UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II

FACOLTA' DI INGEGNERIA
DIPARTIMENTO DI INFORMATICA E SISTEMISTICA

MULTIMEDIA KNOWLEDGE REPRESENTATION AND MANAGEMENT USING ONTOLOGIES

ANTONIO PENTA

DOTTORATO DI RICERCA IN INGEGNERIA INFORMATICA ED AUTOMATICA

XXI CICLO

Tutore:
Prof. Angelo Chianese
CoTutore:
Prof Antonio Picariello

Coordinatore del Dottorato:
Prof. Luigi Corderlla

Dipartimento di Informatica e Sistemistica
Napoli, Novembre 2008

To my family...

To my friends ...

To my mentor and my friend Picus

Acknowledgements

I would like to thank all the people who helped me during these years of my Phd activities:

My first thanks is for Prof Antonio Picariello for his true friendship, for his help in all my works, for looking me first as men and friend and then as phd student. Picus, thanks for all.

Special Thanks is for Prof Angelo Chianese for giving me the possibility to take my phd in interesting and stimulating environments and for always being helpful to me

I will never forget you. Thanks Angelo & Picus.

Thanks to Prof Lucio Sansone for teaching me a lot of things about the enchanting field of Data Base.

Thanks to my friends and colleague Vinni and Pasquale. Every day is been a great day with you.

Thanks to Prof Antonino Mazzeo and Prof Carlo Sansone for giving me the change to work and to prove some of my research for very interesting problems.

Thanks to Flora Amato e Francesco Gargiulo. Thanks guys, I hope to work with you for new problems trying to do our best as we did.

I would remember that Prof Antonino Mazzeo and Flora Amato who give a great help on the development of the methodology, described in this thesis, related to the text data.

Thanks to Ing Roberto Abbruzzese, Ing Biagio Baiano , Ing Clemente Falsino, who helped me a lot during this work and it was my pleasure to have them first as student and then as friends.

Thanks to all the professors, researchers, technicians, phd students and employs that make the DIS a great place where to work.

Now i would to give my special thanks to the people who I met during my visiting phd experiences:

First of all I would remember Prof Letizia Tanca. One of the person that changes my way to do research. She teaches to me a lot of things, she shows to me a different way to work driven by curiosity and humility, something that you don't acquire from any books.

Thanks Letezia to trust my self and my ideas and to give me your friendship and the hope the what is matter is to do our best.
Special thanks to Context-Addict group of Politecnico of Milano: Prof F.A. Schreiber, Prof C. Bolchini, Prof E.Quintarelli, Phd R. Rossato, Phd C. Curino, Phd G. Orsi. They are special people. Thanks a lot.
Special Thanks to Prof Sibel Adali, Prof Maria Luisa Sapino, Prof VS Subrahmanian to give some of their times and to share with me their cleverness.
Special Thanks to all the phd guys who I met in Politecnico of Milano for their friendship. Thanks guys, you are a wonderful people. (Thanks Romolo)
Special Thanks to all the phd guys who I met in Rensselaer Polytechnic Institute, they made my USA visiting experience a great time. Thanks guys for you friendships. (Thanks Onur)

Indice

Capitolo 1

Introduction

Introduction

Despite the great efforts in multimedia processing in both the academic and industrial contexts, nowadays representing, organizing and managing multimedia data and the related semantics by means of a formal framework still remains a challenge. In the data and knowledge engineering community, the formal representation of domain knowledge is systematically treated by using ontologies. Especially in the semantic web field, many papers, several models and language have been proposed in order to define the concept of ontology and to design suitable and efficient systems that really use the ontological framework for real problems. In the multimedia community, a great emphasis has been given to the *extensional aspects* of multimedia ontology: it is easy to find ontologies about images, videos and audios that contain a number of relevant information about technical aspects related to multimedia data, its format and a variety of ways used for annotating complex and rough data. Unfortunately, the same is not true as far as the *intensional aspects* of multimedia ontologies are concerned. Indeed, there is still great work to be done concerning this aspect: starting from the very beginning, it is still not at all clear whether a multimedia ontology is simply a taxonomy, or a semantic network, what is the role of concrete data (if any) or whether it is a simple organization of metadata. In addition, the semantics of multimedia data itself are very hard to define and to capture: for example, in the image domain, the information carried by an image is inherently both complex and uncertain. Therefore, its semantics has a fuzzy nature.

To best understand the aims of the present dissertation, let us describe a sample scenario from common real life situations.

Let us consider a secret service investigation of a large scale anti-terrorristic operation.

In order to carry out the investigation successfully and to avoid dramatic events, the agents use a large number of electronic devices, to conduct surveillance of places, of people involved and suspected members of the terroristic organizations.

In particular, they may use the following devices in order to gather data and information:

- The officer may have video cameras to record activities of suspected persons at various places.
- In addition, the officer may have legally (hopefully) authorized telephone wiretaps, collecting audio involving conversations that the suspects have participated in.
- The agent may also have a number of photographs taken, containing faces of suspected people and or a number of illegal activities.
- The officer may have a great number of textual documents, containing a description of all the previous investigations, judge-court sentences about such people, and so on.
- Eventually, a relational data base may contain several structured information, i.e. bank account transactions, credit card and so on.

In all the previous events, it is clear that the core aspect of the investigation is the idea of multimedia document, containing a variety of formats (textual, pictorial, video, audio) and a variety of metadata, sometime manually added to the multimedia sources and sometimes automatically extracted.

Note that in real cases, the number of multimedia data is very huge and the only way to process such a large number of data is to use automatic tools that can extract information and represent them in a suitable way. In this framework, it is mandatory to provide a novel methodology for storing and accessing multimedia data, taking into consideration both the variety of data sources and the associated uncertainty of automatic analysis and recognition systems.

In this dissertation we will describe a novel formal framework for multimedia ontologies, and in particular for an image and text data. I propose a framework based on a constructivist vision of multimedia data processing and representation. In other words, I provide a suitable knowledge base that can be used for storing and managing different levels of multimedia data in terms of rough data, intermediate and high level concepts, as wall as some abstractions that can be observed over them. In this way, I provide a comprehensive framework that can be used for a variety of purposes: e.g. information storing and management, information extraction, information retrieval and automatic annotations. In order to make our theory understandable, we will concentrate in particular on image data and text data. In other words, I will try to answer the following questions: do we really need yet another knowledge framework for images? What is a multimedia ontology? Can this kind of ontology be suitable for representing both intensional and extensional aspects of multimedia data? What kinds of advantages do we get from image annotation?

Starting from this considerations, I will provide details concerning: i) how I represent and manage the multimedia information; ii) how I derive high level concepts, considering the discovered features, objects and elementary concepts.

The specific contribution of my research can be summarized in the following points:

1. *Multimedia Ontology* - I introduce a novel definition of multimedia ontology with takes into account both: i) low level data; ii) intermediate data structures, iii) semantic description and iv) complex concept representation.

2. *Uncertainty Management* - I extend several models based on fuzzy logic to Ontology Theory.

3. *Query Algorithms for Image Retrieval* - I provide suitable query processing algorithms based on the proposed multimedia ontology theory.

4. *Application to Italian e-Gov Domain* - I describe an application based on the developed theory from the italian applicative domain and in particular fo e-government application

The dissertation is organized as follows.

- Chapter 1 is devoted to give a comprehensive discussion about what we mean for knowledge, why is interesting to model and to use it and how engineers cope with it.
- Chapter 2 is devoted to explain how the problems of knowledge representation and management are studied from computer scientists, and more details about the used symbolic languages, the ontology, the reasoning and querying procedures will be given. Some sections are also devoted to understand how this kind of problems is differently studied in Web based environments.
- Chapter 3 is the related work chapter; it is divided into three parts. The first part is devoted to the descriptio of semantic multimedia management approaches and system. The second one is devoted to Image multimedia management approaches and system and the last one is devoted to Text Multimedia management approaches and system.
- Chapter 4 explains the Image Ontology framework, its algorithm, the systems and in particular the use of the approach on image data.
- Chapter 5 is devoted to explain the Text Ontology framework, its algorithm, the systems and in particular the use of the approach on text data.
- Chapter 6 contains some final discussions, conclusions and future works.

Capitolo 2

From Knowledge to Knowledge Engineering

From Knowledge to Knowledge Engineering

2.1 The Knowledge

According to The Oxford English Dictionary, *Knowledge* is: (i) formation and skills acquired through experience or education; (ii) the theoretical or practical understanding of a subject, (iii) what is known in a particular field or in total; facts and information or (vi) awareness or familiarity gained by experience of a fact or situation. Philosophical debates on its definition in general start with Plato's formulation of knowledge as justified true belief, even if we remark that it is very complex to accept a certain definition instead of another one, and we note that an entire science, epistemology, is focused on the research of knowledge definitions. As a consequence of fact, there is however no single agreed definition of knowledge and there remain a number of competing theories. Knowledge is not a belief, i.e a psychological state in which an individual holds a proposition to be true. Believers typically say that they *know* what they *claim*, but this may not be a knowledge. For example, for many years the flat earth theory was an "knowledge "among the people and only some observations modified this "belief ". Infact around 330 BC, Aristotle provided observational evidence for the spherical Earth noting that travelers going south see southern constellations rise higher above the horizon and this was only possible if their horizon was at an angle to northerners' horizon and thus the Earth's surface could not be flat [82]. What the knowledge is, is not only what we prove or what we see through use of scientific evidence. There is also knowledge that are related to different kind of evidence that involved the ratio process and experience of the subject [55].

From a computer science point of view, we are interested into Knowledge formalization and management in order to build “intelligent” systems having a certain grade of “cognitive power” together with a very high computation capability. In other words, knowledge became the bridge that we need to make our computer process more intelligent. In such a system, one is more focused on the described knowledge as domain facts about the data, then the process that transform such data into knowledge. Starting from Alain Turing in the 50’s[123], scientists tried to think how to built machine that are able to solve problems in automatic way. Around the ’70, scientists understood that the main problem to address was how to represent knowledge so that computer machine could use such kind of knowledge for an efficient problem solving strategy: from that point, research on Knowledge Management and Representation have had a great importance in computer science research and practice.

Multidisciplinary research within the cognitive processes framework – such as perception, learning, communication, association and reasoning – also gave a great contribution to the establishment of a modern information management theory.

Starting from these perspectives and in order to manage and represent knowledge, we have to define: how we divided the kinds of knowledge, how we derive the knowledge, how we store the knowledge, what it is the role of knowledge.

However, from a computer science point of view, we can consider different kinds [87] of knowledge; in particular, the main distinction is about declarative knowledge ("*know that* ") and procedural knowledge ("*knowing how* "). Mainly, when we handle with data and their models, we care about the declarative knowledge. We postpone the problem of definition of procedural knowledge when we address the problems to define the way useful to interact with our models. The "*know that* "is divided in tree main parts:

- *Terminological Knowledge*: it is the knowledge of languages and the concepts that belong to them. For example the knowledge of "car "implies that it is a "vehicle with tree wheels ".

- *Reliable Knowledge*: it is the knowledge of world regularities and general laws. For example the "mother is older that her son ".
- *Factual Knowledge*: it is the knowledge of facts. For example "Chicca "is the "dog of Antonio ".

How we derive knowledge? Our knowledge, typically, comes from: *direct experience*, i.e the experience obtained from an individual by means of her/his interaction with the surrounding world; *reasoning*, i.e. deductive reasoning (from the assumptions to the conclusions), abductive reasoning (from the observations to the possible causes), inductive reasoning (from particular facts to the general laws); *communication* , i.e knowledge obtained from a system of signs.

The second point to address is: where we store all this information? We need complex structures that are able to represent all the connections and the nature of all this information. The best place, where the nature of knowledge is preserved, is our mind. It has some functionality that can handle the complex structure of knowledge and that are able to retrieve all the information in efficient way. We need models and framework that are able to store the information, in such a way that they preserve all the related knowledge.

Last, but not least, we have to consider: what is the role of knowledge? The main role is to understand data in order to determine what happened or what will happen in the feature, thus deriving right planes and reaching significative goals.

Summing up, a computer scientist needs to design and develop models and techniques for building knowledge based systems, i. e. methodologies, competences, creative applications of scientific principles and human thinking that are the foundation of all the engineering research fields.

2.2 The Knowledge Engineering

Considering the previous discussion, knowledge is in other words what someone has after having "understood "a certain kind of information. Often this understanding follows the development of a detailed or long-term relationship with the known objects, persons or general things. Such a process can often be accelerated when the need to use the information for a critical decision arises. However, it should be clear that data, information and knowledge are not static things in themselves but stages in the process of using data and transforming it into knowledge. On this basis they can be considered points along a continuum, moving from less to more usefulness to a human being. The movement from data to knowledge implies a shift from facts or observations to more abstract concepts, as shown in figure 2.1.

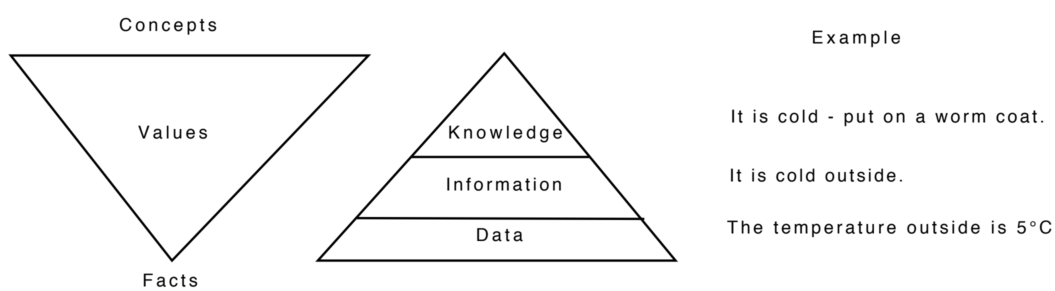


Figura 2.1: Data, Information, Knowledge

According with this vision of knowledge, we need some engineering processes that have the aim to transfer human knowledge into some form of knowledge based system. This could be be done in different activities as:

1. Knowledge acquisition
2. Knowledge validation
3. Knowledge representation
4. Inferencing

5. Explanation and justification.

Knowledge acquisition involves obtaining knowledge from various sources including human experts, multimedia data and existing data sources such as databases and the Internet. In *knowledge validation*, knowledge is checked using test cases for adequate understanding the quality of the obtained knowledge. *Knowledge representation* involves producing a map of the knowledge and then encoding this knowledge into the knowledge base. *Inferencing* means forming new knowledge, the implicit one, through the inferences in the knowledge so that the Knowledge Base System can make a decision, extraction or retrieval of information or to provide advice to the user. *Explanation and justification* involves additional computer program design, primarily to help the computer answer questions posed by the user and also to show how a conclusion was reached using knowledge in the knowledge base. These activities are the purpose of the Knowledge Engineering research field. In fact, the *Knowledge Engineering* is defined as the process of developing knowledge based systems in any field, whether it be in the public or private sector, in commerce or in industry [40],[114]. Knowledge Engineering (KE) is more closed to the empirical sciences that have a domain and make computable predictions about the domain. KE has also the purpose of solving practical problems within some constraints.

2.3 The Knowledge Base System

KE must be able to capture the behavioural skills or knowledge of experts and code these into some Knowledge Base System (KBS). In this way, the Knowledge Based Systems are computer programs that are designed to emulate the work of experts in specific areas of knowledge. The main aims of KBSs are to perform many of the tasks undertaken by humans. However, they do have some limitations. When compared with human expertise which is often not very accessible since only one or a few people can consult as expert. These kinds of systems have been used to capture the knowledge of expert staff who have not a computer ability but they know all the domain details. Where human expertise is difficult to transfer between people, the knowledge within any KBS can be re-used and copied around the world. Where humans can be unpredictable, KBSs are consistent. Where human expertise can be expensive and take decades to develop, KBS can be relatively cheap. On the other hand, humans are creative and adaptable, where KBSs are uninspired and developed for fixed purpose. Humans have a broad focus and a wide understanding. Knowledge-based systems are focused on a particular problem and cannot be used to solve multiple problems. Humans can fall back on common sense knowledge and are robust to error. Knowledge-based systems are limited to the technical knowledge that has been built into them. Humans are also very good at processing sensory information. Moreover, this kind of systems needs to process symbolic information suitable structure for storing and manage them. In particular, these systems include modules coming from the Artificial Intelligence (AI) [104], including expert systems, neural networks, case-based reasoning, genetic algorithms, intelligent agents and data mining. The main purposes of AI methods are to get computer systems to emulate some aspects of intelligent behaviour for example:

- making decisions, diagnosing, scheduling and planning using expert systems or neural networks,
- evolving solutions to very complex problems using genetic algorithms,
- learning from a single previous example, where this is particularly relevant and using it to solve a current problem using case-based reasoning,
- recognized object through the extracted features,
- identifying cause and effect relationships using data mining,
- the ability to take independent actions simulated by intelligent agents.

The application of artificial intelligence tried to emulate all of these characteristics within computer systems. Knowledge engineers have the difficult job of attempting to embed these characteristics into a computer program, using some of AI techniques and models.

Capitolo 3

Knowledge Representation and Management

Knowledge Representation and Management

3.1 The Knowledge Representation

The knowledge engineers have the task of building computable models of some domain for some purpose. They design models and develop knowledge representation systems that have the purpose to define the meaning of objects and their one's relations. Any system, AI-based or not, can be said to have knowledge about its world. This idea of attributing knowledge to a more-or-less complex system is what the philosopher Dennet calls "taking the intensional stance ". The field of knowledge bases, knowledge engineering and so on means more than this. Typically when they talk about that, the researchers have in mind a system that not only know a lot in above sense, but also a system that does what it does using a *representation of that knowledge*. The concept of representation is no doubt as philosophically problematic as that of knowledge. The knowledge representation is the field of study how to represent a collection of propositions believed by some agents with using formal symbols [81]. We would not insist that there be symbols to represent each of the prepositions believed by the agents, they could be infinite but only a a finite number of which are ever represented. It will be the role of services such as querying and reasoning to bridge the gap between what is represented and the full set of the given prepositions. I have to clarify that what makes knowledge-based a system is not the use of logical formalism or the fact the is complex enough to merit an intensional description involving knowledge or the fact that what it believes is true; rather it is the presence of a knowledge container such as knowledge base, that is a collection of symbolic structures representing what it believes and reasons with during the operation of the system itself. R. Davis, H. Shrobe, and P. Szolovits in [99] in important article argue that the notion of knowledge representation can best be understood in terms of five distinct roles it plays:

- A knowledge representation is most fundamentally a *surrogate*. Physical object, events, and relationships, which cannot be stored directly in a computer, are represented by symbols that serve as surrogates for the external things. The symbols and the links between them form a model of the external system. By manipulating the internal surrogates, a computer program can simulate the external system or reason about it.

- A knowledge representation is a set of *ontological commitments*. For database and knowledge base, ontology is used to set the category of things and the facts related to an application domain. This kind of knowledge determines the *ontological commitments* defined by the knowledge engineer. I will give more details about ontology in the future section.

- A knowledge representation is a *fragmentary theory of intelligent reasoning*. The KR have to support reasoning about the things defined inside. In order to do that, it has described the behavior and the interactions among those things. The knowledge becomes as a theory of an application domain. A keypoint is to understand how to put this theory in explicit axioms or into executable programs.

- A knowledge representation is a *medium for pragmatically efficient computation*. The system, based on a knowledge base, have to encode it in a way that can be processed efficiently on the available computing equipment. This is an important requirement in the KR, because we can represent easily some interesting problems but solving them may require an enormous amount of time and effort to compute.

- A knowledge representation is a *medium of human expression*. It is important that the choose representation language should facilitate communication between the knowledge engineers and the domain experts who understand the domain and they could validate the whole theory.

The research in knowledge representation has not the target to find new data structures. At each layer of our knowledge the choices being made are about representation, not data structures. Part of what makes a language representational is that it carries meaning that there is a correspondence between its constructs and things in the external world. That correspondence is made up using some constraint. A semantic net [115], for example, is a representation, while a graph is a data structure. Infact they are different kinds of entities, even though one is invariably used to implement the other, precisely because the net has a semantics. That semantics will be manifest in part because it constrains the network topology: a network purporting to describe family memberships as we know them cannot have a cycle in its parent links, while graphs (i.e., data structures) are of course under no such constraint and may have arbitrary cycles. While every representation must be implemented in the machine by some data structure, the representational property is in the correspondence to something in the world and in the constraint that correspondence imposes. As described above, we need languages able to be processed by computer machines and that provide a set of services that have to reason over the knowledge or to manipulate and/or to interrogate the described knowledge. In this thesis dissertation, I have also focused on studying the linguistic aspects of knowledge representation, and also this research could benefit from understanding how the cognitive state of human being plays a role and what kind of role.

This approach have some interdisciplinary aspects. We know something about the human learning process and the location of different tasks of human brain but we are also far from the reproduction of its complex structures and we only know few notions about the mechanism of storing information in human brain. The symbolic approach is more followed and it could be advanced by being a formal instrument to be processed by machines. This approach fits well what I would like to have: a system where the user could define information about situations, objects, relations, plans, etc., by specifying their aspects and the constraints on those aspects, such that it is ,then, able to retrieve or recognizee a particular situation (object, relation). Infact my aims is to find models/languages useful to describe the knowledge and to develop system able to use that models rather than to understand the knowledge process of human brain. Infact, most conventional applications and database systems are language-dependent and they have the purposes to use a models and its languages to store and retrieve information.

In the following section, I will introduce some aspects of KR in terms of languages and theory as ontology. Then I will introduce some aspects related to the management of KR such as querying and reasoning.

3.2 Formal Language for Knowledge Representation

Research in formal languages deals with understanding how to design a theoretical framework that takes into account some requirements in terms of expressivity and complexity, that is enough powerful to represent the complexity of a real domain and that supports some reasoning and querying services that are useful to manage the knowledge. Usually, to build a knowledge representation language means to make an abstraction over a class of representational structures and introduce a syntactic mechanism to represent that abstraction. In addition, another key point is to define a novel language particularly suitable to communicate and share those experiences that our ”human mind” have done. Someone could argue that there is no point in developing new knowledge representation languages because eventually they all come down to list structures; we can argue again to this observation that we have to clarify the semantics of our formal system in order to choose the best data structure that could describe it. Some efforts were done in that research field to understand the best candidate language to use for that aims. Besides representing knowledge, a language must be able to analyze knowledge in low-level primitives and organize it in high-level structures. For example in natural language the basic unit is the word and the basic structure is the sentences, but higher-level structures like paragraphs, sections, chapters are needed to classy the domain knowledge described in certain manuscript. In the following, I will describe the main knowledge framework used in the knowledge engineering process.

3.2.1 Logic based Language

The first researcher in this field was surely Leibniz, that tried to invent a universal language based on mathematical principals [88].

The main purposes of this scientist were to define a language that was precise enough to rectify our reasoning and general enough to settle all disputes among persons. Nowadays we have different versions of logic, each one having the purpose of representing any factual information that can be stated in any language natural or artificial specializing themselves in different way.

There are different kinds of logic: Propositional Logic, Predicate Calculus known also classic First Order Logic (FOL), Model Logic, Horn Logic, Higher Order Logic. A great variety of framework is derived from classical FOL [94].

Some dimensions used to differentiate among those frameworks are given as follows:

- *Syntax* – some versions differ from the others according to the notation used. It's the least important dimension because the many proposition can be expressed in logically equivalent ways.

- *Subset* – they are subsets obtained by applying some constraint on permissible operators or combination of operators, these reductions were done in order to balance the computable and expressivity power of those languages. For example the Prolog language is based on the Horn-clause [94] a subset of FOL, and that restriction makes Prolog fast enough to be a practical programming language.

- *Proof theory* – some versions of logic restrict or extend the permissible proofs. For example the Non-monotonic logics allow the proof procedures to introduce default assumptions if they are consistent with what is currently known.

- *Model Theory* – some versions modify the denotation or truth values of a statement in terms of some model of the world.

One of the most used multi-valued logic is the fuzzy logic, which uses the same notation as classical FOL but with an infinite range of certainty factors. Among all the varieties of logic, classical first order logic has the most importance and it is widely used as declarative langauge for knowledge representation. The researcher try to use the first-order logic as their representation language. Infact, the predicate calculus was appealing because it has a very general expressive power and well-defined semantics. However, because the language constructs are very fine grained and do not provide adequate facilities for defining more complex constructs, domain expert have difficulty using predicate calculus or understanding knowledge express in it. Another drawback of using FOL as language for KR is the undecidable features of his proof procedures. These have a main importance in the reasoning services that a KR systems offers.

3.2.2 Frame based Language

In his famous paper on frames [91], Marvin Minsky defined a frame as:
"A data structures for representing a stereotyped situation, like being in a certain kind of living room or going to a child's birthday party. Attached to each frame are several kinds of information.[...]. We can think at frame as a network of nodes and relations. The top-levels of the frame are fixed and they represent things that are always true about the supposed situation. The lower levels have many terminal (slots) that must be filled by specific instance data ".
This vision had an exciting reaction in the community, and some systems were implemented according to Minsky's guidelines but its importance was in the his emphasis on the need for structure in organizing a knowledge base. In addition, special-purpose deduction algorithms have been developed that exploit the structural characteristics of frames to rapidly perform a set of inference commonly needed in knowledge system applications. The central inference mechanism in the frame-based system is the *inheritance*. Infact,

the frames are organized as hierarchy with some general concepts. OIL [53] and F-logic [75] are two frame based languages and KL-ONE [109], with his network notation, is the ancestor of many frame system. In particular OIL was the first result to define an ontology langauge well designed (intuitive and with adequate express power), well defined (clearly specified syntax and formal semantics) and compatible with existing (web) standard. I would mention this approach because it was a family of language [117]in which it was possible to define knowledge about a certain domain by introducing a number of concepts and by specifying their interrelations. It was given a formal semantics and introduced inference theory and procedural embedding.

3.2.3 Object-Oriented Language

The Object-Oriented languages (OO) [129] are an evolution of object-based language. The language is object-based if it supports objects as a language feature. Support of objects is a necessary but not sufficient requirement for being object-oriented. Object-oriented languages must additionally support object classes and class inheritance. The object has a set of "operations "and a "state "that remembers the effect of operations. The classes instead is a template from which objects may be created by "create "or "new ". operations. Objects of the same class have common operations and therefore uniform behavior. One of the main feature of that languages is that instead of separating the declarations that define an object from the procedures that operate on them, the O-O systems integrate the declarations and the operations for each type of object in the single model. They provides also mechanism of encapsulation that are useful to distinguish the external behavior of object from their internal structure. These languages differ from other paradigms because their representation seeks to group data around objects (at the conceptual level). According to this philosophy in language design the user can access directly or by inheritance all public information about an object once a handle to that object is obtained. These introduce a key-point in that discussion if it is object useful to represent the entire domain knowledge. The Object-Oriented Modelling (OOM) [47] typically results in several tables which list the objects and the relationships between objects and a mapping between the schema of those data and the object of the application. Furthermore, the communication between objects has to be formalized in order to analyze the overall behavior of the system. This approach could be useful to distinguish between the terminological component and the assertional component (classes refer to the terminological component and objects refer to the assertional component). The limit is that all the knowledge about the domain is codified in the objects representations and their relationship in this way there is a strong coupling between what i represent of a domain and how i can manage that domain. These approach don't help the domain expert to design the knowledge and to follow the knowledge evolution.

3.2.4 Description Logic

So far we have presented different representation languages. Logic-based approach are usually a variant of first order predicate calculus and the reasoning play the role of entail logical consequence. In non-logic based approaches, knowledge is represented by means of some ad hoc data structures, and reasoning is accomplished by similarly ad hoc procedures that manipulate the structures. Among these specialized representations we find object-oriented modeling and frames, but also semantic networks and so on. Due to their more human-centered origins, the systems, that followed the non-logic based approach, were often considered more appealing and more effective from a practical viewpoint than the logic systems. Unfortunately they were not fully satisfactory because of their usual lack of precise semantic characterization. The final result was that every system had a different behavior with respect to the others, despite they manage the same components and even identical relationship names.

The main goal became, so, how to provide semantics to that representation structures, in particular to semantic networks and frames, which carried the intuition that, by exploiting the notion of hierarchical structure, one could gain both in terms of simplicity of representation and in terms of efficiency of reasoning. There were many works [64] focuses on that problems, some of them providing a translation of this kind of structure using FOL, in order to give a precise semantic to that approaches. But it turn out that frames for example did not require all the machinery of first order logic, but could be regarded as a fragment of it. The most important consequence of this fact is the recognition that the typical forms of reasoning used in structure-based representations could be accomplished by specialized reasoning techniques, without necessarily requiring first-order logic theorem provers.

Subsequently, research in the area of Description Logics began under the label *terminological systems*, to emphasize that the representation language was used to establish the basic terminology adopted in the modeled domain.

