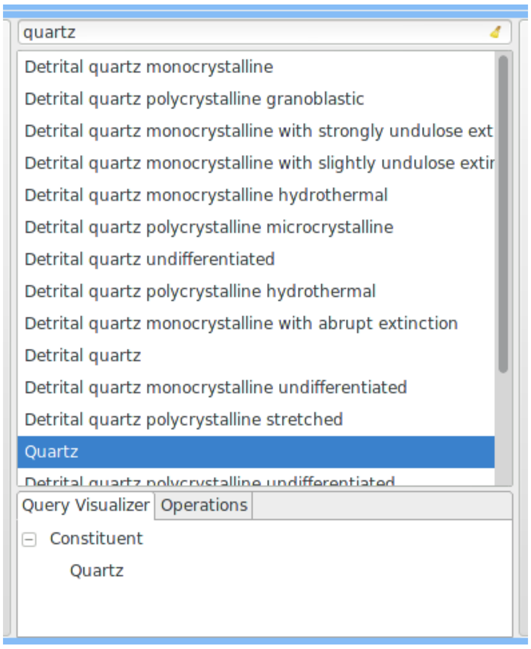
7.4.2 Processing Panel

The processing panel (see Figure 7.6) consists of two sections, a filter text and a list box where it is shown the instances of the selected concept. User can select multiple instances, and while selection occurs, the query visualizer section is updating adding the instance to the tree box. The user can also delete instances in tree box, and the tabular data shown in the analysis panel are updating.

7.4.3 Analysis Panel

Analysis panel (see Figure 7.7) consists of two displays. One is the tabular display, and the other is the data visualization. Both of them are fundamental to facilitate users in their analysis and having them in the same interface is suitable. The quantitative relationship graphs that should be implemented are the stacked bar and the datamap. This will help the geologist to

Figure 7.6: Processing Panel



quartz

Detrital quartz monocrystalline
Detrital quartz polycrystalline granoblastic
Detrital quartz monocrystalline with strongly undulose ext
Detrital quartz monocrystalline with slightly undulose extir
Detrital quartz monocrystalline hydrothermal
Detrital quartz polycrystalline microcrystalline
Detrital quartz undifferentiated
Detrital quartz polycrystalline hydrothermal
Detrital quartz monocrystalline with abrupt extinction
Detrital quartz
Detrital quartz monocrystalline undifferentiated
Detrital quartz polycrystalline stretched
Quartz
Detrital quartz polycrystalline undifferentiated

⊟ Constituent
Quartz

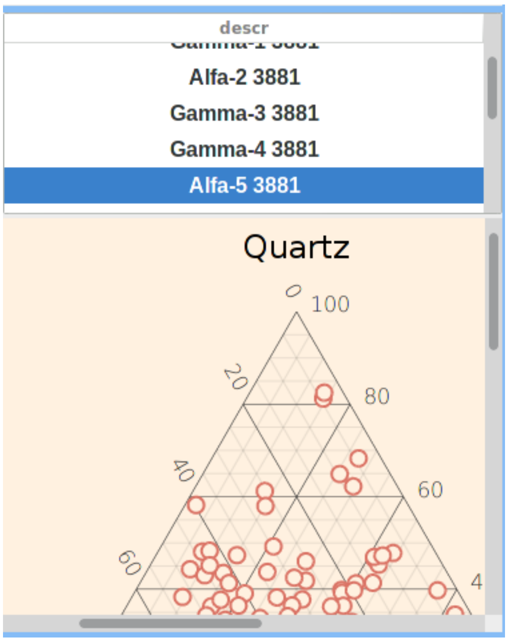
perceive the spatial data. We also included the ternary and scatter plot that PetroQuery® offers.

7.4.4 Application of RockQuery in a case study

In order to evaluate the functionality of the Rockquery interface, this section will describe a sequence of use of the interface through a real petrological case. The initial consultation is *retrieving sample descriptions that include blocky dolomite*. The user begins with the login dialog (see Figure 7.8) where he fills its user name and password. Then, the interface (see Figure 7.9) shows a graph of concepts and relations (ontology) related to the community where the user belongs. After viewing the graph and navigating through it, user can be aware of the concepts and formulate the query. The user selects on one of the nodes that represents the concept and the system lists all instances in the right side. At the same time, the node is added to the query visualizer. The interface allows to hide the panels in order to let more space for the navigation or data visualization. The user can switch from the node inclusion panel to exploration panel as he wishes. The query formulation is an iterative process of selecting a concept and its respective instance. This involves the use of the exploration and processing panel. The feature of text filters in both panels helps in finding the desired term. At the end of the process, the user can save its query.

Following this workflow, our user can iteratively construct its query, refine it and visualize the resulted data. The user's landscape of the areas related to *sensemaking*, following the iterations, is shown in Figure 7.10.

Figure 7.7: Analysis Panel



descr
Gamma-1 3881
Alfa-2 3881
Gamma-3 3881
Gamma-4 3881
Alfa-5 3881

Figure 7.8: Login

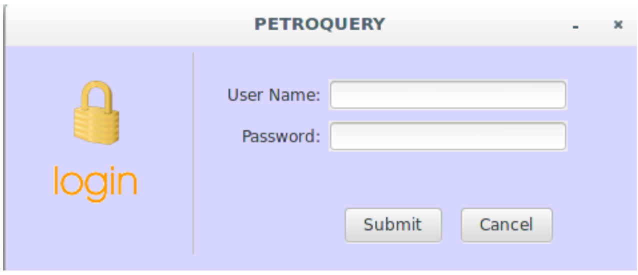


Figure 7.9: Initial Interface after user logged in

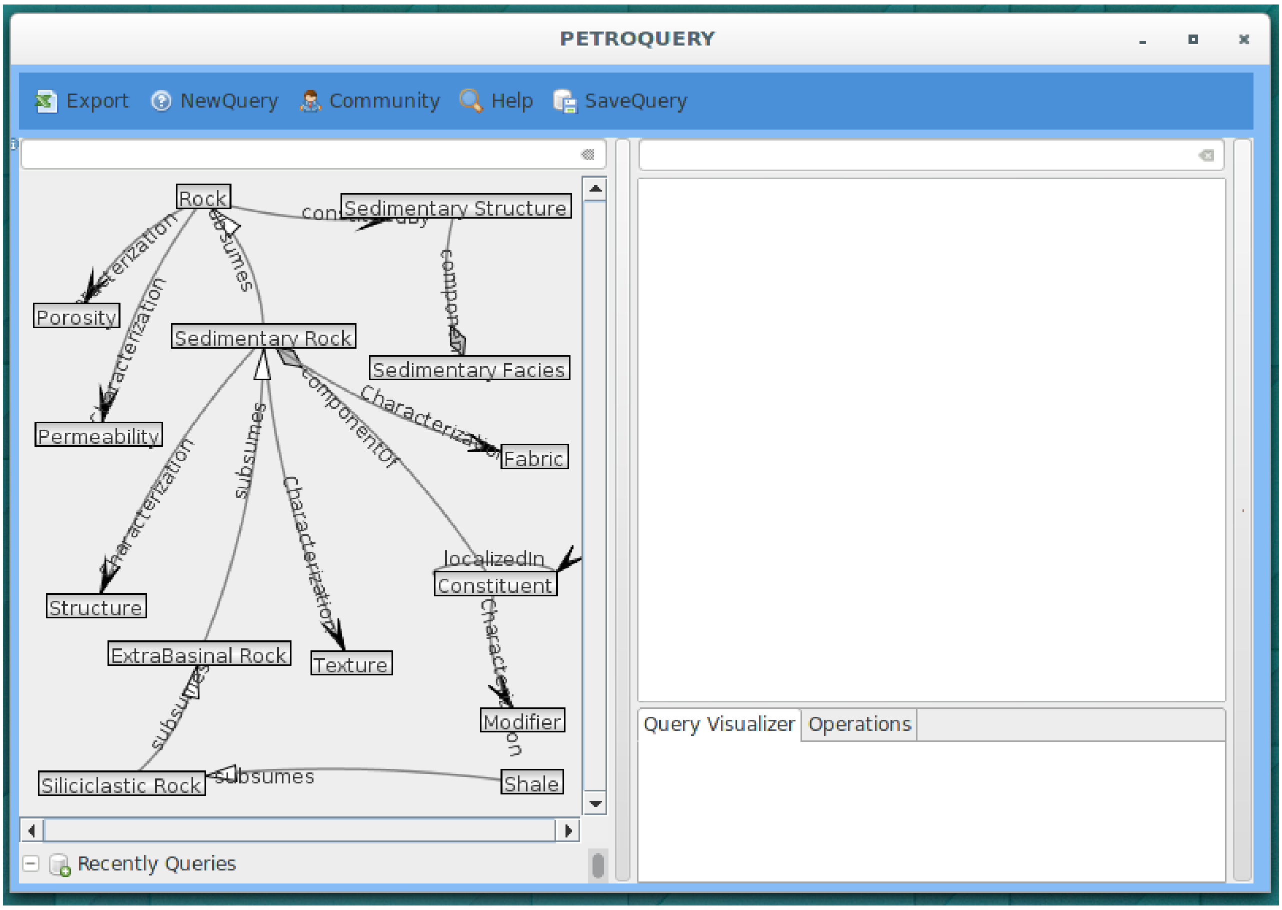
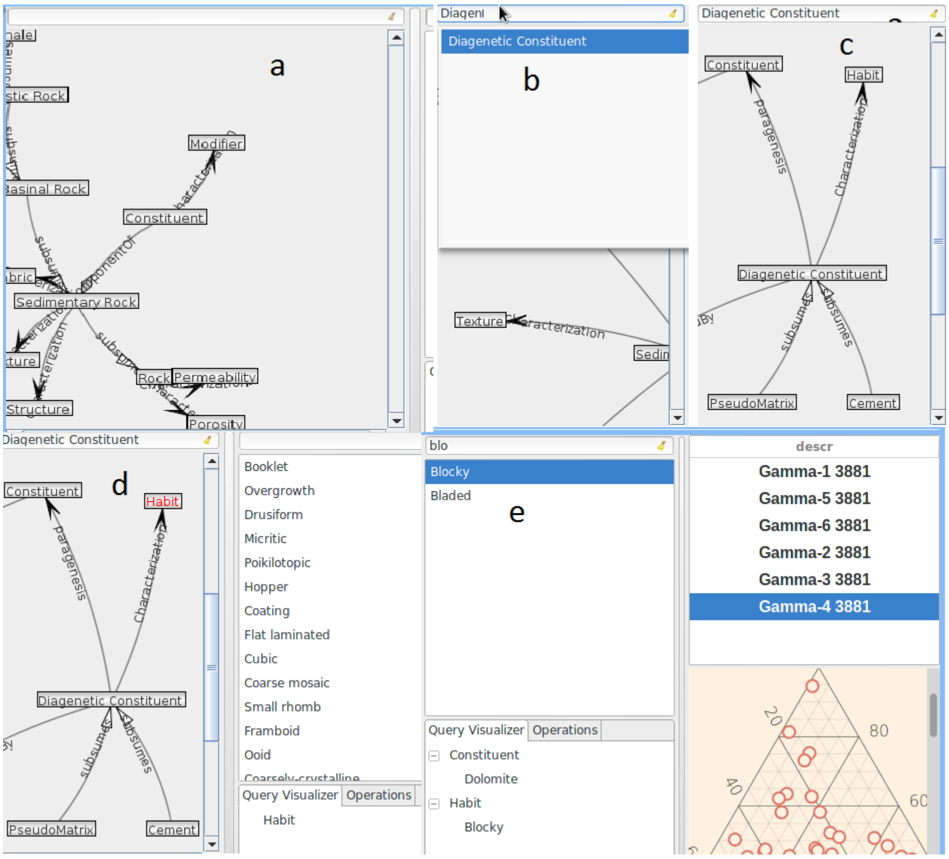


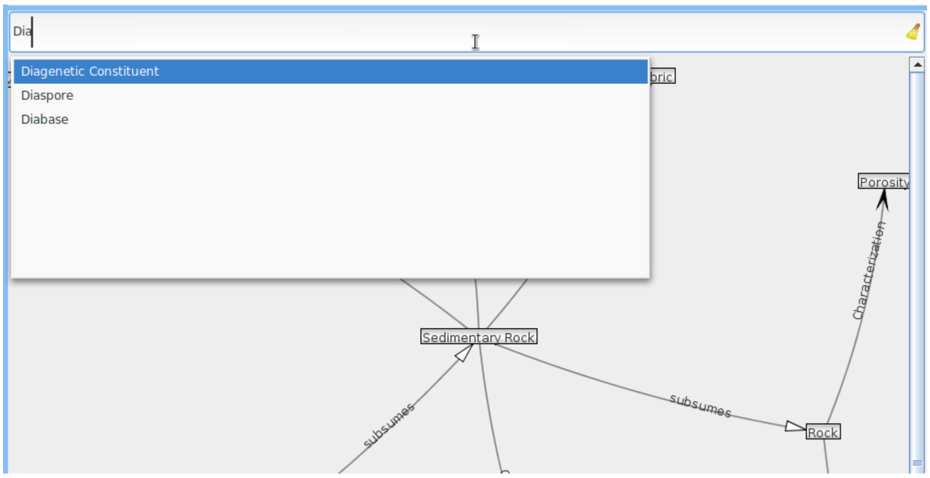
Figure 7.10: Interaction

a)RockQuery's interface at start up, showing the ontology according to his community. b-e) First, user searches a concept in exploration panel and the ontology visualization changes performing the sub-ontology extraction algorithm with the given concept as an input. User finds the concept and selects it; the visualization change the color of the node selected and appears in the processing panel the list of values of that concept. The system offers the capability to filters the data. Through each iteration, the query visualizer is updated. At the same time, it appears the tabular data with a respective data visualization in the analysis panel.



descr
Gamma-1 3881
Gamma-5 3881
Gamma-6 3881
Gamma-2 3881
Gamma-3 3881
Gamma-4 3881

Figure 7.11: Autocomplete Screenshot



7.5 Core Design Rationale

Below we discuss core factors that demonstrate RockQuery's contribution in supporting sensemaking. A key factor in the design of RockQuery was the ontology visualization, which guides the query formulation. Using the structure of the ontology, novel user can better acquire the model of the whole ontology by focusing in a specific part. Therefore, our algorithm is designed to split a well-founded ontology subset of the complete ontology to be visualized. The system retrieves the professional user community and performs internally the sub-ontology extraction algorithm using the target taxonomy approach defined previously in chapter 6. Furthermore, the system autocompletes when user writes in the search area (see Figure 7.11).

Typically, tabular data is used to present query results and the data analysis is performed in a different interface. RockQuery's main difference is that in single interface, user can visualize the tabular data and perceive its visualization. We take care in searching which visualizations will be more pertinent for geologist users. Those data visualizations are ternary graph plot, scatter plot, stacked bar, data-map diagram and balloon diagram.

7.6 Implementation and Development

We have adopted the interaction design process. We started with *understanding and establishing the requirements* through a preliminary PetroQuery® study described in appendix A. Then, we design alternatives presented in appendix B. The final prototype is discussed in this chapter. We also describe the system implementation and the data persistence in the following subsections. Finally, *the evaluation* is presented in subsection 8.2.

7.6.1 Data Persistence

The notion of view to be applied in our case study was conceived through the use of communities. In the community table is established the main terms of the community that will be the input to our algorithm. A user can belong to one or more communities. The terms established for each community was obtained from the experts establishing from one to seven terms.

From Petroledge database, we analyzed the tables that will be necessary for our case study. We add the tables of community, community_user. In order to show the cognitive walkthrough of our proposed system, we implemented the store procedures for mapping the ontology concepts of basin, constituent and description.

7.7 Rock Query Limitations

RockQuery limitations are described according to its three main areas. In the analysis panel, there is a lack of user customization for plotting data visualization. User needs to compare between different visualizations and RockQuery only plots one graphic. In the processing panel, the query visualizer does not visualize the query in a specific query language such as SQL. The list view shows the instances in descending order, but there is no capability of reorganizing this list. In the case of the exploration panel, the ontology visualization does not offer the capability of folding nodes and coloring the paths when a node is selected. The visualization does not contain symbols or icons related to the concept, which in other ontology visualizations enhance the understandability. Also, RockQuery does not deal with synonyms. This means that a user can enter a term that is not in the ontology, but it is a synonym of one of the ontological terms. The query history shows the query names labeled by other users, but the query definition in terms of concepts and instances is not shown.

7.8 Discussion

The limitations of Petroquery® system identified in our study within the community of users include: (1) difficulties in identifying extensions of entities that are collocated in the space, such as, sample (container) and rock (substance); (2) selection of entities and values from a long list of terms, considering that part of the list is unfamiliar for the user; (3) difficulties in the data analysis.

