



Università degli Studi di Salerno

Tesi di Dottorato in Scienze della Comunicazione

NON CLASSICAL CONCEPT REPRESENTATION AND REASONING IN FORMAL ONTOLOGIES

Antonio Lieto

Ph.D. Supervisor:
Prof. Marcello Frixione

Ph.D. Coordinator:
Prof. Alessandro Laudanna

X Ciclo - Nuova Serie 2008-2011

Acknowledgements

I would like to express my appreciation to a number of persons who have provided valuable assistance to me during these years in which I have been enrolled within the doctoral program at the University of Salerno. First of all, I would like to thank my Ph.D. supervisor Marcello Frixione. He has been for me a real guide and I will take his lessons, suggestions and advices with me for the rest of my life.

I would like to thank also Annibale Elia and Alessandro Laudanna for their assistance and support during these years. A special thank goes to Roberto Cordeschi who suggested me to follow interesting courses and that, in each situation in which I had the occasion to meet him, always showed interest for my research asking about the state of advancement of my work.

Then I would like to thank all the persons encountered in international and national meetings and conferences from which I received precious feedback for improving the research I carried out with my thesis.

Pursuing a Ph.D. requires not only theoretical and technical skill but also a big amount of efforts. I would like to thank my parents - Andrea and Anna - and my brothers (Teresa and Nicola) for sharing their unconditional love with me. Last but not least, I would like to thank my girlfriend Paola. Anything I could say would not be enough to acknowledge her.

Contents

Chapter 1. Ontological Languages and Description Logics

- 1.1. Semantic Web Languages pp. 1
- 1.2. Description Logics pp. 4
- 1.3. Description Logics for Semantic Web Languages pp. 8
- 1.4. From Description Logics to Semantic Web languages pp. 9
 - 1.4.1. OWL Lite pp. 10
 - 1.4.2. OWL DL pp. 11
 - 1.4.3. OWL Full pp. 11
- Appendix A and B pp. 12

Chapter 2. Representing Non Classical Conceptual Information

- 2.1. Concepts in Philosophy and Psychology pp. 15
- 2.2. Compositionality pp. 17
- 2.3. Against "Classical" Concepts pp. 18
- 2.4. Concept Representation in Artificial Intelligence pp. 20
- 2.5. Artificial Systems: Why Prototypical Effects are Needed pp. 22
- 2.6. Non-classical Concepts in Computational Ontologies pp. 23
- 2.7. Some Suggestions from Cognitive Science pp. 26

Chapter 3. Models of Cognition: Prototypes and Exemplars to Explain the Typicality

- 3.1. Theory of Concepts – an overview pp. 29
- 3.2. Prototype and Exemplar Theories pp. 30
 - 3.2.1. Prototype Theory pp. 30
 - 3.2.2. Criticisms pp. 33
 - 3.2.3. Exemplar theory pp. 34
- 3.3. Prototype based model of categorization pp. 35
 - 3.3.1. An example of Prototype-based categorization: The Hampton's model pp. 37

- 3.3.2. Categorization with exemplars pp. 39
 - 3.3.2.1 Exemplar categorization: Nosofsky's Model pp. 40
- 3.4. Prototypes and Exemplars pp. 42
 - 3.4.1. In favor of Prototypes: the Random Distortion Pattern Evidence pp. 43
 - 3.4.2. Evidence Against Prototypes pp. 44
 - 3.4.3. Prototypes vs Exemplars in short pp. 46
 - 3.4.4. Criticisms to the Exemplar Paradigm and Hybrid Approaches pp. 48
- 3.5. Towards Hybrid Approaches to Concepts Representation and Reasoning pp. 51
- 3.6. Prototypes and Exemplar Theories in Machine Learning: a brief overview pp. 54
 - 3.6.1. Instance-based Classifier Systems pp. 56

Chapter 4. A Hybrid Approach to Concept Representation and Reasoning

- 4.1 General Description pp. 59
 - 4.1.1 Compositional Module pp. 61
 - 4.1.2 Typical Knowledge Module – Prototypes and Exemplars pp. 62
- 4.2 Cognitive Background pp. 63
 - 4.2.1 Dual Process Approach pp. 63
 - 4.2.2 A “Pseudo-Fodorian” Proposal pp. 65
 - 4.2.3 Prototypes and Exemplars pp. 66
- 4.3 Adaptation of the proposed cascade procedure pp. 67
- 4.4. Implementation pp. 70
- 4.5. Performing Heuristic Categorization pp. 77
- 4.6. Concepts Similarities in Ontology KB pp. 80
- 4.7. Proposed adaptation of the PEL-C algorithm pp. 88
- 4.8. Expected Results pp. 89

4.9. Prototype and Exemplars Representation pp. 91

Chapter 5. Evaluation and Discussions

5.1. Evaluation pp. 95

5.2. Evaluation Results in a nutshell pp. 106

5.3. Discussion and Analysis pp. 108

Chapter 6. Conclusions

Conclusions pp. 111

List of Figures

Figure 1.1. XML representation pp. 2

Figure 1.2. RDF graph based on URIs pp. 3

Figure 1.3. DL Knowledge Base architecture (from Baader and Nutt, 2002) pp. 6

Figure 1.4. Syntax and Semantics of the S family of Description Logics (from Horrocks at al. 2003) pp.8

Figure 3.1. Example of a frame based representation pp. 32

Figure 3.2. Exemplars in the Context Model, adapted from Machery pp. 35

Figure 3.3. A Sequential Schema of Psychological Categorization in Prototype Theories pp. 36

Figure 3.4. A Sequential Schema of Psychological Categorization in Exemplars Theories pp. 39

Figure 3.5. Test and Training Patterns of points (from Smith & Minda 2002) pp. 44

Figure 3.6. Linear Separability pp. 46

Figure 3.7. Categorization Procedure of RULEX (from Machery, 2009) pp. 52

Figure 4.1. General canonical model of the architecture pp. 61

Figure 4.2. Relations between published Linked Open Data (from Bizer et al. 2009) pp. 72

Figure 4.3. Answer space extension through the interconnection of different KB pp. 73

- Figure 4.4. Concept representation in the compositional module pp. 74
- Figure 4.5. Example of a concept description in the compositional module pp. 75
- Figure 4.6. Ex. of connection between and exemplar and the relative concept in DL pp. 77
- Figure 4.7. Representation of colours in terms of Conceptual Spaces pp. 87
- Figure 4.8. Exemplars and Prototypes for the concept BIRD pp. 93
- Figure 5.1. Taxonomy of a toy DL knowledge-base pp. 96
- Figure 5.2. Typology of query performed in the experiment pp. 98
- Figure 5.3. Experimental set-up situation for the evaluation task pp. 99
- Figure 5.4. E1 experimental situation pp. 100
- Figure 5.5. First control situation E2.1.1 pp. 102
- Figure 5.6. Second control situation E2.1a pp. 102
- Figure 5.7. E2.2.1 First control situation pp. 103
- Figure 5.8. Second control situation E2.2a pp. 104
- Figure 5.9. E3. Experimental situation pp. 105

List of Tables

Table 3.1 Prototype models vs Exemplar Models pp 48

Table 3.2: The 5-4 Category Structure (Adapted from Smith and Minda 2000) pp. 49

Table 4.1. The learning algorithm proposed by Gagliardi pp. 79

Table 4.2. Example of the extension of a definition in the Tversky's algorithm pp. 86

Table 4.3. Proposed adaptation of the learning algorithm pp. 88

Table 5.1. Precision and recall pp. 98

Table 5.2. Concept Retrieval results pp. 107

Table 5.3. Precision and Recall results. Synthetic table pp. 107

Abstract

Formal ontologies are nowadays widely considered a standard tool for knowledge representation and reasoning in the Semantic Web. In this context, they are expected to play an important role in helping automated processes to access information. Namely: they are expected to provide a formal structure able to explicate the relationships between different concepts/terms, thus allowing intelligent agents to interpret, correctly, the semantics of the web resources improving the performances of the search technologies.

Here we take into account a problem regarding Knowledge Representation in general, and ontology based representations in particular; namely: the fact that knowledge modeling seems to be constrained between conflicting requirements, such as compositionality, on the one hand and the need to represent prototypical information on the other. In particular, most common sense concepts seem not to be captured by the stringent semantics expressed by such formalisms as, for example, *Description Logics* (which are the formalisms on which the ontology languages have been built). The aim of this work is to analyse this problem, suggesting a possible solution suitable for formal ontologies and semantic web representations.

The questions guiding this research, in fact, have been: is it possible to provide a formal representational framework which, for the same concept, combines both the classical modelling view (accounting for compositional information) and defeasible, prototypical knowledge ? Is it possible to propose a modelling architecture able to provide different type of reasoning (e.g. classical deductive reasoning for the compositional component and a non monotonic reasoning for the prototypical one)?

We suggest a possible answer to these questions proposing a modelling framework able to represent, within the semantic web languages, a multilevel representation of conceptual information, integrating both classical and non classical (typicality based) information. Within this framework we hypothesise, at least in principle, the co-existence of multiple reasoning processes involving the different levels of representation.

This work is organized as follows: in chapter 1 the semantic web languages and the description logics formalisms on which they are based are briefly presented. Then, in

chapter 2, the problem on which this work is focused (e.g. conceptual representation) is illustrated and the general idea of the proposed multi-layer framework is sketched. In chapter 3 the psychological theories about concepts based on prototypes and exemplars are surveyed. In this chapter we argue that such distinction can be useful in our approach because it allows (i) to have a more complete representation of the concepts and (ii) to hypothesise different types of non monotonic reasoning processes (e.g. non monotonic categorization). In chapter 4 the proposed modeling architecture is presented and, in chapter 5, it is evaluated on particular information retrieval tasks. The chapter 6 is dedicated to the conclusions.

Chapter 1.

Ontological Languages and Description Logics

The Semantic Web was originally proposed as an extension to the current Web, as the way to solve the problem of semantic heterogeneity (T.B. Lee 2001). In this view, the proposed solution has been that one of adding, a so called *semantic layer* as an extra layer built on top of the Web, which makes data not only human processable but also machine processable thanks to an enriched semantics. The word “semantics”, in the this research area, assumes a precise connotation: the meaning of the data and documents is assumed to be codified as *metadata*, i.e. data about data (Giunchiglia et al 2010). In this view, data are organized in different levels of increasing expressiveness, each corresponding to a specific representation need. Such levels correspond to different representation languages: XML, XML Schema, RDF and RDF Schema (RDFS) and OWL. In the section 1.1, we briefly summarize the main distinctive elements of the first four mentioned languages. Then, in 1.2, we introduce the basic elements characterizing the standard Description Logics (DLs) and, in 1.3, individuate the connections between DLs and the semantic web languages. Finally, the description of OWL (Ontology Web Language) and of its sub languages is demanded to the paragraph 1.4.

1.2. Semantic Web Languages

XML is designed to represent information using customized tags. Due to this feature, this language is widely used for information exchange on the Web and elsewhere. Strictly speaking XML is not a semantic web language as it codifies no semantics. However, it is important because all the semantic web languages are extensions of XML. Furthermore, it has also a historical importance because, if compared to HTML, it represents a first step towards the semantic web languages.

An example of a XML based representation is the following: let suppose that we have to represent a statement like “DBpedia was last modified on 28 January 2012”. It can be represented in XML using, for example, the tags “DBpedia”, and “modified”, along with a statement indicating the specific XML version of the representation, as shown in fig. 1.1

```
<? Xml version="1.0" ?>
<DBpedia>
<modified>28 January 2012</>
</DBpedia>
```

Fig. 1.1 a XML representation

XML Schema is a XML based format defining the rules that an XML document must respect. From an object oriented programming point of view, XML Schema can be assimilated to a class, while XML documents correspond to instances of that class. It is used for exchanging information between parties that agree on a particular set of rules. However, the terms used in XML Schema have, again, no semantics. Therefore, it results to be difficult for machine to accomplish communication between them when new XML vocabulary terms are introduced. Because of the lack of semantics, XML Schema does not allow to differentiate between polysemous concepts/terms and does not allow to combine the synonymous terms.

RDF language was developed in order to overcome these limits. RDF is used to describe information about web resources. This metadata based description allows to make information machine processable and “understandable”. It is designed to provide flexibility in representing information. RDF is based on a simple data model that allows to make statements about web resources, and provides the capability to perform inferences on the represented statements. The data model of RDF is a directed graph consisting of nodes and edges. Statements about resources can be represented by using this graph. The example in figure 1.2 represents the assertion “Geonames has coverage of all countries” (from De Virgilio et al 2010, p. 30). Edges in RDF graph are labeled. When they connect two nodes, they form a triple. The triples based semantics is one of the main features of RDF and RDF based languages. One of the two nodes represents the subject, the other the object and the edge represents the predicate of the statement.

The direction of the edge is from the subject to the object. RDF usually uses URIs references to identify subjects, objects and predicates (as in fig. 1.2)¹.

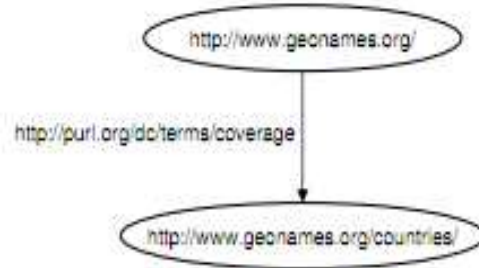


Figure 1.2. RDF graph based on URIs

A statements such as that of fig. 1.2 can be described in RDF as shown below:

```
<? Xml version="1.0" ?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdfs="http://www.purl.org/dc/terms#" >
  <rdf:Description rdf:about="http://www.geonames.org" >
    <dc:coverage rdf:resource="http://www.geonames.org/countries" >
  </rdf:Description>
</rdf:RDF>
```

A limit of RDF is represented by the fact that it does not allow to define a hierarchy between the represented resources. For this reason, RDF Schema (RDFS) was created. RDFS is an extension of RDF able to provide a vocabulary to represent classes, subclasses, properties and relations between properties. The capability of representing classes and subclasses allows users to publish ontologies on the web, but these ontologies are limited, as RDFS cannot represent disjunction or specific cardinality values. Furthermore, RDFS presents also other limits such as, for example: (i) it is not possible to localize the range and domain constraints of a specific property (e.g. it is not possible to express that the range of hasChild is person when applied to persons and

¹ However, in such cases some elements within the triple can be represented without URI. For example: if the statement to be represented in RDF is “DBpedia was modified on 25 January 2012” we have the object of the triple (25 January 2012) which is a literal. Thus it can be represented without URI.

elephant when applied to elephants), (ii) it is not possible to insert inverse or symmetrical properties (e.g. it is not possible to say that `hasPart` is the inverse of `isPartOf` and that the property “touches” is symmetrical).

For these reasons, other languages were developed (see Baader et al 2002, Horrocks et al 2011). One of these is the OWL language. It is based on Description Logics and makes it possible to describe concepts and relations between concepts via logical axioms. OWL is the result of the integration between the OIL and DAML languages. Similarly to RDF, on which it is based, in OWL data are represented as triples: subject, object and predicate. In particular, it is possible to distinguish between three basic OWL languages: OWL Lite, OWL DL and OWL Full. Each of them is characterized by different expressivity and computational complexity. Before describing in major detail these three languages, we will introduce, in the next paragraphs, the main features of the Description Logics and will try to evidence the connection between description logics and the Ontology Web Language.

1.2 Description Logics

Description Logics (from now on DLs) are a family of class-based (concept-based) knowledge representation formalisms (Baader et al 2002). They are characterised by the use of various constructors that allow to build complex classes from simpler ones and by an emphasis on the decidability and computational complexity of some key reasoning tasks. Description Logics had a strong influence on the design of OWL, particularly on the formalisation of semantics and the choice of language constructs. A key feature of Description Logics is that they are logics, i.e., formal languages with well defined semantics. The standard technique for specifying the meaning of a Description Logic is via a model theoretic approach, whose purpose is to explicate the relationship between the language syntax and the models of the language. As reported in Horrocks et al. (2011): a model consists of a domain (which is usually written Δ^I) and an interpretation function (often written \cdot^I), where the domain is a set of objects and the interpretation function is a mapping from individual, class and property names to elements of the domain, subsets of the domain and binary relations defined on the domain, respectively.

So, for an individual Nicola, $\text{Nicola}^I \in \Delta^I$, for a class Person, $\text{Person}^I \subseteq \Delta^I$, and for a property² friend, $\text{friend}^I \subseteq \Delta^I \times \Delta^I$. The interpretation function can be extended from class names to complex class descriptions in the obvious way. For example, given two classes Male and Person interpreted as the sets $\text{Male}^I = \{a, b, c\}$ and $\text{Person}^I = \{b, c, e\}$, then the intersection of Male and Person (i.e., male persons) is interpreted as the intersection of $\{a, b, c\}$ and $\{b, c, e\}$, i.e., $(\text{Male and Person})^I = \{b, c\}$. A domain can be potentially represented by any set of objects. What is important, in fact, is the relationship between objects and sets of objects. In a given model, for example, an individual i is an instance of a class C just in case i is interpreted as an element of the interpretation of C (i.e., $i^I \in C^I$), and a class C is a subclass of a class D just in case the interpretation of C is a subset of the interpretation of D (i.e., $C^I \subseteq D^I$).

The main building blocks of DL knowledge bases are concepts (or classes), roles (or properties), and individuals. Certain concepts (e.g., say, Person) are atomic. Then, using a rich set of concept constructors, it is possible to create complex concepts, by describing the conditions on concept membership. For example, the concept $\exists \text{hasFather} . \text{Person}$ describes those objects that are related through the hasFather role with an object from the concept Person.

The general architecture of a DL System is represented in the figure 1.3 below. Namely: a DL knowledge base typically consists of a TBox T and an ABox A . A TBox (Terminological Box) introduces the terminology, i.e., the vocabulary of an application domain, and contains axioms about the general structure of all allowed worlds, and is therefore akin to a database schema. An ABox (Assertional Box) contains assertions about specific individuals in the terms of the TBox vocabulary, and contains axioms that describe the structure of a particular world. For example, the TBox axiom (1) states that each instance of the concept Person must be related by the role hasFather with another instance of the concept Person. While (2) and (3) represent the assertional knowledge within the ABox and state that Nicola is a Person (2), and that Teresa and Nicola are brothers (3).

² In this case with the term “property” we indicate a two argument relation (Rab). However, usually, the term property is used in order to indicate predicates with one argument (Pa). This distinction, well known in classical logics, is generally not considered within this research area, thus all the types of predicates are named “properties”.

- (1) $\text{Person} \sqsubseteq \exists \text{hasFather} . \text{Person}$
- (2) $\text{Person}(\text{Nicola})$
- (3) $\text{hasBrother}(\text{Nicola}, \text{Teresa})$

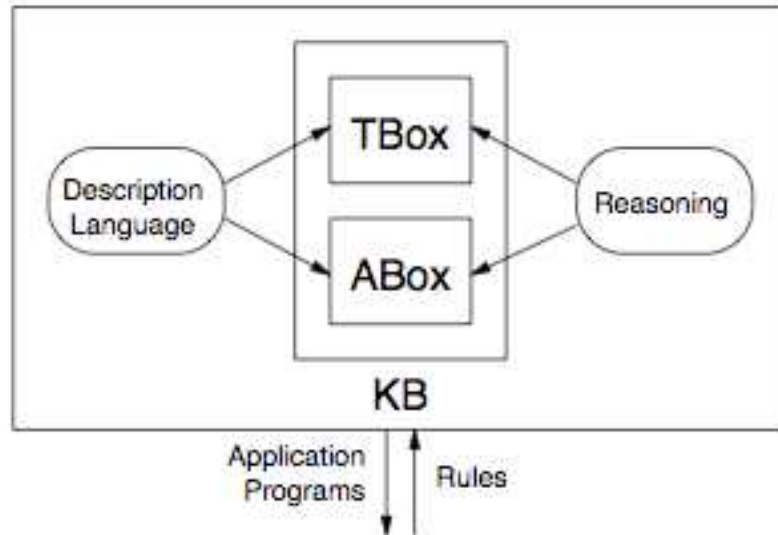


Figure 1.3. DL Knowledge Base architecture (from Baader and Nutt, 2002).

As explained before, the meaning of the expressed axioms is given by corresponding constraints on models. For example: if the knowledge base contains an axiom stating that *Person* is a subclass of *Animal* (written $\text{Person} \sqsubseteq \text{Animal}$), then in every model of the knowledge base the interpretation of *Person* must always be a subset of the interpretation of *Animal*. The meaning of a knowledge base derives from features and relationships that are common to all possible models. If, for example, the interpretation of a class must always be the empty set, then that class is said to be inconsistent, while if there are no possible interpretations, the knowledge base itself is said to be inconsistent. If the relationship specified by a given axiom must hold in all interpretations of a knowledge base, then that axiom is said to be entailed by the knowledge base, and if one knowledge base entails every axiom in another knowledge base, then the first knowledge base is said to entail the second knowledge base. A knowledge base containing the axiom $\text{Person} \sqsubseteq \text{Animal}$, for example, entails that the intersection of *Male* and *Person* is also a subclass of *Animal*.

A DL system, however, not only stores terminologies and assertions, but also offers services that reason about them. Typical reasoning tasks for a terminology are to

determine whether a description is satisfiable (i.e., non-contradictory), or whether one DL description is more general than another one (e.g. whether the first subsumes the second, Baader et al 2002). Reasoning problems concerning an ABox are to find out whether it is consistent (i.e. whether it has a model), and whether it entails that a particular individual is an instance of a given concept description. Satisfiability checking of descriptions and consistency checking of sets of assertions are useful to determine whether a knowledge base is meaningful at all. By performing subsumption tests, one can organize the concepts of a terminology into a hierarchy according to their generality. A concept description can also be conceived as a query, which describes the set of objects one is interested in. Thus, with instance tests, one can retrieve the individuals or concepts that satisfy the query (this query based method will be used in the evaluation of our proposal - see chapter 5 below). One important aspect to keep in mind when dealing with reasoning in Description Logics is that they all follow an open world assumption (OWA). This is especially important since knowledge representation systems bear a superficial similarity with database systems. The TBox is similar to the database schema and the ABox similar to the data stored in it. The important difference is that databases adopt closed-world assumptions when answering queries. Namely: if in a database there is no individual that fulfills the query criteria, then the assumption is that such an individual does not exist and that the statement that no such individual exists is true. On the contrary, in the open-world reasoning of Description Logics, if no individual fulfills the query criteria, then the implication is that there is a lack of information. It is not possible to deduce that, since the query is not fulfilled, the negation of the query is true.

The last element showed by the figure 1.3 is that, in any application, a DL system is embedded into a larger environment. External application programs interact with the DL components by querying the knowledge base and by modifying it. Furthermore, a restricted mechanism to add assertions are rules. Rules are an extension of the logical core formalism, which can still be interpreted logically. For a detailed description of all these elements we refer to the *Description Logic Handbook* (Baader et al. 2002).

1.3 Description Logics for Semantic Web Languages

In this section we will briefly present the syntax and semantics of the Description Logics on which OWL (and previous semantic web languages such as OIL, DAML+OIL) is based. Namely we present the family of description logics extending the S family of DLs. The S family of logics is based on an extension of the well known DL *ALC* (see Schmidt-Schauß and Smolka, 1991; Baader et al 2002 for details).

In the literature, in order to better understand the expressivity of the different DLs, a standard notation has been adopted, in which each letter forming the name of a DL corresponds to a specific expressivity requirement. Namely: the letter *S* stands for the basic *ALC DL* (equivalent to the propositional modal logic $K(m)$) extended with transitive roles, *H* stands for role hierarchies (or, equivalently, to inclusion axioms between roles), *O* stands for nominals (classes whose extension is a single individual), *I* for inverse roles (*I*) and (possibly qualified) number restrictions (*Q* if qualified via datatypes, *N* otherwise). The correspondence between the letters and the corresponding syntax and semantics of each logic of the families *S* and *SH* is provided in the figure 1.4 below.

Construct Name	Syntax	Semantics		
atomic concept	A	$A^{\mathcal{I}} \subseteq I^{\mathcal{I}}$	<i>S</i>	
atomic role	R	$R^{\mathcal{I}} \subseteq I^{\mathcal{I}} \times I^{\mathcal{I}}$		
transitive role	$R \in \mathbf{R}_C$	$R^{\mathcal{I}} \mathcal{D} (R^{\mathcal{I}})^C$		
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$		
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$		
negation	$\neg C$	$I^{\mathcal{I}} \setminus C^{\mathcal{I}}$		
exists restriction	$\exists R.C$	$\{x \mid \exists y. \langle x, y \rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$		
value restriction	$\forall R.C$	$\{x \mid \forall y. \langle x, y \rangle \in R^{\mathcal{I}} \text{ implies } y \in C^{\mathcal{I}}\}$		
role hierarchy	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$		<i>H</i>
nominal	$\{o\}$	$\{o^{\mathcal{I}}\}$		<i>O</i>
inverse role	R^-	$\{\langle x, y \rangle \mid \langle y, x \rangle \in R^{\mathcal{I}}\}$	<i>I</i>	
number restrictions	$\geq n P$	$\{x \mid \sharp\{y. \langle x, y \rangle \in P^{\mathcal{I}}\} \geq n\}$	<i>N</i>	
	$\leq n P$	$\{x \mid \sharp\{y. \langle x, y \rangle \in P^{\mathcal{I}}\} \leq n\}$		
qualifying number restrictions	$\geq n P.C$	$\{x \mid \sharp\{y. \langle x, y \rangle \in P^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \geq n\}$	<i>Q</i>	
	$\leq n P.C$	$\{x \mid \sharp\{y. \langle x, y \rangle \in P^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\} \leq n\}$		

Fig. 1.4 Syntax and Semantics of the S family of Description Logics (from Horrocks et al. 2003)

In figure 1.4, the syntax and semantics of these features is schematized. Namely: A is a concept name, C and D are concepts, R and S are roles, R_C is the set of transitive roles, o is an individual name, P is a simple role (i.e., one that is not transitive and has no transitive sub-roles), and n is a non-negative integer. These logics can also be extended with a simple form of concrete domains known as datatypes; this is denoted by appending (D) to the name of the logic, e.g., $SHOIN(D)$.

The OWL languages that we are going to consider in major detail are directly based on these description logics. Namely: OWL Lite is based on $SHIQ DL$ (it correspond to the $SHOIQ DL$ without nominals, and with only functional number restrictions) while OWL DL is based on the $SHOIN(D)$ where the (D) represent the possibility of adding simple form of datatypes on concrete domain (D).

1.4 From Description Logics to Semantic Web languages

As previously mentioned, the OWL language has three sublanguages (OWL Lite, OWL DL and OWL Full), each with certain characteristics. The first two languages have been explicitly designed in order to provide the possibility of having decidable inferences, thus they are based on Description Logics that provide limitations on the expressivity of the language. In OWL Full, instead, all RDF graphs are allowed. The benefits of this expansive style include total upward compatibility with RDF and a greater expressive power. The price for this increased expressivity is, however, that reasoning in OWL Full is undecidable. In the following we focus in major detail on the these sublanguages, trying to put in evidence the main differences with the above mentioned semantic web languages (e.g. RDF, RDFs).

Before going into the details, however, we underline the differences that even the more-limited versions of OWL have with the standard Description Logics. Are, in fact, these differences that move these versions of OWL from the formal Description Logic world to the Semantic Web world. Namely they can be grouped as follows:

- OWL uses URI references as names, and constructs these URI references in the same manner as RDF. It is thus common in OWL to use qualified names as

shorthands for URI references, using, for example, the qualified name `owl:Thing` for the URI reference `http://www.w3.org/2002/07/owl#Thing`.

- OWL gathers information into ontologies, which are generally stored as Web documents written in RDF/XML. Ontologies can import other ontologies, adding the information from the imported ontology to the current ontology.
- OWL uses the facilities of RDF datatypes and XML Schema datatypes to provide datatypes and data values.

Summing up: what makes OWL a Semantic Web language is not its semantics, which is quite standard for a Description Logic, but instead the use of URI references for names, the use of XML Schema datatypes for data values, and the ability to connect to documents in the World Wide Web.

1.4.1 OWL Lite

OWL Lite allows the use of a subset of OWL and RDF(S) vocabulary. The main goal of this language is to guarantee termination of reasoning processes. OWL Lite language prohibits unions and complements, restricts intersections to the implicit intersections in the frame-like class axioms, limits all embedded descriptions to concept names, does not allow individuals to show up in descriptions or class axioms, and limits cardinalities to 0 or 1. These restrictions make OWL Lite similar to the Description Logic *SHIF(D)* (obtained by adding some constraint to *SHOIQ(D)*). Like *SHIF(D)*, in fact, key inferences in OWL Lite can be computed in worst case exponential time (ExpTime). This improvement in tractability comes with relatively little loss in expressive power. Infact, although OWL Lite syntax is more restricted than that of OWL DL, it is still possible to express complex descriptions by introducing new class names and exploiting the implicit negations introduced by disjointness axioms (Horrocks 2003, 2011). Using these techniques, all OWL DL descriptions can be captured in OWL Lite except those containing either individual names or cardinalities greater than 1. In particular, in OWL Lite language it is possible to use 35 out of 40 OWL constructs and 11 of the 33 RDFs constructs. The list of the 33 RDFs constructs and of the 40 OWL construct is taken from Giunchiglia et al. 2010, and is presented in the Appendixes A and B at the end of

the chapter. To define a class in OWL Lite one must use the OWL construct *owl:Class* instead of *rdfs:Class* which is not allowed. Other not allowed constructs in OWL Lite are: *complementOf*, *disjointWith*, *hasValue*, *oneOf* and *unionOf*.