Successively, the emphasis was on the set of concept-forming constructs admitted in the language, giving rise to the name *concept languages*.

In more recent years, the terms *Description Logic*(DL) become more popular putting more emphasis to the properties of the underlying logical system. DL is a fragment of First Order Logic. The first order logic calculus has a complete property as shown by Gödel in 1930, although this calculus is not a decidable but a semi-decidable procedure. This means that we have program that takes in input $\langle \mathcal{T}, \phi \rangle$ and it stops if ϕ is a logic consequence of $\mathcal{T}$ ($\mathcal{T} \models \phi$) otherwise in the case of $\mathcal{T} \not\models \phi$ the programs could not stop. This was proven by Church and Turing in different way around 1936 [35] [124]. To be a fragment of FOL means that the we have a language that is expressive enough for our domain, that the logic consequence is a decision procedure and that this can be done using limited computer resource as defined by its complexity property.

The representation of DL is at the predicate level: no variables are presented in the formalism. A DL theory is divided into two parts: the definition of predicates: Terminological Box (TBox) and the assertion over constants: Assertion Box (Abox). The terminological parts describes the structural properties of the terms of the domain, while the assertional part depicts a particular configuration of a domain by introducing individuals and asserting their properties using the definitions in the terminology. Thus, statements in the TBox and in the ABox can be identified with formulae in first-order logic or, in some cases, a slight extension of it. A DL system not only stores terminologies and assertions, but also offers services that reason about them.

The following example shows a simple knowledge base written in natural language in which the terminological part describes the properties about the concepts involved in the domain while the assertional part states some properties of two particular individuals "chicca ", "dumbo".

Example 3.2.1.

Terminology

Quadruped ***is-an*** *Animal.*
Elephant ***is-a*** *Quadruped and it* ***has*** *the trunk.*
Dog ***is-a*** *Quadruped and it* ***not has*** *the trunk.*

Assertional

chicca ***is-a*** *Dog.*
dumbo ***is-an*** *Elephant.*

3.3 Syntax and Semantic of Description Logic

The linguistic structures for building descriptions is a characteristic of each DL system, and different systems are distinguished by their description languages. All description logic systems are based on a common family of languages, called concept languages, for describing structured classes of objects. The foundations of concept languages are *concepts* and *roles*: a concept represents a class of objects sharing some common characteristics, while a role represents a binary relation between objects or, in other words, attributes attached to objects. In addition to atomic concepts and roles (concept and role names), all DL systems allow their users to build complex descriptions of concepts and roles. The language is completely described by a formal syntax and a Tarsky-like semantics. A formal definition of the language is essential for knowledge bases characterisation, and for the definition of reasoning services. Description logics form a family of different logics, distinguished by the set of constructors they provide.

The very basic language denoted by the prefix $\mathcal{AL}$ is close to the expressivity of frame based representation systems. Concept descriptions in $\mathcal{AL}$ are formed according to the following syntax rule:

$$
\begin{array}{rcll}
C, D & \rightarrow & A| & \text{(atomic concept)} \\
& & \top| & \text{(universal concept)} \\
& & \bot| & \text{(bottom concept)} \\
& & \neg A| & \text{(atomic negation)} \\
& & C \sqcap D| & \text{(intersection)} \\
& & \forall R.C| & \text{(value restriction)} \\
& & \exists R.\top & \text{(limited existential quantification)}
\end{array}
$$

Note that, in $\mathcal{AL}$, negation can only be applied to atomic concepts, and only the top concept is allowed in the scope of an existential quantification over a role. For example, suppose that in $\mathcal{AL}$ $Engineer$ and $Manager$ are atomic concepts. Then $Engineer \sqcap Manager$ and $Engineer \sqcap \neg Manager$ are $\mathcal{AL}$ concepts describing, intuitively, those engineer that are manager, and those that are not manger. If, in addition, we suppose that $worksFor$ is an atomic role, we can form the concepts $Engineer \sqcap \exists worksFor.\top$ and $Engineer \sqcap \forall worksFor.ArmyProject$, denoting those engineer that works, and those engineers all of whose project, in which are involved, are army ones. Using the bottom concept, we can also describe those engineer that doesn't be enrolled for any project $Engineer \sqcap \forall worksFor.\bot$. In order to define a formal semantics of $\mathcal{AL}$ concepts, we consider interpretations $\mathcal{I}$ that consist of a non-empty set $\mathcal{D}^{\mathcal{I}}$ (the domain of the interpretation) and an *interpretation function*, which assigns a set $A^{\mathcal{I}} \subseteq \mathcal{D}^{\mathcal{I}}$ to every atomic concept A and a binary relation $R^{\mathcal{I}} \subseteq \mathcal{D}^{\mathcal{I}} \times \mathcal{D}^{\mathcal{I}}$ to every atomic role R.

The interpretation function is extended to concept descriptions by the following inductive definitions:

$$
\begin{array}{rcl}
\top^{\mathcal{I}} & = & \mathcal{D}^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
\neg A^{\mathcal{I}} & = & \mathcal{D}^{\mathcal{I}} \setminus A^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} & = & \{a \in \mathcal{D}^{\mathcal{I}} \mid \forall b.\ (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\} \\
(\exists R.\top)^{\mathcal{I}} & = & \{a \in \mathcal{D}^{\mathcal{I}} \mid \exists b.\ (a,b) \in R^{\mathcal{I}}\}
\end{array}
$$

We obtain more expressive languages if we add further constructors to $\mathcal{AL}$ such as

- *Concepts union* are written as $C \sqcap D$ whose semantic is :

  $$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}.$$

- *Full existential quantification* are written as $\exists R.C$ and interpreted as:

  $$(\exists R.C)^{\mathcal{I}}=\{a \in \Delta^{\mathcal{I}} \mid \exists b.\ (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\ \}$$

- *Number restriction* are written as $\leq nR$ (at-least restriction) and $\geq nR$ (at most restriction), where n ranges over the non negative integers. They are interpreted as:

  $$(\leq nR)^{\mathcal{I}}=\left\{a \in \Delta^{\mathcal{I}} \mid \|\{b(a,b) \in R^{\mathcal{I}}\}\| \leq n\right\}$$

  and

  $$(\geq nR)^{\mathcal{I}}=\left\{a \in \Delta^{\mathcal{I}} \mid \|\{b(a,b) \in R^{\mathcal{I}}\}\| \geq n\right\}$$

  $\|.\|$ being the cardinality of the set.

- *Negation* of arbitrary concept are written as $\neg C$ and interpreted as $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$

Using these constructs, we are able to write complex description of concepts and to describe classes of object. In particular we can have *terminological axioms*, which make statements about how concepts or roles are related to each other. If the definition in a terminology contain cycles, we may adopt fix-point semantics to make them unequivocal [18].

In the most of general case the terminological axioms have the form:

$$C \sqsubseteq D \text{ or } R \sqsubseteq S$$
$$C \equiv D \text{ or } R \equiv S$$

C, D being concepts and R, S roles. The first axioms are called *inclusion* and the second one are called *equivalent*. The semantic of those axioms is defined in term of an interpretation $\mathcal{I}$ that satisfies an inclusion $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ or an equality iff $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$, the same for the roles.

The assertion about individuals are defined in the following way. Given a, b, c individuals, one can make assertion using the concept C as $C(a)$ and role R as $R(a, b)$. By the first kind, called *concept assertions*, one states that a belongs to (the interpretation of) C, by the second kind, called role assertions, one states that c is a filler of the role R for b. In a simplified view, an ABox can be seen as an instance of a relational database with only unary or binary relations. However, the semantic of a database and Abox is different as we will see in the following, and in particular due to the difference between "closed-world semantics"of classical databases and "open-world semantics"of ABoxes.

In the definition of ABox semantic, we assume that distinct individual names denote distinct objects. These hypothesis is called *unique name assumption (UNA)*, that is, if a, b are distinct names, then $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. The semantic of ABox is defined in term of an interpretation $\mathcal{I}$: the concept assertion $C(a)$ is satisfied iff $a^{\mathcal{I}} \subseteq C^{\mathcal{I}}$, and it satisfies the role assertion $R(a, b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \subseteq R^{\mathcal{I}}$. An interpretation satisfies the ABox A if it satisfies each assertion in A. In this case we say that $\mathcal{I}$ is a *model* of the assertion ϕ or of the ABox. Finally, $\mathcal{I}$ satisfies an assertion ϕ or an ABox $\mathcal{A}$ with respect to a TBox $\mathcal{T}$ if in addition to being a model of ϕ or of $\mathcal{A}$, it is a model of $\mathcal{T}$. Thus, a model of $\mathcal{A}$ and $\mathcal{T}$ is an abstraction of a concrete world where the concepts are interpreted as subsets of the domain as required by the TBox and where the membership of the individuals to concepts and their relationships with one another in terms of roles respect the assertions in the ABox. Now we are in a position to translate into DL the simple example introduced in the previous section 3.2.1:

Example 3.3.1.

Terminology

$Quadruped \sqsubseteq Animal.$

$Elephant \sqsubseteq Quadruped \sqcap \exists Has.Trunk.$

$Dog \sqsubseteq Quadruped \sqcap \neg(\exists Has.Trunk).$

Assertional

*Dog(*chicca*)* .

*Elephant(*dumbo*)*.

In the following table some notation used to identify different family are described. For more details, please refer to [18] :

DL Name	expressivity
$\mathcal{AL}$	$(C \sqsubseteq D)$,$(C \equiv D)$,$\top$,$C \sqcap D$, $\forall R.C$,
	$\exists R$, assertion as $C(a)$, $R(a,b)$, $a = b$,$a \neq b$.
$\mathcal{ALC}$	$\mathcal{AL}$,$\bot$,$\neg C$, $C \sqcup D$, $\exists R.C$.
$\mathcal{S}$	$\mathcal{ALC}$, $Tra(R)$(transitive role).
$\mathcal{H}$	$R \sqsubseteq S$, $R \equiv S$.
$\mathcal{R}$	$\mathcal{H}$, property disjunction, global reflexivity,
	asymmetric and non reflexivity property, axioms on property based on chain,
	restriction on local reflexivity, negative assertion on role.
$\mathcal{O}$	nominals (one-of).
$\mathcal{I}$	inverse property R^-.
$\mathcal{F}$	functionalty on property $Fun(R)$ or $\leq 1R$.
$\mathcal{N}$	restriction on property non qualified $\leq nR$, $\geq nR$.
$\mathcal{Q}$	restriction on property qualified $\leq nR.C$, $\geq nR.C$.
D_n o D^+	attributes with value on concrete domain.

The works on tractable description logic (also known as DL-lite) are very interesting. In particular, in [29], the authors specifically capture the basic property of knowledge representation languages while keeping low complexity of reasoning, not only in terms of subsumption between concepts and checking satisfiability but also in terms of query answering using conjunctive queries (note that, DL reasoning tasks are polynomial w.r.t. TBox size, and the query answering is polynomial w.r.t. ABox size). There are also some interesting extensions of the description logics, such as Fuzzy Description Logic [119], [120], [118] and Probalistic Description Logic [84], [37].

3.4 Ontology

The terms "ontology"comes from the field of philosophy that is concerned with the ”study of being” or ”existence”, in this case philosophers also use the uppercase "Ontology". In philosophy, one can talk about an ontology as a theory of the nature of existence (e.g., Aristotle’s ontology offers primitive categories, such as substance and quality, which were presumed to account for "All That Is"). In computer and information science, ontology was been introduced by early Artificial Intelligence (AI) researchers who want design computational models that enable certain kinds of automated reasoning applying some procedures coming from mathematical logic. In the 1980’s the AI community came to use the term ontology to refer to both a theory of a modeled world and a component of knowledge systems. In the early 1990’s, an effort to create interoperability standards identified a technology stack that called out the ontology layer as a standard component of knowledge systems. Many papers were published on ontology topics from the once that are more seminal and foundational to the ones that used ontology for different purposes on different kind of data. From a general point of view, the subject of *ontology* is to study the *categories* of things that exist or may exist in some domain. The product of such a study, called an *ontology*, is a catalog of the types of things that are assumed to exist in a domain of interest $\mathbf{D}$ from a prospective of an agent, i.e. a person, who uses a language $\mathbf{L}$ for the purpose of talking about $\mathbf{D}$. The types in the ontology represent the predicate, word sense, or concept and relation types of the language $\mathbf{L}$. There are a relationships between languages and ontology. An uninterpreted logics , such as predicate calculus or semantic networks, are *ontological neutral*. It imposes no constraint on the subject matter or the way the subject may be characterized. By itself logic, says nothing about anything, but the combination of logic with an ontology provides a language that can express a relationship about the entities in the domain of interest. Infact in the context of computer and information sciences, an ontology defines a set of representational primitives with which is possible to model a domain of knowledge or a discourse.

In this framework, an ontology system is quite different from a database system, that is at the moment the most important solution for data management. Ontology can be viewed as a level of abstraction of data models, analogous to hierarchical and relational models, but intended for modeling knowledge about individuals, their attributes, and their relationships to other individuals.

Ontologies are typically specified in languages that allow abstraction away from data structures and implementation strategies; in practice, the languages of ontologies are closer in expressive power to first-order logic than languages used to model databases.

For instance, a conventional database model may represent the identity of individuals using a primary key that assigns a unique identifier to each individual. However, the primary key identifier is an artifact of the modeling process and does not denote something in the domain. The ontology designer are able to state semantic constraints without forcing a particular encoding strategy. For example, in typical ontology formalisms one would be able to say that an individual was a member of class or has some attribute value without referring to any implementation patterns such as the use of primary key identifiers. Similarly, in an ontology one might represent constraints that hold across relations in a simple declaration (A is a subclass of B), which might be encoded as a join on foreign keys in the relational model.

For this reason, ontologies are said to be at the semantic level, whereas database schema are models of data at the logical or physical level. Due to their independence from lower level data models, ontologies are used for integrating heterogeneous databases, enabling interoperability among disparate systems, and specifying interfaces to independent, knowledge-based services.

In the literature there are several different definition of ontology.

The first one was reported in the paper [56] and detailed in [59] in which an ontology is considered as an "explicit specification of a conceptualization"which is, in turn, "the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold among them. ". Gruber claims that "while the terms specification and conceptualization have caused much debate, the essential points of this definition of ontology are reported in the following:

- an ontology defines (specifies) the concepts, relationships, and other distinctions that are relevant for modeling a domain.
- the specification takes the form of the definitions of representational vocabulary (classes,

relations, and so forth), which provide meanings for the vocabulary and formal constraints on its coherent use."

Gruber stresses the conceptual nature of the ontology as a theory that can be used to represent relevant notion about modelling a domain. Domain that is classified in terms of concepts, relationships and constraint on them. Nothing is said about what we mean for the conceptualization of our domain. One of the problem of the knowledge representation is that it is not clear *what* is the knowledge to represent and *how* this could be aware from any context. This is a key point because we know that the knowledge modeling is an expensive operation and we would assure that the final result could be shared. This is possible only if it is valid *for all the users*.

We note that the very task of representation (i.e. modelling) is left to the user, i.e AI researchers focus more on the nature of reasoning than in the nature of the real world. This is source of an essential ontological promiscuity of AI: any agent creates its own ontology based on its usefulness for the task at hand. For example when someone asks for a services, what does she mean: the documents, or the content of the documents, the act in which the request is written? That's why Guarino, in order to define the ontology, starts from a notion of *conceptualization*. This notion was introduced in order to free the definition of ontology by a particular language, vocabulary or state of affairs. For example two ontologies can be different in the vocabulary used (using English or Italian words, for instance) while sharing the same conceptualization. According to Guarino, the conceptualization is a set of conceptual relation within a given domain. The domain is a structure made of a given domain and a set of maximal states of affairs of such domain (also called possible worlds). For instance, a domain may be a set of blocks on a table (a given domain) and with the set of all possible spatial arrangements of these blocks (state of affairs). The conceptual relation is a function that relates each state of affairs with a relation that we could built in the domain. I note that conceptual relation is different from the notion of relation. The first is related to an intensional state, instead the relation means the extensional level and refers to a fixed world structure. Then he introduce the notion of *ontological commitment*. The main idea is to approximate the conceptualization with a language. When we use the language with a given vocabulary we define an interpretation function which is used to find the model for that language (which is an extensional interpretation of the language); something similar happens for the conceputalization: Guarino defines an interpretation function that is called *ontological commitment*. Now the problem is to link the models founded in the language with the onces founded in the ontological commitment;

these models are called the *intended models of the language w.r.t. the ontological commitment*. In this way the ontology becomes a set of axioms designed in a way such that the set of its models approximates as best as possible the set of intended models of a given language according to the ontological commitment. In conclusion the following is the well known definition reported in the paper [60]:

"An ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models."

3.4.1 Ontology Level of Knowledge Representation

In [60] and [61] Guarino, instead, claims as the main component of the ontology studies the nature of the world that we would represent. In this way the ontology can be seen as the study of the organization and the nature of the world independently of the form of our knowledge about it, thus the ontology differs from the epistemology which claims that the knowledge consists of set of a propositions, whose formal structure is the source of new knowledge and the inferential aspect seems to be essential. In practice, Guarino claims that the *formal ontology* can be intended as the "theory of a priori distinctions:

- among the entities of the world (physical objects, events, regions, quantities of matter...);
- among the meta-level categories used to model the world (concepts, properties, qualities, states, roles, parts...)"

In this context, there is a real scientific problem of what the knowledge is rather than how to represent knowledge. Infact the first order logic is notoriously neutral with respect to ontological choices. This is one of its strengths, which shows the power of general ideas like completeness and soundness. However, ontological neutrality is not an advantage any more when applied to KR theories or languages: in this case, such formalisms should reflect the a priori structure of the real world, and the ontological choices made by the user. These aspects could not make explicit the intended models of a KR language, in order to facilitate large-scale knowledge integration and to limit the possibility to state something that is reasonable for the system but not reasonable in the real world. Also Lenat et all in his book [77] writes: "The majority of work in knowledge representation has been concerned with the technicalities of relating predicate calculus to other formalisms, and with the details of various schemes for default reasoning. There has been almost an aversion to addressing the problems that arise in actually representing large bodies of knowledge with content. The typical AI researcher seems to consider that task to be "just applications work". But there are deep, important issues that must be addressed [..]: What ontological categories would make up an adequate set for carving up the universe? How are they related? What are the important things most humans today know about solid objects? And so on. In short, we must bite the bullet.".

We can do that by giving a meta-level characterisation of the language primitives in terms of their ontological nature. Guarino proposes to introduce a new level in knowledge representation formalism that take into account the nature of the domain that we would we represent with an ontology. This level is called *ontological level* and it is between the epistemological and the conceptual level accoriding with the table 3.4.1:

Level	Primitives	Interpretation	Main Feature
Logical	Predicates, functions	Arbitrary	Formalization
Epistemological	Structuring Relation	Arbitrary	Structure
Ontological	Ontological Relations	Constrained	Meaning
Conceptual	Conceptual relations	Subjective	Conceptualization
Linguistic	Linguistic Terms	Subjective	Langauge Dependency

Tabella 3.1: Classification of KR formalism according to the kinds of primitives used

While the *logical level* deals with abstract predicates and the *conceptual level* with specific concepts, at the epistemological level the generic notion of a concept is introduced as a knowledge structuring primitive. The *epistemological level* introduces some structuring choices which may have cognitive and computational significance, and reflects a number of ontological commitments which accumulate in layers from the very beginning of a knowledge base development process in this way the language should be considered as different from the corresponding "flat"logical theory. At the *ontological level*, such ontological commitments associated to the language primitives are specified explicitly by suitably restricting the semantics of the primitives, or by introducing meaning postulates expressed in the language itself. In both cases, the goal is to restrict the number of possible interpretations, characterizing the meaning of the basic ontological categories used to describe the domain: the ontological level is therefore the level of meaning. Of course such a characterization will be in general incomplete, and the result will be an approximation of the set of intended models. At the conceptual level, primitives have a definite cognitive interpretation, corresponding to language-independent concepts like elementary actions or thematic roles. The skeleton of the domain structure is already given, independently of an explicit account of the underlying ontological assumptions. Within a certain application domain, the user is forced to express knowledge in the form of a specialisation of this skeleton. Finally, primitives at the *linguistic level* directly refer to verbs and nouns. I give an example of the use of ontological level. Suppose we have to represent an picture of a dog. At the logical level, a plausible representation may be $\exists x Picture(x) \cap Dog(x)$. At the epistemological level, supposing to adopt a description logic language, we have to decide what is a concept and what is (the filler of) a role. A good choice may be to consider Dog as a concept and $Picture$ as a filler of a $HasImage$ role. However, since the ontological assumptions underlying the meaning of concepts and roles are not made explicit, nothing prevents another user to adopt a different choice: for instance, both $Picture$ and Dog may be considered as concepts, with no role at all. If we want to improve knowledge sharing and reuse, we should be able to somehow restrict the set of possible choices. A possible solution is to go to the ontological level, where terms like role and concept have a formal, standard interpretation. Such an interpretation may forbid $Picture$ to be a concept according to the sense of "Picture"that we have in mind, making clear the ontological assumptions involved in this choice. Another solution may be to go directly to the conceptual level, with the introduction of a pre-defined set of concepts and roles we agree on, which may represent a "standard "for our mini-domain. However, our chances of getting such an agreement and controlling the disciplined development of applications depend in this case on the principles we have adopted for the definition of our basic ontological categories; therefore, the solution of the conceptual level (equivalent to the adoption of "off the shelf "ontologies) can be viewed as a successful one only if it builds on a well defined ontological level. Notice that the necessity of well-founded principles is much more relevant if we want to further specialize logical relations into categories like parts, qualities, properties, states and so on.

3.5 Knowledge Management

Typically the terms knowledge management is used to refer to a set of practices such as the identification, building, representation, distribution, of what it knows and how it knows it. For my aim, i use that term to refer to the methods, algorithms, procedures that have the purposes to manipulate the building knowledge and to produce new one. In particular in the next subsections, i will talk about two main services that make the DL-based knowledge systems an end-user systems: *querying services* and a more powerful systems: *reasoning services*.

3.5.1 Reasoning Services in DL-based Knowledge Systems

A knowledge representation system based on DL is able to perform specific kinds of reasoning. The aims of reasoning procedures are to deduce the implicit knowledge that can be made explicit through inferences. The different kinds of reasoning performed by a DL system are defined as logical inferences. The main inferences are for concepts reasoning, TBoxes reasoning and ABoxes reasoning, and finally TBoxes and ABoxes (together) reasoning. For the concepts the reasoning services are based on the satisfiability, subsumption, equivalence, disjointness problems.

Given a terminology box ($Tbox$), these problems are defined as:

- *Satisfiability*: A concept C is satisfiable with respect to $Tbox$ if there exists a model $\mathcal{I}$ of $Tbox$ such that $C^{\mathcal{I}}$ is nonempty. In this case we say also that $\mathcal{I}$ is a model of C.
- *Subsumption*: A concept C is subsumed by a concept D with respect to $Tbox$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $Tbox$.
- *Equivalence*: Two concepts C and D are equivalent with respect to $Tbox$ if $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $Tbox$.
- *Disjointness*: Two concepts C and D are disjoint with respect to $Tbox$ if $C^{\mathcal{I}} \cap D^{\mathcal{I}} =$ for for every model $\mathcal{I}$ of $Tbox$.

There are some propositions [18] that explain that all the previous problem could be reduce to the only subsumption problem or unsatisfiability one. In particular the knowledge system that used the in intersection operator "$\sqcap$"could use the first approach whereas the systems that allow the negation operator "$\neg$"could use also the second approach. In an $\mathcal{AL}$-language without full negation, subsumption and equivalence cannot be reduced to unsatisfiability in the simple way shown in those propositions and therefore these inferences may be of different complexity.

The $Tbox$ reasoning procedure mainly consist in *expansion* of the concepts according with their definition. This procedure, in the hypothesis of acyclic $Tbox$ could be used to check if the new concepts, obtained from their expansion, are unsatisfiable or satisfiable. Expanding concepts may be computationally costly, since in the worst case the size of the expansion $Tbox$ is exponential in the size of initial $Tbox$. A complexity analysis of the difficulty of reasoning with respect to TBoxes shows that the expansion of definitions is a source of complexity that cannot always be avoided.

In the case of $ABox$ the reasoning procedures consist in the checking if the $ABox$ *is consistent with respect to* a $TBox$. This means that there is an interpretation that is a model of both the $ABox$ and $Tbox$. Similarly as for concepts, checking the consistency of an $ABox$ with respect to an acyclic $TBox$ can be reduced to checking an expanded $ABox$[18]. Other kinds of reasoning services are the *instance check* and *retrieval problem*, the last one is related with the query processing. In The instance check problems means to check whether an assertion is entailed by an ABox. An assertion α is entailed by $ABox$ if every interpretation that satisfies the $ABox$, that is, every model of $ABox$, also satisfies α. If α is of the form $C(a)$, it is possible to reduce the instance check to the consistency problem for ABoxes. I note that also the concept satisfiability can be reduced to ABox consistency. There is some complexity problems in these procedures, in fact in [18] has shown that the ABox consistency can be reduced to concept satisfiability in languages with the "set"and the "fills"constructor. If these constructors are not available, however, then instance checking may be harder than the satisfiability and the subsumption problem.

3.6 Querying Services

The role of a querying and in particular of the query languages should be primarily the selection of data from a collection of them, rather than arithmetic computation on a given data. Typically, in the past the computational capability had to be separate from the retrieval capability. This separation of function was done because it possible to optimize both the computational and the query operations. In the literature ([15]), researchers focused their attention on two principles that a query language should obey. In essence, these principles state (1) that the value produced by a

query should be independent of the manner in which the data are actually stored in a database and (2) that a query language should treat data values as essentially uninterpreted objects, although certain properties, such as a linear ordering on certain domains can be built into the query language. The relational algebra and calculus of Codd [38] satisfy these principles and are often used as models of a query language. One purpose for which Codd introduced these languages was to provide a yardstick for measuring the relative power of query languages. As the two principles state, the main issue of querying processing is to retrieve data, they doesn't take into account that now days the data are distributed, sometimes on different schema, and their semantic is not always explicit. In this way, the intensional representation of relational model is not useful to express the complexity of the whole domain. If we would increase the powerful of model in knowledge terms, the query language could use it in order to retrieve the data taking into account what the data represents in a given domain and how the data are liked among them in conceptual terms.
In that case, the query process could use all the benefic coming from the knowledge model underlaying the system.
In this research fields, important results has been reached based on different knowledge framework for the heterogenous and distributed data sources with and in data and schema integration [79],[78]. Most of these approaches are based on some knowledge frameworks, that can overcome some problems as the incompleteness and inconsistency of the different data sources [76]. These techniques make the query processing more powerful taking. Those approaches also take into account some complexity issue based on the reasoning processing procedure analysis that could make worse the system efficiency [121], [12]. In the following section i will introduce some notation and queries languages for the DL-based knowledge systems. In particular i will focus on a special case of queries called *conjunctive queries*, and i will introduce a query language designed by the web semantic community, that are manly supported by different engines such as [10],[68], [34]. I will conclude this section with a brief introduction on the different assumptions that a query services have to take into account in a knowledge based system.

3.6.1 Querying Services in DL-based Knowledge Systems: The Conjunctive Queries

Starting from a general notion of query in First Order Logic (FOL), a query [14] is an open formula of FOL with equalities. I denote a query q as follows:

$$\{\overrightarrow{x}|\phi(\overrightarrow{x})\}$$

where $\phi(\overrightarrow{x})$ is a FOL formula with free variables $\overrightarrow{x}$. I call the size of $\overrightarrow{x}$ the arity of the query q. Given an interpretation $\mathcal{I}$, $q^{\mathcal{I}}$ is the set of tuples of domain elements that, when assigned to the free variables make the formula true in $\mathcal{I}$ A *boolean query* is a query that does not involve any free variable (i.e. it is a closed formula).
Given a boolean query q :

$$\{|\phi(\overrightarrow{x})\}$$

and an interpretation $\mathcal{I}$, $q^{\mathcal{I}}$ consists of the only empty tuple, i.e. the tuple of arity 0, in the case in which ϕ is true in $\mathcal{I}$, whereas $q^{\mathcal{I}}$ is obviously empty if ϕ is false in $\mathcal{I}$. In the conjunctive queries [14] the data model is asked to find set of values for which a certain pattern of data holds in the model. We shall see that the patterns can be described simply in terms of the existence of data that are connected to each other by equality of some their coordinates. There could be also some queries that cannot be expressed in this manner unless some form of disjunction or union is incorporated. Formally speaking a *conjunctive query* ($\mathcal{CQ}$) q is a query of the form :

$$\{\overrightarrow{x}|\exists\overrightarrow{y}.conj(\overrightarrow{x},\overrightarrow{y})\}$$

where $conj(\overrightarrow{x},\overrightarrow{y})$ is a conjunction of atoms and equalities, with free variables $\overrightarrow{x}$,$\overrightarrow{y}$. A *union of conjunctive query* ($\mathcal{UCQ}$) q is a query of the form :

$$\{\overrightarrow{x}|\bigcup_{i=1\cdots n}(\exists\overrightarrow{y}_i.conj_i(\overrightarrow{x},\overrightarrow{y}_i))\}$$

where each $conj_i(\overrightarrow{x},\overrightarrow{y})$ is, as before, a conjunction of atoms and equalities with free variables $\overrightarrow{x}$ and $\overrightarrow{y}_i$. Obviously, conjunctive queries are a subset of union of conjunctive queries.

