Luis Miguel Sintra Salvo Paiva

Semantic relations extraction from unstructured information for domain ontologies enrichment

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: Professor Doutor Pedro Miguel Negrão Maló
Professor Auxiliar
Faculdade Ciências e Tecnologia, Universidade Nova de Lisboa

Co-orientador: Professor Doutor Celson Pantoja Lima
Professor Visitante, Massachusetts Institute of Technology
Professor Adjunto II, Universidade Federal do Oeste do Pará

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro 2015

Luis Miguel Sintra Salvo Paiva

Semantic relations extraction from unstructured information for domain ontologies enrichment

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: Professor Doutor Pedro Miguel Negrão Maló
Professor Auxiliar
Faculdade Ciências e Tecnologia, Universidade Nova de Lisboa

Co-orientador: Professor Doutor Celson Pantoja Lima
Professor Visitante, Massachusetts Institute of Technology
Professor Adjunto II, Universidade Federal do Oeste do Pará

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro 2015

COPYRIGHT

Semantic relations extraction from unstructured information for domain ontologies enrichment

Acknowledgements

As this was a long path to walk, I hope I can give the deserved merit to all who help me in some way to achieve this great and fulfilling goal.

I would like to acknowledge the following people, as they were important in some point, and provided me the necessary help to walk this path and reach the goals.

I thank my advisor Dr. Celson Lima for proposing me the theme, allowing me to work with him and providing me with the rigour that a work like this needs.

I would also like to thank Dr. Pedro Maló, for being my adviser and especially for all the availability and great help in the final steps.

Additionally, I thank also to MSc. Paulo Figueiras, which provided me a big help to complete the implementation, and was always present and available to help.

I thank to all the professors with whom I had the pleasure and fortune to learn in my course.

Moreover, this path is not only this dissertation. The dissertation is the final step, as such to reach here, I fortunately did not walked alone. I walked with colleagues and friends with whom I had the pleasure to develop myself as well as share experiences and perspectives from each other. I thank to Gonçalo, Abílio, Ricardo, Celso, Manuel, Pedro, Nuno, Arito, Fernando e Márcio. Thanks guys!

Additionally, I would like to thank to my close and dearest friends, outside this academic world, for all the patience related to the parties and invitations I missed out and turned down during this marathon.

Also, my family is my psychological support, in this sense, I would like to thank to “The Clan”, led by my parents, Mário Luiz and Maria Edite, who provided me all (and much more) what was necessary to fulfil all my goals. I love you all.

And lastly, but not least, my dear wife, Carla, and the girls, which gave me the last strength and stability to finally reach the finish line.

Abstract

Based in internet growth, through semantic web, together with communication speed improvement and fast development of storage device sizes, data and information volume rises considerably every day. Because of this, in the last few years there has been a growing interest in structures for formal representation with suitable characteristics, such as the possibility to organize data and information, as well as the reuse of its contents aimed for the generation of new knowledge.

Controlled Vocabulary, specifically Ontologies, present themselves in the lead as one of such structures of representation with high potential. Not only allow for data representation, as well as the reuse of such data for knowledge extraction, coupled with its subsequent storage through not so complex formalisms. However, for the purpose of assuring that ontology knowledge is always up to date, they need maintenance. Ontology Learning is an area which studies the details of update and maintenance of ontologies. It is worth noting that relevant literature already presents first results on automatic maintenance of ontologies, but still in a very early stage. Human-based processes are still the current way to update and maintain an ontology, which turns this into a cumbersome task. The generation of new knowledge aimed for ontology growth can be done based in Data Mining techniques, which is an area that studies techniques for data processing, pattern discovery and knowledge extraction in IT systems.

This work aims at proposing a novel semi-automatic method for knowledge extraction from unstructured data sources, using Data Mining techniques, namely through pattern discovery, focused in improving the precision of concept and its semantic relations present in an ontology. In order to verify the applicability of the proposed method, a proof of concept was developed, presenting its results, which were applied in building and construction sector.

Keywords: Artificial Intelligence, Semantic Web, Ontology Learning, Knowledge Extraction, Association Rules, Pattern Discovery

Sumário

Devido ao crescimento da internet, através da web semântica, aliado à melhoria das velocidades de comunicação e dos suportes de armazenamento, a quantidade de dados e informação aumenta de forma considerável de dia para dia. Desta forma surge a necessidade do aparecimento de estruturas de representação formal com características adequadas, tais como a possibilidade de organização dos dados e informação assim como o seu reaproveitamento para geração de novo conhecimento.

Os vocabulários controlados, e especificamente as ontologias, apresentam-se na linha da frente, como uma destas estruturas de representação com um elevado potencial. Não só permitem a representação dos dados, como a utilização desses mesmos dados para extração de conhecimento assim como o seu posterior armazenamento através de formalismos pouco complexos. No entanto, para que o conhecimento presente nas ontologias possa estar o mais actual possível, estas necessitam de manutenção. *Ontoloy Learning* é uma área que estuda os aspectos da actualizaçao e manutenção de ontologias. Apesar dos esforços para o estudo e desenvolvimento de métodos para a execução totalmente automática deste processo, nos dias de hoje ainda não existe uma forma puramente automatizada para o efeito, o que torna este processo moroso. A geração de novo conhecimento para actualização das ontologias pode ser feito com recurso a técnicas de *Data Mining*, que é uma área que estuda técnicas de tratamento de dados, descoberta de padrões e extracção de conhecimento em sistemas TI.

Este trabalho pretende apresentar uma proposta para a criação de um método de extracção de conhecimento a partir de fontes de dados não estruturados, semi-automatizado, com recurso a técnicas de *Data Mining*, nomeadamente através da descoberta de padrões, com o intuito de melhorar a precisão dos conceitos e das relações semânticas entre conceitos presentes numa ontologia. No sentido de verificação da aplicabilidade do método proposto, foi também desenvolvido um protótipo, com a apresentação dos resultados, aplicado ao sector da construção civil.

Palavras-Chave: Inteligência Artificial, Web Semântica, Ontology Learning, Descoberta de Padrões, Extracção de Conhecimento, Regras de Associação.

Dedico a concretização desta etapa, finalizada por esta dissertação aos meus Pais, Mário Luiz e Maria Edite…

"Always look on the bright side of life!"

Monty Python, in "Life of Brian"

Contents

Figures

Tables

Symbols and Notation

API	**A**pplication **P**rogramming **I**nterface
AR	**A**ssociation **R**ules
ARM	**A**ssociation **R**ule **M**ining
B&C	**B**uilding and **C**onstruction
bcXML	**B**uilding&**C**onstruction XML
BFS	**B**readth **F**irst **S**earch
CV	**C**ontrolled **V**ocabulary
DB	**D**ata**b**ase
DFS	**D**epth **F**irst **S**earch
DM	**D**ata **M**ining
DOKS	**D**ynamic **O**ntology learning with **K**nowledge sources from unstructured data **S**ystem
ECLAT	**E**quivalent **CLA**ss **T**ranformation
FI	**F**requent **I**temset
FIM	**F**requent **I**temset **M**apping
FP	**F**requent **P**atterns
HTTP	**H**yper**T**ext **T**ransfer **P**rotocol
IDE	**I**ntegrated **D**eveloping **E**nvironment
IFC	**I**ndustry **F**oundation **C**lasses
ISO	**I**nternational **S**tandard **O**rganization
IT	**I**nformation **T**echnology
KDD	**K**nowledge **D**iscovery in **D**atabases
NLP	**N**atural **L**anguage **P**rocessing
OET	**O**ntology **E**quivalent **T**erm
OL	**O**ntology **L**earning
OWL	**W**eb **O**ntology **L**anguage
PDF	**P**ortable **D**ocument **F**ile
PHP	**H**ypertext **P**re**P**rocessor
RDBMS	**R**elational **D**ata**b**ase **M**anagement **S**ystem
SEKS	**S**emantic **E**nrichment of **K**nowledge **S**ources
TM	**T**ext **M**ining
UI	**U**ser **I**nterface
USD	**U**ML **S**equence **D**iagram
XML	e**X**tended **M**arkup **L**anguage

1

Introduction

The exponential growth of available information in digital format created the need to discover ways to organize it, in order to be easily accessible. First search engines were essentially word-based, meaning that the results provided by the search process could only be achieved if documents had in their bodies exactly the same words being searched for (Lei et al., 2006). The evolution of search engines motivated by the fact that a simple search by term for the information could not be enough, as the set of terms, or vocabulary available in information being searched could be different from the vocabulary being used. Therefore, it was of great importance to discover approaches for the representation of ideas (concepts), and not just the representation of terms, aiming at getting better results for queries (Almeida and Souza, 2011).

Nowadays, computers systems can represent sets of terms or words (also referred to as vocabularies). However, vocabularies themselves, do not represent ideas or concepts, they just represent words. In order to represent concepts and ideas, one approach can be considered. This approach is the use of mechanisms to represent more than pure words, to represent concepts. These mechanisms are referred to as Controlled Vocabularies (CV) (Lima et al., 2007). CVs are defined subsets of terms from a natural language (e.g. Esperanto), or can be pure symbols of any sort (e.g. sequence of digits)

used to represent concepts, with some sort of organization. CVs represent the concepts by assigning to each, one or more words, or phrases and some describing properties that both translates its meaning. CVs also describe if or how a concept is related to other concept.

Natural languages are very rich in their vocabulary properties. They can have different meanings represented by the same word (Homograph words), in several contexts. Also, there are words that can be pronounced in the same way, however have different spelling and meaning (Homophone words). Homograph and Homophone words can lead to ambiguity and confusion when using the terms by people. CVs address the problems of Homograph and Homophone words solving them by assigning each term to just one concept, and adding properties to explain and provide a better meaning to each concept. For instance, the word "board" can represent a base used in a classroom to write with chalk, or can represent a platform to use in snow sports to ride on top of a mountain hill covered with snow. The way CVs deal with this, is by adding some properties that will increase the precision of the meaning of each term, reducing the ambiguity when these words are used. (N.I.S.O. (US) and others, 2005)

An Ontology is a type of CV that addresses problems like the consistent representation or word ambiguity in information. According to Gruber (Gruber, 1993) an ontology is "*(...) a formal specification of a shared conceptualization of a domain of interest.*" In other words, an ontology represents a formal agreement, where *formal* implies that it has to be machine readable, and *agreement* implies a shared understanding of meaning on the ontological concepts. An Ontology is used when there is the need to share or exchange knowledge within a given domain. Ontologies can be represented as a hierarchically structured set of concepts describing a specific domain of knowledge.

Although ontologies provide structures for concept representation, they face some challenges (Uschold and Gruninger, 1996). So why use an ontology? Inside an organization people from different domains can have different points of view and different words to communicate. In this sense the benefits of using an ontology is to be able to provide a common ground that can lead to a shared understanding for the same concepts. Additionally, when two IT systems need to exchange knowledge, ontologies provides them inter-operability features in order to ease the integration between them (Pouchard et al., 2000). Furthermore, ontologies are useful when there is the need to reuse

its contents and features. There is no need to re-invent the wheel (Gangemi and Presutti, 2009).

Ontology Learning (OL) deals with the creation and maintenance of an ontology, and studies the mechanisms and processes to transform heavy tasks like creation and maintenance of Ontologies, into a semi or complete automatic process. It is worth noting that relevant literature already presents first results on automatic maintenance of ontologies, but still in a very early stage. Human-based processes are still the current way to update and maintain ontology growth (Zhou, 2007).

One of the motors that drive OL itself is the recognition of patterns in the data that could originate new knowledge to further evaluation. For instance, this could be learned from some information not yet known or unpredictable in a specific domain. A pattern, in the area of information retrieval and text mining, can be defined as a predictable occurrence that repeats itself along some text data. Furthermore, Knowledge is defined as "*awareness, familiarity, or understanding of someone or something (e.g. facts, information, descriptions or skills), acquired through experience or education by perceiving, discovering or learning*." (Oxford University Press, 2006) Therefore, OL provides techniques to discover knowledge.

Several processes can be used for a system be able to recognize patterns and further extract knowledge from data and information. Data Mining (also referred in literature as Knowledge Discovery in Databases or KDD) is one of them (Hand et al., 2001). Data mining allows experts to find knowledge in new data or data they already have. Additionally, by adopting data mining techniques, it is expected that decision makers can use new knowledge that otherwise could be unknown, unavailable or difficult to discover, to make better decisions. (Witten et al., 2011)

Having settled the context, urge to say that this dissertation aims at proposing an approach to support part of the process of ontology learning. Specifically, the proposed approach adopts a mechanism suitable for the use of data mining techniques for pattern discovery and extraction, and knowledge discovery from unstructured sources of information from a document corpus. Additionally, it is also proposed an approach to help maintain and update CVs, namely domain ontologies, with the previous discovered knowledge. This means: (i) to discover concepts and relations between them; (ii) to propose an approach to quantify these relations; (iii) to discover new concepts; And finally, (iv) to take advantage of (i), (ii) and (iii) results to update a domain ontology.

Furthermore, a proof of concept to characterize this approach, referred as DOKS (Dynamic Ontology learning with Knowledge sources from unstructured text System), is also part of the results produced.

1.1 Challenges

One of the biggest challenges in information systems when constructing a CV is to find both meaning and relations among concepts and ideas. Furthermore, how to say that a concept is more related to one, than it is to other concept? How to quantify this relation? Similarly, other challenge is to discover knowledge in sources of information that could be later used, for instance, to update a CV. Moreover, is it possible to fully automate this process? Still, other challenge identified relates to the limited amount of information that is inside a single document. This dissertation proposes an approach to help solving these challenges based in the following guiding question:

How to formally discover and quantify semantic relations between concepts in a domain ontology, using external sources of non-structured information?

That question highlights the research path leading the development of this work, as follows:

Semantic relations between concepts from a domain ontology, can be quantified by applying data mining techniques for pattern extraction and knowledge discovery into unstructured sources of information.

1.2 Expected Outcomes

When a study is made, there is a need to consider its contribution and applicability that can arise from it. In this sense, the expected outputs to be provided by this work are the following ones:

- To develop a method to describe how to extract concepts and recognize relations between them from a data document corpus, and to find new knowledge sources in order to update a domain ontology.
- To develop a proof of concept, a software platform, based in the previous method in order to reflect the application of the studied techniques.
- Present results of the semi-automatic OL process. Results composed by patterns discovered in the documents, their relations and the new concepts discovered. They should be presented in an understandable way to the user.
- Finally, publication of scientific documents about the work, to be assessed by the academic community.

1.3 Context of work

The context of the present work arisen from three MSc. Dissertations (Antunes, 2010; Figueiras, 2012; Parada, 2010) in the area of Data Mining and Knowledge Sources. These studies provided the background and inspiration for the reasoned path choice of the present work. The setting made through these studies was provided by CoSPaces. CoSPaces was an European Research project aiming to provide digital solutions in a collaborative workspace between individuals, teams and enterprises. The project expected to achieve the former by improving collaboration methods, like human communication and knowledge sharing support, taking advantage and improving existing IT systems.

EU research project E-Cognos was an inspiration in CV domain. Specifically, it provided the insight and methodology needed to build a domain ontology. Also, provided the ground for the structure representation of the semantics in an ontology applied in the B&C sector.

This work takes advantage of the application domain background based in the Building & Construction sector, which provided the knowledge sources, specifically technical documents (e.g. reports and papers) to be used. They were adopted from (Costa, 2014), a PhD Thesis, that also received a contribution from this study. Namely, "*Semantic enrichment of knowledge sources supported by domain ontologies*", whose main goal was to "*introduce a novel conceptual framework to support the creation of knowledge representations based on enriched Semantic Vectors, using the classical vector space*

model approach extended with ontological support". The respective contribution was the proposal of an ontology learning method based in knowledge discovery techniques.

SEKS (Figueiras, 2012) also provided some resources which were adopted in this work, namely the domain ontology manipulation libraries.

The applicability context of the present work relied in B&C sector, as it was the domain that provided the resources and inspiration. However, in a more abstract sense, the contribution made here can be further used wherever there is the need of a shared communication and understanding of concepts, and in all the fields where knowledge and domain ontologies can be used.

1.4 Document Structure

Following this brief introduction in Chapter 1 with the setting of the problem, the expected outcomes to achieve and the contextualization of the work by the author of the present document, this dissertation will be guided by the following structure.

In Chapter 2, Controlled Vocabularies are the domain of study. Ontology will be the selected CV discussed. It will be explained in more detail what is an Ontology and how to build one. Additionally, it will be presented some existent formalisms to represent them and where are they used.

Chapter 3 will explain what is data mining and knowledge discovery, and describe techniques to discover patterns from unstructured data. One of them, Association Rules will be explained in more detail. FP-Growth, and the concurrent algorithms to discover patterns will be compared, and explained why the former was chosen.

In the following chapter, can be observed the explanation for the solution proposed. Thus, Chapter 4 will present the concept model, an application example describing how to reach from non-structured information to knowledge representation and ontology learning. This chapter also includes the methodology behind FP-Growth and the evaluation of an Association Rule.

With Chapter 5, one can expect to read about the development of a proof of concept. The design and development of a model, with the proposed method to address the question. This will be described with the technologies used, following a description

of the implementation and use cases. The framework developed will also be presented in this chapter.

Chapter 6 will be the assessment of the solution proposal, and the evaluation of the results. Chapter 7 will present some conclusions from the author, namely an overview of this dissertation, the achievement of the proposed outcomes, some possible future directions in this area and some scenarios where this work could be an asset.

2

Controlled Vocabularies

In this chapter it will be presented an introduction to some concepts and definitions about Controlled Vocabularies. Moreover, it will be described forms of knowledge representation. In particular it will be given special attention to Ontologies. Furthermore, it will be explained how to represent a concept (or idea) and the relations between them, into an information system and how ontologies use them. Additionally, a more in depth overview of Ontology Learning will be explained in order to better understand what is it and how does it works, as this is the area that deals with the creation and maintenance of an ontology. Lastly, a brief insight to the project that inspired the idea of Ontology use in the present work, the E-Cognos European project, applied in the Building and Construction sector.

2.1 Controlled Vocabularies – Definition

The exponential growth of available information in digital format created the need to discover ways to organize it, in order to be easily accessible. First search engines were essentially word-based, meaning that the results provided by the search process could only be achieved if documents had in their bodies exactly the same words being searched for (Lei et al., 2006). For instance, if one wanted to do a query on a common search engine (e.g. Yahoo, Google, Bing) for the word "*car*", each result would need to explicitly contain the word searched for. (Figure 2.1)

The evolution of search engines was motivated by the fact that a simple search by term for the information could not be enough, as the set of terms, or vocabulary available in information being searched for could be different from the vocabulary being used.

Referring to the example from Figure 2.1, it is shown that if a user could query a search engine for the concept "*road vehicle, typically with four wheels, powered by an internal-combustion engine and able to carry a small number of people"*(Oxford University Press, 2006) represented by the word "*car*", consequently the results would include the documents containing this search term. Although, the terms "*automobile*" and "*vehicle*" could also describe the same concept. Alternatively, if the term used to search the same concept was "*automobile*", the results would be a different set of documents. Therefore, it was of great importance to discover approaches for the representation of concepts (ideas), and not just the representation of terms, aiming at getting better results for queries (Almeida and Souza, 2011). In this sense, the results provided by the query of Figure 2.1 example would be a sum of the results provided by the terms *"car"*, *"automobile"* and *"vehicle"*.

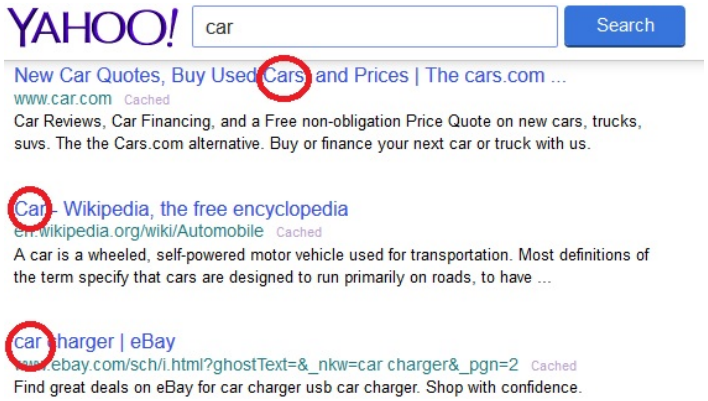


Figure 2.1 - Word search example (Yahoo, 2015)

Nowadays, sets of terms or words (also referred to as vocabularies) can be represented in computers systems. However, vocabularies themselves, do not represent ideas or concepts, they just represent words. Vocabularies are just word lists with no specific organization. Also, words *per se* are just units of a language, they have the responsibility to be the carriers of meaning. One can only understand an idea represented by a word when a meaning is associated to that word, as the meaning is itself the idea that a person wants to express when using that word.

One approach can be considered, in order to represent concepts and ideas. This approach is the use of mechanisms to represent more than pure words, to represent concepts. These mechanisms are referred to as Controlled Vocabularies (CV) (Lima et al., 2007). CVs are defined subsets of terms from a natural language (e.g. Esperanto), or can be pure symbols of any sort (e.g. sequence of digits) used to represent concepts, with

some sort of organization. CVs represent the concepts by assigning to each, one or more words, or phrases and some describing properties that both translates its meaning. CVs can also describe if or how a concept is related to other concept.

Controlled refers to (i) the fact that the vocabulary used needs to be organized based in some logic structure (e.g. Alphabetically, Geographically, Hierarchically, etc.) and defined based in the association of each term to just one meaning, reducing the ambiguity and improving the consistency of a CV; (ii) the fact that the constraints in its use are bigger than in a natural language; a CV can only have one meaning per word. (iii) the fact that the access control to the maintenance of the terms of a CV by the people is restricted. It can have different restrictions to different users (e.g. Normal user, Domain Expert user, Admin user). For example, just domain expert users can propose new words for a CV and just admin users can add new words.

2.1.1 Problems Addressed by a CV

Natural languages are very rich in their vocabulary properties. They can have different meanings represented by the same word, in several contexts, referred to as **Homograph** words. Also, there are words that can be pronounced in the same way, however have different spelling and meaning. These kind of words are referred to as **Homophone** words. Homograph and Homophone words can lead to ambiguity and confusion when using the terms by people or any other entity that uses it (e.g. computer). (Refer to Table a) and b) for examples of Homophone and Homograph words respectively). Moreover, when a word is Homophone and Homograph at the same time, it is called Homonym word, that is to say, **Homonym** words are pronounced and spelled the same way although have different meaning.

CVs address the ambiguity problems of Homograph, Homophone and Homonym words solving them by assigning each term to just one concept, and adding properties to explain and provide a better meaning to each concept. For instance, the homograph word "*board*" (Table -b)) can represent a base used in a classroom to write with chalk, or can represent a platform to use in snow sports to ride on top of a mountain hill covered with snow or can even represent a group of directors from a company. Likewise, the homophone words "*made*" and "*maid*" have the same pronunciation, although the first word refers to the conjugation of the verb "to make" in the simple past tense and past participle, and the second word refers to a female servant. How can a search system (e.g.

Yahoo!, Bing from Microsoft or Google) deal with homograph words by being able to distinguish two different meanings that can be represented by the same word? How can a voice recognition system (e.g. Cortana from Microsoft, Siri from Apple or Google Voice Search) deal with homophone words by being able to recognize accordingly two words that sounds the same? CVs deal with this by adding some properties that will increase the precision of the meaning of each term, reducing the ambiguity when these words are used. (N.I.S.O. (US) and others, 2005).

Table 2.1 - Examples of a) Homophone, b) Homograph and c) Synonym words

Homophone	Homograph		Synonym	
Words	**Word**	**Meaning**	**Word**	**Synonyms**
Board, Bored	Advocate	Speak or write in support of/Person who supports cause of another person	Car	Vehicle, Automobile
Dual, Duel	Board	Base to write/Platform to ride in snowboard	Couch	Sofa, Divan, Chair
Loan, Lone	Desert	A hot, arid region/To leave	Garbage	Trash, Junk, Waste
Made, Maid	Evening	Late afternoon/Making more even	Honest	Honourable, Fair, Sincere, Trustworthy
Sloe, Slow	Match	Wood stick to ignite fire/Sporting event	Intelligent	Smart, Bright, Brilliant, Sharp
Peak, Peek, Peke, Pique, Pick	Object	Thing to see or touch/Goal	Vocabulary	Dictionary, Terminology, Glossary
Rain, Reign, Rein	Tear	To rip/A drop of water from the eye	Woman	Lady, female, girl
a)	**b)**		**c)**	

Additionally, natural languages have more properties that must be addressed by a CV. In particular, there are words that have the same or similar meaning and have different spelling. These words are referred to as **Synonym** words (refer to Table 2.1 c)). As a result, a concept can be represented by more than one word. Referring to Figure 2.1 example, one can infer from it that the concept represented by the word "*car*" have also some other words that can represent them, as any of its synonyms "*vehicle*" or "*automobile*". A CV must allow the use of synonym words, as different people can use different vocabulary for the same concept. And in this case, a query for any of the words from the same concept must return the same or similar results.

2.1.2 Advantages / Disadvantages of CV

When looking at the advantages from the use of CVs, the following can be enumerated:

- **CVs improve the efficiency and precision of retrieval systems.** By providing more than one possible term to search for a concept, they allow for results that do not explicitly contain the search term and still are somehow related to the concept. For instance, a search by cars would provide results containing the word *"cars"* and also the results containing its synonyms words, like *"automobile"* and *"vehicle"*. Likewise, by limiting the terms that can be used, providing a more objective search through the terms used. For example, if one searches documents about *"football"* would not get documents about *"cars"*.
- **CVs remove the ambiguities from natural languages.** Natural languages associate a word to more than one meaning. Consequently, it is hard for an information system to know what the user wants to search. As a result, each term is associated to a specific and unambiguous meaning.
- **CVs activate semantic search**, meaning that the search will be made by idea and not by word. Through the use of each term associated to the concept. In other words, means that the terms used do not need to explicitly be in the data searched for.
- **CVs improve communication through peers in a community or organization,** in the way that they provide the same name to the same thing in the same domain or working context. . When everybody that uses a CV knows the terms to use when referring to a concept, it allows better communication through all people involved in CV use. For instance, when two civil engineers from the same company, work on the same project, a "bar" will always have the same meaning for them, resulting in better communication.
- **CVs provide its reusability in long-term.** The building of a CV can take a lot of time until it can be ready to be used. Requires preparation, planning, and execution time that can be very exhaustive. In this sense, as CVs are built in a way that can be used in several places, several times, taking advantage of the work that was initially aimed for a specific project, into another different project.

In the Disadvantages side, the following can be found:

- **CVs cost time & money to build for the first time.** Building a CV takes time. First to gather all concepts and vocabulary related to a specific domain that will be necessary to include in a CV; second, to find and associate each term to a

specific concept is time-consuming. As a result, companies are reluctant to adopt CVs if they want fast revenue from its investments in short term.

- **CVs allow Human/Domain Expert error.** The concepts are gathered by humans, which should be experts in the CV applicable domain. Although expertise is an asset, the expert is still a human, and humans are prone to errors, even experts. Therefore these errors can lead to imprecise and badly formed CVs.

2.1.3 Types of CVs – Differences, strengths and characteristics

Nowadays there are several ways to represent information in retrieval systems. One of them are CVs. CVs can be divided by complexity, usability needs and level of control.

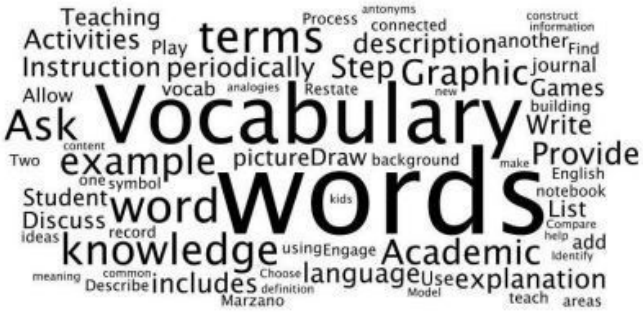


Figure 2.2 - Vocabulary Example

The simplest form of CV is a **Vocabulary**, a list of words or terms without any specific organization logic that gives names to things (Figure 2.2). Although a vocabulary can have some uses, in a retrieval system, most of the times just the words are not enough for semantic retrieval purposes and are the starting point of a CV use.

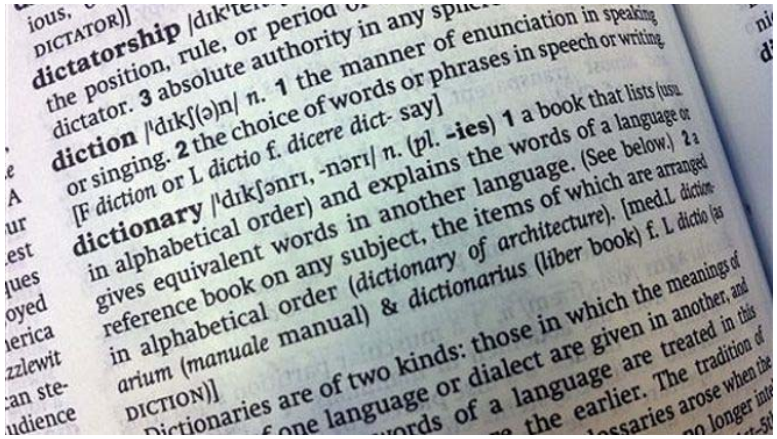


Figure 2.3 - Page from a Dictionary (Oxford University Press, 2006)

When a definition is added to each word from a vocabulary, this vocabulary becomes a **Dictionary** (Figure 2.3). Dictionary is a vocabulary, or a list of words

alphabetically ordered which contains the source of all meaning. Each word has its own meaning described along with some properties. There are several types of dictionaries, in which one of them is a Language Dictionary, which contains all the words that can be used in a particular language (e.g. English Dictionary (Oxford University Press, 2006)). Other type of dictionaries that exist provides the translation of the meaning of every word from one source language to one or more target languages (e.g. Essential Portuguese Dictionary (Oxford University Press, 2012)). This kind of dictionary are used to help when there is the need to communicate between different languages.