1.4.2 OWL DL

OWL DL is based on *SHOIN(D)* syntax and semantics. OWL DL language can use, as OWL Lite language,¹¹ out of the 33 RDFS constructs. In addition, in OWL DL is possible to use all the 40 constructs of OWL. However, some of these constructs are restricted in order to provide the decidability of the language. In particular, classes cannot be used as individuals, and vice versa. Each individual (or instance) must be an extension of a class and must be necessarily categorized in a class (if there is no more specific class, it must be categorized as belonging to the *owl:Thing* class). Furthermore: individuals cannot be used as properties and vice versa, and properties cannot be used as classes and vice versa. Properties in OWL DL are divided into object properties and datatype properties. Object properties connect instances of two classes, datatype properties connect instances of classes and literals. The restriction in OWL DL allow to maintain a balance between expressivity and complexity. In fact, even if computational complexity is higher than OWL Lite (*SHOIN(D)*) is an expressive Description Logic whose worst case complexity is of nondeterministic exponential time: NExpTime) reasoning in OWL DL remains decidable and correspond to that one of its correspondent description logics.

1.4.3 OWL Full

OWL Full is the most expressive OWL language. Like RDF and RDFS (with which has a complete compatibility), it allows classes to be used as individuals. OWL Full goes beyond OWL DL. For example, in OWL Full, it is possible to impose a cardinality constraint on *rdfs:subClassOf*, is desider. This language can use all the 40 constructs OWL without any restriction imposed on OWL-DL. Moreover the construcr *rdfs:class* and *owl:Class* can be used to define a class. The key difference with respect to OWL DL is that in OWL Full what we can say, e.g. classes, properties etc, can be used as

individuals. The penalty to be paid here is two-fold. First, reasoning in OWL Full is undecidable. Second, the syntax for OWL DL (Horrocks, 2003) is inadequate for OWL Full, and the official OWL exchange syntax, RDF/XML, must be used.

Appendix A: RDF(S) Constructs

This appendix provides a list of the thirty-three RDF(S) constructs excluding the sub-properties of `rdfs:member`.

The RDF(S) constructs are `rdf:about`, `rdf:Alt`, `rdf:Bag`, `rdf:Description`, `rdf:first`, `rdf:ID`, `rdf:List`, `rdf:nil`, `rdf:Object`, `rdf:predicate`, `rdf:Property`, `rdf:resource`, `rdf:rest`, `rdf:Seq`, `rdf:Statement`, `rdf:subject`, `rdf:type`, `rdf:value`, `rdf:XMLLiteral`, `rdfs:Class`, `rdfs:comment`, `rdfs:Container`, `rdfs:ContainerMembershipProperty`, `rdfs:Datatype`, `rdfs:domain`, `rdfs:isDefinedBy`, `rdfs:label`, `rdfs:Literal`, `rdfs:member`, `rdfs:range`, `rdfs:seeAlso`, `rdfs:subClassOf`, and `rdfs:subPropertyOf`. Details of the meaning of the above constructs can be found in the RDF(S) manuals. To provide a few examples, `rdfs:Class` allows to represent a concept, `rdfs:subClassOf` to state that a concept is more specific than another, `rdf:resource` to represent a resource (an instance of a concept), `rdfs:label` to represent a human readable label (for a concept or resource or property), `rdfs:comment` to provide a human readable description of a concept or resource or property.

Appendix B: OWL Constructs

This appendix provides the lists of the forty OWL constructs and eleven RDF(S) constructs that can be used in an OWL representation.

The OWL constructs are `owl:AllDifferent`, `owl:allValuesFrom`, `owl:AnnotationProperty`, `owl:backwardCompatibleWith`, `owl:cardinality`, `owl:Class`, `owl:complementOf`, `owl:DataRange`, `owl:DatatypeProperty`, `owl:DeprecatedClass`, `owl:DeprecatedProperty`, `owl:differentFrom`, `owl:disjointWith`, `owl:distinctMembers`, `owl:equivalentClass`, `owl:equivalentProperty`, `owl:FunctionalProperty`, `owl:hasValue`, `owl:imports`, `owl:incompatibleWith`, `owl:intersectionOf`, `owl:InverseFunctionalProperty`, `owl:inverseOf`, `owl:maxCardinality`, `owl:minCardinality`,

owl:Nothing, owl:ObjectProperty, owl:oneOf, owl:onProperty, owl:Ontology, owl:OntologyProperty, owl:priorVersion, owl:Restriction, owl:sameAs, owl:someValuesFrom, owl:SymmetricProperty, owl:Thing, owl:TransitiveProperty, owl:unionOf, and owl:versionInfo. The RDF(S) constructs are rdf:about, rdf:ID, rdf:resource, rdf:type, rdfs:comment, rdfs:domain, rdfs:label, rdfs:Literal, rdfs:range, rdfs:subClassOf, and rdfs:subPropertyOf. To provide a few examples of the meaning of the constructs above, owl:Class can be used to represent a concept, owl:equivalentClass to state that a concept is equivalent to another, owl:Thing to represent an instance of a concept, owl:sameAs to state that two instances refer the same thing.

Chapter 2.

Representing Non Classical Conceptual Information

After the brief introduction regarding the connection between the Description Logics and the Semantic Web languages, in this chapter we introduce the problem investigated in this research. Namely: the problem of concept representation. And, more specifically, the problem of “non classical” concept representation within the field of formal ontologies. The computational representation of concepts is a central problem for the development of ontologies and knowledge engineering. Concept representation is a multidisciplinary topic of research that involves different disciplines such as Artificial Intelligence, Philosophy, Cognitive Psychology and Cognitive Science in general. However, the notion of concept itself turns out to be highly disputed and problematic. In our opinion, one of the causes of this current state of affairs is that the very notion of concept is, to a certain extent, heterogeneous and encompasses different cognitive phenomena. This results in a strain between conflicting requirements such as compositionality, on the one hand, and the need to represent prototypical information on the other. This has several consequences for the practice of knowledge engineering as well as the technology of formal ontologies. In this chapter we propose an analysis of this situation.

The rest of the chapter is organised as follows. In section 2.1, we point out some differences between the way concepts are conceived in philosophy and psychology. In section 2.2 and 2.3 we introduce the conflicting requirements (such as compositionality, on the one hand and the need to represent prototypical information on the other) characterizing the history of concept based representations. Then, in section 2.4 we argue that AI research in some way shows traces of these contrasting needs. In particular, the requirement of compositional, logical style semantics conflicts with the need to represent concepts in terms of the typical traits that allow for exceptions. In section 2.5. we point out the necessity for artificial conceptual

systems to represent and reason on non classical concepts and prototypical information in order to deal with so called “common sense” concepts. This necessity, in our opinion, can be covered taking into account some evidence from cognitive analysis of the human way of organizing and processing information. In this view, our basic assumption is that knowledge representation systems whose design takes into account evidence from experimental psychology may register better performance in real life applications (e.g. specifically in the fields of information retrieval and semantic webs).

In section 2.6, we review several attempts to introduce non classical representation and reasoning in the field of knowledge representation, while paying particular attention to description logics. Finally, in section 2.7, we identify several possible suggestions coming from different aspects of cognitive research in order to overcome this problem. Namely: (i) the distinction between two different types of reasoning processes, developed within the context of the so-called “dual process” accounts of reasoning; (ii) the proposal to keep prototypical effects separate from the compositional representation of concepts; and (iii) the possibility to develop hybrid, prototype and exemplar-based representations of concepts. All these elements representing the cognitive background of our approach will be more deeply described in next chapters.

2.1 Concepts in Philosophy and Psychology

Within the field of cognitive science, the notion of concept is highly disputed and problematic. Artificial intelligence (from now on AI) and, in general, the computational approach to cognition reflect this current state of affairs. Conceptual representation seems to be constrained by conflicting requirements, such as compositionality, on the one hand and the need to represent prototypical information on the other. A first problem (or, better, a first symptom indicating that a problem exists) lies in the fact that the use of the term “concept” in the philosophical tradition is not homogeneous with the use of the same term in empirical psychology (see e.g. Dell’Anna and Frixione 2010).

Briefly³, we can say that in cognitive psychology a concept is essentially intended as the mental representations of a category, and the emphasis is on processes such as categorisation, induction and learning. According to philosophers, concepts are above all the components of thoughts. Even if we leave aside the problem of specifying exactly what thoughts are, this requires a more demanding notion of concept. In other words, some phenomena that are classified as “conceptual” by psychologists turn out to be “nonconceptual” for philosophers. There are, thus, mental representations of categories that philosophers would not consider genuine concepts. For example, according to many philosophers, concept possession involves the ability to make high level inferences explicit and also sometimes the ability to justify them (Peacocke 1992; Brandom 1994). This clearly exceeds the possession of the mere mental representation of categories. Moreover, according to some philosophers, concepts can be attributed only to agents who can use natural language (i.e. only adult human beings). On the other hand, a position that can be considered in some sense representative of an “extremist” version of the psychological attitude towards concepts is expressed by Lawrence Barsalou in an article symptomatically entitled “Continuity of the conceptual system across species” (Barsalou 2005). He refers to knowledge of scream situations in macaques, which involve different modality-specific systems (auditory, visual, affective systems, etc.). Barsalou interprets these data in favour of the thesis of a continuity of conceptual representations in different animal species, in particular between humans and non-human mammals: “this same basic architecture for representing knowledge is present in humans. [...] knowledge about a particular category is distributed across the modality-specific systems that process its properties” (p. 309). Therefore, according to Barsalou, a) we can also speak of a “conceptual system” in the case of non-human animals; b) low-level forms of categorisation which depend on some specific perceptual modality also belong to the conceptual system. Elizabeth Spelke’s experiments on infants (see e.g. Spelke 1994; Spelke and Kinzler 2007) are symptomatic of the difference in approach between psychologists and philosophers. These experiments demonstrate that some extremely general categories are very precocious and presumably

³ Things are made more complex by the fact that also within the two fields considered separately this notion is used in a heterogeneous way, as we shall concisely see in the following. Consequently, the following characterisation of the philosophical and psychological points of view is highly schematic.

innate. According to the author, they show that newborn babies already possess certain *concepts* (e.g., the physical object concept). However, some philosophers (Bermudez 1995, Bermudez and Cahen 2011) have interpreted these same data as a paradigmatic example of the existence of *nonconceptual* contents in agents (babies) who have yet to develop a conceptual system.

2.2 Compositionality

The fact that philosophers consider concepts mainly as the components of thoughts has given greater emphasis to *compositionality*, as well as to other related features, such as productivity and systematicity, which are often ignored by the psychological treatment of concepts. On the other hand, it is well known that compositionality is at odds with *prototypicality effects*, which are crucial in most psychological characterisations of concepts (we shall develop this point in greater detail in the next section). Let us first consider the *compositionality* requirement. In a compositional system of representations, we can distinguish between a set of *primitive*, or *atomic*, *symbols* and a set of *complex symbols*. Complex symbols are generated from primitive symbols through the application of a set of suitable recursive syntactic rules (generally, a potentially infinite set of complex symbols can be generated from a finite set of primitive symbols). Natural languages are the paradigmatic example of compositional systems: primitive symbols correspond to the elements of the lexicon (or, better, to morphemes), and complex symbols include the (potentially infinite) set of all sentences. In compositional systems, the meaning of a complex symbol *s* functionally depends on the syntactic structure of *s* as well as the meaning of primitive symbols in it. In other words, the meaning of complex symbols can be determined by means of recursive semantic rules that work in parallel with syntactic composition rules. This is the so-called *principle of compositionality of meaning*, which Gottlob Frege identified as one of the main features of human natural languages. In classical cognitive science, it is often assumed that mental representations are compositional. One of the clearest and most explicit formulations of this assumption was proposed by Jerry Fodor and Zenon Pylyshyn (1988). They claim that the compositionality of mental representations is mandatory in order to explain some fundamental cognitive phenomena. In the first

place, human cognition is *generative*: in spite of the fact that the human mind is presumably finite, we can conceive and understand an unlimited number of thoughts that we have never encountered before. Moreover, the *systematicity* of cognition also seems to depend on compositionality: the ability to conceive certain contents is systematically related to the ability to conceive other contents. For example, if somebody can understand the sentence *the cat chases a rat*, then (s)he is presumably also able to understand *a rat chases the cat*, by virtue of the fact that the forms of the two sentences are syntactically related. We can conclude that the ability to understand certain propositional contents systematically depends on the compositional structure of the contents themselves. This can easily be accounted for if we assume that mental representations have a structure similar to a compositional language.

2.3 Against "Classical" Concepts

Compositionality is less important for many psychologists. In the field of psychology, most research on concepts moves from the critiques to the so-called classical theory of concepts, i.e. the traditional point of view according to which concepts can be defined in terms of necessary and sufficient conditions. Empirical evidence favours those approaches to concepts that account for prototypical effects. The central claim of the classical theory of concepts (i.e.) is that every concept c is defined in terms of a set of features (or conditions) f_1, \dots, f_n that are individually necessary and jointly sufficient for the application of c . In other words, everything that satisfies features f_1, \dots, f_n is a c , and if anything is a c , then it must satisfy f_1, \dots, f_n . For example, the features that define the concept *bachelor* could be *human*, *male*, *adult* and *not married*; the conditions defining *square* could be *regular polygon* and *quadrilateral*. This point of view was unanimously and tacitly accepted by psychologists, philosophers and linguists until the middle of the 20th century. The first critique of classical theory is due to a philosopher: in a well known section from the *Philosophical Investigations*, Ludwig Wittgenstein observes that it is impossible to identify a set of necessary and sufficient conditions to define a concept such as GAME (Wittgenstein, 1953, § 66). Therefore, concepts exist which cannot be defined according to classical theory, i.e. in terms of necessary and

sufficient conditions. Concepts such as GAME rest on a complex network of *family resemblances*. Wittgenstein introduces this notion in another passage in the *Investigations*: «I can think of no better expression to characterise these similarities than “family resemblances”; for the various resemblances between members of a family: build, features, colour of eyes, gait, temperament, etc. etc.» (*ibid.*, § 67). Wittgenstein's considerations were corroborated by empirical psychological research: starting from the seminal work by Eleanor Rosch (1975), with the psychological experiments that showed how common-sense concepts do not obey the requirement of the classical theory⁴: common-sense concepts cannot usually be defined in terms of necessary and sufficient conditions (and even if for some concepts such a definition is available, subjects do not use it in many cognitive tasks). Concepts exhibit *prototypical effects*: some members of a category are considered better instances than others. For example, a robin is considered a better example of the category of birds than, say, a penguin or an ostrich. More central instances share certain typical features (e.g. the ability of flying for birds, having fur for mammals) that, in general, are neither necessary nor sufficient conditions.

Prototypical effects are a well established empirical phenomenon. However, the characterisation of concepts in prototypical terms is difficult to reconcile with the compositionality requirement. According to a well known argument by Jerry Fodor (1981), prototypes are not compositional (and, since concepts in Fodor's opinion must be compositional, concepts cannot be prototypes). In brief, Fodor's argument runs as follows: consider a concept like PET FISH. It results from the composition of the concept PET as well as the concept FISH. However, the prototype of PET FISH cannot result from the composition of the prototypes of PET and FISH. For example, a typical PET is furry and warm, a typical FISH is greyish, but a typical PET FISH is neither furry and warm nor greyish. Moreover, things are made more complex by the fact that, even within the two fields of philosophy and psychology considered separately, the situation is not very encouraging. In neither of the two disciplines does a clear, unambiguous and coherent notion of concept seem to emerge. Consider for example psychology. Different positions and theories on the nature of concepts are available

⁴ On the empirical inadequacy of the classical theory and the psychological theories of concepts see (Murphy 2002).

(prototype view⁵, exemplar view, theory theory) that can hardly be integrated. From this point of view, the conclusions of Murphy (2002) are of great significance, since in many respects this book reflects the current status of empirical research on concepts. Murphy contrasts the approaches mentioned above in relation to different classes of problems, including learning, induction, lexical concepts as well as children's concepts. His conclusions are rather discouraging: the result of comparing the various approaches is that "there is no clear, dominant winner" (ibid., p. 488) and that "[i]n short, concepts are a mess" (p. 492). This situation persuaded some scholars to doubt whether concepts constitute a homogeneous phenomenon from the point of view of a science of the mind (see e.g. Machery 2005 and 2009; Frixione 2007).

2.4. Concept Representation in Artificial Intelligence

The situation outlined in the section above is, to some extent, reflected by the state of the art in AI and, in general, in the field of computational modelling of cognition. This research area often seems to hesitate between different (and hardly compatible) points of view. In AI, the representation of concepts is faced mainly within the field of knowledge representation (KR). Symbolic KR systems (KRs) are formalisms whose structure is, broadly speaking, language-like. This usually entails assuming that KRs are compositional. In their early development (historically corresponding to the late 1960s and the 1970s), many KRs oriented to conceptual representations attempted to take into account suggestions from psychological research. Examples are early semantic networks and frame systems. Frame and semantic networks were originally proposed as alternatives to the use of logic in KR. The notion of frame was developed by Marvin Minsky (1975) as a solution to the problem of representing structured knowledge in AI systems⁶. Both frames and most semantic networks allowed for the possibility to characterise concepts in terms of prototypical information. However, such early KRs were usually characterised in a rather rough and imprecise way. They lacked a clear

⁵ Note that the so-called prototype view does not coincide with the acknowledgement of prototypical effects: as stated before, prototypical effects are a well established phenomenon that all psychological theories of concepts are bound to explain; the prototype view is a particular attempt to explain empirical facts concerning concepts (including prototypical effects). On these aspects, see again Murphy 2002.

⁶ Many of the original articles describing these early KRs can be found in (Brachman & Levesque 1985), a collection of classic papers of the field.

formal definition, with the study of their meta-theoretical properties being almost impossible. When AI practitioners tried to provide a stronger formal foundation to concept oriented KRs, it turned out to be difficult to reconcile compositionality and prototypical representations. As a consequence, they often chose to sacrifice the latter. In particular, this is the solution adopted in a class of concept-oriented KRs which were (and still are) widespread within AI, namely the class of formalisms that stem from the so-called structured inheritance networks and the KL-ONE system (Brachman and Schmolze 1985). Such systems were subsequently called terminological logics, and today are usually known as *description logics* (DLs) (Baader et al. 2002). We already presented in greater detail this class of formalisms in chapter 1. A standard inference mechanism for this kind of network is *inheritance*. The representation of prototypical information in semantic networks usually takes the form of allowing exceptions to inheritance. Networks in this tradition do not admit exceptions to inheritance, and therefore do not allow for the representation of prototypical information. In fact, representations of exceptions cannot be easily accommodated with other types of inference defined on these formalisms, first and foremost concept classification (Brachman 1985). Since the representation of prototypical information is not allowed, inferential mechanisms defined on these networks (e.g. inheritance) can be traced back to classical logical inferences. In more recent years, representation systems in this tradition have been directly formulated as logical formalisms (the above mentioned description logics, Baader et al., 2002), in which Tarskian, compositional semantics is directly associated to the syntax of the language. Logical formalisms are paradigmatic examples of compositional representation systems and, as a result, this kind of system fully satisfies the compositionality requirement. This has been achieved at the cost of not allowing exceptions to inheritance. However, in so doing, we have forsaken the possibility to represent concepts in prototypical terms. From this point of view, such formalisms can be seen as a revival of the classical theory of concepts, in spite of its empirical inadequacy in dealing with most common-sense concepts. As we have seen in the previous chapter, nowadays DLs are widely adopted within many application fields, in particular within that of the representation of ontologies. For example, the OWL

(Web Ontology Language - see sect. 1.4 of chapter 1) system⁷ is a formalism in this tradition that has been endorsed by the World Wide Web Consortium for the development of the semantic web.

2.5 Artificial Systems: Why Prototypical Effects are Needed

Prototypical effects in categorisation and, in general, in category representation are not only crucial for the empirical study of human cognition, but they are also of the greatest importance in representing concepts in artificial systems. Let us first consider human cognition. Under what conditions should we say that somebody *knows* the concept DOG (or, in other terms, that (s)he possesses an adequate mental representation of it)? It is not easy to say. However, if a person does not know that, for example, dogs usually bark, that they typically have four legs and that their body is covered with fur, that in most cases they have a tail and that they wag it when they are happy, then we probably should conclude that this person does not grasp the concept DOG. Nevertheless, all these pieces of information are neither necessary nor sufficient conditions for being a dog. In fact, they are traits that characterise dogs in typical (or prototypical) cases. The problem is exactly the same if we want to represent knowledge in an artificial system. Let us suppose that we want to provide a computer program with a satisfactory representation of DOG. Then we probably also want to represent the kind of information mentioned above: for many applications, a representation of DOG that does not include the information that dogs usually bark is a bad representation also from a technological point of view. Therefore, if a system does not allow information to be represented in typical/prototypical terms (as is the case of standard description logics), then it is not adequate in this respect. With standard DLs, the only way to tackle this problem should be the recourse to tricks or ad hoc solutions (as often happens in many applications). The concept DOG is not exceptional from this point of view. The majority of everyday concepts behave in this way. For most concepts, a classical definition in terms of necessary and sufficient conditions is not available (or, even if it is available, it is unknown to the agent). On the other hand, it may be that we know the

⁷ <http://www.w3.org/TR/owl-features/>

classical definition of a concept, but typical/prototypical knowledge still plays a central role in many cognitive tasks. Consider the following example: nowadays most people know necessary and sufficient conditions for being WATER: water is exactly the chemical substance whose formula is H_2O , i.e., the substance whose molecules are formed by one atom of oxygen and two atoms of hydrogen. However, in most cases of everyday life, when we categorise a sample of something as WATER, we do not take advantage of this piece of knowledge. We use such prototypical traits such as the fact that (liquid) water is usually a colourless, odourless and tasteless fluid. As a further example, consider the concept GRANDMOTHER. Everybody knows a classical definition for it: x is the grandmother of y if and only if x is the mother of a parent of y . However, in many cases we do not use this definition to categorise somebody as a grandmother. We resort to typical traits: grandmothers are old women who take care of children, who are tender and polite with them, and so on. Once more, the problem is not different in the case of artificial systems: generally a system that has to categorise something as WATER cannot perform chemical analyses, and it must trust to prototypical evidence. With these example our aim is that one of pointing out that the use of prototypical knowledge in cognitive tasks such as categorisation is not a “fault” of the human mind, as it could be the fact that people are prone to fallacies and reasoning errors (leaving aside the problem of establishing whether recurrent errors in reasoning could have a deeper “rationality” within the general economy of cognition). It has to do with the constraints that concern every finite agent that has a limited access to the knowledge which is relevant for a given task. This is the case of both natural and artificial cognitive systems.

2.6 Non-classical Concepts in Computational Ontologies

Within symbolic, logic oriented KR, rigorous approaches exist that make it possible to represent exceptions, and that would therefore be, at least in principle, suitable for representing “non-classical” concepts. Examples are fuzzy logics and non-monotonic formalisms. Therefore, the adoption of logic oriented semantics is not necessarily incompatible with prototypical effects. Nevertheless, such approaches pose various

theoretical and practical difficulties, with many problems remaining unsolved. In this section, we review some recent proposals to extend concept-oriented KR, and in particular DLs, with a view to representing non-classical concepts. Recently, different methods and techniques have been adopted to represent non-classical concepts within computational ontologies. These are based on extensions of DLs as well as standard ontology languages such as OWL. The different proposals that have been put forward can be grouped into three main classes: a) fuzzy approaches, b) probabilistic and Bayesian approaches, c) approaches based on non-monotonic formalisms.

a) As far as the integration of *fuzzy logics* in DLs and in ontology oriented formalisms is concerned, see for example Gao and Liu 2005, and Calegari and Ciucci 2007. Stoilos et al. (2005) propose a fuzzy extension of OWL, f-OWL, able to capture imprecise and vague knowledge, and a fuzzy reasoning engine that lets f-OWL reason about such knowledge. Bobillo and Straccia (2009) propose a fuzzy extension of OWL 2 for representing vague information in semantic web languages. However, it is well known (Osherson and Smith 1981) that approaches to prototypical effects based on fuzzy logic encounter some difficulty with compositionality.

b) The literature offers also several *probabilistic generalizations* of web ontology languages. Many of these approaches, as pointed out in Lukasiewicz and Straccia (2008), focus on combining the OWL language with probabilistic formalisms based on Bayesian networks. In particular, Da Costa and Laskey (2006) suggest a probabilistic generalization of OWL, called PR-OWL, whose probabilistic semantics is based on multi-entity Bayesian networks (MEBNs); Ding et al. (2006) propose a probabilistic generalization of OWL, called Bayes-OWL, which is based on standard Bayesian networks. Bayes-OWL provides a set of rules and procedures for the direct translation of an OWL ontology into a Bayesian network. A general problem of these approaches could consist in avoiding arbitrariness in assigning weights in the translation from traditional to probabilistic formalisms.

c) The role of *non monotonic reasoning* in the context of formalisms for the ontologies is actually a debated problem. According to many KR researches, non monotonic logics are expected to play an important role for the improvement of the reasoning capabilities of ontologies and of the Semantic Web applications. In the field

of non monotonic extensions of DLs, Baader and Hollunder (1995) propose an extension of ALCF system based on Reiter's default logic⁸. The same authors, however, point out both the semantic and computational difficulties of this integration and, for this reason, propose a restricted semantics for open default theories, in which default rules are only applied to individuals explicitly represented in the knowledge base. Since Reiter's default logic does not provide a direct way of modelling inheritance with exceptions in DLs, Straccia (1993) proposes an extension of H-logics (*Hybrid KL-ONE style logics*) able to perform *default inheritance reasoning* (a kind of default reasoning specifically oriented to reasoning on taxonomies). This proposal is based on the definition of a priority order between default rules. Donini et al. (1998, 2002), propose an extension of DL with two non monotonic epistemic operators. This extension allows one to encode Reiter's default logic as well as to express epistemic concepts and procedural rules. However, this extension presents a rather complicated semantics, so that the integration with the existing systems requires significant changes to the standard semantics of DLs. Bonatti et al. (2006) propose an extension of DLs with circumscription. One of motivating applications of circumscription is indeed to express prototypical properties with exceptions, and this is done by introducing "abnormality" predicates, whose extension is minimized. Giordano et al. (2007) propose an approach to defeasible inheritance based on the introduction in the ALC DL of a typicality operator \mathbf{T} ⁹, which allows to reason about prototypical properties and inheritance with exceptions. This approach, given the non monotonic character of the \mathbf{T} operator, encounters some problems in handling inheritance (an example is what the authors call the *problem of irrelevance*). Katz and Parsia argue that ALCK, a non monotonic DL extended with the epistemic operator \mathbf{K} ¹⁰ (that can be applied to concepts or roles) could represent a model for a similar non monotonic extension of OWL. In fact, according to the authors, it would be possible to create "local" closed-world assumption conditions, in order the reap the benefits of non monotonicity without giving up OWL's open-world semantics in general.

⁸ The authors pointed out that "Reiter's default rule approach seems to fit well into the philosophy of terminological systems because most of them already provide their users with a form of 'monotonic' rules. These rules can be considered as special default rules where the justifications - which make the behavior of default rules nonmonotonic - are absent".

⁹ For any concept C , $\mathbf{T}(C)$ are the instances of C that are considered as "typical" or "normal".

¹⁰ The \mathbf{K} operator could be encoded in RDF/XML syntax of OWL as property or as annotation property.

A different approach, investigated by Klinov and Parsia (2008), is based on the use of the OWL 2 annotation properties (APs) in order to represent vague or prototypical, information. The limit of this approach is that APs are not taken into account by the reasoner, and therefore have no effect on the inferential behaviour of the system (Bobillo and Straccia 2009).

2.7 Some Suggestions from Cognitive Science

Even though a relevant field of research exists, in the scientific community there is no agreement on the use of non-monotonic and, in general, non-classical logics in ontologies. For practical applications, systems that are based on classical Tarskian semantics and that do not allow for exceptions (as it is the case of “traditional” DLs), are still preferred. Some researchers, such as Pat Hayes (2001), argue that non-monotonic logics (and, therefore, the non-monotonic “machine” reasoning for the semantic web) can be adopted for local uses only or for specific applications because it is “unsafe on the web”. Nevertheless, the question about which “logics” must be used in the semantic web (or, at least, to what degree and in which cases certain logics could be useful) is still open.

