UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE

ICT International Doctoral School

A TWO-LAYERED KNOWLEDGE ARCHITECTURE FOR PERCEPTUAL AND LINGUISTIC KNOWLEDGE

Mattia Fumagalli

Advisor

Prof. Fausto Giunchiglia

Università degli Studi di Trento

November 2017

Abstract

The lack of generality is a structural weakness of knowledge representation formalisms. Here by lack of generality we mean the inability of any given representation to describe the infinite richness and diversity of the world and also its potentially infinite descriptions which are enabled by language. This lack of generality is the main cause of many of the difficulties encountered so far, just think of the problems which have arisen in the effort of creating reusable ontologies. In this thesis we propose a solution to the problem of generality which is based on the key idea that knowledge should not be modeled a priori, at design time, but it should continuously generated, adapted and evolved, from generation to usage. The thesis provides four main contributions: (i) a shared terminology for the characterization of concepts and for their computational representation; (ii) a formalization of the distinction between substance concepts and classification concepts; (iii) the integration of these two notions of concept into a general representation language that organizes them into a hierarchy of increasing abstraction of what is perceived, and (iv) a two-layered knowledge representation formalism, where the first layer allows to represent concepts, as the main devices for achieving generality, and where the second layer allows to represent concepts as the result of "adapting" a description to the current knowledge representation needs and requirements.

Keywords

[Artificial Intelligence, Conceptual Modeling, Knowledge Representation, Meaning Computation, Philosophy of Mind]

Acknowledgements[*]

First I would like to express my special appreciation and thanks to my advisor Professor Fausto Giunchiglia, you have been a great mentor for me. I would like to thank you for your trust, for encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on my career have been priceless. I would especially like to thank my colleagues from the KnowDive group. All of you have been there to support me with good advice and interesting discussions. You have contributed significantly to my personal and professional growth. A thought to all my friends here in Trento and Milano who supported me in writing, and pushed me to strive towards my goal. A special thanks to my family for all their love and encouragement. Words cannot express how grateful I am to my three pillars, my wife and my two kids. To my beloved wife Ileana, thank you for all of the sacrifices that you've made on my behalf, the sleepless nights, the faithful support in the most difficult moments. Thank you to keep me believing. To my beloved kids, Samuele and Pietro, thank you for all the carefree and funny moments. Your energy and your joy helped me to remember that for doing great things it is also important to keep the focus on fun.

Mattia Fumagalli
University of Trento
November 2017

[*] This work has been supported by QROWD (http://qrowd-project.eu), a Horizon 2020 project, under Grant Agreement No. 732194.

Previously published material and material to be submitted

The chapters of this thesis host texts and ideas from papers that are already published during the PhD or that are going to be published next year (2018):

- We plan to submit a refined version of Chapter 2 to the Artificial Intelligence Journal (AI) (https://www.journals.elsevier.com/artificial-intelligence/), as *a survey of computational approaches to concepts representations*.
- Chapter 3 is derived from: Giunchiglia, Fausto, and Mattia Fumagalli. Concepts as (Recognition) Abilities. In *FOIS*, pp. 153-166. 2016. http://ebooks.iospress.nl/volumearticle/44243
- Chapter 4 is derived from: Giunchiglia, Fausto, and Mattia Fumagalli. Teleologies: Objects, Actions and Functions. In *International Conference on Conceptual Modeling*, pp. 520-534. Springer, Cham, 2017. http://www.springer.com/it/book/9783319699035
- We plan to submit a refined version of Chapter 5 to the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018) (https://www.ijcai-18.org/), as *a teleological approach for tracing and exploiting knowledge diversity*.

Contents

Part I

General Notions

Chapter 1

1. Introduction

1.1. The Context

The key intuition underlying the work in knowledge representation (KR) is that the mind of an Artificial Intelligence (AI) can be modeled as a *representational system* that can be exploited for processing information. Such a system is usually seen as a set of beliefs which construct a *mental model* of the world [**100**]. Many KR formalisms have been devised under this assumption. This work has gone a long way with many success stories. Still, it has soon turned out that all of these formalisms suffer from a lack of *generality* [**66**], [**13**]. The lack of generality is made evident by *(i)* the failure in reducing diverse representations of the world to a single "universal" theory, and *(ii)* the difficulties faced in the creation of representations of the world that can address a changing world.

1.2. The Problem

The problem of (non-)generality is unavoidable and it is entangled with two fundamental issues that must be handled by any representation, namely *world diversity* and *representation use* [**44**]. It is simply impossible to construct a *finite* representation capable of capturing the *infinite richness and diversity of the world* and also the *infinitely many possible descriptions of the world* which are enabled by language. On the one hand, any fixed representation cannot manage the diverse and multiple inputs coming from the external environment. Any new *encounter* with the world may hide some details and highlight others; for any chosen representation, there will always be some aspect of the world that is not captured. On the other hand, any fixed representation depends always on a certain perspective and any change in the goals to be addressed may cause a revision of the current description. For any chosen representation there will always be an alternative way, not yet considered, to represent the same aspect of the world.

The problem of lack of generality has had a huge (negative) impact on AI. So far, the KR problem has been dealt with in isolation, as if knowledge could be modeled "from first principles" independently of the world generating it and of its intended use. The main consequence is that all the attempts to deal with semantic heterogeneity, for instance all the work on (reusable) ontologies and data integration (see, e.g. [**41**], [**12**]) have obtained only partial successes.

1.3. The Solution

In this thesis we propose a new approach to KR which allows to address the problem of generality and which is based on the key idea that knowledge should not be modeled a priori, at design time, but, rather, that it should continuously be generated, adapted and evolved, from generation to usage. The life cycle of the management of knowledge should be constructed as the result of the following two steps:

1. *knowledge acquisition*, where the world input is acquired and stored into an adaptable and extensible KR formalism. This step allows to deal with the diversity of the world;
2. *knowledge (re)use*, where the acquired knowledge is used to generate a fixed world representation which is adapted as a function of the requirements and goals. This step allows to deal with the many possible representations of the world.

Thus, for instance, I can store information about cats as a function of my encounters with them and, in output, I can generate any desired representation of cats which is within the scope of what I know about cats. This thesis provides four main contributions:

1. a shared terminology for the characterization of concepts and for their computational representation;
2. a formalization of the distinction between *substance concepts* and *classification concepts*;
3. the integration of the two above mentioned notions of concept into a general representation language that organizes them into a hierarchy of increasing abstraction of what is perceived;

4. a two-layered KR formalism where the first layer allows to represent *concepts*, as the main devices for achieving generality and where the second layer allows to represent *concepts* as the result of “adapting” descriptions to the current KR needs and requirements.

Our approach, and in particular our proposed architecture is strongly influenced by the work on *biosemantics* (also known as *teleosemantics*) [**62**]. Many of the examples and terminology metaphors used in the following are derived from this field. Biosemantics provides an account of how representations carry meaning by appealing to the teleological notion of *function*. Here the notion of function maps into the one used in the context of neurobiology when attributing functions to components of the brain (as in “the function of processing visual information”). Most relevant to us is Millikan’s account of biosemantics and her explanation of how representations are generated in terms of *consumer*’s and *producer*’s abilities [**74**], [**76**], [**73**], [**69**], [**93**].

1.4. Structure of the Thesis

The rest of the thesis is structured as follows:

- Chapter 2 introduces a survey of computational approaches to concepts representations, with the main goal to classify the heterogeneous computational approaches according the provided terminology, and to provide a reader who may not be very familiar with theories of concepts with introduction to major themes in this research and with pointers to different research projects.
- Chapter 3 presents the central distinction between classification and substance concepts, providing a model of concepts as abilities, with a focus on recognition abilities, and an early version of an Ontology of (Recognition) Abilities (called RAO).
- Chapter 4, starting from the distinction between substance and classification concepts, provides an early proposal for an integrated architecture enabling perception and reasoning. The goal here is to go a step further and to integrate these

two notions into a general theory of concepts which organizes them into a hierarchy of increasing abstraction of what is perceived.

- Chapter 5 describes how the proposed integrated architecture can be used for addressing the problem of generality and the puzzle of sameness. The main goal here is to show how our new approach is a promising solution in supporting the current existing knowledge integration methodologies.
- Finally, Chapter 6 presents the conclusions and future work, respectively.

Chapter 2

2. A Survey of Computational Approaches to Concepts Representation

The key assumption underlying the work in conceptual modeling is that different kinds of conceptual representations are needed in order to account for certain classes of cognitive phenomena [**1**]. Within the field of Artificial Intelligence (AI), many cognitive architectures have been realized adopting different approaches for the organization and the representation of their *conceptual system* [**107**]. A huge work for formalizing, analyzing and depicting the cognitive and ontological principles that ground conceptual modeling processes have been addressed in [**50**] and [**47**]. The formalization of new tools, such as the perceptual symbol system approach [**4**] and the proxytype theory [**86**], gathered from different theories of concepts, has been put forward in [**60**] and [**85**]. Statistical approaches, such as neural nets, implementing dynamic and situated conceptual representations have been exploited (e.g., [**67**]). Computational accounts of approaches (e.g., simulation/embodied approaches) that ground conceptual information in modality-specific systems have been provided (e.g., [**92**]).

It can be generally observed that, so far, all these different representations of concepts have gone a long way with many success stories. Anyhow, none of them can account for all aspects of cognition. Some models, for instance, are used for enabling systems to reason on enormous amounts of data, but fail in accounting for trivial common-sense reasoning [**28**]. Similarly, some conceptual representations are impressively successful when used in well-defined domains, but they are completely inefficient in cross-domains settings [**97**]. Based on this evidence, the main consideration is that artificial systems can take advantage of all these different conceptual representations for addressing different tasks. The key issue becomes then their combination into a unified view. So far, some hybrid approaches have been proposed [**105**]. Here the main goal is to take (some of) the existing representations, to adapt them and to integrate them in a hybrid conceptual model.

However, these approaches are only, partially satisfying, *ad hoc* solutions, and cannot overcome some serious integration problems. The different conceptual representations are, indeed, in many cases, incompatible: (they start from different modelling assumptions and theories of concepts, which more often are left implicit, they adopt different modeling constructs, and so forth) [**101**], [**34**].

The focus of modern AI on concepts and their representations makes the understanding of the notion of 'concept', and the knowledge of the core of conceptual theories, a key factor in this area of research. Making explicit the modelling assumptions behind the different approaches is, indeed, an important issue to be addressed whenever, for instance, a conceptual representation has to be devised and compared, or integrated, to other conceptual representations. This chapter provides a brief survey of the computational approaches to the representation of concepts. The main goals are:

1. to provide a shared terminology for the characterization of concepts and their computational representation;
2. to classify the heterogeneous computational approaches according the provided terminology, and
3. to provide a reader who may not be very familiar with theories of concepts with introduction to major themes in this research and with pointers to different research projects.

Note that this chapter does not attempt to provide a comprehensive review of the state of the art in the representation of concepts. We refer the reader to excellent and thorough reviews, such as [**77**] or [**18**], for that purpose. Our central aim is, indeed, to examine, in the light of the existing theories of concepts, just some of the most relevant approaches, in order to make explicit their modelling assumptions and linking them to a common terminological (and theoretical) ground. The final outcome of this survey could be then used in order to discuss: *a)* criteria for finding similarities/dissimilarities between different ways of modelling concepts and *b)* criteria for devising new other possible integrated conceptual representations.

The chapter is organized as follows: Section 2.1 groups the current approaches to concepts representation into three main classes. Section 2.2 provides a list of dimensions, through which concept representations can be compared and divided. In Section 2.3, 2.4 and 2.5 we provide a brief description of some remarkable computational implementations. Section 2.6 provides a brief overview and comparison of the described approaches.

2.1. Three Broad Classes of Theories

Different theories about the nature of concepts have been proposed in cognitive science, neuroscience and philosophy of mind. Most of these theories are grouped according to the literature into two main classes: *Good Old Fashioned Artificial Intelligence* (GOFAI) theories and *New Fangled Artificial Intelligence* (NFAI), or *post-classical*, theories [**6**]. Along with the GOFAI and NFAI theories, there are theories combining assumptions that ground both GOFAI and NFAI theories. We call these theories *Complementary Fangled Artificial Intelligence* (CFAI) theories.

GOFAI. These theories are also known as *classical-symbolic* theories and provide perhaps one of the best known and most widely accepted view of concepts. According to this view, concepts are explicit representations codified in a language, similar to the first-order predicate calculus. The main features of these type of representations, also called *propositional* [**83**], are *arbitrariness* and *discreteness*. Concepts can be seen indeed as symbols of the language of thought (LOT) [**32**]. They are arbitrary in the sense that the similarity between them and what is represented is not needed. They are discrete because they are complex expressions separable in smaller parts, or they are atomic parts without any internal structure. Arbitrariness and discreteness allow the propositional representations of concepts to be highly formal and abstract.

NFAI. These theories have been developed in recent years and are also known as *post-classical* theories. The NFAI class can be divided into two main sub-classes: the *situated robotics* theories and the *connectionist* theories. The situated robotics program still need to be consolidated and cannot be considered as genuine theory, however it is being tested and used in many AI researches and applications (e.g., dynamical systems). Differently from the situated robotics, the connectionist research program has a long story and dates backs to the 40's [**68**], [**53**]. The many success stories of the symbolic approach around the 50's and 60's put connectionism in the shade for a long period. However, in the late 80's it began to increase again its popularity. Connectionism shares the computational hypothesis of the symbolic approach, but providing a different model for concepts. In particular, according to this view, concepts can be seen as *representations distributed throughout a large number of processing elements* [**34**]. Concepts are embedded in a *network* composed by *interconnected units*, which, at a certain level of abstraction, simulate the behavior of a conglomerate of neural cells. So far, even if they cannot be considered as a proper models of real neural systems, different types of (*artificial*) *neural networks* have been successfully adopted for addressing specific AI tasks.

CFAI. These theories are not in contrast with the GOFAI and NFAI programs. What we call CFAI theories rely, indeed, on assumptions that may be shared by both the previously described classes of theories. They can be seen as complementary views introduced in order to model aspects of cognition that are difficult to be modeled with GOFAI and NFAI frames only. Under the category of CFAI theories we group the *procedural* theories, the *analogical* theories, and the *prototype-exemplar-theory* (PET) theories. Procedural theories raised during the 70s and their slogan says that is not necessary for a concept to be explicitly represented as a mental symbol [**55**]. According to procedural theories concepts can be implicitly represented as a "procedure", i.e., as the execution of a piece of an algorithm. According to this framework, having a concept is having a capability to do something. For instance, having the concept of 'Cat' is having the ability of recognizing something as a 'Cat' or having the capability of using it in inference processes (i.e., it is an animal). Similarly, the analogical theories, around the late 60s, introduced another new interpretation of concepts. According to these theories, concepts are *analogical* (and not propositional, like in the classical-symbolic theories) *representations*. These kinds of representation are defined as mental objects that are similar to the objects they represent, like, for instance a picture of a cat or the image of a cat on my eye retina [**96**]. Differently from the propositional concepts introduced by GOFAI, the analogical concepts are not claimed to be discrete. This means that concepts do not provide a selection of features and the whole perceptual information are collected (this is a value for concreteness and completeness but, for instance, is a problem for compositionality and abstraction. For further information, see sections below). Another interesting issue is that, with their representation of concepts, analogical theories provide an account for simulation (see for instance proxy-types) processes in cognition [**4**], [**86**]. Among the CFAI theories we have what we call PET theories. Here we group three kinds approaches to concepts representation that are very similar, i.e., the prototypical approach, the exemplar approach and the theory approach. According to the prototypical approach, concepts provide the representation of the "most typical" occurrence for a given perceived object. Concepts are prototypes, i.e., a sort of weighted set of features (e.g., the prototype for 'Apple' is something round, green, red or yellow, with a specific range of weight, and so forth). In the exemplar view concepts can be seen as devices storing information about specific example occurrences for a given perceived object (e.g., the information about the apples we encountered in our experience). Within the theory approaches concepts are represented as (micro-)theories. For instance, having a concept for 'Apple' means having (micro-)theory about apples.