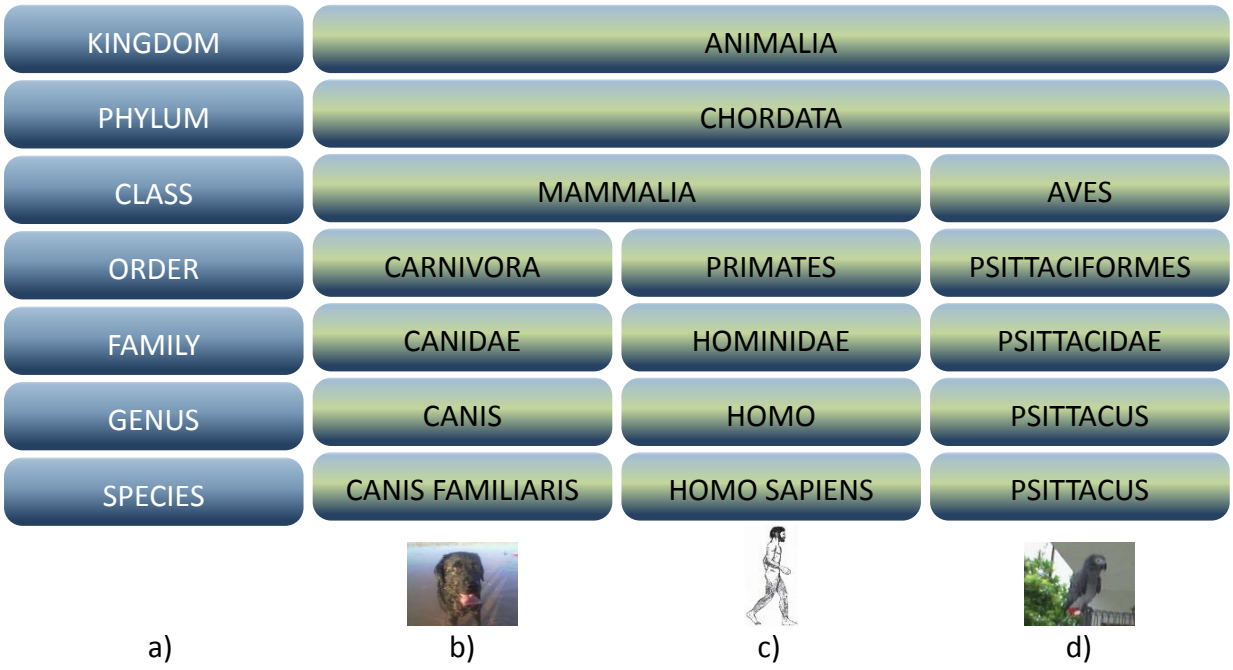
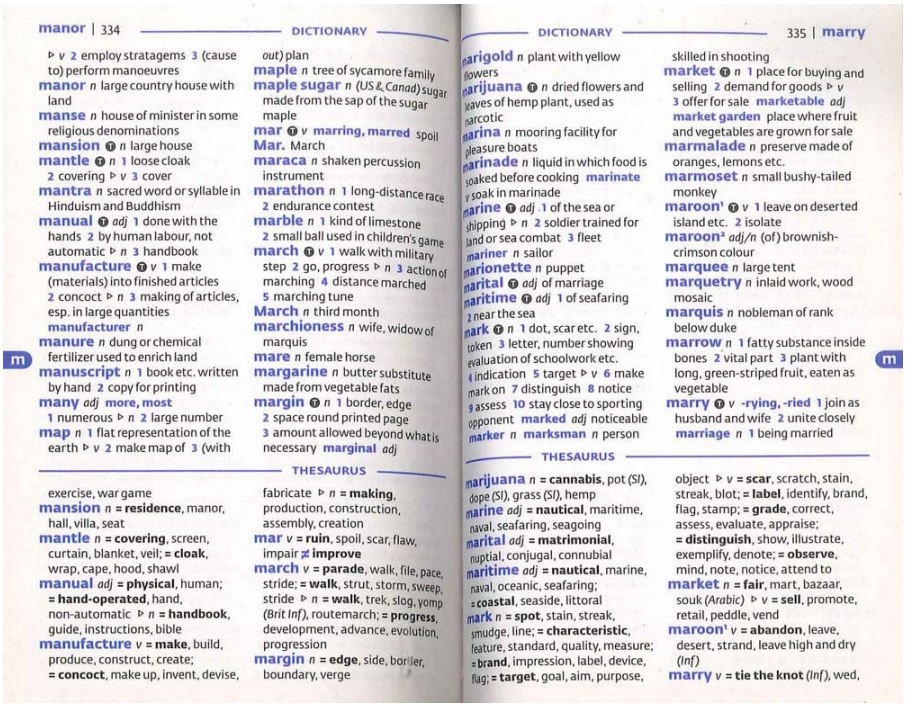


Figure 2.4 - Example of a Species Taxonomy for b) Dog, c) Human and d) Parrot. a) Class Name Hierarchy.

A **Taxonomy** (Figure 2.4) is a structured vocabulary that introduces a hierarchical and a classification layer to a dictionary. Each term is gathered into groups (or classes) in a parent-child-based structure, from the most abstract to the most specific class. It is through a taxonomy that the association between words can be introduced through the parent-child hierarchy. Each term belonging to the same class shares a common characteristic, meaning that each term is associated through this same characteristic. It provides a structured classification mechanism for the terms from a vocabulary.

Adding another type of layer between terms, a sibling-based (on the same hierarchically level) structure, to a taxonomy, results in a **Thesaurus (**Figure 2.5**)**. A Thesaurus takes advantage of a taxonomic structure and associative relations, or semantic relations to its terms. These semantic relations are in the form of synonyms.



manor | 334 DICTIONARY

▷ v 2 employ stratagems 3 (cause to) perform manoeuvres
manor n large country house with land
manse n house of minister in some religious denominations
mansion ❶ n large house
mantle ❶ n 1 loose cloak 2 covering ▷ v 3 cover
mantra n sacred word or syllable in Hinduism and Buddhism
manual ❶ adj 1 done with the hands 2 by human labour, not automatic ▷ n 3 handbook
manufacture ❶ v 1 make (materials) into finished articles 2 concoct ▷ n 3 making of articles, esp. in large quantities
manufacturer n
manure n dung or chemical fertilizer used to enrich land
manuscript n 1 book etc. written by hand 2 copy for printing
many adj **more, most** 1 numerous ▷ n 2 large number
map n 1 flat representation of the earth ▷ v 2 make map of 3 (with out) plan
maple n tree of sycamore family
maple sugar n (US & Canad) sugar made from the sap of the sugar maple
mar ❶ v **marring, marred** spoil
Mar. March
maraca n shaken percussion instrument
marathon n 1 long-distance race 2 endurance contest
marble n 1 kind of limestone 2 small ball used in children's game
march ❶ v 1 walk with military step 2 go, progress ▷ n 3 action of marching 4 distance marched 5 marching tune
March n third month
marchioness n wife, widow of marquis
mare n female horse
margarine n butter substitute made from vegetable fats
margin ❶ n 1 border, edge 2 space round printed page 3 amount allowed beyond what is necessary **marginal** adj

THESAURUS

exercise, war game
mansion n = **residence**, manor, hall, villa, seat
mantle n = **covering**, screen, curtain, blanket, veil; = **cloak**, wrap, cape, hood, shawl
manual adj = **physical**, human; = **hand-operated**, hand, non-automatic ▷ n = **handbook**, guide, instructions, bible
manufacture v = **make**, build, produce, construct, create; = **concoct**, make up, invent, devise, fabricate ▷ n = **making**, production, construction, assembly, creation
mar v = **ruin**, spoil, scar, flaw, impair ≠ **improve**
march v = **parade**, walk, file, pace, stride; = **walk**, strut, storm, sweep, stride ▷ n = **walk**, trek, slog, yomp (*Brit Inf*), routemarch; = **progress**, development, advance, evolution, progression
margin n = **edge**, side, border, boundary, verge

DICTIONARY 335 | marry

marigold n plant with yellow flowers
marijuana ❶ n dried flowers and leaves of hemp plant, used as narcotic
marina n mooring facility for pleasure boats
marinade n liquid in which food is soaked before cooking **marinate** v soak in marinade
marine ❶ adj 1 of the sea or shipping ▷ n 2 soldier trained for land or sea combat 3 fleet
mariner n sailor
marionette n puppet
marital ❶ adj of marriage
maritime ❶ adj 1 of seafaring 2 near the sea
mark ❶ n 1 dot, scar etc. 2 sign, token 3 letter, number showing evaluation of schoolwork etc. 4 indication 5 target ▷ v 6 make mark on 7 distinguish 8 notice 9 assess 10 stay close to sporting opponent **marked** adj noticeable
marker n **marksman** n person skilled in shooting
market ❶ n 1 place for buying and selling 2 demand for goods ▷ v 3 offer for sale **marketable** adj **market garden** place where fruit and vegetables are grown for sale
marmalade n preserve made of oranges, lemons etc.
marmoset n small bushy-tailed monkey
maroon[1] ❶ v 1 leave on deserted island etc. 2 isolate
maroon[2] adj/n (of) brownish-crimson colour
marquee n large tent
marquetry n inlaid work, wood mosaic
marquis n nobleman of rank below duke
marrow n 1 fatty substance inside bones 2 vital part 3 plant with long, green-striped fruit, eaten as vegetable
marry ❶ v **-rying, -ried** 1 join as husband and wife 2 unite closely **marriage** n 1 being married

THESAURUS

marijuana n = **cannabis**, pot (*Sl*), dope (*Sl*), grass (*Sl*), hemp
marine adj = **nautical**, maritime, naval, seafaring, seagoing
marital adj = **matrimonial**, nuptial, conjugal, connubial
maritime adj = **nautical**, marine, naval, oceanic, seafaring; = **coastal**, seaside, littoral
mark n = **spot**, stain, streak, smudge, line; = **characteristic**, feature, standard, quality, measure; = **brand**, impression, label, device, flag; = **target**, goal, aim, purpose, object ▷ v = **scar**, scratch, stain, streak, blot; = **label**, identify, brand, flag, stamp; = **grade**, correct, assess, evaluate, appraise; = **distinguish**, show, illustrate, exemplify, denote; = **observe**, mind, note, notice, attend to
market n = **fair**, mart, bazaar, souk (*Arabic*) ▷ v = **sell**, promote, retail, peddle, vend
maroon[1] v = **abandon**, leave, desert, strand, leave high and dry (*Inf*)
marry v = **tie the knot** (*Inf*), wed,

Figure 2.5 - Page from Oxford Mini School Dictionary & Thesaurus (Allen and Mannion, 2007)

Ontology (Figure 2.6) is the most complex form of a CV. An Ontology includes a set of words (vocabulary), hierarchical and associative relations (taxonomy/thesaurus), a list of concepts and properties for each concept. An ontology is commonly represented in a graph, where each node represents a concept and each connection represents a relation between two concepts. (Please refer to section 2.2 for in depth overview about ontologies)

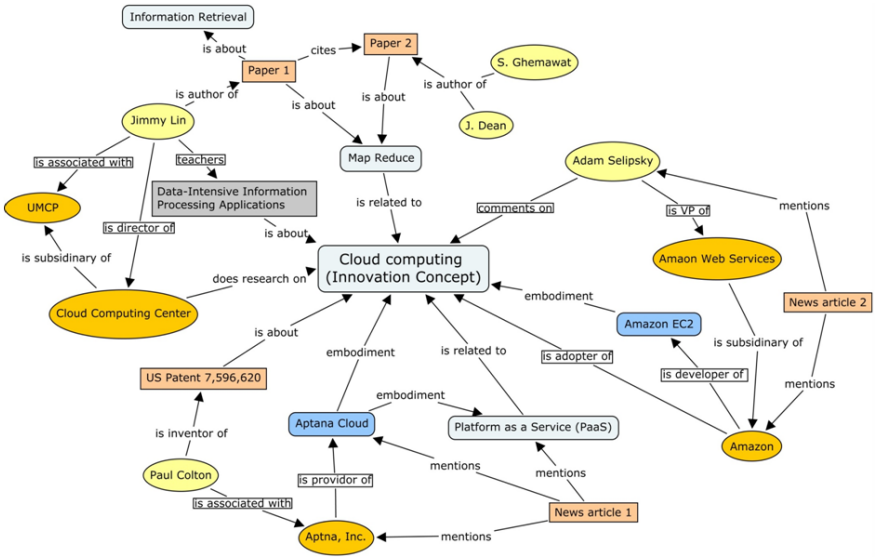


Figure 2.6 - Domain Ontology example (Innovation Ontology adapted from (Stick-iSchool, 2013))

In brief, one can find some arguments that are common to every type of CV presented above:

CVs are mechanisms to structure, classify and represent terms or concepts;

CVs allow a community to agree and use the same terms in the same way;

CVs can be understandable and readable by machines and humans, as well as be used to exchange information between them.

2.1.3.1 Uncontrolled Vocabularies

Another kind of representation structure is worth mentioning. It is not a controlled vocabulary, however is still a managed vocabulary. Is referred to as **Folksonomy**. and is considered an **uncontrolled vocabulary** (Aquino, 2007). Folksonomy provides a user the possibility to associate any word he/she considers adequate to any information element (e.g. documents). This could be understood as the possibility to customize the information from each entity (e.g. person, company, etc) and adapt to the context of such entity. One of differences between a CV and a folksonomy is the control factor. For a folksonomy there is less control over the vocabulary, meaning that the control is not made by experts as in a CV; on the contrary, the control is made by the people that uses it every day. This gives the possibility for final users that access the information to add words that could have been forgotten by domain experts. Although, a folksonomy can be customized by each user, the terms used in the information are not validated by someone who has the expertise knowledge.

Summing up, a folksonomy is a nouvelle representation mechanism that takes advantage of users and social networks to help classify words and build a vocabulary for a specific purpose. This new form of representation is more user-oriented in contrast to the CVs which are more standard-oriented.

2.2 Ontology, a Definition

The term "Ontology" origins from the early 18th century from the modern Latin word *ontologia*, a composition from the Greek words *onto*, which means "being" and *logia* which means "study" (or science, theory). In Philosophy, ontology is the study of the nature of a being or the existence of things and how these things can be related to each other.

In Artificial Intelligence and Information Systems, the most commonly referred definition for ontology is the one presented by Gruber (Gruber, 1993). In particular, an ontology is "*(...) a formal specification of a shared conceptualization of a domain of*

interest." In other words, an ontology represents a formal agreement, where *formal* implies that it has to be machine readable, and *agreement* implies a shared understanding of meaning on the ontological concepts. An Ontology is a type of CV that addresses problems like the consistent representation or word ambiguity in information. An Ontology is used when there is the need to share or exchange knowledge within a given domain. Ontologies can be represented as a hierarchically structured set of concepts describing a specific domain of knowledge.

2.2.1 Ontology Purpose

Nowadays, ontologies can be found across several information-system-related areas. Ontologies can be found in the Semantic Web, in Building and Construction, in Medicine, in Libraries, just to name a few. Although ontologies provide structures for concept representation in all these areas, they face some challenges (Uschold and Gruninger, 1996). Therefore, why use an ontology? Inside an organization people from different domains can have different points of view and different words to communicate. In this sense the benefits of using an ontology relies in the ability to provide a common ground that can lead to a shared understanding for the same concepts. If everyone uses the same words to communicate the same ideas, the understanding of meaning is global across all the peers that access the knowledge in an ontology.

Additionally, the need to communicate remotely and through different types of systems rises each day. Often companies work through different sites or work with information that is not physically located in the same place from where it is accessed (as in a library). Also, for a person can be easy to understand an idea that is being communicated by other people, as they can ask questions to each other to clarify possible doubts. On the contrary, IT systems cannot ask questions. An IT system by its nature, can only understand bits[1]. As a result, when IT systems need to exchange knowledge, they need to be able to understand more than bits. Ontologies provide inter-operability features in order to ease the integration between IT systems (Pouchard et al., 2000). They provide the necessary formalisms to exchange the exact same idea between both. Ontologies provide formal specifications aiming for machine readability, by explicitly defining concepts through terms (eg. words, images, sounds, etc). Therefore, ontologies provide

[1] Bits (also referred as binary digits) are the basic units in a digital system. They commonly can have values of 0 or 1 in which are used to represent data.

to systems exchanging knowledge, the capability to understand the exact same ideas. This understanding can be extended to the point of view of human-machine interaction. Indeed, the formalisms used in ontologies are also human readable. This is a requirement in ontologies and allows a human to understand and work with the knowledge from an ontology.

Furthermore, ontologies are useful when there is the need to reuse its contents and features. There is no need to re-invent the wheel (Gangemi and Presutti, 2009). Would not be worth to build an ontology each time anyone would need one. This step is complex and time consuming, so reuse the ontological resources already available is mandatory to motivate the use of the ontologies.

2.2.2 Ontology Engineering & Components

Specific concepts from a domain are not always easy to understand. Some of them are implicitly understood from other concepts. IT systems do not understand implicit concepts, in contrast they need an explicit conceptualization of ideas in the information in order to be able to understand and work with them easily. Ontology Engineering is a discipline that studies tasks like, Ontology Building (De Nicola et al., 2009; Elsayed et al., 2007) and Ontology Maintenance (Gargouri et al., 2003) which develops approaches for explicit conceptualization of ideas.

One can find ontology tools that can deal with Ontology Engineering tasks (eg. Protegé (Stanford Center for Biomedical Informatics Research, 2011) or OntoEdit (Sure et al., 2002), however these technologies do not have yet sufficient maturation, meaning that the building of an ontology is still a manual, tedious and cumbersome task. Because of this, there is still some reluctance in ontology use. Ontology engineers often face questions and doubts related to ontology development as building time, difficulty, confidence and its maintenance.

In order to help explain why building a domain ontology can be challenging, one first needs to identify the components of an ontology: *Concepts (Ideas), Relations (Meaning), Axioms (Rules)* and *Instances (Individuals).*

Concepts (ideas)

A concept is defined as “An abstract idea; a general notion; an idea formed by mentally combining all its characteristics or particulars” (Oxford University Press, 2006). In other

words, a concept is an idea that can be difficult to understand and is constructed in the mind of someone. It can be anything, as an object, a place, an image, a task, a reasoning process, etc., it can be whatever would fit on a mind.

Meaning and Relations

Meaning is the concept that is represented by a word, phrase. Is the idea that a person desires to express through the use of words, signs, pictures, etc. A relation in an ontology is a connection between two or more concepts, which represents their proximity in meaning. Relations provide more information about concepts related to its meaning. In other words, they help clarify, and position concepts closer to an explicit representation.

Table 2.2 - Examples of relations

RELATIONS	
is-a	*has*
part-of	*is-equal-to*
is-about	*is-similar-to*

The relations can be manifested through either or both hierarchical and associative form. Hierarchical relations are in the form of parent-child connections (or with more levels, like grandparent-grandchild, etc.). These relations can be found in taxonomies, in which case they can be referred as "is-a" taxonomic relations. In previous Figure 2.4 d) from this chapter, it can be seen, as an example, a relation between PRIMATES (parent level) and HOMINIDAE (child level), which can be symbolised in other words as an HOMINIDAE "*is-a*" PRIMATE. Conversely, associative relations are found in connections in the same level, in the form of siblings, called synonyms. This association represents connections to similar or same meaning in a word or concept. One can find some examples for this case in Table c). Several other examples of relations can be found in ontologies. A non-exhaustive example list can be found in Table 2.2 above.

Instances (or Individuals)

Instances (or Individuals) are the units that are used to represent a concept. They can be a word, an image, a number, anything that can be represented and can hold the meaning of a specified concept.

Axioms (or Rules)

Axioms (or Rules) are formal descriptions of the concepts. They describe additional constraints on the ontology and allow to transform implicit facts into explicit ones. (Maedche and Staab, 2001) Axioms provide descriptions for the characteristics and properties of concepts, and can be seen as the concept definitions. They can include collections of descriptions, as restrictions, classes, boolean combinations of descriptions and one or more individuals. (W3C, 2004)

2.2.3 Ontology Languages

There are several formalisms defined that can provide representation of information in an ontology. Table 2.3 provides a non-exhaustive list just for demonstration purposes, of several languages used in Ontology Engineering.

Table 2.3 - Ontology Languages (Lima, 2004)

Language	Description	URL
DAML+OIL	DAML+OIL is a semantic markup language for Web resources. It builds on earlier W3C standards such as RDF and RDF Schema, and extends these languages with richer modelling primitives. DAML+OIL provides modelling primitives commonly found in frame-based languages. It is important to emphasise that this language was the basis of OWL.	http://www.w3.org/TR/daml+oil-reference
EXPRESS / EXPRESS-G	EXPRESS-G is a standard graphical notation for information models. It is a useful companion to the EXPRESS language for displaying entity and type definitions, relationships and cardinality. Used by the ISO DIS 12006-3.	http://www.steptools.com/support/stdev_docs/devtools/devtools-8.html
OIL	OILS stands for Ontology Inference Layer, a language that was developed in the context of the European IST Ontoknowledge project. It is built on top of RDF(S), using as much as possible RDF(S) constructs in order to maintain backward compatibility.	http://www.ontoknowledge.org/oil/
OWL	The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics.	http://www.w3.org/TR/owl-features/
RDF(S)	Resource Description Framework (RDF) defines a language for describing relationships among Web resources in terms of named properties and values. It is particularly intended for representing metadata about Web resources, such as the title, author, copyright and licensing information about a	http://www.w3.org/TR/rdf-schema/

	Web document, or the availability schedule for some shared resource.	
XML	Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. XML has been largely used to represent "semantics" in the Web, here including taxonomies, classification systems, etc..	http://www.w3.org/XML/
Topic Maps	Topic Maps (ISO/IEC **13250)** define a model for the semantic structuring of knowledge networks and are a solution for organising and accessing large and continuously growing information pools. They provide a 'bridge' between the domains of knowledge management and information management. They can also be used to generate navigation for a website, and lots of other metadata tasks. A topic map is a collection of topics (a topic is a resource that acts as a proxy for some subject; the topic map system's representation of that subject), associations, and scopes that may exist in one of two forms: (i) a serialized interchange format (e.g. as a topic map document expressed in XTM syntax); or (ii) Some application-internal form, as constrained by the XTM (XML Topic Maps) Processing Requirements. A topic in a topic Map represents a subject inside the computer.	http://www.topicmap.com/ http://www.topicmaps.org
KIF	Knowledge Interchange Format (KIF) is a language designed for use in the interchange of knowledge among disparate computer systems. KIF, a particular logic language, has been proposed as a standard to use to describe things within computer systems, e.g. expert systems, databases, intelligent agents, etc.. Moreover, it was specifically designed to make it useful as an "interlingua". This means a language useful as a mediator in the translation of other languages. KIF has declarative semantics; it is logically comprehensive (i.e. it provides for the expression of arbitrary sentences in the first-order predicate calculus); it provides for the representation of knowledge about the representation of knowledge; it provides for the representation of non-monotonic reasoning rules; and it provides for the definition of objects, functions, and relations. When the computer system needs to communicate with another computer system, it maps its internal data structures into KIF. KIF is a programmer-readable language and thereby facilitates the independent development of knowledge-manipulation programs.	http://logic.stanford.edu/kif/kif.html

2.3 Ontology Learning

2.3.1 Problems related to the maintenance of an Ontology

The manual creation and maintenance of an ontology is a tedious and cumbersome duty. It is thus desired to reuse and take advantage of the work already done. In this sense,

several tasks can be identified when thinking about the creation and maintenance of an ontology.

Lima (Lima et al., 2003a) identifies two main branches that should be concerns when dealing with the ontology maintenance challenges. The first branch refers to the need for an ontology be adapted to the system and domain in which is being modelled. Specifically, this refers to the ability of an ontology to add, update or delete concepts, relations, instances and axioms, through tasks like acquisition of new knowledge, retrieval and matching of concepts and relations, association of terms to concepts, or definition of constraints and axioms.

The second branch relates to the consistency that needs to be assured between the knowledge representations already existing and the necessary evolutions of the ontology. Noy (Noy and Musen, 2004) identifies some specific tasks related to the consistency assurance needs: Import and reuse ontologies; Translate ontologies from one formalism to another; Provide support for ontology versioning; Specify transformation rules between different ontologies and versions of the same ontology; Merge ontologies; Align and map between ontologies; Extract semantically independent parts of an ontology; Support inference across multiple ontologies; Support query across multiple ontologies.

2.3.2 Definition of Ontology Learning

Ontology Learning (OL) deals with the creation and maintenance of an ontology, and studies the mechanisms and processes to transform heavy tasks like the creation and maintenance of Ontologies, into a semi or complete automatic process. It is worth noting that relevant literature already presents first results on automatic maintenance of ontologies, but still in a very early stage. Human-based processes are still the current way to create, update and maintain ontology growth (Zhou, 2007).

In fact, manual building of an ontology is an extremely intensive and time consuming process, and because of this, the motivation to automate OL is high. OL provides contributions by offering to the ontology community efficiency and overcoming the bottleneck in content discovery for learning ontologies. (Zhou, 2007)

In literature, commonly OL can be found related to several fields such as *machine learning* (Buitelaar et al., 2005), *knowledge acquisition* (Sánchez, 2010), *natural-language processing* (Liu et al., 2011), *information retrieval* (Zhang et al., 2006), *text mining* (Reinberger and Spyns, 2005) and *artificial intelligence*, just to name a few.

Unstructured (non-structured), semi-structured and structured data

The growth of IT systems increased the quantity of data available. This fact created a new challenge, the diversification of the type of information that can be found. These includes, web pages, documents, images and others, holding more or less structure than others.

One of the ways to categorize OL systems is by the data they use to learn ontologies. One can find the data available as structured, semi-structured and unstructured (or non-structured) (Cimiano et al., 2009; Hazman et al., 2011). Consequently, the OL methods to use in each type of data are different.

Structured data is data that is already organized like in databases schemas or in some different type of CVs, like a dictionary or an ontology. As the data is already structured, the main goal in OL from structured data is to find which pieces of data are valuable and can provide interesting knowledge. For instance, one can identify concepts and their relations based in a database schema. (Kashyap, 1999)

Semi-structured data is related to text and data that can be found in HTML pages, XML files, etc. This data already includes some structuring, as a schema and also some free text. It takes advantage of learning methods from structured data, although also needs methods applied on unstructured data to process free text.

Finally, unstructured data relates to text or data in its raw form, without any kind of organization nor processing. This kind of data is related to natural language texts and other kinds of data found in e-books, word, pdf documents, web pages, etc. The methods used to retrieve this kind of data does not rely in any kind of structured information, therefore they are supported by statistical or natural language process approaches. (Hazman et al., 2011)

2.3.3 Ontology Learning State of the Art

The growth of Semantic Web increased the interest to develop methods that could ease the creation and maintenance of semantic resources as ontologies. The automatic learning of ontologies is yet an utopic task, however several researches provided approaches for semi-automatic methodologies for OL.

One example of an approach for OL commonly referred in academic documents is Text2Onto (Cimiano and Völker, 2005). Text2Onto is a tool for ontology learning from unstructured textual sources aimed for the extraction of ontologies from text documents.

In particular, this tool targets the components of an ontology (concepts, taxonomical and non-taxonomical relations, and other properties), to whom are applied different algorithms. For instance, Text2Onto relies in machine learning techniques, to learn concepts.

OntoLearn (Velardi et al., 2005) is an OL system that provides a methodology also for ontology extraction from free text sources. Similar to Text2Onto, it likewise targets several steps in the OL cycle, such as term extraction, natural language definitions extraction, expert parsing of knowledge found and ontology mapping.

OntoEdit[2] (Sure et al., 2002) is a tool aimed for OL from the Semantic Web that proposes a method composed by modules that serve different steps in the ontology engineering cycle. The main steps considered in the methodology of OntoEdit relies in requirements specifications, refinement and evaluation of resources extracted from web documents. In particular, this process includes extracting, pruning, refining, applying, importing and reusing data from web documents.

2.4 Ontologies in Building and Construction Sector – E-Cognos project

The following lines will present a perspective related to the development initiatives of semantic sources by European institutions and companies. Special attention is given to the European Project in Knowledge systems aimed for building and construction sector (B&C), specifically the E-Cognos project.

2.4.1 Historical perspective

Efforts were developed through the last years in Europe, focused in the research and development of controlled vocabularies aimed for B&C sector. Some initiatives include ICONDA®Bibliographic terminology (Fraunhofer, 1986), Industry Foundation Classes (IFC) model from buildingSMART (buildingSMART, 2015), British Glossary for the UK Construction sector (BS6100), bcBuildingDefinitions taxonomy (Lima et al., 2003c) and e-COGNOS ontology (El-Diraby et al., 2005). The initiatives were not limited to Europe. For instance, in North America they included Masterformat™ (Construction

[2] OntoEdit is now OntoStudio, a commercial product from Semafora Systems (Semafora Systems, 2012)

Specifications Institute and Construction Specifications Canada, 2015), OmniClass™ (OCCS Development Committee, 2006) standards, the Canadian Thesaurus and the ANSI/NISO Z39.19 standard for CVs from United States of America (N.I.S.O. (US) and others, 2005).