These limitations were solved in our interaction design by modeling the domain with a well-founded ontology and using the sub-ontology extraction algorithm to show the user the concepts that belongs to his community. Furthermore, the filtering feature reduce the search space. Finally, the data analysis is improved with more data visualizations and all the user interaction is center in a single interface.

8 EVALUATION OF RESULTS

This chapter describes the validation approach of the proposed algorithm and the prototype. The effectiveness of the well-founded ontology view that is retrieved by the sub-ontology extraction algorithm is measured through *precision* and *recall*. In order to test our approach, the analysis of two communities (diagenesis and microstructural) was considered and the development of well-founded ontology base. The well-founded ontology base is described in chapter 5. The method developed for test our approach is described in the following section.

Moreover, in the validation of the proposed prototype, we want to know whether it adequately supports users in their tasks and in the environment in which it is going to be used. In addition, functionality tests are needed to verify the robustness of the implementation. Also, this chapter describes the validation approach of the proposed system and the experimentation for identifying the limitations of our proposed environment. Also we discuss the domain ontology used for our case study. We measure the amount of user satisfaction achieved by the outputs of each step in the query process. A major obstacle in this approach is that the notion of satisfiability is highly subjective, and hence difficult to approximately quantify through a subjective judgment of users. The results are presented in section 8.2.

8.1 Generated subset Evaluation

In the literature of ontology engineering, there is no consensual methodology for assessing the quality of the ontology view. Thus, we designed our evaluation method based on an information retrieval metric. The quality of the sub-ontology extraction is measured by the suitability of the generated subset. Thus, the generated subset evaluation consists of proving that one Well-founded ontology view generated for a community *A* has greater f-measure than other well-founded ontology view generated for a community *B*, when the set of terms extracted are from community *A*. By proving this, we verify that our sub-ontology extraction algorithm extracts the terms required for user task at hand. In other words, a view generated for community *X* should fit better the community *X* rather than other community. We assumed that a community conceptualization is materialized in the literature of this community. For instance, it is expected that scientific articles from Sedimentary Stratigraphy are marked by terms related to the Sedimentary Stratigraphy concepts. In this sense, the idea is to measure the fitness of a generated view for a given community X verifying how much the ontology is fitted to the literature of the community X. In this work, this fitness was measure through f-measure, which is a combination of two metrics broadly used in information retrieval precision and recall.

The more important IR metrics in our study are precision, recall and f-measure. *Precision is the fraction of retrieved instances that are relevant. Recall is the fraction of relevant instances that are retrieved.* The relevance is defined as how well information meets user tasks. Precision and recalled are calculated based on true positive (TP), false positive (FP) and false negative

(FN). Precision measures the ratio of correctly found correspondences (true positives) over the total number of returned correspondences (true positives and false positives). This is supposed to measure the *correctness* of the method. Recall also called true positive rate measures the proportion of actual positives(true positives) over the total number of expected correspondences (true positives and true negatives). This is a completeness measure. *F-measure* is the harmonic mean of precision and recall.

In our evaluation, the meaning of TP,FP and FN are those described below.

- TP case was positive and predicted positive. In our evaluation, it is the intersection between community terms and the generated subset terms.
- FP case was negative but predicted positive. In our evaluation, it is the set difference of the generated subset terms from the community terms.
- FN case was positive and predicted negative. In our evaluation, it is the set difference of community terms from the generated subset terms.

The precision, recall and f-measure in our evaluation method (see Figure 8.1) is defined as:

Definition 2. *Given community terms CT, the precision of the subset generated ST is given by*

$$P(ST, CT) = \frac{|ST \cap CT|}{|ST|} \tag{8.1}$$

and recall is given by

$$R(ST, CT) = \frac{|ST \cap CT|}{|CT|} \tag{8.2}$$

and F-measure is given by

$$F(ST, CT) = 2 * \frac{P(ST, CT) * R(ST, CT)}{P(ST, CT) + R(ST, CT)} \tag{8.3}$$

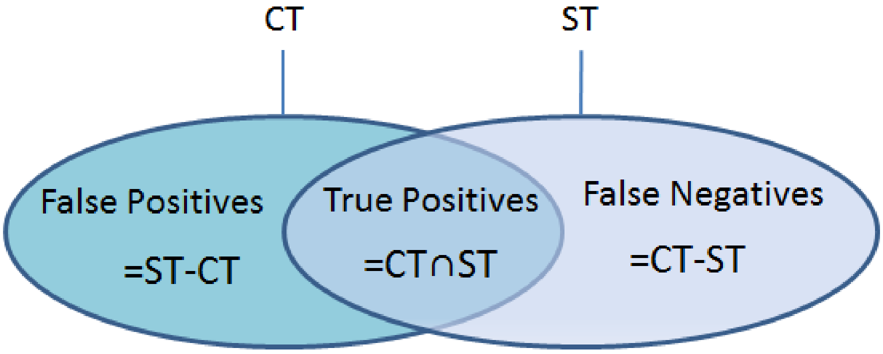
where $|A|$ indicates the cardinality of the set A.

The f-measure is employed to measure the suitability of the generated subset (ontology view). Thus, there are two ontology views *O1* and *O2* for representing community *C1* and *C2*, respectively. Then, it is considered two sets of terms *S1* and *S2*, which corresponds to the sets of representative community terms of *C1* and *C2*, respectively. It is expected that f-measure between *O1* and *S1* ($F(O1, S1)$) is greater than the f-measure between *O2* and *S1* ($F(O2, S1)$). In the same way, it is expected that f-measure between *O2* and *S2* ($F(O2, S2)$) is greater than the f-measure ($F(O1, S2)$) between *O1* and *S2*. In this work, we applied this evaluation approach (see Figure 8.2) considering two communities that are part of the broad geological community: the community of diagenesis and the community of microstructural.

In order to obtain the set of representative community terms, we selected six papers about diagenesis and six papers about microstructural. These sets of papers was selected according to

Figure 8.1: Precision and Recall

ST=Subset terms,CT=Community Terms



Source: The authors

the recommendation made by experts, articles where experts where author and related journals where the expert has published. The selected articles for diagenesis community were (WORDEN; BURLEY, 2009), (HENARES et al., 2014), (CARPENTIER et al., 2014), (MANSURBEG et al., 2012), (GIER et al., 2008) and (KIM; LEE; HISADA, 2007). The selected articles for the microstructural community were (FISHER; KNIPE, 1998), (BUATIER et al., 2012), (MOLLI et al., 2010), (HAERTEL; HERWEGH, 2014), (SCHUELLER et al., 2013) and (BAZALGETTE et al., 2010).

In the next step, the terms extraction from the articles follows the sequence of steps defined in (ABEL, 2001):

- Exclude all common words: prepositions, articles, adverbs and connection verbs.
- Mark all geological terms specific of the domain in study including the terms formalized in the well-founded base ontology.

Furthermore, this process was done manually to guarantee the quality of extraction. In the first step of this sequence, we also exclude the terms that were not exclusive of the diagenesis and microstructural domain. After the second step, we refine the set by excluding the common terms for both diagenesis and microstructural. The result was a list of geological terms by article stored in files labeled with A1, A2, A3, A4, A5, A6 for diagenesis articles and B1, B2, B3, B4, B5, B6 for microstructural articles. It was also generated two files AT and BT that contains all the terms extracted for diagenesis and microstructural, respectively. As a result we obtained geological terms for diagenesis (DT) and for microstructural (MT).

In order to obtain well-founded ontology views, it was taken a well-founded base ontology that covers the domain of diagenesis and microstructural developed by the group. Then, it is applied the sub-ontology extraction algorithm to generate a well-founded ontology view taking as an input key terms established by a representative community expert. These key terms were selected from the base ontology. The terms *detrital constituent*, *diagenetic constituent* and *pore* are the key concepts for diagenesis and the terms *deformational band*, *fault*, *breccia* and *microfracture* are key concepts for microstructural.

The result is an ontology view for diagenesis (DO) and ontology view for microstructural (MO) that, in our evaluation approach, are *O1* and *O2*. Moreover, it is tested the parameters with partonomy (wP) only rigid taxonomy (RT) and formal relation(FR) generating different variations of *DO* and *MO*. Also, the three different approaches of the sub-ontology extraction algorithm (approach 1, approach 2, and approach 3) are evaluated with the parameters combination. Approach 1 consists in performing all the conservation principles. Approach 2 consists in performing only one time the conservation of taxonomy to the target concepts and with this result, applying the other conservation principles. Approach 3 consists in not applying the other conservation principles to the result of the principle conservation of identity.

The next step in our evaluation approach is to compare the f-measure. As depicted in Figure 8.2, we should verify that f-measure ($FDD = F(DO, DT)$) between ontology view for diagenesis (*DO*) and diagenesis geological terms (*DT*) is greater than f-measure ($FDM = F(MO, DT)$) between ontology view for microstructural *MO* and diagenesis geological terms *DT*; in the same way, it is expected that f-measure ($FMM = F(MO, MT)$) between *MO* and *MT* is greater than the f-measure ($FMD = F(DO, MT)$) between *DO* and *MT*.

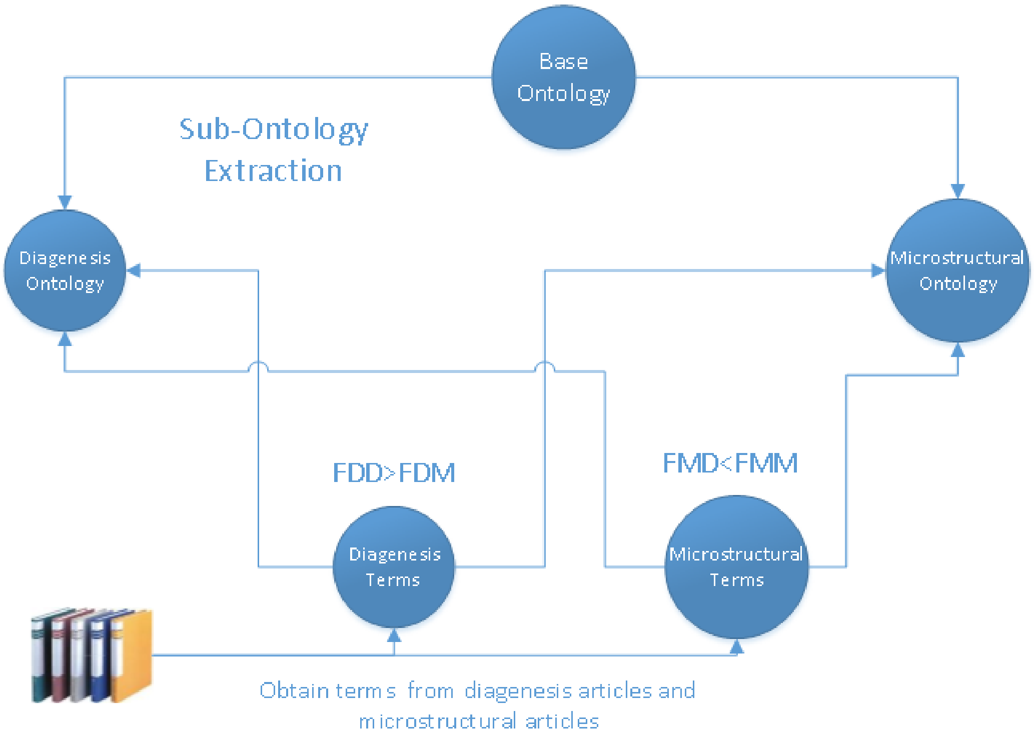
The results of ontology view for diagenesis and ontology view for microstructural is described in the following subsections.

8.1.1 Evaluation of the Ontology View for Diagenesis

For the *approach 1*, we obtained the following results. In the comparisons with the types *wP*, *RT*, *wP-RT* and without parameters (parameter with value false), the P, R, F for diagenesis sub-ontology were greater than the obtained for microstructural sub-ontology. In the rest of comparisons with the other types, R, F for diagenesis sub-ontology were greater than the obtained for microstructural sub-ontology with the exception of file AT. In the case of precision for diagenesis sub-ontology were less than the obtained for microstructural sub-ontology. However, this happens because those types employs the option *FR*. The reason is that the option FR (Formal relation) can take formal relations that are shared with other communities. Recall obtained for diagenesis sub-ontology with all the parameters combinations is always greater than the obtained for microstructural sub-ontology. This means that the sub-ontology generated in the approach 1 for diagenesis community contains more relevant terms than the obtained for microstructural community. In general, *approach 1* results satisfy the fitness of the ontology view for each single file. But in the summarized file AT, only the types *wP*, *FR*, *RT*, *wP-RT* and without parameters (parameter with value false) satisfies the fitness of the sub-ontology (ontology view). The results for the *approach 1* are presented in Table C.1.

Figure 8.2: Evaluation Method

FDD=$F(DT, DO)$, FDM=$F(DT, MO)$,FMD=$F(DO, MT)$,FMM=$F(MO, MT)$
F=F-measure, D=Diagenesis, M=Microstructural, T=Terms, O=Ontology



Source: The authors

For the *approach 2*, for all eight comparisons the P, R, F for diagenesis sub-ontology were greater than the obtained for microstructural sub-ontology. This occurs because when it is applied the principle of taxonomy just once to the target concept, and them analyze with the other conservation principles, the subset obtained is going to depend on the input terms to the algorithm. If these input terms are the key terms of the community, the percentages will be greater. In general, *approach 2* results satisfy the fitness of the sub-ontology for all set of representative terms given in this case study. The results for the *approach 2* are presented in Table C.2.

For the *approach 3*, the results are similar to the *approach 2*. The P, R, F for diagenesis sub-ontology were greater than the obtained for microstructural sub-ontology. However, the difference between f-measures is not as greater as in *approach 2* in the types *FR*, *RT-FR*, *wP-FR*,*wP-RT-FR*. This happens because it was applied the option *FR*. As in *approach 1*, types with the option *FR* can take formal relations are shared with other communities. In general, *approach 3* results satisfy the fitness of the sub-ontology for all set of representative terms. The results for *identity approach* are presented in Table C.3.

8.1.2 Evaluation of the Ontology View for Microstructural

For the *approach 1*, we obtain the following results(see Table C.4). In the comparisons with the types *wP*, *FR*, *RT-FR*,*wP-FR*, *wP-RT-FR* and without parameters (parameter with value false), the P, R, F for microstructural sub-ontology were greater or equal than the obtained for diagenesis sub-ontology. However, the comparison in the type *RT* and *wP-RT*, the P, R, F for microstructural sub-ontology were less or equal than the obtained for diagenesis sub-ontology in the files B1 and B2. The recall was low in all the results. In general, *approach 1* results satisfy the fitness of the sub-ontology for all set of representative terms given in this case study.

For the *approach 2*, results are presented in Table C.5. The comparisons showed that the *approach 2* satisfies the fitness of the sub-ontology. However, the precision in all types for microstructural sub-ontology were less or equal than the obtained for diagenesis sub-ontologyin files B1 and B2. The recall increases in comparison with the *approach 1* and *approach 3*.

For the *approach 3*, results are presented in Table C.6. The comparisons demonstrated the fitness of sub-ontology with five exceptions where the f-measure for microstructural sub-ontology that was less or equal than the obtained for diagenesis sub-ontology in the types *wP*, *RT*, *wP-RT* and without parameters (parameter with value false) in files B2 and BT. Also the comparison in the same types showed that precision for microstructural sub-ontology that was less than the obtained for diagenesis sub-ontology in files B1, B2 and BT.

8.1.3 Summary of results

In the following, we present the proportions of precision, recall, f-measure between the ontology view for diagenesis and the ontology view for microstructural that summarize the results obtained from the evaluation. The expected value of the proportions should be greater than one. This means that the ontology view generated for that community fits community conceptualization materialized in the literature.

The table 8.1 shows the results of proportion of diagenesis over microstructural applied in the file AT. The values should be greater than one in order to satisfy the evaluation approach. However, *approach 1* does not satisfy this condition in the types *RT-FR*, *wP-FR* and *wP-RT-FR*. *Approach 2* obtains better results in precision, recall and f-measure.