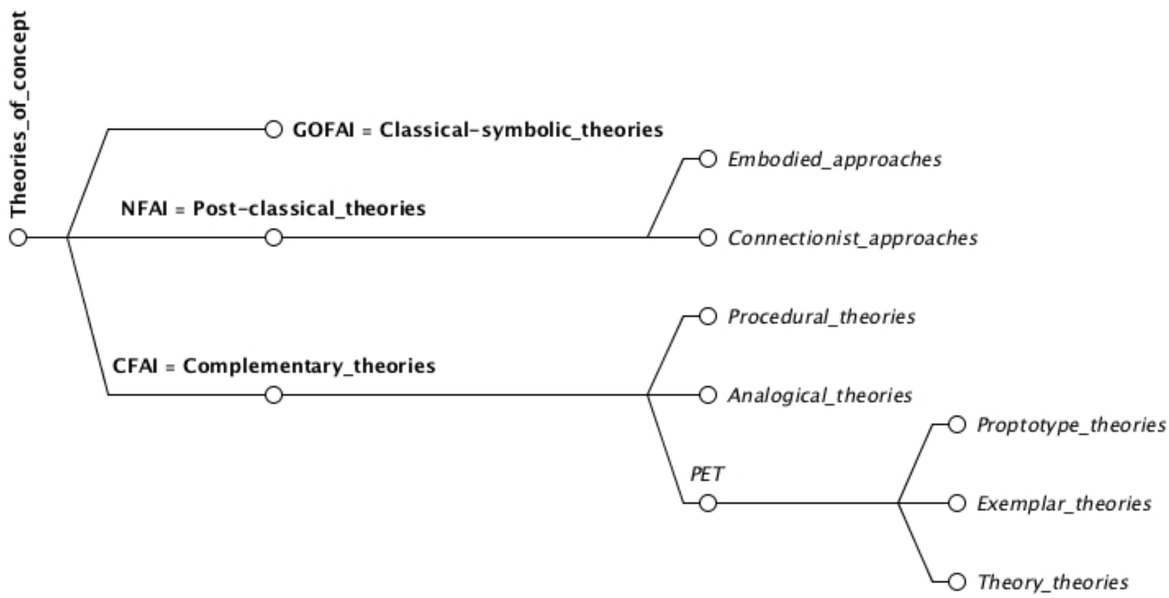

Fig. 1. Theories of concepts: a classification.

2.2. A characterization of Conceptual Representations

GOFAI, NFAI and CFAI programs rely on different assumptions about the nature of concepts and underlie different strategies for their representation. All these strategies of representation can be analyzed along seven different dimensions[1]: *intentionality*, *coverage, shareability, typicality, compositionality*, *formality* and *flexibility*. Any type of conceptual representation can be indeed reduced to a way of addressing (following different approaches) one or more issues related to these dimensions. To make the explanation clear, we describe each dimension using the symbolic and the connectionist theories as reference examples. Let us look briefly at each of these in turn.

Intentionality. The notion of intentionality is needed for giving an account of how concepts can be about, represent, or stand for, things (or state of affairs). Intentionality is essential for explaining the semantics of any given conceptual model. Within the symbolic paradigm, a key line of research that account for intentionality is the so called *causal approach to mental content* [**2**]. According to this view, a concept of something in the world is basically a representation caused by this "something" (articulated in terms of sets of properties). The assumption here is that a concept C represents something S, if and only if S causes C. The basic idea is that any conceptual representation is *derived by* and *covaries with* what it represents, according to a causal relation. Similarly, within the connectionist frame, there are approaches on which the notion of intentionality plays an important role. Here concepts are represented as patterns of activation in a network of simple nodes. Even if these patterns are difficult to be semantically evaluated, they are always to be considered in relation to intentional activities.

Coverage. A *desideratum* of a conceptual model is that it can be used for representing all the types of concepts (see, for instance, *individual concepts*, e.g., `here`, `Venus`, etc., *properties*, e.g., `yellow`, `near`; *living being concepts*, e.g., `animal`, `plant`, *stuff concepts*, e.g., `milk`, `gold`; *abstract concepts*, e.g., `music`, `information`; *role concepts*, `student`, `father`; *action concepts*, `create`, `move`, etc.). Providing a model for concepts considering just few examples and the generalizing the model to all the possible types of concepts lead to unconvincing results. Within the models grounded on the symbolic approach we have an account for a large variety of concepts. The concepts that are more suitable to be modeled in a connectionist frame are kinds that are concrete or

[1] These seven characteristics are derived from an analysis of the explanations provided in [**26**], [**98**], [77], [**18**] and [**83**].

singular concepts, i.e., concepts that are strongly grounded in perception (e.g., *basic level categories*) [**101**].

Shareability. Another important feature of concepts is that they have to be shareable between human and artificial agents [**26**]. The shareability dimension of concepts is typically supported by their explicit representation. In the classical symbolic paradigm concepts are constructed from constituent symbols and syntactic combination of these symbols, and they can be seen as the descriptive product of human designers. This support the shareability of the symbolic models. However, an important issue is that, so far, it has proven impossible to develop fully reusable and shareable symbolic representations (see, for instance, ontologies [**49**]), the motivation being the set of underlying assumptions that always underlies their design. In the connectionist approach concepts are implicit representations associated to an *activation pattern*, distributed over different units (or "nodes"), where each unit is involved in the representation of different concepts (this characteristic is the reason of why the connectionist paradigm is also called sub-symbolic paradigm). The "opacity" of connectionist model of concepts is a classical well-known problem for their shareability. Any of these models behaves as a sort of black box and the interpretation for units and connections weight is always deeply complex.

Typicality. Around the mid-70s of the last century the empirical results of Eleanor Rosch [**89**] demonstrated the necessity of a new model for capturing both the structure of ordinary common sense concepts and the categorization processes. The results obtained by Rosch showed that most of the ordinary concepts often exhibit typicality effects, i.e., they have common features that are central in the recognition and categorization of perceived objects. Approaches addressing typicality can be seen as complementary views of the classical symbolic approaches, they were introduced to address some new cognitive issues. This dimension seems to be well-addressed in the connectionist models as well, where concepts correspond to distributed representations with a position in a multidimensional semantic space [**101**].

Compositionality. This dimension refers to the capability of producing infinite complex concepts starting from a finite set of atomic concepts. This is an essential feature for explaining conceptual systems *productivity* [**33**]. Concepts compositionality is well addressed by symbolic approaches, where it is achieved through the application of certain syntactic rules. In the connectionist approaches compositionality seems very difficult to be addressed [**34**]. It is not clear if this depends only on technical issues and some solutions can be provided with few modifications of the existing models. A connectionist model that seem to be able to capture some compositional properties of the symbolic models is the so called *recurrent neural network* (RNN) [**29**].

Formality. Cognition involves many rational and logical processes. Formality (along with abstraction and discreteness) seems to be a key dimension for modelling conceptual information and using it for enabling rational behaviors. For instance, just think of the reasoning activities performed by an agent. These are built upon inference mechanisms dealing with formal and consistent information. Within the symbolic frames, formality is a key dimension. For this reason, most of the systems devised according to the symbolic principles are targeted for enabling forms of logic-valid automatic processes. Differently, formality is not a central dimension in the connectionist frames and the development of solutions for supporting rational tasks grounded on the connectionist approach is not so widely explored. Anyhow, recently interesting connectionist experiments, suitable, for instance, to address logical deduction, are being devised, see for instance *neural reasoners* [**84**].

Flexibility. The high flexibility of biological cognitive agents is a pivotal feature of their cognition system (see, for instance, *learning*, *evolution* and *adaptation* tasks). The focus on formal semantics and LOT makes flexibility, comparatively speaking, difficult to be addressed for models built upon a symbolic approach. Any update, modification or elimination task have a huge impact on the whole symbolic system. Nevertheless, there have been some developments of flexible symbolic models, see for instance default logic [**9**], fuzzy logic [**57**] systems. In the context of connectionist paradigm, the very foundation has always been learning. This presupposes the high flexibility of connectionist models, which is addressed by tuning of weights or other parameters in huge networks [**101**]. Due to such a flexibility, connectionist models excel at dealing with incompleteness, inconsistency, uncertainty, approximate information, and so forth, and seem to be high capable of simulating some complex cognitive behavior of biological agents.

2.3. Classical-Symbolic Representations

A perfect example of a computational approach to conceptual representation that is influenced by classical-symbolic principles is the modelling of ontologies [Ref.]. The most widely shared definition of ontology in the computer science community is "*a formal, explicit specification of a shared conceptualization*" [**49**]. Such a definition can be seen as derived from the composition of different definitions [**104**]. Ontologies can be seen as complex data structures, i.e., information artifacts, which can be designed (and formalized) using different representational languages, see for instance RDF[2] and OWL[3], following different principles (e.g., OntoClean [**47**]). All the languages used for representing these "conceptualizations" can be reduced to a fragment of *first order logic* (FOL) and can be seen as a perfect instantiation of what in the classical-symbolic frame is taken as LOT [**32**]. The main goal of these artifacts is to support knowledge representation (KR) and integration tasks, but they can be used for other tasks as well (see for instance driving NLP, or providing a data exchange formats). DOLCE[4] and BFO[5] are two example of top-level ontologies, i.e., ontologies representing cross-domain knowledge. Both of them are designed according to well-defined principles and are used in many different ontology design and integration tasks. OWL-Time[6] and the Organization[7] ontology are typical examples of what in the ontology design community are called "core ontologies", i.e., ontologies (more specific that top-level ontologies) expressing and specifying some concepts that can be shared among different area of knowledge. OWL-Time is an ontology expressed in OWL-2[8], describing temporal concepts, enabling the ability to express facts about topological relations among intervals and instants, together with information about temporal position, frequency and durations. The Organization ontology provide a conceptualization representing the structure of organizations (e.g., business organizations, educational organization, an d so forth). It is designed in order to equip specific domain

[2] https://www.w3.org/RDF/
[3] https://www.w3.org/OWL/
[4] http://www.loa.istc.cnr.it/old/DOLCE.html
[5] http://ifomis.uni-saarland.de/bfo/
[6] https://www.w3.org/TR/owl-time/
[7] https://www.w3.org/TR/vocab-org/
[8] https://www.w3.org/TR/owl2-overview/

applications with information about organizations and roles. The Wine[9] ontology is another example of ontology, i.e., a domain ontology. Such an ontology is often used as reference object for tutorials and ontology design tasks, and provide a representations of wines, winery and all the objects needed for expressing this specific area of knowledge.

All these computational artefacts provide an abstraction of (a portion of) the world and are used to enable software systems in addressing some specific (high-level) intelligent tasks. Looking at the dimensions provided in the previous section, there are some main observations. From the point of view of coverage, ontologies can be used for modelling a huge varieties of concepts. As a check we can take the huge number of ontological vocabularies collected in LOV[10]. Let us take, for instance, *Schema.org*[11], which can be expressed and formalized as an ontology (see the its RDF formalization) and the set of its "commonly used types". We have concepts like `CreativeWork`, `Artifact`, `Event`, `Organization`, `Person` and `Place`; concepts like `Action` (e.g., defining actions like `Assess`, `Achieve`, `Move`, `Organize`, along with `Create`, `Reproduce`, and so forth). Similarly, we have concepts of roles like `Creator` or `Student`, concepts of properties like `Gender`, `JobTitle`, `Nationality`, and so forth.

Ontologies may also provide interesting insights on how to address the shareability issue. The main goal of ontologies is indeed "*to enable computers and people to work in cooperation*" [**8**]. Here shareability is addressed by providing an explicit and formal representation of each concept. For instance, the concept `Person` and `Nationality` map into a specific logical formula that can be reused among different software agents and can be used by humans for understanding the intended model behind the concept representations. Along with shareability, the concepts represented with ontologies are characterized by formality and compositionality. Firstly, they are indeed used for enabling logic-based process and their main role is to discretize and schematize the information coming from the external environment. Secondly, they can be seen as symbols of a

[9] https://www.w3.org/TR/owl-guide/wine.rdf
[10] http://lov.okfn.org/dataset/lov/
[11] http://schema.org/

propositional formal language, characterized by a specific syntax and different compositional rules.

For what concerns intentionality ontologies are for sure representation of concepts that aim to capture what these concepts are about. The pivotal point here is that world objects are mapped into key basic constructs, i.e., *instances* which should point to specific world object (or an occurrence of them). This enable the class-relational structure encoding ontologies with a formal semantics. Formal semantics is necessary for supporting intelligent logic-based tasks (see for instance reasoning) but involve losing the tie with the external environment. Loosely speaking, do ontology concepts refer to real world objects, or to a formal representation of them, i.e., instances?

The default features of ontologies do not allow to address the typicality and flexibility dimensions. Anyhow, there is a huge research work on ontology evolution and adaptation (see [**80**] for more details and [**9**], [**57**] as examples of formalism) and typicality is addressed by some complementary approaches that can be integrated to more classical models (for further details see Section 6 below).

2.4. Connectionist and Embodied Representations

Neural networks are typical computational representations inspired by the connectionist view of concepts. These artifacts can be reduced to a set of interconnected units, i.e., abstract representation of neurons, where any connection between these neurons is an abstract representation of a synapse. According to these representations, each unit is associated to a numerical value, i.e., an activation state (or firing, namely the frequency by which a neuron sends signals through synapses). Each connection between neuron representation units is characterized by a weight that codifies the strength of that connection. The influence of a unit x on a unit y is given by the activation value of the unit x multiplied by the weight of the connection from x to y. The weight value can be positive or negative so that the signal sent through the connection can activate or deactivate the neuron reached by the signal. So far, a lot of neural networks have been devised for capturing aspects of cognition. Feed forward networks (FF or FFNN) [**91**] are usually employed on pattern recognition tasks. These are powerful networks characterized by three layers of different units: *input* units, *hidden* units and *output* units. The connections of FF networks are always unidirectional, i.e., they start always from an input unit through a hidden unit until an output unit. One interesting aspect of these networks is that they can be easily trained, i.e., they can learn how to produce results and tune their activation state by using a *back-propagation* mechanism. This allow the network to improve their reactions to given inputs and then improve their results. Besides FF networks we have many other (more or less recent) kinds of neural networks, for instance: radial basis function (RBF) networks [**21**], hopfield network (HN) [**54**], Markov chains (MC) or discrete time Markov Chain, (DTMC) [**51**], deep belief networks (DBN) [**7**] and deep residual networks (DRN) [**52**]. Each of them was devised for enabling some specific artificial activities.