MasterFormat™ originally created in 1963, designed to satisfy the construction sector needs in North America related to a standard for construction specifications, and constructing and procurement requirements. Specifically, it is a list of numbers and titles aimed for organization. Initially, consisted of 16 divisions with 5 digits to represent each item. After 2004 it was heavily updated to 50 divisions with 8 digits representation. (Construction Specifications Institute and Construction Specifications Canada, 2015)

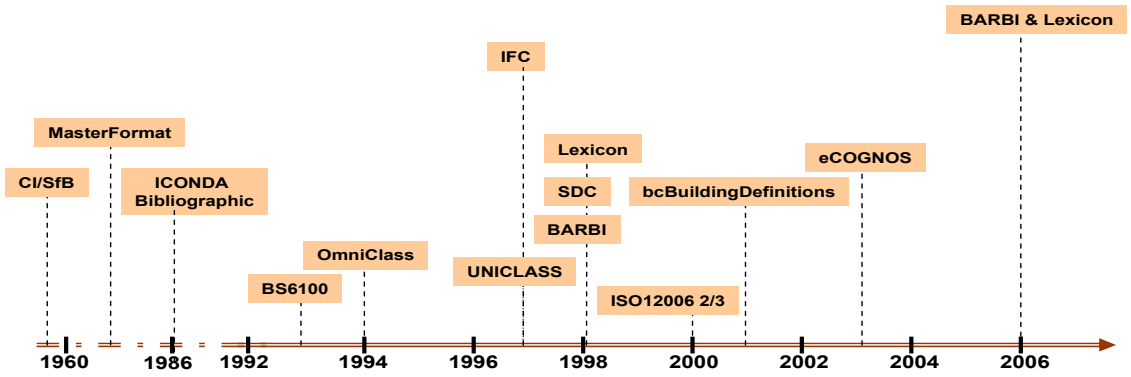


Figure 2.7 - Some examples of CV-focused initiatives in Europe and worldwide (Lima et al., 2007)

ICONDA®Bibliographic terminology created in 1986, is a publications database for the Construction sector. In 2014 included near 900000 records for international publications related to building domains, and grows with a referred rate of near 20000 records per year. (Fraunhofer IRB, 2015)

Omniclass™ is a standard to organize construction information, whose first version was released in 2006. Developed based in standards from ISO and ICIS subcommittees and workgroups from the early 1990s. The classification framework for Omniclass™ is based in the standard ISO12006-2. The basis for the tables origins from MasterFormat for work results, additionally its elements are derived from Uniformat™. Omniclass™ consists of 15 hierarchical tables representing different facets of construction information. (OCCS Development Committee, 2006)

IFC model, which has been developed by buildingSMART (formally known as International Alliance for Interoperability, IAI) since 1997, is an open standard for exchanging Building Information Model (BIM) data, registered under ISO16739:2013 by

ISO. Currently, IFC is now under its fourth version (IFC4), released in 2013.(buildingSMART, 2015)

The bcBuildingDefinitions taxonomy was developed under eConstruct project, the main goal was to present the capabilities of the Building and Construction eXtensible Markup Language (bcXML). This taxonomy contains almost 3000 terms related to *doors* in six different languages. (Lima et al., 2003c)

Finally, the e-COGNOS ontology was a semantic resource developed under the e-COGNOS project, with the goal "*to support the consistent knowledge representation of construction knowledge items considering the e-COGNOS scenario.*" (Lima et al., 2003b) The ontology consists of two taxonomies, one for concepts and other for relations. In the next sub-section this project will be discussed in more detail, with focus on the ontology development, as it was one of the main inspirations for the development of this thesis.

It is worth noting that the previous presented initiatives do not, nor it was intention of the author to reflect the complete universe of the development and research projects related to CVs, on the contrary, they just represent a small sample. Nevertheless, these were the ones that directly or indirectly influenced the present study.

2.4.2 Creation of an ontology in B&C – E-Cognos approach

Developed under a European consortium[3] in 2001, e-COGNOS (COnsistent knowledGe maNagement across prOjects and between enterpriSes in the construction domain – IST-2000-28671) was a project aimed for the management of Knowledge Resources tailored for the B&C industry sector. This project was created with one vision, particularly by assuming that different information and knowledge sources can be shared and used between several actors, in a co-operative approach.

The e-CKMI was the Knowledge Management Infrastructure developed for the IT-based perspective (two development perspectives were identified, IT and managerial) of e-COGNOS, which motivated the development of several components. One of its components and main pillars was the e-COGNOS ontology. E-COGNOS ontology was developed driven by the following: *a group of **Actors** uses a set of **Resources** to produce*

[3] The Consortium included R&D organizations, namely University of Salford (UK) and Centre Scientific et Technique du Bâtiment - CSTB (France), as well as end users, particularly, HOCHTIEF (Germany), OTH (France), YIT (Finland) and Taylor Woodrow (UK).

a set of ***Products*** *following certain* ***Processes*** *within a work environment (****Related Domains****) and according to certain conditions (****Technical Topics****).* (Lima et al., 2005)

The methodology proposed for the creation of the e-COGNOS ontology was the following (Figure 2.8):

- Definition of domain and scope;
- Reuse of ontology-related resources;
- Enumerate the important terms to the taxonomy;
- Define concepts and concept hierarchy based on the relation "*is-a*";
- Define properties of the concepts;
- Define restrictions;
- Populate the ontology.

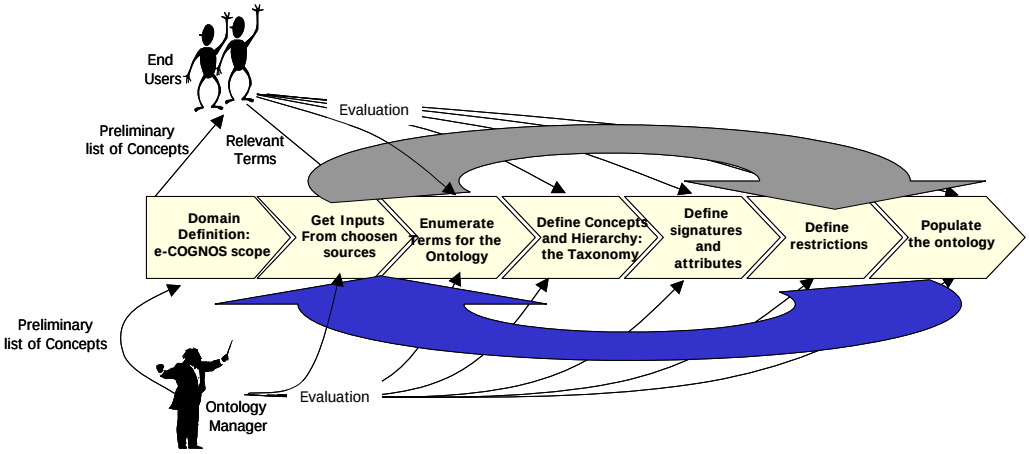


Figure 2.8 - The e-COGNOS ontology creation methodology (Lima et al., 2002)

The components considered in the e-COGNOS ontology were the following: (i) a glossary; (ii) a vocabulary; (iii) a classification system; (iv) a concept taxonomy; and (v) a definition taxonomy. (i) The glossary provides the words from the respective domain. It was adopted from BS6100, because its terms were widely accepted and they include a myriad of synonyms which is very rich. (ii) The vocabulary was an XML vocabulary adopted from bcXML which provided the base to build the bcBuildingDefinitions taxonomy. (iii) The classification needs was provided by ISO 12006-2 classification system. Although this standard did not provide a complete classification system, it provided an identification of classes and their relations, which are necessary for information organization purposes. (iv), (v) both taxonomies, concepts and definitions,

were built based on two sources: O'CoMMA[4] ontology and IFC Model. O'CoMMA provided an initial sample of concepts and the IFC Model also provided a list of concepts, to build the ontology. Later, IFC Model provided, as well, more concepts, attributes and relations to improve this ontology. In Figure 2.9 can be seen both taxonomies of e-COGNOS ontology, concept taxonomy and relations taxonomy.

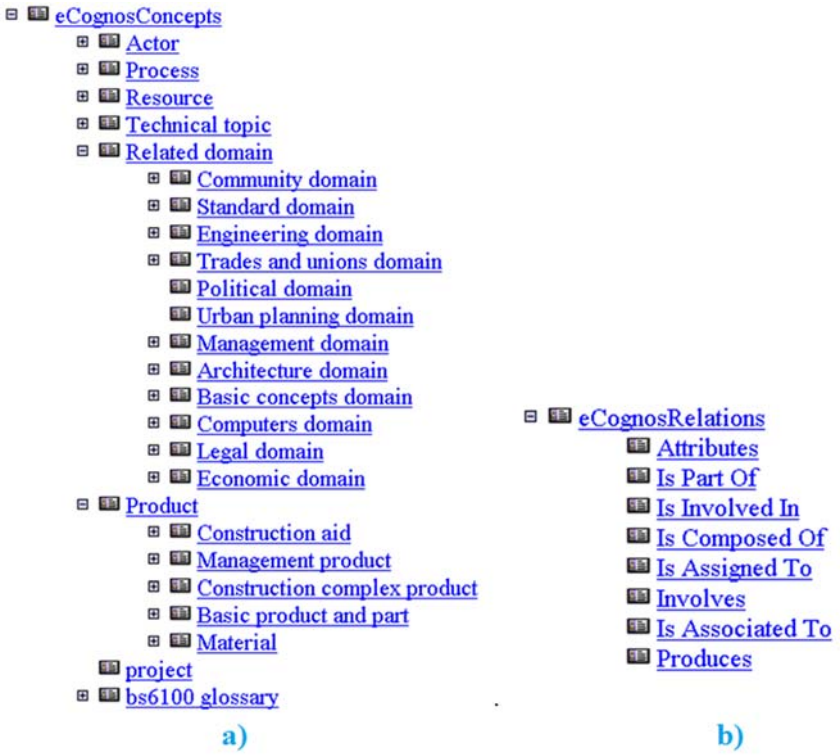


Figure 2.9 - e-Cognos taxonomies a) Concepts; b) Relations (Costa, 2014)

The e-COGNOS ontology first version was created with 800 concepts. Later, achieved more than 17000 concepts with the help of bcXML language to import taxonomies into the e-COGNOS ontology.

To summarize, this chapter presented an overview about controlled vocabularies. It was discussed what are they, and what are they for. That is to say, that CVs reduce the ambiguity that the words of a language may have when these are used. Moreover, the advantages and disadvantages of a CV were discussed. It was presented also the types of CVs available, specifically, Vocabulary, Dictionary, Thesaurus, Taxonomies and

[4] O'CoMMA is a public ontology from CoMMA project. This ontology includes 470 concepts in a taxonomy, 79 relations in a taxonomy, 715 terms in English and 699 in French to label the primitives, and finally 550 and 547 definitions in English and French respectively. (Gandon et al., 2002)

Ontologies, as well as a nouvelle type of vocabulary, Folksonomy, which is an uncontrolled vocabulary, which relies on social tagging.

Secondly, the controlled vocabulary Ontology was discussed in more detail. It was explain its main purpose. Additionally, how to build one and what are the components of an ontology. Specifically, Concepts, Meaning and Relations, Instances and Axioms and Rules. Furthermore, a non-exhaustive list of formalisms to describe an ontology were presented with a brief explanation for each. The ontology will be the CV in which this dissertation will try to propose a method to maintain it.

Thirdly, it was defined what is Ontology Learning, an approach to deal with the cumbersome task of building and maintaining an Ontology. OL aims also to transform these tasks into a semi- or fully automatic processes, which are yet very far to happen. In this sense, OL is a very important task as an ontology can be very difficult to maintain.

Lastly, it was made an overview through the initiatives of CV over history. One of them is the inspiration for this work, specifically, E-Cognos, an ontology from the B&C sector. An adaptation from E-Cognos ontology will be used in work developed under this dissertation.

3

Pattern extraction from unstructured sources of information

This chapter will bring to attention techniques and methods about extracting patterns from unstructured sources of information. It will be presented the area that deals with knowledge discovery, specifically Data Mining and why is it important. Moreover, the problem of pattern discovery will be described and presented as well as possible solutions to this problem, specifically, algorithms for the pattern extraction problem. Another point to read in this chapter is the comparison between the most used algorithm and its competitors, describing the advantages and disadvantages of their adoption. Lastly, it will be explained how to extract interesting rules from frequent patterns supported by Association Rules method, as well as how can a rule be evaluated and measured, how to quantify the interest of a rule and which measures are available.

3.1 Definitions

In order to refresh the memory, and to help better understand the contents of this chapter, is important to bring again to attention definitions such as *pattern, data*, *information* and *knowledge*. These definitions are presented as follows:

- *Data* is related to individual facts or items which are stored in a computer, in its raw digital form, normally in databases.
- *Information* is defined as sets of organized data in order to provide some sense as well as a context to the data itself. In computer science, information is important

or useful facts which results from input processing in a software tool. That is to say, when data is interpreted and associated to a context, it becomes information.

- As stated in Chapter 1, in the area of information retrieval and text mining, a *Pattern* is defined as a predictable occurrence that repeats itself along some (text) data.
- Finally, *Knowledge* is defined as "*awareness, familiarity, or understanding of someone or something (e.g. facts, information, descriptions or skills), acquired through experience or education by perceiving, discovering or learning*." (Oxford University Press, 2006) Knowledge provides meaning to information.

3.2 Pattern Discovery and Knowledge Extraction

The search for patterns aimed to explain and to provide context and meaning for daily common situations is already applied for years by people and is well installed in society motivations. For instance, when a man wants to buy a gift to his girlfriend, tries to remember what she will like. It will search in his mind through all data and information available in it, and try to discover a pattern such as "she likes butterflies" or "she likes red" to make sense of the gift to buy. Similarly, when a farmer wants to know where is the best place to plant seeds and when is the best time to produce a crop, he can search through some data in his mind, or in other place (e.g. a database) to discover. The motivation to discover patterns extends into many more activities, for example, doctors want to discover patterns in the clinical data of their patients, and similarly supermarkets want to discover patterns in the transaction data of their clients and lastly astrophysics want to discover patterns in the data of stellar bodies. One common idea is present in all previous motivations, meaning there is always an entity who wants to discover patterns that could be useful and could originate (produce) knowledge which could help make a good decision.

In the same way, the growth of computer systems through a massification in data storage, with bigger disk drives or online storage, as well as the ease to access a computer by anyone from any area, allow for all data acquired and stored from every field to also grow. Studies estimates a growth of data in world database storages by the double in each 40 month period in the following years (Hilbert and López, 2011). It would be a waste of time, and other resources not to take advantage of the data already stored, to help making better decisions. In this way, new opportunities arise for the discovery of patterns in the

data stored, in which could provide interesting facts to data owners. It would be interesting to have methods and techniques that could help easily search a big volume of information and find knowledge that otherwise would be very hard to find through manual processes.

3.2.1 Data Mining / Knowledge Discovery in Databases.

Several processes can be used for a system be able to discover patterns and further extract knowledge from data and information sources. Data Mining is one of them (Hand et al., 2001) Data Mining (DM), a subfield of Computer Science, is an area that studies methods to discover useful patterns in data sources aimed to help data owners take good decisions. That is to say, the goal of DM is to extract useful patterns from data which could be understandable and later used in the decision making process, in other words, to extract useful information from data sources and transform it into knowledge. Data Mining (DM) allows experts to find knowledge in new data or data they already have. As a result, by adopting DM techniques, decision makers can use newly discovered knowledge that otherwise could be unknown, unavailable or difficult to discover, in order to improve the decision making process. (Witten et al., 2011)

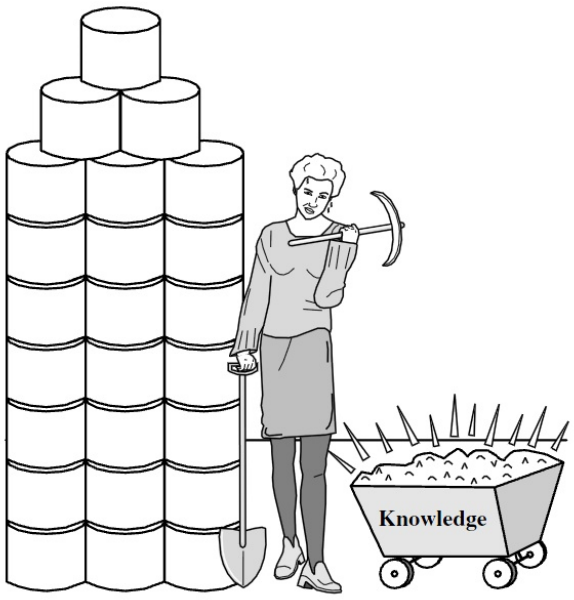


Figure 3.1 - Data mining—searching for knowledge (interesting patterns) in data. (Han et al., 2011)

Data Mining name was inspired in an analogy, easy to understand, from gold mining industry, specifically, in which they search (mine) through raw material, as rocks in order to discover and extract what is really valuable, the gold. In the case of DM, the "gold" is the knowledge that is extracted from the raw data (Figure 3.1). In literature, the

process of pattern discovery and extraction from data sources can be found under several names, such as DM. Likewise, it can also be found as Knowledge Discovery in Databases (KDD) (Fayyad et al., 1996), Information Discovery (Steyvers et al., 2004), Information Harvesting (Memon et al., 2007), Data Archaeology (Brachman et al., 1993), Data Pattern Processing (Inmon and Osterfelt, 1991), Predictive Analytics or Data Science (Waller and Fawcett, 2013). Additionally, is also sometimes confused with other areas as Machine Learning or Information Retrieval as these two areas can overlap DM in arranging solutions to data and information problems. Is worth mentioning, that mining data sources in the form of text, is a particular case of Data Mining, and is referred to as Text Mining (TM). All the previous names can mislead the reader. However, in order to make this point clear, in the present document the process of discovery and extraction of patterns will be referred to as Data Mining and when data is specifically related to text, will be referred to as Text Mining.

DM typically uses data sources already collected. Data can be of any type, structured, semi-structured or unstructured, which include databases, data warehouses, web content, text documents, images, sound, etc. Google search engine (Google, 2013), is a software tool from a well-known company which relies on DM techniques to discover patterns from data sources. They take advantage of search engine queries in order to discover knowledge for further use. One example is the use of such query data to discover what people from a specific country commonly looks for, in order to retrieve faster the desired geographically based query answers. Furthermore, other example of DM use is the Page Rank algorithm that Google uses to search pages and rank them according to the ones that are being pointed by more pages (Loukides, 2010).

In order to go from data to knowledge, Data Mining process relies in several steps (Fayyad et al., 1996). Specifically, these include the following three: *Data Preprocessing, Pattern Discovery* and *Pattern Evaluation* (please refer to Figure 3.2 below for an illustration of this process**).** Initially data is not yet ready to use, it needs to be pre-processed and prepared in order to apply techniques for pattern discovery. *Data Preprocessing* is related to the selection and preparation of appropriate data to use, or is related to execution of operations for data size reduction, all without sacrificing its integrity. For instance, specifically in Text Mining, these can include the removal from

data sources, of specific words[5] without any significant semantic added value to the process. Another illustration of this is the reduction of the size of a data corpus, by reducing each word to its *lemma* word[6] and removing the duplicates.

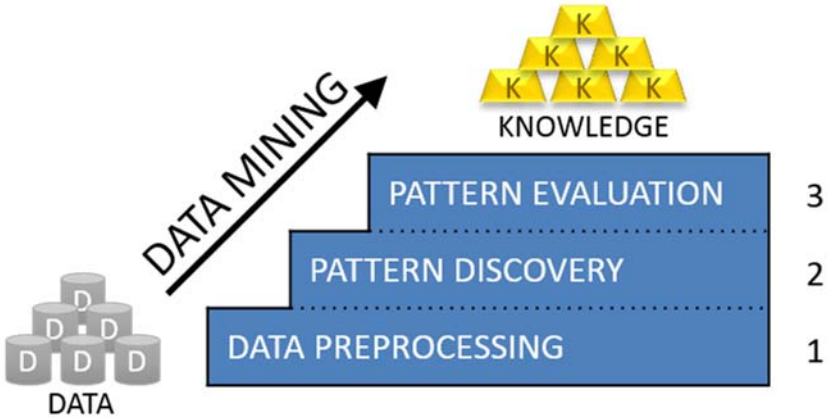


Figure 3.2 - Data Mining Process – Steps from Data to Knowledge

Pattern Discovery step relates to the techniques applied to pre-processed data in order to discover patterns. These include techniques as *Apriori*, *FP-Growth* or *ECLAT* algorithms, which will be further discussed in the following section 3.3.

Finally, *Pattern Evaluation***,** the third step in the DM process, is related to the identification of the useful and interesting patterns which can originate knowledge. This is achieved through *Association Rules* and interest evaluation, discussed in the following section 3.4.

3.3 Pattern discovery - Techniques

The typical techniques (algorithms) used for frequent pattern discovery are based in the following three: Apriori, FP-Growth and ECLAT algorithms. These techniques were introduced by its authors in the academic community as being capable of discovering frequent patterns (frequent itemsets) from a data corpus and to serve as a base for Association Rule Mining (ARM). And how can a pattern be considered frequent? A pattern is considered frequent when the frequency in which it occurs in the data sources is above a minimum threshold value called *support* (refer to sub-chapter 3.4.4) which normally is human defined. For instance, if one consider a set of data related to B&C

[5] These words are referred to, in NLP literature as stop words, and can include articles (e.g. “the”, “that”, “a”, “an”) or prepositions (e.g. “by”, “from”, “in”).
[6] *Lemma* word – a lemma is the canonical form of a set of words from the same family. E.g. Words *friend*, *friendship*, *friendly* have its lemma as *friend*.

materials such as "*brick, back door, window, front door, door knob, floor, wood*" and consider minimum *support* equal to 2, the algorithm will consider "*door*" as a frequent item as it is discovered in three items: "*back door*, *front door* and *door knob*".

3.3.1 Apriori

Apriori algorithm, was one of the pioneers to address the pattern discovery problem. This algorithm was presented by Agrawal and Srikant (Agrawal and Srikant, 1994) who also brought to attention the association rules problem. It is considered as a starting point for many studies in the academic community related to frequent pattern discovery.

Agrawal and Srikant define *Apriori* algorithm as a procedure for candidate itemset generation and test. To put it in another way, this algorithm generates potentially frequent patterns as candidates and evaluates them one by one in order to discover the frequent ones. It uses a search methodology of breadth first search (BFS)[7] technique as it works in horizontal laid out (each row is a transaction with all items in which they occur) data source representation (refer to Table 3.1a)). One of the characteristics of this algorithm is the property described as: every subset of a frequent itemset is also frequent.

Table 3.1 - Horizontal and Vertical layout representations

TID	Itemset
1	Bread, Butter, Jam
2	Butter, Coke
3	Butter, Milk
4	Bread, Butter, Coke
5	Bread, Milk

a) Horizontal Laid Out Representation

Items	TID set
Bread	1, 4, 5
Butter	1, 2, 3, 4
Jam	1
Milk	3, 5
Coke	2, 4

b) Vertical Laid Out Representations

The authors explain how this algorithm works, as follows: This algorithm includes two major steps. The first is a read through the data corpus to calculate the *support* value for all items and determine which ones are large items, *i.e.* have its *support* above a *minimum support* defined value, resulting in a new itemset with the large items as candidate sets to the next iteration. This will be done recursively, using the itemset from the previous iteration, until the largest itemset is discovered, and done as many times as

[7] **Breadth-first search** (**BFS**) is an algorithm for traversing or searching data structures. It starts through the initial elements and explores its neighbours before advance to the next element.

the maximum itemset available in the same data sources. The second step of Apriori algorithm is to make another read through the candidate data sources to discover all frequent items, by counting its frequency. As a result, this algorithm obtains a group of frequent patterns prepared for pattern evaluation, that is to say, association rule mining.

Table 3.3 describes a set of advantages and disadvantages identified in Apriori algorithm.

Table 3.2 - Advantages/Disadvantages for APRIORI algorithm

APRIORI	
ADVANTAGES	**DISADVANTAGES**
Easy to implement	Number of data source scans needed to generate candidate sets – As big as the maximum number of elements in an itemset.
Easily parallelized	Assumes transaction database is memory resident.
Uses large itemset property	

Apriori is an algorithm crossing several scientific areas which deals with data. One example of this is a recent study presented by Kumar and Chadha (Kumar and Chadha, 2012), in which they take advantage of the *Apriori* algorithm to discover association rules in assessment data from the students of an university. Through this study they can discover several and revealing facts such as student's interest, curriculum design, teaching and assessment of methodologies that can affect students who have failed to attain a satisfactory level of performance in the Post-Graduation level.

3.3.2 ECLAT

ECLAT, which stands for Equivalence CLass Transformation, is an algorithm to find frequent patterns in data sources. This algorithm introduced by Zaki (Zaki, 2000) is aimed to improve the performance problems of Apriori-based algorithms, specifically minimization of I/O costs by reducing the number of database reads or even the reduction of the computational costs with more efficient search procedures. ECLAT needs just a reduced number of reads of the data sources and do not require any hash trees whatsoever as it generates frequent itemsets by only simple intersection operations. It can even handle *support* values lower than Apriori *support* values in large datasets.

ECLAT Algorithm uses a search methodology of depth first search (DFS)[8] technique as it works on vertical laid out (each item stores a list of the transactions in which it occurs) data source representation (please refer to Table 3.1b)).

This algorithm works as follows: The first step is to construct a vertical data representation, searching through the data sources and obtaining a list with all items, one per row, each with all transactions in which they occur, as a conditional data source table. The *support* of an item is the count of transactions in which it occurs. In the next iteration, the data source to use is the conditional table from the previous iteration, in which the (n+1)-itemsets are intersected to obtain a list of the transactions in which they co-occur, as the conditional data source table for the (n+1)-itemset. All iterations will continue recursively until all frequent items are discovered or no more candidates are available to test. In each iteration the support is counted. .

Table 3.3 describes a set of advantages and disadvantages identified in ECLAT algorithm.

Table 3.3 - Advantages/Disadvantages for ECLAT algorithm

ECLAT	
ADVANTAGES	**DISADVANTAGES**
Very fast support counting related to Apriori	Intermediate tid-lists may become too large for memory
No complex data structures-no hash trees	
Effective when the discovered frequent itemsets are long.	
Support value easily calculated	

3.3.3 FP-Growth

FP-Growth, which stands for Frequent Pattern Growth is an algorithm introduced by Han et al. (Han et al., 2000), that is applied in data mining, aimed to discover frequent patterns (frequent itemsets) in data sources as a base for further evaluation. In other words, this algorithm searches through a set of data to discover patterns, which are considered frequent. Many studies appreciate the popularity and effectiveness of FP-Growth

[8] **Depth-first search** (**DFS**) is an algorithm for traversing or searching data structures. It starts at an initial arbitrary element and explores as far as possible along each branch before advance to a neighbour element.

algorithm, moreover some authors claim “this algorithm is currently one of the fastest ones to mine association rules.” (Borgelt, 2005; Han et al., 2004; Paiva et al., 2013)

This algorithm is executed in three steps, firstly the build of a compact data structure in the form of a prefix-tree (meaning the nodes will only be based in frequent patterns of size 1), called FP-Tree, from the data corpus. The FP-tree is aimed to store quantitative information (as each node represent an item, with its support value) about the frequent patterns discovered and to prepare for the next step of this algorithm. It is worth adding, that the FP-tree is arranged in such way, that the chance for node sharing is higher between frequently occurring patterns, than between less frequently ones; secondly, the development of a mining method of pattern fragment growth, based in the previous FP-tree, hence the name FP-Growth. This method is referred as a *divide-and-conquer technique*, meaning it separates frequent patterns from not frequent patterns in data sources; lastly, is the extraction of frequent itemsets directly from the FP-tree. The process will start in the less frequent nodes until it reaches the top node (bottom-up approach).

Table 3.4 describes a set of advantages and disadvantages identified in FP-Growth algorithm.

Table 3.4 - Advantages/Disadvantages for FP-GROWTH algorithm

FP-GROWTH	
ADVANTAGES	**DISADVANTAGES**
Uses compact data structure (FP-Tree)	FP-Tree may be expensive to build
One database read	FP-Tree may not fit in memory
No candidate set generation	Rebuilds conditional FP-Trees
Simple – Counting items and building a tree structure only operations	
Very fast to discover frequent patterns	

Due to its popularity and performance, FP-Growth is an algorithm much appreciated and used. An example of this is a study by Korczak and Skrzypczak (Korczak and Skrzypczak, 2012) which illustrates the use of FP-Growth algorithm for frequent pattern discovery in a transaction database from an online store, aimed to discover association rules between the customers transactions. Based in this algorithm, the authors discovered frequent patterns in transactions whose shopping value was above 50 euros and contained more than 2 items, which made them discover interesting behaviour by the

customers, as "*more than 2% of the customers who buys tomatoes and a chicken always buy new potatoes*", or "*more than 2% of the customers who buy white grape and watermelon always buy tomatoes*".

3.3.4 Algorithm comparison

As seen in the previous sub-sections all the referred algorithms have advantages and disadvantages. With this in mind, several comparisons in the academic community were made between them to evaluate their performance and discover if any outperforms the others. Garg and Kumar (Garg and Kumar, 2013) presented an interesting study with a comparison of the algorithms. Specifically, the proposed study compares Apriori, Eclat and FP-Growth under several scenarios, such as testing their behaviour when there is an increase of the transactions quantity, meaning an increase in the data source size (Figure 3.3); moreover, testing their behaviour when there is an increase of attributes and testing their behaviour when last two scenarios are used at the same time. The authors measured the time performance under all the latter scenarios. As a conclusion, is argued that "*FP-Growth is the best among the three, and thus more scalable*". Moreover, the authors add that "*Apriori is the worst in the performance tests*" especially when the data corpus increase in size.

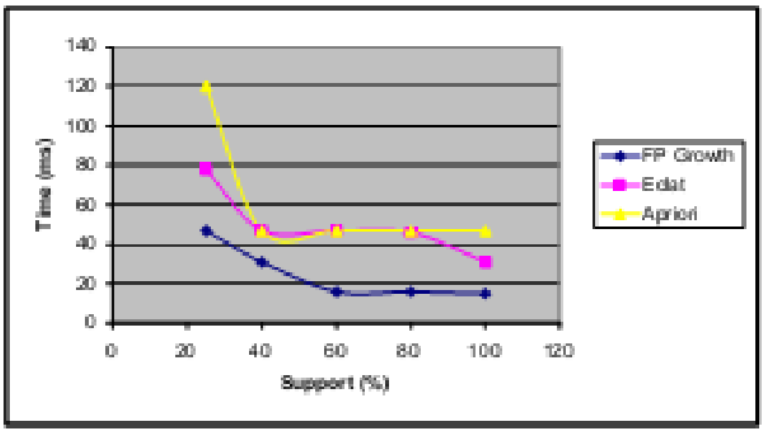


Figure 3.3 - Performance comparison of Apriori, Eclat and FP-Growth (Garg and Kumar, 2013).