In the case of precision, the use of parameter *FR* (formal relation) in *approach 1* produces bad results because this option brings concepts that will belong to microstructural community. But in all the approaches with the different combination of parameters, the proportion of recall of the sub-ontology generated for diagenesis is greater than one that means that the sub-ontology contains relevant terms for the query. The table 8.2 shows the results of proportion of microstructural over diagenesis applied in the file BT. All the approaches satisfy the condition of evaluation approach. In comparison with table 8.1, the values are lower. This means that were few relevant terms retrieved by the different approaches. Furthermore, in *approach 2*, the precision of microstructural sub-ontology was less than diagenesis sub-ontology. This occurs because articles of microstructural community contain terms related with diagenesis and when *approach 2* is applied, only concepts of microstructural community are recovered. Thus, the precision is lower when there is significantly portion of concepts of diagenesis community. A similar situation happens in *approach 3*, the proportion of precision is lower than 1. This occurs because the algorithm will recover concepts that belong only to microstructural community. But the articles contains significantly quantity of terms of diagenesis community causing that the precision of microstructural sub-ontology was less than diagenesis sub-ontology. However, when the algorithm is applied using the parameter of *FR* (formal relation), the proportion of precision is greater than 1. This means that the algorithm recovers concepts that belong to diagenesis community increasing the precision. *Approach 1* obtains for every proportion of precision values greater than 1 because this approach recovers the maximum quantity of terms of three variants containing in the set. Finally, the recall for all the approaches with the different combinations obtain values greater than 1. This occurs because the generated sub-ontology contains great quantity of relevant terms.

In comparison with table 8.1, a better recall was obtained. Thus, depending of the context and application of the geology articles, the taxonomy approach will be the best option. Results over other domains will change depending of the granularity of concepts and relations that exist in the ontology.

In certain cases, the expected result was not obtained because, even if they belong to a

Table 8.1: Proportion D/M over file AT

Type	Measure	Approach 1	Approach 2	Approach 3
	P	1.57	**22.00**	8.80
	R	1.55	**3.88**	**3.88**
	F	1.61	**9.25**	5.29
wP	**P**	1.57	**22.00**	8.80
	R	1.53	**4.43**	3.63
	F	1.52	**9.00**	5.00
FR	**P**	0.70	**15.67**	1.14
	R	1.22	**3.20**	1.38
	F	1.03	**7.60**	1.29
RT	**P**	1.69	**22.00**	8.80
	R	1.24	1.94	**2.82**
	F	1.48	**9.25**	5.29
wP-RT	**P**	1.69	**22.00**	8.80
	R	1.21	2.21	**2.64**
	F	1.40	**9.00**	5.00
RT-FR	**P**	0.71	**15.67**	1.39
	R	1.06	**1.68**	1.38
	F	0.93	**7.60**	1.43
wP-FR	**P**	0.69	**15.67**	1.14
	R	1.15	**3.56**	1.29
	F	0.97	**7.60**	1.23
wP-RT-FR	**P**	0.71	**15.67**	1.39
	R	1.03	**1.88**	1.29
	F	0.90	**7.60**	1.36

Source: The authors

journal of a specific community, some articles used terms from the other community to describe specific situations. The average of terms per articles were of 60 terms for diagenesis articles and 24 terms for microstructural articles.

In addition, we observed that when the *approach 2* is applied the rate of recall increases in comparison with the other approaches. This means that approach 2 recovers more relevant terms. Also, we observed that when the parameter formal relation has the value of true, the precision increases and the recall decreases. In general the precision of our ontology is of 56 % for diagenesis community and 40 % for microstructural community.

Table 8.2: Proportion M/D over file BT

Type	Measure	Approach 1	Approach 2	Approach 3
	P	**1.67**	0.75	0.83
	R	1.62	**3.38**	1.77
	F	**1.75**	1.56	1.31
wP	**P**	**1.59**	0.84	0.74
	R	1.69	**3.69**	1.77
	F	1.67	**1.71**	1.17
FR	**P**	**2.56**	0.72	1.64
	R	1.50	**3.15**	1.42
	F	**1.65**	1.47	1.41
RT	**P**	**1.04**	0.54	0.58
	R	1.38	**5.23**	1.77
	F	1.31	**1.38**	1.13
wP-RT	**P**	**1.04**	0.64	0.52
	R	1.54	**5.62**	1.77
	F	1.28	**1.59**	1.00
RT-FR	**P**	**1.92**	0.52	1.36
	R	1.33	**4.77**	1.42
	F	**1.41**	1.29	1.35
wP-FR	**P**	**2.39**	0.81	1.46
	R	1.46	**3.46**	1.31
	F	**1.61**	1.56	1.33
wP-RT-FR	**P**	**1.86**	0.62	1.21
	R	1.31	**5.15**	1.31
	F	1.39	**1.44**	1.28

Source: The authors

8.2 RockQuery System Evaluation

We follow the Goal Question Metric (GQM) (BASILI; CALDIERA; ROMBACH, 1994) approach to measure the quality of our system. Our usability evaluation of RockQuery system consists of a questionnaire where we evaluate interaction, interface, usefulness and graph exploration. The questions are oriented to measure look and feel, interface layout, ease of use and flexibility, respectively. Those questions are:

Q1 Does the RockQuery enhance the interaction in the consultation process?

Q2 How satisfied are you with the new interface?

Q3 How likely are you in using RockQuery?

Q4 Does graph-based exploration help you understand the data structure for formulating your queries?

Each question has a set of answers that follows the *Likert scale*, with values from one to three. The meaning of the response scale varies according to the question, as presented in Table 8.3. For instance, the question *Q1* with an answer *Not helpful* will have a value of one, question *Q2* with an answer *Neutral* will have a value of two, question *Q3* with an answer *Frequently* will have a value of three.

Table 8.3: Meaning of the response scale varies according to the question.

Question	**Answers**		
	1	**2**	**3**
Q1	Not helpful	Neutral	It improves the interaction
Q2	Unsatisfied	Neutral	Satisfied
Q3	Never	Sometimes	Frequently
Q4	No	Neutral	Yes

Source: The authors

We evaluated five users. User1 *U1* and user2 *U2* are master's students in Geology, one is specialized in carbonate rocks and the other is specialized in siliciclastic rocks. User3 *U3* and user4 *U4* are experts in sedimentology and diagenesis, respectively. The last user *U5* is expert in stratigraphy. *U1* and *U2* use frequently Petroledge and Petroquery, an average of six and two hours per day, respectively. *U3* uses Petroledge one hour and PetroQuery® two hours per day. *U4* uses Petroledge one hour and PetroQuery® ten minutes per day. *U5* uses Petroledge two hours and PetroQuery® thirty minutes. *U1* and *U2* are 22 years old and they can be considered normal expert users. *U3* has 44 years old and *U4* has 57 years old; and both of them are experts in the domain, however, *U4* is not a normal user of PetroQuery® because he does not use frequently. *U5* has 43 years old and can be considered a soft user, not an expert in the domain of diagenesis. In summary, they are all petroleum geologists who have different ages and levels of experience and work focus. All of them have previous experience in using PetroQuery®. The majority of them have basic notion in computer science. *U3*, *U4* and *U5* have notions of ontology. Table 8.4 illustrates the user characterization described above.

In our evaluation procedure, firstly, the users use the RockQuery for performing some queries. After, the users experience was measured through a questionnaire. The following queries were used in the test:

- P1. Select the *Espirito Santo* basin
- P2. Select the thin sections with the constituents' *quartz*, *zeolite* and *sillimanite*, which are localized in the framework or in burrow pore.

Table 8.4: Users Characterization

	Age	**Expert**	**Specialization**	**Petroledge Usage (Hours per day)**	**PetroQuery® Usage (Hours per day)**	**Ontology Notion**	**Sex**
U1	22	No	Geologist	6	4	No	M
U2	22	No	Geologist	6	4	No	M
U3	44	Yes	Sedimentology	1	2	Yes	F
U4	57	Yes	Diagenesis	1	0.01	Yes	M
U5	43	Yes	Stratigraphy	2	2	Yes	F

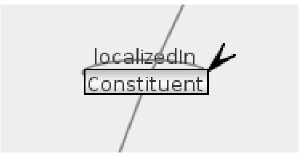
Source: The authors

- P3. Select the diagenetic constituents with blocky habit

The purpose of applying these queries was to evaluate the interaction of the users with the tool in a task of query formulation, where the user had to select a different number of concepts with its instances. For instance, in query *P1* the user is asked to select one concept and one instance. Here the user should use the filter area, because the concept *basin* has many instances. In the query *P2*, the user need to select one concept (*constituent*) related with another (*pore*). In the query *P3*, it is tested in the user is able to select one attribute (*blocky habit*) of a concept (*constituent*). The usefulness and the interface are measured through simple questions presented to the user after their experiences with the RockQuery, asking how they feel about the experiences with the tool and if they liked to use it.

According to the evaluation test, users find the concept and perform the query in a minor time that using the whole ontology. However, the visualization of the ontology does not facilitate the choice of the relationship when it relates the same concept. For instance, all users could not perform the question *P2* because the statement contains the relationship *localized in* and this relation in the layout (see Figure 8.3) was difficult to select.

Figure 8.3: Problem Detected in Graph Layout



Source: The authors

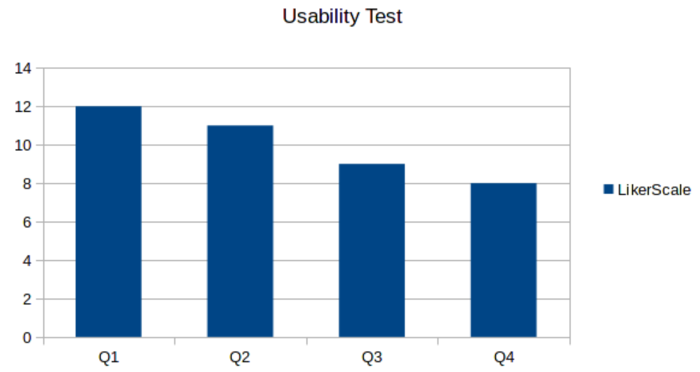
Furthermore, most of the users reported that the question *P2* was not easy to formulate because the relations *in framework* or *filling burrow pore* are instances of *locatedIn* relation, which in the ontology visualization is not clearly visible. The other questions were answered without problems.

According to the majority of users, the ontology could help for novice users, but it makes slower the process for advanced users, because they already know the hierarchy. *U1* made the following observation: ... *the use of the ontology is not clear for me, I prefer to use lists to select the concepts...*. *U2* points out that ...*the ontology contains the principal taxonomy, however, it could be reduced to show just the leaf nodes. Almost for me that I am advanced user, I know that hierarchy...* . As *U1*, *U2* is used to deal with lists and also points out that ...*maybe the use of tree table view will be better...* . *U3* mentions that ...*the use of shortcuts in some interactions will be better than the use of mouse*. *U3 and U5* likes the ontology as an innovation to perform the query. Furthermore, *U4* mentions that ... *the use of filters helps in the formulation of the query, but the ontology graph confuses me. I prefer the use of lists*. Finally, all users note the importance of having an analysis section in the same interface.

Thus, the evaluation suggests the necessity of an alternative to the graph visualization widget for displaying the ontology or more visualization operations that will help to interact with the graph like folding and expanding the node. As a future work, we will experiment the use of our approach with a tree table view in the exploration panel.

Questionnaire results (see Figure 8.4) show that RockQuery enhances the user interaction receiving the majority of points in the scale. Users liked the new interface prototype, but are neutral to use our system because of the graph visualization plugin. The graph-based exploration should be improved with a tree table view and with a better graph layout. Other types of usability evaluation were not applied due to the lack of users availability.

Figure 8.4: Questionnaire Results using the Likert scale



From 1-5 user are not agree with the question, 6-10 user are neutral,11-15 user agree with the question

Source: The authors

Finally, we present a comparison table of our system (see Table 8.5) and some visual query systems (PetroQuery®, VisualSPEED, Graphical RQL and Optique) over terminologies or ontologies described in section 2.5. The criteria contains the following items:

- *Result visualization*: Evaluates if the interface has a section for *data visualization*. This means different diagrams that help user to understand the data.
- *Query History*: Evaluates if the interface has a section for query history. It is common in search interface to have a query history because it helps users to reuse a previous query.
- *Ontology visualization*: Evaluates if the interface has a section for the ontology visualization. An ontology visualization will help in the exploration and navigation of concepts to formulate the query.
- *Text Filter*: Evaluates if the interface has *text filters*. Text filters are important in search interface. It is the widget where the user informs a keyword for searching it within a list of terms.
- *Query Visualizer*: Evaluates if the interface has a section to visualize the query formulation.
- *Knowledge Adaptation*: Evaluates if the system allows presenting in the interface with only the amount of information that is necessary for the task at hand.
- *Codification*: Evaluates if the interface has a section where the user enters his query using any syntax of a query language.

The results shown that RockQuery is the only one that contains a section for data visualization in a single interface. The other systems provide in a separate interface. Also, our system and PetroQuery® have *query history* feature that the other VQSs not have. RockQuery and VisualSPEED are the VQSs that offer a panel for ontology visualization. VisualSPEED's visualization use icons in the representation of concepts. Optique uses filters for searching concepts and attributes. RockQuery uses filters to either navigate the ontology in the exploration panel; or to filter the terms in the processing panel. Moreover, no VQS tries to adapt the information shown in the interface to the user requirements. That is, if the ontology has a huge amount of terms, the interface will list all of them. But, RockQuery try to adapt the information applying the sub-ontology extraction algorithm having as input the key terms of the community where user belongs. Optique and VisualSPEED offer the capacity to codify the query in SPARQL. In summary, our system can be improved by using icons in the concepts. As future work, our system should let advanced users to codify its query. VisualSPEED, Graphical RQL and Optique consult over semantic web data stored in a RDF or OWL database. RockQuery and PetroQuey consult over relational data controlled by an ontology.

Table 8.5: Comparison of four Visual Query Systems.

Criteria	**PetroQuery®**	**VisualSPEED**	**Graphical RQL**	**Optique**	**RockQuery**
Result visualization	No	No	No	No	**Yes**
Query history	Yes	No	No	No	**Yes**
Ontology visualization	No	Yes	No	No	**Yes**
Text Filters	No	No	No	Yes	**Yes**
Query visualizer	Yes	Yes	Yes	Yes	**Yes**
Codification	No	Yes	No	Yes	**No**
Knowledge adaptation	No	No	No	No	**Yes**

Source: The authors

9 CONCLUSION & FUTURE DIRECTIONS

In this chapter, we summarize the contributions of this dissertation, and concentrate on the various possible future directions, given the research material presented in the previous chapters.

9.1 Novel Contributions of the Dissertation

In recent years the development of ontologies has been moving from the realm of Artificial-Intelligence laboratories to the desktops of domain experts. Ontologies have become important to guide the development of knowledge-based systems for a decade by now and currently these applications are achieving their maturity. During this period, the ontology evolves, incorporating new knowledge that is necessary for some of the users. However, sometimes this new knowledge is not fully acknowledged by other users of the application. Thus, the application should provide some capability of reorganizing the knowledge, partitioning the set of concepts in smaller portions according to the user previous knowledge and the task at hand. Moreover, the size and complexity of ontology represent a challenge in retrieving information. Finding a portion of interest that can be used as a virtual substitute for a whole ontology for guiding the consultation is highly desired objective, because it reduces the complexity in the user interaction. This dissertation advocates for the use of ontology views to enhance the query formulation in visual query systems. We claim that, by combining ontology views with Human Computer Interaction (HCI) techniques, the applications can provide a better user interaction in the query and analysis of data.

The main principal contribution of our work are the formalization of well-founded ontology view, the conservation principles, and the different algorithms that satisfied the conservation principles. Also, this research has resulted in a visual tool that helps ontology engineer to perform ontology view extraction. We summarize our contributions:

- We proposed the notion of *well-founded ontology view*.
- We proposed a set of *conservation principles* that an ontology view should follow for being considered a *well founded ontology view*. These conservation principles were specified considering the postulates and ontology meta-properties provided by UFO (Unified foundational ontology).
- We have developed a visual tool that helps the ontology engineer to perform ontology view extraction.
- We have developed a new sub-ontology extraction algorithm based in the Unified Foundational Ontology meta-properties.
- We have created an ontology for the domain of *petrography*, covering the concepts of *diagenesis* and *microstructural* community.

- We analyzed a visual query system PetroQuery® in the domain of Petrography, identifying user interaction problems that can be enhanced by using our novel approach. We present different prototypes and implement RockQuery, a system that uses the idea of well-founded view and the algorithm of sub-ontology extraction and combines with data visualizations and HCI techniques.

9.2 Future Research Directions

As a future work, we intend to improve the set of conservation principles focusing in quality universals and developing a new set of sub-ontology extraction algorithms to deal with events. As a consequence, the sub-extraction algorithm will generate well-founded views not only for endurant universals, but also for perdurant universals. Regarding the interaction design, we can use the query log to enhance the visualization by using the node size as the number of term usage in the query log.

Furthermore, systems evolve over time, being extended, combined, and integrated. A core model for knowledge representation needs to support system evolution by being extensible towards new developments and functional requirements that arise. Thus, our approach can be used for enhancing the interactive ontology evolution process (STOJANOVIC, 2004). Ontology evolution involves challenging tasks. For instance, reduce, increase, or update concepts in an ontology could generate inconsistencies with other parts of the ontology. Our approach of sub-ontologies extraction is capable of identifying all the universals that are intrinsically or importantly related with a specific concept, according to ontological meta-properties. From these two premises, we consider that the ontology view approach can identify a critic region of an ontology (critical sub-ontology) that can suffer inconsistencies in the case of modification in a specific concept. In other words, the use of ontology views will help to reduce the search space of concepts that will suffer collateral effects in an event of concept changing.