Another kind of notation for $\mathcal{CO}$ is the standard datalog one :

$$q(\overrightarrow{x}') \leftarrow conj'(\overrightarrow{x}', \overrightarrow{y}')$$

where the $conj'(\overrightarrow{x}', \overrightarrow{y}')$ is the list of atoms in $conj(\overrightarrow{x}, \overrightarrow{y})$ obtained after having equated the variable $\overrightarrow{x}, \overrightarrow{y}$ according to the equalities in $conj(\overrightarrow{x}, \overrightarrow{y})$.
I call $q(\overrightarrow{x}')$ the head of q and $conj'(\overrightarrow{x}', \overrightarrow{y}')$ the body. Moreover, we call the variable in $\overrightarrow{x}'$ the *distinguished variable* of q and those in $\overrightarrow{y}'$ the non-distinguished variables. The datalog notation could be easy extended for $\mathcal{COQ}$.
If we consider only the Description Logic Knowledge Base DL $\mathcal{K}$, we have differently from the FOL $\mathcal{K}$, only atoms in the forms of $A(z)$ or $P(z_1, z_2)$, where A and P are an atomic concept and atomic role respectively, for *conjunctive queries*. The z, z_1, z_2 are either constants in $\mathcal{K}$ or variables. The same happen for the $\mathcal{UCO}$. We note that the conjunctive query are the formal framework to express the common SQL pattern query: *(Select . . . From . . . Where . . .)*.

3.6.2 The Sparql query language

The query languages for Semantic Web ontologies can be classified under two categories: RDF-based query languages and DL-based query languages.

RDF-based query languages, such as RDQL3, SeRQL4 and the W3C recommendation SPARQL are based on the notion of RDF triple patterns and their semantics is based on matching triples with RDF graphs. RDF stands for Resource Description Framework (section 3.8 will provide further details about it).

It is harder to provide a semantics for these queries language under description logic semantics because RDF representation mixes the syntax of the language with its assertions. The triple patterns in a query do not necessarily map to well-formed DL constructs. DL-based query languages such as the ASK queries of DIG protocol [110] or nRQL queries of Racer- Pro system [62], on the other hand, have well-defined semantics based on the DL model theory. However, DIG queries are limited to atomic (TBox or RBox or ABox) queries whereas nRQL supports only conjunctive ABox queries.

Despite the previous observation the current query engines are focused on Sparql query language, that it is become the standard de facto for the querying procedures in the knowledge base systems that use the web semantic community language representation 3.8. Sparql is query language defined in [48] for querying RDF graphs 3.8. This query language is based on the notation of graph pattern. The objects inside a RDF triple could be seen as a variable of a generic graph pattern. For example, the RDF graph pattern with variable $?x$ $?x\ foaf:name\ "Antonio"$ could contain the triples $\_{:}b\ foaf:name\ "Antonio"$, the $\_{:}b$ is a bnode, that is special resources in RDF [8].

Query processing in Sparql is so defined as to find all the assignments for the variables that make the pattern a logical consequence of the RDF Dataset. The simple entailment could be considered in that case as *subgraph matching problem*. The subgraph matching is a conjunctive query answering problem. For example, if we store RDF data in an relational database, the query processing for Sparql becomes nothing more than conjunctive queries using a single ternary predicate (e.g. triple($?x,\ foaf:name, "Antonio"$)). The results from different graph patterns can be combined using an algebra such as union of answer sets, left outer join, filtering based on XQuery operators and etc. I have to precise that the bnodes could be source of problems in answer set such as the null values in SQL.

3.6.3 Data Base Assumption vs Knowledge Base Assumption

The Data Base assumption, also known as *Close World Assumption (CWA)*, consider that at run-time the data satisfy the schema and therefor the schema is not used. Queries allow for complex navigation paths in the data. In this case the query answering process is more related to a *query evaluation* process on the stored data. For this reason, this kind of process is computationally easy. Let us consider this example 1 related to the UML diagram in figure 3.1

Example 1. *For each concept/ relationship we have a (complete) table in DB.*
DataBase:
$Employee = \{antonio, picus, letizia\}$
$Manger = \{antonio, letizia, angelo\}$
$Project = \{uninaP, polimiP\}$
$worksFor = \{(antonio, uninaP), (antonio, polimiP), (letizia, polimiP)\}$
Query:
$q(x) \leftarrow Manger(x), Project(y), worksFor(x, y)$
Answer:
$\{antonio, letizia\}$

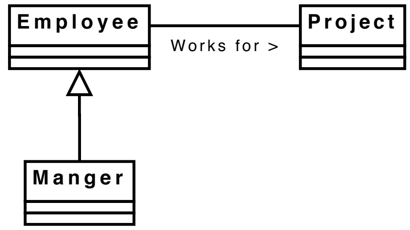


Figura 3.1: A UML Diagram for a simple Knowledge Base

In the case of CWA, we have to make this assumption: everything that is explicitly asserted on the Assertion Box (tables of DB) is true. Instead everything, that is explicitly not asserted, is false. This assumption claims the *complete knowledge about the domain*. It is as if in database model when store a data we express also the complement of this kind of fact, we can do this only if we have a complete knowledge about the domain. This assumption make a use of the *negation as failure* process [14] as a source of new knowledge.

The knowledge assumption, also known as *(*Open World Assumption (OWA)), claims that the ontology or knowledge base imposes some constraints on the actual data. That data could be incomplete or inconsistent w.r.t such constraints. The query answering process has to take into account intensional information to overcome incompleteness or inconsistency. Now the size of data is not considered critical comparable to the size of intensional information. For this reason the query answering process is more related with a *logical inference* process which is computationally more costly. Let us consider an example related to 1:

Example 2. *In that case we have complete table in database only for some conepts/relationship.*
DataBase:
$Manger = \{antonio, picus\}$
$Project = \{uninaP, polimiP\}$
$worksFor = \{(antonio, uninaP), (antonio, polimiP), (letizia, polimiP)\}$
Query:
$q(x) \leftarrow Employee(x)$
Rewritten Query: $q(x) \leftarrow Employee(x) \vee Manger(x) \vee worksFor(x, -)$
Answer:
$\{antonio, picus, letizia\}$

In the OWA, we have a partial knowledge about the domain, we know that same assertions are true, some are false, some are uncertain. These differed assumptions entail not only a different modality of query processing but also they could be give a different answer when we use in the query the negation operator [14]. Which is the better semantic is a problem related to how the full knowledge is represented in our data and the level of our knowledge about the considered domain. There are some models that could take into account both the assumptions, these models use epistemic modal operator[43],[44]. This solution could be used in order to divide the knowledge about the domain in a complete one and in a partial one. This consideration permits us to consider a complete knowledge about some concepts and relationships also in a open world contexts.

I also note that the constraints in the two different assumption have two different meanings. The constraints in CWA refer to integrity constraint that are used to prevent incorrect values from being asserted in a model. In fact in CWA we have a single *model* that contains only the facts asserted. The constraints instead in the OWA refer to logical axioms, i.e restrictions, property domain/range. In fact in the OWA we have multiple possible model that can satisfy the axioms.

In this dissertation, we will mainly work with the OWA assumption, first because it is the underlying semantic of the Ontology Web Language; second we have the need to do logic inference rather than query evaluation; last we will take into account the constraints expressed in the ontology at run time. In addition, OWA is also close to the *Semantic Web Philosophy* in which the knowledge is intentionally under-specified: this allows to reuse and extend the previous knowledge. If the target is an application. it is desirable to turn on the closed world assumption in order to reap the benefits of *negation as failure procedure*. For example in the following table 3.2 i report some context in which it is useful to turn on the both assumptions.

Open World Assumption	Closed World Assumption
Does Antonio Knows Nino's house Address?	Is there a train from Napoli to Roma today?
What are the potential side-effects of drug X ?	Find me drugs that are not licensed for X ?

Tabella 3.2: Questions in OWA or CWA

3.7 Web Semantic Approach for Knowledge Representation

The World Wide Web (WWW) has changed the way people communicate with each other, how information is disseminated and retrieved, and how business is conducted. In 1999 the main protagonist of that revolution Tim Berners-Lee in [20] wrote :

"I have a dream for the Web, in which computers become capable of analyzing all the data on the Web : the content, links, and transactions between people and computers. A *Semantic Web*, which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The *intelligent agents* people have touted for ages will finally materialize."

The term Semantic Web become a challenge for more researchers that try to built techniques that have the aim to dramatically improve the current WWW and its use according with that vision. The roadmap for this purpose was hard and the Web, the users, has shown in last year a new way to publish and use the information. These changes in the Web-shape become to be reported in literature and in the community with names such as *WEB 2.0* and now days *WEB 3.0*. The main change was been in the way of thinking the web. Initially, the Web was a container of hyper-document and the main purpose was to retrieve the information. This was the great revolution of the Web and it helped the people to navigate the data according with different "directions"besides the sequential way in which the people were used to. These "directions"enable the web-user to retrieve documents and to discover new one in all the corners of the net and therefore across the world. In order to help those purposes, the era of search engines started, and different kinds of algorithms were proposed to improve the efficiency and effectiveness of those systems.

In last years the Web becomes a social place, where the people can meet their self, buy something, search friends, organize works, travels and promote their businesses. In this way the data become to be more complex such as the multimedia ones (image and video) and a lot of tools were done in order to insert those data in web space. In spite of the growth of those data, the main obstacle to providing better support to web users is that, at present, the meaning of web content is not machine-understanding. Of course, there are tools that can retrieve data and process them to improve user navigation on them. But when it comes to interpreting sentences and extracting useful information for users, the capabilities of current software are still very limited. It is simply difficult to distinguish the meaning of those data.

The Semantic Web in the original idea, has as main goal to facilitate this aim. This simple idea, however, remains largely unrealized.

Nowadays, what we can see is a Web of complex data, and the people are looking for an integrated access to the whole information. That evolution make the Web as a whole more like to a large database or spreadsheet, rather than just a set of linked documents. Second, Web will be accessible from a growing diversity of networks (wireless, wireline, satellite, etc.) and will be available on a ever increasing number of different types of devices.

Finally, in a related trend, Web applications will become a more and more ubiquitous throughout our human environment, with walls, automobile dashboards, refrigerator doors all serving as displays giving us a window onto the Web. For example the user would recruit the right data to a particular use context, opening a calendar and seeing business meetings, travel arrangements, photographs, and financial transactions appropriately placed on a time line. All of these requests and operations ask for a new kinds of data models which could have the aim to offer an integrate view on the heterogenous and distritbuted data sources. The Integration of data means that we have to understand the meaning of the schema those data fit and we have also to understand how and when to retrieve those data in which way we have to present them [93].

The good news is that a number of technical innovations (RDF which is to data what HTML is to documents, and the Web Ontology Language (OWL) which allows us to express how data sources connect together), along with more openness in information sharing practices, are moving the World Wide Web toward what we have called the Semantic Web. I give more detail in the following section about RDF and OWL.

Progress toward better data integration will happen through use of the key piece of technology that made the World Wide Web so successful: the link.

The power of the Web today, including the ability to find the pages we are looking for, derives from the fact that documents are put on the Web in standard form, and then linked together. The Semantic Web will enable better data integration by allowing everyone who puts individual items of data on the Web to link them with other pieces of data using standard formats. This technology was developed in order to help the vision of web as web of data and helped the machine and user to have integrated model of those data where also the meaning was made explicit.

Ontologies, rulues and inference mechanisms have a great role on that revolution. But most of the requirements in the production of ontology in the context of Web become too much strictly and they loose some of their functionality to define the meaning of the data inside a fixed domain.

For example from one hand we have the US National Center for Biotechnology Information that have build the "Oncology Metathesaurus"[130], this project involved more than 8 people supporting full time and it consists of more than 50,000 classes. It was written according the lows of owl dl and its consistency was proved.

From the other hand we have "Friend Of Friend (FOF)"[3], it is non more than 30 classes it violates DL rules (undecidable) and it is used inconsistently.

In the first case we have high use in medical community but not much data on the web in the second one FOAF more than 60 milion of people uses it and it is used by a number of large providers, becoming the standard the facto of open social networking.

With this example i would claim that the building of ontology have to take into account the target application or domain, because the modelling phase is very expensive and the return on investment must be very high and then the results are not the ones attended.

I can conclude that in web scenario most of the constraint imposed by a well defined theory could compromise the development of a knowledge, what we would is to share and integrate data as much as possible.

Instead when we have a domain application most of theoretical results could helped our goal and the use of web semantic technologies (essentialist the languages recommendation) can be used to spread the result data in web environment for their use across the world. In that wa we could carry out not only a sharing of experience but also something that is called *collective intelligence* [57] that means that you can learn from the collective knowledge (the knowledge produced by web user for example).

3.8 W3C Languages Recommendation for Knowledge Representation

In 1997, the W3C defined the first Resource Description Framework specification. RDF provides a simple but powerful triple-based representation language for Universal Resource Identifiers (URIs). It became a W3C recommendation in 1999 [8]. With URIs we can identify resources and so are central to the Semantic Web enterprise. Using a global naming convention (however arbitrary the syntax), it provides the global network effects that drive the WebŠs benefits. URIs have global scope and are interpreted consistently across contexts. Associating a URI with a resource means that anyone can link to it, refer to it, or retrieve a representation of it. Given the Semantic Webś aims, we want to reason about relationships. URIs provide the grounding for both our objects and relations. URIs identify resources and so are central to the Semantic Web enterprise.3 Using a global naming convention (however arbitrary the syntax) provides the global network effects that drive the Webś benefits. URIs have global scope and are interpreted consistently across contexts. Associating a URI with a resource means that anyone can link to it, refer to it, or retrieve a representation of it. Given the Semantic Webś aims, we want to reason about relationships. URIs provide the grounding for both our objects and relations.

With RDF we can create an RDF graph of nodes and arcs, an URI reference used as a graph node identifies what the node represents; a URI used as a predicate identifies a relationship between the things identified by the connected nodes. RDF also provides an XML-based syntax called RDF/XML for recording and exchanging graphs. However, there are alternative forms that are easier to interpret; for example, see the N3 notation. RDF Schema became a recommendation in February 2004. RDFS took the basic RDF specification and extended it to support the expression of structured vocabularies. It has provided a minimal ontology representation language that the research community has adopted fairly widely