Borgelt (Borgelt, 2005) presents another interesting study which also compares Apriori, Eclat and FP-Growth algorithms. The author uses the FP-Tree building preprocessing, which represents the most expensive (in time performance) operation made by FP-Growth, as the compare element. The conclusion was similar to previous study from Greg and Kumar, specifically the author argues, based on the experimental

results, that FP-Growth "*clearly outperforms Apriori and Eclat, even in highly optimized versions*".

3.3.5 Improvements Attempts

For illustration purpose, Table 3.5 describes a few algorithms found during the research for this dissertation, which are based in the three algorithms discussed, and present themselves as potential improvements. The intention of this table is not to provide an exhaustive list, on the contrary, just a small overview.

Table 3.5 - Improvements attempts for Apriori, Fp-Growth and Eclat algorithms

Name	Description	Source
APRIORI		
MSApriori	Uses the same procedure as Apriori, but uses more than one minimum support value.	(Liu et al., 1999)
A-Close	Uses a closed mechanism to find frequent patterns, specifically it is based in pruning a closed itemset lattice.	(Pasquier et al., 1999)
Apriori-Inverse	Discovers sporadic rules (low support but high confidence) by ignoring all candidate itemsets above a maximum support threshold.	(Koh and Rountree, 2005)
UApriori	Searches frequent sequences in the presence of existentially uncertainty (considers attribute level (existential) probability) in transactions	(Metanat Hooshsadat et al., 2012)
FP-GROWTH		
Top Down FP-Growth	Searches the FP-Tree through the Top-Down approach as opposed to normal FP-Growth (Bottom-Up). Does not generate conditional pattern bases nor sub-FP-trees.	(Wang et al., 2002)
FP-Bonsai	Uses a data reduction technique to reduce FP-Tree structure size, based in pruning operations, aimed for performance improvement, specifically memory size reduction and disk writing frequency reduction. Resulting an FP-Tree referred as FP-Bonsai.	(Bonchi and Goethals, 2004)
nonordfp	Improves the performance (memory efficiency) problem of conditional FP-Tree rebuilding, by using just one compact structure without the need to rebuild conditional trees.	(Rácz, 2004)
Painting-Growth	Improves by just using one scan to the database, and builds an association picture to find the association sets.	(Zeng et al., 2015)
ECLAT		
MaxEclat	Improves Eclat, ordering itemsets in transaction lists by their size (long first) based in their support values. It uses an hybrid search, as opposed to bottom-up search from Eclat. Discovers the long maximal frequent itemsets, and some nonmaximal ones.	(Zaki, 2000)
dEclat	Uses an approach called Diffsets Eclat (dEclat), which uses the count difference between transaction itemsets (tids), as opposed to using the complete tids. Improves the computation size problem, reducing the memory necessary for frequent pattern mining.	(Zaki and Gouda, 2003)
Bi-Eclat	Improves Eclat, by sorting the transactions by its frequencies using descending order. Additionally, in support count uses ascending order.	(Yu and Wang, 2014)

3.4 Association Rules – Pattern Evaluation

Introduced by Agrawal et al.(Agrawal et al., 1993), Association Rules (AR) is an algorithm used after frequent pattern discovery, to evaluate frequent patterns and identify tendencies and relations (or associations) between frequent items in data sources resulting in knowledge, which could be used to try to make predictions over potential scenarios. In other words, the main goal is to use the knowledge discovered from data sources already existent to help improve the decision making process.

$$A \Rightarrow B \wedge A \cap B = \emptyset \quad (1)$$

An AR can be defined as an implication as represented by equation (1). Each AR is composed by two sets of items, *A* and *B*, called *itemsets*. Each itemset can hold one or more items, each belonging to the same data source. Moreover, the intersection of A with B is an empty set, that is to say that A and B are two different itemsets. Additionally, A and B are referred to as premise and conclusion[9], respectively. Premise represents the initial occurrence to evaluate, and conclusion represents the occurrence of what was determined by the rule. Therefore, an Association Rule can be read as follows: If a premise A occurs in a data corpus then conclusion B will likely also occur.

Table 3.6 - Frequent searches in a search engine from a motor store

ID	Search terms
1	{“car”, “motorcycle”, “race car”, “Yamaha”, “luxury car”, “small car”, “boat”}
2	{“race car”, “Yamaha”, “luxury car”, “small car”}
3	{“race car”, “Yamaha”, “small car”, “motorcycle”}
4	{“race car”, “Yamaha”, “small car”}
5	{“luxury car”, “boat”, “Yamaha”, “race car”}
6	{“motorcycle”, “boat”, “small car”}
7	{“Yamaha”, “luxury car”, “boat”}

As a practical example of an AR, one can imagine a database with the most frequent queries in a search engine from a store that sells motors as represented in Table 3.6, and considering a rule with two itemsets, A={“race car”, “small car”} and B={“Yamaha”}, one can say when the items “*race car*” and “*small car*” occur in a query, the item “*Yamaha*” also occur. The former example provides knowledge to the data

[9] Although, these two itemsets can be found in literature with other names as antecedent and consequent (Hoque et al., 2011), this thesis adopts the nomenclature of premise and conclusion (Costa, 2014).

analyser, in the sense it can potentially predict that whenever someone looks for a race car and a small car, it will likely search for a "Yamaha" motor.

Another illustration that also helps to clarify the objective behind AR mining, which is usually referred by the academic community, is an example based on market basket analysis and its transactions in a department store. In such example, the objective is to predict customer behaviour, based on the collected data from several transactions. Specifically, if a customer buys product A, the AR algorithm, based on the stored transaction data of other customers, will be executed and its results will try to predict the behaviour that potential customers will have, or in other words, which product B will the customer likely buy.

Table 3.7 –Example of Association Rules data type- a) Boolean, b) Quantitative

ID	Month	Humidity	Temperature	Weather
1	Jan	HIGH	LOW	RAIN
2	Feb	HIGH	LOW	RAIN
3	Mar	HIGH	LOW	RAIN
4	Apr	HIGH	LOW	RAIN
5	May	HIGH	LOW	CLOUD
6	Jun	HIGH	LOW	CLOUD
7	Jul	MEDIUM	HIGH	SUN
8	Aug	LOW	HIGH	SUN
9	Sep	HIGH	MEDIUM	SUN
10	Oct	MEDIUM	MEDIUM	SUN
11	Nov	HIGH	LOW	CLOUD
12	Dec	HIGH	LOW	CLOUD

a) Weather / Humidity / Temperature average conditions at day 10 of each month of 2013

ID	Age	Married	Cars
1	20	YES	1
2	35	YES	1
3	23	NO	2
4	22	NO	1
5	50	NO	0
6	28	YES	2
7	36	YES	2
8	30	NO	0
9	31	YES	0
10	45	YES	2
11	41	NO	1
12	37	YES	2

b) Number of cars owned by people older than 18.

Association Rules can be expressed in two different data types, *Boolean* or *Quantitative* (Hoque et al., 2011). Boolean Association Rules are expressed by true/false based values. Consequently, premise and conclusion values are also either true or false, for instance, if A = true then B = true, or if Humidity=HIGH and Temperature=LOW then Weather=RAIN or CLOUD (Table 3.7a)). Quantitative Association Rules are expressed by items that holds a numerical representation with more than two values (e.g. age intervals as [18-35]). An illustration of this is presented by Table 3.7b), where if Age > 34 and Married=Yes then Cars > 1.

3.4.1 Association Rules - State of the art

Association Rules is a technique which discovers knowledge in order to improve decision making processes. Since its introduction in 1993 by Agrawal et al. (Agrawal et al., 1993) several initiatives were made to take full advantage of their potential with interesting results across several areas around the world. Such as Medicine, Chemical Science, Business, Culture Heritage, Languages, Education, Sociology or Building and Construction. In the following lines, several initiatives will be presented. These initiatives take advantage of the high potential of AR application to discover interesting and surprising knowledge.

For instance, in Medicine, Mahgoub (Mahgoub, 2006) applied AR to discover knowledge from MEDLINE[10],.a scientific online database. His goal was to discover knowledge from medical publications related to the H5N1 avian influenza virus outbreak. He achieve it by searching through more than 8000 keywords from the abstracts of 100 publications and discovering a set of associations that could later be evaluated. For instance, he was able to discover that when the keywords "*highly, pathogenic and avianinfluenza*" co-occur in the abstracts, the keyword "H5N1" will also occur.

Coming from the Chemical Science area, an interesting application of AR was the study from Azevedo et al.(Azevedo et al., 2005). The authors analysed a massive quantity of molecular simulations, aimed for the discovery of associations between the constituents of the molecular systems. As a conclusion, through the use of this technique, they were able to successfully discover 28 hidden associations between the 127 constituents from the simulations, thus discovering new biochemically relevant knowledge.

In Medicine, Tan et al. (Tan et al., 2009) presented a study which took advantage of the AR potential by trying to discover knowledge from the contents of medical brain images. This process was aimed to help doctors better diagnose brain threats in medical images (specifically, in previous defined regions). After discussing their results with medical specialists, comparing real diagnosis, the authors concluded that their experience was more than 90% effective, which is a relevant value for the use of AR.

[10] Medline (PubMed) – Wide library of scientific publications in the medicine research field. Please refer to http://www.ncbi.nlm.nih.gov/pubmed/ for more information.

From Culture Heritage area, Tomi Kauppinen (Tomi Kauppinen, 2009) examined a semantic portal database (*viz.* CULTURESAMPO), aimed to propose the enrichment of an ontology (*viz.* Finnish General Upper Ontology - YSO) through the use of AR technique. Specifically through the discovery of interesting associations between the concepts used to describe art objects (*viz.* paintings, photographs, etc.) from the database. Using AR technique, the author was able to analyse more than 4700 concepts and successfully discovered relevant associations, which could be proposed for the ontology. Such as an association between "*musical instruments*" (discovered 385 times) and "*folk music*" (discovered 125 times) as they were discovered together in the same descriptions 125 times. Based in this technique, the author were able to propose to the ontology engineer who was responsible for the ontology itself, 23 out of the top 50 associations discovered to be added to the ontology.

More areas took advantage of AR technique, such as Education. In this area, Kumar and Chadha (Kumar and Chadha, 2012) presented recently a case study aimed for the improvement of the education quality of an university. Through the use of AR, the idea was to discover the factors leading to a high student success rate by analysing academic results of the students. For instance, they discovered knowledge such as "*students who have scored badly in their Graduation have done relatively well in their Post Graduation in the subjects which are common in both Graduate and Post Graduate courses.*" Based in the new knowledge discovered, the university could use it to change methodologies or redesign the curriculums aimed for the benefit of both students and university.

Another interesting example of the AR application was presented by Spruit (Spruit, 2007) in the Languages area. The idea of this study was to examine associations between syntactic variables in the 267 Dutch Dialects (from Netherlands, Northern part of Belgium and a small northwest part of France), aimed to discover the geographical distribution of the variations of the words from the dialects. This study evaluated more than 234000 rules, discovering more than 10000 interesting associations with 90% accuracy. By using AR technique the author discovered interesting knowledge about, for instance, the geographical region where a specific word is more used.

The use of AR was also applied to Sociology area. Specifically, to the demography of Washington DC through a study presented by Brin et al. (Brin et al., 1997). These authors took advantage of AR techniques to discover interesting knowledge from a

demography study in the capital of the USA. Based in the census data (more than 30000 entries) from 1990 they were able to discover more than 23700 rules, from which they discover interesting knowledge, such as "*unmarried people have moved in the past five years*" or as "*African-American women are not in the military*".

One study from the Business area, was presented by Hoque et al. (Hoque et al., 2011) aimed to discover knowledge from a company employee database that could provide clues about the behaviour of their employees, to assist in the decision making process for their benefits. From a database of 10 employees, by using AR technique they were able to discover 35 association rules. Such as "*all the low experienced, with medium salary and master education level are young*", or as "*all the PHD education level employees have high salaries*". Through the newly discovered knowledge the organization was able to reflect about some opportunities, as salary increment or a promotion.

Other study from Business area was presented by Korczak and Skrzypczak (Korczak and Skrzypczak, 2012). This study proposes the discovery of interesting knowledge from the analysis of customer transactions in an online store (*viz.* Delicatessen Alma24), aimed to assist decision makers better understand customer behaviour, in order to improve sales performance. Through the application of AR in more than 1000 transactions, the authors were able to discover 32 interesting rules. Based in these rules they were able to discover interesting knowledge, for instance, that "*at least 2% of the customers who buys tomatoes and a quarter chicken, always buy the new potatoes*" or that "*at least 2% of the customers of red pepper and Hojnowka butter always buy tomatoes per kg*". Based in this new knowledge the decision makers were able to better understand the choices of the customers and consequently improve the success of the business.

All in all, Association Rules is a very useful technique to make sense out of data. It is largely used amongst several areas, by data analysts from companies, institutions, stores, museums, health care, etc.

3.4.2 Interest Evaluation in Association Rules

Interest is the feeling of wanting to know or learn about something or someone (Oxford University Press, 2006). *Evaluation* is the making of a judgement about the amount, number or value of something or someone (Oxford University Press, 2006). The

discovery of what will be interesting in a data corpus is the main goal of AR. In order to find interest in useful patterns they need to be evaluated somehow, but how to evaluate an association rule? Moreover, how to know which association rules are interesting? Is an association rule interest in two contexts? The interest of an association rule can be discovered through evaluation of their interestingness, by making a judgment of an association rule based in the context against which is evaluated. In this matter, some considerations and thoughts can be made when evaluating knowledge and specifically an association rule. Firstly, what should be evaluated and what should be considered interesting; lastly, how to evaluate interesting association rules.

In order to discover interest, it is important to know that the first factor to consider is the context in which an association rule needs evaluation. For instance, if the context is building and construction, association rules discovered including houses or buildings could be more interesting than those who include computers or photography. In contrast, if the idea is to discover association rules about houses or photographs of buildings, then the interest on photography rises, transforming one uninteresting context into an interesting one. Therefore, the context is one important factor to consider and should be carefully chosen when evaluating an association rule as it can help to provide the best and results related to the interest considered.

Yao et al. (Yao et al., 2006) argue that there is another factor that could influence the evaluation process of an association rule, and thus its interest. This factor is the user who evaluates an association rule, who can be a person (or an organization or a system). Each user is unique and this can lead to different points of view between different users when discussing about the same idea. Moreover, the background or even the geographical location of a user are factors that can lead to different interpretation or perception about a subject, thus influencing the judgement of the user evaluating an association rule. Therefore, a user can play a crucial role in the judgement of the interest, meaning that the practical results of an evaluation process of association rules could also depend on the subjectivity of a specific user who participates in the evaluation process.

Several studies in the last few years discussed about methods to overcome the drawback of evaluating the *interestingness* of knowledge discovered from association rules. There are two ways to measure the interestingness of an association rule, specifically subjectivity and objectivity measurements (Hilderman and Hamilton, 1999; Silberschatz and Tuzhilin, 1995). Additionally, Mackie (Mackie, 1977) argues that the evaluation of subjectivity is very common when the goal is to evaluate actions or events.

Moreover, the author adds that the objectivity is used when there is the need to quantify a value, thus being reflected by the measures themselves. The subjectivity is influenced by the entity that is evaluating a subject. As explained before, it can also be influenced by factors such as the location or the background. Figure 3.4 illustrates, the classification of the types of measures aimed for interestingness evaluation of an association rule.

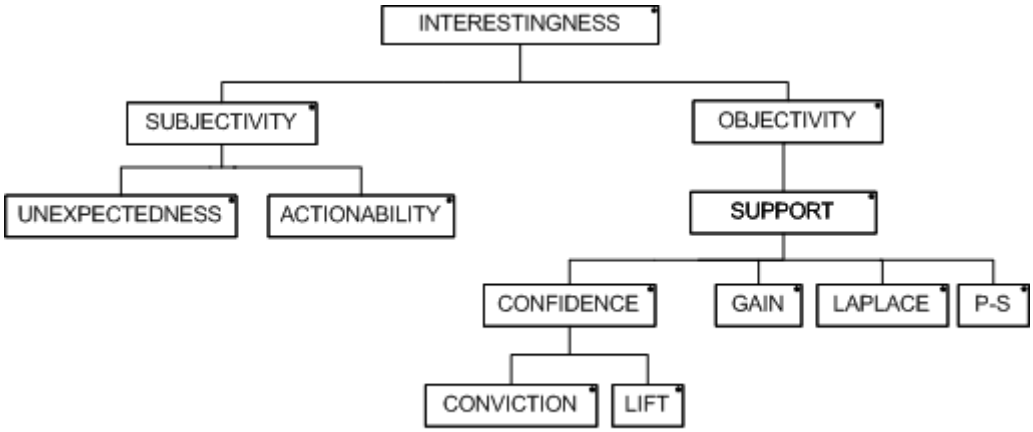


Figure 3.4 – Classification of interestingness measures (adapted from Silberschatz and Tuzhilin, (1995))

3.4.3 Subjective measures

To measure the subjectivity, Silberschatz and Tuzhilin (Silberschatz and Tuzhilin, 1995) identify two concepts, specifically *unexpectedness* and *actionability*. The first concept represents the value of something unexpected or a surprise in an association rule when knowledge is discovered. That is to say, if one could discover an association rule that would not expect, that rule would be an interesting one. The authors argue that, the knowledge discovered which was expected, is knowledge that is already known, and thus, not interesting for the user in this sense. The second concept of subjectivity is actionability, it represents the usability that an association rule could have. In other words, it is the capacity of a rule to be used in an interesting way by its user. One practical example of the use of unexpectedness and actionability concepts applied to association rule mining is the study presented by Gonçalves (Gonçalves, 2005). This author explains how these concepts can be used in association rules, by applying them in the transactions of a department store. The author discovered that when the transactions were made by young couples on a Thursday, there was an association between dippers and beer. Initially, the analysts would think that the act of buying beer would just be associated with the act of buying appetizers or barbecue meat and other alcohol drinks. Surprisingly, when the association rules were discovered, this unexpected knowledge was also

discovered. This is a good illustration of what an unexpected and actionable rule is. As a result, the department store, for now on, on Thursdays, can apply this new and unexpected discovered knowledge to move the dippers and beer closer to each other, so the sales of both could increase.

Is worth adding that, although actionability and unexpectedness concepts are independent of each other, they can be combined to strengthen even more an association rule. Although this could not always be true, frequently, the unexpected rules are also actionable rules, the ones that a user can do something useful with them. Similarly, the actionable rules, are also rules unexpected to appear. If the objective of association rules mining was to discover knowledge already known, what would be the point of doing it? Moreover, what can be done to the knowledge that is already known? Could it be used to improve something? Or can it be discarded? More research is needed in order to discover answers to this questions.

3.4.4 Objective measures

The other type of interestingness measures, are the ones to measure objectivity. These measures quantify the strength of association rules, reflected in a numeric value as a result. Examples of this can be the measures *support*, *confidence*, *conviction*, *lift*, *gain*, *laplace* and *ps*, which will be discussed in this sub-section.

Support and confidence

Since the presentation of association rules by Agrawal et al. (Agrawal et al., 1993), many authors illustrate the use of two specific measures that could assist the evaluation of interest in association rules: *support* and *confidence*. In fact, several authors, proposed studies which take advantage of these two measures to quantify the interest of an association rule (Azevedo et al., 2005; Bayardo and Agrawal, 1999; Bhujade and Janwe, 2011; Brin et al., 1997; Hoque et al., 2011; Kumar and Chadha, 2012; Spruit, 2007). All such studies are a proof that these measures are relevant and should be considered when evaluating the objective interestingness in association rules.

The *support* (also *frequency* or *coverage*) of an association rule is a statistic measure, defined as the number of occurrences in a data corpus where both premise A and conclusion B occurs simultaneously. That is to say, the support of an association rule, is defined as the percentage of the transactions containing both itemsets A and B. The

values from this measure are enclosed between $[0..1]$, and are normally represented in percentage. The support result is proportional to the frequency. That is to say, higher the support value, the more frequent are the itemsets in the data corpus. Support is represented by the following equation (2).

$$Sup(A \Rightarrow B) = \sigma(A \Rightarrow B) \tag{2}$$

Confidence represents an estimation of the probability of observing itemset B given itemset A. When this measure is applied in an association rule, one can immediately quantify the strength of the association between both itemsets. The expression to calculate confidence is given in Equation (3) and its values are enclosed in $[0..1]$. One can also identify that the interest rises also with confidence results. That is to say that there is more interest in an association rule with higher confidence values. Association rules with value equal to 1 mean that given A, B always occur.

$$Conf(A \Rightarrow B) = \frac{\sigma(A \Rightarrow B)}{\sigma(A)} \tag{3}$$

Although both these measures individually can provide some interesting rules, this could not be enough. To get the real interesting rules, one has to consider two additional parameters, *minsup* and *minconf* (Bayardo and Agrawal, 1999). Bayardo and Agrawal argues that when there is lower limit defined, the rules with values above the limit are the real interesting ones. For instance, a rule can have a support value of 20%, however, if the defined minsup is 50% this rule is considered uninteresting. In contrast, if a rule have a support of 80% is considered a very interesting rule.

Conviction and Lift

Support and confidence can provide a good and trustful interestingness objective measure results, however sometimes they are not enough. As a result some other measures were studied and proposed by the scientific community, *Conviction* and *Lift* are two other statistic measures proposed to complement the objectivity measurement of an AR. These measures are commonly used to increase the strength of the values obtained from confidence and support measures, as these two depend on the confidence and support values to be calculated.

Conviction is an implication measure that quantifies the value of the implication. It is represented as $A \Rightarrow B$, meaning that the direction of the rule is important for the interest measurement, hence $A \Rightarrow B \neq B \Rightarrow A$. *Conviction* has some interesting properties, such as:

Property 1: If its value is equal to 1, this means that the concepts are considered totally independent from each other. That is to say, they do not have any kind of association.

Property 2: On rules with 100% as confidence value, the value of conviction is equal to ∞.

$$Conv(A \Rightarrow B) = \frac{\sigma(A) \times \sigma(\tilde{B})}{\sigma(A \cup \tilde{B})} \Leftrightarrow \quad (4)$$

$$\Leftrightarrow Conv(A \Rightarrow B) = \frac{1 - \sigma(B)}{1 - Conf(A \Rightarrow B)} \quad (5)$$

The most interesting rules based in this measure, are reflected by the higher result values, and higher this value higher is the interest of that rule. The result values of *Conviction* are found inside the interval $[0..+\infty[$. *Conviction* is represented by equation (4) or alternatively, by equation (5).

Lift[11] (also found in literature as *Interest* (Brin et al., 1997) or as *Strength* (Dhar and Tuzhulin, 1993)) is a measure to quantify the co-occurrence of a rule. In contrast to conviction, Lift is not an implication measure as it is symmetric in relation to premise and conclusion, hence $Lift(A \Rightarrow B) = Lift(B \Rightarrow A)$. That is to say, it measures how far from independence are itemsets A and B. Lift is defined as a measure to boost ("lift") the confidence of a rule, this suggests a trust increase in confidence results of a rule. This measure have better results in association rules with lower support values. Similarly to conviction, if its value is equal 1 it means the itemsets are totally independent, without any kind of interesting association. Moreover, as far from 1 and as higher the value, higher will be the interestingness of the association rule, which can reflect a higher association between the itemsets. The result values of this measure are included in $[0..+\infty[$. Lift is

[11] As a curiosity, Lift is known in the scientific community as a measure used in the Intelligent Miner from IBM (IBM - International Business Machines, 1996)..

represented by the following equation (6) or equation (7) where this value can be given taking advantage of confidence measure.

$$Lift(A \Rightarrow B) = \frac{\sigma(A \Rightarrow B)}{\sigma(A) \times \sigma(B)} \Leftrightarrow \quad (6)$$

$$\Leftrightarrow Lift(A \Rightarrow B) = \frac{Conf(A \Rightarrow B)}{\sigma(B)} \quad (7)$$

As can be easily observed in both equations (5) and (7) from conviction and lift, there is a relation to confidence measure. Therefore, these measures can be understood as measures to increase or strengthen the trust of confidence results when confidence itself would not be enough to make conclusions and evaluate interesting knowledge in the association rules.

Gain, Laplace and PS

Finally, the remaining measures illustrated in Figure 3.4 are *Laplace*, *PS* and *Gain*. These three measures will be discussed in the following lines.

Laplace is defined as a trust estimator which is function of support. This is reflected by the following: as low as support value is, lower is the interest in the rule considered. Laplace is normally used to rank rules by class. The range of their values are in $[0..1[$. Laplace is represented by equation (8). The constant *k* represents the number of classes chosen when someone is defining the respective classification model. Its value is always higher than 1.

$$Laplace(A \Rightarrow B) = \frac{\sigma(A \Rightarrow B) + 1}{\sigma(A) + k} \quad (8)$$

Gain is an optimization measure proposed by Fukuda et al. (Fukuda et al., 1996) as a measure to solve the optimized gain rules problem. It is a function of support and given by equation (9). The θ parameter is defined as a constant fraction with values between 0 and 1. Additionally, if one wants to decrease the subtractive term, it can only be done by decreasing the support of the premise. When this happens, the support value of the association rule becomes higher.

$$gain(A \Rightarrow B) = \sigma(A \Rightarrow B) - \theta \times \sigma(A) \quad (9)$$

The last of these three measures presented is *PS*. This measure receives its name from their creators, Piatetsky-Shapiro (Piatetsky-Shapiro, 1991)[12]. It was originally used to classify general rules, and later adopted as an association rule measure. As lift, this measure is a boost to the support measure. Its value is included inside the range $[-0.25..0.25[$. When this value is equal to 0 it means that A and B are independent. When this the value is below 0 represents a negative dependency and if the value is higher than 0 it is called positive dependency. Higher values mean more interesting association rules. The definition for PS is represented by equation (10).

$$PS(A \Rightarrow B) = \sigma(A \Rightarrow B) - \sigma(A) \times \sigma(B) \tag{10}$$

Other measures

All the measures illustrated in this section should arguably provide enough means to evaluate any association rule discovered, although, in the academic community several other measures were proposed over the years. For instance, Tan et al. (Tan et al., 2002) propose a study where they evaluate a list of 21 measures for association rules, which includes the ones previously discussed. Moreover, this study also includes the measures such as *Kappa*, *Gini*, *Added Value*, just to name a few. Further research should be made in the direction of more measures, to improve the association rules process reliability.

To summarise, this chapter presented an overview of data mining concepts, and how its process works. It was explained what is and how to analyse data in order to make sense out of it, in other words, how to discover knowledge from raw data, based in a 3 step model from Data Preprocessing, Pattern Discovery and Pattern Evaluation. As the present study is aimed more to the last two steps, these were explained in more detail.

Pattern Discovery step is responsible for the techniques to extract frequent patterns from data. Three main techniques were presented capable of discovering frequent patterns, specifically, Apriori, ECLAT and FP-Growth. These techniques were starting points to several other studies of potential improvements, which a non-exhaustive was presented. A study which demonstrates which one is the best technique from the former three was also discussed, concluding that FP-Growth is the one who performs better.

[12] In the literature *PS* is also found under different designations, for instance, *Leverage* (Azevedo and Jorge, 2007), *Rule Interest* (Gonçalves, 2005) or *novelty* (Lavrač et al., 1999).

Patterns Evaluation, is the step in Data Mining process responsible for the discovery of interesting knowledge. This is achieved through a technique referred to as Association Rules. This technique discovers associations between the patterns and evaluates them. This evaluation of interest is made through specific measures, subjective and objectives. Moreover, the use of this technique crosses several areas, such as Medicine, Education, Culture, Languages, Demography and others. This technique is relevant to use where there is the need to make sense out of data.

In fact, as the present dissertation aims to propose an approach for ontology learning, data mining techniques are going to be used in order to discover knowledge from unstructured sources of information from a document corpus. The former will be achieved based in the concepts discussed in this chapter, specifically, by using FP-Growth algorithm to discover patterns. Moreover, Association Rules technique will be also applied to evaluate the interest of knowledge discovered. Taking advantage of the measures available, as support and confidence. The interesting knowledge will be proposed in order to update a domain ontology.

4

Concept Model

This chapter aims at presenting a conceptual model for the present study, which is defined by the expected outcomes presented in this document (refer to Chapter 1.2). Specifically it will be proposed a model for pattern discovery and knowledge extraction, aimed for the improvement of a collaboration framework. In the following lines, firstly, it will be presented a validation scenario to prove the applicability of the idea; secondly, the actors identified as the ones who will benefit from this system, thirdly, the definition of the inputs and outputs of the system, lastly we will present how all the previous will be integrated in the system.

In order to help the reader visualize the idea of the present work, it will be used a modelling language (UML). UML is a visual representation language for model and system representation commonly used in software development. In this work, will be presented three UML views:

- Functional view, through use case diagrams (section 4.1.2)
- Static view, through class diagrams (section 5.2.5 – Static view)
- Dynamic view, through sequence diagrams (section 5.2.5 – Dynamic view)

4.1 Scenario

Considering an engineering project environment in a company with the participation of several actors from different areas, such as civil engineers, electrical engineers, architects, business analysts, etc. In this sense they need to collaborate in order

to achieve the goal of the project. For instance, this could mean the need to share documentation in all phases or participate in meetings where the understanding of the same ideas is a fundamental factor for the success. In other words, as the experts can have different points of view and originate from different backgrounds, the sharing of knowledge in all phases of the project needs to be smooth amongst all. This means, when every peer understands the same language, the project is easier and the execution time of the project is smaller. With this in mind, the opportunity arises for the need of a platform which could harmonize the communication and sharing issues. This represents an opportunity for the use of a controlled vocabulary and specifically an ontology.

4.1.1 Actors

To extract knowledge from unstructured data sources and use it to improve an ontology, it will be necessary the creation of a computational system aimed also for presenting the results of this process. However, besides the creation attempts of a completely automatic system for ontology maintenance, more research is needed in this area to achieve this vision. Consequently, the systems available are partial automatized, which means that they still need human intervention as a real factor. In this sense it is important to identify who will be the actors which are going to interact with the system.