REFERENCES

ABEL, M. *Study of Expertise in Sedimentary Petrography and their importance for Knowledge Engineering*. 239 p. Thesis (PhD) — Federal University of Rio Grande do Sul, Porto Alegre, 2001.

ABEL, M. et al. Lithologic logs in the tablet through ontology-based facies description. *American Association of Petroleum Geologists*, USA, 2012.

ALENCAR, A. L. de; SALGADO, A. C. A visual query interface for ontology-based peer data management systems. *Brazilian Simposium of Information Systems*, Minas Gerais, 2013.

ALEXAKI, S. et al. Managing rdf metadata for community webs. In: WORKSHOPS ON CONCEPTUAL MODELING APPROACHES FOR E-BUSINESS AND THE WORLD WIDE WEB AND CONCEPTUAL MODELING: CONCEPTUAL MODELING FOR E-BUSINESS AND THE WEB, 2000, Salt Lake City, Utah, USA. *Proceedings...* London, UK, UK: Springer-Verlag, 2000. p. 140–151.

ATHANASIS, N.; CHRISTOPHIDES, V.; KOTZINOS, D. Generating on the fly queries for the semantic web: The ics-forth graphical rql interface (grql). In: MCILRAITH, S.; PLEXOUSAKIS, D.; HARMELEN, F. van (Ed.). *The Semantic Web – ISWC 2004*. [S.l.]: Springer Berlin Heidelberg, 2004. v. 3298, p. 486–501.

BAADER, F. et al. *The Description Logic Handbook: Theory, Implementation, and Applications*. New York, NY, USA: Cambridge University Press, 2003.

BAO, J.; CARAGEA, D.; HONAVAR, V. G. Modular ontologies a formal investigation of semantics and expressivity. In: FIRST ASIAN CONFERENCE ON THE SEMANTIC WEB, 2006, Beijing, China. *Proceedings...* Berlin, Heidelberg: Springer-Verlag, 2006. p. 616–631.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. The goal question metric approach. In: *Encyclopedia of Software Engineering*. [S.l.]: Wiley, 1994.

BASILI, V. R.; SELBY, R. W.; HUTCHENS, D. H. Experimentation in software engineering. *IEEE Trans. Softw. Eng.*, IEEE Press, Piscataway, NJ, USA, v. 12, n. 7, p. 733–743, jul. 1986.

BAZALGETTE, L. et al. Aspects and origins of fractured dip-domain boundaries in folded carbonate rocks. *Journal of Structural Geology*, Amsterdam, Netherlands, v. 32, n. 4, p. 523 – 536, April 2010.

BEDERSON, B. B. Interfaces for staying in the flow. *Ubiquity*, ACM, New York, NY, USA, v. 2004, n. September, p. 1–1, sep. 2004.

BENEVIDES, A. et al. Assessing modal aspects of ontouml conceptual models in alloy. In: HEUSER, C.; PERNUL, G. (Ed.). *Advances in Conceptual Modeling - Challenging Perspectives*. [S.l.]: Springer Berlin Heidelberg, 2009. v. 5833, p. 55–64.

BERCOVICI, N. *Ontology customization and module creation: query-based customization operators and model*. [S.l.], 2008.

BHATT, M. et al. A distributed approach to sub-ontology extraction. In: ADVANCED INFORMATION NETWORKING AND APPLICATIONS, 2004, Fukuoka, Japan. *Proceedings...* [S.l.], 2004. v. 1, p. 636–641.

BHATT, M. et al. Semantic completeness in sub-ontology extraction using distributed methods. In: LAGANA, A. et al. (Ed.). *Computational Science and Its Applications ICCSA 2004*. [S.l.]: Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 3045). p. 508–517.

BLENKINSOP, T. *Deformation Microstructures and Mechanisms in Minerals and Rocks*. [S.l.]: Springer Netherlands, 2000.

BORST, W. N. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. 243 p. Thesis (PhD) — University of Twente, Netherlands, 1997.

BOZSAK, E. et al. Kaon towards a large scale semantic web. In: BAUKNECHT, K.; TJOA, A.; QUIRCHMAYR, G. (Ed.). *E-Commerce and Web Technologies*. [S.l.]: Springer Berlin Heidelberg, 2002. v. 2455, p. 304–313.

BUATIER, M. et al. Origin and behavior of clay minerals in the bogd fault gouge, mongolia. *Journal of Structural Geology*, Amsterdam, Netherlands, v. 34, n. 0, p. 77 – 90, 2012.

BUXTON, B. *Sketching User Experiences: Getting the Design Right and the Right Design: Getting the Design Right and the Right Design*. [S.l.]: Elsevier Science, 2010. (Interactive Technologies).

CARBONERA, J. *Reasoning over Visual Knowledge: An study in Sedimentary Stratigraphy*. Dissertation (Master) — Federal University of Rio Grande do Sul, 2012.

CARD, S.; MACKINLAY, J.; SHNEIDERMAN, B. *Readings in Information Visualization: Using Vision to Think*. [S.l.]: Morgan Kaufmann Publishers, 1999. (Interactive Technologies Series).

CARPENTIER, C. et al. Impact of basin burial and exhumation on jurassic carbonates diagenesis on both sides of a thick clay barrier (paris basin, {NE} france). *Marine and Petroleum Geology*, Amsterdam, Netherlands, v. 53, p. 44 – 70, 2014.

CASTRO, E. et al. Petroquery: a tool for consultation and navigation over ontology. *Regional School of Database*, Porto Alegre, 2005.

CATARCI, T. et al. Visual query systems for databases: A survey. *Journal of Visual Languages & Computing*, Amsterdam, Netherlands, v. 8, n. 2, p. 215 – 260, 1997.

CHIN, J. P.; DIEHL, V. A.; NORMAN, K. L. Development of an instrument measuring user satisfaction of the human-computer interface. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1988, Washington, D.C., USA. *Proceedings...* New York, NY, USA: ACM, 1988. (CHI '88), p. 213–218.

CSIKSZENTMIHALYI, M. *Flow*. [S.l.]: HarperCollins, 2009.

D'AQUIN, M.; SABOU, M.; MOTTA, E. Modularization: a key for the dynamic selection of relevant knowledge components. In: ISWC, 2006, Athens, Georgia, USA. *1st International Workshop on Modular Ontologies, WoMO'06*. [S.l.], 2006.

DAVIS, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.*, Society for Information Management and The Management Information Systems Research Center, Minneapolis, MN, USA, v. 13, n. 3, p. 319–340, sep. 1989.

DEGEN, W. et al. Gol: Toward an axiomatized upper-level ontology. In: INTERNATIONAL CONFERENCE ON FORMAL ONTOLOGY IN INFORMATION SYSTEMS, 2001, New York, NY, USA. *Proceedings...* New York, NY, USA: ACM, 2001. (FOIS '01), p. 34–46.

DEHLER, N. et al. Structural analysis of a core on fractured carbonate reservoir, brazil: Implications for exploration and reservoir modeling. *Journal for E&P Geoscientists*, Amsterdam, Netherlands, 2009.

D'ENTREMONT, T.; STOREY, M. Using a degree of interest model to facilitate ontology navigation. In: VISUAL LANGUAGES AND HUMAN-CENTRIC COMPUTING, 2009, Oregon,USA. *Proceedings...* [S.l.], 2009. p. 127–131.

DERVIN, B. *An Overview of Sense-making Research: Concepts, Methods, and Results to Date*. [S.l.]: The Author, 1983. (Sense-making packet).

DORAN, P. *Ontology Modularization: Principles and Practice*. 150 p. Thesis (PhD) — University of Liverpool, United Kingdom, 2009.

DORAN, P.; TAMMA, V.; IANNONE, L. Ontology module extraction for ontology reuse: An ontology engineering perspective. In: ACM CONFERENCE ON CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 2007, Lisbon, Portugal. *Proceedings...* New York, NY, USA: ACM, 2007. (CIKM '07), p. 61–70.

FALCO, R. et al. Modelling owl ontologies with graffoo. In: ESWC, 2014, Creete, Greece. *Proceedings...* [S.l.], 2014.

FETTER, M.; ROS, L. F. D.; BRUHN, C. H. Petrographic and seismic evidence for the depositional setting of giant turbidite reservoirs and the paleogeographic evolution of campos basin, offshore brazil. *Marine and Petroleum Geology*, Amsterdam, Netherlands, v. 26, n. 6, p. 824 – 853, 2009.

FEW, S. *Information Dashboard Design: The Effective Visual Communication of Data*. [S.l.]: O'Reilly Media, Incorporated, 2006. (O'Reilly Series).

FEW, S. *Data Sensemaking: An interaction of Eyes and Mind*. 2013.

FISHER, Q. J.; KNIPE, R. J. Fault sealing processes in siliciclastic sediments. *Geological Society, London*, London, v. 147, n. 1, p. 117–134, 1998.

FLAHIVE, A. et al. Ontology expansion: appending with extracted sub-ontology. *Logic Journal of IGPL*, Oxford,UK, v. 19, n. 5, p. 618–647, 2011.

FOSSEN, H. *Structural Geology*. [S.l.]: Cambridge University Press, 2010.

GANGEMI, A. et al. Sweetening ontologies with dolce. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING AND KNOWLEDGE MANAGEMENT, 2002, London, UK, UK. *Proceedings ...* London, UK, UK: Springer-Verlag, 2002. (EKAW '02), p. 166–181.

GIER, S. et al. Diagenesis and reservoir quality of miocene sandstones in the vienna basin, austria. *Marine and Petroleum Geology*, Amsterdam, Netherlands, v. 25, n. 8, p. 681 – 695, 2008.

GOMEZ-PEREZ, A.; FERNANDEZ-LOPEZ, M.; CORCHO, O. *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. (Advanced Information and Knowledge Processing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

GRAU, B. C. et al. Automatic partitioning of owl ontologies using e-connections. In: INTERNATIONAL WORKSHOP ON DESCRIPTION LOGICS, 2005, Edinburgh,Scotland, UK. *Proceedings...* [S.l.], 2005.

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

GUARINO, N. Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. In: PAZIENZA, M. (Ed.). *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*. [S.l.]: Springer Berlin Heidelberg, 1997. v. 1299, p. 139–170.

GUARINO, N. Formal ontology and information systems. In: FOIS, 1998, Trento-Italy. *Proceedings ...* [S.l.]: IOS Press, 1998. p. 3–15.

GUARINO, N.; WELTY, C. A. An overview of ontoclean. In: STAAB, S.; STUDER, R. (Ed.). *Handbook on Ontologies*. [S.l.]: Springer Berlin Heidelberg, 2004. p. 201–220.

GUIZZARDI, G. *Ontological Foundations for Structural Conceptual Models*. 441 p. Thesis (PhD) — University of Twente, The Netherlands, 2005.

GUIZZARDI, G.; WAGNER, G. What's in a relationship: An ontological analysis. In: LI, Q. et al. (Ed.). *Conceptual Modeling - ER 2008*. [S.l.]: Springer Berlin Heidelberg, 2008. v. 5231, p. 83–97.

HAERTEL, M.; HERWEGH, M. Microfabric memory of vein quartz for strain localization in detachment faults: A case study on the simplon fault zone. *Journal of Structural Geology*, Amsterdam, Netherlands, n. 0, p. –, 2014.

HEARST, M. A. *Search User Interfaces*. 1st. ed. New York, NY, USA: Cambridge University Press, 2009.

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

HENARES, S. et al. The role of diagenesis and depositional facies on pore system evolution in a triassic outcrop analogue (se spain). *Marine and Petroleum Geology*, Amsterdam, Netherlands, v. 51, n. 0, p. 136 – 151, 2014.

HERRE, H. General formal ontology (gfo): A foundational ontology for conceptual modelling. In: POLI, R.; HEALY, M.; KAMEAS, A. (Ed.). *Theory and Applications of Ontology: Computer Applications*. [S.l.]: Springer Netherlands, 2010. p. 297–345.

HUDLESTON, P. J.; TREAGUS, S. H. Information from folds: A review. *Journal of Structural Geology*, Amsterdam, Netherlands, v. 32, n. 12, p. 2042 – 2071, 2010. Structural Diagenesis.

JUL, S. *From Brains to branch points: Cognitive constraints in navigational Design*. 410 p. Thesis (PhD) — Michigan University, Michigan, 2004.

KIM, J. C.; LEE, Y. I.; HISADA, K. ichiro. Depositional and compositional controls on sandstone diagenesis, the tetori group $middlejurassicearlycretaceous$, central japan. *Sedimentary Geology*, Amsterdam, Netherlands, v. 195, n. 34, p. 183 – 202, 2007.

KLEIN, C. *Manual of Mineral Science*. [S.l.]: John Wiley & Sons Australia, Limited, 2002.

KLEIN, G.; MOON, B.; HOFFMAN, R. Making sense of sensemaking 1: Alternative perspectives. *Intelligent Systems, IEEE*, Washington, DC, USA, v. 21, n. 4, p. 70–73, July 2006.

KOGALOVSKY, M. Ontology-based data access systems. *Programming and Computer Software*, SP MAIK Nauka/Interperiodica, Berlin, v. 38, n. 4, p. 167–182, 2012.

KRIVOV, S. et al. On visualization of owl ontologies. In: BAKER, C.; CHEUNG, K.-H. (Ed.). *Semantic Web*. [S.l.]: Springer US, 2007. p. 205–221.

LAUBACH, S. et al. Fault core and damage zone fracture attributes vary along strike owing to interaction of fracture growth, quartz accumulation, and differing sandstone composition. *Journal of Structural Geology*, Amsterdam, Netherlands, n. 0, p. –, 2014.

LAUBACH, S. et al. Structural diagenesis. *Journal of Structural Geology*, Amsterdam, Netherlands, v. 32, n. 12, p. 1866 – 1872, 2010. Structural Diagenesis.

LOHMANN, S.; NEGRU, S.; BOL, D. The protege vowl plugin: Ontology visualization for everyone. In: ESWC, 2014, Creete, Greece. *Proceedings...* [S.l.], 2014.

LOZANO, J. et al. Ontology view extraction: an approach based on ontological meta-properties. In: ICTAI, 2014, Limassol, Cyprus. *Proceedings...* [S.l.], 2014.

MAGKANARAKI, A. et al. Viewing the semantic web through {RVL} lenses. *Web Semantics: Science, Services and Agents on the World Wide Web*, Amsterdam, Netherlands, v. 1, n. 4, p. 359 – 375, 2004.

MANSURBEG, H. et al. Meteoric-water diagenesis in late cretaceous canyon-fill turbidite reservoirs from the espirito santo basin, eastern brazil. *Marine and Petroleum Geology*, Amsterdam, Netherlands, v. 37, n. 1, p. 7 – 26, 2012.

MARCHIONINI, G. Exploratory search: From finding to understanding. *Commun. ACM*, ACM, New York, NY, USA, v. 49, n. 4, p. 41–46, abr. 2006.

MILLER, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, Washington, DC,USA, 1956.

MODJESKA, D. K. *Hierarchical Data Visualization in Desktop Virtual Reality*. Thesis (PhD) — University of Toronto, Toronto, Ont., Canada, Canada, 2000. AAINQ53695.

MOLLI, G. et al. Fault zone structure and fluidrock interaction of a high angle normal fault in carrara marble (nw tuscany, italy). *Journal of Structural Geology*, Amsterdam, Netherlands, v. 32, n. 9, p. 1334 – 1348, 2010.

MORRIS, A. et al. A filter flow visual querying language and interface for spatial databases. *GeoInformatica*, Kluwer Academic Publishers, Berlin, v. 8, n. 2, p. 107–141, 2004.

NECHES, R. et al. Enabling technology for knowledge sharing. *AI Mag.*, American Association for Artificial Intelligence, Menlo Park, CA, USA, v. 12, n. 3, p. 36–56, sep. 1991.

NICHOLS, G. *Sedimentology and Stratigraphy*. [S.l.]: Wiley, 2009. (Wiley Desktop Editions).

NIELSEN, J. *Usability Engineering*. [S.l.]: Morgan Kaufmann, 1993. (Interactive technologies).

NOY, N.; MUSEN, M. Traversing ontologies to extract views. In: STUCKENSCHMIDT, H.; PARENT, C.; SPACCAPIETRA, S. (Ed.). *Modular Ontologies*. [S.l.]: Springer Berlin Heidelberg, 2009. v. 5445, p. 245–260.

NOY, N. F.; MUSEN, M. A. The prompt suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, Amsterdam, Netherlands, v. 59, n. 6, p. 983 – 1024, 2003.