Empirical results from cognitive psychology show that most common-sense concepts cannot be characterised in terms of necessary/sufficient conditions. Classical, monotonic DLs seem to capture the compositional aspects of conceptual knowledge, but are inadequate in representing prototypical knowledge. However, a “non-classical” alternative, a general DL able to represent concepts in prototypical terms still does not exist.

As a possible way out, we outline a tentative proposal based on several suggestions from cognitive science. Some recent trends in psychological research favour the hypothesis that reasoning is not a unitary cognitive phenomenon. At the same time, empirical data on concepts seem to suggest that prototypical effects could stem from different representation mechanisms. To this end, we identify some suggestions that, in our opinion, could be useful in developing artificial representation systems, namely: (i) the distinction between two different types of reasoning processes, which has been developed within the context of the so-called “dual process” accounts of reasoning (see

Chapter 4.2.1); (ii) the proposal to keep prototypical effects separate from the compositional representation of concepts (sect. 4.2.2); and (iii) the possibility to develop hybrid, prototype and exemplar-based representations of concepts (sect. 4.2.3). In particular in the next chapter we focus our attention on the prototype and exemplar theories of concept representation and reasoning developed in the field of cognitive psychology. As we will argue, in fact, it is our opinion that maintain both the representational level within a unique architectural framework can improve both the quality of the information represented within a knowledge base and, at least in principle, pose the conditions for the realization of a non monotonic reasoning module for approximate categorization based on both the representations.

Chapter 3.

Models of Cognition: Prototypes and Exemplars to Explain the Typicality

In the last 30 years, the results coming from the research in cognitive science demonstrated the inadequacy of the so called “classical” theory of concepts - according to which concepts can be defined in terms of sets of necessary and sufficient conditions – for the explanation of such processes as conceptualization, categorization and common sense reasoning.

The failure of this theory, and of its purely compositional approach to the semantics, revealed, as counterpart, the role played by typicality traits in the above mentioned processes, thus representing a real shift of paradigm in the study of both natural and artificial concept oriented systems.

In this chapter we focus on the models of typicality deriving from the research of the last 30 years in cognitive science and psychology. After a brief, and necessarily not exhaustive, review of the main models (section 3.1) we focus on the differences among the proposed theories, with a particular attention to prototype and exemplar based approaches (section 3.2). In the last part of this section we analyze an emerging research direction trying to provide a unifying approach to typicality. In section 3.3 we investigate the dynamics of the processes implied by these two different views. Then, in section 3.4, we account for the problem of automatic categorization in the field of machine learning, investigating the contact point among this area and cognitive science research about conceptual categorization. In this section we also describe how machine learning research seems to be going towards a unified view of typicality in automatic classification, following, in a certain sense, the way indicated by psychological research.

The main goal of this chapter is to present the theoretical foundations and implications of typicality in concept representation and reasoning. It is our opinion that the suggestions coming from such different research areas - as cognitive sciences and machine learning - could be very fruitful to face the problem of representing and reasoning on typicality even in the field of knowledge representation.

3.1. Theory of Concepts – an overview

Within the field of psychology, different positions and theories on the nature of concepts are available. Usually, they are grouped in three main classes, namely prototype views, exemplar views and theory-theories (see e.g. Murphy 2002, Machery 2009). All of them are assumed to account for (some aspects of) prototypical effects in conceptualisation.

According to the prototype view, knowledge about categories is stored in terms of prototypes, i.e. in terms of some representation of the “best” instances of the category. For example, the concept CAT should coincide with a representation of a prototypical cat. In the simpler versions of this approach, prototypes are represented as (possibly weighted) lists of features.

According to the exemplar view, instead, a given category is mentally represented as a set of specific exemplars explicitly stored within memory: the mental representation of the concept CAT is the set of the representations of (some of) the cats we encountered during our lifetime.

Theory-theories approaches adopt some form of holistic point of view about concepts. According to some versions of the theory-theories, concepts are analogous to theoretical terms in a scientific theory. For example, the concept CAT is individuated by the role it plays in our mental theory of zoology. In other version of the approach, concepts themselves are identified with micro-theories of some sort. For example, the concept CAT should be identified with a mentally represented micro-theory about cats.

These approaches turned out to be not mutually exclusive. Rather, they seem to succeed in explaining different classes of cognitive phenomena, and many researchers hold that all of them are needed to explain psychological data. In the following pages a more detailed overview is presented. We will not focus our attention on the theory-theory

approach, since it is in some sense more vaguely defined if compared with the other two points of view and, for this reason, its computational treatment seems to be less feasible.

3.2 Prototype and Exemplar Theories

Prototypes and exemplars theories represent different approaches that have been developed with the aim of modeling and explaining the aspects of typicality effects in humans' conceptualization and categorization. Historically these two views have been seen as contrasting and unconciliable among them. More recently, however, there is a growing trend in cognitive science to consider these two approaches as complementary in explaining the typicality issues (see the section 3.4 for further details). Following this direction, it is our opinion that the two theories, jointly, can be able to cover and explain some complex aspect of typicality in concept representation and reasoning. Therefore, both the views can provide a strong background from which to extract many suggestions. In the following subsections, we will go into the details of both approaches with the aim of illustrate their main features, their respective pros and cons, and the different assumptions made by the two approaches regarding the reasoning processes in which they are involved.

3.2.1 Prototype Theory

In the psychological literature it is possible to individuate different prototype-based theories. As reported in Machery (2009), they vary depending on how the nature of the knowledge stored in prototypes is characterized. Depending on the theory, prototypes can consist of:

- (i) knowledge about properties that objects either possess or do not possess
- (ii) knowledge about properties that objects possess to a certain degree.

The property *having fins* is an example of the first type of property (an entity can have or not have fins). The property *being salty*, instead, is an example of the second type: a substance can be more or less salty. This second type of property can also be discrete or

continuous. According to Smith and Medin (1981), prototype models that focus on the first type of property are usually called *featural models*, while models that focus on the second type of property are usually called *dimensional models*. Moreover, depending on the theory, prototypes represent either the *typical* properties of categories, the *cue-valid* properties or the properties that are *both cue-valid and typical*. In the first case, prototypes are supposed to store some knowledge about the typical properties of a class. A property P is typical of a class C if and only if the probability that an object possesses P given that it is a member of C is high. For example: *having four legs* is a typical property of dogs (usually the dogs have four legs). Knowing which properties are typical of a class is particularly useful in order to draw inductions about this class. For example: let us suppose that we have an element X that has 4 legs and that is a DOG. The process of induction consists in the inference according to which, starting from these premises, we draw the conclusion that “every DOG has four legs”. The given example, and the related inferential mechanism, can be even easily formalized in First Order Predicate Logic in the following way:

(i) *Known information: usually a Dog has 4legs*

$$\frac{\begin{array}{c} 4legs(a) \\ Dog(a) \end{array}}{\forall x (Dog(x) \rightarrow 4legs(x))}$$

Of course this inference is not *valid* from a logical point of view: it is a non monotonic, defeasible inference, which is cognitively plausible (and in many cases reliable).

According to other theories, prototypes store some knowledge about the highly cue-valid properties of concepts (see e.g., Hampton 1993). A property has a high cue validity if, statistically, it is very informative about the class membership. A high-cue validity feature is one which conveys more information about the category or class variable. For example: *To woof* is a highly cue-valid property of dogs. *Having four legs* is not a highly cue-valid property of dogs even if it is a typical trait of being a dog (a lot of mammals have four legs). Knowing which properties are highly cue-valid for a class is particularly useful for the reasoning task of categorization. For example: if, at a given element X, it is assigned the property *to woof* (which is a high cue-valid property of

dogs), then there is an high degree of probability that X is a DOG. Furthermore, there is also an high degree of probability that humans categorize, by default, the element X as DOG. Because the property “woof” (even if it is neither necessary nor sufficient for being a DOG) represents, from a cognitive point of view, a highly informative feature for that class membership assignment. Therefore, as we have seen in this example, knowing which properties are cue-valid for a class is particularly useful in order to draw non monotonic forms of categorization about this class.

Finally, according to some approaches, prototypes store both typical and cue valid properties. In Jones (1983), for example, prototypes are assumed to store some knowledge about the properties that maximize some weighted function of typicality and cue-validity (for a more detailed literature on the subject see, again, Machery 2009).

A classical, well known, attempt of representing prototypes in Artificial Intelligence has been developed by Marvin Minsky, who introduced, in the mid '70s, the notion of *frame* in the field of Knowledge Representation (Minsky, 1975). A frame is a knowledge representation structure able to represent prototypical information and to perform some forms of non monotonic reasoning (e.g. exceptions to inheritance). In a frame based representation, concepts are represented according to some prototypical, and cognitively relevant, traits expressed in terms of slots, attributes and values. A slot is composed by an attribute and a value. The values assigned to certain attributes can be default values.

The illustration 3.1 graphically shows the typical structure of a frame.

Frame 1	
Concept 1	
Attribute 1	Value 1
Attribute 2	Value 2
Attribute 3	Value 3
...	...

Figure 3.1: Example of a frame based representation

3.2.2 Criticisms

In psychological literature many criticisms to the prototype theories of concepts have been proposed. A first criticism regards how to explain why, among the numerous typical (or cue-valid) properties of the members of a category, prototypical concepts represent only some of them. Psychologists and philosophers have repeatedly highlighted this issue (Smith and Medin 1981; Machery 2009), but answers in this direction are not yet arrived. Another well known criticism comes from Pinker and Prince (1996). They suggest that, in some domains, concepts can be considered as definitions, while, in other domains, concepts are prototypes or exemplars.¹¹ They argue, for example, that kinship concepts (e.g., UNCLE) and legal concepts are well characterized by the classical theory. Also mathematical concepts can be assigned to this category: e.g. in geometry a TRIANGLE can be easily defined as a POLYGON with 3 corners and 3 sides. This criticism, however, rather than demonstrating the invalidity of the prototype theory, demonstrates that it cannot be applied to certain concepts in specific, well structured, domains. Namely: it does not apply to such domains in which there is no space for typicality.

Other criticisms derive from the so-called heterogeneity hypothesis or from the hybrid theory of concepts, according to which a single concept can have a double level representation, or it can correspond two different concepts representing different levels of information (see the section 3.4 below for further details).

Historically, in the psychological literature a direct antagonist of the prototype theory was the exemplars based approach. In the next section we provide an overview of the exemplar theories in order to underline which are the differences between the two approaches.

¹¹ These authors also concede that to some concepts can be associated both a definition and a prototype. In these cases, they seem to endorse the so called “heterogeneity hypothesis”. This position will be described in major details in the section 3.4.

3.2.3 Exemplar theory

The exemplar paradigm of concepts is built around the idea that concepts are *sets of exemplars* that are stored in our long term memory. In this perspective, our knowledge about a certain concept (let consider, for example, the concept DOG) comes from the accumulated knowledge deriving from all the exemplars of dogs encountered during our lifetime (e.g. Fido, Rin Tin Tin, Lassie etc.). An exemplar represents, in other words, *a body of knowledge about the properties possessed by a particular member of a class* (Machery, 2009). Exemplar based models have posed a strong emphasis on categorization process. According to Palmeri and Gauthier (2004, 294): “Exemplar models assume that recognition, categorization and identification depend on stored instances of experienced objects.”. Similarly Medin and Schaffer write (1978, 209-210): “The general idea of exemplars based models is that categorization judgments are based on the retrieval of stored exemplar information”.

In order to better indicate the way in which exemplars have been usually represented according to this theory, we introduce one of the best known exemplar model of concepts: the Context Model proposed by Medin and Schaffer’s (1978). In Medin and Schaffer’s model, the exemplars are represented as follows: four (independent) properties or “dimension” (for instance, color) are given. For each of them, exemplars can assume a dichotomic value. For example, color can have the values *red* or *blue*. So values can be represented by 0 and 1. Some values may not be specified, because people may have selectively attended to some properties of the encountered category members. Thus, Medin and Schaffer represent the exemplar information in the following way:

$$\begin{array}{ll} 111?-A(A_1) & 10?0-A(A_2) \\ 00?1-B(B_1) & 110?-B(B_2), \end{array}$$

where the question marks indicate that information that would differentiate value 1 and value 0 on that dimension either has not been stored, or cannot be accessed.

In the Context Model, exemplars could thus be represented as follows:

Concept A	
Exemplar A₁	
Dimension 1	Value 1
Dimension 2	Value 1
Dimension 3	Value 1
Dimension 4	Unknown value

Concept A	
Exemplar A₂	
Dimension 1	Value 1
Dimension 2	Value 0
Dimension 3	Unknown value
Dimension 4	Value 0

Figure 3.2: Exemplars in the Context Model, adapted from Machery (2009)

A development of this proposal is the Nosofsky's influential Generalized Context Model of categorization (Nosofsky 1986). It will be presented in the section 3.4.2.1 in order to show the basic mechanisms implied by exemplar theories in concept categorization tasks. In such processes, in fact, it is possible to individuate some of the main differences between prototype and exemplar approaches. For this reason the next sections will be dedicated to the models of categorization proposed within both prototype and exemplar perspectives.

3.3 Prototype based model of categorization

Prototype-based models proposed in literature share some key properties. One of the most important properties is that cognitive processes are assumed to involve the

computation of the *similarity* between prototypes and other representations. For example, the categorization decisions are supposed to depend on the computation of the similarity between prototypes and the representation of the target concepts. Let us consider the following example: suppose that we want to categorize the element Fido that has certain characteristics (and suppose also that Fido is a DOG). The process of categorizing Fido as a dog results from the process indicated in Figure 3.3. The first phase starts when we possess some information (perceptual or of some other kind) concerning Fido. Then the available prototypical representations are retrieved from the long-term memory and compared with the representation of Fido the similarity between these representations is computed (the degree of similarity depends on how many properties are represented by both the prototype and the representation of Fido) and, finally, the categorization decision that Fido is a dog follows from the high degree of similarity between the prototype of dog and the representation of Fido (for a brief overview of the different ways to calculate the semantic similarities among concepts. see chapter 4).

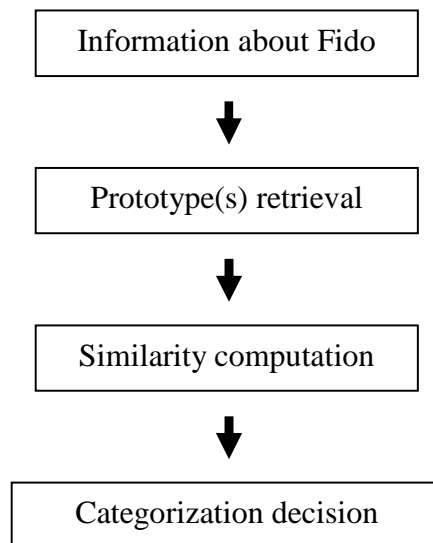


Figure 3.3: A Sequential Schema of Psychological Categorization in Prototype

Another relevant property of these models is that the similarity computation is usually assumed to be *linear* (Medin and Schaffer, 1978). In linear models, a property that is shared by the target and the prototype increases the similarity between the target and the prototype independently of whether other properties are shared by them. For example, the fact that the dog of my neighbor, Fido, has a property that matches my prototype of dog (e.g. *barking*) increases the similarity between the representation of Fido and the prototype of dog independently of whether Fido and my prototype of dog match in other respects. To put it more technically, properties are *independent cues* for categorization. Strictly speaking, the linearity of the similarity function is not required by prototype models.

Finally, prototype models of cognitive processes, for instance, prototype models of categorization, are typically *integrative* (Berretty et al. 1999). That is, it is assumed that our cognitive processes *combine* several cues to produce their outputs. For instance, to decide whether a target is a dog, we are assumed to take always into consideration *several* of its properties.

3.3.1 An example of Prototype-based categorization: The Hampton's model

There exist, in psychological literature, many different models of prototype-based categorization. These models usually specify:

- (i) how the similarity between a prototype and a target is computed—*the similarity measure* -
- (ii) how the decision to categorize the target is made - *the decision rule*.

Typically, nothing is said about whether the matching process between the representation of the target and the prototype is done serially (a property at a time) or in parallel (all properties at the same time).

A classical, well known, model of prototype based categorization has been proposed by Hampton (1995). This model consists of three elements: a prototypical representation of concepts, a similarity measure, and a decision rule. The prototypical representation of

concepts is similar to other models and has a frame-like form (table 3.1). The similarity measure is the following:

- $S(x, C) = f(w(x, i))$,

where $S(x, C)$ is the similarity between the target x and the prototype of the category C , f is a function that ranges over all the properties represented by the prototype, and $w(x, i)$ is the weight of the value (e.g., *red*) possessed by x for the i^{th} attribute (e.g., *color*). According to Hampton (1993, p. 74):

“The simplest, and most common assumption for the function f is a linear combination rule, such that the similarity is proportional to the sum of the attribute-value weights possessed by an instance.”

Thus,

- $$S(x, C) = \sum_{i=1}^n w(x, i)$$

Hampton’s decision rule for categorization is a simple deterministic rule (74):

- $S(x, C) > t \rightarrow x \in C$,

where t represent a threshold on the similarity scale. It is important to note that Hampton’s model assumes that the same process of similarity evaluation underlies both typicality judgments (how typical an object is of its category) and categorization judgments. Typicality ratings are supposed to be strictly related to similarity.

Thus, Hampton’s model of the categorization process involves a matching process between representations as well as a linear measure of the similarity between a prototype and other presentations. These two traits can be considered as the main trademarks of prototype-based models of cognitive processes.

3.3.2 Categorization with exemplars

Like prototype-based models of cognitive processes, exemplar-based models assume that cognitive processes involve the computation of the *similarity* between exemplars and other representations. For instance, categorization judgments are supposed to result from the computation of the similarity between exemplars and the representation of the target. Let us consider an example (from Machery, 2009): when we categorize Fido as a dog, one or several exemplars of dogs are retrieved from our long-term memory (together, maybe, with exemplars of other related categories, such as cats); this exemplar (or these exemplars) is (are) matched with our representation of Fido. Then, the similarity between these representations is computed and, finally, the decision that Fido is a dog results from the high degree of similarity between the retrieved exemplar(s) of dog(s) and the representation of Fido. Figure 3.4 summarizes these processes.

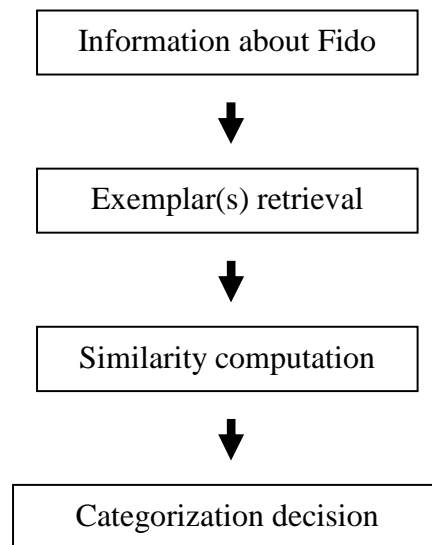


Figure 3.4: A Sequential Schema of Psychological Categorization in Exemplars

Another central property of exemplar-based models is that the similarity measure is usually supposed to be *non-linear*. In non-linear measures, how much a property that is shared by the target and by an exemplar increases the similarity between the target and

this exemplar depends on which other properties they share. For example: Suppose that the pet of my neighbors, Fido, has a property (say, *barking*) that is represented by one of the exemplars of dogs stored in my memory (say, the representation of my own dog, Lassie). How much the similarity between the representation of Fido and of Lassie is increased depends on whether Fido and Lassie share other properties, such as *chasing cats*. Thus, by contrast to linear measures, the degree of similarity in non-linear measures is supposed to be a function of the *configuration* of cues. To put it more technically, properties are *dependent cues* for categorization.

3.3.2.1 Exemplar categorization: Nosofsky's Model

In order to better explain the categorization process in exemplar theory, we briefly describe a well-known exemplar model: the Generalized Context Model of categorization developed by Nosofsky (1986, 1992). This model is an extension of Medin and Schaffer's (1978) Context Model presented above (section 3.2.3). The Generalized Context Model consists of an exemplar model of concepts, a similarity measure and a decision rule. According to this model, each exemplar is represented as a point in a multidimensional space in which each dimension represents a continuous property. Thus, an exemplar is represented by a specific value for each one of the dimensions that constitute the dimensional space.

Regarding the similarity measure: in the Generalized Context Model, each target is compared to all the exemplars that constitute a concept. For instance, a dog, Fido, must be compared to all the exemplars of dogs that constitute my concept of dog as well as to all exemplars of wolves that constitute my concept of wolf. The similarity between Fido and a given exemplar, for instance an exemplar of dog, is a function of the psychological distance between Fido and this exemplar. This psychological distance depends on the extent to which Fido and the exemplar match on each of the relevant dimensions for categorizing Fido. The more different Fido and the exemplar are on a given dimension, say k , the further apart they are on this dimension. Formally, for a given dimension, the distance between the target Fido and the exemplar is:

1. $|x_{tk} - x_{Ek}|$

where x_{tk} is the value of the target, Fido, on dimension k and x_{Ek} is the value of the exemplar on this dimension. Each psychological dimension is weighted. The weight of dimension k , w_k , indicates the attention paid to k . Greater values of this weight capture the idea that mismatch along dimension k increases the dissimilarity between the exemplar of a dog and Fido, decreasing, as consequence, the likelihood that Fido will be categorized as a dog. This parameter is assumed by Nosofsky to be context-dependent. Dimension weights sum to one: this captures the idea that decreasing the attention to one dimension entails increasing the attention to other dimensions. The psychological distance between Fido and the exemplar of a dog depends on whether the relevant dimensions are analyzable (see Ashby and Maddox 1990). Analyzable dimensions can be attended independently of one another. Size and weight are analyzable dimensions of objects. This means that it is possible to attend to the size of an object, independently of its weight. By contrast, non-analyzable dimensions cannot be attended independently of one another. For example, hue, brightness, and saturation of colors are non-analyzable dimensions. When dimensions are non-analyzable, the psychological distance is computed with a Euclidean metric:

$$d_{tE} = \sqrt{\sum_{k=1}^n w_k (x_{tk} - x_{Ek})^2}$$

2.

When the dimensions are analyzable, the psychological distance is computed with a city-block metric:

$$d_{tE} = \sum_{k=1}^n w_k |x_{tk} - x_{Ek}|$$

3.

More generally, the distance between the target and the exemplar for n dimensions is calculated as follows:

$$d_{tE} = c \left(\sum_{k=1}^n w_k (x_{tk} - x_{Ek})^r \right)^{\frac{1}{r}}$$

4.

where r depends on whether the dimensions are analyzable or not. c is a parameter that measures how much the overall psychological distance between a target and an exemplar affects their similarity. The similarity between t and E is an exponential function of the psychological distance between the target and the exemplar. It is calculated as follows:

$$5. S_{tE} = e^{-dtE}$$

Thus, the greater the psychological distance between the target, Fido, and the exemplar of dog, the smaller is their similarity. The overall similarity of the target, Fido, to the concept of dog, that is, to the set of exemplars of dogs, is the sum of its similarities to each exemplar of a dog. Formally,

$$6. S_{tC} = \sum_{E \in C} S_{tE}$$

If two concepts, say DOG and WOLF, have been retrieved from long-term memory, the probability of classifying Fido as a dog is a function of the overall similarity of Fido to the concept of dog divided by the sum of the overall similarities to the concepts of dog and of wolf. Formally,

$$7. P(t \in A) = S_{tA} / (S_{tA} + S_{tB})$$

where A and B are the two relevant concepts.

Concluding: Nosofsky's Generalized Context Model of categorization illustrates the core ideas of exemplar-based models. In this case the process of categorization involves matching the representation of targets with exemplars and calculating, in a non-linear way, their similarity.

3.4 Prototypes and Exemplars

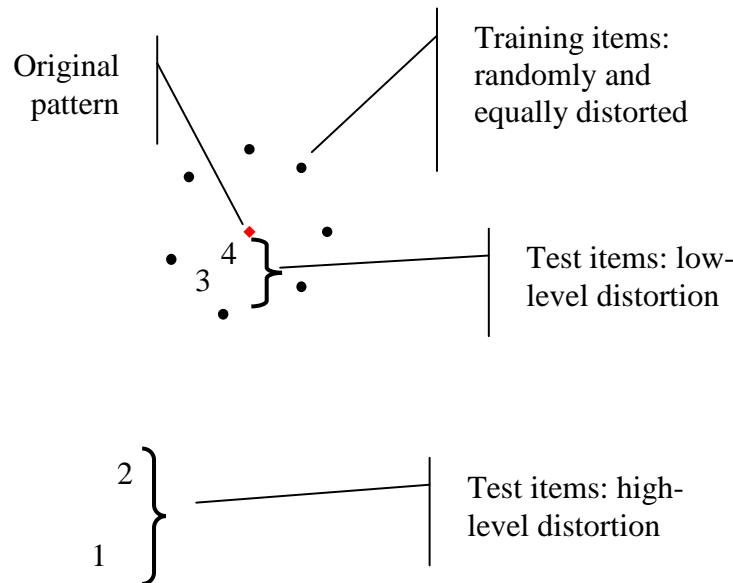
A considerable literature exists comparing prototype and exemplar theories (see e.g. Dopkins and Gleason, 1998; Lalumera 2009). The scope of the comparison has been, usually, to provide evidence in support or in contrast with one of the two theories, in order to enhance the proposed empirical models of human categorization and conceptualization and to establish which theory better explains the typicality phenomenon. In the following sections our goal is to put in evidence the main differences and similarities among these two approaches. In the first part we present

some results that favour one approach with respect to the other one. Then we compare the two approaches in order to underline their similarities and their differences. In the final part, we present a new perspective according to which we can have both prototype and exemplar based representations, and that we can use them in different situations. This choice goes in the direction of a Multi Process Theory, and contrasts the classical Unified View of Cognition (for major details on this issue see Lalumera 2009).

3.4.1. In favor of Prototypes: the Random Distortion Pattern Evidence

An element of evidence in favor of prototype theory derives from the study of Smith and Minda (2002) regarding the categorization prediction of random patterns of points. In the experiment setting proposed by Smith, different answers are expected according to prototype and exemplar theories. More precisely: exemplar and prototype theories make rather similar predictions about the categorization of high-level distortion patterns of points (1 and 2 in the figure below). However, their prediction differs for low-level distortion patterns (3 and 4 in the figure). Exemplar theories predict that the probability of classifying low-level distortion patterns of points should not increase with increasing typicality, or, equivalently, decreasing distortion. The prototype theories, instead, make the opposite prevision. The description of the experiment can be summarized in the following way: there is a starting phase, a training phase and a test phase. The starting phase consists of the creation of a category of patterns of points. In the training phase, training items are patterns of points that are obtained by distorting the original pattern at a similar degree. In the test phase, two different patterns of points are proposed to the subjects of the experiment. In particular, a low-level distortion patterns of points (items 3 and 4 in figure 3.5) and high-level distortion patterns of points (items 1 and 2 in the same figure) are showed to the subjects. Exemplar theories predict that the probability of classifying low level distortion patterns of points should not increase with increasing of typicality or, equivalently, decreasing distortion. Therefore, for the exemplar theories the probability of classifying 4 as a category member should be equal to the probability of classifying 3. The reason is that, for low-level distortion patterns of points, any change in typicality will increase the similarity with some exemplars of patterns of points, but decrease the similarity with others, leaving the overall similarity to the set of

exemplars unmodified. By contrast, prototype theories predict that the probability of classifying low-level distortion patterns of points should increase with increasing typicality, or, equivalently, decreasing distortion. That is, prototype theories predict that the probability of classifying 4 as a category member should be greater than the probability of classifying 3. Figure 3.5 illustrates this argument.



*Figure 3.5: Test and Training Patterns of points
(from Smith & Minda 2002)*

Using existing data sets about subjects' categorization profiles in the dot-distortion classification task, Smith and Minda (2002) shown that prototype models of categorization do better at describing the categorization profiles of normal subjects.

3.4.2. Evidence Against Prototypes

Despite the above mentioned results, in head-to-head competition, exemplar models are in most cases more successful than prototype models. There are, in fact, numerous empirical evidences demonstrating this. For example: a first empirical element that is coherent with the exemplar theory and not with prototypes is the “old-items advantage effect”. This effect consists in the fact that old items are usually more easily categorized

than new items that are equally typical (for a review, see Smith and Minda 1998). For example: it is easier for me to classify my old pet Fido as a dog than an unknown dog that is equally typical. This effect is not predicted by prototype theories of concepts. Prototype theorists assume that people abstract a prototype from the stimuli they are presented with during the learning phase, and categorize old as well as new stimuli by comparing them to the prototype. What matters for categorization is the typicality degree of the items, not whether they have already been seen or not. By contrast, the old-items advantage falls out from the exemplar paradigm.