Two actors can be identified:

- Domain knowledge experts;
- Ontology experts;

Domain knowledge experts are actors that hold deep knowledge in the domain in which the system will be applied. These experts can informally present, evaluate and validate new knowledge aimed for the ontology learning process. Ontology experts are the actors who know how to represent the knowledge and formally add it to the ontology. They will manually improve the ontology with the knowledge shared and validated from the other actor (e.g. adding a new concept or improving a relation)

4.1.2 Functional view

As referred above, the functional view will be described by UML Use Case Diagrams. The Use Case Diagram reflect how the actors will interact with the system. The functionalities defined for the each of the actors are presented by Figure 4.1. It is worth

noting that both Ontology and Knowledge Experts are capable of executing any of the functionalities presented below:

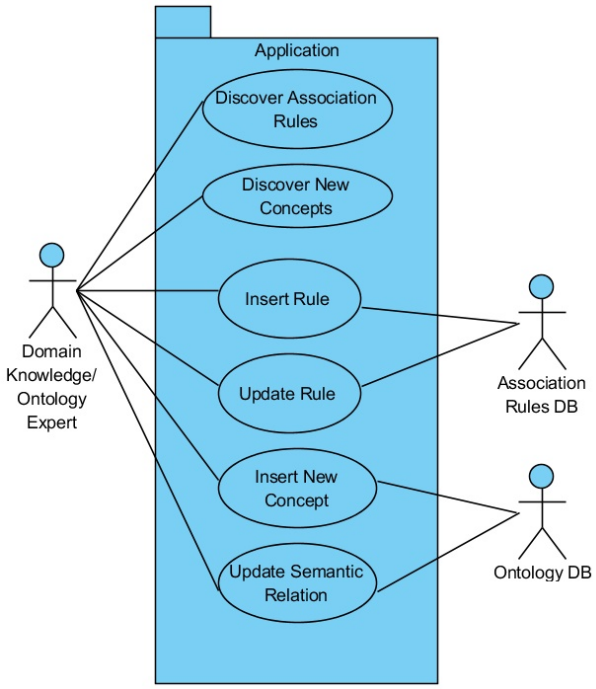


Figure 4.1 - Use case diagram for Domain Knowledge & Ontology Experts

- **Discover Association Rules** – Allows the actors to discover new association rules between the concepts of the knowledge sources.
- **Discover New Concepts** – Allows the actors to discover new concepts that are not yet in the domain ontology, which can be later added if relevant.
- **Insert Rule** – Allows the actor to insert a new rule discovered that is not yet in the association rules database.
- **Update Rule** – Allows the actor to update a rule that was already inserted in the association rules database.
- **Insert new concept** – Allows the actors to insert new concepts in the ontology database, which were discovered before.
- **Update Semantic Relation** – This functionality will allow the actors to update a relation between two concepts from the domain ontology.

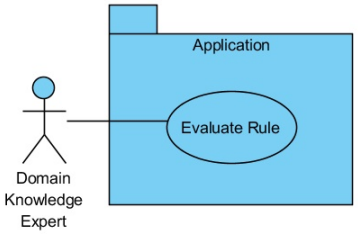


Figure 4.2 - Use Case Diagram for Domain Knowledge Expert

For the Knowledge Expert actor there is one use case which is specifically aimed for him. Figure 4.2 presents the Use Case, precisely *Evaluate Rule*, which allows the knowledge expert to evaluate if a rule makes sense or not in the domain where the system is applied.

4.1.3 Requirements

A set of requirements are identified in order to develop a proof of concept for the present model, which were divided in three types, Functional, Architectural and Technical (Figure 4.3). Although all these requirements are important, in the following lines only some of these requirements will be presented in more detail.

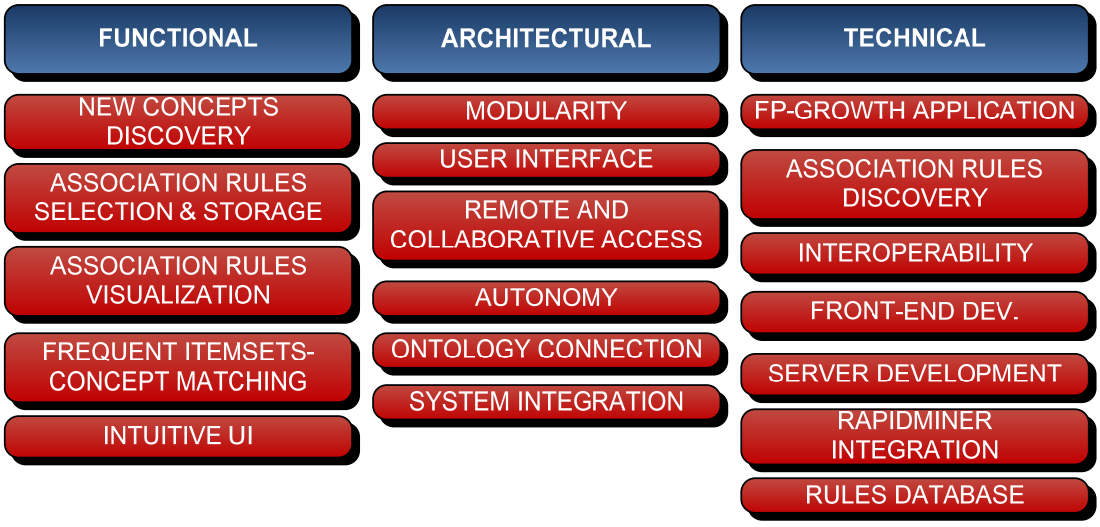


Figure 4.3 - System requirements

The functional requirements are as follows: *new concepts discovery*, this requirement relates to concepts discovered which are not yet in the domain ontology. These concepts need to be identified. Moreover, *frequent itemset-concept matching*, relates to idea to match the concepts discovered by the data mining process with the concepts already in the ontology. Another requirement refers to the presentation of the

association rules discovered, hence *Association Rules Visualization*, as well as the measures results, in an easy and understandable way.

The architectural requirements of the system should be built in a *Modularity* approach in order to simplify it, separate concerns and prepare for future improvements. The second architectural requirement is *Ontology Connection*, meaning it should have a connection to the domain ontology to allow for the matching between frequent itemsets and ontology equivalent terms. Moreover, the system should have a *Remote and Collaborative Access* to allow for several actors to use the system anywhere, if necessary remotely (e.g. Access it from a construction site). Lastly, should allow for *System Integration*, in order to be able to use the several technologies necessary (refer to Chapter 5 for more information about the technologies used).

For the technical requirements, the use of Data Mining techniques, namely *FP-Growth application* and *Association Rule Discovery*, are two requirements which were described in Chapter 3. The users will need a way to interact with the system, hence the creation of a *User interface* is a requirement that is necessary. *Rapidminer Integration*, means the use of Rapidminer API for the data mining process. Other requirement is the creation of a *Rules Database* to store the rules chosen by the actors.

4.2 Model

This work aims at propose a model for knowledge discovery and ontology learning, specifically to discover new concepts and relations between concepts discovered from unstructured data, which could be used to update a domain ontology. The model proposed herein is based in data mining process presented in Chapter 3.2.1 (Figure 3.2), which is composed by 4 components. Specifically, *Data Mining, Repositories, Ontology Learning* e *Knowledge Presentation* (Figure 4.4). A more in depth description of each, will be presented as follows.

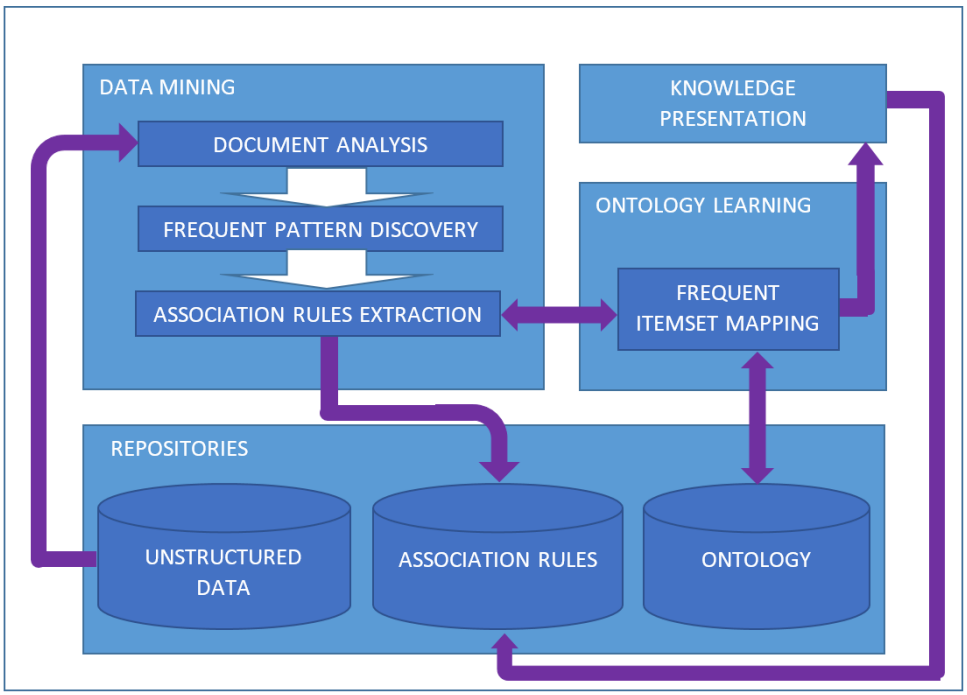


Figure 4.4 - Concept Model

4.2.1 Data Mining

This component is the entry point of the model and is responsible to receive the data input and extract the knowledge. This is achieved through data mining techniques and is adapted from the process presented in Chapter 3.2.1, where it was explained that in order to extract knowledge from data sources, there are three steps which need to be considered *Data Preprocessing* (represented by Document Analysis block in Figure 3.2), *Pattern Discovery* (Frequent Pattern Discovery) and *Pattern Evaluation* (Association Rules Extraction).

The responsibility of this component is related to the knowledge extraction aimed for Ontology Learning of a Domain Ontology. In other words, this component will discover new concepts and associations between them which are not yet known by the actors. As input, this component receives the knowledge sources, as unstructured data, from several documents, files, etc. The output of this component is the knowledge discovered, specifically, the discovered concepts and associations. Each of the steps will be described below.

Document Analysis

The *Document Analysis* block, is the entry point of the model, by which the unstructured data enters. This block is responsible to prepare the data for the next block.

This means, that the data will be preprocessed, by removing irrelevant data without any semantic value. One example is the removing of signalling characters such as ".", or "!" or "?", etc. Moreover, the words could be reduced to its root (e.g. {"Building", "Builder", "build"} have their root as "build") which reduces significantly the data and maintains its semantic value. The output of this block is composed by the words having semantic value for further processing and is delivered to the next block.

Frequent Pattern Discovery

This block, referred as *Frequent Pattern Discovery*, receives as input the pre-processed data from the previous block, to which applies a frequent pattern discovery technique. Based in the comparison study presented in Chapter 3.3.4, the *FP-Growth* technique presents better performance than its competitors, as such, it will be used in this block. The output from this block will be a set of the frequent terms discovered, also referred to as frequent itemsets (FI).

Association Rules Extraction

The *Association Rules Extraction* block is responsible for the extraction of the knowledge, specifically, this block will discover the associations between the concepts. This will be made by discovering association rules between the FI (input) discovered in the previous block, based in evaluation rules. Such evaluation will be made by using objective measures, such as Support, Confidence, Conviction, Lift, Laplace and PS (refer to chapter 3.4.4). The output of this block will be association rules with a positive evaluation, composed by the premise and conclusion concepts and the values for each evaluation measure. Additionally, this output will be saved in a repository and provided to Ontology Learning component.

4.2.2 Repositories

This component relates to all data structures necessary for the operations of the system. Specifically, three data structures are defined: *Unstructured Data*, *Association Rules* and *Ontology*. Unstructured data is the database composed by all documents and files which will be used as input for the Data Mining component. Association Rules is a database responsible to store all the associations discovered from data mining process, as well as the associations that will be chosen by the actors. The Ontology represents the database to hold a Domain Ontology that will be used in this system.

Association Rules Database

This repository is responsible for storing the ontological rules before and after the frequent itemset mapping block. Such rules will be selected by the user and stored under a relational database schema. These stored rules will be the relevant based in the user choice. Table 4.1 below illustrates the structure of a discovered association rule. One line per association rule, and each line will include two columns for *Premise* and *Conclusion* concepts, and additional columns for each of the objective measures used by Association Rules Extraction block. Namely, *Confidence, Conviction, Lift, Total Support, Laplace, Gain and PS*.

Table 4.1 - Representation of Association Rules

#	Premise	Conclusion	Confidence	Conviction	Gain	Laplace	Lift.	Ps	Total Support
1	Concept A	Concept B	Val A	Val B	Val C	Val D	Val E	Val F	Val G

4.2.3 Ontology Learning

Frequent Itemset Mapping

This block is responsible for the connection between the knowledge extracted by the Data Mining component and the knowledge from the domain ontology. Each of the FI appearing on the relevant association rules discovered, will be mapped to the concepts of the domain ontology. The mapping process will be done by calculating the level of similarity between each FI from the rules, and every word (also referred as Ontology Equivalent Term(OET)) representing each concept of the domain ontology. In order to achieve this, it will be used a measure called cosine similarity algorithm.

Cosine similarity algorithm is a measure normally used to calculate the level of similarity between two vectors. In this case, if each frequent itemset or ontology equivalent term can be represented by a vector, the cosine can be used to verify if they are close or far to each other. This measure can be represented by equation (11) which results the correlation between two vectors, vector x and y (Huang, 2008).

$$cos(x, y) = \frac{x}{\|x\|} \cdot \frac{y}{\|y\|} \tag{11}$$

Consequently, equation (11) could be applied to the frequent itemset mapping block, as follows:

$$Sim(FI, OET) = \frac{(FI\ Shared\ Terms) * (OET\ Shared\ Terms)}{(FI\ Total\ Terms) * (OET\ Total\ Terms)} - \delta \qquad (12)$$

Equation (12) represent the level of similarity between FI and OET. The results are presented inside the range [0,1]. Meaning that 0 will represent no similarity between both and 1 represent the maximum level of similarity between them. The terms which appear in FI and OET at the same time are referred as "shared terms". The variable δ was added to deal with itemsets which are similar, although the order of co-occurring shared terms is different. For each equal term in a different order place inside the vector, a value of 0.01 is subtracted to the similarity result value. The variable δ has its values inside the range [0.02, 0.03]. Table 4.2 presents some examples to similarity measure.

Table 4.2 - Cosine Similarity Measure Examples

Example 1	Example 2	Example 3
Similarity ≈ 1	Similarity = 0	0 < Similarity < 1
FI={Wast Manag Facil} OET={Facility Waste Management}	FI={Wast Manag Facil} OET={Complete Chimney System}	FI={Wast} OET={Waste Management Facility}
FI Total Terms = 3 OET Total Terms = 3 Shared Terms = 3 δ = 0,01*3	FI Total Terms=3 OET Total Terms=3 Shared Terms=0 δ = [Not applicable]	FI Total Terms=1 OET Total Terms=3 Shared Terms=1 δ = [Not applicable]
$\boldsymbol{Similarity = \frac{3^2}{3*3} - 0,01*3 = 0,97}$	$\boldsymbol{Similarity = \frac{0^2}{3*3} = 0}$	$\boldsymbol{Similarity = \frac{1^2}{1*3} = 0,33}$

4.2.4 Knowledge Presentation

The present component relates to the way the knowledge is going to be presented to the actors using the system. What to show and how. In this sense there are some elements that need to be presented, namely the association rules discovered, the mapped concepts, the new concepts (if any) and the relevant rules chosen.

Association Rules discovered

It is necessary to show the results from the data mining process. In this sense all the association rules will be presented to the actors in order to be easily chosen. Each will also have an interaction with the possibility to be chosen and sent to the repository.

Mapped concepts

The mapped concepts are the output of the frequent itemset mapping element and are the concepts that are considered similar to the concepts from the domain ontology. In this sense it will be created a scheme in order to differentiate different values of similarity. This scheme will be defined by groups of similarity percentage, each group with a different colour (Table 4.3).

Table 4.3 - Similarity color scheme

Similarity Level (%)	Colour
100	
100-80	
80-60	
60-40	
40-20	
20-5	

New concepts

As already referred above, the new concepts are concepts discovered through the data mining process that do not have any similarity with the concepts from the ontology. That said, whenever a new concept is discovered, it should be proposed to the actor the possibility to add it to the domain ontology.

Relevant Association Rules

The relevant Association Rules are the rules chosen by the actors after evaluation by the domain knowledge expert. The knowledge presentation component will have the responsibility to present all the rules. In this way, the actors can know which rules were already chosen, and which were not.

As a conclusion, in this chapter it was presented a concept model for the present work. It was presented a scenario, defining who will be the actors who will interact, as well as which functionalities and requirements they will have. Moreover, it was described the model dividing it in the components Data Mining, Ontology Learning, Repositories

and Knowledge Presentation. Data Mining will be responsible for all processes related to knowledge extraction, Ontology Learning will be responsible for mapping the knowledge extracted to the one already in the ontology, Repositories will be responsible for all databases and finally, Knowledge presentation will provide the outputs of the model to a user interface.

5

Model Design and Development

This chapter illustrate the design and implementation for the proof of concept (from this point onwards referred to as DOKS, which means **D**ynamic **O**ntology learning with **K**nowledge sources from unstructured data **S**ystem) of the model proposed in the previous chapter. At the end of this chapter the reader will be able to know about the technical architecture of the design, as well as which inputs were used, the tools and technologies adopted, and the implementation of the DOKS system. Lastly, will also know how to use User Interface.

5.1 Design

5.1.1 Technical Architecture

The model presented in the previous chapter (Figure 4.4) will be implemented based on an architectural pattern referred to as MVC (Model View Controller), which is used in software engineering design. This architecture clearly abstracts the system in a 3-tier model, allowing for a clear separation of concerns. (Figure 5.1)

The first layer, the *controller* which is the cerebrum of the system, includes all logic development for data mining and ontology learning. Moreover, the controller is responsible for the connection between the other two layers. Secondly, the *model* layer is responsible for all data related content, holding the repositories necessary for the system. Lastly, the *view* layer is related to the interface of the system to the actors. Figure 5.1 below illustrates the MVC architecture applied to the present model.

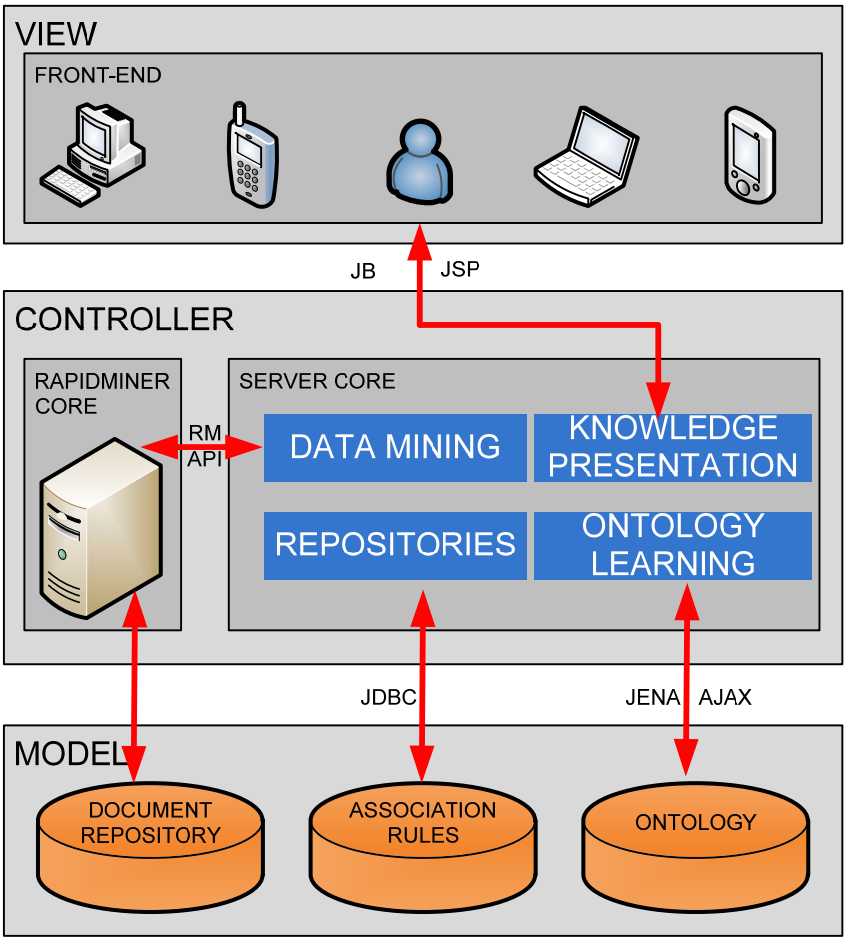


Figure 5.1 - System Architecture

As referred above, in the *Model* layer, the proposed solution will include three repositories for the storage of all the data for the system. The first one, Document Repository, is the initial repository which stores all the initial unstructured data input to be analysed. The second, namely Association Rules, is the database to store the association rules related content. The entity relation diagram which models this database is illustrated in Figure 5.2. This database is composed by four tables, namely *concepts* (responsible for all chosen concepts), *rules* (includes all association rules chosen by the actors), *rules_stemmed* (includes all association rules received after data mining process) and *stemmed_words* (stores all unique frequent items which are included in each association rule). Lastly, the third one referred to as Ontology, is the database to store for the ontology itself, with all concepts and relations from the domain.

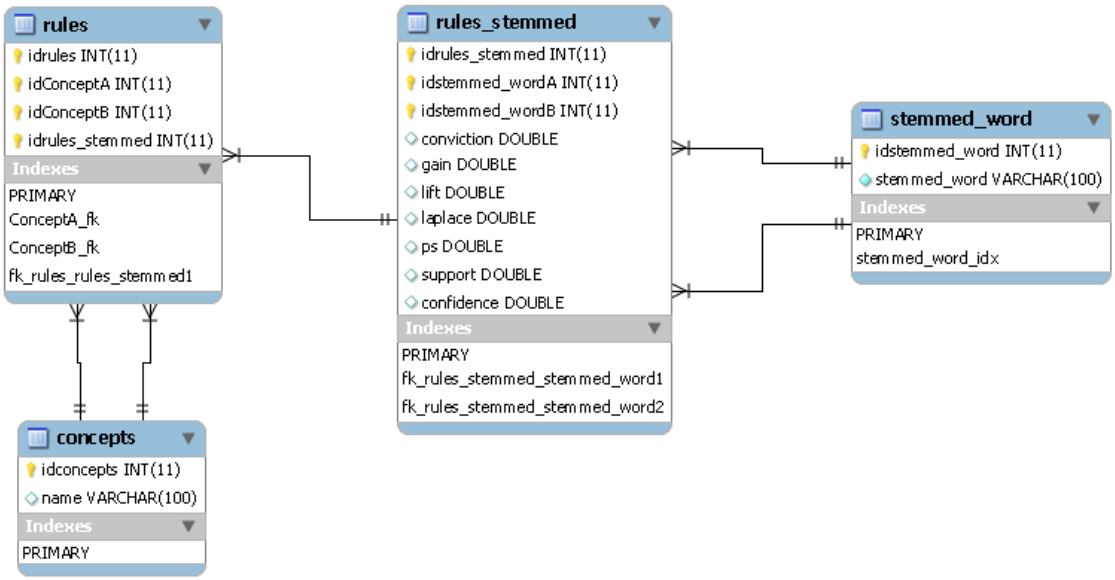


Figure 5.2 – Entity Relation Model

The *View* layer is responsible for the user interface. It is the system connection to the world, where all the actors access the functionalities. This layer will include all the front-end development related to the knowledge presentation component and to the several possible interfaces to use (such as a desktop or laptop pc, mobile, etc).

Controller layer is the place of the whole system where all the logic, analysis and process is developed. Specifically it will be divided in two cores, Rapidminer and Server core. The Rapidminer Core, is responsible to make all the data mining process from the unstructured data until the association rules discovery. The Server Core will have all the logic of the system which will include 4 blocks. Namely, Data Mining, Knowledge Presentation, Repositories and Ontology Learning. Each block responsibility reflects the model proposed in the previous chapter.

5.2 Implementation

This chapter describes all steps of the implementation of DOKS. It will explain what were the technologies and tools used, as well as the specificities of the implementation through a static and a behavioural view of the software.

5.2.1 Input

Unstructured data

The input used to assess the proof of concept was a set of 20 scientific articles published online in ICONDA®Bibliographic library (Fraunhofer, 1986), related to B&C sector,

used as unstructured initial data. Each of the articles provided an average of ca. 3600 terms, making a total of more than 70000 words, which was considered an interesting challenge for the proposed solution in the present work. Although they do not represent entirely the B&C sector, the data used pretended to be a small sample, which could provide a reflect of the sector. Table 5.1 presents the documents used, with a relation of the words for each of them.

Table 5.1 - Data used as input

Doc	Document title	N. Words
1	Evaluation of Deterioration behaviour of Surface Coating for RC Buildings by Permeation-Diffusion	2291
2	A Study on the Carbonation Progress of Concrete Concerning the Influence of Deterioration of the Coating Material for Textured Finish	2869
3	The Experimental Evaluation of Parameters Contributing to the Durability of Coating Materials for Colouring and Protecting External Plastered Surfaces	3744
4	Durability Evaluation of Highly Reflective Coating Materials for Roofing	2819
5	Waste Today Gone Tomorrow Sustainable Waste Management: Malta, a Case Study	6706
6	Waste Management Strategies during Post Disaster Phase: A Case of Sri Lanka	4647
7	Sustainable Construction Waste Management in Malaysia: A Contractor's Perspective	4714
8	Planning for SMEs' Proactive Waste Management in Office Building Retrofit Projects	4075
9	Integration of sustainability solutions in sanitary installations: the example of the AveiroDOMUS "House of the Future"	1801
10	Study of sanitary equipment installed on light-weight partitions	2564
11	Provision scales of sanitary accommodation in public toilets	3085
12	Present state and future challenge on installation number of the sanitary fixture	3790
13	Lighting in New Zealand Homes – Lighting Efficiency as a Sustainability Indicator	3031
14	A Systematic Review on the Therapeutic Lighting Design for the Elderly	5656
15	ICT for Energy Efficiency: Towards Smart Buildings, Manufacturing, Lighting and Grids	4279
16	Light Trespass from Exterior Lighting in Urban Residential Areas of Compact Cities	3491
17	Sustainable Office Building: Should I Focus on HVAC-system Design or Building Envelope	2057
18	Earth, Wind and Fire Towards New Concepts for Climate Control in Buildings	3824

19	Hvac Integrated Control For Energy Saving and Comfort Enhancement	3553
20	Rfid-Based Occupancy Detection Solution for Optimizing HVAC Energy Consumption	3316
Total		72312

Ontology

The ontology used was developed and provided by SEKS project (Figueiras, 2012) and includes Building and Construction sector knowledge. This ontology was specifically built to enable test and validation, as such includes a limited quantity of ontological concept families and properties from B&C sector. Specifically, such limited quantity is shown in Figure 5.3 which describes the main level concepts of the B&C domain ontology.

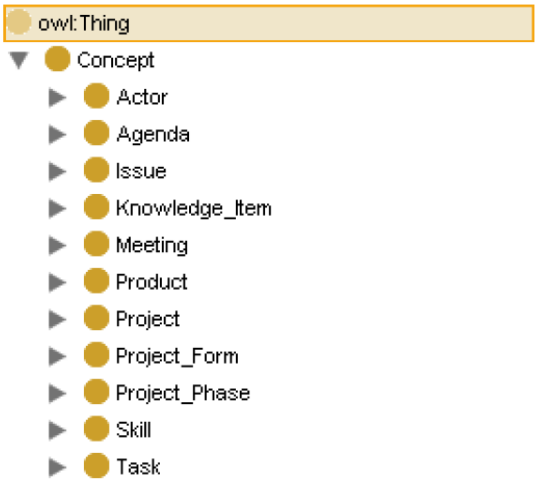


Figure 5.3 - Main level concepts of B&C domain ontology

5.2.2 Tools and Technologies

The following tools and technologies were used to develop the proof of concept of the model presented in the previous chapter (Figure 5.4). One of factors to consider when choosing the right technologies to develop the proof of concept from this dissertation, is the fact that the technologies should be free to use in academic environment.

Visual Paradigm for UML, is a visual design and modelling software tool, which was used to develop all UML diagrams. Specifically, this software was used to make the Class Diagrams, Use Case Diagrams and Sequence Diagrams. It was used the student version, which is free to use in academic projects.

Figure 5.4 - Tools and Technologies adopted

The database was modelled in SQL technology and implemented through the use of *MySQL Workbench* software tool. This tool which is normally used for database modelling, allowed for the creation of the Entity Relation Diagram (ERD), which was later translated into SQL statements necessary to build the database. The database runs on top of an Apache server provided by *XAMPP* software package. This package also provides an admin web portal aimed for database maintenance, specifically PHPMyAdmin.

Rapidminer was the tool used for the Data Mining process. All the workflows for the pre-processing of the unstructured data, FP-growth and Association Rules algorithm were developed in this tool. Rapidminer is normally used in scientific areas such as machine learning, data mining or text mining, just to name a few. In order to provide the output adapted to DOKS, a script based in Groovy[13] technology was implemented through a Rapidminer module. Groovy language is Java based, as a result integrated well

[13] Groovy, a technology for JVM, builds upon the strengths of Java but has additional power features inspired by languages like Python, Ruby and Smalltalk. (Guillaume Laforge, 2012)

with all used Java classes and libraries from Rapidminer. Additionally, the Java based API of Rapidminer allowed for the integration with DOKS.

The development of the proof of concept was based in Java technology through the NetBeans SE IDE. This IDE was used to develop all the code of this project. NetBeans IDE is a freely available open tool which is very complete, it allows the use of Java language, compiling of the code directly to JVM, and also includes a debugger to verify all the possible errors related to the code or to the system execution. All the necessary processes and methods for the Web Front-End were developed using JavaBeans technology through interfaces. The integration with the repository that stores the association rules was developed through JDBC connections.

Protegè was used to work with the ontology. This tool provides a visual graphical interface for ontology interaction. It allowed to open, visualize and add new or update concepts and relations manually. In the code of server side, this functionality was developed with the help of a Java API, Jena Semantic Framework Ontology that supports OWL language (Chapter 2.2.3), in which the ontology was created.