Looking at the dimensions provided in the previous section, there are some main observations. From the point of view of intentionality we may say that connectionist representations of concepts share the same goal of the classical approaches. Anyhow, neural networks, in addition, provide interesting performances on dealing with partial or wrong information. This means that they can infer new information starting from the available partial information. In other words, they can run induction processes by which unknown properties of objects can be discovered starting from the properties of objects that we already known. Formality and compositionality are not pivotal requirements of these models, since they do not commit to a formal semantics. For what concerns the shareability dimension, we can say that this is not a key dimension as well. Here concepts are not explicitly represented, but derived by some properties of the net: this limit their human understandability and the possibility to share them among different situations and agents. Similarly, coverage is not a priority within this framework. By contrast, neural networks can perfectly address the typicality and flexibility dimensions. These information artefacts are indeed able to generate prototypical representations by generalizing from the collected data. This is mainly because they are the result of a training process enabled by the adaptation and evolution capability of the entire net.

Among the NFAI computational representations of concepts we have the so called embedded (or situated) approaches, which are usually implemented by the situated robotics research program. A key exemplification of these approaches is the work by the MIT research group, managed by Rodney Brooks [**20**]. This group is building robots that are equipped with simple sensory-motor devices and a collection of modules. Each of these modules are specialized for addressing a specific task, such as checking for the presence of an obstacle, avoiding an obstacle, exploring, and so forth. Each of these activities is run by a processor that works together with other processors and exchange information with the sensory-motor system and other processors. In these models no explicit representations are provided and no data is stored. The robots are not equipped with a mental model; they are automata that can be described just through finite states [**83**]. All the information used by these agents is grasped from the environment. Here concepts can be seen only as temporary representations, information flows, built upon the different phases of the perceptual process. The main goal is to derive the useful information from the environment, send them to the right processors and then produce an action. Thus, every robot can be seen just as a collection of behaviors in competition [**19**]. From an external point of view, it is possible to detect coherent behavioral patterns. However, locally, these robots are characterized by just casual processes. The robots devised following the situated approaches seem to be able to reproduce the cognitive capabilities of some insects, and, according to recent results, seems that can be evolved by introducing new processing modules connected to the others.

The embodied approaches offer interesting insights for understanding how to address a conceptual representation task. The main findings are the following. Firstly, it seems to be possible to address cognitive tasks without having explicit representations of the external environment. As showed by Rodney Brooks automata different tasks may just be controlled by perceptual-motor loops. Secondly, grounding cognitive tasks in perception may be useful for addressing the so called *frame problem*, i.e., the difficulty of a cognitive agent to adapt to specific and different situations, given the abstract and rigid nature of their internal representations. An agent that is embodied in the context and the environment seem to be more capable of adaptation. It is clear that the pivotal dimension addressed by the embodied approaches is the one of flexibility. It is interesting to notice how, within this framework, differently from neural networks, flexibility must be grounded on sensory-motor devices, and how the main assumption is that "intelligent behaviours are possible without representations", i.e., collecting data only form the external environment.

2.5. Complementary Representations

Besides the approaches supporting the classical idea of concepts as symbols of a language of thought, characterized by a propositional nature, we have approaches supporting the idea that concepts are analogical representations. According to these frameworks concepts are mental entities that are similar to the entities they represent, i.e., they are like pictures of (portions of) reality. In AI this view is well supported by research result like [**96**] and rises the issue of how some cognitive process are grounded on imagination and deal with mental images. The underlying assumption of the analogical approaches is that perception plays a central role in cognition. This lead them to focus on the relevance of simulation processes and share some hypothesis with the embodied approaches to representation. There is a lack of computational frameworks implementing the analogical approach, however, recently, some solution grounded on this paradigm are being developed. For instance, see the work in [**92**], whose attempt is to provide a computational accounts of cognition in modality-specific processing [**6**]. Furthermore, examples of attempts in implementing simulations are in [**23**], [**22**], [**56**] and [**81**]. The difficulty to formalize the analogical approach seems to be the main reason of why a mature computational account of this framework still need to be provided. However, there are not a priori reasons why formalization is impossible. The main conceptual dimension addressed by this view are for sure, intentionality and flexibility. Concepts as analogical representations are taken to be mirror images of something that is outside the mind and, they are considered as simulations of what is experienced and they are taken to be adaptable and flexible according to the variations provided by perception.

Among the complementary approaches we have also the procedural representations. Differently from the analogical approaches, we have more computational frameworks implementing the ideas of these approaches. For those in AI starting from the procedural frame, the key idea is that concepts can be implicitly represented as fragments of algorithms. Concepts can be reduced to a sort of know-how that is not explicitly representable by means of data structures. However, these algorithms need some explicit information, or data structures, to work. Thus, procedural approaches do not exclude the possibility that the mental content is partially determined by some explicit information, but they state that such content is determined by the operations performed over it. Every representation is both involved in a causal relation with the external environment and in a causal relation with some mental operations. Good examples of computational frameworks linked to procedural semantics are semantic networks like KL-ONE [**16**] (for a detailed description see [**30**] [**15**]) and resources like WordNet[12] or FrameNet[13] inspired by *Inferential Role Semantics* (IRS), Lexical Semantics (LS) or *Frame Semantics* [**31**], i.e., a semantic theories that underlie lot of the procedural assumptions. Moreover, we have works like the one in [**40**] with a particular focus on teleosemantics [**69**], providing its (partial) formalization, its application in the context of KR and its integration with the classical approach. Within the procedural framework, for sure, coverage and shareability dimensions are not taken as priorities. Usually, procederual representations apply to some certain kinds of concepts only (e.g., there are no accounts for property concepts), and since these concepts are often modeled as implicit procedures or not formalized networks, they are not claimed to be sharable. Similarly, for what concerns the typicality, there is a lack of results addressing this dimension. Intentionality, on the other side, is of course a central issue, since the main assumption underlying proceduralism is that concepts are causally related to the external environment. Formality and compositionality are in some extent addressed by some computational approaches, but they are not in the agenda of the original procedural program. Finally, flexibility seems to be a well addressed dimension, concepts can be seen indeed as devices that change in relation to the environment and the specific task that need to be addressed. In this regard see for instance the notion of function provided by teleosemantics [**62**].

Among the complementary theories we can find another group of theories, i.e., what we called PET. Is not easy to trace back these approaches to a computational framework. However, we can find some work in AI explicitly taking some of the ideas grounding these theories and trying to better cover some dimensions of their representations. An exemplar

[12] https://wordnet.princeton.edu/
[13] https://framenet.icsi.berkeley.edu/fndrupal/

computational work exploiting some of the features of prototypical an exemplar theories is the one provided by Lieto and Frixione [**60**], which is also partially inspired by the theory of conceptual spaces (see [**38**]). Here the main goal is to combine the typicality effects of a prototypical representation with the compositionality effects of a more classical representation of concepts. The result is a sort of hybrid architecture, i.e., what they call DUAL-PECCS [**61**]. This is basically an integrated KR system aiming at supporting artificial cognitive capabilities such as categorization, by implementing classical, prototypical and exemplar-based representations of concepts. For what concern the theory theories, in some extent, we may say that core ontologies are examples of computational applications. Just think the organization core ontology: as we showed in Section 4, this is a typical formalism grounded on the symbolic frame, however it can be also seen as a (formal) micro-theory representing the corresponding concept.

2.6. A summary Overview of the Computational Approaches

In Table 1 below we list the computational implementations we described, grouping them according to the classification provided in Section 2 and the characterization provided in Section 3.

Table 1. Comparing different computational approaches to concepts representations

CONCEPT REPRESENTATION CLASSES

CHARACTERIZATION DIMENSIONS	Classical Representations	Post-classical Representations	Complementary Representations
Intentionality	✓ Ontologies are devised to capture a representation of what exist and can be easily traced back to a representational approach where intentionality is a pivotal requirements (see [**49**])	✓ Neural networks are always involved in tasks that simulate intentional activities. Example of particular interests are FFNN [**91**] ✓ The existence of an external environment is a key assumption of the embodied approaches [**19**]	✓ A mandatory requirement for both analogical and procedural theories [**15**] ✓ A key dimension for all the PET theories [**36**]

Coverage	✓ Ontologies provide an account for a large variety of concepts, see for instance huge resources as *Schema.org*	× Not a pivotal dimension. Connectionist approaches provide an account for specific kind of concepts (see for instance empirical concepts involved in learning and prediction [**34**]. × According to the embodied approaches concepts are just behavioural patterns. Their classification and characterization is not addressed [**19**]	× This dimension is not covered by procedural and analogical programs, since both of them are focused on empirical concept (most of the examples are grounded on vision [**83**] × The range of concepts that can be represented by PET theories is for sure wider than the one covered by analogical and procedural theories, however it is still limited (abstract concepts are not addressed, property concepts are not considered, and so forth [**26**]

Shareability	✓ A pivotal dimension addressed by ontologies (used in their evaluation, which, as for [**104**], should be characterized by an explicit (human understandable) representation of concepts	✗ Not a pivotal dimension, mainly because of the sub-symbolic nature of neural networks (this issue is discussed mainly in [**101**]) ✗ Not a pivotal dimension, mainly because of the anti-representational view of the embodied approaches [**83**]	✗ According to analogical and procedural programs, concepts are not claimed to be shareable, they are internal (or local) representations/processes and the issue of how these can be shared among agents is not explicitly addressed [**18**], [**83**] ✓ PET theories account for shareability. Prototypical and exemplar effects, for instance, are claimed to be shareable among different agents [**36**]
Typicality	✗ Usually this dimension is not addressed. Ontologies are characterized by a lack of prototypical or exemplar information (for further explanation see [**37**])	✓ Neural networks are able to generate prototypical representations by generalizing from the collected data [**83**] ✗ According to the embodied approaches it is possible to derive behavioural patterns [**20**], but these are far from capturing prototypical or exemplar effects	✗ There is a lack of work in addressing this dimension within the analogical and procedural frames [**83**] ✓ A pivotal feature, perfectly addressed by prototypical and exemplar models [**36**]

Compositionality	✓ A typical characteristic of ontologies, which can be seen as a computational instantiation of LOT [**32**]; see for instance atomic/complex concepts distinction in DLs [**3**]	✗ Not a pivotal dimension, mainly because of the sub-symbolic nature of neural networks, see the discussion in [**34**] ✗ The anti-representational view of the embodied approaches does not allow to cover this dimension [**19**]	✗ This is not a requirement for analogical and procedural representations [**26**] ✗ This dimension is not clearly captured by PET approaches, however there is a lively debate on this issue [**37**], for instance, describes how prototypes can be compositional

Formality	✓ Ontologies enables forms of logic-based systems, some of their most central features are formality, abstractness and discreteness ([**50**], [**49**])	✗ NNs are devised according to a well-defined formalisms and in some cases, see for instances the hybrid approaches described in [**84**], can be mapped to logic, however they are not devised for logic-based systems ✗ The conceptual representation of embodied approaches can just be inferred. Locally, concepts are not represented and there is no formalism depicting a mental model. Each activity is the casual result of the composition of some processes and mechanisms [**19**]	✗ Even if there are some work that are starting to provide a formal account, these dimension is still far to be addressed in the context of procedural and analogical views ✓ A pivotal feature, perfectly addressed by prototypical and exemplar models [**26**]

Flexibility	✗ Not a pivotal dimension, partially addressed by adopting some ad hoc methodologies (see ontology evolutions techniques in [**80**])	✓ A pivotal dimension in NN, which is addressed by tuning of weights or other parameters in huge networks [**101**]. Neural networks are central in capturing these aspects and this makes them essential in tasks involving, for instance, learning and adaptation ✓ Embodied approaches are devised mainly to address this dimension. The main outcome of these approaches is that robots can easily adapt to the external environment [**20**]	✓ Not explicitly addressed by analogical approaches, but clearly addressed by procedural approaches [**15**] ✗ This is cannot be considered a key dimension of PET approaches, and so far their ways of addressing flexibility still need to be clearly discussed [**37**]

Part II

The Proposed Solution

Chapter 3

3. Concepts as (Recognition) Abilities

*Concept*s are an essential notion for the understanding of human thought. They allow us to give an account of phenomena such as knowledge acquisition and representation, language understanding, inference, and categorization [**59**]. A mainstream line of research on this topic, called in the philosophical literature *Descriptionism* [**75**], takes concepts to be classes. According to this view, a concept of something in the world is a *representation* of this "something", articulated in terms of sets of properties. Descriptionism has had a large influence on the work in Artificial Intelligence (AI) and Knowledge Representation (KR) and has motivated various KR languages. The main focus of this work has been (and still is) on how concepts can be used to organize knowledge via the *classification* of instances into classes as a function of their properties. Although KR formalisms have been used in several applications with many success stories, there are still many open issues related, for instance, to the several roles played by concepts in cognition, see, e.g., [**59**] for a discussion of some of the issues which arise with this approach.

Lately, the field of *Teleosemantics* has proposed an alternative approach. According to this school of thought, concepts implement suitable (biological) *functions*. The shift is from the study of the means by which the world is represented to the study of the means by which such representations are generated. Here the notion of function is the same as that used in neurobiology when attributing functions to components of the brain (as in "the function of processing visual information"). According to this view, concepts are components (*devices*) of the human brain characterized by sets of *abilities* of performing, under certain conditions, specific functions. Most relevant to us is the work by Ruth Millikan [**107**]. Millikan's work concentrates on what she calls *substance concept*s, namely, specific types of concepts which can be characterized as *abilities of recognizing* a certain type of items, that she calls *substances*, which are perceived as being part of the real world [**70**]. *Substance concepts* have the main function of collecting and accumulating knowledge from the world.

The goal of the work described in this chapter is to lay the foundations of a unified theory of perception and KR that integrates the results from the two approaches above. The underlying intuition is to think of *all* types of concepts as abilities, to identify the different forms of functions, and corresponding representations, and to study how these functions can be composed as part of an overall *process* enabling cognition. Thus, if substance concepts are recognition abilities, when we concentrate on the classification task, as it is the case in KR, we think of concepts as *classification abilities*, namely as abilities "... *of simplifying the environment, of reducing the load on memory, and of helping us to store and retrieve information efficiently.* …" [**75**], [**64**]. This chapter provides the following three contributions:

a) It provides a model of concepts as recognition abilities by clarifying their role and by defining their main characteristics. This work can be seen as providing a rationalization and formalization of Millikan's work. The main result is a precise characterisation of the similarities but also the (non-trivial) differences between concepts as recognition abilities and concepts as classification abilities.
b) Based on the results above, it provides the definition of an (early version of an) ontology *RAO*, for *Ontology of (Recognition) Abilities*, as the basis for an integrated study of the two types of concepts.
c) It provides the beginning of a methodology for how to use RAO for discovering which concepts as classification abilities, among those contained in the state of the art ontologies, correspond also to recognition abilities.