A second type of empirical evidence in favour of exemplar theories is the following: it can happen that a less typical category member can be categorized more quickly and more accurately than a more typical category member. Furthermore, its category membership can be learned more quickly than the category membership of a more typical instance if it is similar to previously encountered exemplars of the category (e.g., Medin and Schaffer 1978). For example, it may be easier for me to categorize as a dog a three-legged dog than a four-legged one if my own pet dog lost a leg. Medin and Schaffer's (1978) found evidence that supports their prediction. Furthermore, if compared with prototype models, the exemplar models tend to be more conservative about discarding information. They store a major amount of information than the prototypes do. This availability of a wider amount of information facilitates predictions and exemplar models seems to be better than prototype models in predictions support (Machery 2009). Another important blow to the prototype theory derives from the study of linear separable categories (Medin and Schwanenflugel 1981). A category is linearly separable if and only if it is possible to determine whether items belong to this category by summing the evidence offered by each property of this item. For example, suppose that two categories are characterized by two dimensions. These categories are linearly discriminable if and only if one can determine the category membership of each item by summing its value along the x- and y-axes, that is, if a line can be drawn between the members of each category (Figure 3.6).

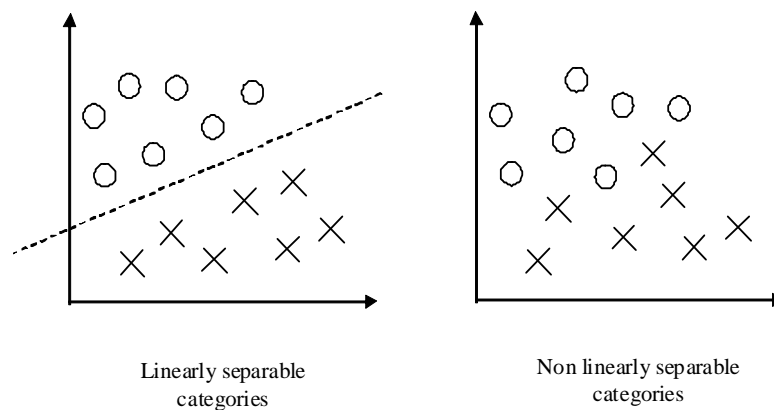


Fig. 3.6 Linear Separability

The study of Medin and Schwanenflugel (1981) demonstrated that it is not possible to claim that linearly discriminable categories are easier to learn. This conflicts with the assumption made by the prototype theory. According to this approach, in fact, people should find it difficult to form a concept of a non-linearly discriminable category (Medin and Schwanenflugel 1981; Murphy 2002). From an operational point of view, subjects should be faster at learning two categories, when such categories are linearly discriminable rather than non-linearly discriminable. Exemplar theories, instead, do not predict that subjects would be better at learning linearly discriminable categories than categories that are not linearly separable. In psychological literature this result has been taken as strong evidence for the exemplar models of concept learning. All these findings are clearly problematic for the prototype paradigm of concepts, while they are consistent with the exemplar view. In the next section a brief summary of the main evidences coming from the comparison of the two approaches is drawn.

3.4.3. Prototypes vs Exemplars in short

Prototype and exemplar approaches present significant differences. A brief summary of such differences is presented here. First of all, exemplar-based models assume that the same representations are involved in such different tasks as identification (e.g., “this is the Tower Bridge”) and categorization (Nosofsky 1986). This contrasts with the prototype models, which assume that different kinds of representation are involved in

these cognitive processes. Furthermore prototype models intend to capture only some central, and cognitively founded, aspects of the features of a concept, while the exemplars models represent *in toto* the particular knowledge of a certain entity. Another aspect of divergence consists in the treatment of the categorization process. As we have seen in the examples of sects. 3.4.1 and 3.4.2.1, in order to account for this reasoning task, both prototype-based and exemplar-based models assume that the *similarity* between prototypical/exemplars representations and target representations must be computed. The decision of whether the target belongs to some category depends on the result of this comparison. Despite this common mechanism, in the prototype view the computation of similarity is usually assumed to be *linear* (a property that is shared by the target and the prototype increases the similarity between the target and the prototype independently of whether other properties are shared by them) while, according to the exemplar view, it is assumed to be *non-linear* (a property that is shared by the target and the exemplar is considered relevant only if there are also other shared properties between the two representations). Another relevant difference among the two approaches regards the different assumptions made on our memory. According to exemplar theorists, we form memories of many encountered category members and we use by default these memories in cognitive tasks. On the contrary, according to prototype theorists, we store in our long-term memory only some parameters that characterize the categories we represent. This difference involves different memory costs: if compared to exemplars, prototypes are synthetic representations and occupy a minor space of memory. On the other hand, the process of creation of a prototype requires more time and cognitive effort, while the mere storage of knowledge about exemplars is more parsimonious and less consuming because no abstraction is needed. The table below summarizes the main traits of the two approaches.

	Prototype models	Exemplar Models
Memory Storage	The prototype of each category is a sort of “average” description of all the exemplars experienced.	Many exemplar encountered are stored along with the category to which it belongs
Memory Costs	Not expensive. Prototypes are “synthetic” representations.	Expensive: the information concerning whole particular exemplars is stored.
Cognitive Efforts	It is expensive to build the prototype. More time is requested.	It is parsimonious to use the exemplars knowledge.
Decision Rule for Categorization	Linear	Not linear
Inferential Prediction	Not so good because it does not keep in memory all the traits.	Better in support predictions based on partial information.
Effects in Categorization	Similarity degree based on typicality.	Old Items Advantage Effect.

Table 3.1 Prototype models vs Exemplar Models

3.4.4. Criticisms to the Exemplar Paradigm and Hybrid Approaches

Despite the success of exemplar theories during the '80s and the '90s of the last century, different results coming from empirical research has shown some weakness point of this approach. Smith and Minda (2000) are the authors of one of the most famous articles in this sense. The authors cast some doubts on the strength of the evidence for the exemplar approach to concepts, categorization, and concept learning. The criticisms have been focused on the fact that many experiments that support the exemplar paradigm of concepts against the prototype paradigm are based on the same category structure¹², which Smith and Minda, called the “5-4 category structure.” There are 2 categories. Category A consists of 5 elements, category B of 4 elements. Apart from

¹² A category structure is an abstract characterization of categories used in experiments. Four properties matter from this point of view: (i) how many categories are used in the experiment, (ii) for each category, how many members belong to it, (iii) how many properties or dimensions characterize the items used in the experiment, (iv) whether the members of the category possess or not each property.

these nine training stimuli, there are seven transfer stimuli. Four binary dimensions distinguish these 16 items. Each item is characterized by a value 1 or 0 along each of these four dimensions. Table 3.2 summarizes the 5-4 category structure (adapted from Smith and Minda 2000).

	Dimensions			
	D1	D2	D3	D4
Category A				
A1	1	1	1	0
A2	1	0	1	0
A3	1	0	1	1
A4	1	1	0	1
A5	0	1	1	1
Category B				
B1	1	1	0	0
B2	0	1	1	0
B3	0	0	0	1
B4	0	0	0	0
Transfer stimuli				
T1	1	0	0	1
T2	1	0	0	0
T3	1	1	1	1
T4	0	0	1	0
T5	0	1	0	1
T6	0	0	1	1
T7	0	1	0	0

*Table 3.2: The 5-4 Category Structure
(Adapted from Smith and Minda 2000)*

Let suppose that a prototype 1111 could be abstracted from category A, while category B would correspond to a prototype 0000. In this case, the category A has four members that share three features with the hypothesized prototype of A and one member that shares two features with this prototype. As emphasized by Smith and Minda (2000), category A has no “exceptional” member, that is a member “sharing more features in common with the opposing prototype,” but an “ambiguous” member, which shares “features equally with both prototypes.” Category B members share 2, 2, 3, and 4 properties with the hypothesized prototype of B. Thus, category B contains two ambiguous members, and no exceptional member. So the average typicality of the members of A and of the members of B is the same. Additionally, the authors noted that natural world categories seem to be more differentiated and are not restricted to a few members. Thus, results found with undifferentiated, small categories may say little about how we learn concepts and categorize in real-world situations. For these two reasons, the validity of many experiments assumed to support the exemplar paradigm has been considered at least as controversial.

Moreover, Smith and Minda argue that “subjects’ performances in experiments that use the 5-4 category structure do not support the exemplar paradigm as clearly as exemplar theorists would have it. Smith and Minda examined 30 data sets from eight articles that were obtained with the 5-4 category structure. They confirm that standard prototype models of categorization do not fit very well the data sets while, by contrast, the Context Generalized Model proposed by Nosofsky (presented in the section 3.4.2.1) successfully fits the data sets. However they show that prototype models can be extended in various ways to fit the categorization of dataset.

Smith and Minda’s (2000) critique has been very influential against the claim that the exemplar paradigm is supported by an overwhelming body of evidence. Moreover, this criticism finds confirmations in many others experimental results, showing that people can use either exemplars or prototypes to solve categorization tasks. This goes in the direction of a hybrid view of concept representation and reasoning. In the next section we will do a brief overview on this approach.

3.5 Towards Hybrid Approaches to Concepts Representation and Reasoning

Hybrid theories of concepts were first proposed for several reasons at the end of the 1970s and at the beginning of the 1980s. They were sometimes motivated by the desire to save the view that concepts consist of definitions (the “classical” view). The argument was the following one: if concepts consist of two parts, a definition and an additional part, experimental findings that cannot be explained by assuming that concepts consist of definitions might be explained by hypothesizing that subjects behavior relied on this additional knowledge. Furthermore, as reported in Lalumera (2005) and Machery (2009), hybrid theories of concepts were also motivated by the shortcomings of the new theories of concepts proposed in the 1970s, such as the prototype theories.

Nowadays many psychologists agree that, for many categories, we have both a definition (e.g. a classical representation based on sets of necessary and sufficient conditions) and another type of representation: for instance, a set of exemplars. Some of them, for example Ashby and colleagues (1998), propose that a definition of a category and, say, a prototype for this category form two concepts, instead of being two parts of a single concept (Ashby et al. 1998; Ashby and Ell 2001). This position is completely different from that supported by other proposals and experimental evidences. Large part of researchers, in fact, argue that the different types of representation should be thought as *parts* of the same concept, thus endorsing a hybrid theory of concepts. An example going in this direction has been proposed by Osherson and Smith (1981). They propose that concepts are made of two parts, a core and an identification procedure: the core of a concept consists of a definition, while the identification procedure consists of a prototype (1981, 57):

“[W]e can distinguish between a concept’s *core* and its *identification procedure*; the core is concerned with those aspects of a concept that explicate its relation to other concepts, and to thoughts, while the identification procedure specifies the kind of information used to make rapid decisions about membership (...). We can illustrate with the concept *woman*. Its core might contain information about the presence of a reproductive system, while its identification procedure might

contain information about body shape, hair length, and voice pitch. Given this distinction, it is possible that some traditional theory of concepts correctly characterize the core, whereas prototype theory characterizes an important identification procedure.”

Osherson and Smith proposed also that some cognitive competences involve only one of these two parts. Particularly, concept composition is assumed to involve exclusively the core while other competences, such as categorization, involve both the definition and the prototype: Categorization is underwritten by two distinct processes—a prototype-based process and a definition-based process. According to the authors we categorize objects by means of the prototype when we need to identify quickly their category membership. This categorization is reliable, but defeasible. We categorize objects by means of the definition when we need to be sure of their category membership. The idea of a hybrid representation of concepts is presented also in the Nosofsky's and colleagues (1994) model called “RULEX” - which stands for “rules plus exemplars”. According to RULEX a concept consists of two parts, a rule and a set of exemplars. A rule is equivalent to a definition. An exemplar is a representation of a category member. During the process of categorization, these two parts are used as follows. When people have to categorize an object in one of two categories, A and B, they first apply a rule that discriminates most members of A from most members of B. Then, they check out whether this object is not one of the instances that are known to be exceptions to the rule (Figure 3.7).

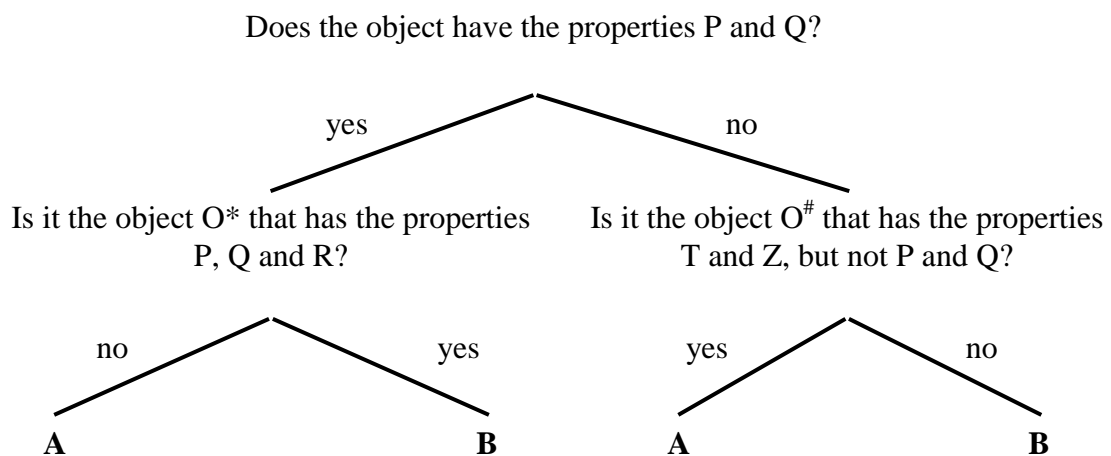


Figure 3.7: the Categorization Procedure of RULEX
(from Machery, 2009)

In other words, this model assumes that a single categorization process uses both parts of our concepts (rule and exemplars), so that the parts of a given concept do not produce inconsistent categorization judgments. Therefore, differently from what has been proposed by Osherson and Smith, they propose a single categorization process on a dual representation.

Another important psychological study supporting the idea of multi-process theory was done by Malt (1989). Her study had the aim of investigating if people categorize and learn categories according to the exemplar approaches or to the prototype based models. Her work, done using behavioral measures such as categorization probability and reaction time, demonstrates that not all subjects retrieve exemplars to categorize. Some use exemplars; a few rely on prototypes, and others appeal to exemplars and to prototypes. A protocol analysis of subjects' description of their categorization strategy confirms this interpretation¹³. Malt writes (1989, 546-547):

“Three said they used only general features of the category in classifying the new exemplars. Nine said they used only similarity to old exemplars, and eight said that they used a mixture of category features and similarity to old exemplars. If reports accurately reflect the strategies used, then the data are composed of responses involving several different decision processes”

This suggests that people can use either prototypes or exemplars to solve Malt's categorization task. These findings are consistent with other well known studies such as Smith et al.'s (1997) and Smith and Minda's (1998) experiments carried out with artificial stimuli. Smith et al. (1997), in fact, found that the performances of half of the subjects of their experiments were best fitted by a prototype model, while the performances of the other half were best fitted by an exemplar model. This suggests that people can learn at least two different types of concepts—prototypes and exemplars—and that they can follow at least two strategies of categorization. Smith and Minda

¹³ A protocol analysis consist in the recording of what the subjects of an experiment say after the experiment about the way in which they performed the assigned tasks.

(1998) replicated these findings. Additionally, they found that during the learning phase, subjects' performances are best fitted by different models, suggesting that, when they learn to categorize artificial stimuli, subjects can switch from a strategy involving prototypes to a strategy involving exemplars. They also found that the learning path is influenced by the properties of the categories subjects are presented with. For example, they show that categories with few, dissimilar members promoted the use of exemplar-based categorization strategies. Thus, psychological evidence suggests that we have at least two different mechanisms for categorizing. These mechanisms rely on different types of knowledge: prototypes and exemplars.

3.6. Prototypes and Exemplar Theories in Machine Learning: a brief overview

The theories of human categorization based on prototypes or exemplars have been considered not only in the field of psychology and philosophy but also in such disciplines as machine learning and automatic classification¹⁴ (Witten, Frank, 2005). Machine learning is the field of artificial intelligence that is concerned with the design of programs that can learn from experience and improve their performance (Russel, Norvig 2002). In the subfield of *supervised learning*, the problem of *classification* concerns the construction of classifier systems that, after a suitable training, can assign instances or objects assumed in input to the proper class among a set of possible classes. Categorization process in a classification system is carried out in two steps: the first one consists of a learning, or training, phase, and the second one is the categorization phase in a strict sense.

In short, the process can be summarized as follows: in the first phase a set of labeled data, called the *training set*, is considered, in order to learn the function which maps data to classes. In the second phase, the classification function learned during the

¹⁴ In the machine learning literature, the terms classification and categorization are usually used as synonyms. However, the two terms can refer to different reasoning processes. For example, in the field of DLs, classification is a (deductive) reasoning process in which superclass/subclass (i.e., ISA) relations are inferred from implicit information encoded in a KB. More in general, in cognitive science, categorization is usually an inferential process through which a specific entity is assigned as an instance to a certain class. In non-monotonic categorization class assignment is a non deductive inferential process, based on typicality. In this section, respecting the terminology of machine learning community, we will use the terms as synonyms whose intended meaning is that one represented by the term "categorization" in psychological literature.

training phase is used to classify new data, for which the appropriate classification is still unknown.

In the last years many classifier systems and algorithms have been developed following both the psychological theories of prototypes and exemplars. There are models developed following the assumption of prototype approaches (e.g. the PRT model proposed by Reed (1972), and exemplar based models of categorization, such as ALCOVE (Attention Learning COVERing map) developed by Kruske (1992) (see Leon and Galea (2007) for major details).

Particularly, in the machine learning area known as Instance based learning, it is possible to individuate different types of classifiers. For example, there is the Nearest Prototype Classifier (NPC), based on prototypes, and the Nearest Neighbour Classifier (NNC) based on exemplars. Before explaining in major detail the main characteristics of these two classifiers, we briefly characterize the Instance Based Learning methodology in general. Instance based learning is basically founded on four elements: the definition of similarity between observations, the representation of classes, the learning algorithm and the classification algorithm (see, again, Gagliardi 2011 for further details). In the following we briefly summarize them:

- ⤴ *Similarity*: is formalized through a definition of a metric in the space of all possible observations, by which it is possible to quantify the distance between objects and thus also between the new instances and the ones stored as representative of the classes.
- ⤴ *Classes representation* consists of a set of couples composed by an instance and the relative class. It is created by the learning algorithm and is used by the classification algorithm.
- ⤴ The *learning algorithm* uses the training set to construct a set of representative instances.
- ⤴ *Classification algorithm* assigns a class to each new observed instance on the base of a criterion of greater similarity to the representative instances.

In instance-based learning systems the knowledge extracted from the training set consists of the storage of directly observed or abstract instances belonging to the set of all possible observations. These instances which are saved in memory form the

categories representation. Classification is performed comparing a new instance, for which the class is unknown, with the labeled instances in memory. According to Gagliardi (2008 pag. 2)

“the instances based representation, unlike other widely used representations in machine learning (e.g. rules, decision trees, etc.), is the only coherent with both the prototypes and exemplars theories and hence, in accordance with the “typicality view” on categorization, is the one to be used in order to develop classifier systems characterized by cognitive plausibility. Other representations (e.g. Classification rules) can be only related to the classical theory of categorization, and therefore, they lack a truly cognitive plausibility”.

3.6.1. Instance-based Classifier Systems

As mentioned above, there are different instance based classifier systems that are based on different categorization theories. For example: the *Nearest Prototype Classifier* (NPC) is one of the simplest classification systems and it is based on the assumptions of prototype theories (Kuncheva, Bezdek, 1998) (Bezdek et al., 1998). In NPC the learning algorithm constructs a single representative instance for every class. Each of these instances is called the *prototype* of the relative class, and it is calculated as the barycentre of the instances belonging to that class. The *NPC* assigns any new observed instance to the class whose prototype is the nearest.

A different approach to automatic categorization is given by the *Nearest Neighbour Classifier* (NNC), which is exemplars-based. The *Nearest Neighbour Classifier* is based on the plain comparison between the new instances and the training set. The learning phase is *de facto* absent because the set of the representative instances coincides with the entire training set. For this reason this classifier is called *memory based*. The *NNC* assigns to any new instance the class of the closest representative instance.

Recently, in the instance based learning research area, different new proposals have been presented. Some of them are modified version of NPC – for example this is the case of the NMPC (Nearest Multiple Prototype Classifier) in which the numbers of prototypes for each class is increased. Other proposals are modified versions of NNC.

These algorithms are variations of the prototype and of the exemplar based approaches respectively.

Another interesting and promising solution goes in the direction of creating hybrid classifiers, in order to consider prototype and exemplar based theories not as two conflicting alternatives, but as two limit cases of the Instance Based Learning technique. Hybrid algorithms allow this “unified view” of typicality because they use a mixed representation of classes, composed by both prototypes and exemplars. And, moreover, they usually have the interesting property of exhibiting as special cases exactly the behavior of NPC and NNC, being able to vary in all possible intermediate cases. Therefore, they seem to be able to satisfy the need of a more inclusive approach to categorization. In our opinion, as we shall show in the next chapter, such a hybrid approach could be fruitful also to face the problem of representing and reasoning on typicality within the field of formal ontologies.

Chapter 4.

A Hybrid Approach to Concept Representation and Reasoning

As anticipated in the chapter 2, the representation of prototypical information and the mechanisms of reasoning on “typicality” have been widely debated in the field of Knowledge Representation. Historically these needs have been contrasted with the requirement of compositionality and the need to perform deductive reasoning. These two groups of requirements have often been viewed as not conciliable, and this dualism determined, in various domains, the realization of “partial systems”, limited both for the type of information expressed and for their reasoning capabilities.

Our proposal is to provide a general architecture able to take into account, in an integrated perspective, these two elements of the common sense knowledge, with the aim of overcome the dichotomy typicality vs compositionality within the ontology based systems. In this sense, we propose a hybrid approach based on a two-layer structure for knowledge representation and reasoning. This architecture has a psychological background (see section 4.2 for major details) based on three different approaches. Namely: (i) the dual theories of reasoning and rationality, stating that human reasoning is the result of the interaction of two different types of cognitive systems (ii) the pseudo-fodorian idea of taking separated the different knowledge components based on compositional and typical information, (iii) the prototype and exemplar theories of concept.

Following this ideas, we propose a model combining a module for classical ontological representation and reasoning with a second one implementing reasoning on prototypical information and information about exceptions. Within the semantic web languages, this integration is now made easier because the linked open data better support (e.g. via

URI, sameAs and other OWL linking statements) the connection of multiple knowledge modules providing different types and/or levels of information for the same concept.

4.1 General Description

The proposed architecture consists of two main interconnected elements, representing the modules of the dual structure, and corresponding to the two types of cognitive system hypothesized by the dual process theory. Such knowledge modules are:

- a compositional part, in which concepts are represented in an Ontology Web Language, and described in terms of necessary and/or sufficient conditions. Such component provides well known types of deductive reasoning such as classical classification, consistency checking and deductive categorization. See chapter 2 for further details.

- A “typical” part, which can represent both prototypical or exemplar-based knowledge concerning a certain concept (the different ways in which prototypical and exemplar information are represented will be discussed in the following pages), and in which some forms of non deductive reasoning can be added to the classical inferences performed by the compositional knowledge bases.

In the general architecture of the system, a connection between these two knowledge modules is provided. This connection represents, in our proposal, a kind of “integration with some limitation” of the two modules. In our view, in fact, the two representations must be kept independent even if interconnected. This separation is motivated by the fact that each representation is associated to a specific type of reasoning. Compositional representation and deductive reasoning must be kept separated from typical information and approximate reasoning. One of the reasons of this separation is to keep safe the results obtained by deductive reasoning in the DL knowledge-base from the results that can be provided only with the second “typical” part. This solution presents a relevant advantage: it does not cause inconsistencies in the case in which the different forms of reasoning would draw to different conclusions¹⁵ This form of “separation within the

¹⁵ It could be possible to obtain different results between monotonic and non monotonic reasoning processes

integration” is possible thanks to the realization of a cascade model, in which one of the components of the system is assumed to have a priority on the other. We can individuate the following general procedural steps in the architecture behavior ¹⁶:

- 1) Perform deductive reasoning on the DL knowledge-module (e.g. classification, consistency checking, monotonic categorization etc.)
- 2) Save the obtained results
- 3) Run specific task tests on the DL system (e.g. queries on the KB)
- 4) If the obtained results are “satisficing” for your purpose then stop, else execute the same tasks on the “typical” knowledge-base and take that results.

According with the above mentioned assumption, the link between the two layers of the architecture is assumed to be unidirectional. In the case mentioned before, in fact, it is only possible to proceed from the compositional (DL) part to the typical part and not *vice versa*. In other words: the results obtained with the task tests executed at the step 3 can only be enriched and /or be substituted, in case they are considered as not relevant, with the results coming from the typical module. But is not possible to operate in the opposite direction. This condition is necessary in order to avoid the overlapping of the two representations and of the relative reasoning processes. However, as we will see in the following pages, in some tasks (for example, tasks in which the results provided by the typical module are heuristically more relevant and “smarter” than those obtained by classical reasoning) , in order to obtain certain types of results closer to the human way of categorizing the world and retrieve information, it is possible to assume that this procedure can be modified assuming that the connection among the knowledge layers is still unidirectional, but in the opposite sense: the starting point is represented by the results coming from the typical part, and the information enrichment process proceed from the typical part to the DL one. Some examples regarding these situations are presented in the section 4.3. The image in figure 4.1 graphically represents the “canonical” direction of the interconnection between the two layers (from compositional to the typical part).

¹⁶ The task to execute in the third step can vary according to the specific application and purpose for which the system is used. In our approach, as will be explained later, we consider applications to information retrieval and information discovery tasks.

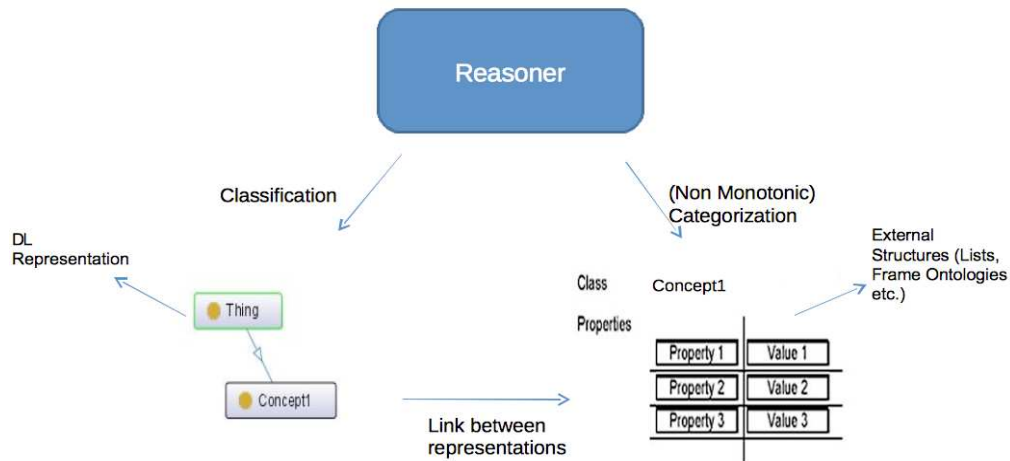


Figure 4.1. General canonical model of the architecture

4.1.1 Compositional Module

The compositional knowledge module of the architecture is supposed to represent, when possible, concepts in terms of necessary and/or sufficient conditions. In the case of an ontology based system, the description of concepts can be expressed using a standard Description Logics formalism, and can be represented according to the classical elements of an OWL ontology (e.g. classes, properties, instances etc., see chapter 2). For example the concept BACHELOR (not married person) can be easily formalized in the following way in a DL knowledge base:

- (i) $BACHELOR \subseteq PERSON$
- (ii) $BACHELOR \equiv MaleAdultPERSON \wedge \neg Married$
- (iii) $BACHELOR(Giordano, Bruno, \dots)$ ¹⁷

This information can be easily expressed in an OWL ontology, representing the concept PERSONS with the above mentioned properties (e.g. $\neg Married$) and with the indication of class membership (e.g. in the example Giordano and Bruno are instances of the class BACHELOR). This module represents, in our architecture, the “first choice”, the first

¹⁷ The expressions in (i) and (ii) belong to the TBOX (terminological box) in a typical DL system. (iii) belongs to the Abox (assertional box) of the knowledge-base.

element to be processed, and on which classical DL based reasoning processes must be performed (step 1 of the above described procedure). Furthermore, this module is also the one on which, in a next phase (step 3) of the procedure, specific tests on the knowledge base can be performed, such as query tests finalized to information discovery and retrieval. In the case in which the results obtained at the step 3 are satisficing the system stops; otherwise, the second module based on “typicality” is activated.