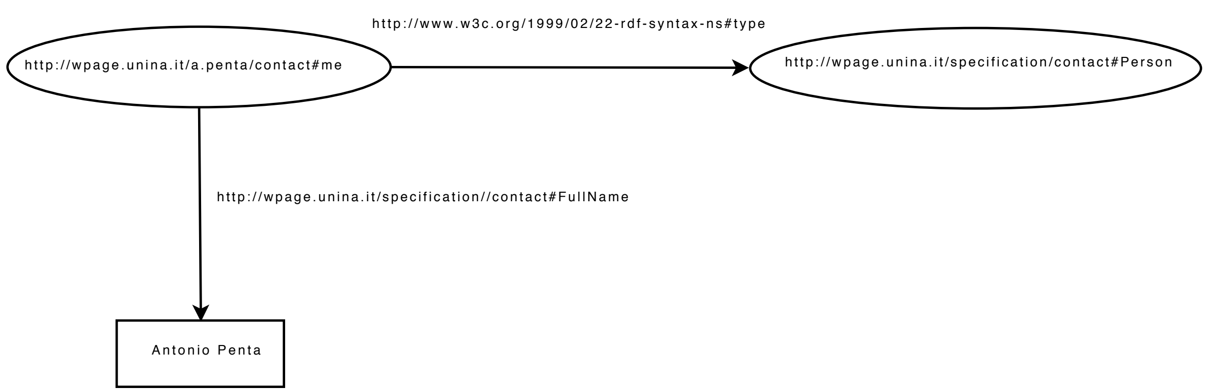


Figura 3.2: RDF Example

This are an example of how the graph structure in 3.2 is serialized in XML/RDF.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:contact="http://wpage.unina.it/a.penta/specification#">
<contact:Person rdf:about="http://wpage.unina.it/a.penta/contact#me">
<contact:FullName>Antonio Penta</contact:FullName>
</contact:Person>
</rdf:RDF>
```

For those who required greater expressivity in their object and relation descriptions, the Web Ontology Language (OWL) specification [7] integrated several efforts. The W3C recommendation presents three versions of OWL (full/dl/lite), depending on the degree of expressive power required. OWLŠs core idea is to enable efficient representation of ontologies that are also amenable to decision procedures. It checks an ontology to see whether itŠs logically consistent or to determine whether a particular concept falls within the ontology. OWL uses the linking provided by RDF to allow ontologies to be distributed across systems. Ontologies can become distributed, as OWL allows ontologies to refer to terms in other ontologies. In this way OWL is specifically engineered for the Web and Semantic Web.

In the tables 3.3 and 3.4, a mapping between some description logics constructs and owl full definitions is reported:

OWL	DL
$oneOf$	$\{a_1 \ldots a_n\}$
$someValuesFrom$	$\exists R.C$
$allValuesFrom$	$\forall R.C$
$hasValue$	$\forall R.a$
$maxCardinality$	$\leq_n R.C$
$minCardinality$	$\geq_n R.C$
$cardinality$	$\doteq_n R.C$
$intersctionOf$	$C_1 \sqcap \ldots \sqcap C_n$
$unionOf$	$C_1 \sqcup \ldots \sqcup C_n$
$complementOf$	$\neg C$
$subClassOf$	$C \sqsubseteq D$
$equivalentClass$	$C_1 \equiv \ldots \equiv C_n$
$disjointWith$	$C \sqcap D \equiv \perp$

Tabella 3.3: Mapping 1 between OWL and DL

OWL	DL
$domain$	$\top \sqsubseteq \forall R^{-}.D$
$range$	$\top \sqsubseteq \forall R.C$
$subPropertyOf$	$R_1 \sqsubseteq R_2$
$equivalentProperty$	$R_1 \equiv \ldots \equiv R_n$
$inverseOf$	$R_1 \equiv R_2^{-}$
$functionalProperty$	$\top \sqsubseteq \leq_1 R$
$invFunctionalProperty$	$\top \sqsubseteq \leq_1 R^{-}$
$symmetricProperty$	$R \equiv R^{-}$
$transitiveProperty$	$Tr(R)$
$someAs$	$a_1 \doteq \ldots \doteq a_n$
$differentFrom$	$a_i \neq a_j$
$allDiffferntFrom$	$\neq (a_1 \ldots a_n)$

Tabella 3.4: Mapping 2 between OWL and DL

Capitolo 4

Related Work

Related Work

4.1 Multimedia Semantic Management

The usefulness of multimedia applications is largely determined by the accessibility of the content, so new challenges are emerging in terms of storing, transmitting, personalising, querying, indexing and retrieval of the multimedia content. Some examples of such challenges include access by business users to multimedia content needed for their work, access by consumers to entertainment content in their home or when mobile, and sharing of content by both professional and private content owners. Clearly, a description and deeper understanding of the information at the semantic level is required in order to efficiently meet the requirements resulting from these challenges. These challenges have as key-point what is called the *semantic gap*.

In fact the the low-level descriptors, metrics and segmentation tools are fundamental building blocks of any multimedia content manipulation technique, they evidently fail to fully capture, by themselves, the semantics of the audiovisual medium; achieving the latter is a prerequisite for reaching the desired level of efficiency in content manipulation and retrieval.

As consequence, there is a need for knowledge representation and processing in many multimedia applications or parts of the whole multimedia value chain.

This has led to an increasing convergence of research in the multimedia and knowledge domains, which we refer to as *semantic multimedia*.

Among the possible domain knowledge representations, the ontologies can be used for expressing multimedia content semantics so that annotation, automatic semantic analysis and further processing of the extracted semantic descriptions are allowed. For example, the main challenge in building a knowledge infrastructure for multimedia analysis and annotation is to link low-level multimedia properties such as spatio-temporal multimedia document structure and the semantic concepts in a clean, extensible, effective and efficient manner.

More specifically, a number of multimedia ontologies have been designed to serve one or more of these purposes [39], [16], [92] :

- Annotation e.g. labelling or tagging of multimedia content;
- Analysis e.g. ontology-driven semantic analysis of multimedia content, for downstream annotation;
- Retrieval e.g. context-based retrieval of images or video from large archives;
- Personalisation e.g. filtering and recommendation of multimedia content according to user preferences;
- Algorithms and processes control e.g. ontologies used to model multimedia processes and procedures;
- Reasoning, which can be applied in various cases such as retrieval and personalisation for creating autonomous content applications.

In this chapter a discussion about the state of art on the representation of multimedia data, and in particular on image and text data, will be presented. The chapter is organized into two main sections. In the first one (4.2), I will describe semantic approaches for image management, introducing (subsection 4.2) some systems proposed in the literature for those aims; in 4.2.2 the problems and proposed solutions about the use of MPEG7 standard for managing multimedia semantic is discussed; several considerations about web systems are also presented 4.2.3. In the second section 4.3 i will describe much more the state of art approach on knowledge extraction from text data and in section 4.3.3 i will consider the legal context in text domain.

4.2 Knowledge Image Management

In the last few years, several papers have been presented about multimedia systems based on knowledge models, image ontologies, fuzzy extension of ontology theories. In almost all the works, multimedia ontologies are effectively used to perform *semantic annotation* of the media content by manually associating the terms of the ontology with the individual elements of the image or of the video [105], [54], thus demonstrating that the use of ontologies can enhance classification precision and image retrieval performance. Instead of creating a new ontology from the scratch, other approaches [31] extend WordNet to image specific concepts, using the annotated image corpus as an intermediate step to compute similarity between example images and images in the image collection. For solving the uncertain reasoning problems, the theory of fuzzy ontologies is presented in several works, as an extension of the ontologies with crisp concepts as the papers [70] that presents a complete fuzzy framework for ontologies. In [100], the authors introduce a description logic framework for the interpretation of image contents. They use a very expressive description logic together with a rules level, typically used to describe the spatial relations among the objects. In their view the annotations describe "real-world"objects and events and they have not the goal to merely "classify "images and attach keywords but to construct a high-level interpretation of the content of a media object. In order to deal with a multiple interpretations of image semantic the introduce an interesting reasoning procedure called abduction. Infact, it is necessary, however, to keep in mind that each media object might consist of multiple modalities, each of which will be the basis of modality-specific interpretation results (ABoxes). In order to provide for an integrated representation of the interpretation of media objects as a whole, these modality-specific interpretation results must be appropriately integrated. A cornerstone of this integration process will be to determine which modality-specific names refer to the same domain object. They assume that the information extracted from a multimedia document through low-level analysis (e.g., image analysis) is formally encoded as a set of Abox assertions. For example, in the context of images for every object recognized in an image, a corresponding concept assertion is found in the assertional knowledge. Usually, the relations that can be extracted from an image are spatial relations holding among the objects in the image. These relations are also represented as role assertions. In order to construct a high-level interpretation of the content, the abduction process will extend the Abox with new concept and role assertions describing the content of the multimedia document at a higher level.In that procedure, better explained in [97],[106], they divide the results of image processing algorithm in two kind of assertions into bona fide assertions and assertions requiring fiats. And they propose an algorithm to understand how is the best set of assertion (Abox) that satisfy the terminological knowledge. They don't propose a complete knowledge system but instead are more focused on new kind of reasoning procedure, but they make strong hypothesis on what a computer vision system could detect. Nothing is said about the complexity of that abduction procedure. In [69] also the author prosed a DL framework for the semantic analysis of the image content, they design a novel multi-modality ontology model that integrates both the low-level image features and the highlevel text information to represent image contents for image retrieval. They construct an ontology in the canine domain that take into account the textual annotation, the image processing results and the domain ontology. A simple reasoning algorithm called matchmaking process is presented, but an first good experimentation is depicted with comparison with keyword base image retrieval engine as Google. In [111] the authors present a multimedia reasoning architecture using the fuzzy extension of expressive $\mathcal{SHIN}$, called f-$\mathcal{SHIN}$. In this approach, first a segmentation algorithm generates a set of over-segmented regions and a classification process is employed to assign those regions with semantic labels. A semantic-based refinement of the segmentation is follows and this information initializes the ABox of a fuzzy-knowledge that is used for multimedia reasoning. By reasoning in this context, they refer to the automatic derivation of high-level semantic annotations from low-level multimedia data (raw and/or preprocessed to acquire audiovisual or conceptual descriptions of varying abstraction levels) through the utilization of the provided (general, domain, structural, etc.) knowledge. In particular the the main reasoning services proposes are entailment and subsumption. Their approaches work manly on a labeling the different section of image with more accurate annotations. All of these previous approaches, that combine multimedia with logic based framework, have some connections with my work. Differently from those instead, i propose a formal definition of multimedia ontology, particularly suitable for capturing the complex semantics of images during several steps of the image analysis process. This is done without proposing any extension of the usual ontology theory and languages, but manage uncertainty implementing ternary properties by means of a reification process, thus taking advantages of the several existing reasoning systems and eventually a complete final system is presented. Grosky et all in [131] describes some of the problems and techniques that the research could be used when they deal with multimedia metadata. Some interesting view are the of subdivision of the multimedia information in :

- Content-independent metadata: data that is not directly concerned with content, but related to it. Examples are image format, authorś name, date, and location.

- Content-based metadata: Non-information-bearing metadata: data referring to low-level or intermediate-level features, such as colour, texture, shape, spatial relationships, and their various combinations. This information can easily be computed from the raw data.

- Information-bearing metadata: data referring to content semantics, concerned with relationships of entities appearing in multimedia documents to real-world entities, as well as data referring to the relationship of a particular multimedia document or sub-document to particular users.

The author note that information-bearing metadata, commonly referred to as semantic information, however, are not extracted directly from visual contents, but represent the relatively more important meanings of multimedia objects that are perceived by human beings. These conceptual aspects are more closely related to usersś preferences and subjectivity. Concepts may vary significantly in different circumstances. Low-level multimedia features are directly related to perceptual aspects of image content. Since it is usually easy to extract and represent these features and fairly convenient to design similarity measures by using the statistical properties of these features. Another noteworthy aspect is the subdivision of the different kinds of gap: *semantic, sensory, subjective* gap.

- *semantic gap* is the principal focus of multimedia retrieval research. A proper definition of semantic gap is given in Smeulders,Worring, Santini,Gupta, and Jain in [112]: "The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation."

- *sensory gap* exists in the multimedia world as the gap between an object and the machineŠs capability to capture and define that object. For example, a person in a picture with the side of his face exposed may not be recognized as a human because a human should have two eyes or a three-dimensional structure represented by a two dimensional image. The lack of resolution can also contribute to this sensory gap. It is possible that different low-level features or representations can be produced by the same object due to distance, partial occlusions, illumination, clutter, camera viewpoint, etc. To bridge the sensory gap, some form of contextual knowledge is required by the retrieval system. This knowledge may come while capturing the multimedia data or may be incorporated as part of domain knowledge. This contextual knowledge may be in the form of physical laws, laws about how objects behave and how people visualize the object.

- *subjective gap* is similar to the semantic gap; it refers to the lack of ability of a user to describe her needs (query) to a retrieval system. The subjective gap also exists due to the non-availability of any features which can define emotions, feelings, smell, touch, and other such features. If a user needs a picture of a sweet food item, there is no method to describe what "sweet"means. At this time, if the image database has text annotations, which includes these abstract features, a query may return some results depending on the quality and level of these annotations.

This could be useful to understand how works to work in order to help a user to express what she wants from a multimedia retrieval system. Therefore, it is important that the system itself try to reconstruct them from a userś browsing and querying history.

Some interesting notes are written by Santini in two papers [108], [107]. He poses some questions about the nature of meaning and similarity in the image databases. His main review is that we can not use the same approaches used in the relational database to give a semantic to an image. In fact in the relational database it is the schema that sets the semantic of the tuples. They become true assertions on the given relational schema, without those schema the data could be mapped in different possible meaning. This happens because the symbolic nature of the data used in relational systems could be described in compositional way. Symbols have a combinatorial syntax and semantics in which there is a distinction between structurally atomic and aggregate. Instead a relation between the image and its meaning is considerably more complicated than in symbolic systems. In the image the information is not encoded in symbols, but it is distributed on the whole set of pixels. In this way, the whole object can be used as symbol carrying information and it cannot be divided into meaningful fragments. The semantic of the image could not be defined only thought a system of signs or symbols. Those complex meanings and implications depend not only on *how* and *by whom* were produced, but also on why are they being searched. As consequence we could not based all the meaning of the image on a similarity concepts, that is inadequate to establish a sign relation between the image objects and its meaning. In his work he suggest to use different modality of interaction with image in which the text context surrounded the image and/or the user interaction and needed become the constraint for defining the relations of signs that are the bases of image meaning. I think that this is an important vision that a researcher could have when he deals with image semantic management but an interpretation of image could be view in different aspects. A set of labeling on the image content could be used to formulate different interpretations of the same image but a general knowledge (that could include contextual one) constraints these interpretations, and the first relation of sign could be produced in compositional way looking on the meaningful entities inside an image and how they are related. A second meaning of these signs could be produces according to a model that could be catch the useríntension through his interaction with a system, his level of knowledge, his experiences, and his social interaction all this aspect could be deduced may be from his web behavior.

4.2.1 Image Retrieval Systems

At the moment, the system for Knowledge Image Management are strictly related to Content Base Information Retrieval (CBIR) systems. In that kind of systems, the main goal is to retrieve content (image, video, text) in more efficient and effective way. In the CBIR framework, a significant amount of research has been done in the communities of Computer Vision [51, 32] and Information Retrieval [83]. Traditionally, CBIR has the purpose of finding images relevant to the users' information needs from image databases, principally on the basis of low-level image global descriptors (color, texture and shape features) for which automatic extraction methods are available. The architecture of these systems is composed of the following points:

- define the representation of information, for example, the space colors used and also determine the components to be considered;
- extraction of features, most of the times they are multidimensional vectors, that require dimensional scaling algorithms;
- storing images: determine the most appropriate methods for compact storage of large collections of images;
- defining the metrics for comparison and more sophisticated search methods for the image retrieve .

In the past decade, systems for retrieval by visual content have been presented in the literature, proposing visual features that, together with similarity measures, could provide an effective support to image retrieval [113]; different systems like *PICASSO* [42], *SIMPLIcity* [127], and *Blobworld* [30] have been developed, using both global and local descriptors [63, 25]. In the last years in literature, some systems were developed in order to improve the capabilities offered by simple CBIR systems with the addition of services reasoning. The best in terms of results are:

- ALIPR (Automatic Linguistic Indexing of Pictures - Real Time) [128]: it is a system that can automatically annotate entire collections of photographs and ALIPR uses distributions of color and texture to characterize images and it infers new kinds of annotation looking at the text of the images that have similar visual content, but the humans annotated in a different way;

- AceMedia [1]: In addition to the information medium level obtained starting from the colors, shapes and textures are particularly used the spatial relationships to define the regions and their semantic meaning.

- Cortina [49] in which the authors announced to be the first system to break the 1 Million image barrier and more. In particular, the authors propose a similarity search in a combined feature space that includes color and texture; successive classifiers automatically classify image content using these descriptors. In addition, several efforts have been devoted to annotate pictures exploiting users interaction and trying to capture their specific semantic by the concept of *relevance feedback*.

The proposed system architecture is similar to the ones presented in the literature, but contains innovative automatic technique annotations and reduces the semantic-gap between high-level concepts and multimedia concepts associated with individual images.

Some interesting web based systems are :

- Google Image Search [4] and Yahoo Image Search [11] that are keyword base image search engine, where the keywords are the ones related within the web pages that host the images. Their is nothing in their searches related with the nature of the image.

- Riya [9] that it look inside the image, not only at the text around it and use color, shape and texture of images. and Flicker that is the most used system for image retrieval on .

- Flickr [2], [80], [74] which search capability is also keywords based but that keyword, called tag, are produced by users and are focused on image contents. In this case the search is based on what is called *collective knowledge* [58], and this search could be improved by the social relationship that this system builds.

4.2.2 MPEG7

MPEG-7 [6],[27] is conceived for describing multimedia content data. MPEG-7 is used to store meta-data about multimedia in order to describe particular events. MPEG7 has standardized tools for describing different aspect of multimedia at different levels of abstraction. It has proposed as an instrument to improve the current multimedia representation and applications. The MPEG-7 framework consists of Descriptors(Ds) and Descriptor Schemes (DSs) that represent features for multimedia, and more complex structures grouping Ds and DSs, respectively. In particular, the MPEG-7 standard includes tags that describe visual features (e.g., color), audio features (e.g., timbre), structure (e.g., moving regions and video segments), semantic (e.g., object and events), management (e.g., creator and format), collection organization (e.g., collections and models), summaries (e.g., hierarchies of key frames) and, even, user preferences (e.g., for search) of multimedia. In this way the standard includes descriptions of low-level media-specific features that can often be automatically extracted from media types.

Unfortunately, MPEG-7 is not currently suitable for describing top-level multimedia features, because i) its XML Schema-based nature prevents the effective manipulation of descriptions and its use of URNs is cumbersome for the web; ii) it is not open to the web standards for representing knowledge.

Some efforts was done in order to translate the semantic of the standard in some knowledge representation languages [71], [96], [125]. All these methods perform a one to one translation of MPEG-7 types into OWL concepts and properties. This translation does not, however, guarantee that the intended semantics of MPEG-7 is fully captured and formalized. On the contrary, the syntactic interoperability and conceptual ambiguity problems remains.

An interesting work [101], [17] was done in order to define a multimedia ontology They try to define a new multimedia ontology that take into account the semantic of MPEG-7 standard. They started using some patterns derived from a foundational ontology DOLCE [28]. In particular they used two design patterns Descriptions & Situations (D & S) and Ontology of Information Objects (OIO), which are two of the main patterns provided by DOLCE. The ontology already covers a very large part of the standard, their modeling approach has the aim to offer even more possibilities for multimedia annotation than MPEG-7 since it is truly interoperable with existing web ontology. This approach put some constraints on the image semantic thought the use of foundational ontology but their work are more focused on the interoperability purpose.

4.2.3 Image Management in Web Environment

The aim of the Semantic Web is to augment the existing Web so that resources (Web pages, images etc.) are more easily interpreted by programs (or "intelligent agents "). The idea is to associate Web resources with semantic categories which describe the contents and/or functionalities of Web resources. As described in the report [5], the problem in web scale it is to share a common and complete (for the image data) vocabulary, to understand the meaning of the URI. Many of the relevant vocabularies have been developed prior to the Semantic Web. Most notably, the key International Standard in this area, the Multimedia Content Description standard, widely known as MPEG-7, is defined using XML Schema. At the time of writing, there is no commonly accepted mapping from the XML Schema definitions in the standard to RDF or OWL as described in the previous section. Many annotations on the Semantic Web are about an entire resource. For example, a <dc:title> property (Dublin Core Vocabulary) applies to the entire document. For images and other multimedia documents, one often needs to annotate a specific part of a resource (for example, a region in an image). Sharing the metadata dealing with the localization of some specific part of multimedia content is important since it allows to have multiple annotations (potentially from multiple users) referring to the same content. It is important that the rest of the annotation file focuses on the description of the content of the image. In order to provide a complete description of the content, it is often useful to try to answer the questions "when", "where"and "why "the photo has been taken, and "who"and "what"is depicted on the photo, because these are the most probable questions that the end-user would like to be queried during the retrieval process. Another problem is how to deal with URIs, these are unique identifiers but in the image we could deal with the image itself that could have different copy on web and its content.

4.3 Knowledge Text Management

Text is the second kind of data managed during my research activity. Typically, in the literature we have a research field called *text mining*[52], that refers generally to the process of deriving high quality information from text. High quality information is typically derived through the use of patterns such as statistical pattern learning. Text mining usually involves the process of structuring the input text, deriving patterns within the structured data, and finally evaluation and interpretation of the output. "High quality"in text mining usually refers to some combination of relevance, novelty, and interestingness. Although Text Mining and Data Mining are related as they are mining processes they differ in point of the following issues:

- Text mining deals with unstructured or semi-structured data, such as text found in articles, documents, etc. However Data Mining is related to structured data from large databases. In addition, another characteristic of text mining is the amount of textual data. The concepts contained in a text are usually rather abstract and can hardly be modelled by using conventional knowledge representation structures.

- Furthermore, the occurrence of synonyms (different words with the same meaning) or homonyms (words with the same spelling but with distinct meanings) makes it difficult to detect valid relationships between different parts of the text. Text mining techniques enable to discover and use the implicit structure of the texts (e.g. grammar structure) and they usually integrate some specific Natural Language Processing (Corpus Linguistics).

Text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling (i.e., learning relations between named entities). The most important text mining tasks are document clustering and text summarization. The basic idea in clustering is that similar documents are grouped together to form clusters. The summarization usually consists in producing summaries that contain not only sentences that are present in the document but also new automatically constructed phrases that are added to the summary to make it more intelligible. Instead in my activity I choose the name *Knowledge Text Management* because I am much more interesting in approaches that has the aim to extract the underlying semantic through the using of external knowledge and not only the single entities or classes. I will describe two main approaches for discover the semantic inside the text. The first is founded on algebraic and statistic methods and it has the aim to make text mining and to extract topic from text collections, their main ideas coming from Information Retrieval community. The second are more focused on the algorithm able to structure the knowledge of a given document. This is done in order to manage the full knowledge inside the document in way that also the machines could be use these obtained structures. The ontology could help to see at text as a set of concept,relations, instances of a given knowledge base. In the last case the ontology and their techniques are the core elements of those approach. In particular i will introduce the problem of structuring legal document, that is the particular domain I have used as case study.

4.3.1 Text Mining and Topic Detection

In the literature, several techniques are used to discover or to represent the knowledge inside the text. Some of these approaches are derived from the Information Retrieval community.

Most of these approaches are "synthetic", i.e. they try to represent the text data in a useful space where it is more easy to discover the semantic properties, the knowledge content of the text data. This set the first main question: What is the best model for representing the knowledge inside a text? A second question, that is strictly related with the first choice, is: what kind of knowledge are we looking for in the text document?

Algebraic and statistical approach reduces the problem representing the document by means of a model the provides application of techniques coming from pattern recognition theories and improving some methods taking into account the production process of those text data. Hardly, these aspects are taken into account when a document is mapped in those models and the semantic discovery becomes a problem to find the right categories of a set of documents . The main model used is the *vector space model* [86], mainly because of its conceptual simplicity and the appeal of the underlying metaphor of using spatial proximity for semantic proximity. With that model documents and queries are represented in a high-dimensional space,in which each dimension of the space corresponds to a word in the document collection. The most relevant documents for a query are expected to be those represented by the vectors closest to the query, that is, documents that use similar words to the query. The vectors are built according to different terms weighting. Once we have represented the document in those spaces, we can apply also several techniques coming from machine learning theory because, we have now a simple features vector. One of the most famous methods, in the field of text analysis, is *Latent Semantic Indexing*(LSI) [41]. LSI is a technique that projects queries and documents into a space with latent semantic dimensions. Co-occurring terms are projected onto the same dimensions, non-co-occurring terms are projected onto different dimensions. In the latent semantic space, a query and a document can have high cosine similarity even if they do not share any terms, as long as their terms are semantically similar according to the co-occurrence analysis. We can look at LSI as a similarity metric that is an alternative to word overlap measures like term frequency or inverse document frequency. The latent semantic space that we project into has fewer dimensions than the original space (which has as many dimensions as terms). LSI is dimensionality thus a method for dimensionality reduction. A dimensionality reduction reduction technique takes a set of objects that exist in a high-dimensional space and represents them in a low-dimensional space, often in a two-dimensional or three-dimensional space for the purposes of visualization. In the last five year some powerful approaches take place, known as *latent topic model*. Most of them are studied to discover topic inside a text collections. Here the topic becomes to be something different from LSI method. Infact the LSA topic is more closed to "soft-cluster "idea, in the latent topic model the topic becomes a statistic distribution of words over a collections of documents. In this category i would name Probabilistic Latent Semantic Analysis (PLSA) [67], that is an extension of LSI, and the Latent Dirichlet Allocation (LDA) [24]. The LDA has studied a lot and different version of this approach, and is proposed in order to improve the infer process of word distribution (topic generation). The LDA differs from PLSA because it is not only a process for describing topic but also it is generative method able to generalaize the word distributions starting from the only observations of word inside a given texts collection. This is useful to describe all possible documents that could be written with a given domain and vocabulary.

4.3.2 Knowledge Extraction from Text Document

In this section I will explain some approaches that have the aim to structured the knowledge inside the documents using external knowledge such as ontology. These methods differ from the previous approaches because they would preserve in suitable structure the whole knowledge inside a document. Most of the papers in that field distills structured data or knowledge from unstructured text by identifying references to named entities as well as stated relationships between such entities. This was done using Natural Language Processing Algorithm [73], [65] and pattern classification techniques [45]. In works like [46], starting from ontology they formulate rules to extract constants and keywords and apply a recognizer to organize extracted constants as attribute values of tuples in a generated database schema. In the last years a new research field takes place related with this aims, called *ontology learning*[36]. The term ontology learning was originally coined by Alexander Madche and Steffen Staab and can be described as the acquisition of a domain model from data. Obviously, ontology learning needs input data from which to learn the concepts relevant for a given domain, their definitions as well as the relations holding between them. One crucial requirement is thus that the input data is representative for the domain one aims to learn an ontology. The process of learning the extensions for concepts and relations is commonly referred to as ontology population. Further, we will speak of knowledge markup or annotation if the population is done by selecting text fragments from a document and assigning them to concepts. The subtasks related with the ontology learning are :