It is important to notice that, within KR, various approaches have attempted to provide broader notions of concepts and/or to overcome some of the existing limitations. Some examples are: methodologies for making explicit the semantics of the underlying conceptual models inside KR languages [**50**], the analysis of cognitive and ontological principles that ground knowledge engineering processes [**46**], the implementation of new conceptual theories, with a sound cognitive foundation, such as conceptual spaces [**2**], [**38**], the perceptual symbol system approach [**6**], [**85**], the proxy-type theory [**60**], [**86**]. In addition to these theories we may find works addressing the problem of empirical classification and of how to build representations from "observations" [**11**], [**58**]. The work described in this chapter is orthogonal to this work and, as far as we know, it is the first attempt to provide a unified view of concepts as recognition abilities and as classification abilities.

This chapter is organized as follows. Section 3.1 defines the notion of substance concept. Sections 3.2 and 3.3 analyse the various kinds of substance concepts. Section 3.4 provides a comparison between concepts as recognition abilities and concepts as classification abilities. Section 3.5 introduces RAO and its main categories. Section 3.6 provides an example of how to use RAO for the identification of substance concepts among the concepts which are used in state of the art ontologies. Finally, Section 3.7 analyses the implications of the results presented in this chapter for the development of complex AI systems which integrate recognition and knowledge representation, as a first (small) step towards a unified architecture for cognition.

3.1. Substances and Substance Concepts

We model how things are in terms of *subjects* able to experience the *world*, where by world we mean anything that is external to the subjects themselves. We call these subjects, *perceptual-cognitive systems (PCSs)* [**71**] to emphasize our focus on the study of systems where perception and knowledge are integrated.

Time is the horizon over which PCSs and the world "meet". A PCS experiences the world through *encounter*s. An encounter is the event through which (a portion of) the world *manifests* itself to a PCS. We call such part of the world, *substance*, where, quoting Millikan, "*… substances are those things about which you can learn from one encounter something of what to expect on other encounters, where this is no accident but the result of a real connection*" [**75**]. The uniquely identifying characteristic of substances is their ability to manifest some form of invariance through multiple encounters. This invariance is grounded in what we call the substance *causal factor* [**72**], meaning by this *an inner characteristic which is associated to a substance and which is the cause of its invariance* *across encounters*. In turn, this invariance takes the form of a set of *outer characteristics* which occur across encounters and allow for the recognition of a substance. Thus, for instance, cats[14], like all species, are characterized *by a homeostatic mechanism* which, in turn, causes them to possess a certain set of common traits (e.g., in shape or weight) and, often but not always, to look similar.

As from the above quote, substances are subjects of learning, namely, of the generation of new knowledge from perception. This process is enabled by *substance concepts*, where substance concepts are taken to be *recognition abilities*, namely *abilities which allow a PCS to realize that the substance involved in the current encounter is the same substance as from previous encounters*. Substance concepts implement functions that allow to recognize a substance as such and to learn and to cumulate the new knowledge about it through a sequence of encounters. They allow to recognize sameness of content in time and also to *group pieces of information together, as being from the same substance* [**70**]. Substance concepts are innate abilities, which are at the core of cognition, which match the stimuli coming from substances (what we call *signals*) and which allow humans to generate knowledge from signals. Consider, for instance, the substance concept "cat"[15]: we can observe today that cats drink milk or that scratch when we disturb them and this knowledge will be confirmed in future encounters.

The set of (outer) characteristics that a substance manifests over encounters are matched with a set of *substance property concept*s, or simply, *(substance) properties*, which are associated to its corresponding substance concept. Substance properties play a central role in the recognition of substances. A substance property is an *ability to discriminate a substance characteristic over encounters*. This ability is manifested in sameness of reaction to substance characteristics. There are two types of properties that we call *determinables* and *determinates* [**73**], where determinables can be thought as slots that collect determinates. Thus, for instance, *colour* is a determinable which is used to collect

[14] Throughout the chapter we write cat meaning *Felis catus*.

[15] To distinguish between substance concepts and substances we write the former in "quotes". Thus for instance, "cat" is an example of substance concept which corresponds to the substance cat.

determinates such as *blue*, *red*, or *yellow*. From a biological point of view, determinable properties correspond to the use of neurons located in certain early sensory areas of the brain (e.g., colour is tuned to neurons in *visual* cortex) while, on the other hand, determinate properties (e.g., red) would represent single states produced as a reaction to perception (e.g., a red stimulus) [**63**].

3.2. Kinds of Substance Concepts

There are two types of substances and, correspondingly, two types of substance concepts, namely *individuals* and *real kinds*. Individuals are single units, scattered in space, enduring through time. In language, individuals are usually revealed by the use of proper nouns or definite descriptions. Examples of named individuals are Barack Obama, my cat Garfield and the Empire State Building. On the other hand, we usually think of real kinds as clusters of elements, what we usually call the real kind *members*, which are characterized by *a common, empirically observable, connection grounded in some*, most often natural, *law*. Real kinds "*... allow successful inductions to be made from one or a few members to other members of the kind not by accident*" [**72**]. Examples of real kinds are: stuff, e.g., gold or water, biological species, e.g., cat and *Quercus Alba*, artefacts, e.g., chair and car, and also social roles, e.g., doctor and father. The members of real kinds, what we perceive as a "generic" chair or cat, are substances as well.

A first observation relates to the statement that real kinds, their members and individuals are all substances, a statement which is somewhat counter-intuitive for anybody working in KR. For someone coming from this field, the most obvious way to think of the world is to map real kinds to classes and individuals to instances which, in turn, are members of classes. This mapping is discussed in detail in Section 5 below. Here it is worthwhile noticing that with substance concepts we focus on recognition, modelled as an ability. In this respect, both individuals and real kinds share the property that, during an encounter, they are only partially perceived by PCSs. In the same way as we *always* perceive only one or a few members of a kind, we *always* get only a partial view of an individual (e.g., the back or the front). The best way to understand this commonality is to think of substances, no matter whether they are individuals or real kinds, as *wholes* which are only partially presented, with some of their *parts*, by their manifestations to a PCS. In perception there are neither sets nor instances, there are *only* wholes (substances) that are perceived *only* partially. There is however a key difference between real kinds and individuals which is at the basis of the KR representation of the world in terms of classes and instances. Real kinds have the property, not possessed by individuals, of being in multiple places at the same time, meaning by this the fact that any kind can have, at the same time, multiple occurrences inside one or more (contemporary) encounters. This property, clearly, does not hold for individuals: a PCS will perceive at most one individual as part of the same encounter. Thus, for instance, I can perceive two cats together on top of the wall in front of me, but I can only perceive (at most) one occurrence of Garfield per encounter.

A second observation relates to the fact that the inductive grounding that allows the recognition of the same real kind across encounters is very much the same as for individuals. The key observation is that a real kind manifests itself through its members. Both in the case of individuals and of real kinds, the PCS is faced, in time, with similar characteristics that allow a substance to be recognized as being the same from a previous encounter. Thus, for instance, the members of the real kind cat, what we usually call cats, like all species, possess a certain set of common properties (e.g., similar shape and weight) and, consequently, often look similar. Analogously, Garfield, like all individuals, looks pretty much the same across encounters. The ability of substances to manifest some form of invariance through multiple encounters is grounded in their causal factor, as defined above. But the nature of this causal factor is very different between real kinds and individuals. In the first case it consists of some causal connection that is shared by all members of a real kind while in the second case it is related to the fact that the same individual usually changes slowly in time.

The third observation relates to the process by which substances get recognized through substance concepts. This observation is also crucial to understand the distinction between individuals and members of a real kind, a distinction that in KR is blurred into the notion of instance. This distinction is, again, deeply rooted in the profound difference which exists between recognition and classification. Consider for instance an encounter with Garfield. What will the PCS recognize: the individual Garfield or the (member of the) real kind cat, what we usually call “a cat”? We have the first case when recognition happens via the individual substance concept, the second case when the real kind substance concept is enabled. The “selection” of the substance concept is related to the substance properties being recognized. This process is not univocal and depends on many factors. The most important seems to be the actual goal of the PCS (is it looking for Garfield because it wants to feed it or is it just trying to avoid hitting a cat running in front of the car?), but it also depends on the context (e.g., it is harder to recognize an individual at night), on which characteristics are manifested and/or grasped (it is harder to recognize an individual from the back) and so on. Notice that the recognition of Garfield will most likely exploit different characteristics from those used in the recognition of a cat. In the first case, the PCS will exploit those characteristics that uniquely identify Garfield among the other cats, while in the second case it will exploit those characteristics that uniquely identify cats among the other animals. These two sets of characteristics overlap only partially.

The fourth and last observation is that the same substance changes over different encounters thus presenting a set of continuously evolving characteristics. Thus, for instance, two encounters with the real kind cat may produce very different manifestations, though looking similar to two other manifestations which in turn look similar to two other manifestations which …, eventually, will look similar. As a paradigmatic example, under what conditions a person is (recognized as being) the same person as 30 years ago? If I meet a person after 30 years, most likely I will not recognize her as being the same individual. Dually, with an individual with no salient distinguishing marks, there is a high probability to fail its recognition over encounters. Think for instance of forks. In this case what usually happens is that only the real kind fork is recognized as there is no interest in distinguishing among the various individuals. We just look for any fork. This of course will not be the case with that specific fork that I was playing with when I was a kid.

3.3. Kinds of Real Kinds

Real kinds can be further divided into two more specific categories, i.e., *eternal kind*s and *historical kind*s.

Eternal kinds correspond to what is often called *stuff*, e.g., “gold” and “water”. The members of eternal kinds share some fundamental characteristics without being historically related to one another. This inner structure remains stable over time without exceptions. Eternal kinds are often expressed through their mass and/or atomic number, are named using uncountable nouns and are said to have “essences” in a very classical sense, i.e., essences that can be discovered through empirical investigation.

Historical kinds are real kinds “… *for which historical location does play a role in explaining likeness*” [**72**]. Examples of historical kinds are species, artefacts and social roles, e.g., “doctor” and “baker”. Historical kinds are named in this way because their members bear a certain common relation which has evolved in time. For instance, consider species. All their members have a connection with some prior member from which they derive their characteristics. Similarly, all artefacts can be seen as being derived from some prior member, i.e., a prototype, a model of a chair. Finally, younger doctors learn how to act from older doctors.

Historical kinds are strongly correlated and, for what we have figured out so far, include as sub-kinds what the psychologist Eleanor Rosch calls *basic level categories* (and *objects*). These are the concepts that children learn first and use to categorize the world. They are the easiest to recognize via sensory (e.g., visual) and motor interaction with substances. Basic level categories can be detected by running experiment(s) like the one described in [**90**]. As shown in this experiment, in a hierarchy (a classification) of categories, basic level categories maximize the number of characteristics shared by their members and minimize the number of characteristics shared with the members of their sibling categories. A further characterization of basic level categories is that, usually, the members of their superordinate categories share a very small number of characteristics while the members of their subordinate categories, usually, share a large number of characteristics that, however, are shared also by the members of the sibling categories. The consequence is that the members of basic level categories have a much higher probability of successful recognition than the members of their superordinate or subordinate categories. Recognizing a cat, for instance, is much easier than recognizing an animal or a Siamese cat. In other words, basic level categories provide the ideal balance between the similarity of their members and the dissimilarity of the members of their sibling categories. One interesting observation is that, contrarily to what was initially expected by anthropological and linguistic researchers, biological basic level objects are at the level of abstraction of species, namely one level up from the level of abstraction of the basic level objects which are artefacts (e.g., furniture, as from the experiment by Rosch).

Following Rosch we can further distinguish basic level categories, namely historical kinds, into *biological* and *non-biological basic kinds*. The former are the basic units of biological classification, i.e., biological species, while the latter are defined as the complement of the former and are therefore not well characterized. Examples of non-biological basic kinds, are artefacts like “car” or “chair” (subsumed by superordinate categories like vehicle and furniture, respectively) or social roles like “doctor” and “baker”.

3.4. Classification and Recognition

In KR, the main focus so far has been on classification more than on recognition. As a result, knowledge is modelled in terms of *instances* (e.g., Garfield), *concepts* (e.g., cat), namely sets of instances and *properties* defined as the Cartesian product of two classes, e.g., being of colour yellow, being near something). Concepts are associated to sets of properties and the values of the latter allow to make distinctions among the members of the former. We call below this kind of concepts, *classification concepts*, or simply *classes*, to distinguish them from substance concepts. We also talk of *classification properties* when we need to distinguish them from substance properties.

The mapping between the work and notions defined in this chapter and these notions coming from KR can be established based on the following steps:

a) We think of *classification as* the *ability of organizing instances into classes as a function of their properties*. This is the ability that generates and manipulates classes, classification properties and instances as representations of the world.
b) With an overloading of the terms, we talk of substance concepts and substance properties meaning not only the corresponding functions and abilities but also the

representations generated by such functions, and dually for classification concepts. This allows to eliminate the difference in approach between us and the "usual" KR approach. For both classification and recognition, we distinguish among devices, abilities, functions and representations only when needed.

c) We acknowledge that recognition and classification are two distinct abilities which generate and manipulate distinct representations of the world, the first being a perception-oriented representation the second being a semantic language-oriented representation of the world. This implies that classes and substance concepts, classification properties and substance properties, instances and individuals are actually distinct representations. It is important to notice that this assumption is coherent with the most recent discoveries in neuroscience which provide evidence that perception and "semantic" oriented representations are actually stored in two different parts of the brain [**65**].

As a result of these assumptions we are now in the condition of studying the pair-wise similarities between the recognition and classification representations of the world. The existence of these similarities is the obvious consequence of the fact that *substance concepts and classification concepts are both representations of substances*. However, this mapping is far less obvious than one would expect, the motivation being rooted in the very nature of the functions of recognition and classification. Substance concepts allow to recognize substances over encounters and to acquire knowledge about them, while classification concepts allow to group together substances about which we already have some knowledge. Thus the former are *representations of sets of occurrences of substances*, while the latter are *representations of sets of substances*. With substance concepts we describe substances *over* time, while with classification concepts we describe substances *in* time. Similarly, individuals are *representations of sets of occurrences of substances*, while instances are *representations of (single) substances*. This generates various crucial distinctions.