4.1.2 Typical Knowledge Module – Prototypes and Exemplars

According to our proposal, the typical knowledge module can represent typical information using both prototype and “exemplars based” representations. Prototypes describing concepts according to typicality traits should be implemented as data structures that are external to the DL knowledge base. Such structures could be lists of (possibly weighted) attribute/value pairs that are linked to the corresponding concept in the DL module. Some attributes of the list correspond to attributes of the DL concept, for which the value is further specified. Other attributes of the prototype could be absent from the corresponding DL concept. The exemplar based representation, instead, are assumed to be internal to the DL module, even if further levels and or “pieces” of related information can be stored in external data structures. They represent the specific traits of a certain entity. The representation of the exemplar Fido, belonging to the concept DOG, for example, could contain such peripheral characteristics as the information that Fido has got distemper. The prototypical representation of DOG, on the contrary, will describe the concept DOG according to a subset of cognitively central and relevant traits associated to dogs: e.g. they woof, they wag tail, and so on.

In our view, prototype based representation of concepts can be described according to the classical “format” of a frame (Minsky 1975) . A frame can be considered as a cognitive founded model of a specific concept. It can represent a single conceptual class or can be related to other frame representations, forming, in this case, the so called frame system. Usually a single frame is composed by three main elements: slots, values and facets. The slots represent the attribute assigned to a concept. They correspond to the “properties” (named binary relations between concepts) of a DL representation.

Facets constrain the values taken on by slots, such as, for example, the minimum or maximum value of a slot. The values specify a punctual information for a specific attribute.

On the other hand, exemplar based representations express a specific information within a wider and more general DL knowledge-Base. According to our hypothesis, it is not possible that exemplar information is in contrast with that one more general provided in the compositional module. The presence of contradiction between these two levels represent, in our perspective, a symptom of a wrong modeling. Let us consider a typical example from medicine (Motik et. al 2006): suppose that the knowledge to be modeled is the following “the people have the heart on the left, but some people (called dextrocardiacs) have it on the right”. In their paper Motik et al. state that such a domain cannot be modeled in a classical compositional OWL knowledge base because the axioms: $\text{Human} \sqsubseteq \text{HeartOnLeft}$, $\text{Dextrocardiac} \sqsubseteq \text{Human}$, and $\text{Dextrocardiac} \sqsubseteq \neg\text{HeartOnLeft}$ make the concept Dextrocardiac unsatisfiable and produce a contradiction. In our opinion, it would be an error to model the domain in these terms¹⁸. The correct way, that allows to account for the “exception” represented by the concept dextrocardiac , would be the following one: the class Human is represented in terms of necessary and sufficient conditions in the DL module, HeartOnLeft is represented as a prototypical properties of human beings (and, therefore, it is represented in the typical part of the concept Human), while the “atypical” situation of being a dextrocardiac can be modeled both linking this state to a specific exemplar within the DL knowledge base (representing the state of being dextrocardiac as a specific property of an exemplar within the knowledge base) or creating a class Dextrocardiac defined as: $\text{Dextrocardiac} \sqsubseteq \text{Human}$ and with HeartOnRight as a necessary and sufficient condition for a being dextrocardiac . This example allows us also to show the way in which both prototypical and exemplar representations interact among them in the general presented framework.

4.2 Cognitive Background

¹⁸ An important remark: Motik et al consider this kind of KB in ontology web language profiles with a low expressivity while we move at the expressivity level of OWL Full profile.

The empirical results coming from cognitive psychology show that most common-sense concepts cannot be characterised in terms of necessary/sufficient conditions. Classical, monotonic DLs seem to capture the compositional aspects of conceptual knowledge, but are inadequate to represent prototypical knowledge. However a “non classical” alternative able to represent concepts in prototypical terms does not still emerge.

Some recent trends of psychological research favour the hypothesis that reasoning is not an unitary cognitive phenomenon. At the same time, empirical data on concepts seem to suggest that prototypical effects could stem from different representation mechanisms. In this spirit, we individuate some point of reference from cognitive sciences that, in our opinion, could be useful for the development of artificial representation systems and seems to go in the direction prospected with the proposed architecture. Namely: (i) the distinction between two different types of reasoning processes, which has been developed within the context of the so-called “dual process” accounts of reasoning (sect. 4.2.1 below); (ii) the proposal to keep prototypical effects separate from compositional representation of concepts (sect. 4.2.2); and (iii) the possibility to develop hybrid, prototype and exemplar-based representations of concepts (sect. 4.2.3).

4.2.1 Dual Process Approach

Cognitive research about concepts seems to suggest that concept representation does not constitute an unitary phenomenon from the cognitive point of view. In this perspective, a possible solution should be inspired by the experimental results of empirical psychology, in particular by the so-called dual process theories of reasoning and rationality (Stanovich and West 2000, Evan and Frankish 2008). In such theories, the existence of two different types of cognitive systems is assumed. The systems of the first type (type 1) are phylogenetically older, unconscious, automatic, associative, parallel and fast. The systems of the type 2 are more recent, conscious, sequential and slow, and are based on explicit rule following. In our opinion, there are good prima facie reasons to believe that, in human subjects, classification, a monotonic form of reasoning which is defined on semantic networks, and which is typical of DL systems, is a task of the type 2 (it is a difficult, slow, sequential task). On the contrary, exceptions play an important role in processes such as categorization and inheritance, which are

more likely to be tasks of the type 1: they are fast, automatic, usually do not require particular conscious effort, and so on.

Therefore, a reasonable hypothesis is that a concept representation system should include different “modules”: a monotonic module of type 2, involved in classification and in similar “difficult” tasks, and a non-monotonic module involved in the management of exceptions. This last module should be a "weak" non monotonic system, able to perform only some simple forms of non monotonic inferences (mainly related to categorization and to exceptions inheritance). This solution goes in the direction of a “dual” representation of concepts within the ontologies, and the realization of hybrid reasoning systems (monotonic and non monotonic) on semantic network knowledge bases.

4.2.2 A “Pseudo-Fodorian” Proposal

Fodorian theory also represents an important point of reference for our proposal. According to Fodor, concepts cannot be prototypical representations, since concepts must be compositional, and prototypes do not compose. On the other hand, in virtue of the criticisms to “classical” Aristotelian theory (stating that concepts can be described in terms of necessary and sufficient conditions), concepts cannot be definitions. Therefore, Fodor argues that (most) concepts are atoms, i.e., are symbols with no internal structure. Their content is determined by their relation to the world, and not by their internal structure and/or by their relations with other concepts (Fodor 1987, 1998). Of course, Fodor acknowledges the existence of prototypical effects. However, he claims that prototypical representations are not part of concepts. Prototypical representations allow to individuate the reference of concepts, but they must not be identified with concepts. Consider for example the concept DOG. Of course, in our minds there is some prototypical representation associated to DOG (e.g., dogs usually have fur, they typically bark, and so on). But this representation does not coincide with the concept DOG: DOG is an atomic, unstructured symbol.

We borrow from Fodor the suggestion that compositional representations and prototypical effects are demanded to different components of the representational architecture. We assume that there is a compositional component of representations,

which admits no exceptions and exhibits no prototypical effects, and which can be represented, for example, in the terms of some classical DL knowledge base. In addition, a prototypical representation of categories is responsible for such processes as categorisation, but it does not affect the inferential behaviour of the compositional component.

It must be noted that our present proposal is not entirely “Fodorian”, at least in the following three senses:

i. We leave aside the problem of the nature of semantic content of conceptual representations. Fodor endorses a causal, informational theory of meaning, according to which the content of concepts is constituted by some nomic mind-world relation. We are in no way committed to such an account of semantic content. (In any case, the philosophical problem of the nature of the intentional content of representations is largely irrelevant to our present purposes).

ii. Fodor claims that concepts are compositional, and that prototypical representations, in being not compositional, cannot be concepts. We do not take position on which part of the system we propose must be considered as truly “conceptual”. Rather, in our opinion the notion of concept is spurious from the cognitive point of view. Both the compositional and the prototypical components contribute to the “conceptual behaviour” of the system (i.e., they have some role in those abilities that we usually describe in terms of possession of concepts).

iii According to Fodor, the majority of concepts are atomic. In particular, he claims that almost all concepts that correspond to lexical entries have no structure. We maintain that many lexical concepts, even though not definable in the terms classical theory, should exhibit some form of structure, and that such structure can be represented, for example, by means of a DL taxonomy.

4.2.3 Prototypes and Exemplars

As anticipated in the chapter 3, within the field of psychology, different positions and theories on the nature of concepts are available. They are generally grouped into three main classes, namely prototype views, exemplar views and theory-theories (see e.g. Murphy 2002, Machery 2009). All of them succeed in accounting for (some aspects of) the prototypical effects in conceptualisation. According to the prototype view,

knowledge about categories is stored in terms of prototypes, i.e. in terms of some representation of the “best” instances of the category. For example, the concept CAT should coincide with a representation of a prototypical cat. In the simpler versions of this approach, prototypes are represented as (possibly weighted) lists of features.

According to the exemplar view, a given category is mentally represented as a set of representations of specific exemplars explicitly stored within memory: the mental representation of the concept CAT is the set of the representations of (some of) the cats we have encountered during our lifetime.

Theory-theories approaches adopt a holistic attitude towards concepts. According to some versions of the theory-theories, concepts are analogous to theoretical terms in a scientific theory. For example, the concept CAT is identified by the role it plays in our mental theory of zoology. In other versions of the approach, concepts themselves are identified with micro-theories of some sort. For example, the concept CAT should be identified with a mentally represented micro-theory about cats.

These approaches turn out to be not mutually exclusive. They seem to succeed in explaining different classes of cognitive phenomena, and many researchers hold that all of them are needed in order to explain psychological data. In this perspective, we propose integrating some of them in computational representations of concepts. More precisely, we propose combining prototypical and exemplar based representations in order to account for category representation as well as the prototypical effects (for a similar, hybrid prototypical and exemplar based proposal developed in the field of machine learning, see Gagliardi 2008). We do not take into consideration the theory-theories approach, since it is in some sense more vaguely defined when compared to both prototypes and exemplar based approaches. As a consequence, at present its computational treatment seems to be more problematic.

4.3 Adaptation of the proposed cascade procedure

At a first view, a difference between our proposal and one of the above mentioned psychological theory (namely: the dual process theory, section 4.2.1) can be individuated. In the procedure described in sect. 4.1, the various steps seem do not

completely respect the assumption made by the psychological theory. In fact, in our proposal, the knowledge module associated to the non monotonic, typicality based categorization is only assumed to be used in a second phase of the process (Step 4). This approach has been preferred in order to use the typical knowledge only as an extension of the compositional one. It is, in a certain sense, a more conservative approach because it minimizes the risk of errors since it considers as strict propriety the trustful and sure results coming from the DL part and from classical reasoning. However, according to the "dual theory", the typical component is assumed to be the "faster" and automatic module of the cognitive system. In our opinion, this situation can be, in certain cases, plausibly hypothesized within the proposed architecture. Let consider, for example, the following situation: the individual Anna is an usual customer of a cinema. According to the management board of the cinema she has to be assigned to a predefined class of customers (a cluster) in order to plan the execution of targeted promotional activities when new movies arrive.¹⁹ Let suppose that Anna, and all the clients of the cinema, are described (in the DL KB) with properties registering the previously watched movies. And let even suppose, for sake of simplicity, that Anna watched in that cinema only two movies regarding superheroes. She is therefore described as: `haswatchedSupermanMovie`, `haswatchedSpidermanMovie`. Following the dual process theory then a non monotonic categorization process must be performed. Let assume that, in this case, the instance/exemplar Anna is assigned to the class "LoversOfSuperHeroesMovies" that we suppose to be represented within the typical component of the system²⁰. Of course this assignment is based on a non monotonic and also defeasible reasoning (e.g. Anna could dislike that genre and have seen that movies only because, in that situations, she was with her children that likes the super heroes). However, according to the limited amount of data available, the drawn conclusion

¹⁹ Targeted activities have a major percentage of success and minor costs because they are specifically performed on targets (e.g. groups of persons) which can be potentially interested to the promoted activity and not to all the possible audience. For example: in a book store if a customer is assigned to the class of "Lovers of Science Fiction genre" he can be contacted through a target activity when a new book of the genre is available. Of course this contact has a major probability of success (the success, in this case, can be measured by the numbers of book sold to the members of the target class) if compared with the probability of success of the same contact presented to another cluster of customers/readers (e.g. let suppose the readers belonging to the class "Lovers of the romance level genre").

²⁰ This typical class can be characterized by typical properties representing, for example, the fact that the instances belonging to that class usually watch movies whose based on super heroes stories.

seems to be plausible. What follows from this situation is that if we perform a concept retrieval task based on query such as: Does Anna likes super heroes movies? (this query corresponds to instance checking task that we will further describe in section 4.8) then an answer can be only provided by the typical component (in the case of the above mentioned example the answer would be affirmative). It is however, possible to hypothesize a successive check even on the classical knowledge base in order to control if the result obtained from the typical component is “confirmed” (in the case in which the exemplar is categorized in the same class even in the DL²¹) or not. In this last case the unique possibility of keeping trace of this (uncertain but plausible) information is demanded exclusively to the typical component. Therefore, in these situations, we claim that it is more plausible to start from the prototype based knowledge and not from the classical one. In cases such as that one in the example, the procedure describing the system behaviour would be, therefore, different and runs as follow:

- Perform non deductive reasoning on the typical knowledge-module (e.g. non monotonic categorization)
- Save the results obtained from the typical part
- For information retrieval task such as instance checking (see paragraph 4.8) consider as priority the results obtained by the non monotonic categorization.
- If the obtained results are “satisficing” then stop, else perform deductive reasoning on the DL component (e.g. classification, consistency checking, deductive categorization) and execute the same tasks on the “DL” knowledge-base.
- If the results are the same stop. If the results are different (e.g. the concept X is categorized in different manners in the two knowledge bases) then take the result coming from the prototypes if you want to adopt a “risky” strategy offering uncertain but plausible (and potentially smart) answers. Otherwise consider the results coming from the DL.

²¹ Of course we hypothesize that the same class prototypically represented to which the instance has been assigned is even represented within the DL knowledge base with a classical description.

Summarizing: in the presented procedure in a first time non monotonic categorization of instances in the typical component is performed. Then the obtained results are “saved” and for certain tests such as, for example, instance checking (whose objective is to check whether an instance belongs to a certain class), the system retrieve, as first choice, the results obtained by the non monotonic categorization. Only at a second step it accesses, if needed, to the results obtained by deductive reasoning processes.

Therefore in cases like these, in which both approximate reasoning and typical knowledge level are taken in consideration, the procedural process completely follow the suggestions coming from the psychological theory. It is important to note that this procedure can drive to errors and can be maybe suggested for technology (such as, for example, search technologies) in which is not crucial to have, at the first attempt, the correct answer; while, conversely, the access to “smart answers” obtained thanks to non monotonic reasoning - even if not valid from a logical point of view - could really improve the system performance in terms of user experience.

4.4 Implementation

In the field of web ontology languages, the development of the architecture sketched above appear nowadays, technologically easier to implement. Within the Semantic Web research community, in fact, the Linked Data perspective is assuming a prominent position (Bizer et al. 2009). According to this view, in recent years, one of the main goals of the Semantic Web community is the integration of different data representations (often stored in different data sources) within a unique, semantically linked, representational framework. The main technical result coming from this integration is represented by the possibility of enlarging the answer-space of a query through the realization of “semantic bridges” between different pieces of data (and, often, data sources). Such integration is made possible through constructs provided by Semantic Web languages, such as OWL, or schemas such as SKOS²² etc. According to Bizer et al (2009, p. 2) “Linked Data provides a publishing paradigm in which not only documents, but also data, can be a first class citizen of the Web, thereby enabling the extension of the Web with a global data space based on open standards - the Web of

²² <http://www.w3.org/2004/02/skos/>

Data”. Publishing a data set as Linked Data on the Web involves the following three basic steps (T.B.Lee 2006):

- Assign URIs to the entities described by the data set and provide for dereferencing these URIs over the HTTP protocol into RDF representations.
- Set RDF links to other data sources on the Web, so that clients can navigate the Web of Data as a whole by following RDF links.
- Provide metadata about published data, so that clients can assess the quality of published data and choose between different means of access.

An indication of the range and scale of the Web of Data originating from the Linking Open Data project is provided in Figure 4.2 below (a version updated to September 2011 is available at: richard.cyganiak.de/2007/10/lod)

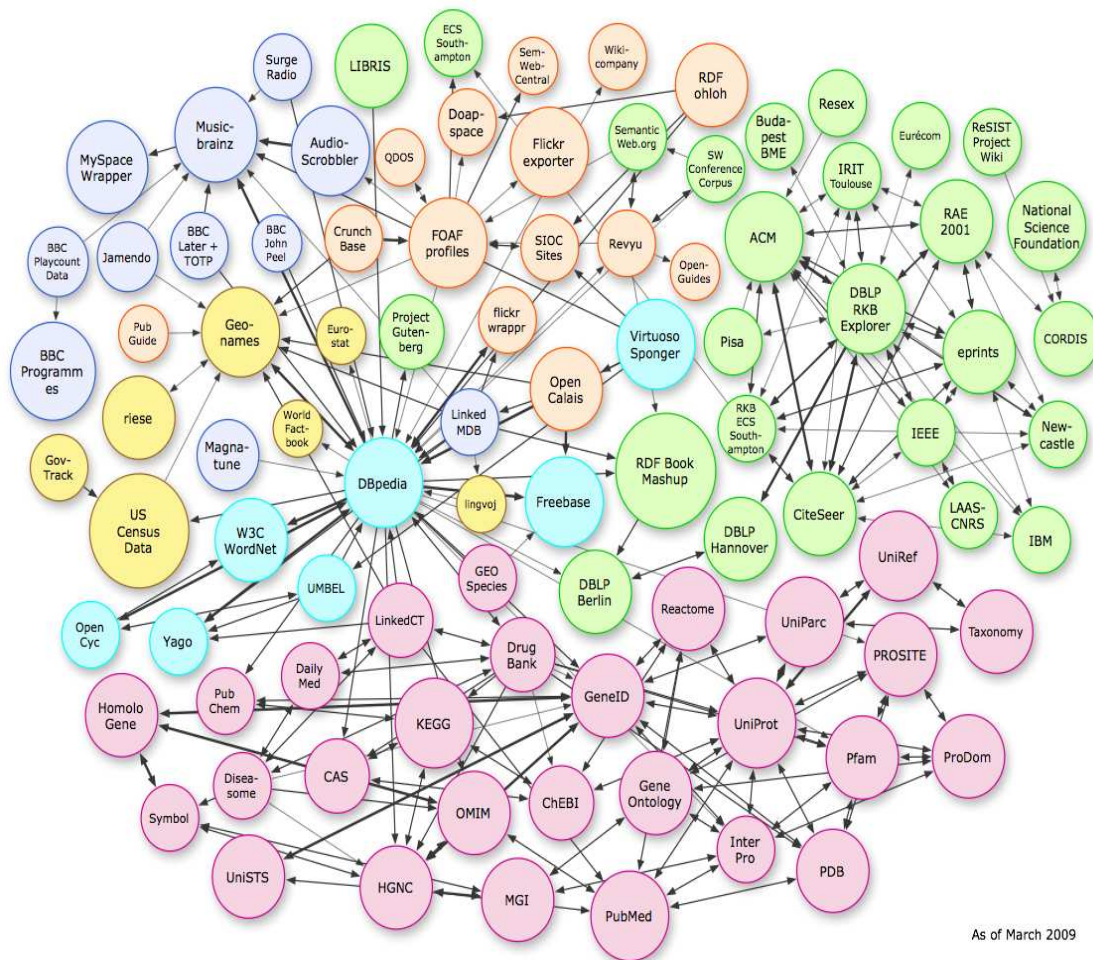
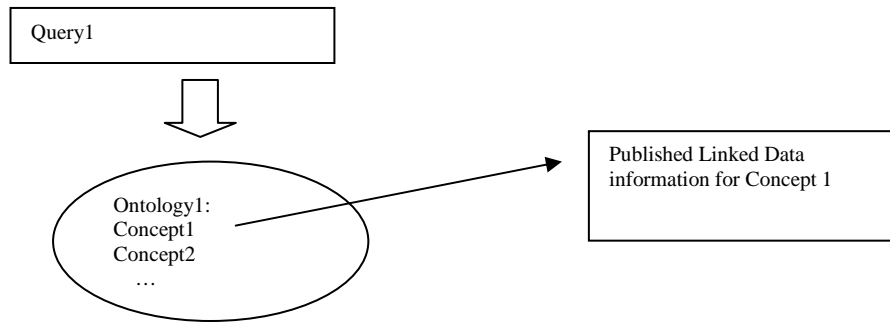


Figure 4.2. Relations between published Linked Open Data (from Bizer et al. 2009)

As is possible to see, the content of the cloud is diverse in nature comprising data about geographic locations (Geonames), people (FOAF), companies (IBM), scientific publications (DBLP), images (Flickr), etc.

The arcs in the figure indicates the links between interconnected data sets. In our case, the way in which Linked Data allows to expand the answer space of a query is represented by the fact they represent other possible representations of a certain concept. This representations are interrogable by - and can extend the knowledge of - different knowledge bases via the RDF based links (represented, in the figure 4.3 below by the black arrow).



4.3. Answer space extension through the interconnection of different KB

An example of RDF links, stating that the URIs in subject and object correspond to the same entity, is provided below:

```
SUBJECT: http://dbpedia.org/page/Venus
PREDICATE: http://www.w3.org/2002/07/owl#sameAs
OBJECT: http://dbpedia.org/page/Phosphorus_(morning_star)
```

Consider now the opposition between exemplar and prototype theories (see sect. 4.2.3 and the chapter 3 above). Both theories can be implemented in a representation system using the Linked Data perspective.

Let us consider first the case of prototype theory. A “dual” representation of concepts and reasoning mechanisms appears to be possible through the following approach: first a concept is represented in a formal ontology based on a classical, compositional DL system. Concepts in the compositional module (expressed with DL formalisms) are represented as in fig. 4.4. Every concept can be subsumed by a certain number of superconcepts, and it can be characterised in the terms of a number of attributes, that relate it to other concepts. Concepts correspond to one-argument predicates, and attributes to two-argument relations. To each attribute, it can be associated a restriction on the number of possible fillers. Concept/superconcept relations and attributes are assumed to correspond to necessary conditions for the application of a concept. DL formalisms allow to specify which of such conditions are also as sufficient conditions. Every concept can have one or more individual instances.

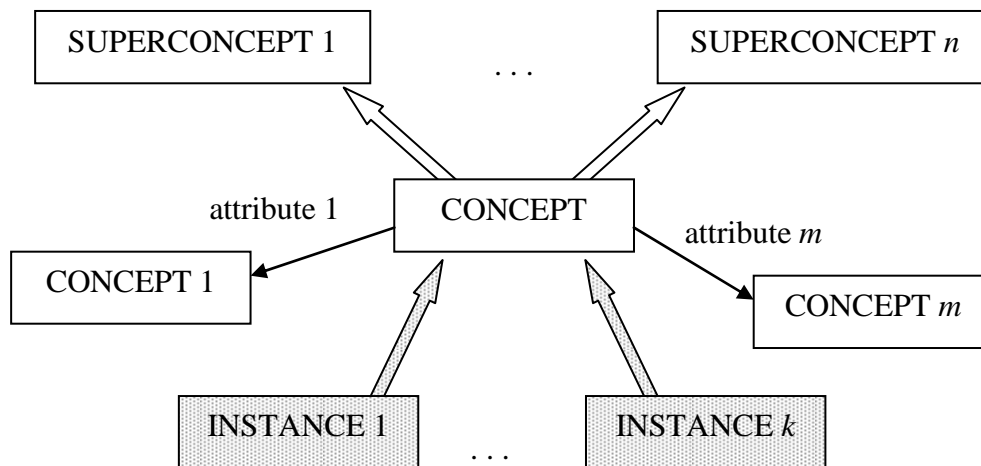


Fig. 4.4. Concept representation in the compositional module

As an example, consider fig. 4.5. The concept DOG is described as a subconcept of MAMMAL. DL concepts express only necessary and/or sufficient conditions; therefore, some details must be very loose. So, for example, according to fig, YYY, a DOG can have or have not a tail (this is expressed by the number restriction 0/1 for the attribute has_tail), and has an unspecified number of limbs (some dogs could have lost some limbs, and teratological dogs could have more than four limbs). LASSIE and RIN TIN TIN are represented as individual instances of DOG (of course, concepts describing individual instances can be further described, for example by specifying the values of attributes inherited from parent concepts).

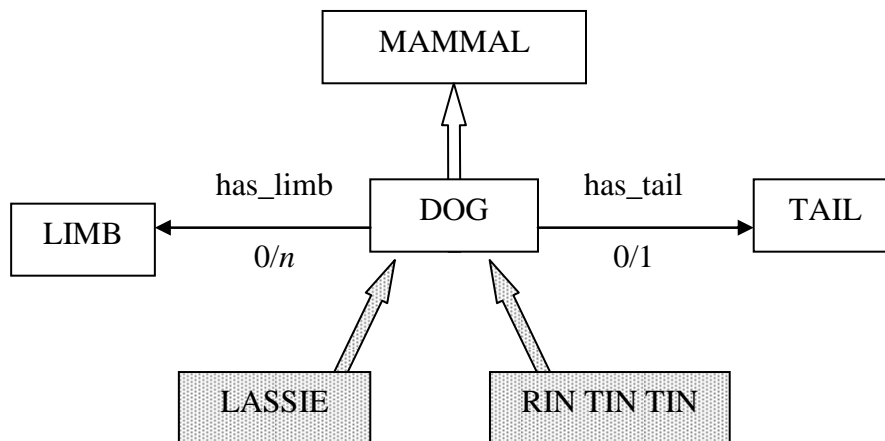


Fig. 4.5. Example of a concept description in the compositional module

At a second step the prototypical representation of the same concept is implemented using the Open Knowledge-Base Connectivity (OKBC) protocol. The knowledge model of the OKBC protocol is supported and implemented in the so called Frame Ontologies²³ that represent a possible solution for the prototypical representations of concepts and, if compared with other possible solutions, present the advantage of being easily interoperable with the classical DL system. Following the above mentioned example, we can suppose to represent a prototypical DOG in a frame ontology characterised by such slots as: hasFur, hasTail and Woof. A fragment of code a frame ontology about DOG is presented below.

```

<class>
<name>Dog</name>
<type>:STANDARD-CLASS</type>
<own_slot_value>
<slot_reference>:ROLE</slot_reference>
<value value_type="string">Concrete</value>
</own_slot_value> <superclass>:THING</superclass>
<template_slot>Fur</template_slot>
<template_slot>Tail</template_slot>
<template_slot>Woof</template_slot>
</class>
  
```

²³ Protege Frames is an ontology editor that supports the building of Frame Ontologies and that implements the knowledge model of the OKBC protocol.

According to the classical format of the frame representation, each conceptual frame is represented in terms of slots, facets and values. Obviously each slot can be, and should be, further specified, as in the example below in which the information that “a prototypical DOG has usually exactly 1 Tail” is expressed stating that the Slot Tail has MaxCardinality = MinCardinality = 1.

```

<slot>
  <name>Tail</name>
  <type>:STANDARD-SLOT</type>
  <own_slot_value>
    <slot_reference>:SLOT-MAXIMUM-
CARDINALITY</slot_reference>
    <value value_type="integer">1</value>
  </own_slot_value>
  <own_slot_value>
    <slot_reference>:SLOT-MINIMUM-
CARDINALITY</slot_reference>
    <value value_type="integer">1</value>
  </own_slot_value>

```

Fragment Code of a Frame ontology specifying cardinality constraints

Since it is possible to export (without losing the prototypical information) the Frame Ontologies in OWL language, the connection between the two types of representation can be done using the standard formalisms provided by the Semantic Web community within the linked data perspective (e.g. using the owl:sameAs or other “linking” constructs). In the case of sameAs, the model of the connection that can be provided between the DL representation and the prototypical one is the following:

<owl:Class rdf:ID="Dog">; DL CLASS representing the concept DOG

<owl:sameAs rdf:resource="URI_FrameOntology/#Class_NameDOG"/>; URI of the external frame based representation of the concept DOG
 </owl:Class>²⁴

In a similar way, an exemplar based information of a given concept can be expressed in a Linked Data format, and be connected to a DL ontological representation. Returning to the example before: the specific representation of the exemplar Fido (a specific

²⁴ Please note that, in this case, is assumed that OWL Full language is used.

DOG) can be linked via URI to an external resource representing, for example, the image of Fido it self (as shown in the figure 4.6 below)²⁵.