In the DOKS Web Front-End developed the technologies used were HTML and CSS to create all pages supporting the visual presentation of the results to the users, and on the server side the interface was created in JSP. The user interface was tested and can be used in all modern browsers in the market, like Google Chrome, Mozilla Firefox or Internet Explorer.

5.2.3 Data Mining Process

The workflow for the Data Mining process was implemented in Rapidminer (Figure 5.5). The next lines will explain the implementation of this workflow.

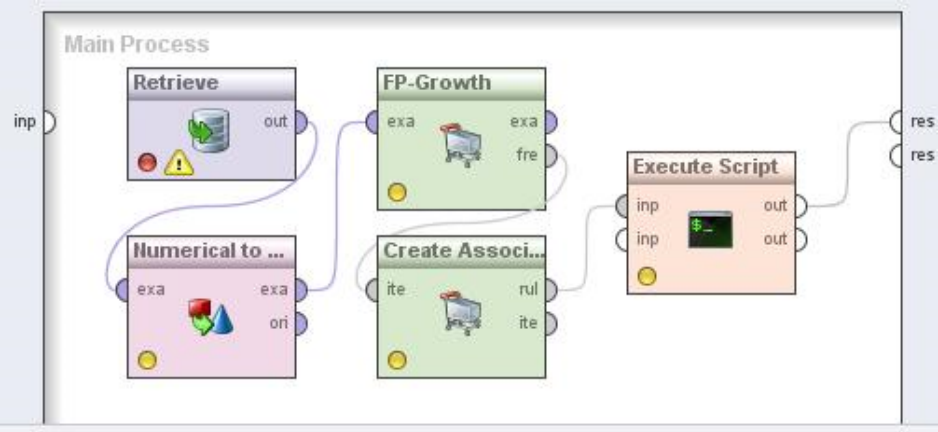


Figure 5.5 – Rapidminer Main Process

Document Analysis

The first step in data mining process is to preprocess the input. The sub-processes used were Tokenization, Transform Cases, Filter Stopwords, Stem (Snowball) and Filter Tokens. (refer to Figure 5.6). This set of procedures is the Document Analysis Block. In fact, is composed by 6 blocks following a specific order.

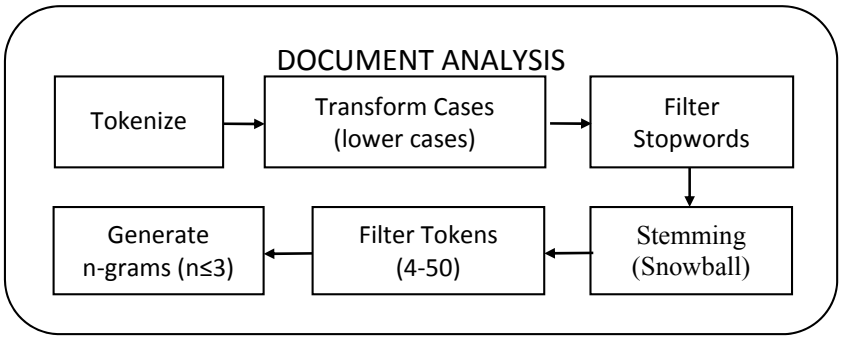


Figure 5.6 - Document Analysis Pipeline Block

The first sub-process is the *Tokenize* which is responsible for the separation of the full text into a sequence of tokens. Tokens can be understood, for the purpose of the present work, as a set of letters. As a result, everything that does not include letters is discarded from this process (e.g. punctuation, numbers and other characters).

The second sub-process is *Transform Case*. Its goal is to transform all tokens to lower case. This is a necessary step so that all tokens that differ in a letter case would be considered the same. For instance, the tokens *Token* and *token* are considered the same, as they have exactly the same letters in the same order, having just a capital letter different, consequently the one with the capital letter will be case lowered to *token*.

Next sub-process is *Filter Stopwords.* This is responsible for the removal of stop words, such as *the*, *each*, *a*, etc. All these stopwords are removed from the set of tokens.

Stem (Snowball), is the next sub-process and corresponds to the execution of the stemming algorithm. This algorithm has the responsibility to transform the word in its stem, in other words, it will remove all the affixes and prefixes from the word. In this project the stemming algorithm used is the Snowball variation algorithm. This process can be optional, however its use can be of great value, as it reduces the words to its stem, gathering them into the same word family. Meaning that as more words are grouped by their stems, more representative is the word in the document. Another good advantage of the stemming process is the reduction of the data size augmenting each stem precision.

After all words are in their stem form, they enter in the *Filter Tokens* process. This process prunes all tokens (words) that are lower than 4 and higher than 50 characters. This process is necessary to remove all unnecessary tokens that have no semantic relevance for the study, like chains of random letters.

The last step of the document analysis is the *Generation of n-grams*. The n-grams generation is the creation of sequences from 1 to N words. It was considered N=3 as representative of the whole. (eg. Waste Management – 2-gram or bigram and Electric Power Product – 3-gram or trigram). The purpose for this generation is to discover concepts and groups of words representing concepts.

Table 5.2 – Numerical to Binomial regulation

NumBinMax	Support	Confidence	Association Rules
0,012	0,25	0,01	18
0,012	0,25	0,60	18
0,012	0,25	0,70	18
0,013	0,20	0,65	102
0,013	0,25	0,70	12
0,014	0,20	0,01	92
0,015	0,20	0,01	92

The output of document analysis was provided to a sub-auxiliary process, referred to as *Numerical to Binomial*, whose function is to change the nominal values of the received output into binomial values (true or false). In other words, it changes every value inside a defined interval to false and to true the ones outside. Which means that the words with no significant semantic value are filtered out of the document corpus. For the purpose of this work, the interval values were choose as follows: Min – 0.0 Max 0.013. As Table 5.2 shows, tests were made to fine tune the configuration of this sub-process to reflect the wider number of Association Rules to examine.

FP-Growth

This block is responsible to discover frequent patterns from the preprocessed data using the FP Growth algorithm. The *minSup* (minimum frequency or support) value, for each

FI was 20% (0.20). All the FI with support value below were not considered frequent and as a result, pruned from the word set.

Association Rules

This is the block responsible to receive the FI from the FP-Growth block and discover all AR. The output are the AR discovered, which includes two FI, premise and conlusion, as well as the measures used to classify each of them. Namely, *Confidence*, *Conviction*, *Gain*, *Laplace*, *Lift*, *Ps* and *Support*. After several tests with minConf values, the value of 65% (0.65) was considered the best for the goals of the present dissertation, as it provided the biggest set of association rules (refer to Table 5.2).

5.2.4 Ontology Learning

Frequent Itemset Mapping

As already referred, Frequent Itemset Mapping block represents a process executed after the Data Mining process, applied to the FI discovered by Association Rules block. It is worth referring that each FI is stemmed and each OET is not, meaning that a comparison of both will always result different. The solution to this was to compare the FI with the first characters of the OET. For instance, if the FI stem is *manag*, this will be compared to all OET which start by this stem (e.g., *Management, Manager*).

Table 5.3 gives an example of the results of a comparison between a FI and an OET, where it can be observed a list with all the concepts that have OET starting with the FI *manag*. It should be noted that this process finds the exact matches as well as candidate concepts.

Table 5.3 – Concept mapping for FI *manag*

Concept mapping for manag	Similarity (%)
Management Actor	**100**
Trainer	**100**
Manual	**50.00**
Waste Management Product	**50.00**
Chief Executive	**50.00**
President	**50.00**

Facility Manager	**50.00**
Managing Personnel Resources	**33.33**
Middle Management Actor	**33.33**
Executive Management Actor	**33.33**
Information Management Facility	**33.33**
Energy Management Facility	**33.33**
Managing Material Resources	**33.33**
Report	**33.33**
Team Assembly Phase	**33.33**
Managing Financial Resources	**33.33**
Resource Management Skill	**33.33**
Managing Time	**33.33**
Waste Management Facility	**33.33**

Moreover, when the results of the mapping process is empty, it means that the FI represents a new concept. In that situation, the user has the opportunity to learn the ontology by adding a new concept.

5.2.5 DOKS Architecture Implementation

In this section is explained in detail the implementation of the architecture design of DOKS (refer to section 5.1.1). DOKS was implemented as a Java Web application built to run on Apache Tomcat 7 server. The system was developed using Eclipse IDE. Figure 5.7 presents the Java packages and class structure implemented.

SEKS package includes all basic routines to access the ontology, taking advantage of Jena Framework API. The ontology was mapped into a MySQL database, however it was stored in an .owl file. The jenaConfig.xml provided the configuration for access to the database.

The view layer is represented by the UI, which was implemented as a Web Portal, using Java Server Pages.

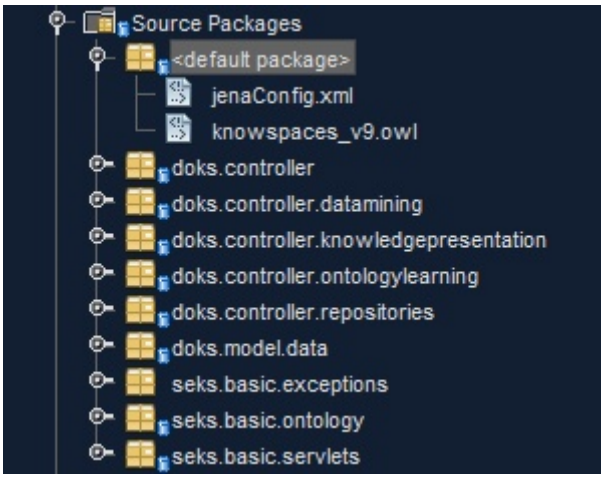


Figure 5.7 - DOKS Architecture Class Structure

Static view

DOKS packages reflects the Model and Controller abstraction layers. These packages were designed in UML Class Diagrams (UCD). UCDs reflect the static view, describing the classes to develop in a software system. To view the diagrams, please refer to Appendix C at the end of this document.

The Controller layer includes the following packages:

- doks.controller – This package includes the base classes that connects all the packages in the controller.
- doks.controller.datamining – This package relates to the classes and objects used to obtain the results from the data mining process, which includes methods to execute rapidminer workflows.
- doks.controller.knowledgepresentation – This relates to the classes that deal with the interaction to the UI and the way the knowledge will be presented.
- doks.controller.ontologylearning – This includes the classes for the ontology learning block, specifically, frequent itemset mapping related methods.
- doks.controller.repositories – Repositories package includes the classes for the association rules database manipulation.

The view layer includes the following for the Web UI package:

- index – web home page of the UI.
- ARResultsConcepts – web page to show association rules results
- ARResults – web page to show data mining output

- ARToDatabaseResults – web page to execute data mining process in a new set of data.
- rulessaved – web page to show the association rules already chosen
- ConceptTree – small window to insert new concept in the domain ontology.

The Model layer included:

- seks.basic packages – relates to classes used for interaction of the domain ontology. Provided by SEKS project (Figueiras, 2012).
- The database classes to support the data models.

Dynamic view

The dynamic view presents the interactions between the entities of the system, providing the responses for the actions made by the users. Such view will be expressed by UML Sequence Diagrams (USD). Each USD will reflect a particular functionality presented in chapter 4.1.2, explaining the interactions between DOKS components, the UI and the actors. Please refer to Appendix A for the diagrams.

5.3 User interface

This section presents the interface available to show the results of this dissertation. The users can learn what the capabilities of the interface are. What can they do with it, what will they visualize and the way to use it. It will be described with the help of screenshots representing each of the pages or a specific functionality from the User Interface (UI). The UI presented is the version aimed for web browsers, such as Google Chrome or Mozilla Firefox.

5.3.1 Home Page

Figure 5.8 – UI –AR Home Page

The first screen to load is the Home Page (Figure 5.8) which includes a menu with three options, namely:

- *Discover Association Rules (No Concepts)*;
- *Discover Association Rules* and *Analyse in RM and Renew DB;*
- *Analyse files in RM and renew DB.*

The first option, "*discover association rules (no concepts)"* is, as the name describes, to discover all association rules in the repository. This is the option to proceed with only the data mining process, or in other words, the association rules are discovered and the results are presented to the user without the execution of the *frequent itemset mapping* block from Ontology Learning component.

The second option, *discover association rules,* provides the results of the data mining process and the results of the frequent itemset mapping. This option includes the AR discovery and the mapping of the frequent itemsets with the ontology equivalent terms

Finally, the third option, *Analyse files in RM and renew DB* is the operation responsible execute the data mining process with a new set of unstructured data. The objective here is to save the results of a new set in the AR database.

5.3.2 Association Rules Page

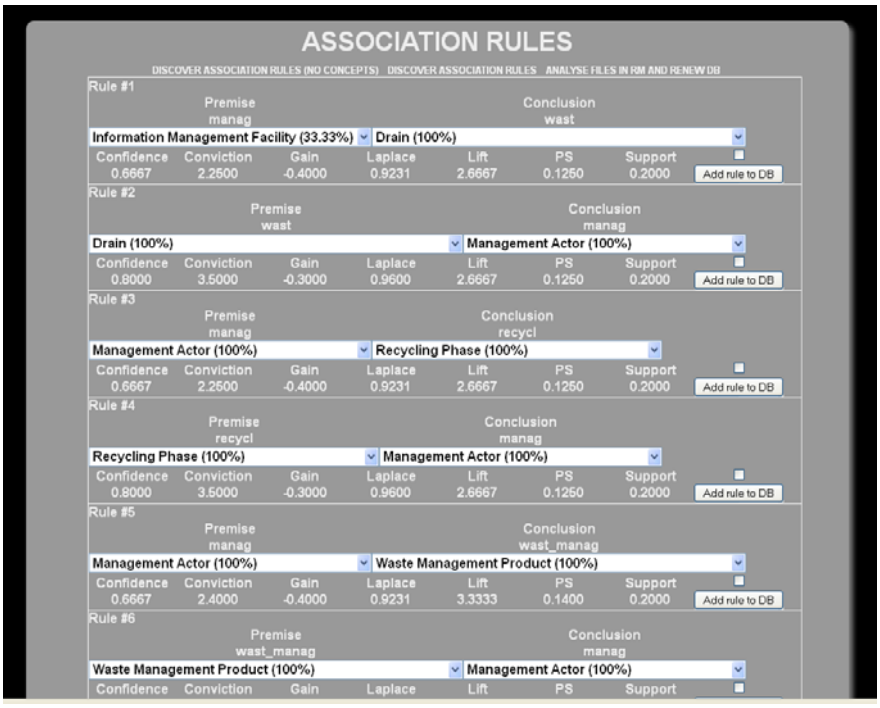


Figure 5.9 – UI - Association Rules result page

When the system finishes the DM and FIM processes, its results are presented to the user, as show in Figure 5.9. This screenshot shows a page with the results received from the server. The next sub-section describes how this is organized.

Association Rule Box

All ARs discovered are presented in boxes. Each box describes one AR, and includes a premise, a conclusion and a set of measures. Both the premise and the conclusion are represented by a FI and is available a dropdown list with the similar concepts mapped from the ontology for the user be able to choose the concept which best suits the AR. Additionally, a set of measures is also available for the user evaluate the AR. This includes Confidence, Conviction, Gain, Laplace, Lift, PS and Total Support values. Finally, a checkbox to select the rule if the intention is to save it in the AR database and a button to execute the save operation (namely "Add rule to DB") are also included in the rule box.

Mapped concepts

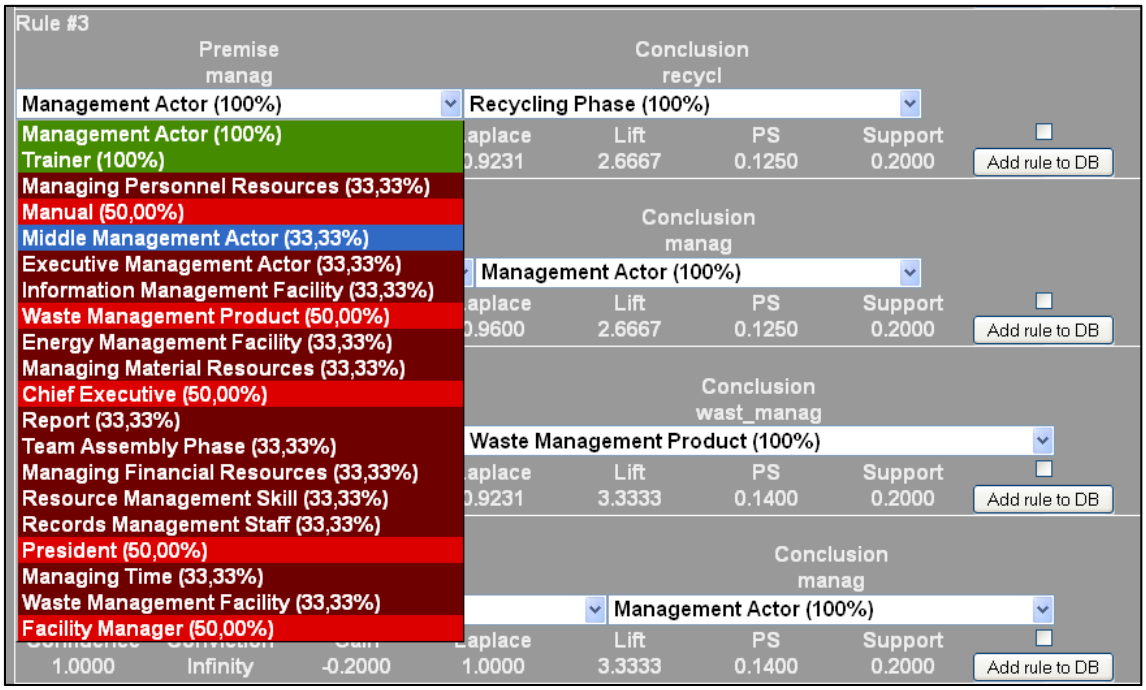


Figure 5.10 – UI – Exact and Candidates concepts dropdown list

In order to choose a concept there are some considerations worth be noted. After each FI has been mapped into a list of candidate concepts, each concept is presented with two additional information items relevant to support its choice. The first item is the similarity value. This value represents how similar is a concept to the FI which is

presented in the form of percentage. (refer to chapter 4.2.3). The second item worth notice is the similarity colour scheme (refer to chapter 4.2.4). Green is for exact matches and as higher the value of similarity becomes, lighter is the red colour. These two items provide a visual and intuitive aid for the best possible choice of the concept. Figure 5.10 provides an example of this items described, where it can be seen a list of concepts, each with its similarity value and colour. For instance, the concept Facility Manager, presented in a light red, is 50% similar to *manag*.

New Concepts

The cases where the FI does not have any kind of similar concept in the ontology, means that the FI is new, and the user can add it to the ontology as a new concept. The UI provides a link "*New...*" next to the dropdown list (Figure 5.11), which is empty, in order to add a new concept to the ontology. After pressing the link, a box pops up allowing the user to choose where in the ontology structure, the new concept should be added, as well as what will be its name. The new concept will be added to the ontology as an Individual with the respective FI as its first keyword.

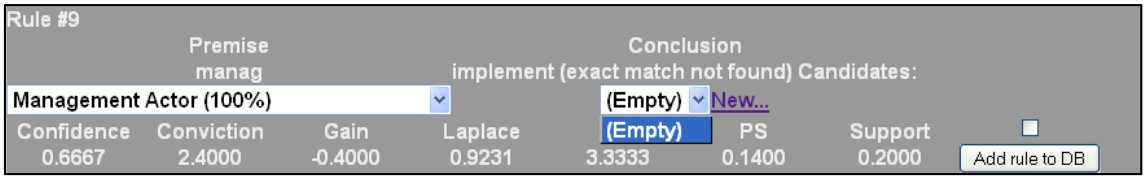


Figure 5.11 – UI – AR case with a new concept discovered

5.3.3 Chosen Association Rules visualization page

In order to see which association rules were already chosen by the users, it was created a visualization page based in the format of the Association Rules database (refer to chapter 4.2.2). This page serves two purposes. One, is the visualization of the knowledge already stored in the database and two is a confirmation for the user to know that the chosen rule was stored in the AR DB. This page can also be for the ontology expert to know which associations will be used to learn the ontology. A screenshot of this situation is presented in Figure 5.12, where an example can be observed. The first relevant information to appear is a confirmation line providing the concepts of the chosen association rule, which in this case it was chosen the association rule between concepts *Temperature Measuring Instrument* and *Complete Heating System*. After this appears a

list with all the ARs already in the DB, which also includes the recently added highlighted in red.

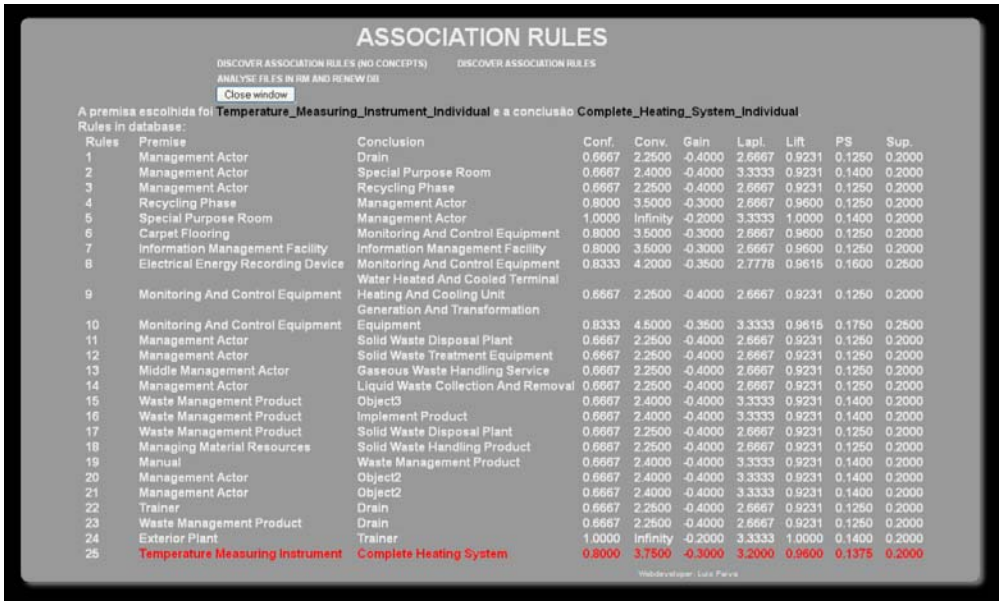


Figure 5.12 – UI – Chosen Association Rules view page

6

6 Assessment

This chapter presents an assessment over the method proposed in this document. It will be evaluated with some validation scenarios with some discussions on the respective domains, in fact, if the results were as expected. In order to achieve the latter, the method proposed was assessed using DOKS system interface (Chapter 5.3). This assessment tests the applicability of the method proposed in the following scenarios:

1. Knowledge Discovery
    a) Semantic relations;
    b) New concepts;
2. Proposal of semantic relations for enrichment of a domain ontology

As stated in chapter 5.2.1, the present project received nearly 20 document files to evaluate, which originated from sub-domains of Building and Construction Sector, specifically from *Climate Control*, *Lightning*, *Coating*, *Sanitary* and *Waste Management*. The initial unstructured data included ca. 70000 words to be assessed. These were the initial data to be applied in this project and used in all scenarios. Additionally, an ontology was used as input, as referred in previous chapter, with contents already included, in order to also assess the mapping process.

6.1 Knowledge Discovery

6.1.1 Semantic Relations

The user starts by accessing the Home Page of the interface and by pressing "DISCOVER ASSOCIATION RULES (NO CONCEPTS)" will make a request to the server, in order to start the DM process. The server initiates the process of association rules discovery in the background, through Rapidminer API. These API is responsible for the Data Mining process of the method, specifically, Document Analysis, Frequent Pattern Discovery and Association Rules Extraction blocks. It is worth remembering that this process was fine tuned to output the wider results amount possible (refer to Table 5.2 from previous chapter). If everything runs smoothly, this process should provide an output.

Table 6.1 - All unique FI with one term discovered

Unique FI			
manag	Wast	recycl	wast_manag
plan	implement	energi	consumpt
temperatur	Indoor	Heat	energi_consumpt
electr	Power	Oper	hvac
cool	Toilet	sanitari	climat
offic	offic_build	coat_materi	coat

The output from this process was a total of 102 AR discovered (refer to Appendix B in order to see a table with all AR discovered), which included 24 unique FI as premise and conclusion. These FI can be observed in Table 6.1. It is worth noting, that the AR only considered FI in the premise and conclusion with one term. If the amount of terms in the FI could be raised to 2 or more, the AR amount would probably be higher and more knowledge would be discovered. Conversely, the case could be different, if the amount of terms in the FI were too high, the number of rules discovered could be huge, which would be harder for the users to evaluate them. The quantity of ARs discovered was considered by the author of the present document as a good value for the administrator could proceed several good ontology enrichment.

The output for the DM process, specifically the unique FI from Table 6.1, is then used as input for the ontology learning process, specifically, for the FIM block, which is also automatically executed in the server. Moreover, a domain ontology with knowledge from B&C is also used as input to provide the concepts to map. This process retrieves a list of possible concepts, which are recognized as being similar to the FI. As an example,

Table 6.6 below shows the list of mapped concepts for FI *Plan*. In this case, there are 7 concepts which are 100% similar to the FI. When all concepts are mapped, these will appear in the AR results page in the form of a dropdown list. Afterwards, the user needs to choose manually the ontology mapped concepts which are related, and will appear in the AR results page in the form of dropdown list. For instance, for rule #3 manag-recycl, the user can choose as concepts respectively Trainer-Recycling Phase.

6.1.2 New concepts

In order to discover new concepts the process is much similar as the one to discover semantic relations. A new concept is a concept which is not yet in the domain ontology, so, in order to discover new concepts the user needs to discover the semantic relations from previous sub-section and execute the FIM process. After this, just needs to look at the AR results page, and see if any mapping was empty. Which is represented by the word "Empty" in the dropdown list. From the 102 AR, only one new concept was discovered, *implement*. Which could mean that the ontology is updated or the input data used was predictable. More tests would need to be done to assess this issue. The user, then needs to manually insert this new concept in the ontology. To achieve this, press "New" which will open a new window with the ontology tree structure. Write the name of the concept, for instance, Implementation and associate the FI to this concept, and press "ok". The new concept is now in the ontology.

6.1.3 Discussion

Some interesting knowledge can be observed in the AR result set, for instance, the FI *Manag* appears in 5 rules as a premise, which originates 5 different conclusions (Table 6.2). Additionally, it can be observed that it relates less to *Wast* and *Recycl*, than to the other three. Moreover, when these rules conclude to *Manag*, the total sum of the rules is also 5 (refer to Table 6.3). Although *Manag* is present in the same number of sets in premise and conclusion, it can be argued that the rules are not directional. Sub-section 3.4.4 demonstrates that the conviction value can be different from each direction, justifying the former situation. In fact, the conviction values for the each premise comparing to its conclusion, are not the same. In other words, it can be argued that the semantic strength of the relation between *Manag* and *Wast*, for instance, depends on the direction of the rule.

Table 6.2 – AR with FI Manag as premise

#	Premise	Conclusion	Confidence	Conviction	Gain	Laplace	Lift.	Ps	Total Support
1	Manag	Wast	0.667	2.25	-0.4	0.9231	2.667	0.125	0.20
2	Manag	Recycl	0.667	2.25	-0.4	0.9231	2.667	0.125	0.20
3	Manag	Wast_Manag	0.667	2.40	-0.4	0.9231	3.333	0.140	0.20
4	Manag	Plan	0.667	2.40	-0.4	0.9231	3.333	0.140	0.20
5	Manag	Implement	0.667	2.40	-0.4	0.9231	3.333	0.140	0.20

Even though the rules with different conviction values from the same FI as premise and conclusion, could present some interesting knowledge, there are cases where the rules are bidirectional. In other words, both FI are related to each other in the same sense. Consequently, this could arguably mean that these concepts belong to the same knowledge domain. As a result, these could provide good indicators for the improvement of their relations in the domain ontology or even to create new ones. Although the support value could be in the lower bound of minsup, if the Confidence value of an association rule is 100% in both directions, this is a good indicator of a strong relation. Refer to Table 6.4 for examples of this knowledge discovered from the DM process.

Table 6.3 – AR with FI *Mang* as conclusion

#	Premise	Conclusion	Confidence	Conviction	Gain	Laplace	Lift.	Ps	Total Support
1	Wast	Manag	0.80	3.5000	-0.30	0.96	2.67	0.125	0.20
2	Recycl	Manag	0.80	3.5000	-0.30	0.96	2.67	0.125	0.20
3	Wast_Manag	Manag	1.00	∞	-0.20	1.00	3.33	0.140	0.20
4	Plan	Manag	1.00	∞	-0.20	1.00	3.33	0.140	0.20
5	Implement	Manag	1.00	∞	-0.20	1.00	3.33	0.140	0.20

Nevertheless, the rules considered unidirectional, or in other words, the rules that do not have the same evaluation results in both direction (refer Table 6.5 for examples of this situation) could also present unexpected and relevant knowledge. For instance, whenever *Plan* occurs, *Wast* also occurs with 100% confidence. In contrast, whenever *Wast* occurs, *Plan* occurs with 80% confidence, which also indicates a relation between both.

Table 6.4 - AR - bidirectional rules

#	Premise	Conclusion	Confidence	Conviction	Gain	Laplace	Lift.	Ps	Total Support
1	Recycl	Wast	1.00	∞	-0.25	1.00	4.00	0.187	0.25
2	Wast	Recycl	1.00	∞	-0.25	1.00	4.00	0.187	0.25
3	Hvac	Cool	1.00	∞	-0.20	1.00	5.00	0.160	0.20
4	Cool	Hvac	1.00	∞	-0.20	1.00	5.00	0.160	0.20
5	Wast_Manag	Plan	1.00	∞	-0.20	1.00	5.00	0.160	0.20
6	Plan	Wast_Manag	1.00	∞	-0.20	1.00	5.00	0.160	0.20

The present work presents several association rules measures which could be used to help an ontology expert learn an ontology. For instance, considering confidence, when this value is 100%, there is a total trust of the semantic relation between both keywords. In such case, the conviction value equals to ∞. Even if the support values are as low as 20%. For example, this case can be observed in rule #4 of Table 6.5.