- acquisition of the relevant terminology,
- identification of synonym terms / linguistic variants (possibly across languages),
- formation of concepts,
- hierarchical organization of the concepts (concept hierarchy),
- learning relations, properties or attributes, together with the appropriate domain and range,
- hierarchical organization of the relations (relation hierarchy),
- instantiation of axiom schemata,
- definition of arbitrary axioms.

4.3.3 Knowledge in Text Legal Domain

The legal domain is very complex compared to others, because it involves knowledge of the physical and social worlds, as well typical legal knowledge that actually creates a novel layer over the social world. The legal practitioner applies conceptual thinking and legal structural knowledge that he gained over long-term training. The complexity of law demands an abstract, differentiating, economical and functional technical language, "legal language ", which is able to represent the structures and meanings in law. The law is not just a collection of mechanical if/then rules. Based on the same facts and legal rules, legal expert may indicate contradictory solutions. A correct syllogism may be overruled by social conventions, principles or circumstances. Although where the explicit knowledge exists, some legal problem may be not resolved simply and legal decision could be not predictable. Law is based on text and language, and the language could be the source of different interpretation. In those case (from on hand) we have to deal with a new kind of symbolic representation and reasoning such as modal theory [23] if we would predict and infer the relationship between the facts and the related laws. From the other hand, we need to build lexicons and knowledge bases that have to be shared mach as possible among the domain expert. The ontology helps to describe the meaning and a context of a given information. Their possible fields of application in law are manifold. The use of ontologies for the formalization of the law is, however, not a new approach. Infact several works to represent legal knowledge has been proposed, such as: Valente et all Functional Ontology of Law [26], Frame-based Ontology of Visser [126], McCarty's Language of Legal Discourse (LLD) [89] and Stamper's Norma [116]. As a consequence of such theories, several ontologies are now available, such as: ON-LINE (Ontology-based Legal Information Environment), DUBA (Dutch Unemployment Benefits Act), CLIME (Cooperative Legal Information Management and Explanation): Maritime Information and Legal Explanation (MILE) and Knowledge Desktop Environment (KDE) [89]. Several approaches based on the wordNet project have been also done: in particular in Italy, JurWordNet [122] is the first Italian legal ontology [1].It is worth noticing that, despite the vast amount of efforts, several challenging problems still remain opened, especially related to the *automatic ontology building process*. The use of Pattern Recognition techniques on the sentence level for the identification of concepts and document classification for automatic document description is described in several works, as SCISOR[72] and FASTUS [66]. In the system BREVIDOC, documents are automatically structured and important sentences are extracted. These sentences are classified according to their relative importance [90]. From the Natural Language Processing (NLP) point of view, legal research concentrates on the automatic description of documents. In particular, the main focuses are: development of thesauri,

[1]We gratefully acknowledge ITTIG - CNR, Italy, and particularly dott.Tiscornia, for the use of JurWordNet in this work

machine learning for features recognition, disambiguation of polysems, automatic clustering and neural networks. The most important systems are the HYPO [102] and SALOMON [89].

Capitolo 5

Image Ontology System

Image Ontology System

5.1 Introduction

In this chapter I will describe the framework and the system used to analyze and process the content of multimedia objects, i.e. images.

The framework is based on the main idea of linking low and intermediate features detected by a computer vision system to a general knowledge produced by a domain expert. The bridge , among these different kinds of information, is done by means of a *Multimedia Ontology*. In particular, this chapter is organized as follows. Section 5.2 describes at a glance the underlying vision theory that is at the basis of our ontological framework and provides theory for multimedia knowledge base and multimedia ontology foundations. Section 5.3 describes the underlay data models. Section 5.4 depicts the query algorithms in Image Ontology Theory. Section 5.5 describes the proposed system architecture that has been realized and experimented, as described in section 6.5.
The following running example will help to explain the purpose of my approach. Let us consider the picture in Figure 5.1.

Figura 5.1: Running Example

Using an image processing and analysis system, it is possible to obtain a description of the content of this image in terms of color and shape, together with the grade of uncertainty that we expect each image processing algorithm to produce. An intelligent system, using a classifier, might associate some elementary concepts to the extracted multimedia feature, related to the image itself, e.g.$\{\langle person, horse, grass, sand\rangle\}$. Thus, it is clear that the representation requirements of image data and, more generally, of multimedia data can be improved if there is a model that is able to describe more complex concepts. For example, $jockey$ or $racing - track$ concepts could be redefined through the use of more elementary concepts obtained by an intelligent system. In this context, we need a system that should allow specifications for: *Special Relationships* that exist between the different media entities characterizing a media object or an event - for example, geometrical, spatial, temporal relationships and so on; *Uncertainty* that is produced by Computer Vision systems when processing and recognizing multimedia contents - for example, object detection in an image or a video is always associated to a certain membership degree; *Association between low-level properties and semantic properties of images* - for example, the semantics of an image can be enriched/learned/guessed by observing the relationships of its color and shape with real-world concepts; An *associated reasoning service* which can use the available feature observations and concept description to infer other probable concepts in presence of uncertain relationships - for example, some shape, texture and color properties might be used to infer that a certain media object is an instance of a given concept: e.g., colors=$yellow$ with a grade μ_y, shape=$circle$ with a grade μ_c may be associated with the concept of the sun with a grade $min\{\mu_y, \mu_c\}$. Considering the image of Figure 5.1, the proposed framework will provide utilities for: i) representing spatial relationships, such as: a person sits above a horse; ii) managing the uncertainty that is related to all the detected objects, such as the person and the horse; iii) representing suitable features (color, shape and texture) for each detected object, such as the person and the horse; iv) providing an appropriate reasoning service that could infer that in the image there is indeed a jockey riding a racing horse. It is not easy to represent and use this kind of knowledge using classical data models, such as the relational one. In our perspective, new kinds of theory are required to express the semantics of multimedia data, i.e. a novel model of ontology and related languages.

5.2 The Model

In this section, i describe a novel model for representing and managing multimedia data: in particular, i first start from several considerations about how a human vision system is able to store and manage multimedia information for high level image understanding purposes. Furthermore, a formal representation of the processed data will be given, having the aim of designing an automatic, content-based multimedia query engine.

5.2.1 The Human Vision System

Given an image I, a human decodes its knowledge content after different cognitive steps, as described in Figure 2. Each step is related to a human perceptive process and some of these steps are iterated in order to derive more complex concepts. Several steps are image processing blocks that approximate the human vision process on the whole image or on parts of an image. Psychological theories propose two different approaches for recognizing concepts related to an image: the holistic and the constructive one [22]. According to the holistic theory, and image is processed and recognized by humans considering the whole image. In contrast, the constructive theory considers image understanding as a progressive process: a human first recognizes an interesting part of an image and infers the knowledge of the whole image from the knowledge of its parts, in a recursive fashion. I follow this latter approach. In addition, i also need a further environmental knowledge that describes all the necessary knowledge as evidences by the classical “meaning triangle” [114] : in a given media, i detect symbols, objects and concepts; in a certain image we have a region of pixels (symbol) related to a portion of multimedia data; this region is an instance (object) of the certain concept. In other words, i can detect concepts but i am not able to disambiguate among the instances without some specific knowledge.

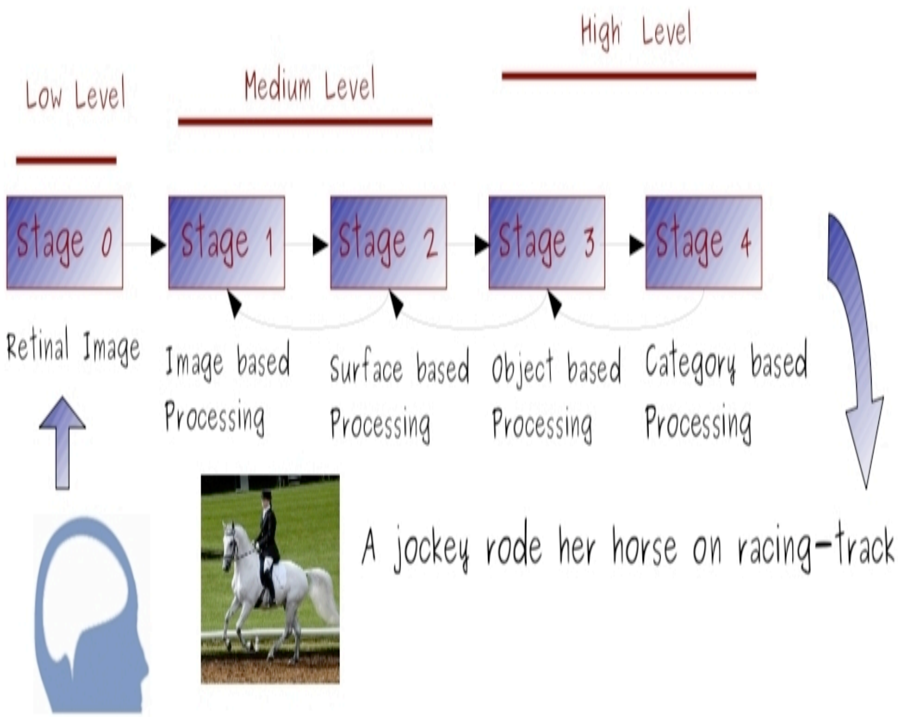


Figura 5.2: The Process of Visual Perception

Figura 5.3: An useful image for recognizing the visual perception process

A simplified version of the described vision process will consider only three main levels: *Low, Medium and High*, as depicted in Figure 2. In fact, the knowledge associated to an image is described at three different levels: *Low level*: raw images, computationally and abstractly thought of as discrete functions of two variables defined on some bounded and usually regular regions of a plane, i.e a pixel map used to structure the image perceptual organization and in filtering processes in order to obtain new maps; *Intermediate level*: an aggregation of data related to the use of spatial features - including points, lines, rectangles, regions, surfaces, volumes - color features, textures and shape features, for example colors are usually described by means of *color histograms* and several features have been proposed for texture and shapes, all exploiting spatial relations between a number of low level features (pixels) in a certain region; *High level*: this layer is related to axioms that involved concepts and their relations conveyed by an image; looking at Figure 5.1 i could use these sentences to define the high level: "A jockey rode her horse on a racing-track". The features associated to these layers should be characterized in terms of a *fuzzy value*. These *fuzzy values* represent a certain *degree of uncertainty* that each image processing algorithm produces, i.e we might say the shape is "highly" trapeze, or that it is "a little bit" rectangular. Expressions such as *highly*, *a little bit*, and so on, recall this notion of fuzziness implicitly related to the *similarity of visual stimuli*. We can associate this fuzziness to some regions inside the image related to colors, shapes and textures. Considering the running example image, the derived features are the following: Colors:$\{\langle Green, 0.8\rangle, \langle White, 0.89$ $\langle Black, 0.7\rangle$, $\langle Brown, 0.75\rangle\}$, Shapes: $\{\langle Rectangle, 0.6\ \rangle$,

$\langle Trapeze, 0.9\rangle\}$. Textures: $\{\langle Animal, 0.8\rangle$, $\langle Nature, 0.6\rangle\}$.

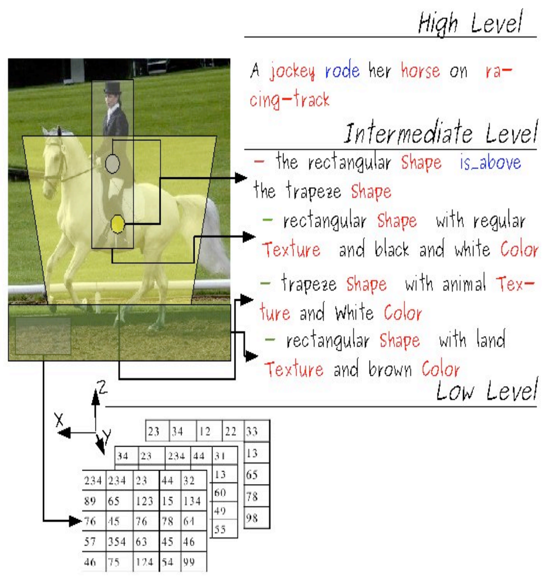


Figura 5.4: The Levels Description of the Running Image

5.2.2 The Image Knowledge Framework

The Image Knowledge Framework is a logical theory used to express the semantic underlay an image taking into account a fusion between the general concepts inside a domain ontology and media formulation of those concepts. This frameworks is the core of the Multimedia Knowledge Representation System that is able to compute a query answering process over an image database reducing the semantic gap between the (automatic or manual) image annotations and their semantic content. That framework is based as set axioms in Description Logic format and also the instances are a set of assertion over the defined DL concepts and role.

The Description Logic framework considered in this dissertation is the $\mathcal{SHOIN}(D)$, corresponding to the ontology description language OWL-DL. In the following, we will briefly present its syntax and semantics, starting from a general knowledge base, then this general framework will be used to define the proposed image ontology theory.

$\mathcal{SHOIN}(D)$ allows reasoning on concrete data types, such as strings and integers, using the so-called *concrete domains* (D). The $\mathcal{SHOIN}(D)$ Knowledge Base is a knowledge base defined as $\mathcal{K}$=$\langle\mathcal{T},\mathcal{R},\mathcal{A}\rangle$ where $Tbox$ ($\mathcal{T}$) is a finite set of concept-inclusion axioms, $Rbox$ ($\mathcal{R}$) is a finite set of transitivity and role-inclusion axioms, $Abox$ ($\mathcal{A}$) is a finite set of concepts and role assertion axioms and individual (in)equality axioms.

I give more details about the features of $\mathcal{SHOIN}(D)$ in the following list:

- **Concrete Domains** :

  A concrete domain D is a pair $\langle\Delta_D,\Phi_D\rangle$, where Δ_D is an interpretation domain and Φ_D is the set of concrete domain predicates d with a predefined arity n and an interpretation $d^D \subseteq \Delta_D^n$. For instance, the assertion $Rectangular \sqcap \exists has\_value. \geq 0.7$ will denote an object having a rectangular shape and value greater or equal to 0.7. The last value is associated through the relation has_value and only further hypothesis can also read that value as a certainty degree .

- **Alphabethes**:

  The Alphabets $\mathtt{C}$, $\mathtt{R}_a$, $\mathtt{R}_c$,$\mathtt{I}_a$, $\mathtt{I}_c$, are non-empty, finite and pair-wise disjoint sets of *concepts names, abstract roles names, concrete roles names, abstract individual names* and *concrete individual names*.

- **RBox**:

  An abstract role that does not have transitive sub-roles is called *abstract simple* role. An RBox R consists of a finite set of transitivity axioms $trans(R)$, and role inclusion axioms of the form $R \sqsubseteq S$ and $T \sqsubseteq U$, where R and S are abstract roles, and T and U are concrete roles. The reflexive-transitive closure of the role inclusion relationship is denoted with $\sqsubseteq^*$. A role not having transitive sub-roles is called simple role.

- **Concepts**:

  The set of $\mathcal{SHOIN}(D)$ concepts C is defined by the following syntactic rules:

$$\begin{aligned} C \quad \longrightarrow \quad & \top|\bot|A|C \sqcap C_1|C \sqcup C_1|\neg C|\forall R.C|\exists R.C| \\ & |(\leqslant nS)|(\geqslant nS)|\ \{a_1,...,a_n\}|(\leqslant nT)|(\geqslant nT)| \\ & \forall T_1,...,T_n.D|\exists T_1,...,T_n.D \end{aligned}$$

A being an atomic concept, R an abstract role, S an abstract simple role, T a concrete role, d a concrete domain predicate and a_i and c_i abstract and concrete individuals, respectively; the concrete domain D is defined as $D \longrightarrow d|\{a_1, .., a_n\}$ with $n \in \mathbb{N}$.
For example, the concept "Jockey", introduced in the previous example and its related concepts are defined in this way:

$$\text{Jockey} \equiv \text{Person} \sqcap \forall \text{ride.(RaceHorse)}$$
$$\text{HorseRacerTrack} \equiv \text{Track} \sqcap \exists \text{rounded.Grass.}$$
$$\text{RaceHorse} \equiv \text{Horse} \sqcap \forall \text{ located\_in HorseRacerTrack.}$$
$$\text{Track} \sqsubseteq \forall \text{ made\_of.Sand}$$
$$\text{Horse} \sqsubseteq \text{Animal} \sqcap \forall \text{ has\_legs(=4)}$$

- **TBox**:
  A $\mathcal{SHOIN}(D)$ $TBox$ $\mathcal{T}$ consists of a finite set of concept inclusion axioms :

$$C \sqsubseteq D$$

  where C and D are concepts. We can use also $C = D$ in $\mathcal{T}$ in place of $C \sqsubseteq D, D \sqsubseteq C$ in $\mathcal{T}$. An abstract simple role S is called functional if the interpretation of role S is always functional (see later for the semantics). A functional role S can always be obtained from an abstract role by means of the axiom $\top \sqsubseteq \leq 1S$. Therefore, whenever we say that a role is functional, we assume that $\top \sqsubseteq \leq 1S$ is in the TBox.

- **ABox**:
  An ABox $\mathcal{A}$ consists of a finite set of concept and role assertion axioms and individual (in)equality axioms $a : C, (a, b) : R, (a, c) : T, a \neq b$ and $a = b$, respectively.

The semantics of the previous abstract syntactic rules is given in Tarski-style. We define an $interpretation$ $\mathcal{I}$ with respect to a concrete domain D as a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ consisting of a non empty set $\Delta^{\mathcal{I}}$ (called the $domain$), disjoint from Δ_D, and of an $interpretation\ function$ $.^{\mathcal{I}}$ that assigns to each $C \in \mathtt{C}$ a subset of $\Delta^{\mathcal{I}}$, to each $R \in \mathtt{R}_a$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, to each $a \in \mathtt{I}_a$ an element in $\Delta^{\mathcal{I}}$, to each $c \in \mathtt{I}_c$ an element of Δ_D, to each $T \in \mathtt{R}_c$ a subset of $\Delta^{\mathcal{I}} \times \Delta_D$ and to each n-ary concrete predicate d the interpretation $d^D \subseteq \Delta_D^n$. The mapping $.^{\mathcal{I}}$ is extended to concepts and roles as usual:

$$\begin{array}{ccc}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
\neg C^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C_1 \sqcap C_2)^{\mathcal{I}} & = & C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\
(C_1 \sqcup C_2)^{\mathcal{I}} & = & C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}} \\
(S^-)^{\mathcal{I}} & = & \{\langle y, x\rangle : \langle x, y\rangle \ \in S^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} & = & \{x \in \Delta^{\mathcal{I}} | R^{\mathcal{I}}(x) \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} & = & \{x \in \Delta^{\mathcal{I}} | R^{\mathcal{I}}(x) \cap C^{\mathcal{I}} \neq \emptyset\} \\
(\geq nS)^{\mathcal{I}} & = & \{x \in \Delta^{\mathcal{I}} : |S^{\mathcal{I}}(x)| \geq n\} \\
(\leq nS)^{\mathcal{I}} & = & \{x \in \Delta^{\mathcal{I}} : |S^{\mathcal{I}}(x)| \leq n\} \\
\{a_1, \ldots, a_n\}^{\mathcal{I}} & = & \{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}
\end{array}$$

and similarly for the other constructs where $R^{\mathcal{I}}(x) = \{y : \langle x, y\rangle \in R^{\mathcal{I}}\}$ and $|X|$ denotes the cardinality of the set |X|. In particular:

$$(\exists T_1, \ldots, T_n.d)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} : [T_1(x)^{\mathcal{I}} \ldots T_n(x)^{\mathcal{I}}] \cap d^D \neq \emptyset\}$$

Example 3. *Let us consider the following axioms of a simple ontology ($TBox\mathcal{T}$) about the people that works in a software company, with a empty RBox ($R = \emptyset$):*

$ProjectPhase \sqsubseteq (= 1\ has\_leader) \sqcap (= 1\ has\_duration)$

$Project \sqsubseteq (\geq 1\ has\_woker) \sqcap (= 3\ has\_phase)$

$\top \sqsubseteq \forall has\_leader.ProjectManager$

$\top \sqsubseteq \forall has\_worker.\mathbb{N}$

$\top \sqsubseteq \forall has\_duration.month/man$

$\top \sqsubseteq \forall has\_phase.ProjectPhase.$

$SwManger \sqsubseteq ProjectManager$

$$Requirements\_Analysis \sqsubseteq ProjectPhase \sqcap \exists\ has\_duration. \leq_2$$

$$Design \sqsubseteq ProjectPhase \sqcap \exists comeAfter.Requirements\_Analysis$$

$$Development \sqsubseteq ProjectPhase \sqcap \exists comeAfter.Design\_Project \sqcap has\_Leader.SwManger$$

$$Infomation\_System\_Project = Project \sqcap \exists$$

$$hasPhase.(Requirements\_Analysis \sqcup Design \sqcup Development) \sqcap \exists has\_woker. \geq_6$$

In $\mathcal{T}$, the value for has_worker ranges over the concrete domain of natural numbers while the value for duration ranges over the concrete domain of month for man. The concrete predicate $\leq_2$ is true if the value is smaller or equal than to 2 and the one $\geq_6$ is true if the value is greater or equal than to 6

The ABox A contains the following assertions:

$$ra : Requirment\_Anaysis \sqcap \exists has\_Leader.\{picus\}$$
$$da : Development \sqcap \exists has\_Leader.\{antonio\}$$
$$InfoSurgeon : Information\_System \sqcap \exists has\_phase.\{ra\} \sqcap \exists has\_phase.\{da\}$$

I use this framework in order to define a multimedia knowledge base that takes into account the specificity of multimedia data and in particular of images. In the image domain, each image may be decomposed into a set of regions that are characterized in terms of texture, color and shape, as described in the previous sections; some of these regions can be associated to the instances of some concepts as derived from image analysis algorithms. In addition, we can infer new kinds of concepts that cannot be derived from intermediate image features used by image processing and analysis algorithms. In the following, we will describe the proposed Image Knowledge Base.

Let us give a reference domain of IMages (Δ_{IM}). We describe the image knowledge in terms of: *AGgregate concept* (C_{Ag}), *High Media concept* (C_{HM}) and *Intermediate Media concept* (C_{IM}), $Image$ concept, $SubImage$ concept and in terms of relations among them.

We informally define $C_{Ag}, C_{HM}, C_{IM}, Image, SubImage$ as in the following:

- C_{IM} is the set of auxiliary concepts describing the shape, texture, colors properties of object belonging to the reference domain of IMages (Δ_{IM}) with a certain *degree of fuzziness*.
- C_{HM} is the set of concepts, whose semantics is completely inferred by their multimedia features through the automatic computer vision machines and they have some relations to the C_{IM} and to the objects belonging to the domain Δ_{IM}. These relationships have a certain *degree of fuzziness* (e.g. water, sand, elephant, grass, horse).
- C_{Ag} is the set of concepts belonging to a general domain defined through the axioms over high media concepts and/or aggregate concepts or/and some relations to them (e.g. jockey, racing-track) and those kinds of axioms can relate those concepts with the object of Δ_{IM}.

- $Image$ and $SubImage$ is the set of concepts whose individuals are belonging to the individuals of Δ_{IM}.

In order to associate a fuzzy membership among the elements of the previous concepts, a *reification pattern* is used. This is a solution used to express n-ary relations with binary relations. Infact in the syntax of Description Logic only binary and unary predicates are provided, instead the proposed fuzzy membership involves ternary predicates. This pattern is based on the use of auxiliary variable that represents the relation instances itself with links from the subject of the relation to this instance and with links from this instance to all participants that represent additional information about this instance.

For example, if we have a sentences such as "The SubImage s has_recognized as person with fuzziness 0.4 ", we can express it in DL as :

$$s : SubImage \sqcap (s, r)hasRecognized \sqcap r : ReifConcept \sqcap (r,'' 0.4'') : \\ hasFuzzyValue \sqcap (r,'' person'') : hasObjectValue$$

The *r:ReifConcept* is the introduced concept used to simulate the ternary relation.

Note that the two relations *hasFuzzyValue* and $hasObjectValue$ have to be functional and some restrictions have to be used, such as existential quantification qualified for the relation *hasObjectValue* and universal quantification qualified for both *hasFuzzyValue* and *hasObjectValue*.

Regarding the W3C languages, in order to do that we can use *rdf blank nodes* or the anonymous classes obtained through the owl restrictions : *owl:someValuesFrom*, *owl:allValuesFrom*.

Through the use of a *reification pattern*, we associate a value belonging to the concrete domain (i.e xsd:float). In particular we use that pattern between the elements of C_{HM} and the individuals of the concepts $Image$ or $SubImage$ and between the elements of C_{IM} and the individuals of the concepts $Image$ or $SubImage$. These float values are the fuzzy measures produced by image engines and used by the query engine to formulate an output ranking as described in the next sections.

Formally speaking:

Definition 1 ($IMage$ Knowledge Base)**.** *The $IMage$ $\mathcal{SHOIN}(D_n)$ knowledge base ($\mathcal{K}_{IM}$) is a $\mathcal{SHOIN}(D_n)$ knowledge base defined as:*

$$\mathcal{K}_{IM} = \langle \mathcal{T} \sqcup \mathcal{T}_{IM}, \mathcal{R} \sqcup \mathcal{R}_{IM}, \mathcal{A} \sqcup \mathcal{A}_{IM} \rangle.$$

the Tbox $\mathcal{T}_{IM}$ being a finite set of inclusion axioms related with C_{Ag}, C_{HM}, C_{IM}; $\mathcal{R}_{IM}$ being a finite set of role-assertion axioms on properties involved also $Image$ and/or $SubImage$ concepts; Abox $\mathcal{A}_{IM}$ being a finite set of concepts and role assertions on the relations belonging to $\mathcal{R}_{IM}$ and C_{Ag},C_{HM}, C_{IM}.

The syntactic rules of $\mathcal{K}_{IM}$ are defined in this way:

$$\begin{array}{lcl} C & \rightarrow & \top|\bot|A|C \sqcap C^1|C \sqcup C^1|\neg C|\forall R.C|\exists R.C| \\ & & |(\leqslant nS)|(\geqslant nS)|\ \{a_1, ..., a_n\}|(\leqslant nT)|(\geqslant nT)| \\ & & \forall T_1, ..., T_n.D|\exists T_1, ..., T_n.D \\ D_{Ag} & \rightarrow & C \\ C_{Ag} & \rightarrow & D_{Ag} \mid C_{HM} \mid C_{Ag} \sqcap C^1_{Ag}|C \sqcup C^1_{Ag}|\neg C_{Ag}|\forall R.C_{Ag}|\exists R.C_{Ag} \end{array}$$

The following assertions are also provided:

1. $C \sqsubseteq D$
2. $C_{Ag} \sqsubseteq D_{Ag}$
3. $\exists hasVisionContent.(SubImage \sqcap \exists hasHighConceptFuzzyValue.Xsd : Float) \sqsubseteq C_{HM}$
4. $Image \sqsubseteq \exists partOf^{-}.SubImage$
5. $SubImage \sqsubseteq \exists partOf.Image$
6. $\exists hasColorReification.Color \sqcap \exists hasShapeReification.Shape \sqcap \exists hasTextureReification.Texture \sqsubseteq SubImage$
7. $\exists hasColorValue.Xsd : String \sqcap \exists hasIntermediaConceptFuzzyValue.Xsd : Float \sqsubseteq Color$
8. $\exists hasTextureValue.Xsd : String \sqcap \exists hasIntermediaConceptFuzzyValue.Xsd : Float \sqsubseteq Texture$
9. $\exists hasShapeValue.Xsd : String \sqcap \exists hasIntermediaConceptFuzzyValue.Xsd : Float \sqsubseteq Shape$

10. $func(hasShapeValue), func(hasColorValue),$
$func(hasTextureValue)$ $func(hasHighConceptFuzzyValue),$
$func(hasIntermediaConceptFuzzyValue), func(hasVisionContent),$
$func(hasVisionContent^{-}), func(partOf)$

by $func(R)$ we mean that the role R is functional. The $Color, Shape, Texture$ are the C_{IM} concepts that are also used to define the fuzzy measures through the reification pattern, in that case we have different color, shape and texture for each images . $Xsd : Float$ and $Xsd : String$ are the two concrete domain used to express the fuzzy values and the property value respectively.
In that core, also some topological relations among elements of $SubImage$ exist: by means of those relations, we will represent how the $SubImage$ elements are correlated among them in spatial way. The set of these relations is called R_S= { $onTheTopOf, onTheLeftOf, onTheRightOf,$ $onTheBottomOf$,$SpatialDisjoint, SpatialOverlap, onTheLeftTopOf, onTheRightTopOf,$ $onTheLeftBottomOf, onTheRightBottomOf$ }

The previous assertion from 2 to 10 are the core of $\mathcal{T}_{IM}$ and $\mathcal{R}_{IM}$.

The semantics of the previous syntactic rules is given in terms of the interpretation function in addition to the concrete image domain. For example, we use the interpretation function $\mathcal{I}$ for defining the semantics of C_{Ag}, C_{HM}, C_{IM}. In other words, the previous set of axioms takes into account some intuitive relations in the image domains. In particular, they capture the idea that each image may be formed by components of sub-images. In addition, the relations between rough multimedia data and extracted features (color, shape, texture) and high level knowledge are captured, together with the related uncertainties. The semantics of the remaining roles and concepts involved in $\mathcal{K}_{IM}$ will be given as follows. We assign the concepts $SubImage$ and $Image$ to a subset of Δ_{IM}, and we assign to the relations:.

- $hasVisionContent$ a subset of $\Delta^{\mathcal{I}} \times \Delta_{IM}$
- R^* a subset of $\Delta_{IM} \times \Delta^{\mathcal{I}}$
- $hasIntermediaConceptFuzzyValue$ a subset of $\Delta^{\mathcal{I}} \times XSD : Float \subseteq \Delta_D$
- $hasHighConceptFuzzyValue$ a subset of $\Delta_{IM} \times XSD : Float \subseteq \Delta_D$
- $partOf$ a subset of $\Delta_{IM} \times \Delta_{IM}$

- $r \in R_S$ a subset of $\Delta_{IM} \times \Delta_{IM}$

R^* being one of the relations $hasColorReification, hasShapeReification, hasTextureReification$

Eventually, we can now give a simple and formal definition of *image ontology*:

Definition 2 ($IMage\ Ontology$)**.** *An* $IMage\ Ontology$ $\mathcal{O}_{IM}$ *is an* OWL*-*DL *ontology defined according to the IMage* $\mathcal{SHOIN}(D)$ *knowledge base.*

5.2.3 An example of Image Ontology

Let us consider our running example. An example of associated schema is informally given in the following:

Jockey is a *Person* on the top of the *Racehorse*.

Racehorse is an *Horse* located in the *Horse-race-track*.

Horse-race-track is a *Track* surrounded by *Grass*.

The *Horse* is on the top of the *Track*.

Track is made of *Sand*.

Person, on the top of the *Horse*, rides that *Horse*.

Person, *Horse*, *Grass* and *Sand* are described by a *SubImage* of *Image* with a related fuzziness value and some *Shape*, *Texture*, *Color* are associated to them with a related fuzziness value.

In order to best understand the extensional level of the proposed ontology, let us consider Table 5.1 which contains images and/or part of images of the analyzed image, belonging to the Δ_{IM} . In Table 5.1, these instances are labeled 1, 2, 3, 4, 5. In addition, an example of some instances associated to the previous schema is informally given as follows:

An instance of *Person* is associated with the instance 1 of $SubImage$, with a degree of fuzziness 0.58; the instance 1 has as *Color* an instance having as associated concrete data$''black''$ and a related degree of fuzziness 0.65; and as *Shape* an instance, named $shape\_p1$, having an associated concrete data “$rectangular$“ and a related degree of fuzziness 0.7 ; and as *Texture* an instance having an associated concrete data$''skin''$ and a related degree of fuzziness 0.55. An instance of *Grass* is associated to the instance 2 $SubImage$, with a degree of fuzziness 0.9; the instance 2 has as *Color* an instance having as associated concrete data$''green''$ and a related degree of fuzziness 0.9; and as *Shape* an instance, named $shape\_g1$, having an associated concrete data “$rectangular$“ and a related degree of fuzziness 0.7; and as *Texture* an instance having an associated concrete data $''nature''$ and a related degree of fuzziness 0.67. An instance of *Horse* is associated with the instance 3 of $SubImage$, with a degree of fuzziness 0.7; the instance 3 has as *Color* an instance having as associated concrete data $''white''$ and a related degree of fuzziness 0.65; and as *Shape* an instance, named $shape\_h1$, having as associated concrete data "$trapeze$" and a related degree of fuzziness 0.5; and as *Texture* an instance having as associated concrete data $''animal''$ and a related degree of fuzziness 0.55. An instance of *Sand* is associated with the instance 4 of $SubImage$, with a degree of fuzziness 0.6; the instance 4 has as *Color* an instance having as associated concrete data $''brown''$ and a related degree of fuzziness 0.5; and as *Shape* an instance, named $shape\_s1$, having as associated concrete data "$rectangular$", and a related degree of fuzziness 0.65; and as *Texture* an instance having as associated concrete data $''organic''$ and a related degree of fuzziness 0.55. The instances $1, 2, 3, 4$ of $SubImage$ are image parts of the $Image$ instance 5. The instance $shape\_p1$ is *on the top* of the instance $shape\_h1$, $shape\_h1$ is *on the top* of the instance $shape\_s1$, $shape\_g1$ is *on the top* of the instance $shape\_s1$.

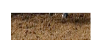
1	2	3	4	5
				

Tabella 5.1: The Images and SubImage Instances

According to our theory, the previous concepts are illustrated in Table 2. From the above

C_{IM}	C_{HM}	C_{Ag}
Texture	Person	Jockey
Color	Grass	Racehorse
Shape	Horse	Horse-race-track
	Sand	Track

Tabella 5.2: Aggregate, High Media, Intermediate Media Concepts

example, we explicitly note that in this theory we have *a full integration of data and knowledge* *levels*. We also note that, although only a few spatial relations have been described, such as on the top of, our model is sufficiently general to implement different spatial and geometrical relations (for example, direction, orientation and so on). In Figure 5.5 and 5.6, a portion of the *image ontology* related to the running example is depicted, without axioms and assertions on the general knowledge involved in the running example.

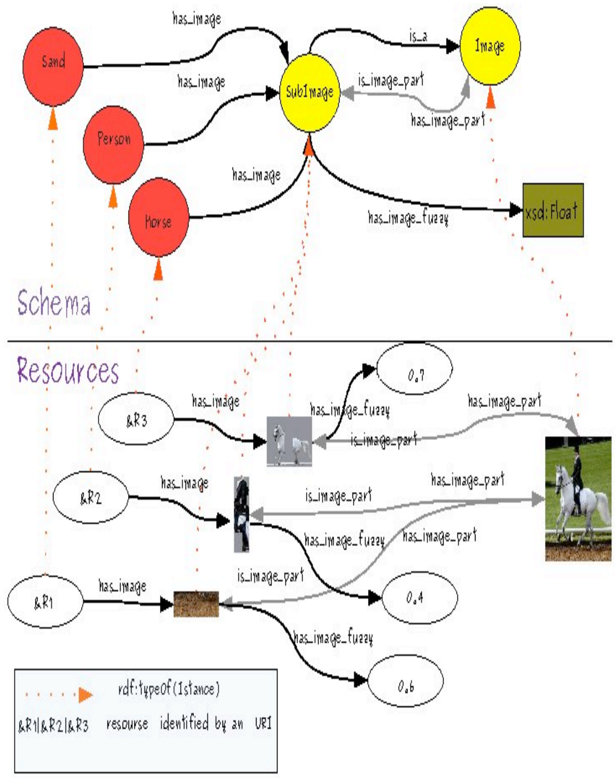


Figura 5.5: The First OWL Snapshot of the Image Ontology for the Running Example

5.3 Data models

In this section how the multimedia data are stored according with the previous framework is described. We essentially have two choices: in the first one we store the data in OWL data model the second one in Relation data model. The difference among these solutions are related to the query

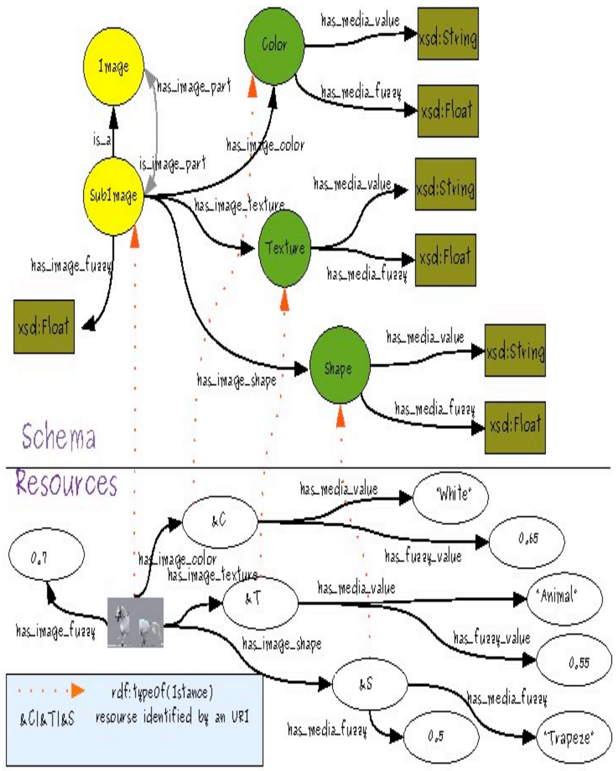


Figura 5.6: The Second OWL Snapshot of the Image Ontology for the Running Example

processing step. In OWL data model a SPARQL query language may be used to retrieve instances and it is the natural solution in order to create instances in the ontology refereed to the previous framework. In the Relation Data Model we have to store the data in relations and in this case we have to consider the different semantic between the ontology schema and the relational one.

5.3.1 Image OWL Data Model

In that case the data are defined as instances of the classes defined in the image framework theory. Note that we deal with multimedia data (image data), according with that type of data i create resources for high media, intermediate media, image and sub-image concepts and their data properties , instead i use the general knowledge only for reasoning and querying purposes. In this way in my ontology i have additional resource one of each images or sub-image that are used to take into account the relationships among the different sections in which an image is subdivided. In that case the image or sub-image individuals are the pointer on the key of image data base that are able to store blob files. In the following i have an snapshot of data related to Image OWL Data Model:

```
<owl:Class rdf:ID="Color"/>
   <Color rdf:ID="Color_25">
       <hasColorValue rdf:datatype="&xsd;string">Brown</hasColorValue>
   </Color>
   <Color rdf:ID="Color_26">
       <hasColorValue rdf:datatype="&xsd;string">Yellow</hasColorValue>
   </Color>
   <Color rdf:ID="Color_27">
       <hasColorValue rdf:datatype="&xsd;string">Blue</hasColorValue>
   </Color>
   ...
    <owl:Class rdf:ID="Image"/>
   <Image rdf:ID="Image_21">
       <partOf- rdf:resource="#SubImage_23"/>
       <partOf- rdf:resource="#SubImage_24"/>
       <partOf- rdf:resource="#SubImage_22"/>
   </Image>
   ...
     <owl:Class rdf:ID="SubImage"/>
   <SubImage rdf:ID="SubImage_22">
       <hasHighConceptFuzzyValue rdf:datatype="&xsd;float">0.8
        </hasHighConceptFuzzyValue>
       <hasColorReification rdf:resource="#Color_26"/>
       <hasColorReification rdf:resource="#Color_25"/>
       <partOf rdf:resource="#Image_21"/>
   </SubImage>
```

```
<SubImage rdf:ID="SubImage_23">
    <partOf rdf:resource="#Image_21"/>
</SubImage>
<SubImage rdf:ID="SubImage_24">
    <partOf rdf:resource="#Image_21"/>
</SubImage>
```

5.3.2 Image Relation Data Model

To best understand the proposed model, let us consider the image model provided by [33]; i particular, the authors provide a general *fuzzy image database model* that can be interpreted as extensions of traditional data models using fuzzy set theory and possibility theory [98]. The usual set theoretic operators can be extended to fuzzy sets in different ways, depending on the specific semantics associated with the fuzzy logical connectives.

The proposed NF^2 image database model is an extension of the standard NF^2: the considered domains and set operators are fuzzy domains and fuzzy set operators respectively.

Definition 5.3.1 (fuzzy tuple)**.** Let $D_1, \ldots, D_n$ be n domains. A fuzzy n-tuple is any element of the cartesian product $2^{D_1} \times \ldots \times 2^{D_n}$, being 2^{D_i} the fuzzy powerset of D_i, that is, the set of all fuzzy subsets of D_i.

According to the definition, any n-tuple is an array $\langle v_1, \ldots, v_n \rangle$, where each v_i is a set of elements from the corresponding fuzzy domain D_i.

Analogously, i consider as a special case the presence of a membership degree equal to 0, which represents the certain non-membership. The presence of a pair $\langle v, 0 \rangle$ in an attribute value does not give any information in addition to what we would have if i did remove that pair from the tuple. This is because domain values that do not appear in an attribute value are implicitly associated to the membership degree 0. To give an example, the attribute values $\{\langle \text{ green}, 0.5 \rangle\}$, and $\{\langle \text{ green}, 0.5\rangle, \langle \text{ green}, 0.0 \rangle\}$, provide the same information. Thus, i assume that our relations do not contain any such pair. This is not a restriction, since all the fuzzy values are either returned by some automatic vision systems or result from fuzzy algebraic operations.

A fuzzy relation schema is used to associate attribute names to the domains of tuples.

Tabella 5.3: An NF^2 relation related to only to color and shape features of the motivating example

File	Color	Shape	Content
RunningImage.jpeg	$\{\langle$ Green,0.6$\rangle$, $\langle$ Black,0.7 $\rangle$, $\langle$Gray,0.89$\rangle$ $\langle$ Brown,0.65$\rangle\}$	$\{\langle$ Rectangle ,0.8$\rangle$, $\langle$ Trapeze ,0.7 $\rangle\}$.	$\{\langle$ Jockey rode her horse in HorseRaceTrack ,0.7$\rangle\}$

Definition 5.3.2 (NF^2 relational schema)**.** A NF^2 relational schema is defined as a symbol R, that is the name of the NF^2 relation, and a set ($X = \{A_1, \ldots, A_n\}$) of (names of) fuzzy attributes. The schema is usually denoted as $R(X)$.

A NF^2 relation is an instance of a NF^2 relation schema; that is, a NF^2 relation is a set of fuzzy tuples, as stated in the following definition.

Definition 5.3.3 (NF^2 relation)**.** Let $R(\{A_1, A_2, \ldots, A_n\})$ be a relational schema. A NF^2 relation, defined over R, is a set of fuzzy tuples $t = \langle v_1, \ldots, v_n \rangle$ such that each v_i is a fuzzy subset of $dom(A_i)$

The fuzzy relation of Table 5.3 has four attributes: **File**, **Color**, **Shape** and **Content**. As can be seen in this example, each attribute value is a set of $\langle domain\_value, fuzzy\_value \rangle$ pairs.

Note that the above data model is particularly suited for representing: low level features, intermediate level features and high level features, together with the associated uncertainty. But in case 5.3 we have not the separation in the data model how the single regions of an image are described in terms of intermediate media and high media concept as described in the OWL data models. We can see the table 5.3 as result of a conclusion reasoning process over a domain ontology and owl image data model. I could also use image relation data model where each tuples are related with the sections of the images and its multimedia content as described by the image ontology framework:

File	Color	Shape	Content
Sec1RunningImage.jpeg	$\langle$ Gray,0.6 $\rangle$	$\langle$ Rectangle ,0.8 $\rangle$	$\langle$ Horse,0.7 $\rangle$

We can preserve the link between an image and its sections through an bridge simple relation with two attributes one for the image and the second for its section and the key is made by the two attributes; the link between the two relation is obviously made through the foreign constraint.

This model is very interesting for some reasons:

- one could store annotation of image from an external system as simple database update operation.
- one could run query using the typical plan optimization and index features of object-relation data base model

This model is also used to build an ontology after an image annotation process according with the following definitions.

5.3.2.1 Extending an NF^2 schema to an Extracted Image Ontology

So far I have described how a NF^2 schema can represent much more information than a classical relational schema. In this section I propose a novel strategy having the aim of extending the NF^2 schema *with* an ontology derived from it.

In this way, we can use the expressivity of the ontology language in order to express both explicit and newly derived implicit knowledge from the image data. In addition, we explicitly note that the use of an ontology as an intermediate level between the application modules and the data can enhance the integration capability of different and distributed sources,the semantic retrieval of the data and more efficient content-based querying. The main problem to address is to define a methodology for building an ontology schema $\mathcal{I}m\mathcal{O}$ starting from an NF^2 relational schema , re-defining in a suitable way standard *data base reverse-engineering* methodologies. In particular object-to-object relationship will be inferred from the foreign keys, while data properties will be derived from the other attributes. In the following we show an example of methodology that, without loss of generality, is particularly useful for describing the used approach. Let us consider a the NF^2 schema we have the following classes, defined under the name space *OntoImage*:

- the *OntoImage:Image* class and the *OntoImage:ImageBlob* class, created considering the key attribute *File* of the NF^2 table ;

- a set of *owl:class*, one for each non-key attribute (color, shape, content): the *OntoImage:ImageColor* class for the color attribute, the *OntoImage:ImageShape* class for the shape attribute, the *OntoImage:ImageContent* for the content attribute;

- a set of *owl:objectproperty*, connecting the previous classes: *OntoImage:has fuzzy color*, *OntoImage:has fuzzy shape*, *OntoImage:has fuzzy content*, *OntoImage:has fuzzy image* ;

- a set of *owl:dataproperty*: the *OntoImage:has degree value*, *OntoImage:has color*, *OntoImage:has shape*, *OntoImage:has content*, *OntoImage:has image* that has as range the Image Domain values .

Several problems have to be addressed in order to transform NF^2 relations into ontologies. First, we note that NF^2 schema is made up of n-ary relations in non-first normal form; differently, in standard ontology languages we can express only unary or binary predicates. In order to solve this problem, i use the owl-reification pattern, thus using - in a suitable way - some classes with some restriction over them, as described by a W3C working group note [95]. The second main problem is related to the identifiers of the objects belonging to such classes: we propose that these identifiers be generated by means of an appropriate invertible mapping, operating on NF^2 instances. In other words, we are addressing the well known *impedence mismatch* between a relational schema and an ontology schema. The mapping function can be implemented using a hash table – in NF^2 form – containing the pairs $\langle$ NF^2 istance, owl:class identifier $\rangle$, as follows:

NF^2 values	owl:classes identifier
Im1.jpeg	$\langle$ OntoImage:Image,324$\rangle$
$\{\langle$ Rectangle ,0.8$\rangle$, $\langle$ Trapeze ,0.7 $\rangle\}$	$\langle$OntoImage:ImageShape,346 $\rangle$
....	.....

Eventually, we give the following definition:

Definition 5.3.4 (Extracted Image Ontology)**.** *An Extracted Image Ontology $\mathcal{EImO}$ is a triple:*

$$\mathcal{EImO} = (\mathcal{ImO}, \mathcal{R}, \mathcal{M}) \tag{5.1}$$

where $\mathcal{ImO}$ is an image ontology, $\mathcal{R}$ is an NF^2 schema, and $\mathcal{M}$ is a mapping between $\mathcal{ImO}$ and $\mathcal{R}$.

We explicitly notice that the extended image ontology approach has more advantages than the one based on the NF^2 schema. Indeed, the use of ontologies allows to make inferences over image data, in order to retrieve semantically important facts from low level and intermediate level properties. In particular, the initial semantic content of an image data would be increase by associating (e.g. merging) the image ontology with a domain ontology or a thesaurus, Note that the connection between terms and concepts could be weighted in different ways, as based on concept affinity describe in [85]. Accordingly the use of fuzzy domain ontologies is not required, once the degree of concept affinity determines the fuzzy truth value of the returned image.

5.4 Query Language In Image Ontology Theory

In the above discussed framework, the extensional level of our ontology is made of a set of data that could be considered as set of descriptions of image or sub-image instances in terms of values of high-media or intermediate-media concepts or predefined relations. Each data could be stored in one of the model defined in 5.3. The query process algorithm changes according to the selected image data model, infat we have to different eventual languages in query plan execution a SPARQL that is based on Basic Graph Pattern Matching and the SQL that is based on tuples calculus. Both the approaches are followed by a ranking algorithm that build a suitable order based on the fuzziness value of the data. In this section I will describe how to retrieve the instances given a suitable query language. From a theoretical point of view, we focused on a particular class of queries, called *conjunctive query*. Given this kind of queries, we first give a definition of *high media concept conjunctive query*:

Definition 5.4.1 (High Media Concept Vision Query). *Given an $conj(x_1)$, that is an high media concept, the High Media Concept Vision Query is a conjunctive query in the following datalog form:*

$$\begin{aligned} q_m(x_1, x_2, x_3, x_d) \quad \leftarrow \quad & conj(x_1) hasVisionContent(x_1, x_2) SubImage(x_2) \\ & partof(x_2, x_3) Image(x_3) hasHighConceptFuzzyvalue(x_2, x_d) \end{aligned}$$

where $hasVisionContent$, $SubImage$, $Image$, $partof$, $hasHighConceptFuzzyvalue$ are the predicate belonging to the $\mathcal{K}_{IM}$.

Definition 5.4.2 (High Media Concept Query Formulation)**.** *Given a conjuntive query in the form of the standard datalog one for a single high media concept* $conj(x_1)$ *:*

$$q(x_1) \leftarrow conj(x_1)$$

High Media Concept Query Formulation is a transformation ϕ*:*

$$\phi : q(x_1) \Rightarrow q_m(x_1, x_2, x_3, x_d)$$

where $q_m(x_1, x_2, x_3, x_d)$ *is a High Media Concept Vision Query for the high media concept* $conj(x_1)$.

The main idea, that underlies this definition, is that we can retrieve the image and sub-image related to a given high media concept rather than the individuals of high media concept. I am now in a position to introduce a simple relational schema that i will use inside the query algorithm, it is called $Image - Table$ schema and it has the following table format:

Image	SubImage	FuzzyValue
im1.jpg	sub1.jpg	0.6
im1.jpg	sub2.jpg	0.3
im2.jpg	sub1.jpg	0.4
im3.jpg	sub4.jpg	0.8

In this table are reported the image and related sub-images with fuzziness that are the representation of a given high media concept retrieved applying the ϕ.
I use the term 5.4 to refer at that table or at its projection.

Definition 5.4.3 (SpatialTrasformation)**.** *Given a binary predicate R(x,y) that expresses an action or an general relation among concepts in the domain ontology, there is a transformation* χ*:*

$$\chi : R(x, y) \Rightarrow S(x, y)$$

being S(x,y) a spatial relation belonging to R_S= { $onTheTopOf$, $onTheLeftOf$, $onTheRightOf$, $onTheBottomOf$,$SpatialDisjoint$, $SpatialOverlap$, $onTheLeftTopOf$, $onTheRightTopOf$, $onTheLeftBottomOf$, $onTheRightBottomOf$ }

Definition 5.4.4 (Query Hyhypothesis). *In order to describe the query algorithm at this stage, some hypothesis essential for the right execution of the next query algorithms are introduced:*

- *We do not take into account the concrete domain predicate described into domain ontology (preliminary semplification).*
- *The axioms in the domain ontology don't have to be recursive.*
- *Each concept in the domain ontology has to be redefined in terms of high media concepts or aggregate media concepts,to do this i can use the algorithm 1 that returns a list of concepts that needs to introduce more axioms related to image ontology framework.*
- *The query can't use negation of concept or relationship and also the negated concepts in domain ontology are not take into account.*
- *The χ transformation is defined for each relation in the domain ontology that is not a spatial one.*
- *I have two kinds of relations obtained from the χ transformation. The first relates in spatial way all the simple subimages and these are codified by relations in the owl image data model and the second relates in spatial way two aggregate regions by means of a proper key. The aggregate regions are the regions made by two or more simple subimages.*
- *I choose the Owl Image Data Model as base model that store the main multimedia information and an image table to evaluate the intermediate results of the query evaluation algorithm.*
- *The universal quantification qualification is used only for high media concepts.*

5.4.1 Query Algorithms

In this subsection the algorithms used to evaluate queries over the multimedia resources defined as describe in the image ontology theory are described. The $MainQueryAlgorithm$ 2 algorithm is the main algorithm and it involves tree sub-algorithms $rewriting, queryPlan, queryEvaluation$.

Algorithm 1: $getMediaUndefinedConcept$ algorithm

Input: $\mathcal{T}, \mathcal{L}_{HM}$

$\mathcal{T}$ is the reference Tbox .

$\mathcal{L}_{HM}$ is the list of High Media concept

Output: $\mathcal{L}^*_{Ag}$,

$\mathcal{L}^*_{Ag}$ is the the set of aggregate concepts that one have to define in terms of high media concept

begin
- $\mathcal{L}^*_{Ag} = \emptyset$
- **foreach** $\psi : \mathcal{T} \models \psi$ **do**
  - **foreach** $C\ defined\ in\ \mathcal{T}\ \wedge\ C \notin \mathcal{L}_{HM}$ **do**
    - **if** $!(\ C \equiv \psi\ \wedge\ (\forall\ C^*\ defined\ in\ \psi \rightarrow C^* \in \mathcal{L}'_{HM} \subseteq \mathcal{L}_{HM}))$ **then**
      - $\mathcal{L}^*_{Ag} = \{C\} \cup \mathcal{L}^*_{Ag}$
    - **end**
  - **end**
- **end**
- return $\mathcal{L}^*_{Ag}$

end

- $rewriting$ algorithm 3 has the aim to redefined each concept inside the query or the given reference domain ontology in terms of High Media Concept, the result of that algorithm is a query formula that involved only spatial relationships and high media concepts
- $queryPlan$ is the common algorithm used to create a tree structure where each node is a TreeNode ad described in ref used to evaluate the query
- $queryEvaluation$ is the core query algorithm 4 that combines sparql query on the leaf of tree structure and relational query over imageTable on generic TreeNode and applies some joint operations on intermediate imagetable results according to the previous query plan.

Definition 5.4.5 (TreeNode)**.** *The TreeNode is an Abstract Data Structure that has the following internal fields:*

Algorithm 2: $MainQueryAlgorithm$ algorithm

Input: $\mathcal{T}, Q$

$\mathcal{T}$ is the reference Tbox,

Q is the query

Output: IT,

IT is a table of images (tableImage) that satisfy the given query

begin

$Q' = rewriting(Q, \mathcal{T})$

$TreeS = queryPlan(Q')$

$IT = queryEvaluation(getRoot(TreeS))$

return IT

end

- *a γ formula that described a concept to be queried according with syntax and semantic of Image Ontology Theory and the hypothesis in 5.4.4*
- *a logic operator op $\in \{\cap, \cup\}$*
- *a set of pointer to other TreeNodes, called $\mathcal{P}$.*

and the following interface :

- $setOperator(op)$*, setQuery(*γ*),* $setPointerSet(\mathcal{P})$ *that are function used to build an instances of TreeNode.*
- *getOperator() return the logic operator inside the TreeNode.*
- *getPointerSet() return the* $\mathcal{P}$.
- *getQuery() return the γ formula.*

I algorithm 3, the rewriting query algorithm for a query formula is depicted, in order to obtain a query to evaluate on the Owl Image Data Model described above. It uses the following functions:

- $getEquivalentAxioms$ is function that applies on $\mathcal{T}$ returns the set of equivalence axioms.

- $getSubsumptionAxioms$ is function that applies on $\mathcal{T}$ returns the set of subsumption axioms.

- $leftComponet$ is a predicate function that returns true if the concept is left member of the axioms otherwise false.

- $rightComponent$ is a predicate function that returns true if the concept is right member of the axioms otherwise false

- $reduceEquiv$ substitutes the input concept with the right component of the input equivalence axiom.

- $reduceSub$ substitutes the input concept with the left component of the input subsumption axiom.

In algorithm refalg:queryEvaluation, the $queryEvaluation$ algorithm is described; it makes use of the following functions:

- $isAllLeafFather$ is a predicate function that returns true if all the pointers of the $\mathcal{P}$ are pointers to leaf TreeNodes.

- $evalSparqlQuery$ is a function that applies the query formula inside the nodes as SPARQL according to the reference table 5.4.1

- $evalRelationQuery$ is a function that applies a suitable relational join query (explained in table 5.4.1)over the input image tables, belonging to the input set, according with the query formula.

- $isLeaf$ is predicate function that returns true if the input points to a leaf TreeNode otherwise false.

- $applyOperator$ is a function that converts the logic operator inside the TreeNode in a suitable joint relation query.

In table 5.4.1, the query patterns used to translate the query formula obtained by query plan in sparql query are described. In this table, the concepts involved are high media concepts and the queries are the leafs of the query plan tree.

These patterns are used in the $evalSparqlQuery$ function in the algorithm 4 and they are the implementation of High Media Concept Query Formulation . The results of these queries are relation tables in the Image Table schema which expresses a relation among images, their sub-images and their fuzzy values.

When the query TreeNode has as concept an aggregate one, we use the function $evalRelationalQuery$; that means that we use a transitional relation to store the results related to the images that represent the aggregate concepts. In table 5.4.1 there are the queries patterns used in this function. According with the hypothesis with aggregate concept i have only existential qualified quantification. The main idea is to take into account the spatial relationship among image complex regions by means of relation joint operation.

The function $applyOperator$ has as input a set of Image Tables and gives as result a new Image Table. This result is built according with the logic operator given in input:

- if the operator is a $\cup$ i have the a union of all tuples of the input image tables.
- if the operator is s $\cap$ i have a joint operation among the attribute identified the image, i retrieve the key identified the aggregate regions and then i apply an appropriate membership function to compute a unique fuzzy values for each tuples according to the Image Table Schema.

Note that if the query involved more that one concept linked by relations, we can apply the above described process for each concept; then we can use the relation among concepts to build a join among the results Image Table.

5.5 An ontological system for Image Retrieval

So far I have described the theoretical aspects of multimedia ontologies. In order to describe the effectiveness of my approach, I also present a prototypical system for image retrieval that extensively makes use of the proposed ontological framework. As depicted in Figure 5.7, the system is made up of four main layers:

- The *Graphical User Interface* is used as administration panel for the whole system: in particular, it provides a query editor and two navigation interfaces for visualizing the concepts specified in the ontologies. We use this module to compute two kinds of query: *a query by example* (QBE), where an image input is given and the most similar images are retrieved and *a Textual Query* (TQ), where a simple textual query is used to express concepts and relations within the knowledge base.

- The *Automatic Knowledge Discovery Engine* contains some novel algorithms to extract regions inside an image; it uses image features related to those regions to derive high media concepts and their related membership degree. It also associates with them some intermediate media concepts, together with their associated membership degree. This engine has three main modules: The *Segmentation* module has the aim to subdivide an input image in a set of regions that are consistent with respect to predefined criteria; we use an intelligent segmentation algorithm based on an improvement of K-means clustering on color distribution. This procedure uses some tricks in order to improve the accuracy of the segmentation, i.e suitable space color (Lab), a pre-cluster that has the aim to find the optimum K^* for the K-means based on density measures of regions and region cuts in order to delete the useless parts. The *Feature Extraction* module is used to derive some features related to colors, shapes and textures. The features are extracted using the Java API interface of Oracle Multimedia 11g. This API is able to detect a image global vector feature, called image signature, and we use that signature as input in its similarity function to derive which kinds of values the images have in terms of shape, feature, texture. The similarity function gives a score about the similarity with the image stored in the database and we consider them as a pattern of shape, color or texture. Inside this module, we have developed some functions to detect several spatial relationships among the shape associated with regions. The *Classification* module that implements a hierarchical classification strategy is based on Support Vector Machine classifier which is able to recognize the high media concepts related to each region using some hierarchical processing on the global feature vector returned from the *Automatic Knowledge Discovery Engine* module.

- *Query Engine* is the main query engine module and it is made of two submodules: *Query Formulation* is used to compose a SPARQL query involving the concepts that are derived from TQ or that is extracted from the previous *Automatic Knowledge Discovery Engine* module in case of QBE. *Query Processing Engine* has the aims to execute this set of queries on the media data stored in *Data Layer* and it combines the output results using the membership values associated with each high and intermediate media concepts deriving a ranking according to the well-known Fagin algorithm [50].

- The *Data Layer* is formed by: An *OWL* repository that contains all the axioms and assertion of the image ontology. An *Image Database* used to store images, regions, pattern images used by Oracle Multimedia.

All the modules have been implemented in Java Technology, using frameworks such as WEKA, JENA, and external c++ routines for the image segmentation algorithm. More details about computer vision algorithm could be found in [19].

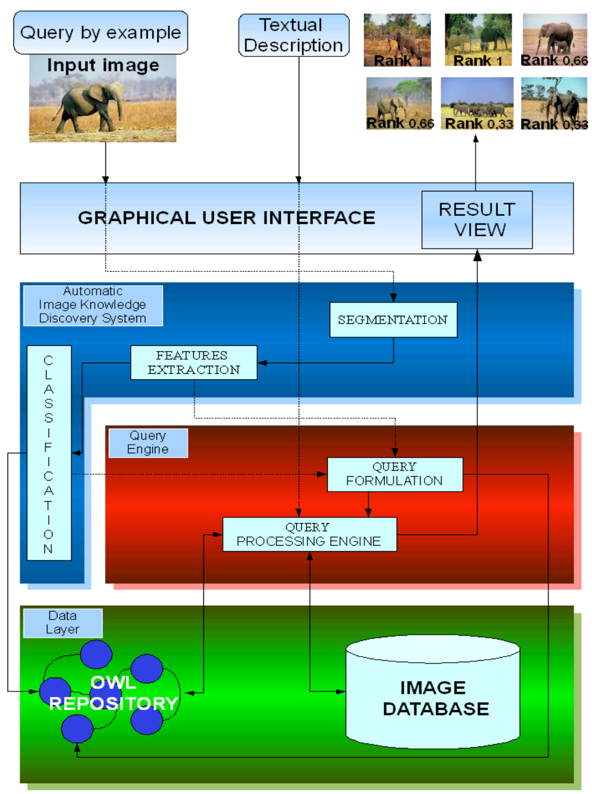


Figura 5.7: The System

5.6 Experimental Setting and Results

I have tested the system described above on a database of about 600 images (200 from standard corel database and 400 extracted from google image engine). I have developed an ontology with 8 high media concepts : sky, water, savannah, grass, elephant,horse, sand and person and 10 axioms built over them. The images are subdivided in:

- 263 containing horse and/or grass and/or sky and person and sand
- 337 containing elephant and/or grass and/or sky, and/or water and/or savannah.

The test have the aim to derive:

- the precision and recall of the system given a textual query.
- the accuracy of the image knowledge discovery system

For the classification purpose, I choose the 30% of images that are used in the training phase for the Image Knowledge Discovery System classifiers. For the queries by example, note that the results are related with the once obtained from the classification algorithm; in this way it is not used a general ontology in the query evaluation process because no aggregate concepts are involved.
The following table express the results(Precision and Recall)for the Automatic Image Knowledge Discovery: Results for the query engine:

Predicati	Elephant	Horse	Savannah	Sky	Water	Grass	Person	Sand	Precision
Elephant	134	8	2	0	4	0	0	0	0.91
Horse	3	96	2	3	0	0	3	0	0.90
Savannah	2	2	101	3	0	0	0	8	0.87
Sky	0	4	0	104	3	0	0	0	0.94
Water	1	0	0	0	51	0	0	0	0.98
Grass	0	0	5	0	2	220	0	6	0.97
Person	0	3	0	0	0	0	50	0	0,94
Sand	0	0	15	0	0	6	0	30	0.68
Recall	0.96	0.87	0.92	0.95	0.85	1	0.70	0.83	

Figura 5.8: Precision and Recall for Automatic Image Knowledge Discovery

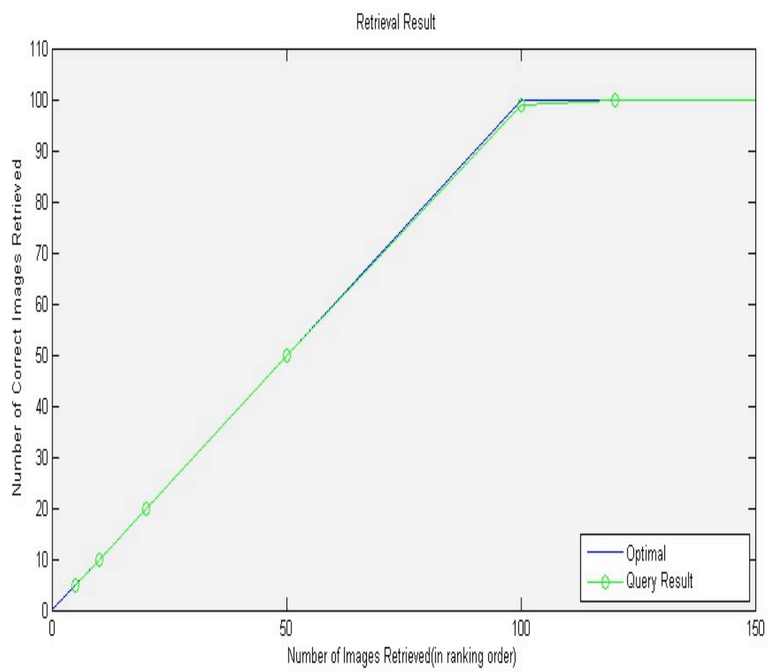


Figura 5.10: The results for textual query : $AfricanElephant$

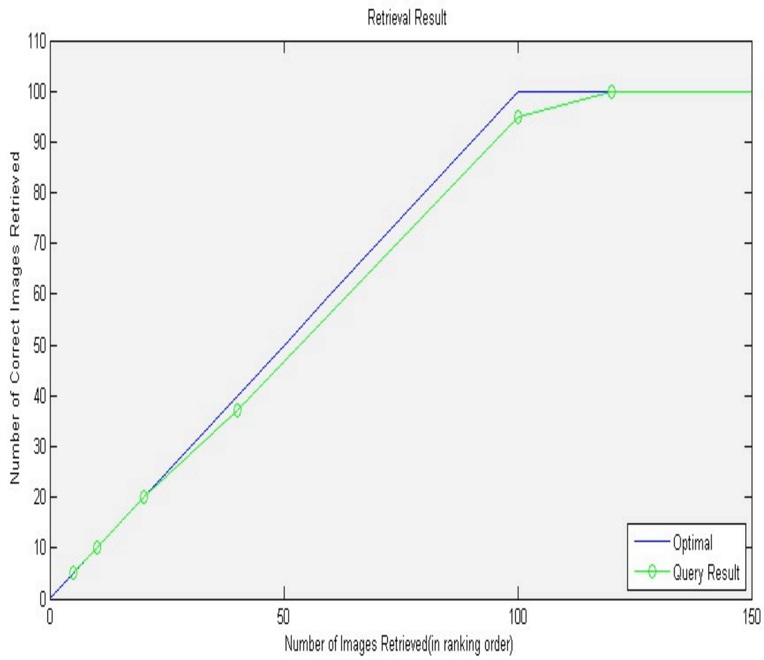


Figura 5.9: The results for textual query : $Jockey$

Note that we obtain an average Recall of 0.88% and average Precision of 0.90%. The figures 5.10 and 5.9 compare the number of retrieved images in ranking order against the number of correct images in ranking order.

An optimum function that represents the ranking order designed by humans is also depicted and we compare this optimal function to the one obtained from the previous system.

The queries use a set of axioms defined over the high media concepts:

- $Jockey$:

$$\text{Jockey} \equiv \text{Person} \sqcap \exists \text{ ride.(RaceHorse)}$$
$$\exists \text{ rounded.(Grass} \sqcap \text{ Sand)} \sqsubseteq \text{ HorseRacerTrack}$$
$$\text{RaceHorse} \equiv \text{Horse} \sqcap \exists\ located\_in \text{ HorseRacerTrack}.$$

- $AfricanElephant$:

Elephant $\sqcap\exists$ lives.Savannah $\sqsubseteq AfricanElephant$.

The ranking order in the case of $Jockey$, as described in figure 5.9, doesn't follow the optimum ranking because, as depicted in table 5.6, some high media concept involved in the definition of $Jockey$ have lower recall or precision than the others.

Algorithm 3: $Rewriting$ algorithm

Input: $\mathcal{T}, Q, \mathcal{L}_{HM}$

$\mathcal{T}$ is the reference Tbox,

Q is the query,

$\mathcal{L}_{HM}$ the set of high media concepts.

Output: Q,

is the rewriting query

```
begin
    while !(∀ c belonging to Q → c ∈ L_HM) do
        foreach c belonging to Q do
            foreach A_EQUI ∈ getEquivalentAxioms(T) do
                if leftComponet(A_EQUI, c) then
                    Q = reduceEquiv(Q, c, A_EQUI)
                end
            end
            foreach A_SUB ∈ getSubsumptionAxioms(T) do
                if rightComponet(A_SUB, c) then
                    Q = reduceSub(Q, c, A_SUB)
                end
            end
        end
    end
    return Q .
end
```

Algorithm 4: $queryEvaluation$ algorithm

Input: $TreeNode, \mathcal{L}_{HM}$

$TreeNode$ is a tree node that are base object of a tree structure.

$\mathcal{L}_{HM}$ the set of high media concepts.

Output: IT,

IT is a table of images (tableImage) that satisfy the given query