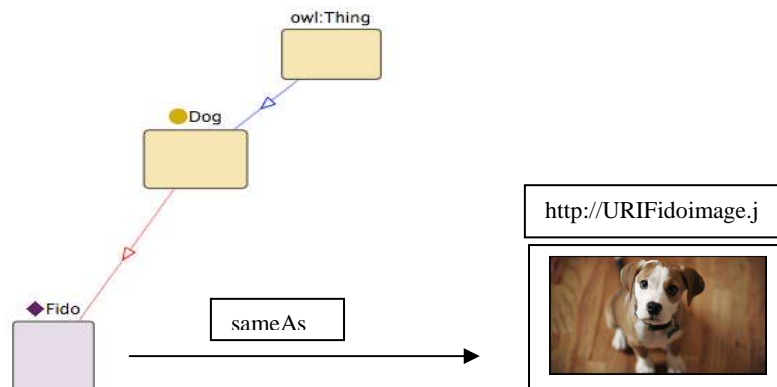


Figure 4.6. Ex. of connection between and exemplar and the relative concept in DL.

In this way, according to our hypothesis, different types of reasoning processes can follow different paths. For example, classification and other classical forms of reasoning could involve only the DL ontology, while different types of reasoning (such as non monotonic categorization) could involve exemplars and/or prototypical information.

4.5 Performing Heuristic Categorization

The main goal of the introduction of this type of reasoning regards the attempt of modeling a KR able, in a certain measure, to give results more similar to those of

²⁵ It is important to note that this type of representation, made possible by the integration between the dual architecture and the linked data approach, has – at least in principle - important consequences in the world of search technologies. In fact, the connection of different pieces of data for the same concept allows, in such fields in which the technology is already mature, to rethink the notion of search. Up to now, in fact, this notion is mostly related to textual search. Even when we search images, for example, we are forced to type text in a search box and the returned results are given according to the textual description of the image. With this type of information architecture, instead, is possible to think, for example, at an image retrieval in which the input is represented by an image itself and not by a text referred to an image. In fact, assuming the use of techniques of image processing and pattern recognition (well known in Artificial Visions and Robotics), and an interface supporting the upload of photo and /or of other types of data, is possible to imagine a direct connection between the external information provided via URI (containing, in our example, an image of a specific exemplar) and the “multimedia” input provided in the query process.

human cognition. Within common sense reasoning, humans often draw non deductive conclusions that are heuristically relevant in our life, and “rational” within the general economy of cognition. Non monotonic categorization (e.g. the assignment of an individual to a class according to incomplete and uncertain information) is one of these cases, and the introduction of the possibility of drawing heuristic categorization in ontological knowledge-bases would represent a relevant improvement for these systems. In the following pages we will propose a possible approach to this problem, taking into account suggestions coming from the field of machine learning and automatic categorization. In particular, within machine learning, two approaches have been adopted for the realization of different classifiers²⁶ the Nearest Prototype Classifier (NPC), based on prototypes, and the Nearest Neighbour Classifier (NNC), based on exemplars (Gagliardi 2010). Recently, different proposals of hybrid classifiers have been developed, in order to overcome the dichotomy between prototypes and exemplars, and to take advantage from both approaches. In our opinion, such a hybrid approach could be fruitful to face the problem of reasoning on typicality within the dual architecture proposed above.

In particular, we shall take into account the *PEL-C algorithm, Prototype-Exemplar Learning Classifier*, developed by Gagliardi (Gagliardi 2008, 2009, 2010). *The PEL-C* is a hybrid machine learning algorithm able to account for typicality in the categorization process, using both prototype and exemplar based representations. It is based on the nearest neighbour (NN) classification algorithm, according to which any new observed instance is assigned to the class of the nearest instance among the representative ones (RI). The PEL-C algorithm works as follow: in the starting step of the learning phase a prototype for every concept is calculated using a barycentric measure, then the distance between the training set (TS) and representative instances (RI) is calculated. For any new learning iteration, the instance of the training set (TS) that is farthest from the individuated prototype is added as candidate instance and compared with the prototype it self. This instance may or may not undergo to an abstraction process according to which the prototype can be re-calculated. If the

²⁶ Note that in the field of machine learning the terms classification and categorization are often synonymous.

abstraction takes place, the considered instances generate a new prototypical concept otherwise it is stored as an exemplar belonging to the prototypical concept. The termination conditions of the classifier can be defined *a priori* (e.g. in the case in which the number of learning iteration is known or, adaptively, during the its own performance. The learning algorithm proposed by Gagliardi is presented below. TS indicates the training set, RI the representative instance set and C(k) the items of a class k.

Table 4.1. The learning algorithm proposed by Gagliardi

1. Initialize RI with the barycenter of the class C(k)
2. WHILE NOT (Termination Condition)
 - [Find a new candidate instance]
 - 2.1 Calculate the distance between every instance of TS and every instance of RI
 - 2.2 Among the misclassified instances of TS, find the new instance which is the farthest from the nearest instance of RI belonging to class C(k)
 - 2.3 Add X to RI
 - [Update RI]
 - 2.4 Consider only the instances of RI and TS belonging to C(k). Call them RI(k) and TS(k) respectively
 - 2.5 Update the position of RI using the k-means clustering algorithm applied only to TS(k) with starting conditions Ri(k):
 - 2.5.1 Apply the Nearest Neighbor rule to the items of TS(k) respect to the RI(k)
 - 2.5.2 Iteratively re-calculate the locations of instances of RI(k) by updating the barycenters calculated respect to the subclasses determined with the NN rule
3. END

The application of this algorithm requires the choice of a metric of semantic similarity between concepts within the prototype and exemplar based component of the architecture. In the next paragraph I give an overview of the ways of calculating concept similarities in ontologies. Some of these measures have been taken into account in order to propose an adaptation of the PEL-C algorithm. It must be noted that PEL-C is in certain sense more general if compared to our present needs. For example, some steps of the learning phase are not needed because both the prototypes and exemplars are already available. Therefore, there is no need to calculate (at the initial stage) and recalculate (during the learning phase) the “prototype” of the representation.

4.6. Concepts Similarities in Ontology KB

Different methods and techniques have been developed to calculate semantic similarity between concepts within the ontologies. A first type of methods is based on the calculation of the geometric distance between concepts. This model is also known as edge counting model or network model because concepts are arranged in a graph structures (see Rada et al. 1989). In this perspective, similarity calculation is performed by counting the number of edges that need to be traversed to get from one concept to the other. In this approach, the less is the distance, the greater is the concept similarity. However, this method is rather simple, and it mainly considers the “is-a” relationship that cannot accurately reflect other semantic aspects of similarity between concepts. Furthermore it depends on arbitrary aspects of the representation: among the concept DOG and the concept ANIMAL there can be an arbitrary number of intermediate concepts and this depend from contingent factors that do not deal with the similarity between concepts.. The techniques based on link counting were already criticized in the KL-ONE systems.

A different method is based on information content algorithms (Resnik 1999). In this approach the semantic similarity between two concepts is determined taking into account both the amount of information that the two concepts have in common in their last common ancestor - called Most Specific Common Abstraction (MSCA) - and the probability of concept occurrence in the same corpus. According to the information content approach, similarity is obtained calculating the entropy of concepts. The more information two concepts have in common, the closer semantics they have. A major drawback of this method (as reported in Zhang and You 2010), is represented by the fact that it is entirely dependent by the statistics of occurrences of the corpus²⁷ rather than on the analysis of the characteristics of ontology definitions.

Semantic similarity methods are in their turn usually distinguished between single ontology similarity methods, which assume that the compared concepts are from the same ontology, and cross ontology similarity methods, which compare concepts from two different ontologies. Edge counting and information content methods work by exploiting structure information of the concepts in the hierarchy (i.e., position of terms) and are best suited for comparing concepts from the same ontology, while, for cross

²⁷ The ontology can be seen as vocabularies and therefore as lexical corpora of defined terms.

ontology concept similarity, hybrid approaches - obtained through a mix of different methods - are used.

Another method for concept similarity calculation is based on the lexical based matchmaking algorithms. They work directly on the name of the elements to compare (Williams et al. 2003) through the so called Edit Distance (ED), a basic function that calculates the number of substitutions which are necessary to transform the first word into the second one. This method do not consider the semantic of the concepts but only the string of characters composing the words. For example: for the words LOGIC and LOGICIAN the ED (LOGIC, LOGICIAN) = 3 since three letters have to be added the transform the first word into the second one. ED is usually incorporated into a weighted formula, which takes into account also the length of the shorter of the two words. So, the resulting formula is the following

$$similarity(L1, L2) = \frac{\max(0, \min(|L1|, |L2|) - ED(L1, L2))}{\min(|L1|, |L2|)}$$

This formula gives a similarity measure included between 0 and 1, where 0 is a bad match and 1 is a perfect match. So, in the above example, the complete similarity measure would be the following:

$$similarity(\text{Logic}, \text{Logician}) = \max(0, (5-3)/3) = 2/3.$$

Despite its simplicity, this method present a lot of disadvantages because it completely ignores the semantics of the terms and, therefore, the semantics of the concepts represented by that terms. For example different words pairs such as (DOG, DOGS) and (ACE, FACE) result to have the same semantic similarity event if their similarity it is not the same.

Another well known approach is based on dictionary matchmaking. The difference with the previous method is represented by the fact that now there is a common vocabulary used as reference for the concept comparison. One of the most used vocabularies is

Wordnet²⁸. The relations expressed in Wordnet represent the basis for the similarity measurement, which is calculated analogously to the above mentioned network approach (the number of the relations to be traversed to go from one concept to another is counted). In addition, each type of relation is differently weighted (Castano et al 2003). For example: two words connected via synonymic relations have 1 as similarity value, while words connected via the hyperonym relation have a lower similarity value, e.g. 0.7. The problem of this approach is related to the usage of the synset in Wordnet. In Wordnet, in fact, the same word often has different synonyms, but it may happen that not all the members of the synset are synonyms in the same way. Let us consider an example taken by Hall (2006): consider the word FOREST. FORESTS has WOOD and WOODS as synonyms. But the problem here is that WOOD and FOREST do not have exactly the same meaning. WOOD, in fact, is a growth of trees that is smaller than a FOREST. In addition, the word WOODS usually denotes an area that is much larger than a WOOD. These differences are totally lost. A further problem is that the principles on which the semantic relations are constructed are often different for each concept.

A different approach is based on structural matchmaking. It uses the ontology structures to calculate the concept similarity. The basic idea is that similar concepts have a similar set of surrounding concepts. In its simplest version, similarity is calculated taking into account the number of children, of parents and of properties that the two concepts have (Maedche and Staab 2002, Castano et al. 2005). However, the idea that similar concepts have similar surrounding concepts is based on the wrong assumption that different people model the same domain in a similar way. But this is not always the case. There are, in fact, a lot of modeling differences for the same domain depending by the specific modeling needs, or simply by idiosyncratic choices of the modellers. As a consequence, the same concept is often represented with a different structure in different knowledge bases. Such differences make this methods quite imprecise. Moreover, the structural matchmaking completely ignores the semantics of the concepts to be compared.

²⁸ WordNet is an on line lexical database for the English language developed by Miller since the 1995 (Miller 1995). It categorize the words according to four syntactic categories (nouns, verbs, adjectives, and adverbs) and represent the semantic relations between the terms expressing: synonymic relations, hyponymic/hyperonymic relation, meronymic/holonymic relations, entailment relations, antonymic relations, troponymic relations.

Another relevant method is the Description Logics matchmaking (Paolucci et al. 2002). This approach is useful when concepts are encoded in ontology languages based on DLs and belong to different KBs. The Description Logics matchmakers works as follows: it attempts to use subsumption to insert the concept from one knowledge base in the other one. Depending on whether and if the concept from first KB is inserted into the hierarchy of the second KB, different levels of matching are distinguished. There is an exact match in the case in which the reasoner determined the equivalence from a DL point of view (Klen et al 2004, and Horrocks 2004); there is a plug-in match if the concept from the first ontology is subsumed by the concept of the second ontology (in this case the concepts are connected via an ISA relation and, even if that is not completely correct, are assumed to be similar. In the opposite case of the plug-in match (i.e., when the concept of the first ontology subsumes the concept of the second one), there is a subsume match. This match is considered at a lower level with respect to the plug in match because the relation concept vs super concept is seen as asymmetric: the sub concept is more similar to the super concept than the super concept to the sub concept. The intersection match is when the concepts cannot be arranged in a subsumption hierarchy but are not formally in conflict between them (Li and Horrocks 2004). Finally, a disjoint match is when the definitions of the two concept are in conflict among them (see Li and Horrocks 2004, Lemmens and Arenas 2004).

If compared with the other approaches, the DL method takes into account the semantics of the concepts, represented by the DL description defining their meaning. The others are mainly based on schema comparison. However, there are different problems also with this type of algorithms. The first one is represented by the situation in which a concept of the first ontology is subsumed by (in the case of plug-in match) or subsumes (in the case of subsume match) more than one concept of the other ontology.. Namely: for the plugin match a problem emerge if there are, for example, two target concepts that subsumes the source concept and that are not arranged in some kind of is-a relation. In this case the algorithm can only assume that the two concept target have the same similarity with the source concept- However, one concept could be very specific and almost the same as the source concept and the other one could be more abstract (see Hall 2006 for details). For the subsume match, indeed, in the case in which the source concept subsumes more than one target concept (let say two concepts) it is nearly

impossible to say which of two subsume matches is better. Another problem regards the intersection match. Here the basic assumption of the algorithm is that two concepts that are not in a subsumption relation probably are not very similar. However, many concepts are not in a hierarchy relation between them and can be very similar. Finally, another drawback of the DL approach is that it only calculates the number of definitional parts that match and those that not, but it does not say anything about how closely the matching parts match and which is the “semantic distance” between the non matching parts. Let us consider, for example, the situation sketched below:

$$\begin{aligned} \text{Lake} &= \text{WaterArea} \wedge \forall \text{ hasWater.}(\text{Standing} \wedge \text{Fresh}) \\ \text{Inland Water} &= \text{WaterArea} \wedge \forall \text{ hasWater.Inland} \\ \text{Water} &= \text{WaterArea} \end{aligned}$$

In this case the DL based algorithm will find that LAKE and WATER are more similar than LAKE and INLAND WATER even if LAKE and INLAND WATER give more information on the type of water that they contain and are intuitively closer among them. This because the definition of the type of water contained conflicts from a DL point of view (it is not possible for the algorithms to compare the two types) while LAKE and WATER are fully compatible.

A completely different approach for the calculation of concept similarities is based on cognitive models. Here the basic assumption is that semantic similarity measures in artificial systems should give results analogous to those given by human experts. Therefore, it assumed that the calculation of the semantic similarities between concepts must be based on a cognitive model.

A first well known model is the Feature Based Model proposed by Amos Tversky (1977). The basic assumption behind this approach is that concepts are defined by unstructured lists of features that, together, compose their description. The similarity of two concepts C1 and C2 is a function of the features common to C1 and C2, of those in C1 but not in C2, and of those in C2 but not in C1. (i) is the formula for the similarity of two concepts.

$$2. \text{sim}(C1;C2) = \frac{[\text{Shared features}]}{[\text{Shared features}] + [\text{Features only in C1}] + [\text{Features only in C2}]}$$

In order to evaluate pros and cons of this approach, let us consider the simple example reported below:

$$\begin{aligned} \text{Forest} &= (\text{Vegetation}, \text{Trees}) \\ \text{Coniferous Forest} &= (\text{Vegetation}, \text{Coniferous Trees}) \end{aligned}$$

According to the Tversky's algorithm, the similarity between the two concepts would be 1/3 because:

$$\text{sim}(\text{Forest}; \text{Coniferous Forest}) = \frac{1}{1 + 1 + 1}$$

Unfortunately, as soon as one concept is compared to two or more other concepts, then problems in the similarity measure become evident. For example, if we compare a third concept Scrub Vegetation = (Vegetation, Scrub) to the two concepts of table 1, then the similarity between the first two concepts turns out to be equal to that between the first and the third concepts, in spite of the fact that Forest is more similar to Coniferous forest than to Scrub vegetation .

An additional problem is that the comparison between features is limited to the fact that they have or have not the same name. So, for example, the fact that a Coniferous Tree is a Tree cannot be modeled, and thus the comparison of Forest to Coniferous and to Scrub vegetation produces the same value. This problem can be solved by extending the definitions as shown in the table 4.2 below so that the hierarchy (and its relations) is represented in the feature list.

$$\begin{aligned} \text{Forest} &= (\text{Vegetation}, \text{Trees}) \\ \text{Coniferous Forest} &= (\text{Vegetation}, \text{Trees}, \text{Coniferous Trees}) \\ \text{Scrub Vegetation} &= (\text{Vegetation}, \text{Scrub}) \end{aligned}$$

$$\begin{aligned} \text{sim}(\text{Forest}; \text{Coniferous Forest}) &= \frac{2}{2 + 1 + 1} = \frac{1}{2} \\ \text{sim}(\text{Forest}; \text{Scrub Vegetation}) &= \frac{1}{1 + 1 + 1} = \frac{1}{3} \end{aligned}$$

Table 4.2. Example of the extension of a definition in the Tversky's algorithm

Another cognitive method is based on the theory of Cognitive Spaces proposed by Gärdenfors (Gärdenfors 2000, 2004). In the cognitive spaces model, concepts are points or areas in a hyperspace. Each property or aspect of the concept is modeled as a separate dimension, and each dimension can in its turn have an internal structure. Such an internal structure allows the cognitive space model to closely reflect human cognitive abilities. Similarity in conceptual spaces is defined either as city block or Euclidean distance. City block metric is used for those dimensions that are separable and do not influence each other, while the Euclidean metric is used for the inseparable dimensions. Additionally the conceptual spaces model also contains weights for the different dimensions, so that the relative relevance of the different dimensions can be considered in the similarity calculation.

The problems with the cognitive space model are twofold. First, it cannot model relations between concepts. So, SAUSAGE DOG cannot be described as a DOG with certain characteristics. This could be modeled as another domain but does not fully capture the semantics of the concept. In addition all dimensions apply to the complete concept. It is not possible to define that a certain dimension is only relevant for parts of the concept. For example the fact that in a mixed housing/park urban area the type of building is only relevant for the housing part cannot be modeled.

The second problem is that it is often hard to identify the internal structure of the dimensions. For some dimensions it is easy to describe their structure, as in the case of a conceptual space describing the human colour space in terms of hue, luminosity and saturation (figure 4.7).

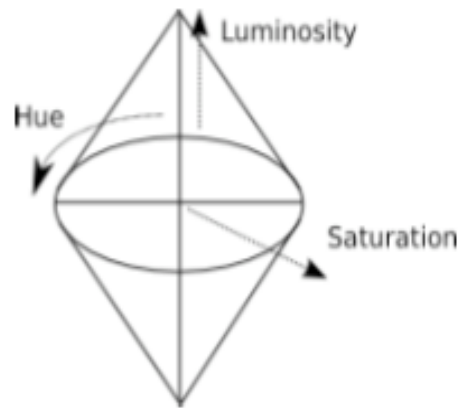


Figure 4.7. Representation of colours in terms of Conceptual Spaces

Unfortunately, for a large number of dimensions it is very hard to have a correct description. In addition, it is not quite clear how to handle concepts that have no definite values for a certain dimension.

These problems represent, at the current state of art, the main obstacles to employing the cognitive spaces model in an integration scenario. However, if these problems will be fixed, this approach represent a really powerful model for taking into account the cognitive aspects of concept similarities in ontology based structures. Progress, in this sense, have been made with the development of the CSML (Conceptual Space Markup Language) language- - (a XML based representation language, see Abams and Raubaul 2009, 2010). However this developments are still at an early stage.

In conclusion, the best method for concept similarities calculation suitable for our proposal seems to be the feature based model. In the next paragraph we propose to integrate this approach developing an adaptation of the PEL-C algorithm.

4.7 Proposed adaptation of the PEL-C algorithm

As mentioned in the previous paragraph, the PEL-C algorithm as proposed by Gagliardi need an adaptation in order to be used in the dual architecture context. In facts, in our proposal, the prototypes of the concept are already given (in the typical knowledge part), and there is no need of the learning phase of the algorithm, in which the prototypes of concepts are calculated. In a certain sense, such prototypes represent the “barycenter” of the concepts. From them, the semantic distance of the new instances is calculated, and this allows to determine if they belong to the prototypical class or not. An important aspect to take into account is the method for the calculation of the concept similarities. Here we propose to follow the model inspired by the Tversky’s feature theory (see paragraph above), according to which the similarity of two concepts can be calculated as the ratio between the shared features and those features that are only in one or the other concept. In this way, the adapted learning phase of the PEL-C algorithm is provided introducing what we call Category Set (CS) instead of the classical Training Set (TS) item. CS represent the set of the instances (new or already presented in KB but not assigned to a class) that need to be categorized. The adapted algorithmic procedure result to be the following:

1. Consider the representation in the typical component (TC) as the barycenter of the concept C
2. WHILE NOT (*Termination Condition*)
 - [*Find a new candidate instance*]
 - 2.1 Calculate the distance (using the Tversky’s feature model) between every instance of the Category set (CS) and the prototypical concepts of the typical component (TC)
 - 2.2 Create a list containing the results of the semantic similarity between concepts in CS and in TC.
 - 2.3 For each concept in CS: IF the semantic similarity result is OVER a predefined threshold THEN [assign the concept Ck(S) to the prototypical concept in TC]
 - ELSE [do not categorize the concept Ck(S) as belonging to the typical concept in TC]
3. END

Table 4.3. Proposed adaptation of the learning algorithm

The termination conditions here is determined by the end of the of the categorization process between the concepts in CS and the concepts in TC. Please note that if the typical component is composed only by a single “frame” the algorithm only perform a one to one comparison between the prototypical representation and the candidate instances. The result of this process consists in an instantiation task performed according to non monotonic reasoning. It, in fact, is based on the Tversky’s cognitive model and on its relative algorithms. As we will see in the paragraph below, the result of this non monotonic categorization turn out to be useful for the improvement of certain types of performances related to the instance checking task.

4.8. Expected Results

The general presented architecture can be realized for different purposes and tested in different ways. In our case we focus our attention on the information retrieval and on the reasoning processes performed on ontology knowledge bases. Therefore, it is our intention to evaluate our proposal by comparing its performance with that of a traditional ontology based system representing the same domain. What we expect is a double result (Frixione M., Lieto 2011). From the information retrieval point of view we expect an enriched query-answering mechanism that should take advantage from the integration of different types and/or levels of information provided for the same concept. The evaluation task²⁹ for this issue is based on a control known as “property checking”. It consists answering such questions as “does the class A have the property b?”. In the following example we explain in which sense a better result is expected. Let us suppose that an user runs an informational query³⁰ on a “dual” knowledge base representing information concerning fruit in order to know which kind of citrus is yellow (that is an indirect formula to ask: ”does any citrus have the property of being yellow?”). The expected answer that fits the informational needs of the user is “lemon”. However, does not exist in the compositional knowledge base any kind of citrus that

²⁹ The evaluation tasks that are proposed are referred to the step 3 of the above mentioned procedures of the system behavior.

³⁰ According with the Information Retrieval literature, *informational queries* are different form *transactional* and *navigational* queries. In *informational queries* , the user intention is to obtain a specific information concerning a given object (see Jansen et al 2008).

has the property of being yellow as a defining condition. Being yellow is not a necessary condition for being a lemon and, therefore, this property is not represented into the class lemon of a DL ontology. However the property “to be yellow” is relevant from a cognitive point of view to characterize the concept “lemon”, and, according to our hybrid approach, can be represented into the prototypical component of the class “lemon”. In this way is possible to retrieve the desired information from the prototypical and/or exemplar part of the representation. So, given a query on the knowledge base such as:

```
SELECT? citrus
```

```
WHERE {?citrus :has colour :YELLOW }
```

the result returned from the DL representation should be null, while the “correct” answer (correct with respect to the intention of the user) will be generated from the prototypical component of the representation. Improving, in this way, the answering mechanism of the system.

Another expected result is based on the improvement of the inferential mechanisms provided by ontology based systems. Our cognitively inspired architecture, in fact, would make possible to consider a new type of reasoning with the introduction of a non monotonic, heuristic, process of categorization³¹ performed as indicated in the previous section. In this case the evaluation tasks consist in the “instance checking” control based on the interrogation of the knowledge base. Instance checking aims to answer at such questions as “is a particular instance member of a given concept?”. We expect that, the prototypical and exemplar based representations, performing a non monotonic reasoning process, could provide a different answer if compared to a traditional DL ontology. For example: it could result that an instance A is not a member of the Class A* in the DL component while it is an instance of the Class A** in the prototypical representation of the same concept. This result does not cause inconsistencies or create any problem because of the separation of representation and reasoning process and, in

³¹ While monotonic categorization is already performed on classical DL ontologies, non monotonic categorization is not yet performed and forecasted.

addition, it gives to the system the possibility of considering an enlarged space of answer provided through non deductive and cognitively founded reasoning mechanisms.

4.9. Prototype and Exemplars Representation

In the previous chapter we introduced the prototype and exemplar theories of concept representation and suggested that the use of both the forms of representation, within the proposed modeling approach, can provide interesting insights.

In the following we try to argue why, in our opinion, it is important to keep both these representational forms. A first motivation is represented by the possibility of improving the representational capabilities of the ontology based systems allowing, for example, to attach prototypical information to the exemplars of a specific class. More in general we claim that the possibility of representing prototypical information at the exemplars level allows to take into account of more aspects within a representation, augmenting the quantity and the quality of data made available.

The importance of keeping multiple views (classical, prototype based and exemplar based) on the same representation can be explained by the fact that they allow to have artificial representation which are closer to the reality.

A simple example of different possible views for the same concept is taken by Lukyanenko and Parsons (2011): when professors think about its students each student retains a plethora of individual features. Some students may require more attention than others. The distribution of attention for each student may also change over time. However, a classical university domain ontology, ignoring this information, typically defines a Student class using the same set of properties. And, furthermore, this representation usually does not include any information regarding the individual differences between each student³². Even if this representational choice is understandable and seems to be reasonable under a certain perspective, we claim that, with this modeling approach, some information on individual/exemplars differences is lost. Namely what is lost is the information regarding the typical features of the exemplars.

³² It is important to note that the individual differences can be mainly described in terms of typical properties (e.g. the Student 1 can be described as funny, the Student as shy etc.).

However, this information can turn out to be very useful in many circumstances as expressed in the example of Anna and the cinema provided in the chapter 4.3. As we have seen, in fact, in that case the presence of both the representations within the proposed modeling framework allows, at least in principle, the retrieval of prototypical information which is linked to the exemplars³³. In our opinion there are other cases able to illustrate why a dual, prototype and exemplar based, representation of concepts could turn out to be useful for the representation of non classical concepts in ontological knowledge bases also from a technological point of view. In the first place, there are kinds of concepts that seem to be more suited to be represented in terms of exemplars, and concepts that seem to be more suited to be represented in terms of prototypes. For example, in the case of concepts with a small number of instances, which are very different from one another, a conceptual representation in terms of exemplars should be more convenient. An exemplar based representation could be more suitable also for non linearly separable concepts (see the previous section). On the other hand, for concepts with a large number of very similar instances, a representation based on prototypes seems to be more appropriate. Consider for example an artificial system that deals with apples (for example a fruit picking robot, or a system for the management of a fruit and vegetable market). Since it is no likely that a definition based on necessary/sufficient conditions is available or adequate for the concept APPLE, then the system must incorporate some form of representation that exhibits typicality effects. But probably an exemplar based representation is not convenient in this case: the systems has to do with thousands of apples, which are all very similar one another. A prototype would be a much more natural solution. Thus, the presence of both a prototype and an exemplar based representation seems to be appropriate. Let us consider the concept BIRD (fig. 4.8). And let us suppose that a certain number of individuals b_1, \dots, b_n are known by the systems to be instances of BIRD (i.e., the system knows *for sure* that b_1, \dots, b_n are birds). Let us suppose also that one of these b_i 's (say, b_k) is a penguin. Then, a prototype P_{BIRD} is extracted from exemplars b_1, \dots, b_n , and it is associated with the concept BIRD. Exemplar b_k concurs to the extraction of the prototype, but, since penguins are rather

³³ The cautulative expression “in principle” is necessary in this case because in that example, the retrieval is subordinate to the realization of a non monotonic categorization process operating transversely among the two layers.

atypical birds, it will result to be rather dissimilar from P_{BIRD} . Let us suppose now that a new exemplar b_h of penguin must be categorized. If the categorization process were based only on the comparison between the target and the prototype, then b_h (which in its turn is rather dissimilar from P_{BIRD}) would be categorized as a bird only with a low degree of confidence, in spite of the fact that penguins are birds in all respects. On the other hand, let us suppose that the process of categorization takes advantage also of a comparison with known exemplars. In this case, b_h , due to its high degree of similarity to b_k , will be categorized as a bird with full confidence. Therefore, even if a prototype for a given concept is available, knowledge of specific exemplars should be valuable in many tasks involving conceptual knowledge. On the other hand, the prototype should be useful in many other situations.