Table 6.5 - AR - unidirectional rules examples

#	Premise	Conclusion	Confidence	Conviction	Gain	Laplace	Lift.	Ps	Total Support
1	Plan	Wast	1.00	∞	-0.20	1.00	4.00	0.150	0.20
2	Wast	Plan	0.80	4.00	-0.30	0.96	4.00	0.150	0.20
3	Toilet	Sanitari	0.80	4.00	-0.30	0.96	4.00	0.150	0.20
4	Sanitari	Toilet	1.00	∞	-0.20	1.00	4.00	0.150	0.20
5	Electr	Power	0.80	4.00	-0.30	0.96	4.00	0.150	0.20
6	Power	Electr	1.00	∞	-0.20	1.00	4.00	0.150	0.20

Furthermore, the mapping of the FI and ontology concepts can also provide great and valuable knowledge for its enrichment. If the confidence of an association rule is lower than 100%, such scenario still creates doubts. However, when the mapping process is introduced here, the correspondence of the FI with the ontology keyword, might help to dissipate or at least reduce the doubts. Furthermore, as higher the similarity value, higher is the certainty of the presence of that concept in a rule, in other words, the semantic distance of the two concepts evaluated in a rule is higher, and the uncertainty is lower, with the presence of this mapping.

Table 6.6 - Example of Mapping results of the FI Plan

Concept	Equivalent term mapped	Sim (%)
Agenda	plan	100
Architect	planner	100
Exterior Plant	plant	100
Planner	planner	100
Planting	planting	100
Planting Accessory	planter	100
Special Purpose Room	planetarium	100
Administrative Agenda	administrative plan	50
Board	planning board	50
Commitee	planning commission	50
Concept Phase	project planning	50
Cooling And Freeze Plant	freeze plant	50
Drawing	design plan	50

Engineer	building plan	50
Financial Agenda	financial plan	50
Intermediate Meeting	planning meting	50
Liquids Treatment Plant	chlorination plant	50
Non-Clothes Storage Furniture	plan chest	50
Planetarium Equipment And Furnishing	planetarium projector	50
Planning Actor	planning responsible	50
Plant And Planting Product	plant product	50
Plant And Service Control	plant control	50
Plant Display Furniture	decorative planter	50
Planting Product	planting product	50
Preliminary Design Phase	planning phase	50
Solid Waste Disposal Plant	crusher plant	50
Technical Agenda	technical plan	50
Garden And Park Furniture	tub for plants	33,33
Industrial Plant Performance Control	centralized plant control	33,33
Plant And Control Room Unit	plant office shelter	33,33
Plant And Service Test Equipment	plant test equipment	33,33
Report	preproject planning report	33,33
Waste Water Treatment Plant	sewage pumping plant	33,33
Liquid Waste Treatment	packaged sewage treatment plant	25
Roof Specialty And Accessory	vegetated roof planting module	25
Water Management Facility	residual water treatment plant	25

Nonetheless, if the similarity value is lower, this could be also provide unexpected knowledge, which could be interesting (refer to chapter 3.4.2). Table 6.6 shows an example for the mapping of the FI *Plan*, which provides also some unexpectedness in their results. Examples of this are the concepts *Liquid Waste Treatment* or *Roof Specialty and Acessory*. More tests should be made in order to increase the trust of these results.

6.1.4 Proposal of semantic relations for improvement of a domain ontology

In order to assess this goal, a user needs first to execute the "DISCOVER NEW CONCEPTS" at the Home Page of the UI to discover rules and map all concepts. After choosing the similar concepts for each of premise and conclusion of an AR, press "Add Rule to DB" button. This will save the rule in the database. The FrontEnd opens a page with the chosen rule and the ones already chosen before, if any. These rules are proposals of semantic relations, which will be evaluated by the domain expert in order to improve

the relations of the respective concepts in the database. As the selection of rules is a manual process, Table 6.7 shows a list of examples of proposals of semantic relations, in order to update their relations in the domain ontology.

Table 6.7 - Examples of proposals of semantic relations for domain ontology improvement

#	Premise	Conclusion	Confidence	Conviction	Gain	Laplace	Lift.	Ps	Total Support
1	Management Actor	Drain	0.6667	2.2500	-0.40	2.6667	0.9231	0.1250	0.20
2	Recycling Phase	Management Actor	0.8000	3.5000	-0.30	2.6667	0.9600	0.1250	0.20
3	Carpet Flooring	Monitoring and Control Equipment	0.8000	3.5000	-0.30	2.6667	0.9600	0.1250	0.20
4	Trainer	Drain	0.6667	2.2500	-0.40	2.6667	0.9231	0.1250	0.20
5	Exterior Plant	Trainer	1.0000	Infinity	-0.20	3.3333	1.0000	0.1400	0.20

As a conclusion, this chapter provided an assessment translated by three scenarios and the respective results provided. These scenarios present a proof of the validity of the proposed method. From the method, one can discover semantic relations, new concepts, and use these relations to improve a domain ontology. The results produced include, not only but also interesting and unexpected knowledge, which could be an added value to assist in the ontology learning, providing good trust indicators to improve the knowledge inside. It is worth adding that the inputs (e.g. documents and ontology) used for assessment are from B&C, but this method can use inputs from any other domain.

7

Conclusion and Future directions

Through the following lines will be presented an overview of the work developed in this thesis. The objectives defined in Chapter 1 intended to guide the path of the study. For these objectives it will be described which ones were achieved and which ones were not, describing also the problems and difficulties found during the development and research, and also, how were these difficulties solved. Similarly, this will also bring to attention some possible future research topics, where achievements addressed by this work can constitute a solid basis.

7.1 Work overview

As presented through this dissertation, it was described the importance of concept representation in contrast to word representation, in the Semantic Web area. It was highlighted the advantage of the use of mechanisms (e.g. Controlled Vocabularies), as these provides means for semantic representation, which allows more than just simple word representation. These mechanisms provide the possibility to make semantic search in contrast to the currently used search technology, providing results more adapted to what users want.

Similarly, frequent pattern discovery in texts may enhance the recognition of semantic relations between words. As a result this recognition can help discover the meaning associated to a word. The Data Mining techniques adopted to achieve this task were FP-Growth to discover frequent patterns and Association Rules to provide more than the just recognition of relations between the words (refer to chapter3.3.3 and 3.4 respectively).

Based in the AR algorithm, this work demonstrated that it is possible to measure the strength of a relation.

Pattern recognition by itself can be an indicator of relations between words, however this can be enhanced through the use of a domain ontology. In this sense, this work proposed Frequent Itemset Mapping, a process to match frequent items discovered in a document corpus and keywords associated to concepts from a domain ontology related to B&C (refer to chapter 4.2.3).

As explained in Chapter 2.3, Ontology Learning is the area related to the automatic or semi-automatic (meaning without human supervision) maintenance of an ontology. Through newly discovered knowledge sources it is possible to learn a domain ontology, in the sense that one can use this new knowledge that could otherwise be unknown, be difficult to discover or be unavailable to improve concept relations inside the ontology. The method proposed in this dissertation, based in the AR algorithm, provides metrics in the form of numeric values to evaluate the strength of semantic relations between concepts. Through these value is possible to know if a concept A is more related to a concept B than to a concept C, therefore learn or maintain the concepts from a domain ontology, as the one used in this work related to B&C.

7.2 Contributions

The development of this work proposed four expected outcomes in Chapter 1 as follows:

- To develop a method to describe how to extract concepts and recognize relations between them from a data document corpus, and to find new knowledge sources in order to update a domain ontology.

The proposed method relies on applying Data Mining techniques to discover knowledge in documents that could be useful to update a domain ontology. Knowledge, meant the discovery of new concepts, relations or the improvement of the relations between the concepts already in a domain ontology (e.g. the ontology used in this work was adopted from the B&C domain). The initial resources were a set of documents from the ICONDA[14] database and a domain ontology adopted from SEKS framework developed

[14] ICONDA is a large database of technical documents related to B&C domain.

under a MSc. Dissertation (Figueiras, 2012) both related to B&C domain. The documents were initially processed in the Rapidminer software tool. Rapidminer proved to be a satisfactory tool, as it also allowed to apply the algorithms FP-Growth to discover frequent patterns and Association Rules to discover the relations. The process created for the matching between the frequent itemsets discovered in the documents and the ontology equivalent terms associated to the concepts from the domain ontology was the Frequent Itemset Matching (refer to chapter 4.2.3). This process allowed to search through the ontology in order to verify if the frequent itemsets discovered in the documents were associated to any concept inside the ontology, or if it originated new knowledge.

This work tries to develop a method for Ontology Learning (OL) where it is possible to turn a domain ontology more up to date. Even with a small sample, this process provided some good results (refer to chapter 6), as it discovered new concepts, and also provided some interesting relations between the concepts. However, OL relies on automatic methods, this work did not intended to provide a full automatic method to learn an ontology. Alternatively, it was intended to develop a semi-automatic method that relies in human interaction to complete the OL task with the knowledge discovered through all the results.

- To develop a proof of concept, a software platform, based in the previous method in order to reflect the application of the studied techniques.

In order to execute all the steps from the method proposed it was developed a software tool, DOKS (refer to section 5.2.5). DOKS is a client-server application developed using Java technology to implement all the processes and components in this tool. To interact with the ontology, it was used Jena API. The communication to the database was made by JavaBeans technology. The ontology was developed in OWL. Rapidminer provided an API to access its results, and they were exported through a script represented in Groovy. To hold the results for later access, it was created an XML message. Both DBs for the ontology and for the AR results were saved in a MySQL RDBMS.

- Present results of the semi-automatic OL process. Results composed by patterns discovered in the documents, their relations and the new concepts discovered. They should be presented in an understandable way to the user.

To present the results from DOKS, a FrontEnd was implemented in web technology. Here the set of technologies used were: (i) Html5+CSS3 as a base to support the layout; (ii) The communication with the server was made through HTTP requests based on PHP technology to send the results; (iii) To present the results in the web page, the technology chosen was PHP + XPath. The results were presented in a first page, in which the user could choose two concepts, based on the Frequent Itemset Mapping, and see the values of the metrics from each association rule presented by the FrontEnd. This way, a relation between two concepts could be chosen for later processing. It is worth mentioning, the creation of a colour scheme for the Frequent Itemset Mapping process, in order to help the user choose the concept from the domain ontology that best matches the frequent item.

- Finally, publication of scientific documents about the work, to be assessed by the academic community.

The following scientific documents were published after assessment by the academic community during the development of this work:

- Luis Paiva, Ruben Costa, Paulo Figueiras, Celson Lima, "Discovering Semantic Relations from Unstructured Data for Ontology Enrichment - Association rules based approach", 8ª Conferência Ibérica de Sistemas e Tecnologias de Informação: CISTI'2013, pp 579-584, 2013
- Ruben Costa, Paulo Figueiras, Luis Paiva, Ricardo Jardim-Gonçalves, Celson Lima, "Capturing Knowledge Representations Using Semantic Relationships An Ontology-based Approach", Sixth International Conference on Advances in Semantic Processing: SEMAPRO 2012, pp 75-81, 2012
- Paulo Figueiras, Ruben Costa, Luis Paiva, Ricardo Jardim-Gonçalves, Celson Lima, "Information Retrieval in Collaborative Engineering Projects-A Vector

Space Model Approach", International Conference on Knowledge Engineering and Ontology Development: KEOD2012, pp 233-238, 2012

7.3 Future Directions

As this work relates to some areas from Semantic Web and Ontology Engineering, some possible directions can be identified for further work and improvement. Two paths are proposed, one related to the improvement of the presented method, the second related to its applicability and reuse.

Sometimes, the knowledge that results from the method proposed herein can be huge, and if the process is not fully automated it can be an exhaustive task to analyse these results. This suggests further research related to DOKS ability to deal with the size growth of data used in the Ontology Learning process, can be identified in three areas: (i) speed to process large sets of data as it can be really slow. Research can be taken in methods to, for instance, take advantage of multi-core processor technology in order to use parallelization techniques to improve the speed of the matching process; (ii) way to present results for evaluation by an expert, although this work provided a colour scheme to represent the strength of the matching process (refer to Table 4.3 above). This means to improve the way in which the results are shown, by using more graphics (e.g. graphs to represent relations). This will provide a better efficiency of the method itself and allow for faster reasoning of the results; (iii) method to process large/huge and complex sets of data, also known as Big Data. Big Data is the nouvelle sub domain of Data Mining that studies solutions to the problem of big and complex sets of data.

Searching for patterns in a document, was proved by this work that it is not an easy task, although it is possible. The relation between words in a document can lead to the discovery of a central concept or idea that could represent its context or domain, for instance a document including the words "*bridges*" and "*buildings*". The central concept from this document could be identified as "*Civil Engineering*". However, how can one discover the central idea in a document? Is it even possible? Can this discovery be done?

How to find the central idea? Syntactic Context[15] or Latent Semantic Analysis[16] are areas that tries to address this questions, and can be a promising future direction.

It is worth mentioning that the intention of this research was not to develop a fully functional model to deal with data mining. However, the author thinks that it could be a good contribution to the following areas:

- Information Systems: Search engines like Google, Bing or Yahoo, just to name a few, could use semantic search capabilities to improve its results instead of just statistical ones. For instance, if one would like to search for a car, the search engine could provide the pages where the word "*car"* appears, and also the pages where the word *"automobile"* appears based on the relation between these two words. This means that one could search for a concept, instead of having to know every word that represents it. Additionally, it could also provide suggestions, for instance, related to their brands, turning the search results more close of what the user really searched for.
- Cybersecurity: This is an area of great interest nowadays, based on several world events related to terrorism. The method proposed in this work, could help in this area, for instance, if one could use a search engine to look up in the web for a person A that could be known as related to Al-Qaeda. After using the method proposed, one could also discover a person B that appears frequently in some pages related to person A, although not directly related to Al-Qaeda. This could be a proof of the relation between both people, and the discovery of the relation of person B to Al-Qaeda.
- Cybersecurity and human rights: MEMEX is a project from DARPA with an initial goal of using search technology to help fight human trafficking, as they identified this as a serious problem to solve. The secondary goal of this project was identified as to improve the search mechanisms and tools that are used today. Semantic search could help in the sense that it could discover pages with terms related to human traffic, for example *human trade* or *modern slavery*, which could represent the same idea.

[15] Syntatic Context relates to the order of the words in a sentence, and states that through language rules, one can infer the context of a sentence.
[16] Latent Semantic Analysis is the area that analyse the relations between documents, trying to find correspondence between its terms and concepts in order to infer its context.

- Team Sports: GloballCoach is a software tool idealized by former Liverpool and Chelsea coach, Rafael Benitez, that targets Football Teams and their coaches, providing them data analysis capabilities. Amongst others, this software aims to recognize patterns in game data to show to team players and improve their tactical and technical behaviour. This kind of software systems are focused in the individual behaviours of each player. In this sense, the method proposed in this work could be a great aid in the recognition of relations between the players, augmenting the analysis from an individual up to team perspective analysis capabilities. For instance, the coach could analyse which is the best relation in their left side. Meaning that if he wants to select player A for a game, he could know if the relation between players A and B provides more goals than with A and C. Other example could be, to whom a player A provides more assistances[17], meaning that the relation between player A and B provides more goals to the team than a relation between player A and C.

Summing up, the semantic search is here to stay and is spreading along all research areas. In this sense, controlled vocabularies are useful tools to enhance semantic search capabilities in information systems. Ontologies themselves are great mechanisms to provide search capabilities to users, experts or not, in theirs daily search quests. Consequently, knowing how to provide or obtain the best results will throw companies or entities one technological step ahead of their competitors.

[17] Assistance is the word used in a football game and represents the moment when a player passes the ball to a teammate, and this teammate scores a goal.

8

Bibliography

Agrawal, R., Imieliński, T., Swami, A., 1993. Mining Association Rules Between Sets of Items in Large Databases, in: SIGMOD '93. ACM, New York, NY, USA, pp. 207–216. doi:10.1145/170035.170072

Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules, in: Proc. of 20th Intl. Conf. on VLDB. pp. 487–499.

Allen, R.E., Mannion, J., 2007. Oxford Mini School Dictionary & Thesaurus. Oxford University Press.

Almeida, T., Souza, R.F. de, 2011. O vocabulário controlado como instrumento de organização e representação da informação na FINEP [WWW Document]. URL http://repositorio.ibict.br/handle/123456789/88 (accessed 2.6.15).

Antunes, J.P.D., 2010. Design and implementation of an autonomous, proactive, and reactive software infrastructure to help improving the management level of projects.

Aquino, M.C., 2007. Hipertexto 2.0, folksonomia e memória coletiva: um estudo das tags na organização da web. Rev. E-Compós 18.

Azevedo, P.J., Jorge, A.M., 2007. Comparing Rule Measures for Predictive Association Rules, in: Kok, J.N., Koronacki, J., Mantaras, R.L. de, Matwin, S., Mladenič, D., Skowron, A. (Eds.), Machine Learning: ECML 2007, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 510–517.

Azevedo, P.J., Silva, C.G., Rodrigues, J.R., Loureiro-Ferreira, N., Brito, R.M.M., 2005. Detection of Hydrophobic Clusters in Molecular Dynamics Protein Unfolding Simulations Using Association Rules, in: Oliveira, J.L., Maojo, V., Martín-Sánchez, F., Pereira, A.S. (Eds.), Biological and Medical Data Analysis. Springer Berlin Heidelberg, pp. 329–337.

Bayardo, R.J.. J., Agrawal, R., 1999. Mining the Most Interesting Rules, in: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '99. ACM, New York, NY, USA, pp. 145–154. doi:10.1145/312129.312219

Bhujade, V., Janwe, N.J., 2011. Knowledge Discovery in Text Mining Technique Using Association Rules Extraction. Presented at the 2011 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 498–502. doi:10.1109/CICN.2011.104

Bonchi, F., Goethals, B., 2004. FP-Bonsai: The Art of Growing and Pruning Small FP-Trees, in: Dai, H., Srikant, R., Zhang, C. (Eds.), Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 155–160.

Borgelt, C., 2005. An Implementation of the FP-growth Algorithm, in: Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations. ACM, pp. 1–5.

Brachman, R.J., Selfridge, P.G., Terveen, L.G., Altman, B., Borgida, A., Halper, F., Kirk, T., Lazar, A., McGuinness, D.L., Resnick, L.A., 1993. Integrated support for data archaeology. Int. J. Intell. Coop. Inf. Syst. 2, 159–185.

Brin, S., Motwani, R., Ullman, J.D., Tsur, S., 1997. Dynamic Itemset Counting and Implication Rules for Market Basket Data, in: Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, SIGMOD '97. ACM, New York, NY, USA, pp. 255–264. doi:10.1145/253260.253325

buildingSMART, 2015. IFC Overview summary — Welcome to buildingSMART-Tech.org [WWW Document]. URL http://www.buildingsmart-tech.org/specifications/ifc-overview (accessed 4.19.15).

Buitelaar, P., Cimiano, P., Magnini, B., 2005. Ontology learning from text: An overview.

Cimiano, P., Mädche, A., Staab, S., Völker, J., 2009. Ontology learning, in: Handbook on Ontologies. Springer, pp. 245–267.

Cimiano, P., Völker, J., 2005. Text2Onto, in: Natural Language Processing and Information Systems. Springer, pp. 227–238.

Construction Specifications Institute, Construction Specifications Canada, 2015. MasterFormat: Introduction & Guides [WWW Document]. URL http://www.masterformat.com/about/history/ (accessed 4.18.15).

Costa, R., 2014. Semantic Enrichment of Knowledge Sources Supported by Domain Ontologies. Faculty of Science and Technology - New University of Lisbon, Lisbon.

De Nicola, A., Missikoff, M., Navigli, R., 2009. A software engineering approach to ontology building. Inf. Syst. 34, 258–275. doi:10.1016/j.is.2008.07.002

Dhar, V., Tuzhulin, A., 1993. Abstract-driven pattern discovery in databases. IEEE Trans. Knowl. Data Eng. 5, 926–938. doi:10.1109/69.250075

El-Diraby, T.A., Lima, C., Feis, B., 2005. Domain taxonomy for construction concepts: toward a formal ontology for construction knowledge. J. Comput. Civ. Eng. 19, 394–406.

Elsayed, A., El-Beltagy, S.R., Rafea, M., Hegazy, O., 2007. Applying data mining for ontology building. Proc ISSR.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996. The KDD process for extracting useful knowledge from volumes of data. Commun. ACM 39, 27–34.

Figueiras, P.A., 2012. A framework for supporting knowledge representation – an ontological based approach.

Fraunhofer IRB, 2015. Fraunhofer IRB | ICONDA©Bibliographic - Find and Access Publications on Planning and Building [WWW Document]. URL http://www.irb.fraunhofer.de/iconda/login/ICONDA/iconda-start-info.jsp (accessed 4.19.15).

Fraunhofer, I.R.B., 1986. ICONDA Bibliographic-Find and Access Publications on Planning and Building.

Fukuda, T., Morimoto, Y., Morishita, S., Tokuyama, T., 1996. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. ACM SIGMOD Rec. 25, 13–23.

Gandon, F., Dieng-Kuntz, R., Corby, O., Giboin, A., 2002. Semantic Web and Multi-Agents Approach to Corporate Memory Management, in: Musen, M.A., Neumann, B., Studer, R. (Eds.), Intelligent Information Processing, IFIP — The International Federation for Information Processing. Springer US, pp. 103–115.

Gangemi, A., Presutti, V., 2009. Ontology Design Patterns, in: Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Springer Berlin Heidelberg, pp. 221–243.

Garg, K., Kumar, D., 2013. Comparing the Performance of Frequent Pattern Mining Algorithms. Int. J. Comput. Appl. 69, 21–28.

Gargouri, Y., Lefebvre, B., Meunier, J., 2003. Ontology maintenance using textual analysis, in: Proc. 7TH World Multi Conference on Systemics, Cybernetics and Informatics, USA. List of Figures Figure. Citeseer.

Gonçalves, E.C., 2005. Regras de associação e suas medidas de interesse objetivas e subjetivas. INFOCOMP J. Comput. Sci. 4, 26–35.

Google, 2013. Google.com [WWW Document]. URL https://www.google.com/ (accessed 7.7.14).

Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowl. Acquis. 5, 199–220. doi:10.1006/knac.1993.1008

Guillaume Laforge, 2012. Groovy [WWW Document]. URL http://groovy.codehaus.org/

Hand, D.J., Mannila, H., Smyth, P., 2001. Principles of Data Mining. MIT Press.

Han, J., Kamber, M., Pei, J., 2011. Data mining: concepts and techniques, 3rd edition. ed. Elsevier.

Han, J., Pei, J., Yin, Y., 2000. Mining frequent patterns without candidate generation, in: ACM SIGMOD Record. ACM, pp. 1–12.

Han, J., Pei, J., Yin, Y., Mao, R., 2004. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowl. Discov. 8, 53–87.

Hazman, M., El-Beltagy, S.R., Rafea, A., 2011. A survey of ontology learning approaches. database 7, 6.

Hilbert, M., López, P., 2011. The world's technological capacity to store, communicate, and compute information. science 332, 60–65.

Hilderman, R.J., Hamilton, H.J., 1999. Knowledge discovery and interestingness measures: A survey. Department of Computer Science, University of Regina.

Hoque, A.M.S., Mondal, S.K., Zaman, T.M., Barman, P.C., Bhuiyan, M.., 2011. Implication of association rules employing FP-growth algorithm for knowledge discovery, in: 2011 14th International Conference on Computer and Information Technology (ICCIT). Presented at the 2011 14th International Conference on Computer and Information Technology (ICCIT), pp. 514–519. doi:10.1109/ICCITechn.2011.6164843

Huang, A., 2008. Similarity measures for text document clustering, in: Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand. pp. 49–56.

IBM - International Business Machines, 1996. IBM Intelligent Miner User's Guide, Version 1 Release 1. SH12-6213-00 edition, July.

Inmon, W.H., Osterfelt, S., 1991. Understanding data pattern processing: the key to competitive advantage. QED Information Sciences, Inc.

Kashyap, V., 1999. Design and creation of ontologies for environmental information retrieval, in: Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management. Citeseer, pp. 1–18.

Koh, Y.S., Rountree, N., 2005. Finding Sporadic Rules Using Apriori-Inverse, in: Ho, T.B., Cheung, D., Liu, H. (Eds.), Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 97–106.

Korczak, J., Skrzypczak, P., 2012. FP-Growth in Discovery of Customer Patterns, in: Aberer, K., Damiani, E., Dillon, T. (Eds.), Data-Driven Process Discovery and Analysis, Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, pp. 120–133.

Kumar, V., Chadha, A., 2012. Mining association rules in student's assessment data. Int. J. Comput. Sci. Issues 9, 211–216.

Lavrač, N., Flach, P., Zupan, B., 1999. Rule evaluation measures: A unifying view. Springer.

Lei, Y., Uren, V., Motta, E., 2006. SemSearch: A Search Engine for the Semantic Web, in: Staab, S., Svátek, V. (Eds.), Managing Knowledge in a World of Networks, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 238–245.

Lima, C., 2004. Final draft CWA4 proposal "European eConstruction Ontology "version 2004–03–26, in: Workshop on eConstruction N.

Lima, C., El-Diraby, T., Stephens, J., 2005. Ontology-based optimization of knowledge management in e-construction. J. IT Constr. 10, 305–327.

Lima, C., Fiès, B., Lefrançois, G., Diraby, T., 2003a. The challenge of using a domain Ontology in KM solutions: the e-COGNOS experience. Presented at the International Conference on Concurrent Engineering: Research and Applications, Funchal - Portugal, pp. 771–778.

Lima, C., Fiès, B., Zarli, A., Bourdeau, M., Wetherill, M., Rezgui, Y., 2002. Towards an IFC-enabled ontology for the Building and Construction Industry: the e-COGNOS approach. Presented at the European Conference of Information and

Communication Technology Advances and Innovation in the Knowledge Society, Salford, UK, pp. 254–264.

Lima, C., Fies, B., Zarli, A., El-Diraby, T., Ferneley, E., 2003b. The E-Cognos Project: Current Status and Future Directions of an Ontology-Enabled IT Solution Infrastructure Supporting Knowledge Management in Construction, in: Construction Research Congress. American Society of Civil Engineers, pp. 1–8.

Lima, C., Stephens, J., Böhms, M., 2003c. The bcXML: supporting eCommerce and knowledge management in the construction industry [WWW Document]. URL http://www.itcon.org/cgi-bin/works/Show?2003_22 (accessed 9.6.14).

Lima, C., Zarli, A., Storer, G., 2007. Controlled Vocabularies in the European Construction Sector: Evolution, Current Developments, and Future Trends, in: BSc, G.L., MSc, BEng, R.C. (Eds.), Complex Systems Concurrent Engineering. Springer London, pp. 565–574.

Liu, B., Hsu, W., Ma, Y., 1999. Mining Association Rules with Multiple Minimum Supports, in: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '99. ACM, New York, NY, USA, pp. 337–341. doi:10.1145/312129.312274

Liu, K., Hogan, W.R., Crowley, R.S., 2011. Natural Language Processing methods and systems for biomedical ontology learning. J. Biomed. Inform., Ontologies for Clinical and Translational Research 44, 163–179. doi:10.1016/j.jbi.2010.07.006

Loukides, M., 2010. What is data science.

Mackie, J., 1977. Ethics: Inventing right and wrong. Penguin UK.

Maedche, A., Staab, S., 2001. Learning Ontologies for the Semantic Web. Presented at the Semantic Web Workshop 2001, Hong Kong.

Mahgoub, H., 2006. Mining association rules from unstructured documents, in: Proc. 3rd Int. Conf. on Knowledge Mining, ICKM, Prague, Czech Republic. pp. 167–172.

Memon, N., Hicks, D.L., Larsen, H.L., 2007. Notice of Violation of IEEE Publication PrinciplesHarvesting Terrorists Information from Web, in: Information Visualization, 2007. IV '07. 11th International Conference. Presented at the Information Visualization, 2007. IV '07. 11th International Conference, pp. 664–671. doi:10.1109/IV.2007.60

Metanat Hooshsadat, SAMANEH BAYAT, PARISA NAEIMI, MAHDIEH S. MIRIAN, OSMAR R. ZA?ANE, 2012. UAPRIORI: AN ALGORITHM FOR FINDING SEQUENTIAL PATTERNS IN PROBABILISTIC DATA, in: Uncertainty Modeling in Knowledge Engineering and Decision Making, World Scientific Proceedings Series on Computer Engineering and Information Science. WORLD SCIENTIFIC, pp. 907–912.

N.I.S.O. (US), others, 2005. Guidelines for the construction, format, and management of monolingual controlled vocabularies. NISO Press.

Noy, N.F., Musen, M.A., 2004. Ontology versioning in an ontology management framework. IEEE Intell. Syst. 19, 6–13. doi:10.1109/MIS.2004.33

OCCS Development Committee, 2006. OmniClass: A Strategy for Classifying the Built Environment [WWW Document]. URL http://www.omniclass.org/index.asp (accessed 4.18.15).

Oxford University Press, 2012. Oxford Essential Portuguese Dictionary. Oxford University Press.

Oxford University Press, 2006. Oxford Dictionary of English. Oxford University Press, London.

Paiva, L., Costa, R., Figueiras, P., Lima, C., 2013. Discovering Semantic Relations from Unstructured Data for Ontology Enrichment - Asssociation rules based approach. Presented at the CISTI'2013 - 8ª Conferência Ibérica de Sistemas e Tecnologias de Informação, AISTI, Lisboa, pp. 579–584.

Parada, V.M.M., 2010. Desenho e implementação de um sistema computacional para apoiar a gestão de projectos utilizando técnicas de data mining.

Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L., 1999. Discovering Frequent Closed Itemsets for Association Rules, in: Proceedings of the 7th International Conference on Database Theory, ICDT '99. Springer-Verlag, London, UK, UK, pp. 398–416.

Piatetsky-Shapiro, G., 1991. Discovery, analysis and presentation of strong rules. Knowl. Discov. Databases 229–238.

Pouchard, L., Ivezic, N., Schlenoff, C., 2000. Ontology engineering for distributed collaboration in manufacturing, in: Proceedings of the AIS2000 Conference. Citeseer.