ORSTROM, P.; ANDERSEN, J.; SCHARFE, H. What has happened to ontology. In: DAU, F.; MUGNIER, M.-L.; STUMME, G. (Ed.). *Conceptual Structures: Common Semantics for Sharing Knowledge*. [S.l.]: Springer Berlin Heidelberg, 2005. v. 3596, p. 425–438.

PARENT, C.; SPACCAPIETRA, S. An overview of modularity. In: *Modular Ontologies*. [S.l.: s.n.], 2009. p. 5–23.

PASSCHIER, C.; TROUW, R. *Microtectonics*. [S.l.]: Springer, 2005.

PIROLLI, P.; CARD, S. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In: INTERNATIONAL CONFERENCE ON INTELLIGENCE ANALYSIS, 2005, McLean, VA, USA. *Proceedings ...* [S.l.], 2005. v. 2005, p. 24.

POLSON, P. G. et al. Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. *Int. J. Man-Mach. Stud.*, Academic Press Ltd., London, UK, UK, v. 36, n. 5, p. 741–773, may 1992.

POSNER, M. *Foundations of Cognitive Science*. [S.l.]: Bradford, 1993. (A Bradford book).

PRESS, F. et al. *Para entender a Terra*. [S.l.]: Bookman, 2006.

PRIOR, D. J.; RUTTER, E. H.; TATHAM, D. J. *Deformation Mechanisms, Rheology and Tectonics*. [S.l.]: Geological Society of London, 2011. 349 p.

ROGERS, Y.; SHARP, H.; PREECE, J. *Interaction Design: Beyond Human - Computer Interaction*. [S.l.]: Wiley, 2011. (Interaction Design: Beyond Human-computer Interaction).

RUSSELL, D. M. et al. The cost structure of sensemaking. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2003, Amsterdam, The Netherlands. *Proceedings...* New York, NY, USA: ACM, 1993. (CHI '93), p. 269–276.

SCHUELLER, S. et al. Spatial distribution of deformation bands in damage zones of extensional faults in porous sandstones: Statistical analysis of field data. *Journal of Structural Geology*, Amsterdam, Netherlands, v. 52, n. 0, p. 148 – 162, July 2013.

SEBRECHTS, M. M. et al. Visualization of search results: A comparative evaluation of text, 2d, and 3d interfaces. In: ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 1999, Berkeley, California, USA. *Proceedings ...* New York, NY, USA: ACM, 1999. (SIGIR '99), p. 3–10.

SEIDENBERG, J.; RECTOR, A. Web ontology segmentation: Analysis, classification and use. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 2006, Edinburgh, Scotland. *Proceedings of the 15th*. New York, NY, USA: ACM, 2006. (WWW '06), p. 13–22.

SHAUGHNESSY, J.; ZECHMEISTER, E.; ZECHMEISTER, J. *Research Methods In Psychology*. [S.l.]: McGraw-Hill, 2006.

SHAW, M. et al. vsparql: A view definition language for the semantic web. *Journal of Biomedical Informatics*, Amsterdam, Netherlands, v. 44, n. 1, p. 102 – 117, February 2011.

SHNEIDERMAN, B. The eyes have it: a task by data type taxonomy for information visualizations. In: VISUAL LANGUAGES, 1996, Boulder, CO. *Proceedings ...* [S.l.], 1996. p. 336–343.

SHNEIDERMAN, B.; PLAISANT, C. *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*. [S.l.]: Pearson Addison Wesley, 2004.

SIBSON, R. Fault rocks and fault mechanisms. *Journal of the Geological Society*, London, v. 133, n. 3, p. 191–213, March 1977.

SILVA, L. A. *Aplication of problem solving methods in tasks of rock interpretation*. 160 p. Dissertation (Master) — Federal University of Rio Grande do Sul, Porto Alegre, 2001.

SMITH, B.; WELTY, C. Ontology:towards a new synthesis. In: INTERNATIONAL CONFERENCE ON FORMAL ONTOLOGY IN INFORMATION SYSTEMS, 2001, Ogunquit, Maine, USA. *Proceedings ...* New York, NY, USA: ACM, 2001. (FOIS '01), p. .3–.9.

SMITH, S. L.; MOSIER, J. N. *Guidelines for designing user interface software*. [S.l.], 1986.

SNOKE, A.; TULLIS, J.; TODD, V. *Fault-related Rocks: A Photographic Atlas*. [S.l.]: Princeton University Press, 1998. (Princeton Legacy Library).

SNYDER, C. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. [S.l.]: Morgan Kaufmann, 2003. (Interactive Technologies Series).

SOYLU, A. et al. Optiquevqs – towards an ontology-based visual query system for big data. In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF EMERGENT DIGITAL ECOSYSTEMS, 2013, Luxemburg. *Proceedings...* Luxemburg: ACM, 2013.

STOJANOVIC, L. *Methods and Tools for Ontology Evolution*. 249 p. Thesis (PhD) — Universitaet Karlsruhe, Germany, 2004.

STUCKENSCHMIDT, H.; SCHLICHT, A. Structure-based partitioning of large ontologies. In: STUCKENSCHMIDT, H.; PARENT, C.; SPACCAPIETRA, S. (Ed.). *Modular Ontologies*. [S.l.]: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5445). p. 187–210.

STUDER, R.; BENJAMINS, V.; FENSEL, D. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, Amsterdam, Netherlands, v. 25, n. 12, p. 161 – 197, 1998.

TEIXEIRA, W. et al. *Decifrando a Terra*. [S.l.]: Companhia Editora Nacional, 2008.

TORABI, A.; FOSSEN, H. Spatial variation of microstructure and petrophysical properties along deformation bands in reservoir sandstones. *AAPG Bulletin*, USA, v. 93, n. 7, p. 919–938, 2009.

TROCHIM, W.; DONNELLY, J. *Research Methods Knowledge Base*. [S.l.]: Atomic Dog/Cengage Learning., 2008.

TSARKOV, D.; PALMISANO, I. Divide et impera: Metareasoning for large ontologies. In: OWL: EXPERIENCES AND DIRECTIONS WORKSHOP, 2012, Heraklion, Crete, Greece. *Proceedings...* [S.l.], 2012.

TUNNING, B. *Visual query*. [S.l.]: Google Patents, 2005. US Patent App. 10/786,453.

VESCOVO, C. D. et al. Empirical study of logic-based modules: Cheap is cheerful. In: INTERNATIONAL WORKSHOP ON DESCRIPTION LOGICS, 2013, Ulm, Germany. *Proceedings...* [S.l.], 2013. p. 144–155.

VOLZ, R.; OBERLE, D.; STUDER, R. Views for light-weight web ontologies. In: ACM SYMPOSIUM ON APPLIED COMPUTING, 2003, Melbourne, Florida, USA. *Proceedings...* New York, NY, USA: ACM, 2003. (SAC '03), p. 1168–1173.

WARE, C. *Information Visualization: Perception for Design*. [S.l.]: Elsevier Science, 2004. (Interactive Technologies).

WORDEN, R. H.; BURLEY, S. D. Sandstone diagenesis: The evolution of sand to stone. In: ____. *Sandstone Diagenesis*. [S.l.]: Blackwell Publishing Ltd., 2009. p. 1–44.

WOUTERS, C. et al. A practical walkthrough of the ontology derivation rules. In: HAMEURLAIN, A.; CICCHETTI, R.; TRAUNMULLER, R. (Ed.). *Database and Expert Systems Applications*. [S.l.]: Springer Berlin Heidelberg, 2002. v. 2453, p. 259–268.

WOUTERS, C. et al. Extraction process specification for materialized ontology views. In: DILLON, T. et al. (Ed.). *Advances in Web Semantics I*. [S.l.]: Springer Berlin Heidelberg, 2009. v. 4891, p. 130–175.

ZLOOF, M. M. Query-by-example: The invocation and definition of tables and forms. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 1975, Framingham, Massachusetts. *Proceedings...* New York, NY, USA: ACM, 1975. (VLDB '75), p. 1–24.

Appendices

AppendixA PRELIMINARY PETROQUERY® STUDY

This appendix contains a conceptual analysis and experimentation performed over PetroQuery® system, discussed in section A.1 and A.2 respectively. Concretely, we crafted this study to help us learn:

- What are the conceptual problems inside PetroQuery System?
- Does PetroQuery System require a better user interaction?

We discussed the results of the experimentation in section A.3.

A.1 Conceptual Analysis

As a first step towards understanding the impact of software evolution, we conducted a conceptual analysis, which consists of analyzing the query history and knowledge model used to implement the database structure. Thus, we make an ontological analysis over the original knowledge base model used to implement the database. This analysis helps us understanding the misconceptualization of some terms used in the knowledge model, which collapses different definitions in some terms.

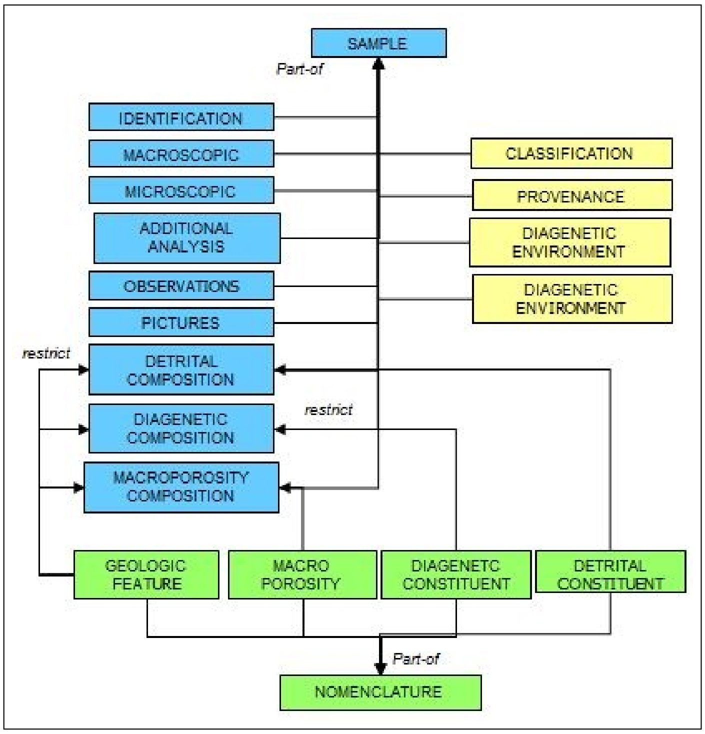
A.1.1 Analysis of Petroledge Knowledge Model

The knowledge model, described in (ABEL, 2001), was designed to describe siliciclastic reservoir rocks. The original model (see Figure A.1) describes the extensional portion of knowledge requested to describe rock samples and the ontological concepts. The terminology was collected from both experts and scientific entities that are responsible for the definition and divulgation in standards in Sedimentary Geology. In the last years, the ontology was expanded to describe other compositional classes of rocks and additional petrographic features, such as structural aspects of the rock and the description of the pore system of reservoirs. This knowledge model was mapped to a database model, described in (SILVA, 2001).

The database model design prioritizes queries over multidimensional data that apply several different attributes for selection of few instances in the database. The dimensions applied for selection are those defined in the domain ontology and are applied dynamically by the system as the ontology grows.

It has been developed different systems over this database model, such as Petroledge and PetroQUery. PetroQuery, described in section 2.6.1 follows the QBE (Query By Example (ZLOOF, 1975) paradigm using a mapping table that plays the role of ontology to guide the consultation. This table mapping includes other terms from the knowledge model, which were not ontologically consistent. Furthermore, the knowledge model is analyzed using the following

Figure A.1: Original Knowledge model of Petroledge®.



The concepts described as subparts of Nomenclature composes the ontology concepts. The other concepts of the models represent the extensional knowledge that supports the petrographic description task.

Source: (ABEL, 2001)

ontological properties, according to the proposal of (GUARINO; WELTY, 2004) and (GUIZZARDI, 2005).

- P1. Identity: the properties supply some own (O) identity criteria, which are not inherited from the subsuming properties, or the criteria of identity are inherited along property subsumption hierarchies. or only hold for some instances and not for the others (I);
- P2. Unity (U): the property defines countable instances;
- P3. Existential dependence (E): the concept X is existential dependent of another Y, if it exists intrinsic individualized properties of X dependent of Y.
- P4. Relational dependence (D): the concept X is relational dependent of another one Y, if every instances of X is related to Y.

Considering the point of view that the system use an ontology to guide the consultation, the terms that appears in the interface, such as *sample description, classification, diagenetic com-*

position, detrital composition and macroporosity should be considered ontological concepts. The terms from the knowledge model with its analysis are listed below:

- *Sample Description* is a piece of information where is registered the petrographic characteristics of what is observed about the rock. Because it is an artifact, it not clear the ontological properties. However, we can view as a relator of petrographer and a rock sample. In addition, this concept was mapped to a table in the database. It contains as attributes: *basin*, *well* and *outcrop*. *Basin* is an area where sediments have been deposited during a stratigraphic event. Ontologically, it has its own identity and unity. *Well* is the identification of the well from where the rock sample was extracted. It has its own identity and unity. *Outcrop* is a visible exposure of bedrock on the surface of the earth. It has its own identity and unity because we can delimit visually the part of bedrock in the surface. All these characteristics are collapsed as attributes of *Sample Description*. This term appear in the interface as ontology concept, which is correct.

- *Macroporosity* is a term that involves *pore* and types of pore. Ontologically it has no sense. It was represented as a table and appears in the interface as an ontology concept, which is incorrect.

- *Detrital composition* is part of sedimentary rock. It has no identity, but it is rigid. It is the set of mineral, fragments of rock or bioclasts that build the sedimentary rock. Thus, it is relational dependent of sedimentary rock. It can be considered as a role mixin. It is represented as a table and appears in the interface as ontology concept, which is correct.

- *Diagenetic composition* is the set of diagenetic constituents, which are minerals that were crystallized by physical and chemical reactions after the sediment deposition. It was mapped to a table. It has no identity, neither unity, but it has rigidity.

- *Classification* is a property of the sample description that defines the compositional or textural petrologic class of the rock. Ontologically has no sense. It was represented as a table and it appears in the interface as carbonate classification and siliciclastic classification, which is incorrect.

The analysis shows that there are terms from the knowledge model that ontologically have no sense. Also, there are terms that appear in the interface that represent measures, which are not ontological concepts. For instance, the term *Total* appears in the interface as a concept, but it represents a resume of the rock composition, describing the proportion of each mineral class in the rock. Several other operations accomplished over the data by system modules are mixed with the static description of concepts and instances in the query system.

A.1.2 Analysis of Query History

The query history contains the list of terms employed by user in each saved query. It was analysed 453 queries in total from 25 users registered in the database. The analysis shows a strong repetition of queries among users, applying few concepts. The concepts are mainly related to basin and author. This means that users do not explore the combination of other concepts because the interface does not offer the exploration of concepts in a suitable manner. This is an interface disadvantage, since it provides full possibility of complex geological analysis over the data, which is hardly reachable by eventual use of spreadsheets or statistic tool, and it is not applied by the users.

A.2 Experimentation

Interaction problems from PetroQuery's users were reported. We design an controlled experiment to understand the issues in the interaction between end-users and the PetroQuery system. In the following we describe methodology and result of such experiment. The experiment follows the methodology described in (BASILI; SELBY; HUTCHENS, 1986) and is presented in the following subsections. For the purposes of this study, we only analyze user's comments, provided either during interviews or as results of thinking aloud protocol during the sessions of demonstration. All users' comments have been analyzed in order to describe the user perception about PetroQuery.

A.2.1 Definition

Our research questions were:

- *How users initiate an exploration task in PetroQuery Interface:* An important step to not frustrate the formulation of query is how the user begins the exploration of concepts in the interface.

- *How users navigate and browse through data in PetroQuery*: Important aspect to be measured in order to identify problems that make complex the interaction between the user and system.

- *Data filtering is supported or not*: Part of a good formulation of query implies that the interface offers filters to choose the correct term.

- *What additional utilities PetroQuery provides that is being used*: We want to identify the frequency of use of other utilities incorporated in PetroQuery that help users to do a better analysis and interpretation of data.

Table A.1: Question and Metric

Question	Metric
Q1.Description of situation when using	Subjective evaluation of use context
Q2.What is the frequency of reusing your own defined queries	% of use of the module Query Saved
Q3.What is the frequency of reusing queries defined	% of use of the module Query Saved
Q4.What is the frequency of using triangular Classification	% of use analytics module
Q5.What is the frequency of using query with retrieval images	% of use photos of the samples

Source: The authors

A.2.2 Design

In the phase of data collection, we elaborated questionnaires and performed interviews. In addition, the user demonstrates how he or she develops a query. The time of consultation is measured and observed the difficulties of interaction. All the interaction is recorded. In the elaboration of the questionnaire, it was considered the approach of GQM where the goal is the Analysis of Query Process in PetroQuery and the defined questions with respective metric are shown in Table A.1.