```
begin
    P = getPointerSet(TreeNode).
    ImageTableSet = ∅
    if isAllLeafFather(P) then
        foreach p ∈ P do
            I = evalSparqlQuery(p)
            ImageTableSet = ImageTableSet ∪ {I}
        end
        ImageTable = applyOperator(ImageTableSet, getOperator(TreeNode))
        return ImageTable.
    end
    else
        foreach p ∈ P do
            if isLeaf(p) then
                I_1 = evalSparqlQuery(p)
                ImageTableSet = ImageTableSet ∪ {I_1}
            end
            else
                I_2 = queryEvaluation(p, L_HM)          /* Recursive Call */
                I_2 = evalRelationQuery(I_2)
                ImageTableSet = ImageTableSet ∪ {I_2}
            end
        end
        ImageTable = applyOperator(ImageTableSet, getOperator(TreeNode))
        return ImageTable.
    end
end
```

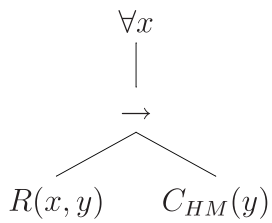
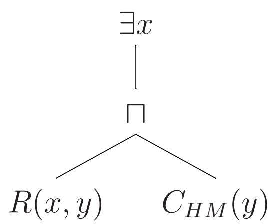
<table>
<tr>
<td></td>
<td>PREFIX ex: <http://example.com/#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?Image ?subImage ?fuzzy
WHERE {
?Image ex:partof- ?subImage.
?subImage ex:hasHighConceptFuzzyValue ?Fuzzy.
Optional {
?1 :$\chi(R)$?x.
FILTER (?1 a ex: C_{HM})
}
FILTER(!(Bound (?x)).
}</td>
</tr>
<tr>
<td></td>
<td>PREFIX ex: <http://example.com/#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?Image ?subImage ?fuzzy
WHERE {
?Image ex:partof- ?subImage.
?subImage a _:R .
_:R a owl:Restriction .
_:R owl:onProperty ex:$\chi(R)$.
_:R owl:someValuesFrom ex:C_{HM}.
?subImage ex:hasHighConceptFuzzyValue ?fuzzy.
}</td>
</tr>
<tr>
<td>$C_{HM}(x)$</td>
<td>PREFIX ex: <http://example.com/#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT ?Image ?subImage ?fuzzy
WHERE {
?subImage
?Image ex:partof- ?subImage.
?subImage ex:hasVisionContent- ex: C_{HM}.
?subImage ex:hasHighConceptFuzzyValue ?fuzzy.
}</td>
</tr>
</table>

Tabella 5.4: Sparql Query Pattern for the leaf of Query Plan Tree. I use the $\chi(R)$ to mean the translation of general relations to the spatial one

$\exists x$ $\sqcap$ $R(x, y)$ $C_{Ag}(y)$	The y is a key,that is used to understand the spatial relation among aggregate image regions. The $C_{Ag}(y)$ is an $ImageTable(Img, IdRegion, FuzzyV)$ We thus have the following result queries: $Im1$=$\rho_{Im1 \leftarrow ImageTable}(ImageTable)$ $Im2$=$\rho_{Im2 \leftarrow ImageTable}(ImageTable)$ I=$\sigma_{Im1.Img=Im2.Img}(Im1 \bowtie_{Im1.IdRegion=\chi(R).IdRegion1} \chi(R) \bowtie_{\chi(R).IdRegion2=Im2.IdRegion} Im2)$ $I = \Pi^*(I)$ Π^* is a relation projection redefined in order to apply a suitable membership function on the two fuzzy attributes in order to obtained only one and to obtain an Image Table Schema. One of the choice is the min value.

Tabella 5.5: Relational Query Pattern for the intermediate node of Query Plan Tree. I use the $\chi(R)$ to mean the translation of general relations to the spatial one

Capitolo 6

Tex Ontology System

Text Ontology Stystem

6.1 Introduction

In this chapter the use of NLP techniques and ontologies as the core for building novel legal based information systems is desribed. In almost all legal traditional activities, most of the processes are characterized by the presence of paper documents that have to be properly managed: processed, archived and prepared for long term preservation. Despite the introduction of automatic tools, no significant reduction in the volume of paper documents created has been registered: an intense and extensive dematerialization activity is still necessary. This problem is not trivial, and it could be technically formulated as in the following: starting from a collection of unstructured documents, related to a particular bureaucratic process, (e.g., documents in public administration offices or legal notary documents or investigation reports), the dematerialization process implies the application of syntactic-semantic methodologies in order to automatically transform the unstructured legal document into a formally structured, machine readable document. The described process requires the use of different techniques from interdisciplinary fields: in particular, several efforts have been done regarding *legal ontologies*, from both theoretical – in order to define legal lexical dictionaries – and application points of view, as for instance can be evidenced from the large number of e-gov initiatives in Europe – putting a great emphasis on the study of the *structures* and *properties* of legal information, as well on organization, storage, retrieval and dissemination within the context of legal environments. Note that, the main approaches are focused on describing either general knowledge models or general NLP systems. Differently, to the best of our knowledge, my work is one of the first attempt of proposing a unifying model in the notary domain and a complete semantic processing system that transforms legal documents into structured RDF files. In order to describe the peculiarities of our work, throughout the paper we will use a running example, as discussed in the following.

Example 6.1.1 (Notary Documents)**.** *Let us consider the Italian juridic domain, and in particular the notary one: a notary is someone legally empowered to certify the legal validity of a document. Let us suppose to analyze a* buying act. *In real estate market, in Italy and also in some other european countries, when someone has the intention of buying or selling a property, such as houses, pieces of lands and so on, a notary document, certifying the property transaction from an individual to another one, is signed. Such document is generally composed by an* introduction part *containing the caption, a part containing the* biographical data *of the individuals involved in the buying act, a section containing* data about the property *and a sequence containing several rules regulating the sales contract. Consider for example the Italian sales contract fragment, proposed in figure 6.1; an Italian reader can easily detect the areas concerning the caption, the personal data and the property attributes. In a similar way, we propose a system that: i) detects the several sections containing relevant information (segmentation), and ii) transforms the unstructured information within the retrieved section into a structured document, by means of iii) lexical, structural, and domain ontologies.*

This document is explained in figure 6.1. Note that the methodology could be applied for any domain which documents are legal ruled.

This chapter is organized as follows: in section 6.4, the general system architecture is outlined; section 6.2 describes the theory underlying our work, in particular the ontological levels for legal information management; the RDF document building strategy is described in section 6.3; some details about system architectures are presented in section 6.4 and eventually some results are discussed in section 6.5.

6.2 Theory

In this section several definitions that are at the basis of several proposed algorithms will be given. In the legal domain, almost all the documents is still written using natural languages. Even though, the unstructured form of document follows a well determined sequence: in a notary act, for example, the notaries use a certain subset of natural language and in addition they use a certain pre-defined structure, that can be codified by laws or normative rules. For these reasons, we say

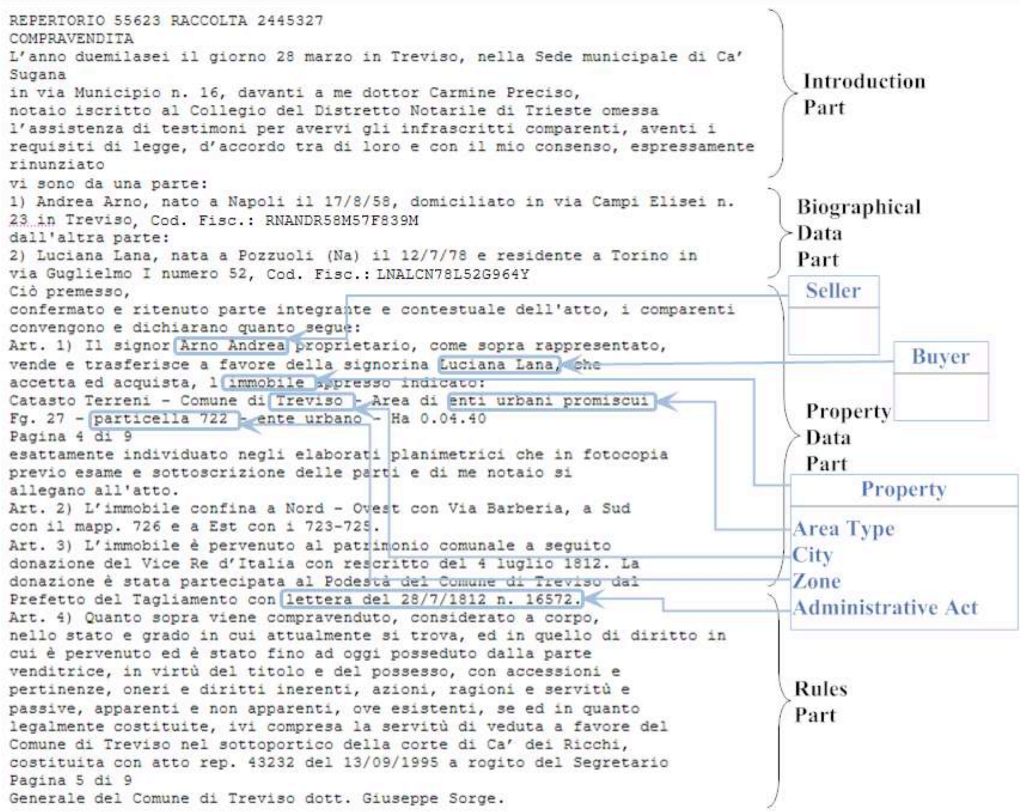

REPERTORIO 55623 RACCOLTA 2445327
COMPRAVENDITA
L'anno duemilasei il giorno 28 marzo in Treviso, nella Sede municipale di Ca' Sugana
in via Municipio n. 16, davanti a me dottor Carmine Preciso,
notaio iscritto al Collegio del Distretto Notarile di Trieste omessa
l'assistenza di testimoni per avervi gli infrascritti comparenti, aventi i
requisiti di legge, d'accordo tra di loro e con il mio consenso, espressamente rinunziato
vi sono da una parte:
1) Andrea Arno, nato a Napoli il 17/8/58, domiciliato in via Campi Elisei n. 23 in Treviso, Cod. Fisc.: RNANDR58M57F839M
dall'altra parte:
2) Luciana Lana, nata a Pozzuoli (Na) il 12/7/78 e residente a Torino in
via Guglielmo I numero 52, Cod. Fisc.: LNALCN78L52G964Y
Ciò premesso,
confermato e ritenuto parte integrante e contestuale dell'atto, i comparenti
convengono e dichiarano quanto segue:
Art. 1) Il signor Arno Andrea proprietario, come sopra rappresentato,
vende e trasferisce a favore della signorina Luciana Lana, che
accetta ed acquista, l'immobile appresso indicato:
Catasto Terreni - Comune di Treviso - Area di enti urbani promiscui
Fg. 27 - particella 722 - ente urbano - Ha 0.04.40
Pagina 4 di 9
esattamente individuato negli elaborati planimetrici che in fotocopia
previo esame e sottoscrizione delle parti e di me notaio si
allegano all'atto.
Art. 2) L'immobile confina a Nord - Ovest con Via Barberia, a Sud
con il mapp. 726 e a Est con i 723-725.
Art. 3) L'immobile è pervenuto al patrimonio comunale a seguito
donazione del Vice Re d'Italia con rescritto del 4 luglio 1812. La
donazione è stata partecipata al Podestà del Comune di Treviso dal
Prefetto del Tagliamento con lettera del 28/7/1812 n. 16572.
Art. 4) Quanto sopra viene compravenduto, considerato a corpo,
nello stato e grado in cui attualmente si trova, ed in quello di diritto in
cui è pervenuto ed è stato fino ad oggi posseduto dalla parte
venditrice, in virtù del titolo e del possesso, con accessioni e
pertinenze, oneri e diritti inerenti, azioni, ragioni e servitù e
passive, apparenti e non apparenti, ove esistenti, se ed in quanto
legalmente costituite, ivi compresa la servitù di veduta a favore del
Comune di Treviso nel sottoportico della corte di Ca' dei Ricchi,
costituita con atto rep. 43232 del 13/09/1995 a rogito del Segretario
Pagina 5 di 9
Generale del Comune di Treviso dott. Giuseppe Sorge.

Figura 6.1: An example of Notary Documents

that notaries manage *semi-structured documents* written in a simplified natural language. These considerations are at the basis of the following preliminary definitions.

Definition 6.2.1 (Structure-UnarySet)**.** *Let us give a domain* $\mathcal{D}^S$*; a* $Structure\text{-}UnarySet$ $(\mathcal{SU})$ *over* $\mathcal{D}^S$ *is the set of unary predicates, called* $structure\text{-}concepts$ (sc)*,*

$$\mathcal{SU} = \{sc_1, ...sc_n\}$$

$sc_i \in \mathcal{D}^S, i \in \{1..n\}$

Definition 6.2.2 (Document-Structure-UnarySet)**.** *A* $Document\text{-}Structure\text{-}UnarySet$ $(\mathcal{DS})$ *is a non empty subset of* $\mathcal{SU}$ *containing all the necessary concepts for defining the structure of a given document according to a experts domain description.*

Definition 6.2.3 (Structure-BinarySet)**.** *Let us give a domain* $\mathcal{D}^S$*; a* $Structure\text{-}BinarySet$ $(\mathcal{SR})$ *over* $\mathcal{D}^S$ *is the set of binary predicates, called* $structure\text{-}relations$ (sr)*,*

$$\mathcal{SR} = \{sr_1, ...sr_m\}$$

$sr_i \in \mathcal{D}^S$, $i \in \{1..m\}$.

Example 6.2.1 (Structures example)**.** *According to definition 6.2.1, a possible* $\mathcal{SU}$ *for the italian notary documents considered is:* $\{person, component, date, location, organization, article, section, biographical\_section, notary\_section, buying\_act, parties\_section\}$*; using example 6.1.1, according to definition 6.2.2,* $\mathcal{DS}$ *can be* $\{article, section, biographical\_section, notary\_section, buying\_act, parties\_section\}$ *, and according to definition 6.2.3,* $\mathcal{SR}$*=*$\{has\_number\_act, is\_part\_of, is\_kind\_of, has\_name, has\_surname, has\_section, has\_article\ has\_sold, is\_born\_at, has\_SSN\}$

The following definition also stands.

Definition 6.2.4 (Base-Document)**.** *Let a* $Paragraphs\text{-}Sections$ (S^P) *be the set of textual line inside a document. A* $Base\text{-}Document$ $(\mathcal{D}^B)$ *is:*

$$\mathcal{D}^B = \{S_1^P, ..., S_m^P\}$$

$S_i^P \cap S_j^P \supseteq \emptyset$,$i, j \in \{1..m\} \wedge i \neq j$.

In other word, a document is a set of overlapping text-areas; note that I can have different $\mathcal{D}^B$, depending on the different set of partition criteria used.

In order to capture the knowledge about the structure and the content of the document, let us describe the used ontologies, in terms of their intensional level. First I introduce the $TBox\text{-}Module$ ($\mathcal{TM}$) that is used to characterize a fragment of a TBox $\mathcal{T}$:

Definition 6.2.5 (TBox-Module)**.** *Let* $\mathcal{T}$ *be* $TBox$*, a* $TBox\text{-}Module$ $\mathcal{TM}$*, is a set of axioms* χ *that are:*

- *logically correct: any formula that is provable in* χ *is provable in* $\mathcal{T}$
- *logically complete: any formula in the scope of* χ *that is provable in* $\mathcal{T}$*, should be provable in* χ*.*

We can now define $Tbox$ as a $Structure\text{-}TBox$

Definition 6.2.6 (Structure-TBox)**.** *A $Structure\text{-}TBox$ ($\mathcal{ST}$) is a finite set of axioms over concepts and roles belonging to $\mathcal{SU}$ and $\mathcal{SR}$ respectively, expressed according to the syntatic rules and the semantic of $\mathcal{SHOIQ}(D_n)$ description logic.*

This kind of intensional knowledge takes into account the document's *implicit* structure used from domain experts to write these legal documents. Considering the notary example, a $Structure\text{-}TBox$ for a $buyingAct$, may be formed by several axioms selected by a domain experts, e.g the "biographical-section" of a given document, that contains concepts and relations describing "name", "surname" of "person", "address" and "security social number", is represented with the following axioms:

$buying\_act \equiv = 4has\_section.section$,

$biographical\_section \sqsubseteq section$,

$biographical\_section \equiv \geqslant 2has.person$,

$person \equiv \exists hasName \sqcap \exists hasSurname \sqcap \exists hasSSN \sqcap \exists is\_born\_in.city$.

These are the set of axioms of the $Structure\text{-}TBox$, i.e. the $TBox\text{-}Module$ related to the $biographical\_section$ of the $buyingAct$. Each $TBox\text{-}Module$ has to be characterized by means of a proper key.

In particular, at each key is assigned a feature set associated to regular expressions, keywords occurrences, entity recognition, and a related *score* is computed considering the positive matching in the feature set; we thus use the best score to detect what is the best module that describes the given fragment. In the following, we will give several definitions used to structure the information related to a document.

Definition 6.2.7 (KnowledgeKey-Function)**.** *A $KnowledgeKey\text{-}Function$ (ψ) is an invertible function:*

$$\psi\colon \mathcal{TM} \longrightarrow k \in \mathcal{K}$$

k being a unique key used to identify $\mathcal{TM}$ and $\mathcal{K}$ the set of these keys.

In the notary example, $\mathcal{TM}$ is identified by a key k^* and the related feature is $feat(k^*)$ $=\{CODICE\backslash s * FISCALE\backslash s * [A-Z0-9\backslash s], nat[o,a], an\_entity\_of\_type\_person\}$; i.e. a mixture of regular expressions and named entity recognition.

I am now in a position to introduce others concepts related to further levels description of a document D.

Definition 6.2.8 (Structured-Document)**.** *A $Structured\text{-}Document$ $\mathcal{SD}$ is a set of 2-tuples:*

$$\mathcal{SD}=\{\langle S^P_1, k_1\rangle, ..\langle S^P_h, k_h\rangle\}.$$

S^P_i*, and* $k_i \in \mathcal{K}$ $i \in \{1 \dots h\}$ *being $Paragraphs\text{-}Sections$ and a knowledge key (obtained by applying the ψ function to a $\mathcal{TM}$) respectively.*

Note that different $\mathcal{TM}$ (domain, structure, or lexical) may point to the same $Paragraphs\text{-}Sections$; so, some tuples in $\mathcal{SD}$ may have the same $Paragraphs\text{-}Sections$ and different keys. In our vision, the knowledge related to the notary legal domain should be expressed in a *domain ontology*, including a *structural ontology*, together with a *lexical ontology*.

For example, in an italian notary act one could use a specific legal domain ontology built over the top of JurWordNet [122], several ontologies describing the structure of a particular juridic document produced by domain experts, in addition to a lexical ontology based on ItalWordNet [103].

Given these tree different kinds of knowledge, i.e. structural, domain and lexical knowledge, we propose to use the first one for text segmentation aims, the second and third ones to infer more specific concepts related to the semantic content of the documents: in particular, the individuals and the keywords extracted from a section are interpreted as concepts and the relative relations are then inferred using both domain and lexical ontology modules.

Eventually, the extensional knowledge contained in each section in which the document is subdivided is also represented as follows:

Definition 6.2.9 (Knowledge-Chunk)**.** *A $Knowledge\text{-}Chunk$ (kc) is an RDF triple kc=$\langle r, p, a\rangle$, r being a resource name, p being a property name, a being a value.*

Let us now introduce the last level of description of our legal document:

Definition 6.2.10 (KnowledgeChunk-Document)**.** *Let D be a document; a $KnowledgeChunk\text{-}Document$ ($\mathcal{KC}^D$) is:*

$$\mathcal{KC}^D \in \{D, kc_1....kc_l\}$$

kc_i, $i \in \{1..l\}$ *being the Knowledge-Chunk and D the related document.*

For example for the "buyingAct", called $ID\text{-}Do$-01, I should have three Knowledge-Chunk:

Example 6.2.2 (Knowledge-Chunk).

$kc_1 =$ $\langle myxmlns{:}ID\text{-}Do\text{-}01, buyingAct{:}asset,$
$\text{“}Immobile''\rangle,$

$kc_2 =$ $\langle myxmlns{:}ID\text{-}Pe\text{-}01, foaf{:}name, \text{“}Ludovico''\rangle,$

$kc_3 =$ $\langle myxmlns{:}ID\text{-}Pe\text{-}01, buyingAct{:}seller,$
$myxmlns{:}ID\text{-}Do\text{-}01\rangle,$

$\mathcal{KC}^D =$ $\{ID\text{-}Do\text{-}01, kc_1, kc_2, kc_3\}$

$myxmlns$, $foaf$ and $buyingAct$ being predefined xml name space.

6.3 Algorithms

In this section I will describe the several algorithms that are used in our system.

In order to detect the parts in which every act is composed, we define and apply a text segmentation algorithm, assigning each extracted segments to a structured document, according to a *predefined schema*, characterizing the legal documents under study.

In order to extract simple fragments of the text, we use some partition rules, that are dependent from: *i)* normative prescriptions; *ii)* tradition of single notary schools; *iii)* common use of the single notary. A variety of rules may thus be detected, using several criteria formalized using real notaries expertises, more details can be found in [13].

In figure 6.2 an example of applying three partition criteria on the same act fragment is depicted.

Once several partitions are defined on a given text, we determine the optimal *act partition* in order to associate the most suitable act part to an appropriate ontology module, that contains the concepts and the relations to be extracted. We thus apply a scoring criterion, realized comparing the pattern extracted from each text segment with the concept contained in the ontology module. For text segmentation purpose, the following function is defined:

Definition 6.3.1 (Segm-Function). *A $Segm\text{-}Function$ (ρ) is a function that associates an element of $Base\text{-}Document$ to a $\mathcal{SD}$:*

$$\rho : \mathcal{D}^B \longrightarrow \mathcal{SD}$$

Note that a $Segm$-$Function$ may be implemented in a variety of way; in this dissertation, an association between an S^P and a k according to a minimum score computed comparing the patterns extracted from text and those represented by the key is proposed. An implementation of ρ function is given by algorithm 5.

Note that in this algorithm, $scoreVec$ is an array of scores; $getParagraphsSections$ takes in input a given Document and, using a certain partition criteria, returns a $Base$-$Document$; $structuredFunction$ has the role of matching a $Paragraphs$-$Section$ with one of $TBox$-$Modules$ in input and of retrieving the tuple having the best score together with the score itself; $getStructuredDocument$ computes the best $Structured$-$Document$ dinamically built considering those sections having the best sum of the scores previously computed. Once I have obtained the $Structured - Document$, I extract the $knowledge$-$chunk$s from the text as described in algorithm 6.

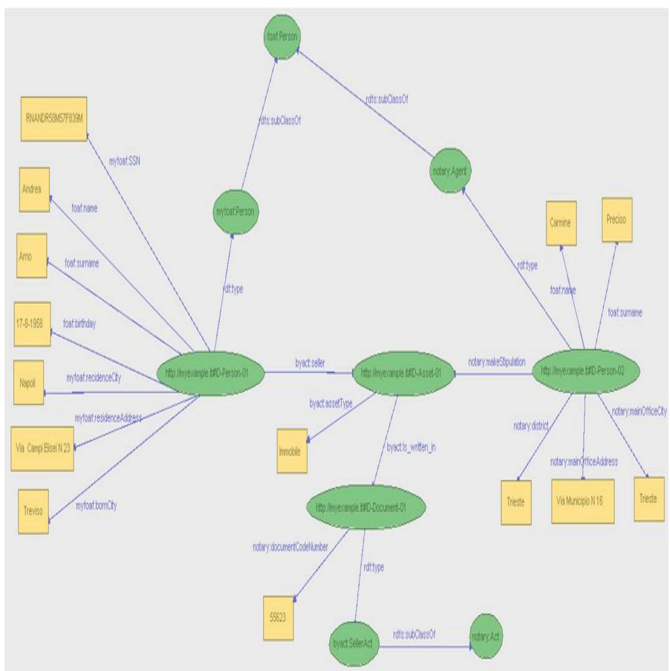


Figura 6.3: A section of RDF graph extracted from a Notary Act in fig 6.1

In this algorithm, the $InferenceProcedure$ extracts $knowledge\text{-}chunks$ from text using inference mechanism, concepts and relations extraction techniques. For example, a generic rule could be formed by a combination of token and syntactic patterns, in order to derive the instances of some concepts or relations, and eventually using subsumption on $TBox\text{-}Module$ for deriving more specific concepts.

Example 6.3.1 (RDF triples extraction)**.** *For the act reported in the example of figure 6.1, the system extracts, as shown in the figure 6.3, several triples from notary act between the notary and the people involved into the buying-selling process with their generalities and in particular who is the seller and who is the buyer, and the related relationship property.*

6.4 System Architecture

Figure 6.4 shows the architecture of the proposed system at a glance. I briefly discuss the core module of the system: The Information Extraction Module.
Its main functionalities are:

- *Structural analysis*: this functionality performs some processing of digital semistructured text. It takes in input the textual macrostructures which allow the recognition of text sections, according to the information provided by the structural ontology, that represents the organisation of the documents in the legal domain. The subdivision of the document into segments of text makes the further syntactic and semantic analyses more accurate.

- *Lexical analysis*: this functionality performs a syntactic-semantic annotation of the text by means of a labelling strategy; in particular, each text element is associated with a grammatical category (verb, noun, adjective and so on) and a syntactic role (subject,predicate, complement, etc.). In order to do that, several traditional NLP components are then used, i.e., a Stop Word List in order to eliminate the irrelevant words in the sentence, such as pronouns, articles and so on, a Stemmer to remove the more common morphological and inflexional forms from words in Italian language , a Part of Speach Tagger to detect the several grammatical parts of a sentence, and a Syntactic Analyser to recognise the logic-syntactic relation existing between "sintagms", to these aims, we use ontologies based on the ItalWordnet lexical database.

- *Semantic analysis*: this core functionality performs novel information extraction techniques, by means of structural, legal domain and lexical ontologies, this module detects concepts and relations among concepts. The proposed semantic analyzer produces a proper semantic annotation that is codified in RDF triples. In particular, it associates an appropriate concept to each discovered single entity.

In the described architectures, the Linguistic Analysis module uses standard and well-established NLP algorithms, while the semantic analysis tool for the extraction of the information represents novel approaches and techniques. The classification and segmentation modules are the standard modules of text mining application and they can implement classical techniques [21].

The figures 6.4 and 6.4 explain the process flow from the recognition of the entities to the retrieval of ontological patterns until the building of RDF triples.

6.5 Preliminary Results and Discussion

In order to quickly analyze the effectiveness of the proposed methodology, the algorithm has been tested over a collection of about 20 notary documents belonging to the categories of buying-selling document. Each document has been labeled by a notary practitioner in order to highlight the main information characterizing the document itself. The results of notary practitioner has been used as ground-truth. Despite the low number of documents, we preferred to give some preliminary results The practitioner's processing is still in progress: this is the first essential bottleneck of all the evaluation procedures over complex documents, i. e. the