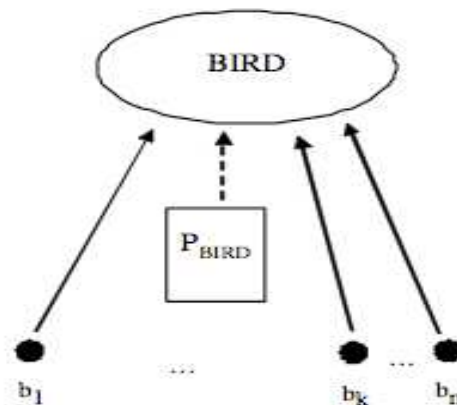


Figure 4.8. Exemplars and Prototypes for the concept BIRD

Beyond the representational advantages, there is also at least one reason that goes in the direction of suggesting this double level of representation (which is made possible by proposed modeling framework). It is related to the reasoning issues. In fact, as mentioned in the chapter 3, there are different dynamics involving the process of categorization for exemplar and prototype based representations. Therefore, in certain cases, for the non monotonic categorization task, it could result to be better to have a categorization process involving the new item and an exemplar, while (in other cases) it could result to be more useful to have a comparison between the new item and the prototype. Moreover, following the proposed modeling approach, it is possible to

hypothesize the realization of different reasoning modules operating independently, on the different representations. The interesting issue, at this level, is represented by the fact that this independence cannot be cause of contradictions because the reasoning modules can be run on different pieces of knowledge according to the “cascade model” proposed in section 4.1. However, this part regarding the enhancement of reasoning processes trough the realization of a non monotonic reasoning module able to take into account of aspects of typicality, has been at this moment only theorized but not yet implemented, tested and optimized. In the next chapter an evaluation of the proposed modeling approach is provided and the obtained results are analyzed and discussed

Chapter 5. Evaluation and Discussions

An evaluation study has been conducted in order to test the behavior of the proposed architecture and to compare its results with that ones obtained by a standard DL based representation. The main aim of the evaluation regarded the observation and the analysis of the answers provided by the proposed modeling approach considering different representational configurations. In the section 5.1 the experimental set-up created for the evaluation phase is presented. In 5.2 the obtained results are summarized and in section 5.3 they are discussed.

5.1. Evaluation

The evaluation has been done on an information retrieval task. The system performance, in fact, has been tested through its capability of retrieving information starting from apparently irrelevant traits for certain defined concepts. The importance of retrieving information starting from typicality traits is given by the fact that this represent one of the most common – and successful - heuristics of search used by humans. Humans, in fact, often use peripheral, secondary and typical traits in order to retrieve and or acquire information on a specific domain. These traits, even if not formally relevant for the information definition and structuring, represent, usually, central features from a cognitive point of view and are very useful for information extraction and retrieval. For these reasons, the goal of our evaluation is to reflect on the behavior of the proposed architecture for a particular well known kind of query named - in IR literature - informational query (see footnote 30 of the previous chapter for details).

The evaluation test on this task has been realized with the following experimental setup: in a first phase a toy DL knowledge base has been built and connected with a typical representations of the concepts.

The created toy knowledge base is composed of 20 classes, 10 attributes and 10 instances. It has been built according to the modeling requirements provided by the

proposed approach. Namely: the concepts in DL are only represented, when possible, as sets of necessary and sufficient conditions, while the representation of prototypical traits are demanded to an external typical component realized as a Frame Ontology following the Linked Data approach (it has been published as a RDF based statements available and interrogable on the web).

The fact that the experimental KB respects the modeling requirements proposed in our approach is not secondary. In fact, many existing DL knowledge bases are based on approaches which mix, in a unique DL based representation, different types of information using the same formalism. In this sense, a typical example of a wrong modeling is represented by the case of the concept “dextrocardiac” presented in the section 4.1.2 by Motik et al. (2006).

The entire knowledge base used in the pilot study is available on the web at the following address: <http://www.dualontologyarchitecture.net/ontology.owl>. It is a KB representing specific types of fruits, in which the concepts (such as, for example LEMON and ORANGE) have been modeled in a distributed way according to the proposed approach. The classes of the ontology for the classification of different types of fruits (and, in particular, different types of citrus) are represented in Fig. 5.1.

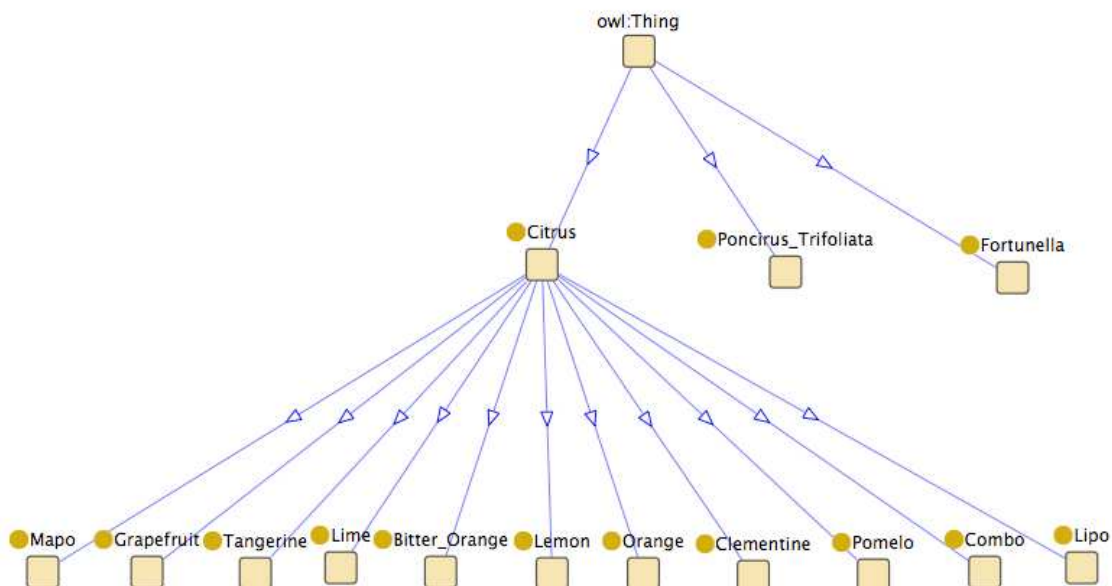
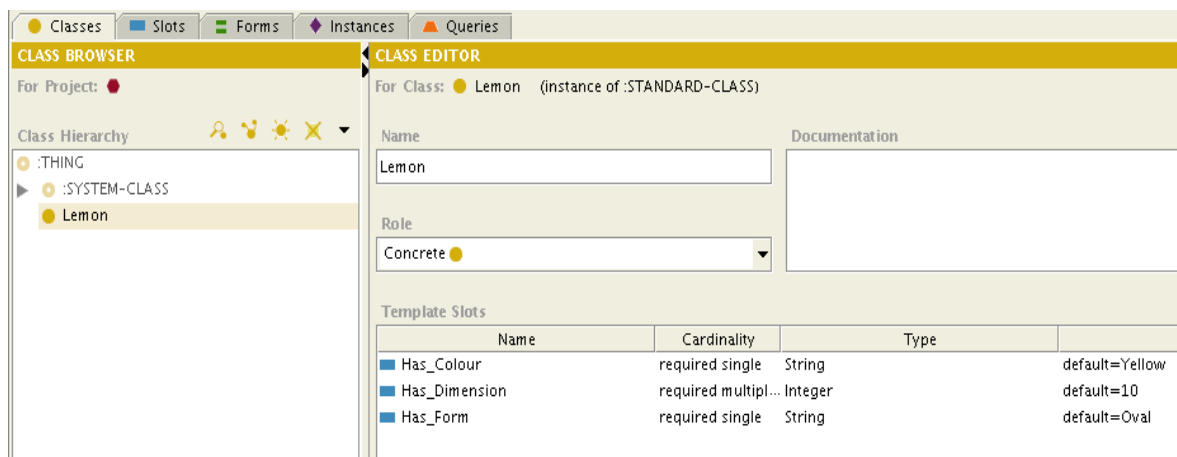


Figure 5.1. Taxonomy of a toy DL knowledge-base

Other non domain dependent classes (such as Country, Field of Use, Vitamins etc.) are represented in the KB, in order to allow, jointly with object properties such as contains, is_contained_in, is_produced in etc., to represent, for example, that LEMONS contain Vitamine C and are produced, with a certain percentage, in certain specific countries. The instances in the KB belongs to the classes Vitamin (Vitamine_A, Vitamine_B etc.) and Country (Brazil, Spain, Italy etc.).

Concepts such as LEMON and ORANGE are also represented as prototypes in a Frame Ontology, modeling only typical information such as: has_colour, has_dimension, has_form etc. The screenshot below shows how the first three slots have been filled with default values in the case of LEMON. For example: the representation of a typical LEMON contains the information that usually a Lemon is Yellow and has an oval form.



A similar representation has been produced for the other concepts. The full list of the prototypes created for the experiment is available at the following address: <http://www.dualontologyarchitecture.net/typicalrepresentations.txt>.

Prototypes are linked to the correspondent concepts in the DL knowledge base via the *owl:sameAs* construct. A battery of queries has been run first on the classical DL representation alone, and then on the representation obtained through the interconnection of both the prototypes and the DL component. The queries performed on the two KB have the following SPARQL form:

SELECT ? CONCEPT

WHERE {?CONCEPT :has typical PROPERTY : Typical Value of the typical PROPERTY .
}

Fig 5.2. Typology of query performed in the experiment

This type of query correspond to the “property checking” task explained in chapter 4.8. In this case, the way in which the query has been built shows that the information extraction request is based on the typical features of the concepts. This evidence comes out from the WHERE clause inserted into the query system. The full list of query performed on the DL KB and then repeated on the dual architecture during the pilot study is presented here: <http://dualontologyarchitecture.net/sparql.txt>. The obtained results have been evaluated using precision and recall measures and, in some cases, using a simple yes/no counting approach regarding the success/unsucces of the concept retrieval task.

Precision and recall are two standard measures within the information retrieval field. They are used, for instance, to evaluate the efficiency of a search engine in order to understand if the retrieved information is relevant with respect to the information need of the user. They are usually calculated as follow:

- PRECISION= Relevant Retrieved/ retrieved ($R,R / (R,R+NR,R)$)
- RECALL= Relevant Retrieved/ relevant ($R,R / (R,R+R,NR)$)

	RELEVANT	NOT RELEVANT
RETRIEVED	R,R	NR,R
NOTRETRIEVED	R,NR	NR,NR

Table 5.1. Precision and recall

In other words, precision is the ratio between the relevant retrieved information and the totality of all retrieved information; recall (which is a coverage measure) is the ratio between the retrieved relevant information and the totality of relevant information that is retrievable in principle.

In our case, we calculate the precision value for a specific query Q , as the number of relevant concepts that have been retrieved divided by the total number of the retrieved concept. Recall, instead, is the number of retrieved relevant concepts divided by the total number of relevant concepts.

Four main experimental situations have been considered for the evaluation (we have called them E1, E2.1, E2.2 and E3). To each of them has been associated a specific configuration of the two representations (the compositional one and the typical one) composing the proposed architecture. In certain cases (E2.1, E2.2) we individuated two control situations within the experimental situation itself. As we will see further, the analysis of these situations, within the same general set-up of the experiment, allowed us to explain in which representational cases the proposed architecture obtained good or bad results if compared to classical representations. The four experimental situations and the relative data emerging from the evaluation, are described in major detail in the following pages. The general picture of the different experimental set-up considered is schematized in the figure 5.3 below. The main experimental situations are represented by the blocks E1, E2 and E3. The four control situations investigated in E2 are represented by E2.1.1, E2.1a, E2.2.1 and E2.2a respectively.

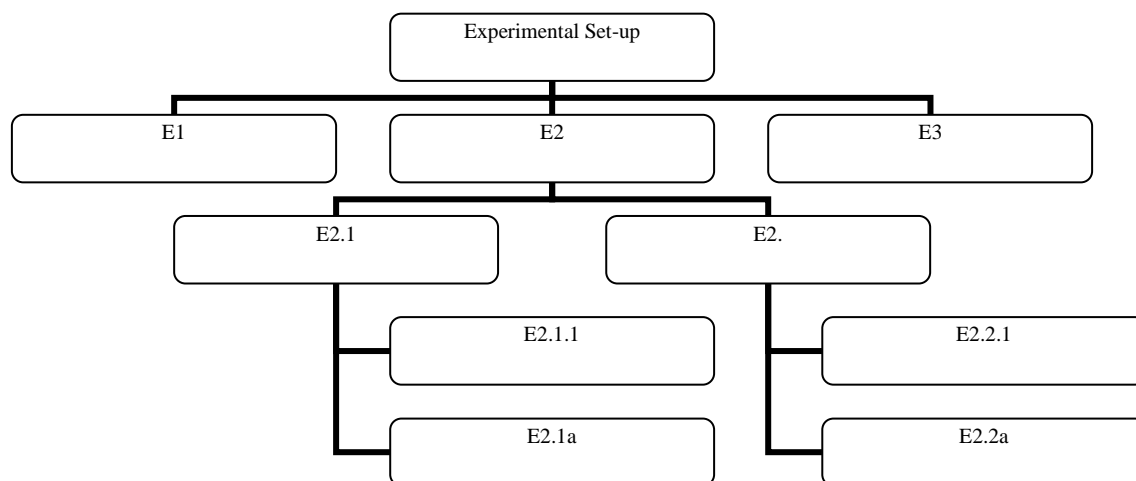


Figure 5.3. Experimental set-up situation for the evaluation task

In the first experiment E1, the prototypical properties have been not represented in the DL ontology, and have been completely demanded to the external component. For example, in E1 the DL ontology (<http://www.dualontologyarchitectures.net/ontology.owl>) represents definitional properties such as: contains, is_contained_in (regarding the chemical composition of the fruits) et cetera, while it does not represent properties such as has_colour, has_dimension, has_taste etc. The general structure of the E1 knowledge base is represented in the figure 5.4 below.

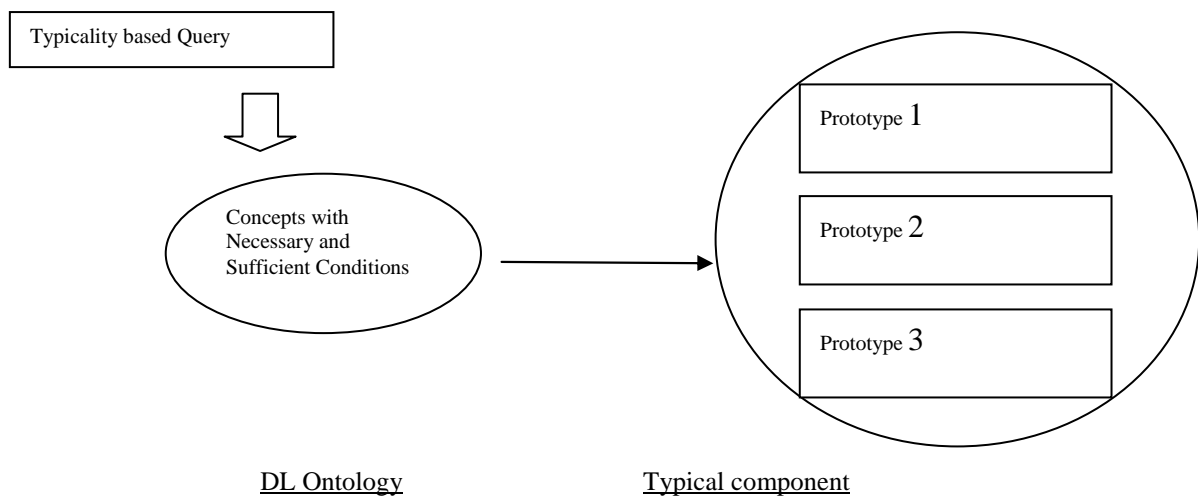


Figure 5.4. E1 experimental situation

In the case of E1, the obtained results are in line with the expected ones. In fact, on a battery of 30 queries based on typical features, it resulted to be impossible to retrieve such information from the DL component because such information was not represented in it (we have 30 “concept not found” results). On the other hand, for the same battery of queries, it was possible to retrieve, the desired information from the prototype representation using both a simple query rewriting/adaptation process and the *owl:sameAs* construct as a semantic bridge between the two representations (these two “linking elements” are represented by the black arrow in fig. 6.1). Thus, in this case, we had 30/30 of “concepts found results”. Since in E1 the frames representing prototypes are isolated (e.g., frame representing the prototype of LEMON is not connected with the

frame representing the prototype of ORANGE, and so on) we use a binary metric in order to evaluate the results based on the success/unsuccess of the retrieval. In this case, in fact, does not have sense to calculate precision and recall because the denominator of both measures would always assume a value between 0 and 1 (in E1, in fact, it is only possible a single concept retrieval).

In the second experiment E2, we inserted in the DL knowledge base properties considered as “typical” (e.g. the property of “being yellow”, was inserted in the DL representation of LEMON). These properties have been, of course, represented even in the typical component as “slots” of the frames. According to our analysis, this way of modelling concepts is not correct. However, it represents one of the most common approaches in the development of ontology based representations. In E2 we divided a first experiment in two control situations. In the first one (E2.1.1) each typical property represented in the DL component was only applied to one concept within the ontology (e.g. the typical property has colour: Yellow is applied only to the DL concept LEMON and not to other DL concepts). Moreover, the typical component has been still considered as composed by single, independent, frame based ontologies.

The figure 5.5 shows this particular control situation. In particular it shows the mechanisms activated by the query on the dual knowledge base. The mechanism is the following: the query is firstly executed on the DL knowledge base, where the concept C1 is retrieved, and then it is rewritten and reformulated on the typical component of C1. Due to the fact that the property Typ1 is represented also in the typical component of C1 (as Slot1), then the obtained answers from the two representation are identical and correspond to the same concept (C1, in the example).

The result obtained from the 30 queries executed in the control situation E2.1.1 consist in an exact match between the concept obtained by the DL and that ones obtained by the external knowledge base. This result put in evidence the fact that if the typical property inserted in the DL module has a unique counterpart with the corresponding slot of the prototype, therefore the obtained results from the two components cannot be different (they are necessarily the same).

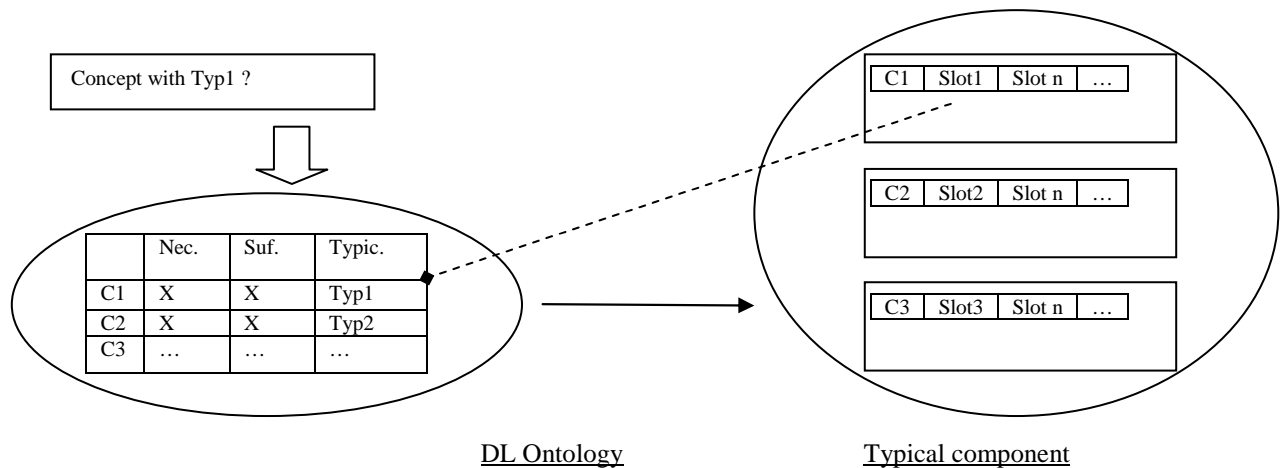


Figure 5.5 First control situation E2.1.1

Within the same experimental situation E2 we provided, modifying the initial knowledge base used in E2.1.1, a second control situation (E2.1a) in which the typical property Typ1 in DL (corresponding to Slot1 in the prototypes), resulted to be represented in different concepts within the typical component (e.g. in concept C1, C2 etc.). Despite this representational difference, even in the second control situation, we obtained, for all the 20 queries executed, the same result from both the compared representations. This thanks to the link between the two representations expressed via the *owl:sameAs* construct. This construct, in fact, allows to define that the concept C1 in DL is the same of C1 in the typical component, thus identifying the unique path that the query rewriting process have to follow in order to interrogate the typical component. Without this element the two knowledge bases could provide discordant results.

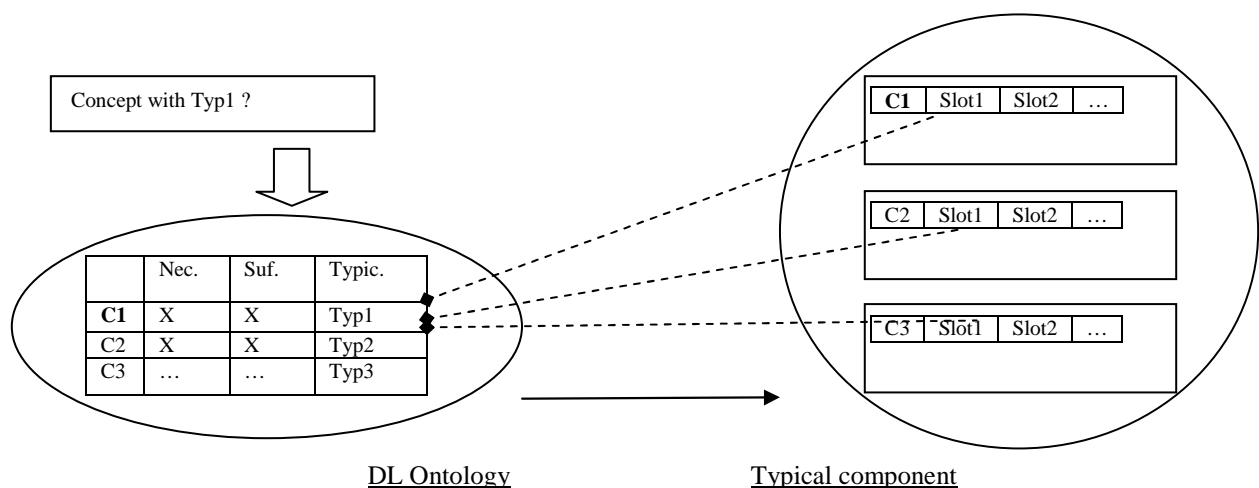


Figure 5.6. Second control situation E2.1a

After these first experiments, a new experimental situation (E2.2), has been individuated for the evaluation. In E2.2 the typical properties inserted into the DL ontology have been considered to be applied to multiple domains (e.g. the property *has_colour* : ORANGE has been applied to different concepts within the DL ontology such as, for example, BITTER ORANGE, ORANGE and so on).

More precisely: in E2.2 situation 5 typical properties have been considered as “multiple” and inserted into the DL knowledge base as belonging to 3 different concepts. These choice, even if arbitrary (one can imagine that the same property can be shared by more and more concept within a knowledge base), has been considered only in order enhance the manageability of the evaluation. Our aim, in fact, has been mainly finalized at discovering the different dynamics of the architecture behaviour when the same type of search stimulus (the typicality based query) is exposed to different representational situations. The new DL knowledge base obtained through the mentioned modifications is available at: <http://www.dualontologyarchitecture.net/ontologyE2.2.owl>. Thus, in E2.2 it has been possible to calculate Precision and Recall because the possible retrieved concepts can be > 1 (in other words it is possible to have, as answer, a list of possible results.). Even in this condition we maintained the assumption of the typical representations as isolated blocks.

A first control situation (E2.2.1) investigated for E2.2 is illustrated in the figure 5.7 below.

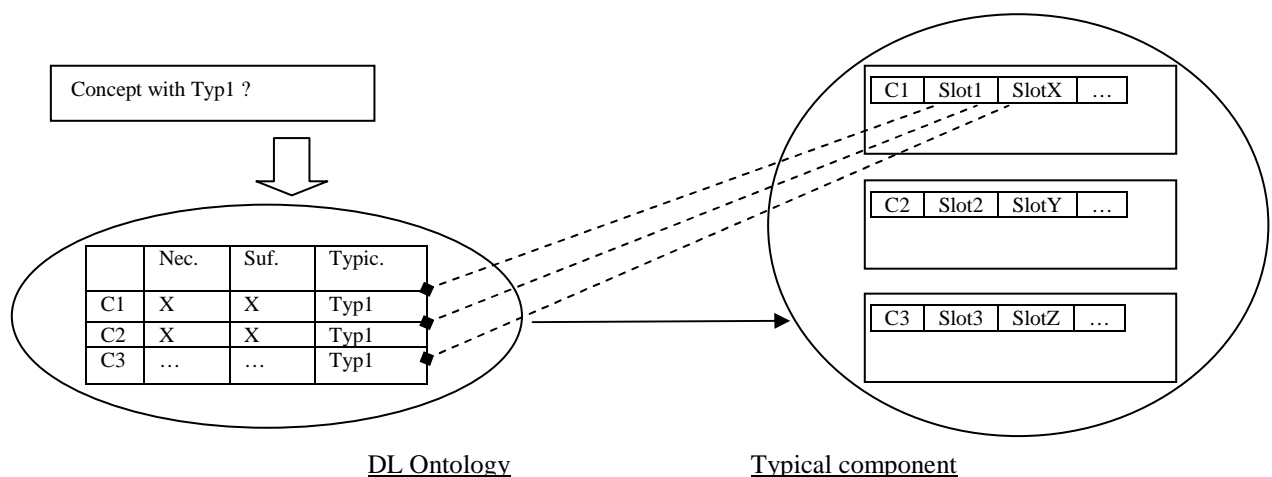


Figure 5.7. E2.2.1 First control situation

This situation is characterized by the fact that the same typical property (Typ1) is shared by a set of DL concepts (C1, C2 and C3 in figure) and is also represented as slot only in one of the external representations of the typical component. We tested this experimental situation and, with 20 queries and, for all the performed queries (20/20), we obtained a better precision and recall values through the use of the typical knowledge base (in these specific case the improvement has been of the 66 %).

For a second list of query we considered a different control situation (E2.2a) which has been illustrated in the figure 5.8 below.

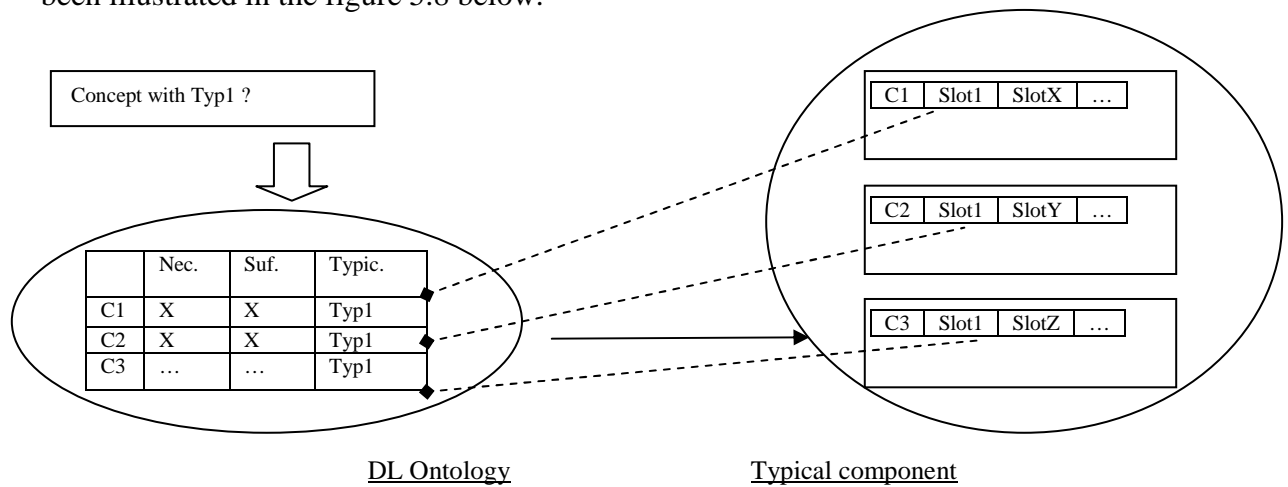


Figure 5.8 Second control situation E2.2a

The figure 5.8 shows that the same property (Typ1) results to be distributed in the DL ontology and even in the typical component of the architecture. In this case the result provided by the proposed architecture is uncertain. Or, better, it depends functionally from the first prototype considered after the query performed on the DL ontology.