In the phase of data analysis we identified the way of browsing and navigating through ontology by the audio and mouse track files. Also, we annotated the difficulties found during the interaction.

A.2.3 Implementation and Execution

We conducted the evaluation with six participants from different oil companies that use PetroQuery, two from each company. Although they are all petroleum geologists, the users show different ages and levels of experience and work focus. All of them have previous experience in using PetroQuery. The majority of them have basic notion in computer science. The tasks to be completed by the user were a questionnaire and a test, in which the user needs to propose queries and evaluate the level of difficulty of building the query in the system.

In the interview, it was applied a questionnaire and it was recorded. For the part of demonstration, it was used a mouse tracker software to identify the flux and the time of performing the query. Users reported several difficulties to find information. Also they mentioned some interesting functionalities that are not available in the system, and it were recorded for further studies.

A.3 Results

In this section, we present the results from the different methods used along the experimentation. First of all, we analyze the findings in terms of the answers provided in the interviews. Then we illustrate the problem of interaction in terms of the observation and comments during the demonstration.

A.3.1 Interview Analysis

The interview analysis covers 6 users, which were geologists with different backgrounds. A user information is provided in Table A.2, which contains the media of hours per day of use of Petroledge and PetroQuery, the specialization and the age. The results of the questionnaire is illustrated in Figure A.2.

According to users, the context of use is oriented to generate graphics, to retrieve data from previous years and discover patterns through comparisons. The user U3 illustrates it: *"Two geologists are talking about porosity that is decreasing due to the occurrence of microcrystalline quartz precipitated between grains of the matrix. Then, they formulated the question whether this happens locally in a surface or in the whole basin. Thus, they used PetroQuery to identify if there are other wells that have the same pattern. Using the structure provided by PetroQuery the question will be following the concepts and instance: Basin > Diagenetic constituent > microcrystalline quartz > porosity inter-granular filling porosity..."* Also, user U5 mentions *"PetroQuery is used to retrieve specific descriptions, that have determined characteristics, to make classifications of set of samples and export them..."*. But, we think PetroQuery can be used for other purposes in other geology domains where the object of consultation is not going to be the *Sample Description*. Thus, the consultation interface should be designed to offer an easy navigation in order to build the query.

Questions Q2 and Q3 are related to the percentage of reuse of saved queries. The user will reuse the queries as much as more complex they are and demand great number of concepts. Simple consultation with few concepts for selection will always be developed in the time of consultation. As it was mention, when working in a particular project, more frequently they reuse queries because specific studies involve more complex queries with more than 5 concepts. User U5 identifies two types of users according the type of query, as mentions *"Users that generated petrographic data just make basic queries like petrographers, and users that do complex queries mixing different concepts are the interpreters like reservoir engineer. These interpreters don't know about petrographic concepts..."*. Thus, we should turn more natural the formulation of queries for users who do not have background on petrography.

Finally, in the case of writing queries, some users did not achieve the task of writing queries with more than five concepts, even having a good experience in the use of Petroledge and PetroQuery. We concluded that the basic path of queries is *Basin-Well* or *Sample description*

Table A.2: User Information for contextualizing the use of PetroQuery

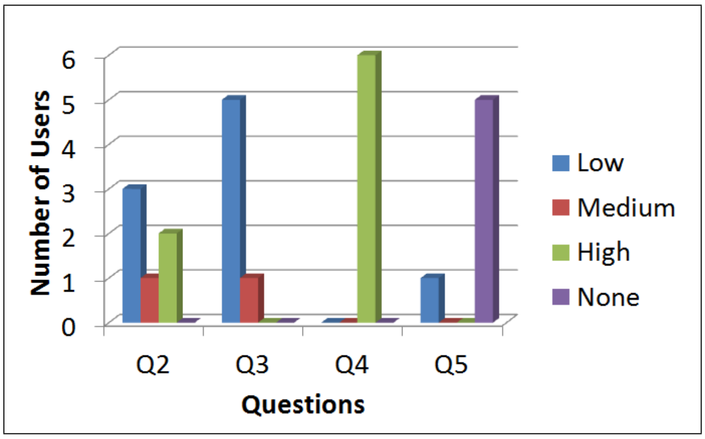
Abbreviations for specializations: Sedimentology SED, Stratigraphy STR, Geology G, Petrography PG. The following abbreviations for reuse of queries: H=high, M= medium, L=low.

Interviewers	**1**	**2**	**3**	**4**	**5**	**6**
Hours per day of Use Petroledge	5	6	2	6	1	1
Hours per day of Use PetroQuery	1/2	2	1/2	2	1/10	2
Specialization	SED	PG	STR	G	PG	SED
Age	31	34	43	22	57	44

Source: The authors

and then any attribute of the *Sample description*. Thus, user has not employ other concepts combinations because the interface not supports a good exploration and analysis. For instance, U5 mentions that *...different tools from Endeeper are not well integrated that let an easy analysis of petrography data....*

Figure A.2: Frequency of use according to the defined Questions



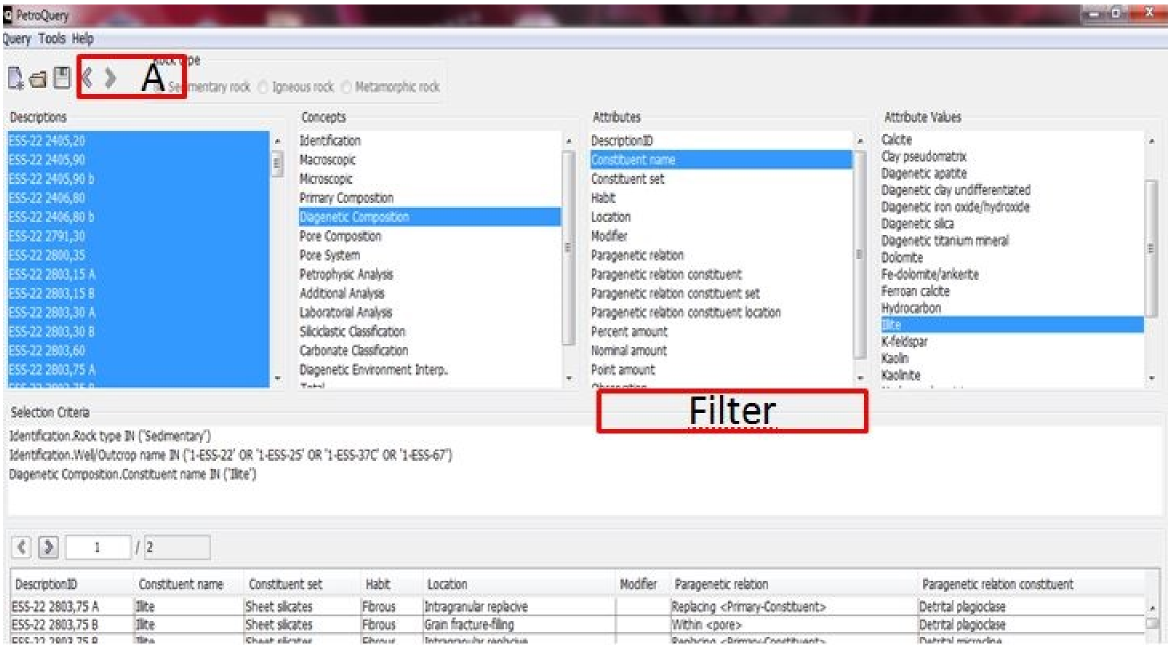
Source: The authors

A.3.2 Observation Analysis

Users is required to start with the concept sample description navigating through its attributes. This is a request for query optimization, since it reduces drastically the search space for the database management system, making possible to support complex multidimensional consultation over databases as large as 10^{th} or 15^{th} instances. Otherwise, in the view of ontological modelling, this initial filter makes no sense since any attribute of the rock are potentially consultable. Moreover it reduces flexibility of the query, requiring the user to understand the Sample concept to continue formulating the query.

In the part of navigation and browsing of concepts, the user has to come back in some cases to redo the query and has to click in the button A from the Figure A.3. This not satisfies the principle of Recognition rather than recall from the Nielsen heuristics. The user U1 takes ten minutes to formulate a query because he could not identify the sequence of concepts scoring this query as difficult. Although, for other users, this query was solved in one minute. According to the cognitive walkthroughs (POLSON et al., 1992) method, we define that the goal was to perform five queries and measure the time of each one. The actions were the selection of appropriate concepts to retrieve the desired information. During this, it was identified the absence of some buttons like *Clear*, *Erase selection*. Also, we observed the absence of a filter (see Figure A.3) when the user has to select an instance that was in the last position. Furthermore, there is no explicit indication of how to group concepts like using the operators *or* or *and*.

Figure A.3: Screenshot of PetroQuery



Identification.Rock type IN ('Sedmentary')
Identification.Well/Outcrop name IN ('1-ESS-22' OR '1-ESS-25' OR '1-ESS-37C' OR '1-ESS-67')
Diagenetic Composition.Constituent name IN ('Illte')

DescriptionID	Constituent name	Constituent set	Habit	Location	Modifier	Paragenetic relation	Paragenetic relation constituent
ESS-22 2803,75 A	Illte	Sheet silicates	Fibrous	Intragranular replacive		Replacing <Primary-Constituent>	Detrital plagioclase
ESS-22 2803,75 B	Illte	Sheet silicates	Fibrous	Grain fracture-filling		Within <pore>	Detrital plagioclase
ESS-22 2803,75 B	Illte	Sheet silicates	Fibrous	Intragranular replacive		Replacing <Primary-Constituent>	Detrital microcline

Source: The authors

For this part, we conclude the following:

- The query is forced to start in *Sample Identification* concept.
- There is No Filtering support.

- There is No good exploration and browsing.

We concluded from the observations that the lack of intention and homogeneity in the design of the PetroQuery consultation system limit the power of using multidimensional consultation associated to a heavy and mature domain ontology. The user does not acknowledge PetroQuery's potential as a consequence of the interface is not well designed to approximate the cognitive understanding of the domain by the user and the exposition of the data by the system.

AppendixB DESIGN ALTERNATIVES

This appendix contains the design attempts resulted from the preliminary PetroQuery® study. In the study were identified the following requirements: a text filter, button for save and create new query, use of a new exploration structure for query formulation, a module for analysis of petrographic data and the enhance of user interaction. All of these must be in one interface.

Those were designed with Lucidchart [1] and Balsamiq [2]. The first design (see Figure B.1) is an arrangement of the PetroQuery® interface increasing two new features, which are profile and recommender queries. The interface contains three panels. The first panel covers the profile and a list view containing the sample descriptions. The second panel covers the formulation and results. The third panel is the recommender section and conditions criteria. This design lacks of a query visualizer. Thus, it was discarded.

The second design has two horizontal panels. The first panel contains two sections, the first section has four list views for selecting descriptions, concepts, attributes, values. In the center of this list views, the user selects if they were going to study sedimentary, igneous or metamorphic rock. The second section has the query visualizer, which consists of a conceptual structure that will be built when adding new concept to the query formulation. The second panel contains the result table. However, this design was discarded because it not offers analysis capability and the process of query interaction is difficult to perform.

The third design has two vertical panels. The first panel contains a text filter and a tree view. In the tree view it is displayed the taxonomy of the queried term. The second panel contains the query visualizer, text query visualizer and a table of results. The query visualizer is a diagram. The lack of this design is that the term could be related with other terms that are not in the taxonomy. This design was discarded because it not offer the analysis capacity. However, the use of a text filter was considered as a widget that should be in the system.

The fourth design has three vertical panels. The first panel contains the text filter, a ontology visualization plugin, and a section for recommender queries. The second panel contains also a text filter, a tag cloud, the query visualizer (a simple table) and a text query visualizer. The third panel contains a result table with a button called *Analytics*, which will pop up a new window with a graphic. It also contains a section for brief description of each sample and the buttons of new and save. On the top, next to the title bar is the user profile section.

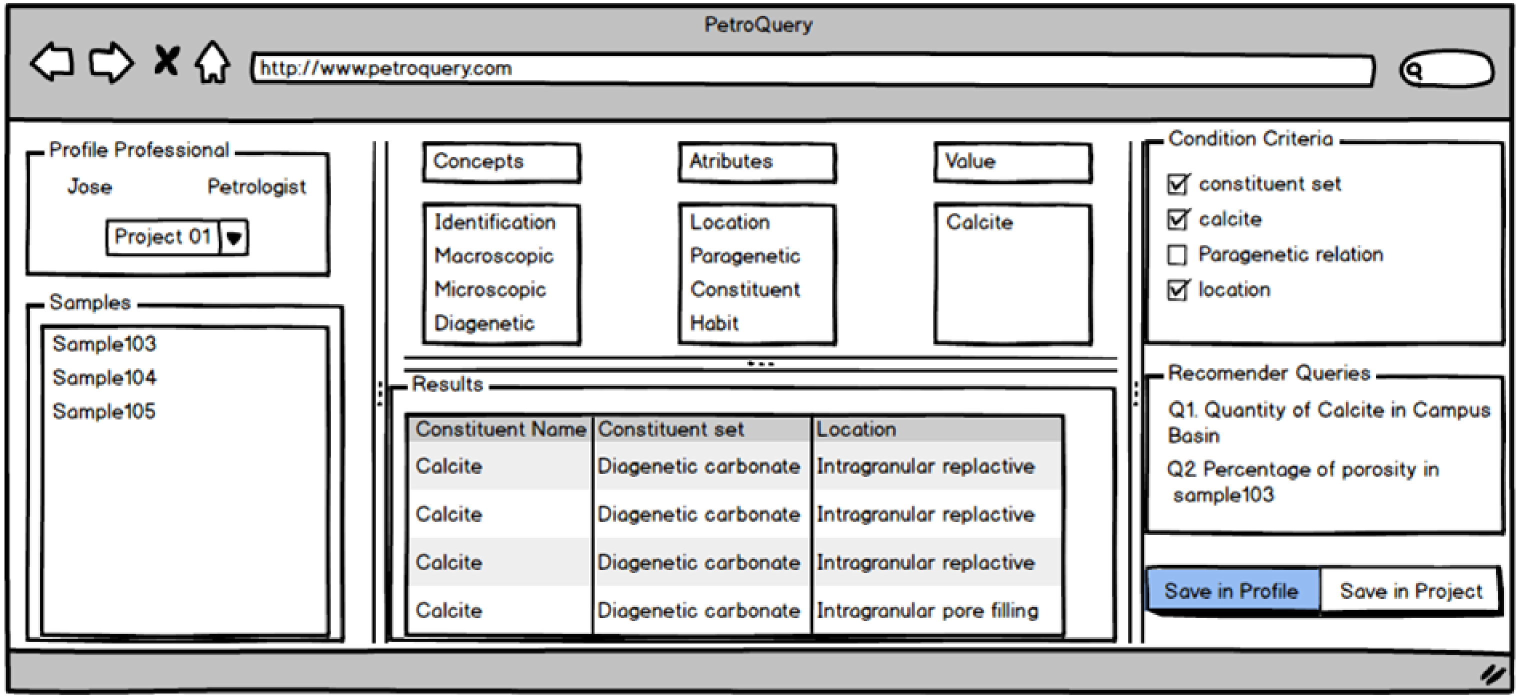
This design was adopted and adapted during the prototype construction. The justification of each component is detailed below. The use of text filter was a requirement from the experimentation. In the use of another structure for query formulation, we employ a graph visualization plugin of an ontology because it has the main structure. However, we should show just part of it and not all the concepts. Thus, the use of our approach will help. The recommender queries

[1] https://www.lucidchart.com/

[2] https://balsamiq.com/

was one of our ideas, because in many recently visual query systems, they have a section for recommender queries. The panel of analysis visualization is because the comment done by U5 in the experimentation about the necessity of an easy analysis of petrographic data.