Let us start from individuals. An individual is a set of occurrences of the same substance and, as such, it can be mapped to the single instance representing that substance. A crucial difference is that individual substance concepts need not have names. Names play no role in the recognition process, while they are crucial in the deployment of classification abilities: you need an identifier to be able to refer to an instance, this is a prerequisite to classification. Furthermore, the mapping individual – instance is not one-to-one. Thus, I can have two or more individuals for the same instance because I did not recognize them as being (sets of) manifestations of the same substance, e.g., myself at the age of five and the age of fifty, or myself dressed as Santa Claus and being recognized as such. The contrary, namely having two or more instances for the same individual, seems to be the case only when there is a need to reason about occurrences of individuals in different moments in time, e.g., because reasoning of the color of my hair at the age of five and at the age of fifty. Furthermore, I can have an instance which does not correspond to an individual, e.g., the *Minotaur*, because it is a product of the mind with no existence in the real world, or *Homer* who I have never seen in person or described in any form; but I can also have an individual which is not an instance, e.g., a specific part of the mountain I see every day from the window of my office. I look at this view every day, I love it but I do not need to name it because there is no need for me to classify it and reason about it.

Real kinds can be mapped to classes. Classes are sets of instances where real kinds group sets of encounters, one set per instance. Again, as in the case of individuals, and for the same reason, real kinds and their members need not have names. It is interesting to notice that in natural language, when we speak of a member of a real kind, e.g., “cat”, we speak of *a cat* meaning a generic cat while we speak of Garfield meaning the specific individual. And the kind of mental image and reasoning that is performed in the two situations is usually different. In most cases we have a one-to-one mapping between a class and a real kind. However, as for individuals, this mapping presents lots of exceptions. Thus we may have a real kind with no corresponding class, similarly to what happens for individuals. We may also have a class with no corresponding real kind. Following Millikan, we call these types of classes, *Nominal Kinds* or simply, *Nominals*. Nominals are sets of instances which do not share a causal factor but that, rather, *are grouped together according to a definition provided in terms of the set of properties they share*.

With nominals we have two possible situations. In the first case we have one nominal kind that can be mapped to two or more real kinds. This situation arises with concepts like “stone” or “animal”. Thus, “animal” can be thought as the union of the real kinds “cat”, “dog”, “lion”, and so on. With stones, for instance we may focus on their shape, weight, composition and so forth. These types of nominal kinds are concepts that are high in the abstraction hierarchy and are very useful to classify and organize real kinds. But this conventional, theory driven, characterization of stones has nothing to do with the rich, recognition driven substance concepts. When I say “animal” which image of which animal should come to my mind? A cat, a crocodile, or …? In the second case we have a real kind that can be mapped to two or more classes. This situation arises any time we distinguish among the members of a real kind by assigning them some specific property. Thus, for instance, “cat” can be thought as the union of “white cat”, “black cat”, and so on.

If we concentrate on the mapping between substance properties and classification properties we have pretty much the same situation as for instances and real kinds. Even if some mapping exists, it is not one-to-one and it presents various exceptions. Thus, for instance, as from [**63**], the (substance) properties that we use in recognition are often quite different, and much more complex from the classification properties we use to describe substances. There is also a further interesting twist. While most classification properties map to substance properties, this turns out not to be the case for *social roles*. Social roles are real kinds that in KR are modelled as properties. Social roles are real kinds as a consequence of the fact that their members, e.g., doctors, manifest similar behaviours. No matter the concrete individual who is playing the role of doctor, given a certain situation, her activities will be similar to those of other doctors, as they would be needed to carry out their duty.

3.5. An ontology of (Recognition) Abilities

Figure 1 introduces *RAO*, for *Ontology of (Recognition) Abilities,* a very early version of an ontology which organizes the notions introduced above. RAO must be understood as follows. As from the root, we concentrate on substance concepts taking them to be abilities as well as the representations generated by these abilities. Nominal kinds are added (in dashed lines) for completeness meaning by this that they can be thought as extreme cases of substance concepts with no recognition ability.

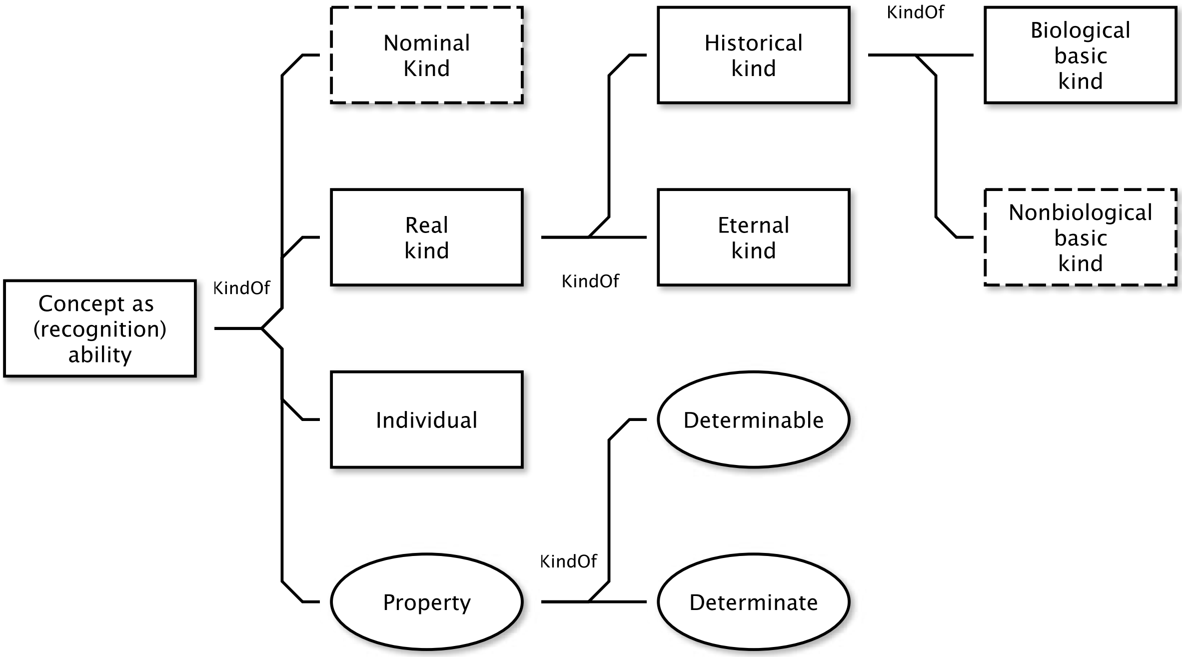

Fig. 2. The RAO ontology.

The structure of RAO is motivated by causal factors, as defined in the rest of this section. Thus, with individuals we have the following:

a) *A natural conservation law.* Individuals have the ability to preserve their properties from day to day [**72**]. Take for instance a person. If she has brown eyes, is tall, is a good tennis player and knowledgeable about informatics, it is likely that she will have these same traits also tomorrow. A similar argument applies to the other kinds of individuals, e.g., to artefacts.

It is important to observe that the causal factor of individuals works very much in the same way as the physics law of the conservation of energy.

The members of real kinds share an empirically observable connection grounded in some law. Real kinds are taken to be the union of historical kinds and eternal kinds. The connections characterizing historical kinds are provided by the possession of one or more of the following four causal factors:

b) *Being the result of a copying activity*. In this case, historical kinds share determinate properties because of some form of previous "copying" activity. We say that a substance B is a "copy" of a substance A, or that B is modelled on A, or that B is a reproduction of A. B can be a true copy of A, as in the case of genes and viruses. B can also be an indirect copy of A resulting from a wider reproduction process. Thus for instance the heart of a person is an indirect reproduction of the heart of her parents while an artefact is another form of indirect reproduction from some abstract model.

c) *A function.* In this case, historical kinds are associated with a function which defines their purpose. This property is possessed in particular by artefacts, and its concrete appearance is often influenced by the cultural context [**72**]. Chairs for instance are defined by the function of allowing people to sit on them, and Japanese chairs are very different from European chairs. Social roles, e.g. mother, are examples of human functions.

d) *A similar training.* In this case, historical kinds are living beings, e.g., persons, who have characteristics or skills that are transferred across generations through training. Example kinds are socially constructed substances such as roles, e.g., doctors and bakers.

e) *A homeostatic mechanism.* For instance, the members of biological species can be seen as "*… homeostatic systems [...] amazingly well-buffered to resist change and maintain stability in the face of disturbing influences*" [**70**]. The key observation here is that, despite having many different properties, the members of a species remain stable and relatively similar in time (e.g., adult weight, internal temperature and so forth). This is because species evolve as a result of continuous adaptation and, at the same time, of the necessity for the various genes in a gene pool to be compatible with one another.

The members of biological basic kinds possess factors (b) and (e) while the members of historical kinds that are held together according to (b) or (c) or (d), or their combination, can be grouped in the catch-all category of non-biological kinds.

Finally, eternal kinds can be characterized by the following causal factors:

f) *An inner structure or a single underlying cause.* Eternal kinds have a sort of real essence that can be discovered by empirical investigation.

Notice that the above list of properties, and therefore RAO, is neither claimed to be final nor complete. It is a first characterization that organizes the state of the art and which can be further extended. Among others, an open research issue is whether non-biological kinds can be divided into more fine-grained categories according to some specific applications of (b), (c) or (d). For instance, a possible subordinate category of non-biological kinds could be the "artefact kind", whose members are alike due to a special case of (b), which should be applied in a non-biological sense and (c), applied as in the chair example above.

3.6. From classification abilities to recognition abilities – a case study

Starting from RAO it is possible to devise the beginning of a methodology for identifying which classification concepts from existing ontologies correspond to substance concepts. In this section we show how this can be done by mapping RAO to the light version of DOLCE[16], i.e., DOLCE-Lite, and to the PIZZA domain ontology[17]. DOLCE-Lite provides a large repertoire of very abstract concepts while the PIZZA ontology classifies more concrete concepts. The resulting mapping is depicted in Figure 2 below.

[16] http://www.loa.istc.cnr.it/old/DOLCE.html
[17] http://protege.stanford.edu/ontologies/pizza/pizza.owl

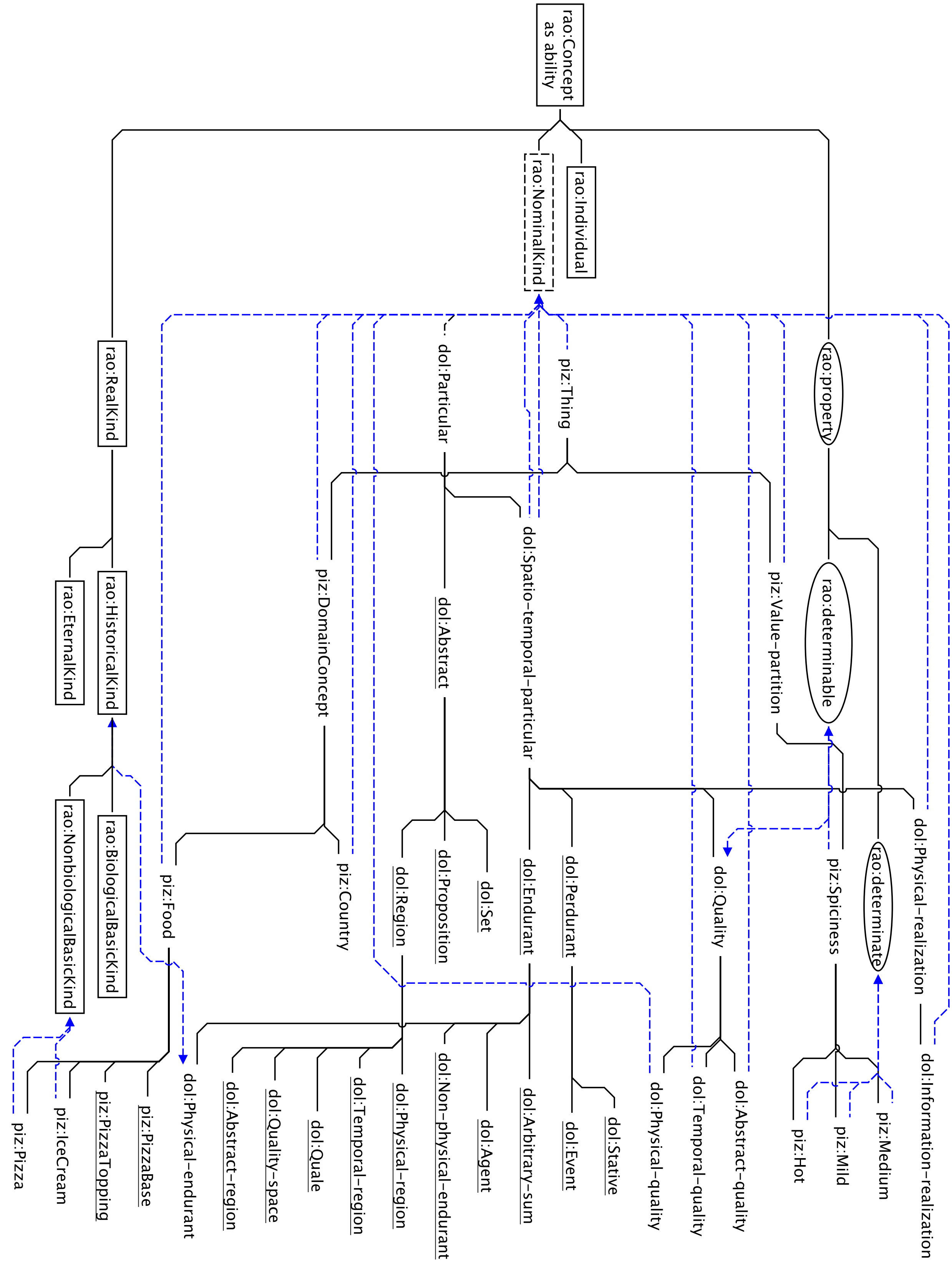

Fig. 3. Mapping RAO to DOLCE and the PIZZA Ontology.

Due to lack of space we have reported our results only for the first three levels of the input ontologies. The RAO concepts are identified with boxes (kinds/individual) and ellipses (properties). `dol:`, `piz:`, `rao:`, stand for concepts coming from DOLCE, PIZZA and RAO, respectively. The underlined terms denote classes which are not mapped. Dashed edges denote the mapping, where two arrows mean equivalence and one arrow means subsumption.

Focusing on DOLCE-Lite, 36% of its classes are mapped to RAO. Two classes are mapped with an equivalence relation. The first is `dol:PhysicalEndurant`, which is mapped with `rao:RealKind`, the second is `dol:Quality`, which, according to our current understanding, maps to `rao:determinable` category. 28% of the DOLCE classes (78% of all the mapped classes) are mapped to `rao:NominalKind` via a subsumption relation. As an example of the kind of reasoning which motivates the mapping with nominal kinds, consider the `dol:SpatioTemporalParticular` class, (Figure 2), which is labelled as nominal. The reason for this choice is that the class "spatio-temporal particular", as from DOLCE-Lite, is a "*dummy class for optimizing some property universes*". Here we may find concepts such as endurants (i.e., `dol: Endurant`), perdurants (i.e., `dol:Perdurant`) and physical realizations (i.e., `dol:PhysicalRealization`), which cannot be grounded in a causal factor.