In order to better explain the dynamics of the situation in E2.2a, we provide a simple example. Let's consider the case of the property "being orange" (e.g.: "has_colour : Orange"). In E2.2a ontology this information has been associated to 3 different concepts within the DL ontology: ORANGE, BITTER ORANGE and CLEMENTINE. Therefore for the query Q1 "find all the citrus that are orange" the query answer mechanism recover all the concepts in DL having this characteristics (for a total of 3 concepts retrieved). In this case the function of the typical representation is that one of refine the obtained results from the DL ontology. Of course the problem of this "refinement" is

that it depends functionally from the relative prototype considered. In fact, since the typical representations are supposed to be not connected in this situation, if the considered prototype is, for example, that one of CLEMENTINE (<http://www.dualontologyarchitecture.net/clementine.rdfs>), instead of – let suppose the prototype of the concept ORANGE – therefore the concept which results as “refined” would be that one of CLEMENTINE. While, in the opposite case, the refined result would be that one of ORANGE. We obtained this kind of results for all the 10 queries performed in this situation.

These answers are not satisfactory from our point of view. Because they do not allow to really have a better result in terms of the quality of the information retrieved. In case of uncertainty for specific queries, in fact, we believe that the solution provided by the DL representation (a list of possible results) can be considered a better choice if compared to the answers provided using also the typical component.

Finally we performed another set of experiment (E.3) in which the typical component has been considered as composed by a unique representation instead of multiple, federated, representation without any contact among them. The unified representation of all the typical categories created for the E3 is available at: <http://dualontologyarchitecture.net/framesystem.rdfs>. In this case we also considered the case of multiple shared properties in DL and in the typical representation. The figure 5.9 graphically shows the situation obtained in E3.

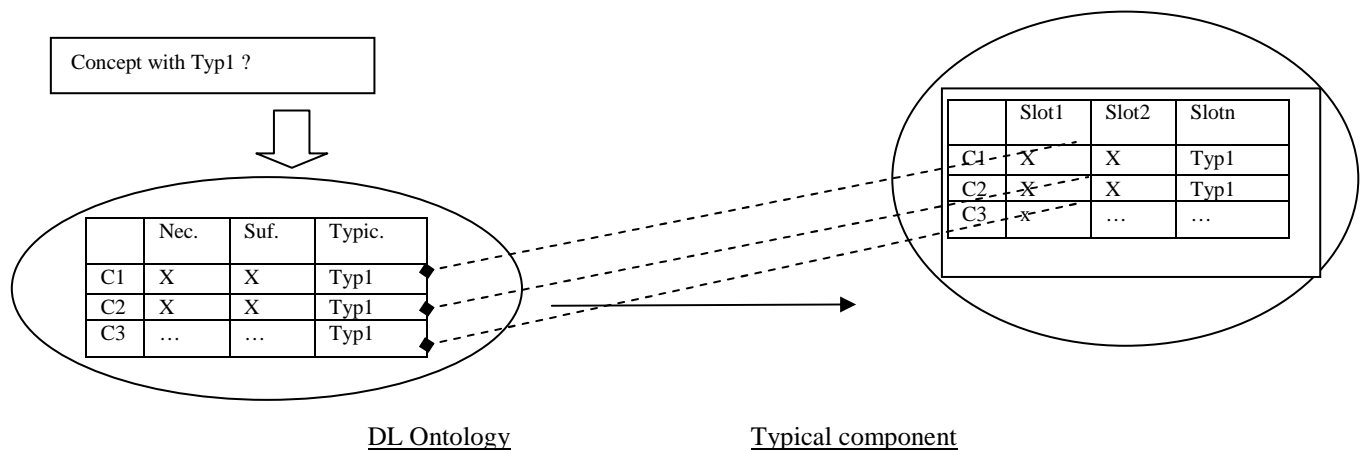


Figure 5.9. E3. Experimental situation

In the E3situation we obtained, for 30 of 30 queries performed on the DL knowledge base and then repeated using the typical component, the same results in both the situations. Even the values obtained for the precision and recall are the same. In other words the results are fully superimposable. We will comment and analyze in major detail this, and the other obtained results, in the next paragraph.

5.2. Evaluation Results in a nutshell

In the tables below we provide a synoptic summary of the results obtained for the different experimental situation illustrated. We do not report the specific number of the concept retrieved and or the obtained precision and recall results because, as we explained before, the main goal of our evaluation was to observe the behaviour of the proposed architecture for different search tasks. Furthermore, the numbers of the concept retrieved, and the relative percentage, could not have a valuable relevance because they have been obtained on a small knowledge base (modified, as indicated, for each experimental situation) with a limited battery of queries. For this reason we use, in the table below, some terms in order to indentify the different situations emerging from the evaluation.

Namely: we use the term “null” in order to identify the fact that the result obtained through the query corresponds to a situation in which there is not a concept/information retrieval; the term “full” to identify that, according to the battery of query launched on both the knowledge bases (and according to the representational situation on which this query has been run) we obtain a complete information/concept retrieval. Furthermore, we use the term “refined” in order to identify the result obtained by the first experimental situation in E2.2 in which the support coming from the typical knowledge base has been that one of refine the results obtained from the DL component, improving the performance of precision and recall measures and the semantic quality of the information extracted. Finally we use the term “limited” in order to identify the second situation encountered in E2.2a in which the use of typical component does not enhance the quality, and the trust, of the obtained results but only reduce the list of the concepts obtained. The problem here is represented by the fact that this reduction depends from

the order of interrogation of the typical representations of the concepts and this represent an evident limit for the quality of the obtained results.

	Concept retrieval in DL	Concept retrieval in Typical Component
E1	Null	Full
E2.1.1 (first situation)	full	Full
E2.1a (second situat.)	full	Full
E2.2.1 (first sit.)	full	Refined
E2.2a (second sit.)	full	Limited
E3	full	Full

Table 5.2. Concept Retrieval results

It is important to note that when both the representations are signed with the term full therefore it also means that the obtained results are identical (exactly the same concepts are retrieved). In the second table represented below we take into account the measures of precision and recall for each of the experimental situation individuated. Even in this case we do not consider the specific metrics obtained but we simply indicate if the different situations proposed present relevant differences for the indicated measures.

	Precision DL	Recall DL	Precision Dual KB	Recall Dual KB
E1	Not calculated	Not calculated	Not calculated	Not calculated
E2.1.1 (first situation)	Not calculated	Not calculated	Not calculated	Not calculated
E2.1a (second situat.)	Not calculated	Not calculated	Not calculated	Not calculated
E2.2.1 (first sit.)	Calculable	Calculable	Improved	Improved
E2.2a (second sit.)	Calculable	Calculable	Improved	Improved
E3	Calculable	Calculable	Same	Same

Table 5.3. Precision and Recall results. Synthetic table.

In the first experimental situations (E1 and E2.1) we do not calculate precision and recall measures because these tasks have been based on a single concept retrieval. Therefore the denominator of both measures would, unnaturally, have assumed only one of the value of 0 and 1.

In the control situation E2.2, instead, we obtained an improvement of both precision and recall measures guaranteed by the access to the typical component.

In the second situation E2.2a, we also noted an improvement of the precision and recall values. This fact depends from the structure of the situation described. In fact, in E.2.2, we always assist to the passage from a list of results to a unique concept obtained by the typical component. Improving, in this way, always the considered metrics but, as we have seen, not the quality of the information extracted.

Finally, in E3, we obtained the same values of precision and recall results for both the compared representational solutions. In the next paragraph the presented results are commented and discussed.

5.3. Discussion and Analysis

The results obtained through the evaluation shows some pros and cons of the proposed architecture in a real situation. As before mentioned, the main aim of our analysis has been that one of making a comparison between the proposed approach and the classical way of representing concepts in formal ontologies.

The general insight emerging from the obtained results seems to suggest, at a first glance, that in the major part of the experimental situation investigated there is not an improvement of the concept retrieval mechanisms.

More specifically: one could argue that, in E1, the contraposition among the obtained results (null vs full) is, in a certain sense, encoded within the representation itself. Continuing: it is evident that, in E2.1 (for both the control situations) and in E3, the results obtained by the two modeling approaches are the same (the concept retrieval gives the same results). Finally, the results obtained in the second subcondition individuated in E2.2 (e.g. E2.2a, see figure 6.5) can be considered better in the classical approach than in the proposed one.

In the following we try to provide arguments in order to interpret the emerging results in a different perspective. We suggest, in fact, that certain situations between the two approaches appearing as “neutral” are not neutral at all, and that, therefore, the proposed modeling approach for the representation of non classical concepts presents many advantages even when it express the same results of the classical one.

Going more into the details: in the case of E1, E2.1 and E3, we claim that the proposed approach represent an improvement with respect to classical one. The advantage is given

by the fact that the architectural solutions presented in these situations are much more closer to the real world scenarios (in which different levels of information for the same concept are stored in different data sources) than the classical approach (in which a single, monolithic, block of representation is demanded to represent all the needed information for a certain set of concepts). Thus, the fact of obtaining the same retrieval results in a more realistic scenario represent, in our opinion, a plus in favour of the proposed approach. In addition, we argue that another relevant advantage is given by the fact that the typical components of our architecture are expressed according to the Linked Data format. This means that they are usable and interrogable by other data sources, providing different modeling view for the same concept, improving the interoperability and the level of re-use of the knowledge bases. This issue, which is related to the aspects of knowledge and data integration, represent nowadays one of the main objective within the Semantic Web research community.

Furthermore, continuing our analysis, let consider the results coming from the first situation encountered in E2.2 (figure 6.4). In this case the proposed approach present a relevant improvement both on the side of the quality of retrieved information (there is an improvement of the recall value) and on that one of the quantity. The external typical component, in fact, has basically a pruning function and produce, from a list of possible answers, the result which can be considered cognitively more relevant. Thus improving the capabilities and the intelligent behavior of the system.

The last, and unique, condition in which the proposed architecture obtained results that can be interpreted as negative with respect to the classical one, is the second experimental situation encountered in E2.2a. In this case, in fact, even if the obtained structural values registered a better performance (because the precision and recall assumed enhanced values), the fact that the obtained results depend from the casual order of the considered prototypical concepts represent a minus from the point of view of the trust of the information retrieved.

As a possible extenuating circumstance we assume that this situation is oversized by our experimental evaluation. Despite this fact, however, this remain a negative situation for our approach.

Going deeper in our analysis it is important to point out some limits of the evaluation. A first one is represented by the fact that is has been not performed on existing well known

large ontological knowledge bases. The main reason of this situation is represented by the difficulty of recovering, from ontological search engines (such as Swoogle:) ontologies in <http://swoogle.umbc.edu/>) which represent, in our approach, the language to use for the DL representation. The reason of this lack of available OWL Full ontologies is given by the fact that all the ontologies shared and used in large semantic applications are, for computational complexity reasons, in OWL DL. However, we try to mitigate this cons providing, on the links mentioned in the experimental situations section, the knowledge bases on which we performed our experiments. Furthermore, the experimental conditions described both in E2.1 and E2.2 situations have been realized exactly to identify and to test our model with that the modeling approach mainly used in large ontological representations.

Chapter 6.

Conclusions

In this work we presented a cognitively inspired modelling approach aimed at facing the problem of non classical concept representation and reasoning in formal ontologies. Many other approaches have been developed in literature in order to face these problems; however, as presented in the chapter 2 (section 2.6), they pose various theoretical and practical difficulties, with many problems remaining unsolved.

The proposed modelling approach has been illustrated in the chapter 4. The main element characterizing this approach can be summarized as follows:

- (i) Division between the representations of the compositional and typical components
- (ii) Possibility of integrate these representations using the Linked Data approach and specific linking constructs provided by the ontological languages.
- (iii) Division of the type of reasoning processes operating on the interconnected knowledge base.

Our theoretical proposal has been partially implemented and evaluated. More properly: the implemented part has been that one regarding the representational modules of the architecture and the corresponding links between the two conceptual components.

This part has been evaluated on an information retrieval task concerning property checking based on prototypical information. The results obtained have been presented and discussed in the chapter 5. They seems to suggest that, in the major part of the situations, the proposed approach obtain the same or, in certain cases, enhanced results for the task considered for the evaluation. Furthermore, the proposed approach presents the advantage of presenting real world scenarios characterized by distributed

information systems and federated knowledge bases. However, some critical points are a matter of discussion and shall be further investigated and developed. The principal one is represented by the fact that the reasoning module able to perform certain forms of approximate reasoning (such as the non monotonic instance categorization) has been, at the current state of the art, theorized but not yet realized. So its implementation and testing represents a future work plan to be done in order to complete the evaluation. Despite that, however, we claim that the proposed approach presents relevant insights also about the reasoning processes. The presented hybrid approach, in fact, allows, at least in principle, to hypothesize the co-existence of different reasoning procedures (one classical and deductive and another one non monotonic logics) providing a cascade model able to avoid possible inconsistencies caused by eventual discordant results.

In this view, this work can be considered as an initial step aimed at the realization of a complete modeling framework including both the representational and reasoning aspects of typicality. The road traced seems to be encouraging but it still needs to be further investigated.

References

- Adams B., Raubal M., (2009). Conceptual Space Markup Language (CSML): Towards the Cognitive Semantic Web. In *IEEE International Conference on Semantic Computing*.
- Adams B., Raubal M., (2010). The Semantic Web needs more Cognition. In *Semantic Web*. Vol 1. 69-74.
- Ashby, F. G., Alfonso-Reese, L. A., Turken, A. U., & Waldron, E. M. (1998). A neuropsychological theory of multiple systems in category learning. *Psychological Review*, 10, 442-481.
- Ashby, F. G., & Ell, S. W., (2002). Single versus multiple systems of category learning: Reply to Nosofsky and Kruschke (2002). *Psychonomic Bulletin & Review*, 9, 175-180.
- Ashby, F. G., & W.T., (1990). Integrating information from separable psychological dimensions. *Journal of Experimental Psychology, Human Perception and Performance*, 16, 598-612.
- Baader F., Nutt W., (2002). Basic Description Logics. In *The Description Logic Handbook*. Cambridge University Press. Pp. 47-104.
- Baader F., Calvanese D., McGuinness D., Nardi D., Patel-Schneider P, (2002). *The description logic handbook: theory, implementation, and applications*, Cambridge University Press, New York, NY.
- Baader, F., Hollunder, B., (1995). Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning* 14(1), 149--180.
- Barsalou, L.W., (2005). Continuity of the conceptual system across species. *Trends in Cognitive Science* 9(7), 305-311
- Bermudez, J. L., (1995). Nonconceptual content: From perceptual experience to subpersonal computational states. *Mind and Language*, 10, 333–369.
- Bermudez, J. L. and Cahen, A., (2011). Nonconceptual mental content. *Stanford Encyclopaedia of Philosophy*, <http://plato.stanford.edu/>
- Berners Lee T, Lassila H, Hendler J. (2001). “*The Semantic Web*”. *Scientific American* 284(5): 34-43.
- Berretty, P. M., Todd, P. M., & Martignon, L. (1999). Categorization by elimination: Using few cues to choose. In G. Gigerenzer, P. M. Todd, & the ABC Research Group.,

- Simple heuristics that make us smart* (pp. 235-254). New York: Oxford University Press.
- Bizer, C., Heath, T., Berners-Lee, T., (2009). Linked Data - The Story So Far, *International Journal on Semantic Web and Information System*, 5 (3), pp. 1-22.
- Bobillo F., Straccia U., An OWL Ontology for Fuzzy OWL 2 (2009), *Proceedings of the 18th International Symposium on Foundations of Intelligent Systems*, Prague, Czech Republic.
- Bonatti, P. A., Lutz, C., Wolter, F., (2006). Description logics with circumscription. *Proc. of KR*, pp. 400-410.
- Brachman R., Hector J. Levesque H.J., (1985). Readings in Knowledge Representation, Morgan Kaufmann Publishers Inc., San Francisco, CA.
- Brachman, R., (1985). I lied about the trees. *The AI Magazine*, 3(6), 80--95.
- Brachman, R., Schmolze, J. G., (1985). An overview of the KLONE knowledge representation system, *Cognitive Science*, 9, 171--216.
- Brandom, R., (1994). *Making it Explicit*. Cambridge, MA: Harvard University Press.
- Calegari S., Ciucci D., Fuzzy Ontology, (2007). Fuzzy Description Logics and Fuzzy-OWL, *Proceedings of the 7th international workshop on Fuzzy Logic and Applications: Applications of Fuzzy Sets Theory*, Camogli, Italy.
- Castano S. , Ferrara S., Montanelli S. (2003). H-Match: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems. *Proceedings of the First Workshop on Semantic Web and Databases (SWDB-03)*.
- Castano S. , Ferrara S., Montanelli S. (2005). Ontology-based Interoperability Services for Semantic Collaboration in Open Networked Systems. In *Proc. of the 1st Int. Conference on Interoperability of Enterprise Software and Applications*.
- Da Costa P.C., Laskey K., (2006). PR-OWL: A Framework for Probabilistic Ontologies, *Proceedings of the 2006 conference on Formal Ontology in Information Systems: (FOIS 2006)*, p.237-249.
- De Virgilio R., Giunchiglia F., Tanca L. (2010) *Semantic Web Information Management*, Springer Verlag. Berlin.
- Dell'Anna A., Frixione M., On the Advantage (If Any) and Disadvantage of the Conceptual/Nonconceptual Distinction for Cognitive Science, *Minds and Machines*, v.20 n.1, p.29-45, February 2010 .

- Ding, Z., Peng, Y., Pan, R.. 2006. BayesOWL: Uncertainty modeling in Semantic Web ontologies. In Z. Ma (ed.), *Soft Computing in Ontologies and Semantic Web*, vol. 204 of *Studies in Fuzziness and Soft Computing*, Springer.
- Donini F. M. , M. Lenzerini , D. Nardi , W. Nutt , A. Schaerf, (1998). An epistemic operator for description logics, *Artificial Intelligence*, v.100 n.1-2, p.225-274.
- Donini F., Nardi D., Rosati R., (2002). Description logics of minimal knowledge and negation as failure, *ACM Transactions on Computational Logic (TOCL)*, v.3 n.2, p.177-225.
- Dopkins S., Gleason T. (1998). Comparing exemplar and prototype models of categorization. *Canadian Journal of Experimental Psychology*, Vol 51(3), 212-230.
- Evans, J. S. B. T., Frankish, K. (eds.), 2008. *In Two Minds: Dual Processes and Beyond*. New York, NY: Oxford UP.
- Fodor, J., 1981. The present status of the innateness controversy. In J. Fodor, *Representations*, Cambridge, MA: MIT Press.
- Fodor, J., 1987, *Psychosemantics*, Cambridge, MA: The MIT Press/A Bradford Book.
- Fodor, J., 1998. *Concepts: Where Cognitive Science Went Wrong*, Oxford, UK: Oxford University Press.
- Fodor, J., Pylyshyn, Z., 1988. Connectionism and cognitive architecture: A critical analysis.
- Frixione, M., Lieto A., 2011. Representing and Reasoning on Typicality in Formal Ontologies. *Proceedings of the 7th International Conference on Semantic Systems*, Sept. 7-9, 2011, Graz, Austria. ACM Internat. Conference Proceedings Series. New York.
- Gagliardi, F. (2010). Cognitive Models of Typicality in Categorization with Instance-Based Machine Learning. In: *Practices of Cognition. Recent Researches in Cognitive Science*, University of Trento Press. pp. 115-130.
- Gagliardi, F. 2008. A Prototype-Exemplars Hybrid Cognitive Model of "Phenomenon of Typicality" in Categorization: A Case Study in Biological Classification. in *Proc. 30th Annual Conf. of the Cognitive Science Society*, Austin, TX, 1176--1181.
- Gagliardi, F. 2010. Cognitive Models of Typicality in Categorization with Instance-Based Machine Learning. In *Practices of Cognition: Recent Researches in Cognitive Sciences*, pp 115--130, University of Trento Press.
- Gao M., Liu C., Extending OWL by Fuzzy Description Logic, *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence*, p.562-567, November 14-16, 2005.

- Gardenfors. P. (2000). *Conceptual Spaces: The Geometry of Thought*. The MIT Press.
- Gardenfors. P. (2004). How to Make the Semantic Web More Semantic. In *Formal Ontology in Information Systems, Proceedings of the Third International Conference (FOIS 2004)*, pages 17-34.
- Giordano L., Gliozzi V., Olivetti N., Pozzato G, (2007). Preferential description logics, *Proceedings of the 14th international conference on Logic for programming, artificial intelligence and reasoning*, p.257-272, October 15-19, 2007, Yerevan, Armenia.
- Giordano L., Gliozzi V., Olivetti N., Pozzato G, (2009). ALC + T: a Preferential Extension of Description Logics, *Fundamenta Informaticae*, v.96 n.3, p.341-372.
- Giordano L., Gliozzi V., Olivetti N., Pozzato G., (2008). Reasoning about Typicality in Preferential Description Logics, *Proceedings of the 11th European conference on Logics in Artificial Intelligence*, September 28-October 01, 2008, Dresden.
- Giunchiglia F., Farazi F., Tanca L, De Virgilio R. (2010). *The Semantic Web Languages. Semantic Web Information Management*, Springer Verlag. Berlin.
- Hall M. (2006)., *A Semantic Similarity Measure for Formal Ontologies*. Master Thesis. Alpen-Adria Universitat Klagenfurt.
- Hampton J.A. (1993). Prototype models of concept representation. In I. van Mechelen, J. A. Hampton, R. S. Michalski, & P. Theuns (Eds.), *Categories and concepts: Theoretical views and inductive data analysis* (pp. 67–95). London: Academic Press
- Hampton J.A. 1995. Similarity based Categorization: The Development of the Prototype Theory. *Psychologica Belgica*. 35-2/3, 103-125.
- Hayes, P., 2001. Dialogue on rdf-logic. Why must the web be monotonic? *World Wide Web Consortium (W3C)*. Link: <http://lists.w3.org/Archives/public/www-rdf-logic/2001Jul/0067.html>
- Horrocks I, McGuinness D., Patel-Schneider P., Welty C, 2003. OWL: a Description Logic Based Ontology Language for the Semantic Web, in *Description Logic Handbook*, Cambridge University Press. 458-486.
- Horrocks Ian, Peter F. Patel-Schneider, and Frank van Harmelen, 2003. From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *J. of Web Semantics*, 1(1):7-26.
- Jansen B., Danielle L. Booth , Amanda Spink, Determining the informational, navigational, and transactional intent of Web queries, *Information Processing and Management: an International Journal*, v.44 n.3, p.1251-1266, May, 2008

- Jones, G. V. (1983). Identifying basic categories. *Psychological Bulletin*. 94, 423-428
- Katz, Y., Parsia, B., 2005. Towards a non monotonic extension to OWL. *Proc. OWL Experiences and Directions*, Galway, November 11-12.
- Klien E., Einspanier U., Lutz M., Hubner S.(2004). An Architecture for Ontology-Based Discovery and Retrieval of Geographic Information. *In Proceedings of the 7th Conference on Geographic Information Science*.
- Klinov, P., Parsia, B., 2008. Optimization and evaluation of reasoning in probabilistic description logic: Towards a systematic approach. *In Proceedings of ISWC 2008*.
- Lalumera, E. (2009). Cosa sono i concetti. Roma-Bari : Laterza.
- Lemmens R., Arenas H. (2004). Semantic Matchmaking in Geo Service Chains: Reasoning with a Location Ontology. *In Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04)*, pages 797 - 802.
- Leon F. And Galea D., (2007). Rules Prototypes and Exemplars – A survey on Categorization Techniques. *Proceedings of the 9th International Symposium on Automatic Control and Computer science*, Iasi, Romania.
- Li L., and I. Horrocks I. A Software Framework for Matchmaking Based on Semantic Web Technology. *In International Journal of Electronic Commerce*, 8(4): pages 39-60, 2004.
- Lukasiewicz T., Straccia U., Managing uncertainty and vagueness in description logics for the Semantic Web, *Web Semantics: Science, Services and Agents on the World Wide Web*, v.6 n.4, p.291-308, November, 2008.
- Machery, E., 2005. Concepts are not a natural kind. *Philosophy of Science*, 72, 444–467.
- Machery, E., 2009. *Doing without Concepts*. Oxford, UK: Oxford University Press.
- Maedche A., Staab S. (2002). Measuring Similarity between Ontologies. *In Lecture Notes in Computer Science*, 2473: page 251.
- Malt, B.C. (1989). An on-line investigation of prototype and exemplar strategies in classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15(4), 539-555.
- Medin, D.L. and Schaffer, M.M. (1978). Context theory of classification learning, *Psychological Review*, 85(3), 207-238

- Medin, D.L. and Schwanenflugel, P.J. (1981). Linear separability in classification learning, *Journal of Experimental Psychology: Human Learning and Memory*, 7(5), 355-368.
- Minsky, M., 1975. A framework for representing knowledge, in Patrick Winston (eds.), *The Psychology of Computer Vision*, New York, McGraw-Hill. Also in Brachman & Levesque (2005).
- Motik B., Horrocks I., Rosati R., Sattler U. (2006). *Can OWL and Logic Programming Live Together Happily Ever After?*. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *Proc. of the 5th Int. Semantic Web Conference (ISWC 2006)*, volume 4273 of LNCS, pages 501–514, Athens, GA, USA, November 5–9 2006. Springer.
- Murphy, G. L., 2002. *The Big Book of Concepts*. Cambridge, MA: The MIT Press.
- Nosofsky, R.M. 1992. Exemplar-based approach to relating categorization, identification and recognition. In F. G. Ashby (Ed.), *Multidimensional models of perception and cognition*. Scientific psychology series (pp. 363-393). Hillsdale, NJ, England: Lawrence Erlbaum Associates, Inc.
- Nosofsky, R. M., Palmeri, T. J., & McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychological Review*, 101 (1), 53-79.
- Nosofsky, R. M. (1986). Attention, similarity, and the identification categorization relationship. *Journal of Experimental Psychology*, 115, 39-57.
- Osherson, D. N., Smith, E. E., 1981. On the adequacy of prototype theory as a theory of concepts. *Cognition*, 11, 237--262.
- Palmeri, T.J., & Gauthier, I. (2004). Visual object understanding. *Nature Reviews Neuroscience*, 5, 291-303.
- Peacocke, C., 1992. *A Study of Concepts*. Cambridge, MA: The MIT Press. Rosch, E., 1975. Cognitive representation of semantic categories. *Journal of Experimental Psychology*, 104, 573-605.
- Pinker, S., Prince, A. (1996), The nature of human concepts: evidence from an unusual source. *Communication and Cognition*, 29, 307-361.
- Quillian, M.R. (1968). Semantic memory, in M. Minsky (ed.), *Semantic Information Processing*, MIT Press, Cambridge, MA.
- Rada R., H. Mili, E. Bicknell and M. Blettner. Development and application of a metric on semantic nets. *In IEEE transactions on systems, man and cybernetics*, 19(1): pages 17-30, 1989.

- Resnik P. (1999). Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal Of Artificial Intelligence Research*, Volume 11, pages 95-130.
- Rosch, E. (1975). Cognitive representation of semantic categories. *Journal of Experimental Psychology* 104: 573-605.
- Russell S.J., Norvig P. (2002) *Artificial Intelligence. A Modern Approach (2nd Ed.)* Prentice Hall, Englewood. Cliffs, NJ.
- Schmidt-Schauß and Smolka, 1991. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1-26.
- Smith, E. E. & Medin, D. L. (1981). Categories and concepts. Cambridge, MA: Harvard University Press.
- Smith, J. D., & Minda, J. P. (1998). Prototypes in the mist: The early epochs of category learning. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 24, 1411-1436.
- Smith, J. D., & Minda, J. P. (2000). Thirty categorization results in search of a model. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 26, 3-27.
- Smith, J. D., Murray, M. J., & Minda, J. P. (1997). Straight talk about linear separability. *Journal of Experimental Psychology: Learning, Memory, and Cognition*. 23, 659-68.
- Smith, J. D., & Minda, J. P. (2002). Distinguishing prototype-based and exemplar-based processes in dot-pattern category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(4), 800-811.
- Spelke, E.S., 1994. Initial knowledge: six suggestions. *Cognition*, 50, 431-445.
- Spelke, E.S., Kinzler, K.D., 2007. Core knowledge. *Developmental Science*, 10(1), 89–96.
- Stanovich, K. E., West, R., 2000. Individual Differences in Reasoning: Implications for the Rationality Debate? *The Behavioural and Brain Sciences* 23 (5), 645--665.
- Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J. Z., Horrocks, I., 2005. Fuzzy OWL: Uncertainty and the Semantic Web. *Proc. Workshop on OWL: Experience and Directions (OWLED 2005)*. CEUR Workshop Proceedings, vol. 188.
- Straccia U., Default inheritance reasoning in hybrid KL-ONE-style logics, *Proceedings of the 13th IJCAI*, p.676-681, August 28-September 03, 1993, Chambery, France.
- Tversky A. Features of Similarity. In *Psychological Review*, 84(4): pages 327-352, 1977.

Williams, A. Padmanabhan and M. B. Blake, (2003). Local Consensus Ontologies for B2B-Oriented Service Composition. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 647-654.

Witten, I.H., Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. (2nd Ed.)* Morgan Kaufmann, San Francisco, CA.

Wittgenstein, L. (1953). *Philosophische Untersuchungen*. Oxford, Blackwell.