Figure B.1: Design One



☑ constituent set
☑ calcite
□ Paragenetic relation
☑ location

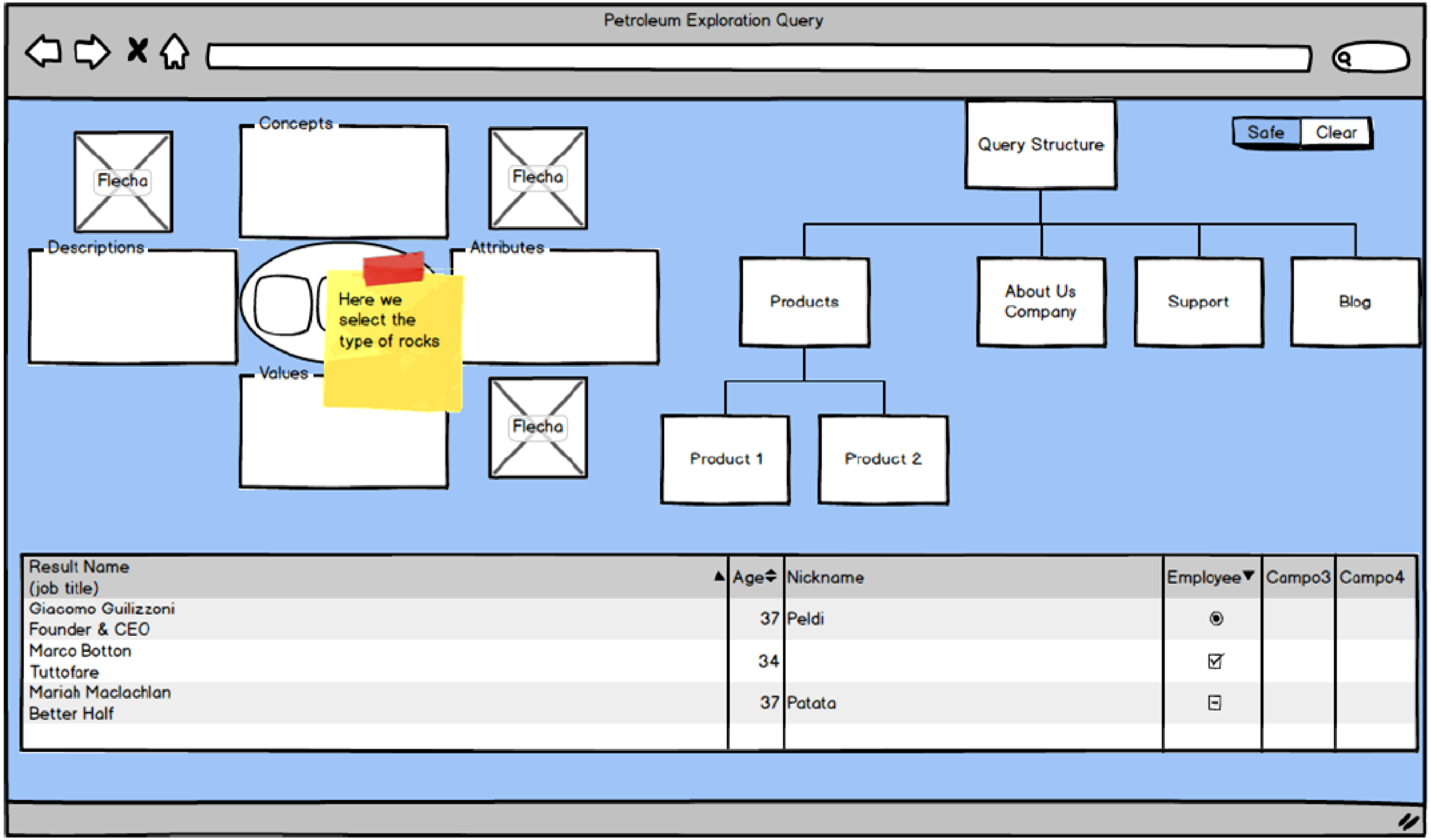
Q1. Quantity of Calcite in Campus Basin

Q2 Percentage of porosity in sample103

Constituent Name	Constituent set	Location
Calcite	Diagenetic carbonate	Intragranular replactive
Calcite	Diagenetic carbonate	Intragranular replactive
Calcite	Diagenetic carbonate	Intragranular replactive
Calcite	Diagenetic carbonate	Intragranular pore filling

Source: The authors

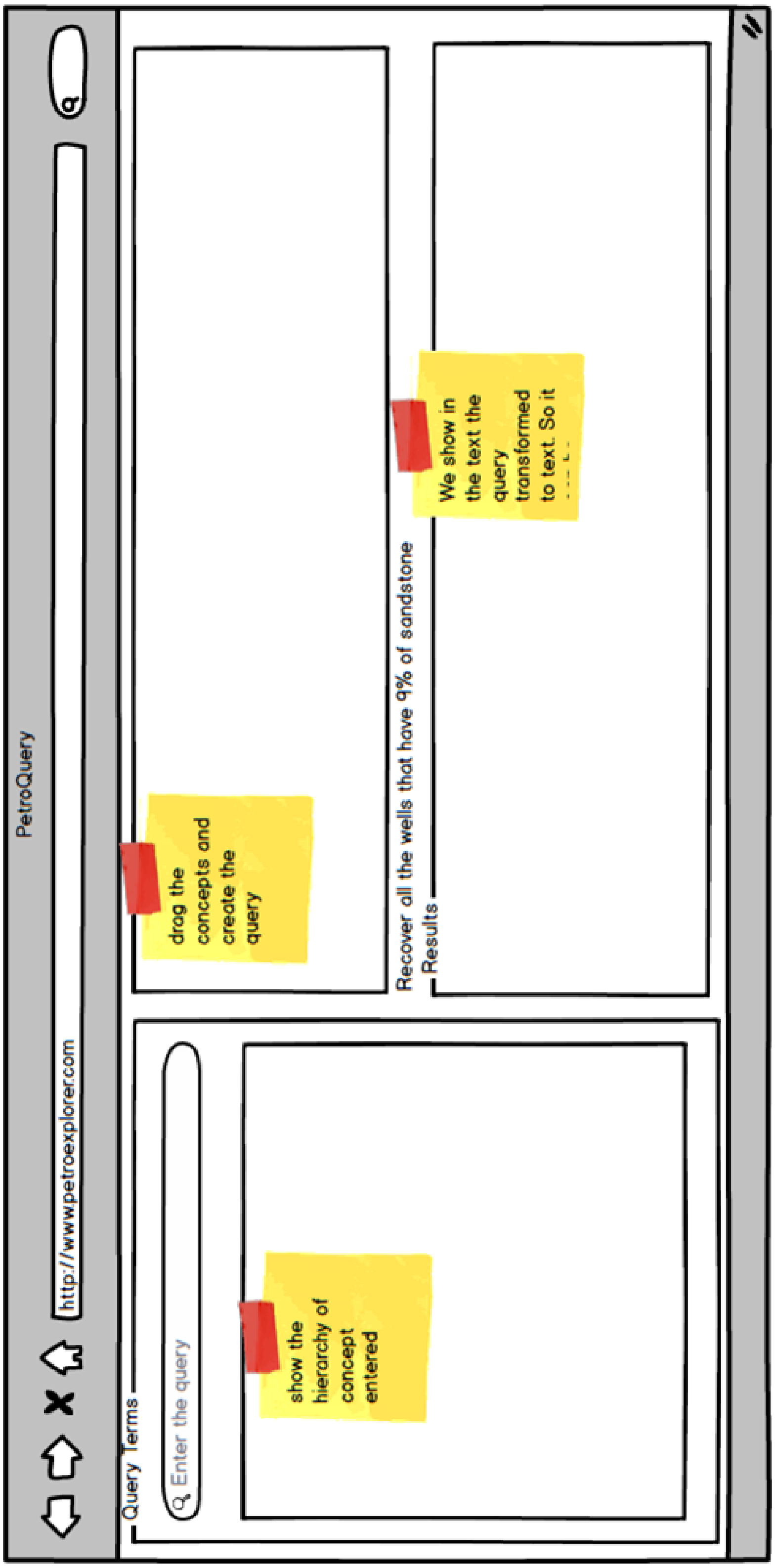
Figure B.2: Design Two



Result Name (job title) ▲	Age⬍	Nickname	Employee▼	Campo3	Campo4
Giacomo Guilizzoni Founder & CEO	37	Peldi	◉		
Marco Botton Tuttofare	34		☑		
Mariah Maclachlan Better Half	37	Patata	⊟		

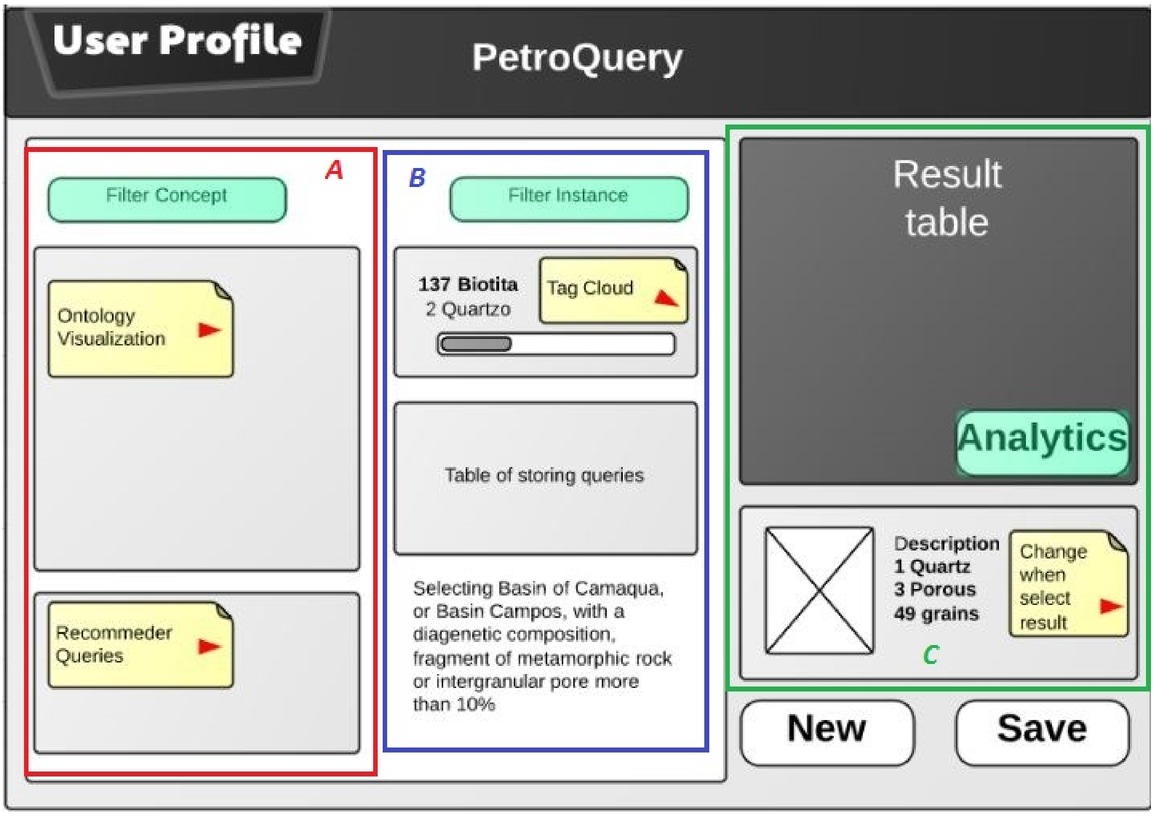
Source: The authors

Figure B.3: Design Three


PetroQuery
http://www.petroexplorer.com
Query Terms
Enter the query
show the hierarchy of concept entered
drag the concepts and create the query
Recover all the wells that have 9% of sandstone
Results
We show in the text the query transformed to text. So it

Source: The authors

Figure B.4: Design Four



Selecting Basin of Camaqua, or Basin Campos, with a diagenetic composition, fragment of metamorphic rock or intergranular pore more than 10%

Source: The authors

AppendixC TABLE RESULTS

Table C.1: Approach 1:Precision and Recall

Type	**Measure**	**A1**		**A2**		**A3**		**A4**		**A5**		**A6**		**AT**	
		D	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**
	P	0.47	0.18	0.62	0.22	0.56	0.17	0.58	0.28	0.51	0.31	0.61	0.28	0.44	0.28
	R	0.15	0.05	0.17	0.06	0.05	0.02	0.18	0.09	0.11	0.07	0.12	0.05	0.31	0.20
	F	0.23	0.08	0.27	0.09	0.10	0.03	0.28	0.13	0.18	0.11	0.20	0.09	0.37	0.23
wP	P	0.47	0.18	0.62	0.22	0.56	0.17	0.58	0.28	0.51	0.31	0.61	0.28	0.44	0.28
	R	0.14	0.05	0.16	0.06	0.05	0.02	0.17	0.09	0.10	0.06	0.11	0.05	0.29	0.19
	F	0.21	0.08	0.25	0.09	0.09	0.03	0.26	0.13	0.17	0.11	0.19	0.09	0.35	0.23
FR	P	0.56	0.70	0.68	0.86	0.61	0.67	0.61	0.81	0.59	0.77	0.69	0.86	0.50	0.71
	R	0.17	0.12	0.18	0.12	0.06	0.03	0.18	0.13	0.12	0.09	0.13	0.09	0.33	0.27
	F	0.26	0.20	0.28	0.22	0.10	0.07	0.28	0.23	0.20	0.16	0.22	0.16	0.40	0.39
RT	P	0.47	0.14	0.62	0.20	0.56	0.17	0.58	0.28	0.51	0.31	0.61	0.28	0.44	0.26
	R	0.15	0.06	0.17	0.08	0.05	0.02	0.18	0.12	0.11	0.09	0.12	0.08	0.31	0.25
	F	0.23	0.08	0.27	0.11	0.10	0.04	0.28	0.17	0.18	0.14	0.20	0.12	0.37	0.25
wP-RT	P	0.47	0.14	0.62	0.20	0.56	0.17	0.58	0.28	0.51	0.31	0.61	0.28	0.44	0.26
	R	0.14	0.06	0.16	0.07	0.05	0.02	0.17	0.12	0.10	0.09	0.11	0.07	0.29	0.24
	F	0.21	0.08	0.25	0.11	0.09	0.04	0.26	0.17	0.17	0.14	0.19	0.12	0.35	0.25
RT-FR	P	0.56	0.68	0.68	0.86	0.61	0.72	0.61	0.82	0.59	0.79	0.69	0.89	0.50	0.70
	R	0.17	0.13	0.18	0.15	0.06	0.04	0.18	0.16	0.12	0.11	0.13	0.11	0.33	0.31
	F	0.26	0.22	0.28	0.25	0.10	0.08	0.28	0.27	0.20	0.19	0.22	0.19	0.40	0.43
wP-FR	P	0.56	0.72	0.68	0.88	0.61	0.72	0.61	0.82	0.59	0.79	0.69	0.89	0.50	0.72
	R	0.16	0.12	0.17	0.13	0.05	0.04	0.17	0.13	0.11	0.09	0.12	0.09	0.31	0.27
	F	0.24	0.20	0.27	0.22	0.10	0.07	0.27	0.23	0.19	0.16	0.21	0.17	0.38	0.39
wP-RT-FR	P	0.56	0.68	0.68	0.86	0.61	0.72	0.61	0.82	0.59	0.79	0.69	0.89	0.50	0.70
	R	0.16	0.13	0.17	0.14	0.05	0.04	0.17	0.16	0.11	0.10	0.12	0.11	0.31	0.30
	F	0.24	0.22	0.27	0.25	0.10	0.08	0.27	0.27	0.19	0.18	0.21	0.19	0.38	0.42

Table C.2: Approach 2:Precision and Recall

Type	**Measure**	**A1**		**A2**		**A3**		**A4**		**A5**		**A6**		**AT**	
		D	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**
	P	0.47	0.02	0.62	0.02	0.56	0.00	0.58	0.04	0.51	0.03	0.61	0.00	0.44	0.02
	R	0.15	0.03	0.17	0.03	0.05	0.00	0.18	0.05	0.11	0.03	0.12	0.00	0.31	0.08
	F	0.23	0.02	0.27	0.02	0.10	0.00	0.28	0.04	0.18	0.03	0.20	0.00	0.37	0.04
wP	**P**	0.47	0.02	0.62	0.02	0.56	0.00	0.58	0.04	0.51	0.03	0.61	0.00	0.44	0.02
	R	0.15	0.02	0.17	0.02	0.05	0.00	0.18	0.05	0.11	0.02	0.12	0.00	0.31	0.07
	F	0.22	0.02	0.26	0.02	0.10	0.00	0.27	0.04	0.18	0.02	0.20	0.00	0.36	0.04
FR	**P**	0.51	0.04	0.66	0.02	0.61	0.00	0.60	0.04	0.54	0.03	0.67	0.00	0.47	0.03
	R	0.16	0.05	0.18	0.02	0.06	0.00	0.18	0.05	0.11	0.02	0.13	0.00	0.32	0.10
	F	0.24	0.04	0.28	0.02	0.11	0.00	0.28	0.04	0.19	0.02	0.22	0.00	0.38	0.05
RT	**P**	0.47	0.02	0.62	0.02	0.56	0.00	0.58	0.04	0.51	0.03	0.61	0.00	0.44	0.02
	R	0.15	0.05	0.17	0.05	0.05	0.00	0.18	0.11	0.11	0.05	0.12	0.00	0.31	0.16
	F	0.23	0.03	0.27	0.03	0.10	0.00	0.28	0.05	0.18	0.03	0.20	0.00	0.37	0.04
wP-RT	**P**	0.47	0.02	0.62	0.02	0.56	0.00	0.58	0.04	0.51	0.03	0.61	0.00	0.44	0.02
	R	0.15	0.05	0.17	0.05	0.05	0.00	0.18	0.09	0.11	0.05	0.12	0.00	0.31	0.14
	F	0.22	0.03	0.26	0.03	0.10	0.00	0.27	0.05	0.18	0.03	0.20	0.00	0.36	0.04
RT-FR	**P**	0.51	0.04	0.66	0.02	0.61	0.00	0.60	0.04	0.54	0.03	0.67	0.00	0.47	0.03
	R	0.16	0.10	0.18	0.05	0.06	0.00	0.18	0.10	0.11	0.05	0.13	0.00	0.32	0.19
	F	0.24	0.05	0.28	0.03	0.11	0.00	0.28	0.05	0.19	0.03	0.22	0.00	0.38	0.05
wP-FR	**P**	0.51	0.04	0.66	0.02	0.61	0.00	0.60	0.04	0.54	0.03	0.67	0.00	0.47	0.03
	R	0.15	0.05	0.18	0.02	0.06	0.00	0.18	0.05	0.11	0.02	0.13	0.00	0.32	0.09
	F	0.24	0.04	0.28	0.02	0.11	0.00	0.28	0.04	0.19	0.02	0.21	0.00	0.38	0.05
wP-RT-FR	**P**	0.51	0.04	0.66	0.02	0.61	0.00	0.60	0.04	0.54	0.03	0.67	0.00	0.47	0.03
	R	0.15	0.08	0.18	0.04	0.06	0.00	0.18	0.08	0.11	0.04	0.13	0.00	0.32	0.17
	F	0.24	0.05	0.28	0.03	0.11	0.00	0.28	0.05	0.19	0.03	0.21	0.00	0.38	0.05