presence of some experts for comparison aims. These results are given in terms of *Precision* and *Recall* defined as follows:

$$Recall = \frac{R}{R + RNR} \times 100$$

$$Precision = \frac{R}{R + NR} \times 100$$

R being the set of retrieved features, RNR the number of Relevant features Not Retrieved by the system and NR the number of Not-Relevant features retrieved by the system.

General Entity	Number	Special Entity	Number
Person	67	Notary	20
Place	123	Seller	28
Society	10	Buyer	24
Real Estate	20	Participant	51
SSN	43	Relation	234
Date	43		

The results shows a Precision 99% and a Recall 100%, that are relative values due to the dimension of data used. Despite the low number of analyzed acts, these experiments are very encouraging and are very useful to understand the power of the proposed methodology and to investagate where it needs any improvements. The richness of the approach is that the system can have better performances thorough some efforts in the external knowledge resources rather than through the internal changes. For example an error : "Fintecna Finanziaria per il Settore Industriale e dei Servizi S.p.A "→ "Servizi S.p.A "needs to be overcome only more accurate linguistic resources.

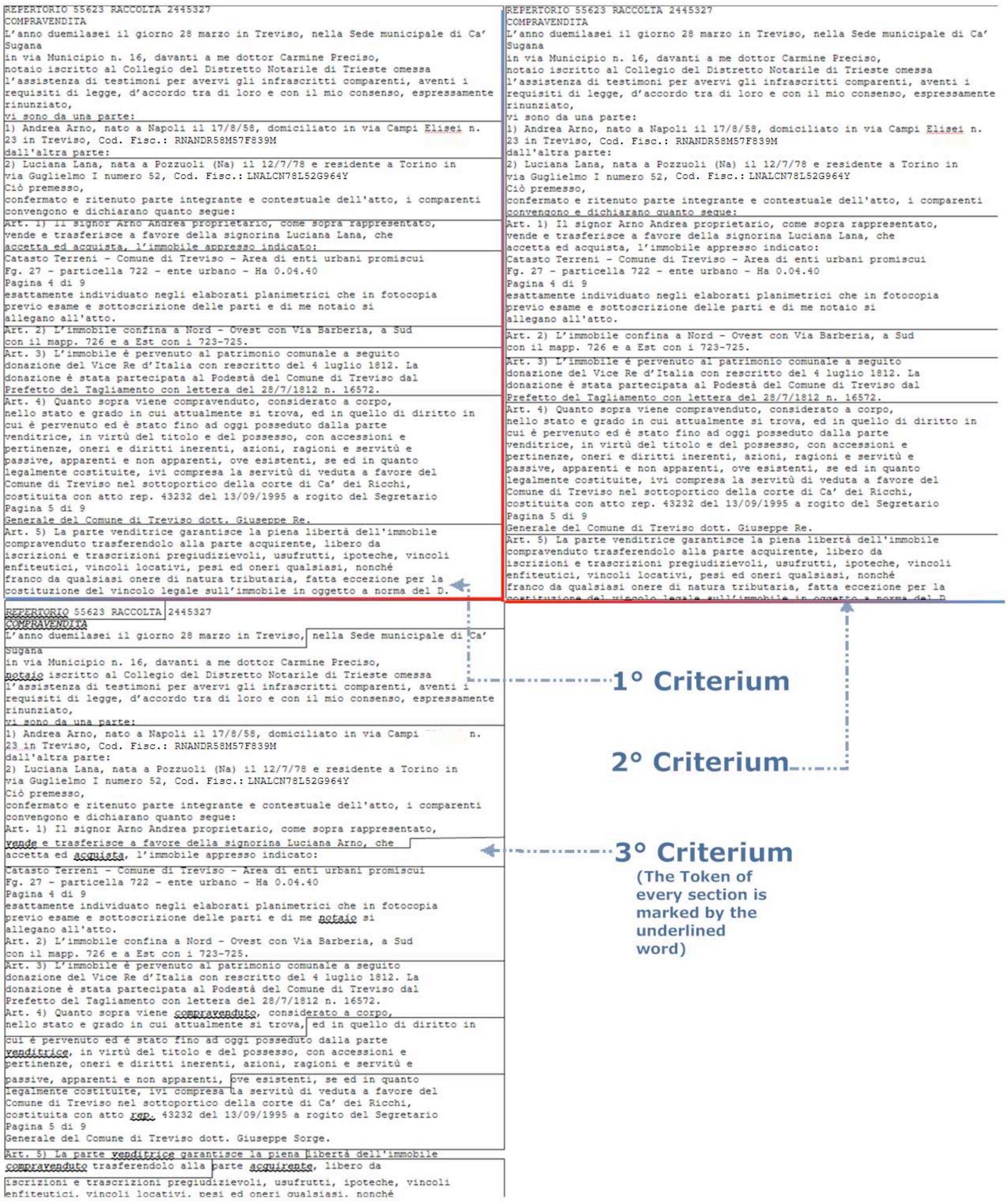


Figura 6.2: Application of Three Partition Criteria on the same Act fragment based on suitable key Selection

Algorithm 5: $Segm\text{-}Function$ algorithm

Input: $D, \mathcal{K}_{\mathcal{ST}}, \mathcal{K}_{\mathcal{DO}}, \mathcal{K}_{\mathcal{LO}}, N_C$

D is the document,

$\mathcal{K}_{\mathcal{ST}}$,$\mathcal{K}_{\mathcal{DO}}$,$\mathcal{K}_{\mathcal{LO}}$ is the range of $KnowledgeKey\text{-}Function$ for the structure Tbox and domain, lexical ontology respectively,

N_C is the enumeration of the partion criteria,

Output: $\mathcal{DS}$,

$\mathcal{DS}$ is the $Structured\text{-}Document$

begin

    $\mathcal{SD}^* = \{\emptyset\}$

    foreach $i \in N_C$ **do**

        $scoreVec[i] = 0$;

        $\mathcal{SD} = \{\emptyset\}$;

        $\mathcal{D}^B = getParagraphsSections(D, i)$;

        foreach $S_j^P \in \mathcal{D}^B$ **do**

            $\langle \mathcal{SD}, i \rangle$,$scoreVec$=

                $structuredFunction(S_j^P, \mathcal{K}_{\mathcal{ST}},$

                    $\mathcal{K}_{\mathcal{DO}}$,$\mathcal{K}_{\mathcal{LO}}$,$\mathcal{SD}$,$scoreVec$)

        end

        $\mathcal{SD}^* = \mathcal{SD}^* \cup \{\langle \mathcal{SD}, i \rangle\}$;

    end

    $\mathcal{SD} = getStructuredDocument(\mathcal{SD}^*, scoreVec)$;

end

Algorithm 6: $RDF\text{-}Extractor\ (RDFex)$ algorithm

Input: DS

DS is the $Structured\text{-}Document$.

Output: $\mathcal{KC}^D$,

$\mathcal{KC}^D$ is the $KnowledgeChunk\text{-}Document$

begin

$\mathcal{KC}^D = \{D\}$

foreach $\langle S_i^P, k_j \rangle \in \mathcal{SD}$ **do**

$\mathcal{SM} = \psi^{-1}(k_j)$,

$kc = InferenceProcedure(\mathcal{SM}, S_i^P)$

$\mathcal{KC}^D = \mathcal{KC}^D \cup kc$

end

end

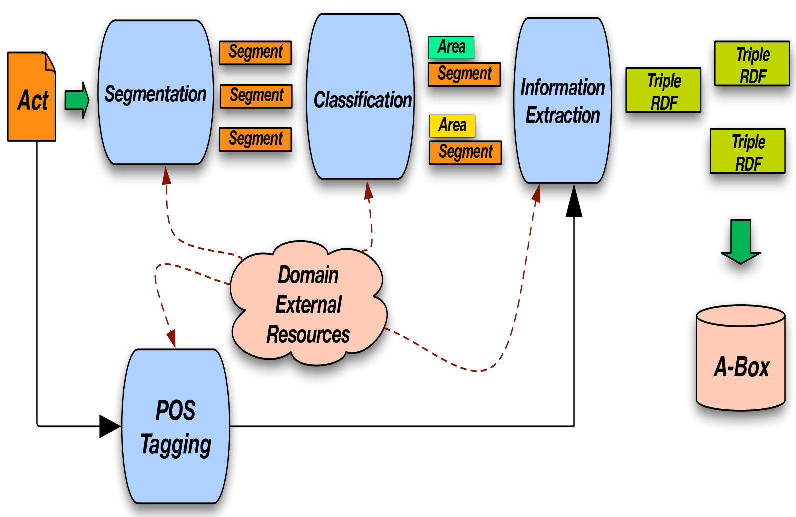


Figura 6.4: System Architecture

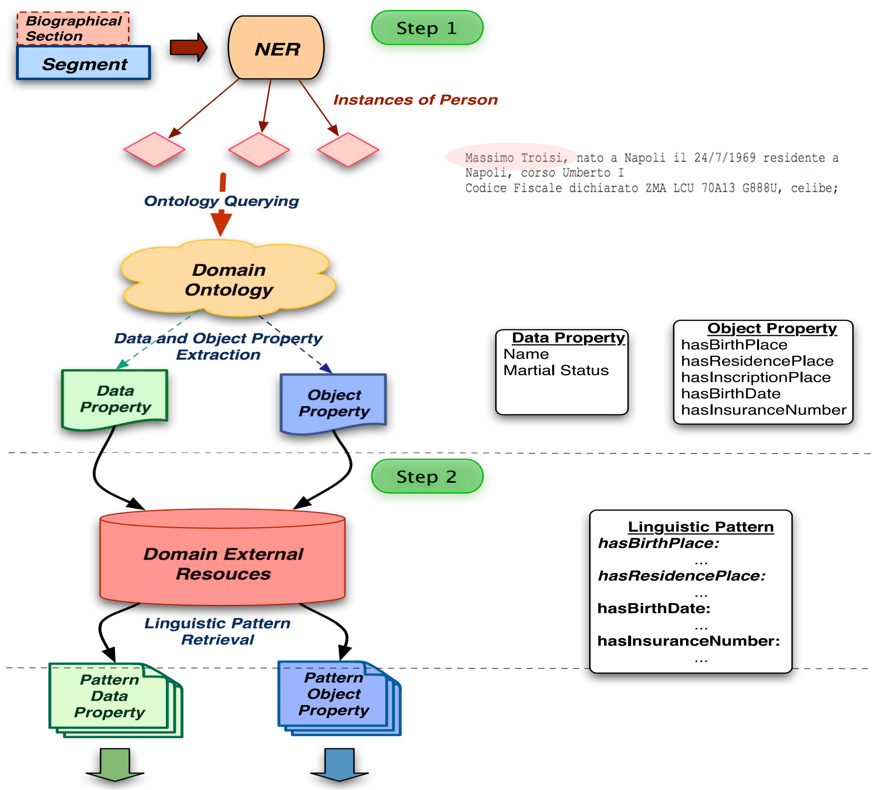


Figura 6.5: Process Flow 1

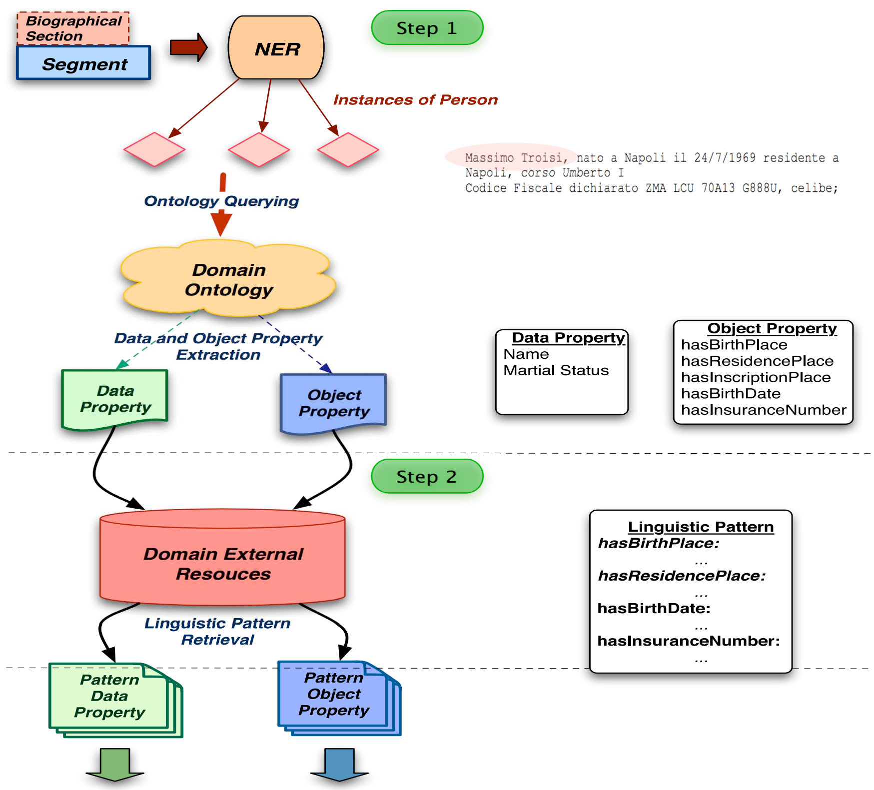


Figura 6.6: Process Flow 2

Capitolo 7

Conclusion and Future Works

Conclusion and Future Works

In this ph. d. dissertation I described a novel formal framework for multimedia ontologies, and in particular for an image and text data. I proposed a framework based on a constructivist vision of multimedia data processing and representation, thus providing a suitable knowledge base that can be used for storing and managing different levels of multimedia data in terms of rough data, intermediate and high level concepts, as well as some abstractions that can be observed over them.

In this way, I provided a comprehensive framework that can be used for a variety of purposes: e.g. information storing and management, information extraction, information retrieval and automatic annotations.

In order to make our theory understandable, I focused in particular on image data and text data. This kind of research is very important because the number of multimedia data is very huge and the only way to process such a large number of data is to use automatic tools that can extract information and represent them in a suitable way. In this framework, it is mandatory to provide a novel methodology for storing and accessing multimedia data, taking into consideration both the variety of data sources and the associated uncertainty of automatic analysis and recognition systems.

To the best of my knowledge, this is the first attempt in the scientific literature of the field the tries to furnish a high level conceptual framework for multimedia knowledge management and processing.

Some works could be done to improve the proposed methodology in some directions:

- To extend this approach to video and audio data.
- To build an integration system for all multimedia data based on the methodology describes in this dissertation .

- To consider different data set and To improve in number the experimental results in order to understand how scalable is the performance of those systems.
- To study in more depth the relationship among the expressivity of the language used and the complexity of the querying and reasoning algorithms.
- To generalize some proposed querying algorithms, thus removing some simplifying hypothesis.
- To use the Multimedia Web Sources as case study for a new search engine with the idea to collect special information around the wild web world.

Elenco delle figure

Elenco delle tabelle

Bibliografia

[1] http://www.acemedia.org/aceMedia.

[2] Flickr. http://www.flickr.com/.

[3] The friend of a friend (foaf) project. http://www.foaf-project.org.

[4] Google image search engine. http://images.google.it/.

[5] Image annotation on the semantic web. http://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/.

[6] Mpeg-7. multimedia content description interface. standard no. iso/iec n15938, 2001.

[7] Ontology web language. http://www.w3.org/2004/OWL/.

[8] Resource description framework (rdf). www.w3c.org/RDF/.

[9] Ryya. http://www.riya.com.

[10] Sesame query engine. http://www.openrdf.com/.

[11] Yahoo image search engine. http://images.search.yahoo.com/.

[12] *Querying the Semantic Web: A Formal Approach*, London, UK, 2002. Springer-Verlag.

[13] Roberto Abbruzzese. Un sistema di information extraction di documenti giuridici basato su ontologie. Master's thesis, Universita'ă Federico II di Napoli, Facolta'ă di Ingegneria , Corso di Laurea in Ingegneria Informatica, 2008.

[14] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[15] Alfred V. Aho and Jeffrey D. Ullman. Universality of data retrieval languages. pages 110–119, 1979.

[16] Wasfi Al-Khatib, Y. Francis Day, Arif Ghafoor, and P. Bruce Berra. Semantic modeling and knowledge representation in multimedia databases. *IEEE Trans. on Knowl. and Data Eng.*, 11(1):64–80, 1999.

[17] Richard Arndt, Raphaël Troncy, Steffen Staab, Lynda Hardman, and Miroslav Vacura. Comm: Designing a well-founded multimedia ontology for the web. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference,ISWC/ASWC*, volume 4825 of *Lecture Notes in Computer Science*, pages 30–43. Springer, 2007.

[18] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[19] Biagio Baiano. Automatic knowledge discovery in image data. Master's thesis, Universitaă Federico II di Napoli, Facolta'ă di Ingegneria , Corso di Laurea in Ingegneria Informatica, 2008.

[20] Tim Berners-lee. *Weaving the Web*. Harper Collins Publishers, 1999.

[21] Michael W. Berry. *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer, 2004.

[22] Irving Biederman. Human image understanding: recent research and a theory. In *Papers from the second workshop Vol. 13 on Human and Machine Vision II*, pages 13–57, San Diego, CA, USA, 1986. Academic Press Professional, Inc.

[23] Maarten; Blackburn, Patrick; de Rijke and Yde Venema. *Handbook: of Modal Logic*. North Holland, 2006.

[24] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

[25] G. Boccignone, A. Chianese, V. Moscato, and A. Picariello. Context-sensitive queries for image retrieval in digital libraries. *Journal of Intelligent Information Systems*, on-line first, 2007.

[26] Joost Breuker. A functional ontology of law. *Arti intelligence and law*, 7:341–361, 1994.

[27] Thomas Sikora B.S. Manjunath, Philippe Salembier. *Introduction to MPEG-7: Multimedia Content Description Interface*. 2002.

[28] A. Gangemi N. Guarino A. Oltramari C. Masolo, S. Borgo and L. Schneider. The wonderweb library of foundational ontologies (wfol). Technical report, WonderWeb Deliverable 17, 2002.

[29] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Dl-lite: Tractable description logics for ontologies. In *The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference AAAI*, pages 602–607, 2005.

[30] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[31] Yih-Chen Chang and Hsin-Hsi Chen. Approaches of using aword-image ontology and an annotated image corpus as intermedia for cross-language image retrieval. In *Proceedings of Cross-Language Evaluation Forum, C. Peters et al. (Eds.), Lecture Notes in Computer Science,*, 2006.

[32] Y. Chen, J. Li, and J. Z. Wang. *Machine Learning and Statistical Modeling Approaches to Image Retrieval*. Kluwer Academic Publishers, 2004.

[33] Angelo Chianese, Antonio Picariello, Lucio Sansone, and Maria Luisa Sapino. Managing uncertainties in image databases: A fuzzy approach. *Multimedia Tools Appl.*, 23:237–252, 2004.

[34] Andreas Schultz Chris Bizer. Berlin sparql benchmark (bsbm) specification - v2.0, 2008. `http://www4.wiwiss.fu-berlin.de/bizer/BerlinSPARQLBenchmark/spec/`.

[35] A. Church. A note on the entscheidungsproblem. *Journal of Symbolic Logic*, 1:40–41, 1936.

[36] Philipp Cimiano. *Ontology Learning and Population from Text*. Springer, 2006.

[37] Thomas Lukasiewicz Claudia d'Amato, Nicola Fanizzi. Tractable reasoning with baesian description logic. In Springer, editor, *Scalable Uncertainty Managment*, pages 146–159, 2008.

[38] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970.

[39] S. Dasiopoulou, C. Saathoff, Ph. Mylonas, Y. Avrithis, Y. Kompatsiaris, and S. Staab. *Semantic Multimedia and Ontologies: Theory and Applications*. Springer-Verlag, Heidelberg, March 2008, 2008.

[40] J. K. Debenham. *Knowledge Systems Design*. Prentice-Hall: Englewood Cliffs, NJ, 1988.

[41] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[42] A. Del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(2):121–132, 1997.

[43] F. M. Donini, M. Lenzerini, D. Nardi, W. Nutt, and A. Schaerf. An epistemic operator for description logics. *Artif. Intell.*, 100(1-2):225–274, 1998.

[44] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Andrea Schaerf, and Werner Nutt. Adding epistemic operators to concept languages. pages 342–353, 1992.

[45] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[46] David W. Embley, Douglas M. Campbell, Stephen W. Liddle, and Randy D. Smith. Ontology-based extraction and structuring of information from data-rich unstructured documents. In *In Proceedings of the Conference on Information and Knowledge Management (CIKM'98), Washington D.C*, pages 52–59. ACM Press, 1998.

[47] M. Ensing, R. Paton, P.H.W.M. Speel, and R. Rada. An object-oriented approach to knowledge representation in a biomedical domain. *Artificial intelligence in medicine*, 6(6):459–482, 1994.

[48] Andy Seaborne Eric Prud'hommeaux. Sparql query language for rdf, 15 January 2008. `http://www.w3.org/TR/rdf-sparql-query/`.

[49] S. Manjunathan et al. Cortina: Searching a 10 milion + images database. In *Technicak Report, VRL, ECE, University of California, Santa Barbara*, 2007.

[50] Ronald Fagin. Fuzzy queries in multimedia database systems. In *PODS*, pages 1–10. ACM Press, 1998.

[51] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc.of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05)*, volume 2, pages 524–531, San Diego, USA, 2005.

[52] Ronen Feldman and James Sanger. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, December 2006.

[53] Dieter Fensel, Ian Horrocks, Frank van Harmelen, Deborah L. McGuinness, and Peter F. Patel-Schneider. Oil: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–44, 2001.

[54] Stamou G, Van Ossenbruggen J, Pan J Z, Schreiber G, and Smith J R. Multimedia annotations on the semantic web. *Multimedia, IEEE*, 13:86 – 90, 2006.

[55] Luigi Giussani. *Religious Sense*. McGill-Queen's University Press, 1997.

[56] Thomas R. Gruber. A translation approach to portable ontology specifications. volume 5, pages 199–220. Academic Press Ltd., 1993.

[57] Tom Gruber. Collective knowledge systems: Where the social web meets the semantic web. *Web Semant.*, 6(1):4–13, 2008.

[58] Tom Gruber. Collective knowledge systems: Where the social web meets the semantic web. *Web Semant.*, 6(1):4–13, 2008.

[59] Tom Gruber. Ontology. *Encyclopedia of Database Systems*, 2008.

[60] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. *Int. J. Hum.-Comput. Stud.*, 43(5-6):625–640, 1995.

[61] Nicola Guarino and Roberto Poli. Editorial: The role of formal ontology in the information technology. *Int. J. Hum.-Comput. Stud.*, 43(5-6):623–624, 1995.

[62] Volker Haarslev and Ralf Moller. Description of the racer system and its applications. In *Proceedings of the International Workshop on Description Logics*, pages 132–141, 2001.

[63] J. S. Hare and P. H. Lewis. On image retrieval using salient regions with vector-spaces and latent semantics. In *Proc. of Image and Video Retrieval (CIVR 2005)*. Springer, 2005.

[64] P. Hayes. The logic of frames. *Readings in Artificial intelligence, B.L. Webber and N.j. Nilson, eds*, pages 451–458, 1981.

[65] Hermann. *Knowledge Representation and the Semantics of Natural Language*. Springer Berlin Heidelberg, 2006.

[66] J. R. Hobbs, D. Appelt, M. Tyson, J. Bear, and D. Israel. Sri international: Description of the fastus system used for muc-4. *Fourth Message Understanding Conference, Morgan Kaufmann*, pages 143–147, 1992.

[67] Thomas Hofmann. Probabilistic latent semantic analysis. In *In Proc. of Uncertainty in Artificial Intelligence, UAIâĂŹ99*, pages 289–296, 1999.

[68] HP-Labs. Sparql query engine. `http://jena.sourceforge.net/ARQ/`.

[69] Song Liu Huan Wang and Liang-Tien Chia. Image retrieval with a multi-modality ontology. In Springer, editor, *Multimedia System*, volume 13. Springer Berlin / Heidelberg, 2007.

[70] Chang-Shing Lee; Zhi-Wei Jian; Lin-Kai Huang;. A fuzzy ontology and its application to news summarization. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 35:859 – 880, 2005.

[71] Jane Hunter. Adding multimedia to the semantic web - building an mpeg-7 ontology. In *In International Semantic Web Working Symposium (SWWS*, pages 261–281, 2001.

[72] P.S. Jacobs and L.F. Rau. Scisor: Extracting information from on-line news. *Comm ACM*, 33(11):88–97, 1990.

[73] Steve R. Kao, Anne; Poteet. *Natural Language Processing and Text Mining*. Springer London, 2007.

[74] L. Kennedy, M. Naaman, S. Ahern, R Nair, and T Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. In *ACM Multimedia*, pages 631–640, 2007.

[75] Michael Kifer, Georg Lausen, and James Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42:741–843, 1995.

[76] Domenico Lembo. *Dealing with Inconsistency and Incompleteness in Data Integration*. PhD thesis, Dipartimento di Informatica e Sistemistica, Universita' di Roma La Sapienza, 2004.

[77] Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[78] Maurizio Lenzerini. Data integration needs reasoning. In *Logic Programming and Non-monotonic Reasoning, 6th International Conference, LPNMR*, volume 2173 of *Lecture Notes in Computer Science*, pages 54–61. Springer, 2001.

[79] Maurizio Lenzerini. Data integration: A theoretical perspective. pages 233–246, 2002.

[80] K. Lerman and L. Jones. Social browsing on flickr. *CoRR*, abs/cs/0612047, 2006.

[81] Hector J. Levesque and Gerhard Lakemeyer. *The Logic of Knowledge Base*. The MIT Press, 2001.

[82] G. E. R. Lloyd. Aristotle: The growth and structure of his thought. *Cambridge: Cambridge Univ. Pr.,*, pages 162–164, 1968.

[83] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[84] Thomas Lukasiewicz. Expressive probabilistic description logics. *Artif. Intell.*, 172(6-7):852–883, 2008.

[85] A. M. Rinaldi M. Albanese, A. Picariello. A semantic search engine for web information retrieval: an approach based on dynamic semantic networks. In ACM Press, editor, *ACM SIGIR Semantic Web and Information Retrieval Workshop.*, 2004.

[86] Chris Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. 2000.

[87] Prof Colombetti Marco. *Lectures Notes - Knowledge Engineering Course*. Politecnico di Milano, 206.

[88] Davis Martin. *The Universal Computer: The Road from Leibniz to Turing*. W.W. Norton., 2000.

[89] L. T. McCarty. A language for legal discourse i. basic features. In *ICAIL '89: Proceedings of the 2nd international conference on Artificial intelligence and law*, pages 180–189, New York, NY, USA, 1989. ACM.

[90] S. Miike, E. Itoh, K. Ono, and K. Sumita. A full-text retrieval system with a dynamic abstract generation function. *in Proc SIGIR 94*, pages 152–161, 1994.

[91] Marvin Minsky. *A Framework for Representing Knowledg*. Patrick Henry Winston (ed.), The Psychology of Computer Vision, 1975.

[92] M.R. Naphade and T.S. Huang. Extracting semantics from audio-visual content: the final frontier in multimedia retrieval. *Neural Networks, IEEE Transactions on*, 13(4):793–810, Jul 2002.

[93] Wendy Hall Nigel Shadbolt and Tim Berners-Lee. The semantic web revisited. *IEEE Intelligent Systems*, 21-issue 3:96–101, 2006.

[94] Ulf Nilsson and Jan Maluszynski. *Logic, Programming and Prolog (2ed)*. John Wiley & Sons Ltd, 1995.

[95] Natasha Noy and Alan Rector. Defining n-ary relation on the semantic web. Technical report, W3C, 2006.

[96] Jacco Van Ossenbruggen, Frank Nack, and Lynda Hardman. That obscure object of desire: Multimedia metadata on the web, part 2. *IEEE Multimedia*, 12:54–63, 2005.

[97] S. Espinosa Perald, A. Kaya, S. Melzer, R. Moller, and M. Wessel. Towards a media interpretation framework for the semantic web. In *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 374–380, Washington, DC, USA, 2007. IEEE Computer Society.

[98] Antonio Picariello and Maria Luisa Sapino. A fuzzy algebra for image data bases. In *Multimedia Information Systems*, 2002.

[99] H. Shrobe R. Davis and P. Szolovits. What is a knowledge representation? *AI Magazine*, 14:17–33, 1993.

[100] Bernd Neumann Ralf Maller. Ontology-based reasoning techniques for multimedia interpretation and retrieval. In Springer, editor, *Semantic Multimedia and Ontologies*, pages 55–98. Springer London, 2008.

[101] Steffen Staab Richard Arndt, Raphael Troncy and Lynda Hardman. Adding formal semantics to mpeg-7: Designing a well-founded multimedia ontology for the web. Technical report, Department of Computer Science. Technical Report of University of Koblenz., 2007.

[102] Ashley K D Rissland E L. A case-based system for trade secrets law. In *Proc. 1st Intl. Conf. on Artificial Intelligence and Law*, pages 61–67, 1987.

[103] A. Roventini. Italwordnet: Building a large semantic database for the automatic treatment of the italian language. *In Zampolli, A., Calzolari, N., Cignoni, L. (eds.), Computational Linguistics in Pisa, Special Issue of Linguistica Computazionale*, Vol. XVIII-XIX, 2003.

[104] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence, 2003.

[105] Petridis K. Bloehdorn S. Saathoff C. Simou N. Dasiopoulou S. Tzouvaras V. Handschuh S. Avrithis Y. Kompatsiaris Y. Staab S. Knowledge representation and semantic annotation of multimedia contentvision. In *Image and Signal Processing, IEEE Proceedings*, volume 153, pages 255–262, 2006.

[106] S. Melzer R. Maller T.Nath M. Wessel S. Espinosa, A. Kaya. Multimedia interpretation as abduction. In *International Workshop on Description Logics*, 2007.

[107] Simone Santini. Image semantic without annotatiions. *Annotation for the Seamantic Web*, 96:156–180, 2003.

[108] Simone Santini and Ramesh Jain. Do images mean anything? In *ICIP (1)*, pages 564–567, 1997.

[109] James G. Schmolze, Bolt Beranek, and Newman Inc. An overview of the kl-one knowledge representation system. *Cognitive Science*, 9:171–216, 1985.

[110] Ralf Mller Sean Bechhofer and Peter Crowther. The dig description logic interface. In *International Workshop on Description Logics*, 2003.

[111] N. Simou, Th. Athanasiadis, V. Tzouvaras, and S. Kollias. Multimedia reasoning with f-shin. 2nd International Workshop on Semantic Media Adaptation and Personalization, London, December 17-18, 2007, London, United Kingdom (SMAP 2007), 2007.

[112] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

[113] A.W.M. Smeulders, M. Worring, S. Santini, A Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:1349–1379, 2000.

[114] John F. Sowa. Knowledge representation: Logical, philosophical, and computational foundations. In *Brooks Cole Publishing Co., Pacific Grove, CA*, 2000.

[115] John. F. Sowa. Semantic networks (extended version of an article written for the Encyclopedia of Artificial Intelligence, edited by Stuart C. Shapiro, Wiley in 1987, second edition 1992), 2002. `http://jfsowa.com/pubs/semnet.htm`.

[116] R.K Stamper. The role of semantics in legal expert systems and legal reasoning. *Ratio Juris*, 4(2):219–244, 1991.

[117] Luc Steels. Frame-based knoledege representation. *AI working Papers-MIT Artificial Intelligence Laboratory*, 1071.

[118] Umberto Straccia. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research-JAIR*, 14:137–166, 2001.

[119] Umberto Straccia. A fuzzy description logic for the semantic web. In Elie Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence, chapter 4, pages 73–90. Elsevier, 2006.

[120] Umberto Straccia. Managing uncertainty and vagueness in description logics, logic programs and description logic programs. In *Reasoning Web, 4th International Summer School, Tutorial Lectures*, number 5224 in Lecture Notes in Computer Science, pages 54–103. Springer Verlag, 2008.

[121] Sergio Tessaris. *Questions and answers: reasoning and querying in Description Logic*. PhD thesis, University of Manchester, 2001.

[122] D. Tiscornia. Some ontological tools to support legal regulatory compliance, with a case study. *Workshop on Regulatory Ontologies and the Modeling of Complaint Regulations (WORM CoRe 2003) Springer LNCS*, November 2003.

[123] Alan Turing. Computing machinery and intelligence. *A Quarterly Review of Psychology and Philosophy*, 236:433, 1950.

[124] A.M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, Series 2,42:230–265, 1936.

[125] Jacco van Ossenbruggen, Frank Nack, and Lynda Hardman. That obscure object of desire: Multimedia metadata on the web, part 1. *IEEE MultiMedia*, 11(4):38–48, 2004.

[126] P.R.S. Visser and T.J.M. Bench-Capon. The formal specification of a legal ontology. In *Proceedings of JURIX-96*, pages 15–24. Tilburg University Press, 1996.

[127] J. Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for pictures libraries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23:1–16, 2001.

[128] James Z. Wang and Jia Li. Real-time computerized annotation of pictures. In *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, volume 30, pages 985–1002, 2008.

[129] Peter Wegner. Dimensions of object-based language design. *SIGPLAN Not.*, 22(12):168–182, 1987.

[130] Barry Smith Werner Ceusters and Louis Goldberg. A terminological and ontological analysis of the nci thesaurus. *Methods of Information in Medicine*, 44:498âĂŞ507, 2005.

[131] Rajeev Agrawal William I. Grosky and Farshad Fotouchi. Mind the gaps-finding the appropriate dimensional representation for semantic retrieval of multimedia assets. *Semantic Multimedia and Ontologies*, pages 229–252, 2008.