85% of the PIZZA ontology classes are successfully mapped, resulting in eleven subsumption relations. For instance, `piz:hot` and `piz:mild` are subsumed by `rao:determinate`. These concepts map well to our definition of determinate. They are distinguishable because of a set of common features (in this case, features related to spiciness, i.e., `piz:spiciness`, which perfectly map to `rao:determinable`). Similarly, classes like `piz:Pizza` and `piz:IceCream` are subsumed by `rao:NonBiologicalBasicKind`. In fact, if, as from the experiment by Rosch, `piz:Food` is a nominal, pizzas are artefacts grounded in a "copying" causal factor. 38,5% of "pizza" ontology classes (45,5% of all the mapped classes) are mapped to nominal kinds via subsumption. These classes are clear examples of conventionally defined, theory driven, groupings (e.g., `piz:Value-partition`).

This exercise provides the highlights of a general methodology for identifying substance concepts. At the same time, it also provides evidence of the fact that there is a need to further refine RAO: 64% of the concepts from DOLCE-Lite and 15% of the concepts from the PIZZA ontology are unmapped. Let us consider some examples. For instance, following Millikan's suggestion that *events* are substance concepts, the classification concept `dol:Perdurant`, should be mapped to a new recognition ability. As another example, the concept `dol:Endurant` and its more specific concepts should be linked to RAO through several more classes which are more specific than the ones we have provided so far. For instance, `dol:AmountOfMatter`, is a `dol:PhysicalEndurant` which can be successfully mapped to what Millikan would call "stuff", i.e., a kind of `rao:EternalKind`. A further issue is whether quality spaces, as defined in DOLCE, can be used as kind of "determination dimensions" and whether they can be employed to guide the linking between determinables and determinates. This analysis is essential to explore the relation between the concept of "determinate" and the concept of "quale" thus providing a contribution to the modelling of properties [**82**]. All these examples define a path of research that will allow us to provide a clear mapping of how, in practice, we could deploy concepts which implement a recognition function, a classification function, or both in an integrated manner.

3.7. Implications on AI systems

The differences between substance concepts and classification concepts highlighted above provide interesting insights on how to build integrated AI perception and reasoning systems. The general question is which concepts should be selected for artificial recognition (e.g., vision, sensory) systems and how we should treat them in relation to the classification concepts which are represented inside KR systems, for instance as elements of ontologies. How to create a mapping between these two kinds of concepts is a very well-known open problem, i.e. the *semantic gap* problem, which has been solved only in very particular situations [**62**].

From the point of view of recognition, substance concepts are the ones where most efforts should be concentrated, as they are the concepts that, thanks to the causal factor in which they are grounded, have a more immediate mapping with their appearance. Thus for instance, the recognition of cats and dogs from a set of pictures will need to handle less diversity than the recognition of animals. Dually, the function of classification is very valuable and worthwhile with nominals, not only because it allows to organize substance concepts (thus delegating them the recognition function) but also because the definitions of nominals are very stable, with essentially no exceptions. These definitions can in fact be provided as sets of properties with no need to map with the complexity and infinite variety of the real world. Animals, for instance, are best defined and thought of as cats or dogs or …, without trying to recognize them in terms of the sensory input.

A further interesting situation arises when there is a need to recognize some specific sub-kinds of real kinds, for instance when we need to distinguish cats by their color. This type of categorization turns out to be useful in substance recognition from sensors as it allows to apply to recognition the compositionality of meaning which is intrinsic in knowledge representation. The work described in [**103**] is a rather successful experiment in this direction. We are aware of the consequences of what discussed by Millikan in her critique to Fodor [**70**], namely that the compositionality of properties in KR may not correspond to the compositionality of properties in recognition. Thus, even if I know how to recognize a person and how to recognize a hat I may fail to recognize that very same person wearing that very same hat. However, our early attempts to apply compositionality to recognition hold a lot of promise, the main reason being that they exponentially decrease the cost of training of the learning components [**103**].

The big challenge is how to manage and reason about those substances for which we have both classes and substance concepts. The problem is that the static definition of classes does not fit well with the variance of appearance of their substances, and therefore, with the corresponding variability of substance concepts. A long discussion on this issue leading, among other things, to the distinction between *conception* and concept can be found in [**75**]. As a small example, it is essentially impossible to provide a definition of what the real kind "cat" is, as what makes a cat "a cat" is its causal factor while its apparent characteristics change in time. A discussion of this issue is out of the scope of this chapter. Our general approach, which will be the topic of a follow-up work, is that the (substance) properties which are chosen when defining, e.g., the class “cat”, should not be fixed a priori but, rather, should be *adapted* at run time as a function of the goal which the definition must serve, for instance the alignment with what is being recognized by a vision system.

Chapter 4

4. Teleologies: Objects, Actions and *Functions*

A crucial characteristic of humans is their ability to build and exploit representations of what they perceive, what we usually call *the world*. Such representations usually consist of complex combinations of *concepts*, where we take a concept to be *an abstract idea generalized from particular instances*. However, the very notion of concept is controversial [**40**]. Thus, for instance, on one side we have the *Biosemantics* approach which takes a concept to be a device and a representation supporting certain biological processes, in particular, perception (e.g, human vision) [**87**], while, on the other side, we have the so-called *Descriptionist* approach which takes a concept to be a class, namely a set of instances characterized by some shared set of properties, as the basic construct enabling knowledge representation, classification and reasoning. The former and latter notions of concept underlie the work in Computer Vision (CV) [**35**] and in Knowledge Representation (KR) [**102**], respectively.

The work desribed in the previous chapter shows how the two notions above have different characteristics and calls them *substance concepts* and *classification concepts*, respectively. Substance concepts represent what we perceive and, therefore, are characterized by a notion of *perceptual identity* (and diversity) while classification concepts represent what we reason about and, therefore, are characterized by a notion of *reasoning identity* (and diversity). *While perceptual identity captures invariance over the occurrences of what we perceive, reasoning identity captures invariance over the occurrences of what we reason about*. Thus, for instance, we recognize a *rock* as being such depending on what we perceive, while we reason about the same rock as an *obstacle* when it is in our way, or as a kind of *weapon* when throwing it at someone.

In this chapter we show how to integrate substance and classification concepts into a hierarchy of increasing abstraction from what is perceived. Thus, at the first level, we have *objects* (which roughly correspond to substance concepts), which are *representations of what is perceived* (e.g., *a car*); at the second level we have *actions*, which *represent how* *objects change in time* (e.g., *move*, where, among others, cars can move); while, at the third level, we have *functions* (which roughly correspond to classification concepts), which represent *the expected behavior of objects* as it is manifested in terms of "an object performing a certain set of actions" (e.g., *a vehicle,* where vehicles, e.g., cars, can perform many actions, e.g., move and stop). The intuition is that, by performing *actions*, *objects interfere* with other objects, this being the basic mechanism by which the world evolves. In this perspective, *functions model the expected interference among objects*. Object interference, and therefore function, is captured via the notions of *producer* and *consumer*, where an object is a producer when it performs an action affecting another object and a consumer when it is affected by it.

The patterns by which producers affect consumers provide the basis for the construction of *Teleologies*.[18] *Ontologies*[19] are defined as explicit formal specifications of the terms in a domain [**45**]. The same definition can be applied for teleologies but with the proviso that teleologies focus on function and on how a chosen representation fits a certain purpose, this being the basis for a general model for the *diversity of knowledge* [**44**]. In this respect, the distinction between objects and their multiple functions is the *first source of heterogeneity*, modeling the diversity between the representation of what we *perceive* and the representation of what we *reason about*. The *second source of heterogeneity* is our ability to represent and reason about what we perceive at different levels of abstraction, as function of the problem to be solved. Thus, for instance, I can describe a person as *moving her legs*, as *walking*, or as *moving*, depending on my focus.

This work is a first step towards a solution to the problem of managing knowledge diversity not in the sense that we are able to define the ultimate teleology which can be reused in general (which is impossible) but, rather, in the sense that we provide the basis for a general methodology for the construction, integration and/or adaptation of data and knowledge coming from multiple heterogeneous sources. We organize the chapter as follows. In

[18] The word teleology builds on the Greek words *telos* (meaning "end, purpose") and *logia*, (meaning "a branch of learning").
[19] The word *ontology* builds on the Greek words *ont* (meaning "being") and *logia*, (meaning "a branch of learning").

Section 4.1, we introduce objects, actions and functions. In Section 4.2 we introduce producers, consumers and producer – consumer (PC) patterns. In Section 4.3 we introduce the three PC pattern transformations which can be used to *reduce* one pattern to another pattern, *preserving the pattern intended meaning*. In Section 4.4, we provide a small example of how to build and how to adapt a teleology, using the pattern transformations from Section 4.4, *adaptation* being they key for handling diversity in knowledge. Finally, in Section 4.5, we provide the related work.

4.1. Object, Action and Function

We live immersed in a *spatio-temporal continuum* where space and time are the *a-priori* forms of perception [**10**]. We do not perceive space or time, but anything we perceive is *part* of a precise spatial or temporal ordering, and fills it. We perceive these parts through *encounters*, namely events during which such parts *manifest* themselves to an observer. We call such parts, *substances*, where, as from [**75**], "… *substances are those things about which you can learn from one encounter something of what to expect on other encounters, where this is no accident but the result of a real connection*".

People represent substances as *concepts*. However, the mapping between substances and concepts is not one-to-one. Thus, I may perceive a substance as a cat that I am trying to avoid hitting, as *my* cat, as an animal, or as an obstacle. Even more, there are substances for which we do not have a concept. One such example, the part of the mountain that I can see from the window of my office. *Concepts represent those parts of the spatio-temporal continuum that are relevant to us, in the way which is most convenient for us,*[20] *as the world where we live*.[21] But if the world, as we perceive it, *is* representation, and if there is a certain degree of freedom in what we represent and in how we represent it, is there a general principle to which we all adhere and that allows us to live in the same world, or at least in worlds which are very similar?

[20] The concepts we use are also largely influenced by our language, culture, history, place where we live, and many other contextual factors.
[21] Interestingly enough, the ancient Latin word for world is mundus, meaning "clean, elegant", itself a translation of the Greek word cosmos, meaning "orderly arrangement".

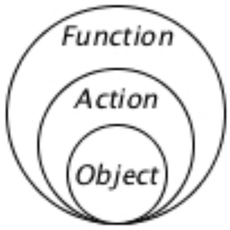

Fig. 4. Object, Action and Function.

Our answer to the above questions is based on a distinction among three types of concepts, namely *objects*, *actions*, and *functions*, which represent what is perceived, across encounters, at increasing levels of abstraction (see Figure 1).

We take *Objects* to be those concepts which *represent substances, i.e., what is perceived* across encounters. Examples of objects are: cats, cars, rivers. As from described in the previous chapter, an object can be thought as the set of all of the representations of how the same substance "fills" space, any time we encounter it. Objects can be *individuals* (what in KR we call instances, e.g., my cat Garfield) or *kinds* (i.e., generic instances of what in KR we call classes, e.g., any cat that I can encounter while walking). Objects are *first level abstract representations* in the sense that they abstract over multiple occurrences of the same substance (as recognized during encounters) and collect them in clusters (one cluster per object). An object, e.g., "a cat", is nothing else but the set of representations of all the times we have perceived (e.g., seen) it.

We take *Actions* to be those concepts which *represent how objects change in time*. Examples of actions are: running (performed by, e.g., cats), carrying (performed by, e.g., cars) and flowing (performed by, e.g., rivers). As with objects, actions are generated any time we encounter a substance. Actions are *second level abstract representations* in the sense that they abstract over multiple occurrences of changes in time of a substance (as recognized during encounters) and collect them in clusters (one cluster per action). An action, e.g., "running", is taken to be the set of representations of all times we have perceived a running object, e.g., "a cat" (or "a dog"), where the representation of "a running cat" or ("a running dog"), is a temporal sequence of "cat" ("dog") occurrences. Notice how actions are independent of the specific object carrying them out; objects are abstracted away to keep track only of what changes.

We say that *a certain object O performs a certain action A* when we perceive O subject to the change described by A. Notice that there are only so many actions that can be performed by an object. For instance a car cannot be used to fly. We capture this intuition by saying that any object O is associated to *a set of admissible actions* $\{A\}_{a:O}$, where A is an action and "a:O" stands for "admissible for O". We have the following:

$$\mathrm{AaO(O)} = \{\mathrm{A} \mid \textit{for any } \mathrm{A} \in \{\mathrm{A}\}_{a:O}\}$$

$$\mathrm{OaA(A)} = \{\mathrm{O} \mid \textit{for any } \mathrm{O} \textit{ such that } \mathrm{A} \in \{\mathrm{A}\}_{a:O}\}$$

where AaO and OaA are to be read, respectively, *(admissible) Actions of (Object)* and *Objects of (admissible Action)*. Thus, for instance we have AaO(car) = {move, transport, trap, …} and OaA(move) = {car, bus, person, table, …}. For any object, its set of admissible actions, as well as its set of not admissible (*inadmissible*) actions, is infinite, as infinite are the ways in which an object can evolve in time. At the same time, an admissible action can be performed only under certain contextual conditions. For instance, a car needs gas to run its engine and move around. Admissible actions are similar in spirit to Millikan's *abilities* and somewhat related to the notion of *affordance,* as formalized by Gibson [**39**] and then taken up in various contexts, see, e.g., [**94**], [**82**]. The crucial difference is that affordances are related to what an environment enables an object to do, more than what an object is, by itself, able to do.

Certain admissible actions occur quite rarely. For instance, a car can be used as a trap for certain animals, but this is rather unusual. Many admissible actions are instead quite common. Thus for instance, a car usually moves around and transports people, while a person usually eats, sleeps and walks. Similarly, at school, quite often an older person (that we call "a teacher") explains some topic to a younger person (that we call "a student"), she gives homework, she grades it, and so on. The fact that certain sets of actions are repeatedly performed by the same object allows humans to make predictions about the future behaviour of objects and to reason about this. We formalize this fact through the notion of *function*. *The function of an object formalizes the behavior that an object is expected to have*. This expected behavior may be due to the object's *purpose* (as it is the case with artifacts, e.g., a car) or to its *role*, for instance in the world and society (as it is the case with living organisms, e.g., a cat, a tree or a person). Sometimes the word used to denote a function is the same used to denote the object performing it (e.g., car, cat); in many cases language provides dedicated words (e.g., teacher, parent) possibly with a negative connotation (e.g., obstacle, enemy, garbage).