Table C.3: Approach 3:Precision and Recall

Type	**Measure**	**A1**		**A2**		**A3**		**A4**		**A5**		**A6**		**AT**	
		D	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**
	P	0.47	0.02	0.62	0.02	0.56	0.06	0.58	0.07	0.51	0.10	0.61	0.08	0.44	0.05
	R	0.15	0.01	0.17	0.01	0.05	0.01	0.18	0.05	0.11	0.05	0.12	0.04	0.31	0.08
	F	0.23	0.01	0.27	0.01	0.10	0.02	0.28	0.06	0.18	0.07	0.20	0.05	0.37	0.07
wP	**P**	0.47	0.02	0.62	0.02	0.56	0.06	0.58	0.07	0.51	0.10	0.61	0.08	0.44	0.05
	R	0.14	0.01	0.16	0.01	0.05	0.01	0.17	0.05	0.10	0.05	0.11	0.04	0.29	0.08
	F	0.21	0.01	0.25	0.01	0.09	0.02	0.26	0.06	0.17	0.07	0.19	0.05	0.35	0.07
FR	**P**	0.56	0.51	0.68	0.56	0.61	0.50	0.61	0.56	0.59	0.54	0.69	0.58	0.50	0.44
	R	0.17	0.12	0.18	0.12	0.06	0.04	0.18	0.14	0.12	0.09	0.13	0.09	0.33	0.24
	F	0.26	0.20	0.28	0.20	0.10	0.07	0.28	0.22	0.20	0.15	0.22	0.15	0.40	0.31
RT	**P**	0.47	0.02	0.62	0.02	0.56	0.06	0.58	0.07	0.51	0.10	0.61	0.08	0.44	0.05
	R	0.15	0.02	0.17	0.02	0.05	0.02	0.18	0.06	0.11	0.06	0.12	0.05	0.31	0.11
	F	0.23	0.02	0.27	0.02	0.10	0.02	0.28	0.07	0.18	0.08	0.20	0.06	0.37	0.07
wP-RT	**P**	0.47	0.02	0.62	0.02	0.56	0.06	0.58	0.07	0.51	0.10	0.61	0.08	0.44	0.05
	R	0.14	0.02	0.16	0.02	0.05	0.02	0.17	0.06	0.10	0.06	0.11	0.05	0.29	0.11
	F	0.21	0.02	0.25	0.02	0.09	0.02	0.26	0.07	0.17	0.08	0.19	0.06	0.35	0.07
RT-FR	**P**	0.56	0.40	0.68	0.48	0.61	0.33	0.61	0.47	0.59	0.44	0.69	0.50	0.50	0.36
	R	0.17	0.12	0.18	0.12	0.06	0.03	0.18	0.14	0.12	0.09	0.13	0.09	0.33	0.24
	F	0.26	0.18	0.28	0.20	0.10	0.06	0.28	0.22	0.20	0.15	0.22	0.16	0.40	0.28
wP-FR	**P**	0.56	0.51	0.68	0.56	0.61	0.50	0.61	0.56	0.59	0.54	0.69	0.58	0.50	0.44
	R	0.16	0.12	0.17	0.12	0.05	0.04	0.17	0.14	0.11	0.09	0.12	0.09	0.31	0.24
	F	0.24	0.20	0.27	0.20	0.10	0.07	0.27	0.22	0.19	0.15	0.21	0.15	0.38	0.31
wP-RT-FR	**P**	0.56	0.40	0.68	0.48	0.61	0.33	0.61	0.47	0.59	0.44	0.69	0.50	0.50	0.36
	R	0.16	0.12	0.17	0.12	0.05	0.03	0.17	0.14	0.11	0.09	0.12	0.09	0.31	0.24
	F	0.24	0.18	0.27	0.20	0.10	0.06	0.27	0.22	0.19	0.15	0.21	0.16	0.38	0.28

Table C.4: Approach 1:Precision and Recall

Type	**Measure**	**B1**		**B2**		**B3**		**B4**		**B5**		**B6**		**BT**	
		M	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**
	P	0.47	0.31	0.43	0.40	0.45	0.19	0.63	0.11	0.56	0.06	0.62	0.00	0.40	0.24
	R	0.08	0.05	0.08	0.08	0.08	0.03	0.07	0.01	0.05	0.01	0.04	0.00	0.21	0.13
	F	0.14	0.09	0.14	0.13	0.13	0.06	0.12	0.02	0.09	0.01	0.08	0.00	0.28	0.16
wP	**P**	0.47	0.34	0.46	0.40	0.55	0.26	0.63	0.16	0.69	0.06	0.62	0.00	0.43	0.27
	R	0.08	0.06	0.09	0.07	0.09	0.04	0.06	0.02	0.06	0.01	0.04	0.00	0.22	0.13
	F	0.14	0.10	0.14	0.12	0.16	0.07	0.12	0.03	0.11	0.01	0.08	0.00	0.30	0.18
FR	**P**	0.75	0.31	0.80	0.40	0.68	0.23	0.74	0.11	0.56	0.06	0.62	0.00	0.64	0.25
	R	0.07	0.05	0.08	0.07	0.06	0.04	0.04	0.01	0.03	0.01	0.02	0.00	0.18	0.12
	F	0.13	0.09	0.15	0.12	0.11	0.06	0.08	0.02	0.05	0.01	0.04	0.00	0.28	0.17
RT	**P**	0.22	0.31	0.29	0.40	0.26	0.19	0.42	0.11	0.44	0.06	0.46	0.00	0.25	0.24
	R	0.05	0.05	0.08	0.08	0.06	0.03	0.06	0.01	0.05	0.01	0.05	0.00	0.18	0.13
	F	0.09	0.09	0.12	0.13	0.10	0.06	0.11	0.02	0.09	0.01	0.08	0.00	0.21	0.16
wP-RT	**P**	0.22	0.34	0.31	0.40	0.35	0.26	0.42	0.16	0.56	0.06	0.46	0.00	0.28	0.27
	R	0.05	0.06	0.08	0.07	0.08	0.04	0.06	0.02	0.07	0.01	0.04	0.00	0.20	0.13
	F	0.08	0.10	0.13	0.12	0.13	0.07	0.10	0.03	0.12	0.01	0.08	0.00	0.23	0.18
RT-FR	**P**	0.50	0.31	0.66	0.40	0.48	0.23	0.53	0.11	0.44	0.06	0.46	0.00	0.48	0.25
	R	0.05	0.05	0.08	0.07	0.05	0.04	0.03	0.01	0.02	0.01	0.02	0.00	0.16	0.12
	F	0.10	0.09	0.14	0.12	0.09	0.06	0.06	0.02	0.05	0.01	0.04	0.00	0.24	0.17
wP-FR	**P**	0.75	0.34	0.83	0.40	0.77	0.29	0.74	0.16	0.69	0.06	0.62	0.00	0.67	0.28
	R	0.07	0.05	0.08	0.07	0.07	0.04	0.04	0.01	0.03	0.00	0.02	0.00	0.19	0.13
	F	0.13	0.09	0.15	0.12	0.13	0.08	0.08	0.03	0.06	0.01	0.04	0.00	0.29	0.18
wP-RT-FR	**P**	0.50	0.34	0.69	0.40	0.58	0.29	0.53	0.16	0.56	0.06	0.46	0.00	0.52	0.28
	R	0.05	0.05	0.08	0.07	0.06	0.04	0.03	0.01	0.03	0.00	0.02	0.00	0.17	0.13
	F	0.10	0.09	0.14	0.12	0.11	0.08	0.06	0.03	0.06	0.01	0.04	0.00	0.25	0.18

Table C.5: Approach 2:Precision and Recall

Type	**Measure**	**B1**		**B2**		**B3**		**B4**		**B5**		**B6**		**BT**	
		M	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**	**M**	**D**
	P	0.19	0.31	0.20	0.40	0.26	0.19	0.21	0.11	0.44	0.06	0.38	0.00	0.18	0.24
	R	0.15	0.05	0.18	0.08	0.21	0.03	0.10	0.01	0.18	0.01	0.13	0.00	0.44	0.13
	F	0.17	0.09	0.19	0.13	0.23	0.06	0.14	0.02	0.25	0.01	0.19	0.00	0.25	0.16
wP	**P**	0.19	0.34	0.23	0.40	0.35	0.19	0.21	0.11	0.56	0.06	0.38	0.00	0.21	0.25
	R	0.14	0.06	0.19	0.08	0.26	0.03	0.10	0.01	0.21	0.01	0.12	0.00	0.48	0.13
	F	0.16	0.10	0.21	0.13	0.30	0.06	0.13	0.02	0.31	0.01	0.18	0.00	0.29	0.17
FR	**P**	0.19	0.31	0.20	0.40	0.26	0.23	0.21	0.11	0.44	0.06	0.38	0.00	0.18	0.25
	R	0.15	0.05	0.17	0.08	0.20	0.04	0.10	0.01	0.17	0.01	0.12	0.00	0.41	0.13
	F	0.16	0.09	0.18	0.13	0.22	0.06	0.13	0.02	0.25	0.01	0.19	0.00	0.25	0.17
RT	**P**	0.12	0.31	0.14	0.40	0.19	0.19	0.16	0.11	0.38	0.06	0.31	0.00	0.13	0.24
	R	0.21	0.05	0.26	0.08	0.32	0.03	0.16	0.01	0.32	0.01	0.21	0.00	0.68	0.13
	F	0.16	0.09	0.19	0.13	0.24	0.06	0.16	0.02	0.34	0.01	0.25	0.00	0.22	0.16
wP-RT	**P**	0.12	0.34	0.17	0.40	0.29	0.19	0.16	0.11	0.50	0.06	0.31	0.00	0.16	0.25
	R	0.18	0.06	0.27	0.08	0.41	0.03	0.14	0.01	0.36	0.01	0.18	0.00	0.73	0.13
	F	0.15	0.10	0.21	0.13	0.34	0.06	0.15	0.02	0.42	0.01	0.23	0.00	0.27	0.17
RT-FR	**P**	0.12	0.31	0.14	0.40	0.19	0.23	0.16	0.11	0.38	0.06	0.31	0.00	0.13	0.25
	R	0.19	0.05	0.24	0.08	0.29	0.04	0.14	0.01	0.29	0.01	0.19	0.00	0.62	0.13
	F	0.15	0.09	0.18	0.13	0.23	0.06	0.15	0.02	0.32	0.01	0.24	0.00	0.22	0.17
wP-FR	**P**	0.19	0.34	0.23	0.40	0.35	0.23	0.21	0.11	0.56	0.06	0.38	0.00	0.21	0.26
	R	0.14	0.06	0.18	0.07	0.25	0.04	0.09	0.01	0.20	0.01	0.11	0.00	0.45	0.13
	F	0.16	0.10	0.20	0.13	0.29	0.06	0.13	0.02	0.30	0.01	0.18	0.00	0.28	0.18
wP-RT-FR	**P**	0.12	0.34	0.17	0.40	0.29	0.23	0.16	0.11	0.50	0.06	0.31	0.00	0.16	0.26
	R	0.17	0.06	0.25	0.07	0.38	0.04	0.12	0.01	0.33	0.01	0.17	0.00	0.67	0.13
	F	0.14	0.10	0.20	0.13	0.33	0.06	0.14	0.02	0.40	0.01	0.22	0.00	0.26	0.18

Table C.6: Approach 3:Precision and Recall

Type	Measure	B1		B2		B3		B4		B5		B6		BT	
		M	D	M	D	M	D	M	D	M	D	M	D	M	D
	P	0.22	0.31	0.20	0.40	0.29	0.19	0.32	0.11	0.31	0.06	0.46	0.00	0.20	0.24
	R	0.08	0.05	0.08	0.08	0.11	0.03	0.07	0.01	0.06	0.01	0.07	0.00	0.23	0.13
	F	0.12	0.09	0.12	0.13	0.16	0.06	0.12	0.02	0.10	0.01	0.12	0.00	0.21	0.16
wP	**P**	0.22	0.34	0.20	0.40	0.29	0.26	0.32	0.16	0.31	0.06	0.46	0.00	0.20	0.27
	R	0.08	0.06	0.08	0.07	0.11	0.04	0.07	0.02	0.06	0.01	0.07	0.00	0.23	0.13
	F	0.12	0.10	0.12	0.12	0.16	0.07	0.12	0.03	0.10	0.01	0.12	0.00	0.21	0.18
FR	**P**	0.47	0.31	0.54	0.40	0.48	0.23	0.42	0.11	0.31	0.06	0.46	0.00	0.41	0.25
	R	0.06	0.05	0.08	0.07	0.06	0.04	0.03	0.01	0.02	0.01	0.03	0.00	0.17	0.12
	F	0.11	0.09	0.14	0.12	0.11	0.06	0.06	0.02	0.04	0.01	0.05	0.00	0.24	0.17
RT	**P**	0.16	0.31	0.14	0.40	0.19	0.19	0.26	0.11	0.25	0.06	0.31	0.00	0.14	0.24
	R	0.08	0.05	0.08	0.08	0.10	0.03	0.08	0.01	0.06	0.01	0.06	0.00	0.23	0.13
	F	0.11	0.09	0.10	0.13	0.13	0.06	0.12	0.02	0.10	0.01	0.11	0.00	0.18	0.16
wP-RT	**P**	0.16	0.34	0.14	0.40	0.19	0.26	0.26	0.16	0.25	0.06	0.31	0.00	0.14	0.27
	R	0.08	0.06	0.08	0.07	0.10	0.04	0.08	0.02	0.06	0.01	0.06	0.00	0.23	0.13
	F	0.11	0.10	0.10	0.12	0.13	0.07	0.12	0.03	0.10	0.01	0.11	0.00	0.18	0.18
RT-FR	**P**	0.41	0.31	0.46	0.40	0.35	0.23	0.37	0.11	0.25	0.06	0.31	0.00	0.34	0.25
	R	0.07	0.05	0.08	0.07	0.06	0.04	0.04	0.01	0.02	0.01	0.02	0.00	0.17	0.12
	F	0.12	0.09	0.14	0.12	0.10	0.06	0.07	0.02	0.04	0.01	0.04	0.00	0.23	0.17
wP-FR	**P**	0.47	0.34	0.54	0.40	0.48	0.29	0.42	0.16	0.31	0.06	0.46	0.00	0.41	0.28
	R	0.06	0.05	0.08	0.07	0.06	0.04	0.03	0.01	0.02	0.00	0.03	0.00	0.17	0.13
	F	0.11	0.09	0.14	0.12	0.11	0.08	0.06	0.03	0.04	0.01	0.05	0.00	0.24	0.18
wP-RT-FR	**P**	0.41	0.34	0.46	0.40	0.35	0.29	0.37	0.16	0.25	0.06	0.31	0.00	0.34	0.28
	R	0.07	0.05	0.08	0.07	0.06	0.04	0.04	0.01	0.02	0.00	0.02	0.00	0.17	0.13
	F	0.12	0.09	0.14	0.12	0.10	0.08	0.07	0.03	0.04	0.01	0.04	0.00	0.23	0.18