Rácz, B., 2004. nonordfp: An FP-growth variation without rebuilding the FP-tree., in: FIMI.

Reinberger, M.-L., Spyns, P., 2005. Unsupervised text mining for the learning of dogma-inspired ontologies. Ontol. Learn. Text Methods Appl. Eval. 29–43.

Sánchez, D., 2010. A methodology to learn ontological attributes from the Web. Data Knowl. Eng. 69, 573–597.

Semafora Systems, 2012. Semafora systems: OntoStudio [WWW Document]. URL http://www.semafora-systems.com/en/products/ontostudio/ (accessed 4.18.15).

Silberschatz, A., Tuzhilin, A., 1995. On subjective measures of interestingness in knowledge discovery., in: KDD. pp. 275–281.

Spruit, M., 2007. Discovery of association rules between syntactic variables. Citeseer.

Stanford Center for Biomedical Informatics Research, 2011. The Protègè Ontology Editor and Knowledge Acquisition System [WWW Document]. URL http://protege.stanford.edu/ (accessed 4.4.15).

Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T., 2004. Probabilistic Author-topic Models for Information Discovery, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04. ACM, New York, NY, USA, pp. 306–315. doi:10.1145/1014052.1014087

Stick-iSchool, 2013. Innovation Ontolgy [WWW Document]. URL http://stick.ischool.umd.edu/newsite/innovation_ontolgy (accessed 3.24.15).

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D., 2002. OntoEdit: Collaborative Ontology Development for the Semantic Web, in: Horrocks, I., Hendler, J. (Eds.), The Semantic Web — ISWC 2002, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 221–235.

Tan, P.-N., Kumar, V., Srivastava, J., 2002. Selecting the Right Interestingness Measure for Association Patterns, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02. ACM, New York, NY, USA, pp. 32–41. doi:10.1145/775047.775053

Tan, X., Pan, H., Han, Q., Ni, J., 2009. Domain knowledge-driven association pattern mining algorithm on medical images, in: Internet Computing for Science and Engineering (ICICSE), 2009 Fourth International Conference on. IEEE, pp. 30–35.

Tomi Kauppinen, H.K., 2009. Extending an Ontology by Analyzing Annotation Co-occurrences in a Semantic Cultural Heritage Portal.

Uschold, M., Gruninger, M., 1996. Ontologies: Principles, methods and applications. Knowl. Eng. Rev. 11, 93–136.

Velardi, P., Navigli, R., Cucchiarelli, A., Neri, F., 2005. Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies. Ontol. Learn. Popul.

W3C, 2004. OWL Web Ontology Language Semantics and Abstract Syntax Section 2. Abstract Syntax [WWW Document]. URL http://www.w3.org/TR/owl-semantics/syntax.html#2.3 (accessed 4.12.15).

Waller, M.A., Fawcett, S.E., 2013. Data Science, Predictive Analytics, and Big Data: A Revolution That Will Transform Supply Chain Design and Management. J. Bus. Logist. 34, 77–84. doi:10.1111/jbl.12010

Wang, K., Tang, L., Han, J., Liu, J., 2002. Top down FP-Growth for association rule mining. Springer.

Witten, I.H., Frank, E., Hall, M.A., 2011. Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Yahoo, 2015. Yahoo [WWW Document]. Yahoo. URL https://www.yahoo.com/ (accessed 3.22.15).

Yao, Y., Chen, Y., Yang, X., 2006. A Measurement-Theoretic Foundation of Rule Interestingness Evaluation, in: Lin, P.T.Y., Ohsuga, P.S., Liau, D.C.-J., Hu, P.X. (Eds.), Foundations and Novel Approaches in Data Mining, Studies in Computational Intelligence. Springer Berlin Heidelberg, pp. 41–59.

Yu, X., Wang, H., 2014. Improvement of Eclat Algorithm Based on Support in Frequent Itemset Mining. J. Comput. 9, 2116–2123. doi:10.4304/jcp.9.9.2116-2123

Zaki, M.J., 2000. Scalable algorithms for association mining. IEEE Trans. Knowl. Data Eng. 12, 372–390. doi:10.1109/69.846291

Zaki, M.J., Gouda, K., 2003. Fast vertical mining using diffsets, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 326–335.

Zeng, Y., Yin, S., Liu, J., Zhang, M., 2015. Research of Improved FP-Growth Algorithm in Association Rules Mining. Sci. Program. 2015, e910281. doi:10.1155/2015/910281

Zhang, G.-Q., Troy, A.D., Bourgoin, K., 2006. Bootstrapping ontology learning for information retrieval using formal concept analysis and information anchors, in: Proc. 14th Int'l Conf. Conceptual Structures (ICCS'06). Citeseer.

Zhou, L., 2007. Ontology learning: state of the art and open issues. Inf. Technol. Manag. 8, 241–252.

Appendix A – UML Sequence Diagrams

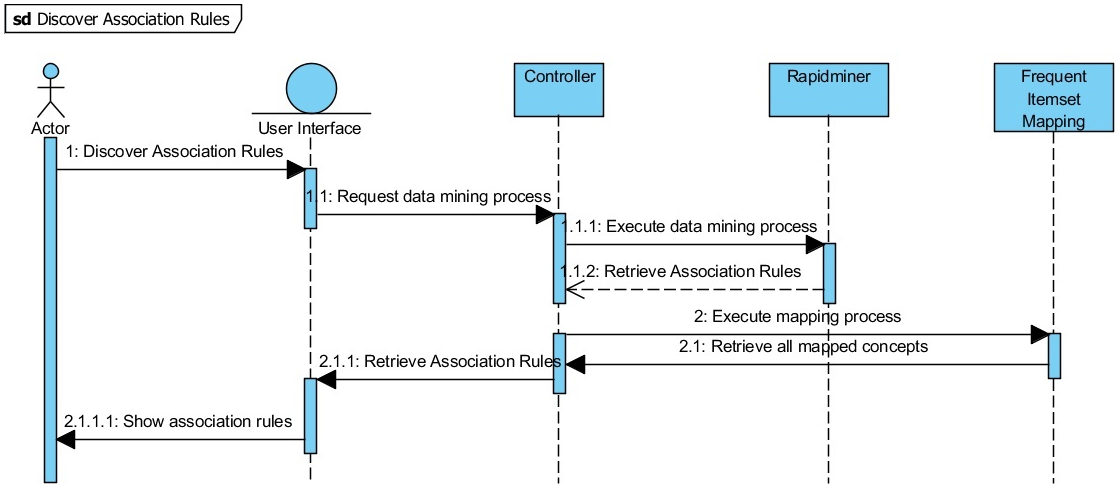


Figure 9.1 - USD for Discover Association Rules

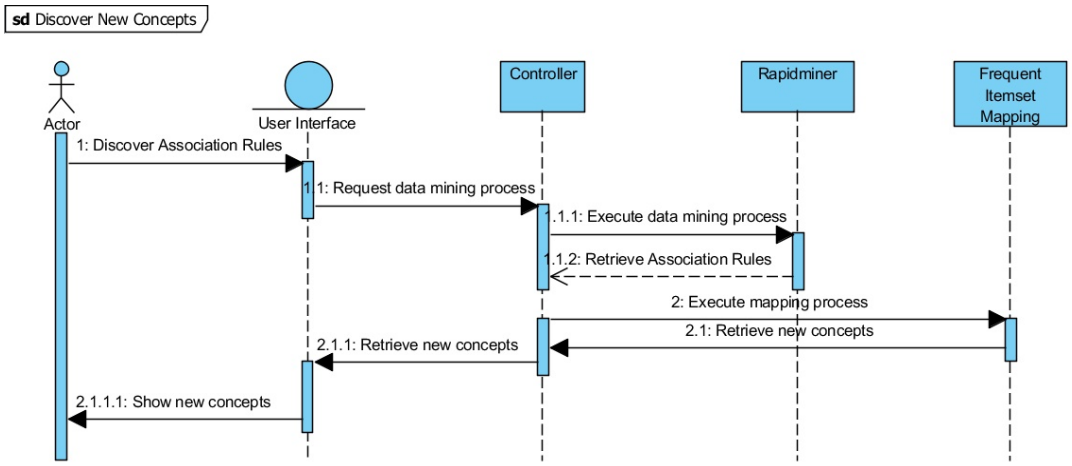


Figure 9.2 – USD for Discover New Concepts

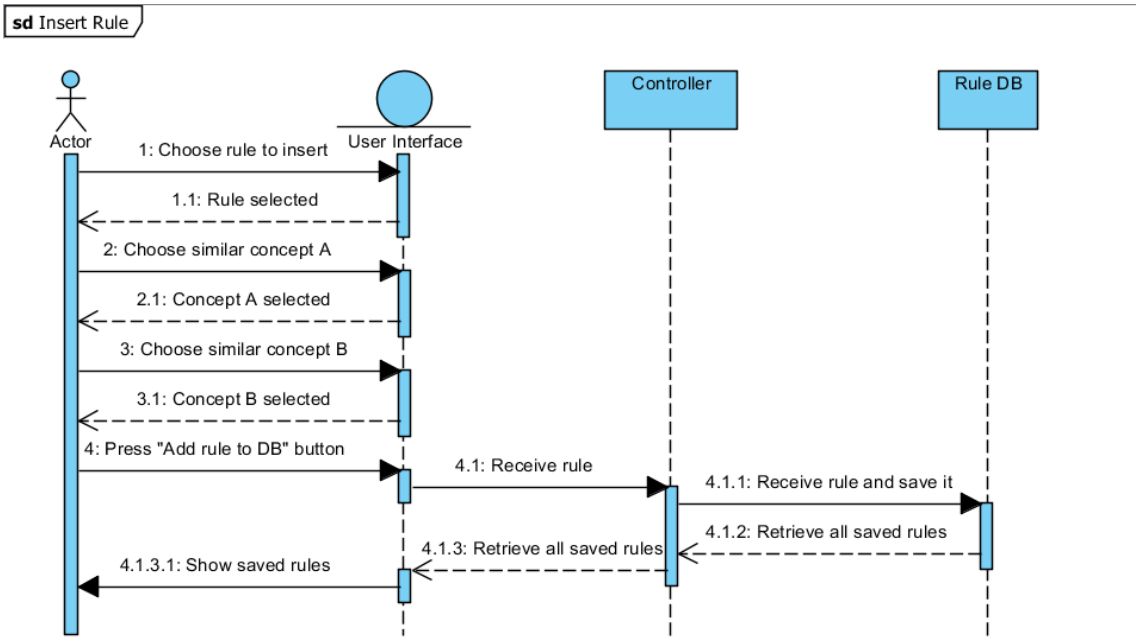


Figure 9.3 – USD for Insert Rule

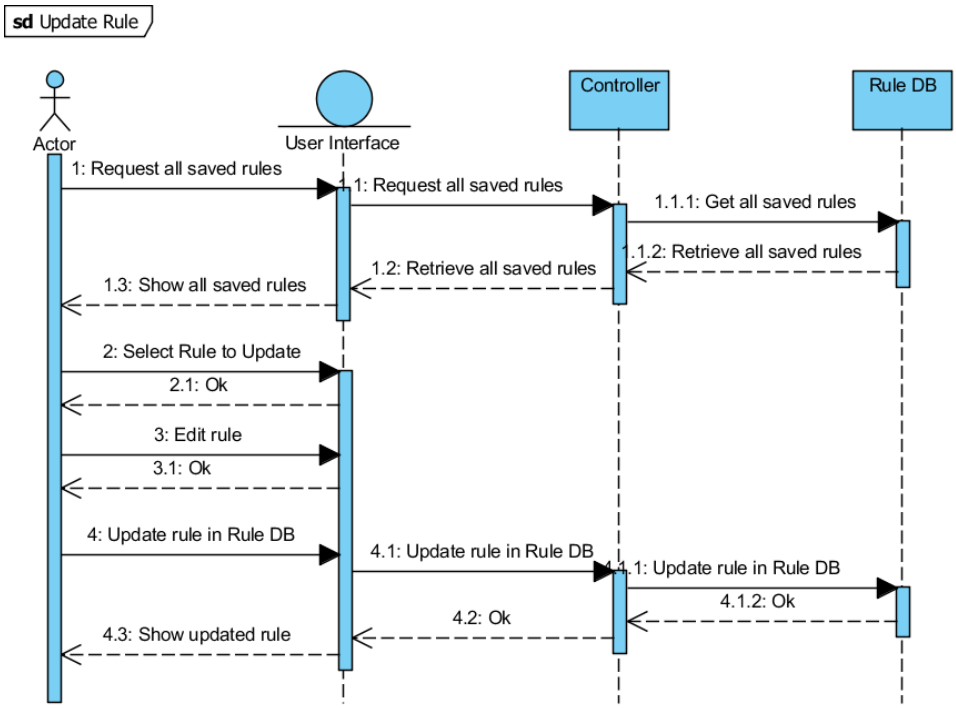


Figure 9.4 - USD for Update Rule

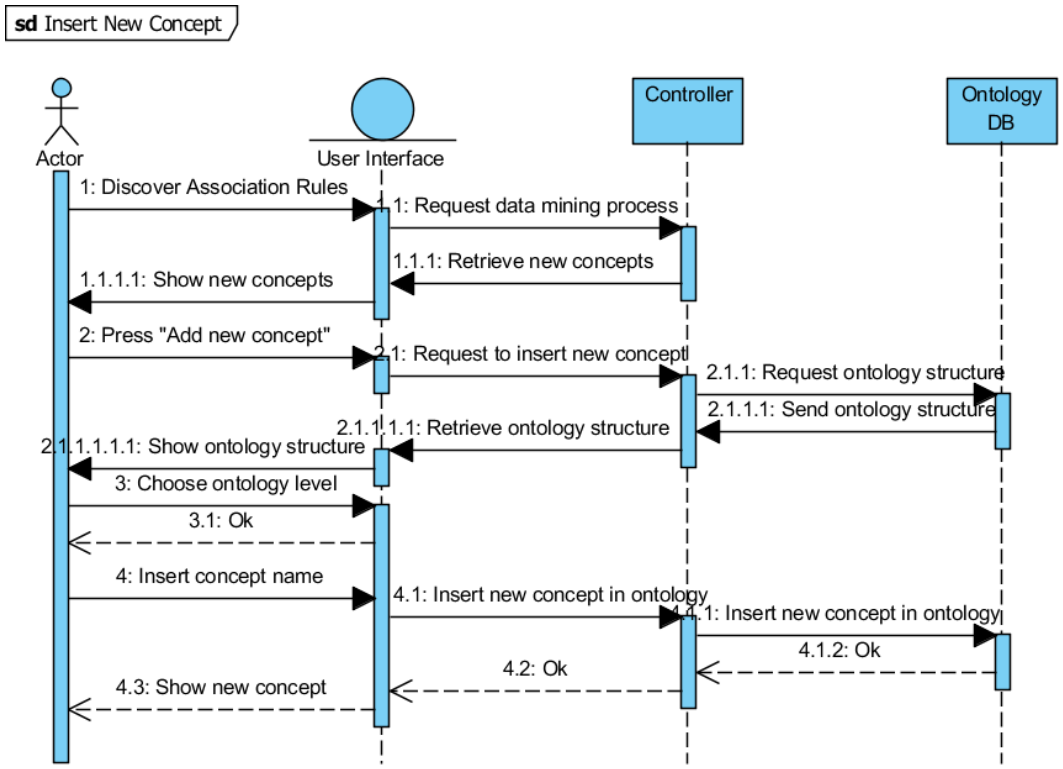


Figure 9.5 – USD for Insert New Concept

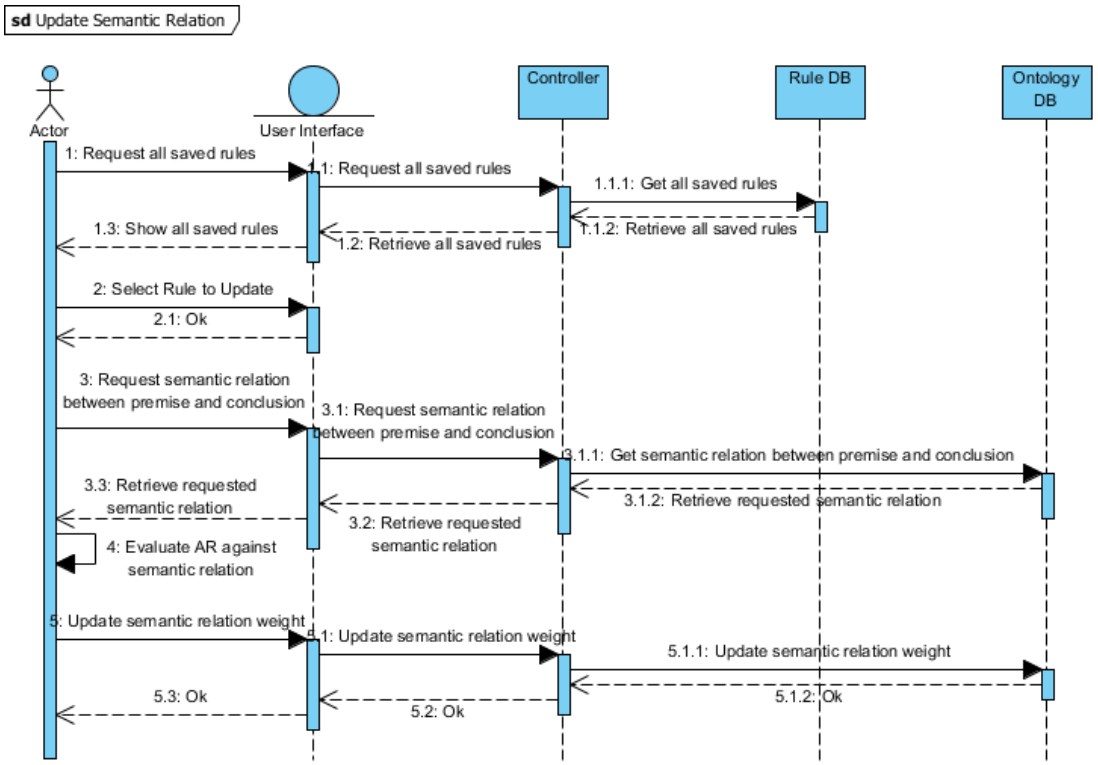


Figure 9.6 – USD for Update Semantic Relation

Appendix B – Association Rules results

This section includes a table with all the association rules discovered in this work. There are 102 discovered. It is worth remember that this results are the output of AR process, which means that the FI are not yet mapped.

Table 9.1 - Association Rules process results

#	Premise	Conclusion	Confidence	Conviction	Gain	Laplace	Lift	Ps	Total Support
1	manag	wast	0.6667	2.2500	-0.4000	0.9231	2.6667	0.1250	0.2000
2	wast	manag	0.8000	3.5000	-0.3000	0.9600	2.6667	0.1250	0.2000
3	manag	recycl	0.6667	2.2500	-0.4000	0.9231	2.6667	0.1250	0.2000
4	recycl	manag	0.8000	3.5000	-0.3000	0.9600	2.6667	0.1250	0.2000
5	manag	wast_manag	0.6667	2.4000	-0.4000	0.9231	3.3333	0.1400	0.2000
6	wast_manag	manag	1.0000	Infinity	-0.2000	1.0000	3.3333	0.1400	0.2000
7	manag	plan	0.6667	2.4000	-0.4000	0.9231	3.3333	0.1400	0.2000
8	plan	manag	1.0000	Infinity	-0.2000	1.0000	3.3333	0.1400	0.2000
9	manag	implement	0.6667	2.4000	-0.4000	0.9231	3.3333	0.1400	0.2000
10	implement	manag	1.0000	Infinity	-0.2000	1.0000	3.3333	0.1400	0.2000
11	energi	consumpt	0.8333	4.2000	-0.3500	0.9615	2.7778	0.1600	0.2500
12	consumpt	energi	0.8333	4.2000	-0.3500	0.9615	2.7778	0.1600	0.2500
13	energi	temperatur	0.6667	2.2500	-0.4000	0.9231	2.6667	0.1250	0.2000
14	temperatur	energi	0.8000	3.5000	-0.3000	0.9600	2.6667	0.1250	0.2000
15	energi	indoor	0.6667	2.2500	-0.4000	0.9231	2.6667	0.1250	0.2000
16	indoor	energi	0.8000	3.5000	-0.3000	0.9600	2.6667	0.1250	0.2000
17	energi	heat	0.6667	2.2500	-0.4000	0.9231	2.6667	0.1250	0.2000
18	heat	energi	0.8000	3.5000	-0.3000	0.9600	2.6667	0.1250	0.2000
19	energi	energi_consumpt	0.8333	4.5000	-0.3500	0.9615	3.3333	0.1750	0.2500
20	energi_consumpt	energi	1.0000	Infinity	-0.2500	1.0000	3.3333	0.1750	0.2500
21	energi	electr	0.8333	4.5000	-0.3500	0.9615	3.3333	0.1750	0.2500
22	electr	energi	1.0000	Infinity	-0.2500	1.0000	3.3333	0.1750	0.2500

23	energi	power	0.6667	2.4000	-0.4000	0.9231	3.3333	0.1400	0.2000
24	power	energi	1.0000	Infinity	-0.2000	1.0000	3.3333	0.1400	0.2000
25	energi	oper	0.6667	2.4000	-0.4000	0.9231	3.3333	0.1400	0.2000
26	oper	energi	1.0000	Infinity	-0.2000	1.0000	3.3333	0.1400	0.2000
27	energi	hvac	0.6667	2.4000	-0.4000	0.9231	3.3333	0.1400	0.2000
28	hvac	energi	1.0000	Infinity	-0.2000	1.0000	3.3333	0.1400	0.2000
29	energi	cool	0.6667	2.4000	-0.4000	0.9231	3.3333	0.1400	0.2000
30	cool	energi	1.0000	Infinity	-0.2000	1.0000	3.3333	0.1400	0.2000
31	consumpt	temperatur	0.6667	2.2500	-0.4000	0.9231	2.6667	0.1250	0.2000
32	temperatur	consumpt	0.8000	3.5000	-0.3000	0.9600	2.6667	0.1250	0.2000
33	consumpt	indoor	0.8333	4.5000	-0.3500	0.9615	3.3333	0.1750	0.2500
34	indoor	consumpt	1.0000	Infinity	-0.2500	1.0000	3.3333	0.1750	0.2500
35	consumpt	heat	0.6667	2.2500	-0.4000	0.9231	2.6667	0.1250	0.2000
36	heat	consumpt	0.8000	3.5000	-0.3000	0.9600	2.6667	0.1250	0.2000
37	consumpt	energi_consumpt	0.8333	4.5000	-0.3500	0.9615	3.3333	0.1750	0.2500
38	energi_consumpt	consumpt	1.0000	Infinity	-0.2500	1.0000	3.3333	0.1750	0.2500
39	consumpt	electr	0.6667	2.2500	-0.4000	0.9231	2.6667	0.1250	0.2000
40	electr	consumpt	0.8000	3.5000	-0.3000	0.9600	2.6667	0.1250	0.2000
41	consumpt	oper	0.6667	2.4000	-0.4000	0.9231	3.3333	0.1400	0.2000
42	oper	consumpt	1.0000	Infinity	-0.2000	1.0000	3.3333	0.1400	0.2000
43	consumpt	hvac	0.6667	2.4000	-0.4000	0.9231	3.3333	0.1400	0.2000
44	hvac	consumpt	1.0000	Infinity	-0.2000	1.0000	3.3333	0.1400	0.2000
45	consumpt	cool	0.6667	2.4000	-0.4000	0.9231	3.3333	0.1400	0.2000
46	cool	consumpt	1.0000	Infinity	-0.2000	1.0000	3.3333	0.1400	0.2000
47	wast	recycl	1.0000	Infinity	-0.2500	1.0000	4.0000	0.1875	0.2500
48	recycl	wast	1.0000	Infinity	-0.2500	1.0000	4.0000	0.1875	0.2500
49	wast	wast_manag	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
50	wast_manag	wast	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000

51	wast	plan	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
52	plan	wast	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
53	toilet	sanitari	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
54	sanitari	toilet	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
55	temperatur	indoor	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
56	indoor	temperatur	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
57	temperatur	heat	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
58	heat	temperatur	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
59	temperatur	energi_consumpt	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
60	energi_consumpt	temperatur	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
61	temperatur	hvac	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
62	hvac	temperatur	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
63	temperatur	cool	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
64	cool	temperatur	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
65	temperatur	climat	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
66	climat	temperatur	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
67	recycl	wast_manag	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
68	wast_manag	recycl	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
69	recycl	plan	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
70	plan	recycl	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
71	offic	offic_build	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
72	offic_build	offic	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
73	indoor	heat	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
74	heat	indoor	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
75	indoor	energi_consumpt	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
76	energi_consumpt	indoor	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
77	indoor	hvac	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
78	hvac	indoor	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000

79	indoor	cool	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
80	cool	indoor	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
81	heat	energi_consumpt	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
82	energi_consumpt	heat	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
83	heat	hvac	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
84	hvac	heat	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
85	heat	cool	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
86	cool	heat	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
87	energi_consumpt	electr	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
88	electr	energi_consumpt	0.8000	3.7500	-0.3000	0.9600	3.2000	0.1375	0.2000
89	energi_consumpt	oper	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
90	oper	energi_consumpt	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
91	energi_consumpt	hvac	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
92	hvac	energi_consumpt	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
93	energi_consumpt	cool	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
94	cool	energi_consumpt	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
95	electr	power	0.8000	4.0000	-0.3000	0.9600	4.0000	0.1500	0.2000
96	power	electr	1.0000	Infinity	-0.2000	1.0000	4.0000	0.1500	0.2000
97	wast_manag	plan	1.0000	Infinity	-0.2000	1.0000	5.0000	0.1600	0.2000
98	plan	wast_manag	1.0000	Infinity	-0.2000	1.0000	5.0000	0.1600	0.2000
99	hvac	cool	1.0000	Infinity	-0.2000	1.0000	5.0000	0.1600	0.2000
100	cool	hvac	1.0000	Infinity	-0.2000	1.0000	5.0000	0.1600	0.2000
101	coat_materi	coat	1.0000	Infinity	-0.2000	1.0000	5.0000	0.1600	0.2000
102	coat	coat_materi	1.0000	Infinity	-0.2000	1.0000	5.0000	0.1600	0.2000

Appendix C – UML Class Diagram

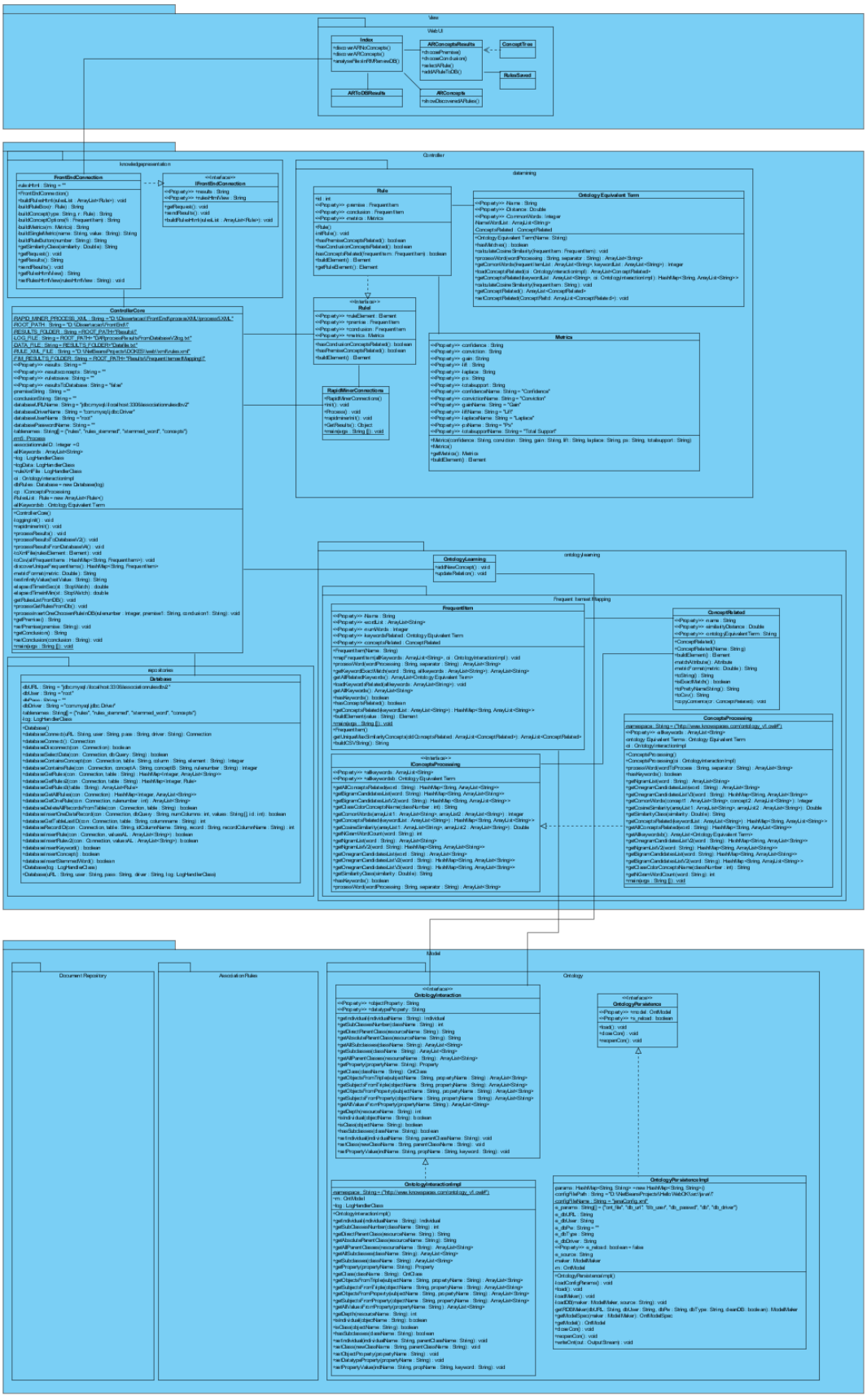


Figure 9.7 - UML Class Diagram