We capture this intuition by saying that *an object can perform one or more functions*, where a *function is defined as a set of actions*. Let O be an object and $\{F_O\}_{p:O}$ a set of *proper functions* F_O (where *proper* emphasizes the fact that these are functions which are "expected"). Then we have the following ("p:O/p:F_O" stands for "proper for O/F_O"):

$$FpO(O) = \{F_O \mid \textit{for any } F_O \in \{F_O\}_{p:O}\}$$
$$OpF(F_O) = \{O \mid \textit{for any } O \textit{ such that } F_O \in \{F_O\}_{p:O}\}$$

with:

$$ApF(F_O) = \{A \mid \textit{for any } A \in \{A\}_{p:Fo}\}$$
$$FpA(A) = \{F_O \mid \textit{for any } F_O \textit{ such that } A \in \{A\}_{p:Fo}\}$$

where: FpO and OpF are to be read *(proper) Functions of (Object)* and *Objects of (proper) Function*, respectively, and ApF and FpA are to be read *(proper) Actions of (Function)* and *Functions of (proper Action)*, respectively. Thus, for instance, we have ApF(vehicle) = {move, transport, …} and FpA(move) = {vehicle, person, …}. Obviously, $\{A\}_{p:Fo} \subset \{A\}_{a:O}$. $\{F_O\}_{p:O}$ is assumed to be finite. The finiteness of $\{F_O\}_{p:O}$, in the case of artifacts follows from the fact that we build artifacts with a specific *purpose* in mind. The finite functionality of living beings is not connected to the fact that we know their purpose but to the fact that they have shape and behavior which comes from nature and is replicated through reproduction, and from the fact that we model it as their *role*. It is a fact that (the functions of) living beings are more easily recognized and perceived than (those of) artifacts [**90**]. At the same time, *$ApF(F_O)$ contains (again) a possibly infinite number of actions*, this meaning, in practice, that there is always the possibility to characterize a specific change of an object/function as a new action. If language allows us to precisely denote an object or a function with a word, a precise characterization in terms of its possible actions is impossible.

As from Figure 1, functions are *third level abstract representations* in the sense that they abstract over multiple occurrences of objects performing actions (as recognized during encounters) and collect them in clusters (one cluster per function). A function, e.g., “mover”, consists of the set of representations of all the times we have perceived an object performing a certain expected action, e.g., “a running cat” or “a walking person”.

4.2. Producer – Consumer Patterns

We model the interaction between objects, actions and functions using patterns like the one in Figure 2. More precisely, the pattern in Figure 2, is a specific instance of what we call an *OAO (for Object-Action-Object) pattern*. In OAO patterns, round boxes represent objects, arrow boxes represent actions and square boxes represent functions. t_1 and t_2 define start and end of the action. The specific pattern in Figure 2 instantiates what in natural language we would describe as *‘a car transporting a person’*. In Figure 2, *Transport* is the action, *Car* is the producer object, *Person* is the consumer object, *Vehicle* is the function performed by the producer while *Passenger* is the function performed by the consumer. The intuition is that *an object plays the function of a producer when it performs an action affecting another object, possibly itself, and that the function of consumer is played by the object being affected by this action*. The intuition of what *“an action affecting another object”* means is that an object is associated with a *state* and that this state changes any time an object is a consumer. The state of an object includes its physical properties (e.g., position, shape, beauty), the actions it performs (a subset of the set AaO(O)), namely the patterns where it is a producer and the state of its functions (being, e.g., active, idle, malfunctioning, sick, in love, angry, …).

Fig. 5. ‘OAO pattern – A car transporting a person’.

In Figure 2, the arrows from/to objects represent two crucial aspects of the model:

1. an object is *always* both a producer and a consumer, being embedded in the continuous evolution of the world;
2. an object may occur in multiple OAO patterns while an action may occur only inside a single OAO pattern.

OAO patterns have the form of the pattern in Figure 2 with three possible variations: *(i)* producer and consumer may be dropped when the relevant concept is not lexicalized or it is lexicalized with the same term as the object, *(ii)* the producer and the consumer may be the same object (as in, e.g., *“a person walking”*), in which case the pattern forms a cycle, and *(iii)* the action may be in passive form (as in, e.g., *“a person transported by a car”)*, this being useful to compose OAO patterns, as described below.

4.3. Basic Patterns

OAO patterns model the world evolution. Clearly there are infinitely many such patterns. However, that there are only *four primitive OAO patterns* and corresponding *primitive functions*, which model *the world evolution basic modalities*. These patterns model *(i)* how new objects are conceived, *(ii)* how they are realized and *(iii)* how they are destroyed, and *(iv)* how they affect the state of other objects. The first such pattern, called *Conception,* or *OCO pattern,* defines the function *conceiver.* See Figure 3.

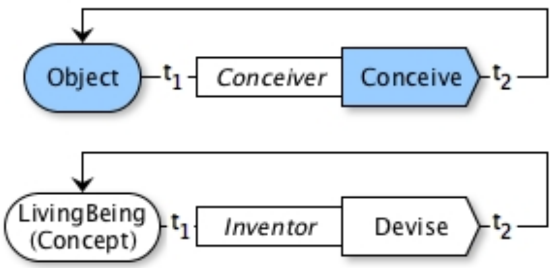

Fig. 6. 'OCO - Object conception pattern'.

Conception represents the process by which a concept, which was not lexicalized before, is conceived. Concept conception amounts not only to the creation of the new concept in the mind of a living being, e.g., a person, but also to the creation, via perception, of the causal relation between the concept and the substance being perceived. For instance, Johannes Gutenberg in 1439 conceived the first printing press. Notice that living beings are the only objects which can conceive new functions and that they do this by reflexively "enriching" their state with a new concept, where the word in parenthesis in Figure 3 represents the concept being conceived.

The second primitive pattern, that we call *Realization*, or *ORO pattern*, defines the function *maker*. See Figure 4. The realization of an object coincides with the moment when an object assumes its (recognizable) identity in the world. For instance, my car was realized in 2014, 15 days before I bought it. For an object to be realized, its defining functions must have been previously conceived. Figure 4 depicts three important specializations of the pattern, namely: *(i)* the capability of living beings to procreate, *(ii)* the manufacturing skills by which a factory (or a person) can realize objects, e.g., a press, and *(iii)* the ability of "intelligent" machines to assemble new objects.

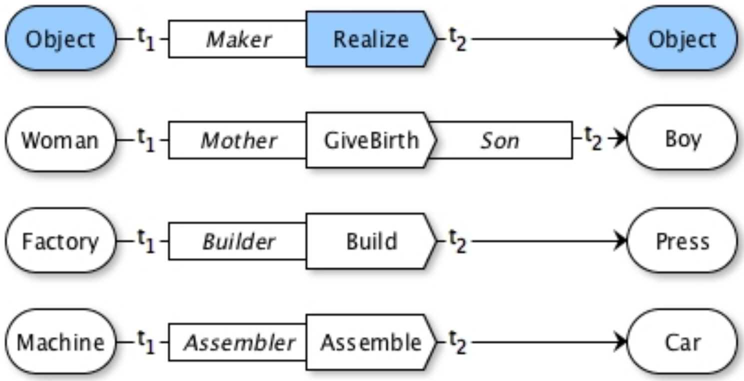

Fig. 7. 'ORO - Object realization pattern'.

The third primitive pattern, called *Destruction,* or *ODO pattern*, defines the function *destroyer*. See Figure 5.

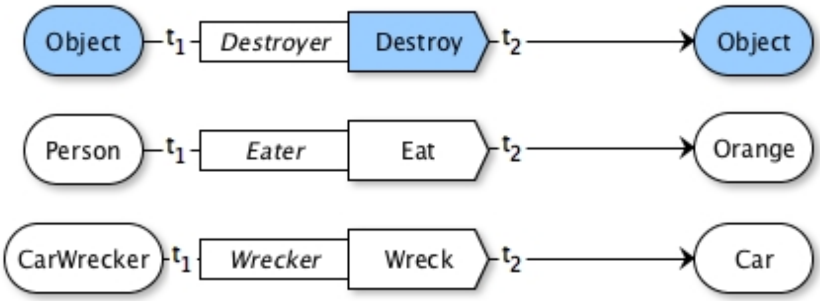

Fig. 8. 'ODO - Object destruction pattern'.

ODO patterns represent the process by which an object "disappears" because losing its identity. This is the inverse pattern of realization. Thus, eating an orange and a car wrecker destroying my car are both instantiations of this pattern.

The last primitive pattern, called *Service Provision,* or *OSO pattern*, defines the functions *provider* and *receiver*. See Figure 6.

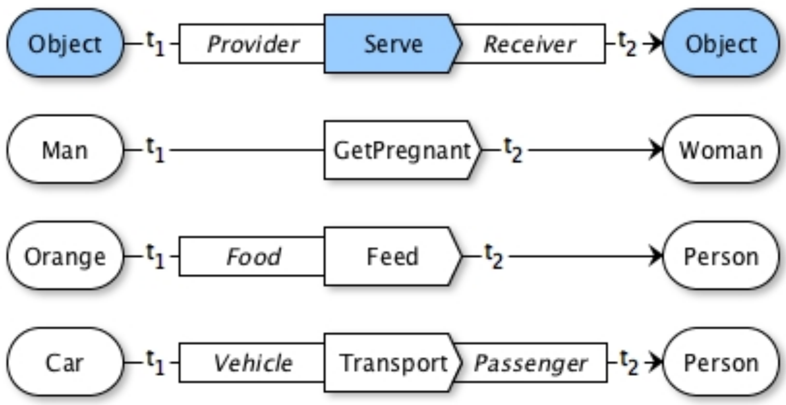

Fig. 9. ‘OSO - Object service pattern’.

This is the pattern that models the process by which any two objects may affect one another. The specialization patterns represent some important specializations, namely: *(i)* the inception of a living being (it is a “service” in the sense that the state of the consumer is changed), *(ii)* a living being acquiring the energy needed to live by eating, and *(iii)* an object affecting the state, e.g., the position, of another object.

The key observation is that *the world evolution can be modeled by suitably specializing/generalizing and/or by composing OAO patters to produce complex patterns.* We call the patterns obtained in this way, *Producer – Consumer (PC) patterns*. The figures above provide examples of specializations. PC patterns compose OAO patterns by making the consumer of a former pattern coincide with the producer of a latter pattern.

4.4. Complex patterns

PC patterns can produce graphs of arbitrary complexity. The simplest versions of PC patterns are OAOAO patterns. These patterns are of particular relevance since they represent how the application of the function in the first OAO pattern provides input to the function applied in the second OAO pattern. Examples of relevant OAOAO patterns are: *reproduction,* which models how something is constructed as a copy of some object, *transformation* which models how objects change their function (e.g., a car transformed into a cage or a rock into a chair); *undo* by which the second function, under certain conditions, cancels the effects of the first function, this allowing to define the *inverse* function; *service composition*, which models how complex services are provided, online or in the world, and so on. As an example of OAOAO pattern, Figure 7 depicts *Creation*.

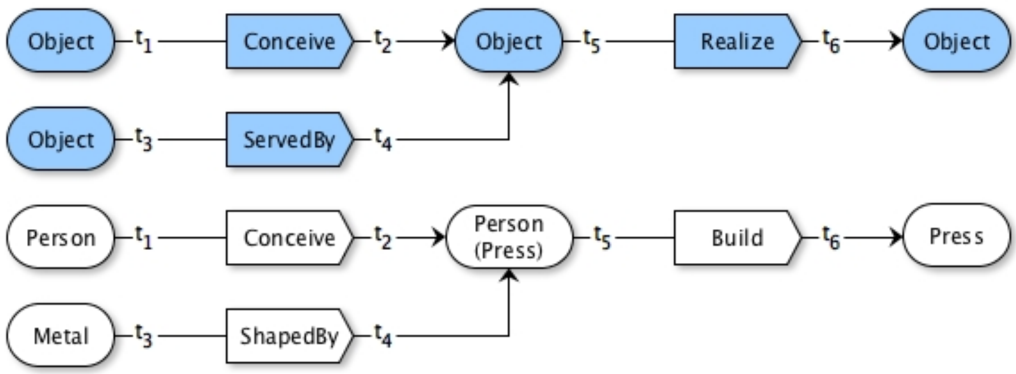

Fig. 10. 'Creation compound pattern'.

Creation allows for the construction of a new type of object (e.g., presses, in the case of Gutenberg's press). Notice that the central object has two inputs which may occur at different times. The first observation is that the double input captures the fact that nothing can be created but can only be "transformed" from something else. The second is that what we represent is always an approximation, e.g., we could further complicate the above pattern to consider more materials, human effort, and so on.

4.5. Pattern transformations

PC patterns allow for a *uniform* representation of the spatio-temporal continuum. However, they do not give us the means for univocally representing this continuum as (the evolution of) the world where we live. And this could not be the case! As from Section 2, there is a many-to-many mapping between substances and substance concepts (i.e., objects) and, as from the previous chapter, there is a many-to-many mapping between substance concepts and classification concepts, this latter intuition being captured by the two relations FpO and OpF introduced in Section 2. These two mappings are at the core of the phenomenon of knowledge diversity and formalize two levels of freedom in the representation of the spatio-temporal continuum. The first, from substances to objects, corresponds to the many possible ways in which the same substance can be *perceived* as a certain object. The second, from objects to functions, corresponds to the many possible ways in which the same object can be *reasoned about* in terms of the function it performs.

Our solution to the problem of managing diversity in knowledge is to exploit the uniform representation provided by PC patterns and define a set of *PC pattern transformation operators* that allow, given any two PC patterns, to reduce one to the other, *preserving their intended meaning*. The intuition is that the existence of such a reduction will be evidence that the two PC patterns represent the same or similar configurations of substances, and the contrary when this is not the case. Notice how this *does not avoid* the possibility of multiple descriptions of the same (set of) substances, but it *does provide* a systematic approach for absorbing diversity.

We have identified three PC pattern transformation operators, that we call *Granularity*, *Abstraction* and *Partiality,* where the combined effects of these three operators allow to transform patterns, still preserving the underlying semantics.[22]

The *Granularity operator* allows for two types of transformation: *(i)* substituting *parts* with *wholes* or vice versa, and *(ii)* substituting *more specific* concepts with *more general* concepts or vice versa. The examples in the previous section are all applications of this operator. Figure 8 provides a further example where the pattern at the bottom is obtained from the one at the top via a *whole-part* transformation and a *more general-more specific* transformation.

[22] A general formalization of this intuition, not provided here for lack of space, will be provided in a follow-up work and will be based on the work described in [**43**], which provides a formalization of the problem of theory transformation in terms of abstraction operators.

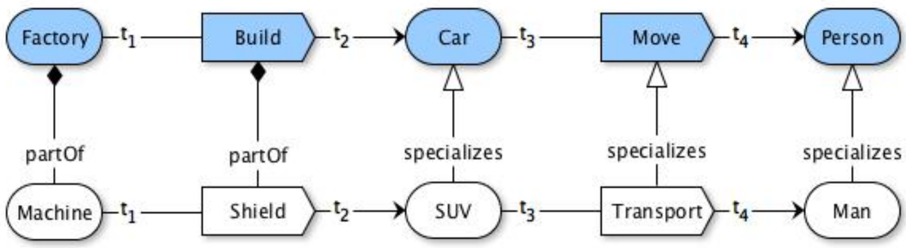

Fig. 11. ‘Granularity operator’.

The *Abstraction (concretization) operator* enables the (un)folding of concepts, towards a less (more) fine-grained structure; making some concepts implicit (explicit). Figure 9 provides an example of abstraction (top) and one of concretization (bottom). Notice how an Action Object Action pattern gets reduced to a single action and vice versa.

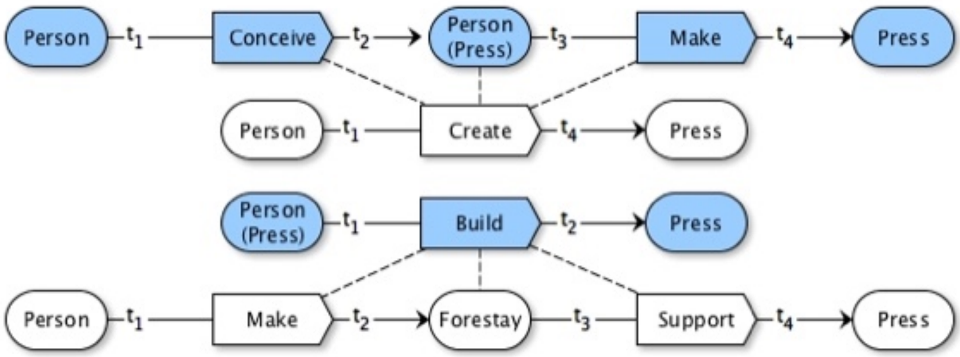

Fig. 12. ‘Abstraction/concretization operator’.

The granularity and the abstraction operators output PC patterns. This is not the case for the *Partiality operator* which outputs two patterns, namely *(i)* patterns containing only actions and functions, that we call *AA patterns*, and *(ii)* patterns containing only objects and functions, that we call *OO patterns.* The *Partiality operator* achieves this result by dropping all elements of certain kinds (O or A). Consider for instance Figure 10, where the top pattern is obtained from the middle one by dropping objects (and functions) and where the bottom pattern is obtained from the middle one by dropping actions. Notice how AA patterns focus *on the process,* as it is done, e.g., in planning and activity recognition (see, e.g., [**78**]) while OO patterns focus *on objects and their functions*, as it is done, e.g., in

Schema.org [**5**]. The choice of where to focus depends on the purpose of the modeling. We call the union of PC patterns, OO patterns and AA patterns, *teleology patterns*, to capture the idea that any representation is chosen to best fit the problem to be solved.

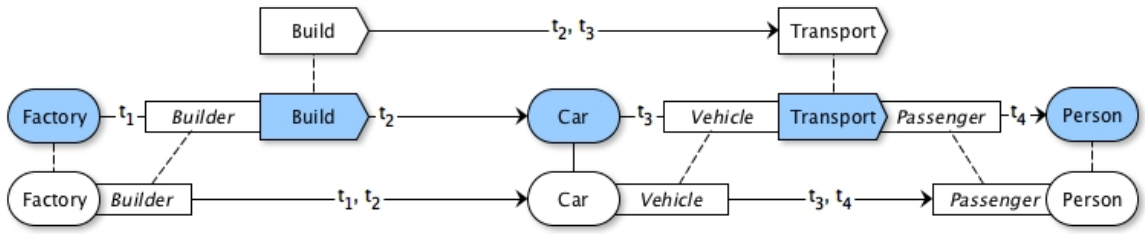

Fig. 13. 'Partiality operator'.

4.6. Teleologies

Teleology patterns are the basic constituents of *Teleologies*. Teleologies are nothing else but structured organizations of teleology patterns where the horizontal dimension is given by the teleology patterns themselves while the vertical dimension follows the "usual" more/less general hierarchy. In this respect the name "teleology" has a double motivation as, on one side, teleologies allow for the explicit representation of function, while, on the other side, are organized as needed for the problem to be solved.

The top part of teleologies is organized in two levels. The root is "Concept", meaning that the focus is on representation rather than on what is the case, as it happens in (upper level) ontologies (where, for instance, the root of *DOLCE* is "Thing" [**25**] and the root of *SUMO* is "Entity" [**79**]). In turn, the root has three children, namely "Object", "Action" and "Function", the last being then further subdivided into "Producer" and "Consumer". Furthermore, functions, objects and actions are linked by the relations defined in Section 2, i.e., "AaO", "OaA", "FpO", "OpF", "FpA" and "ApF".

Teleologies are designed to satisfy two main properties:

1. to allow for the representation of teleology patterns, as the way *to provide a uniform view of the concepts recognized via perception and the concepts used and derived via reasoning*;
2. to allow for their continuous modification, via pattern transformation operators, as the way by which a teleology can be *adapted to integrate new inputs*, e.g., new concepts needed to represent a new input from perception or from a heterogeneous dataset, or concepts coming from another teleology.

Let us start with the first property. For the sake of argumentation, as an example, we can assume that we have, as initial set of "relevant" concepts, those which are reported in Table 1 and are not tagged with "*". Notice that in Table 1 we have "Person" and "Car" but also "LivingBeing" and "Machine", with the latter two concepts being more general than the former two. It is in general a good practice to use a set of high level concepts as collectors of functions and actions. Thus, for instance, the functions and actions of "LivingBeing" can be inherited by, e.g., "Cat". The idea is to avoid unnecessary diversity as the more general concepts drive the instantiation of their more specific concepts.

Table 2. An example of relevant concepts.

Object	**Function**	**Action**
LivingBeing	{LivingBeing}	{Conceive}
Machine	{Machine}	{Transport}
Person	{Person, Driver, Maker, Passenger, *Rider*}	{Conceive, Drive, Make, *Ride*}
Car	{Car, Vehicle, Transportation}	{Transport}
Motorcycle	{*Motorcycle*, Vehicle}	{Transport}

A snapshot of the resulting teleology is reported in Figure 11.

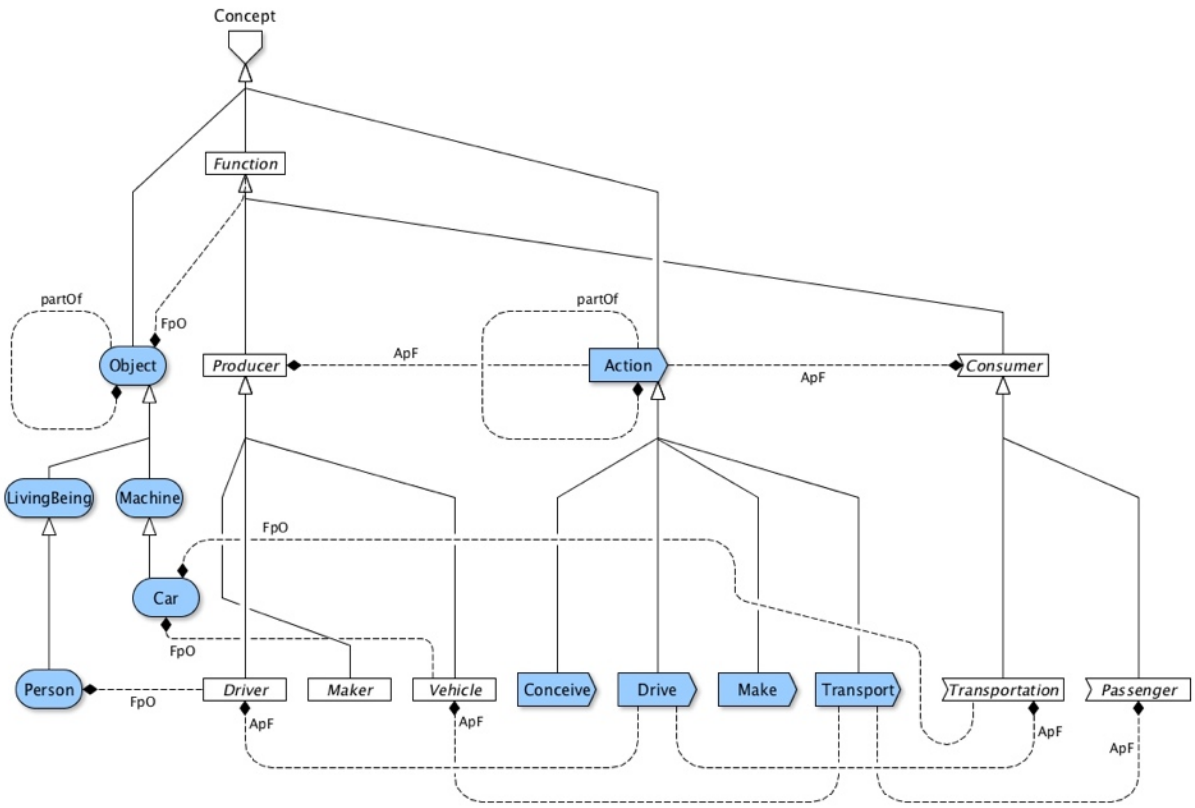

Fig. 14. 'A small example of teleology'.

The white arrows represent *more/less general* relations, the black diamonds represent *associative* relations, e.g., "partOf", "FpO" and "ApF". The "ApF" links in Figure 11 must be read as follows: producers enable the actions in ApF(producer) while consumers are affected by the actions in ApF(consumer). Notice how the PC pattern in Figure 2 is reconstructed via the associative relations linking "Car", "Person", "Vehicle", "Transport" and "Passenger". Notice also how *roles* such as, e.g., "driverOf" or vehicleOf", not represented in Figure 11 for lack of space, are more specific concepts than the relation resulting from the composition of FpO and ApF. Roles are crucial for the representation of OO patterns like the one represented in Figure 10 (bottom).

Let us now see how we can use the same process as above to adapt, e.g., extend/change, the current teleology in the presence of a new concept (for instance coming from another vocabulary). Consider, for instance, the concept "Car", classified in *Schema.org*[23] as a

[23] http://schema.org/Car

“Product” and as a “Vehicle”. This concept is perfectly aligned with that with the same name in the teleology in Figure 11. The only (optional) addition is to add “Product” (as a function, more precisely as a consumer). Consider now the more complex situation of updating the teleology in Figure 11 by adding the object “Motorcycle”, as defined in *Schema.org*[24], as a specialization of “Machine”. “Motorcycle” is not present in Figure 11 nor is there any PC pattern to which it can be connected. The relevant PC pattern(s) can be added by applying the granularity operator, more specifically by specializing the function “Driver” with “Rider” and the action “Drive” with “Ride”. Figure 12 represents a focus on the relevant part of Figure 11 where the new concepts (marked with “*” in Table 1) are added. The resulting teleology is now capable of modeling the PC pattern described by the natural language sentence ‘*a person riding a motorcycle*’.

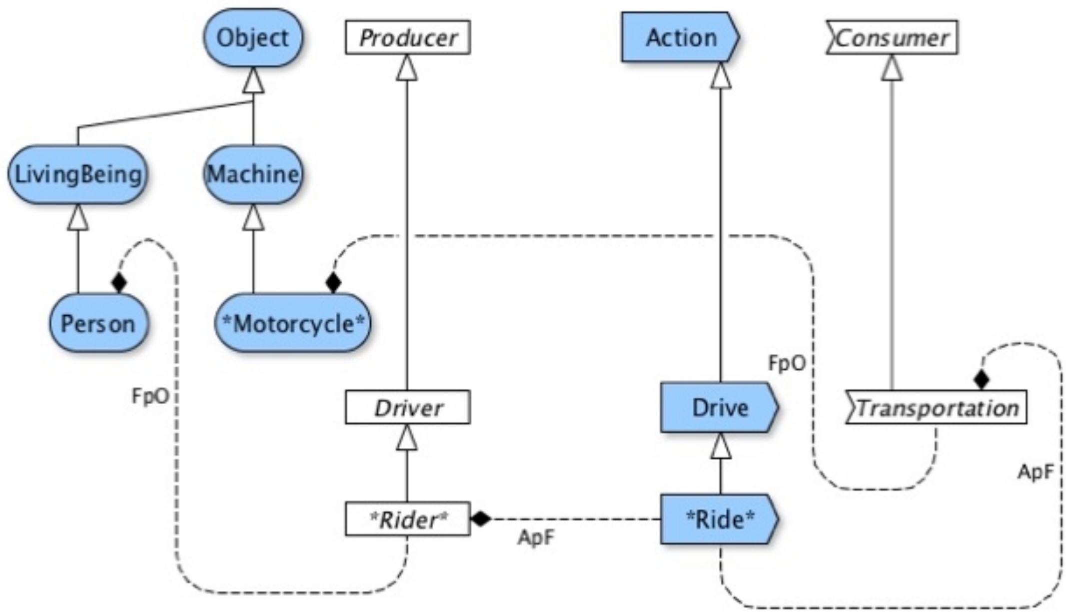

Fig. 15. ‘An example of teleology update’.

[24] https://auto.schema.org/Motorcycle

Chapter 5

5. Tracing and Exploiting Knowledge Diversity via a Causal Model

Our knowledge about the world is entangled with one fundamental issue that must be handled by any representation, namely world change. The central question is: *how can we know something as the same thing despite its change?* Let us take for instance an everyday research item like an article, i.e., a paper. Each encounter with it provides us with multiple different pieces of information. Does the article that is on the online repository is the same article that is on the table of my apartment in the evening? Does the article that is now dusty and sketched is the same article that one week ago was completely clean? Does this article is the same article even after cutting away its title page? Does this article that I use as rough paper for taking some notes is the same article that was written by my advisor one month ago? Does this article remain the same even after being cut into small pieces?

In the context of KR, three are the main state-of-the-art strategies of approaching this challenge (see chapter 2 for a detailed survey). The first strategy is to define properties that something *must have* for being that something, where a property is taken as *a quality, an attribute, or a characteristic describing something*. For instance, for being an article something must have 'pages', a 'title', an 'author', and so forth. For being the same article, something must have the same 'author(s)', same 'DOI', and so forth. Typical information artifacts implementing this strategy are *top-level* (i.e., foundational) *ontologies*, like *Dolce*[25], for very abstract notions, or *core ontologies* like *Vivo*[26], devised for modeling more concrete notions, like 'article' or 'academic article'. The second strategy is to keep-track of all the properties that something *may have*, i.e., that can be present when this something is encountered *in every possible context*. For instance, an article along with 'author', 'title' and 'DOI', may have a 'format', an 'illustrator', a 'review', a 'pagination', and so forth. The same article may have different formats, many reviews, many number of pages, and so forth. Typical information artifacts implementing this strategy are huge

[25] http://www.loa.istc.cnr.it/old/DOLCE.html
[26] http://vivoweb.org/ontology/core

"catch-all" schemas for structuring a large amount of data like *Schema.org*[27] or *DBpedia.org*[28]. The third strategy is to provide the properties that something is *likely to have* over encounters *in a given context*. For instance, an article, in the context of a library catalog, is likely to have a 'location', an 'availability', a 'sound recording' and so forth. The same article is likely to have the same subject, the same author(s), but not always the same location or availability, and so forth. Typical information artifact implementing this strategy are database schemas or domain ontologies like the schema underlying *Osikat*[29] (just check the advanced search) or the *bibliographic domain ontology*[30], which are designed for very specific application needs.

All these strategies have significant drawbacks. The first strategy is very difficult to be applied. The task of finding identifying properties is indeed an almost impossible task for most of the things that can be encountered. Thinks about animals, artifacts, plants, general stuff; most of the time it is impossible to find a set of properties that do not change over time and that can be used for identifying them over encounters. The definition of article provided by Vivo is a perfect example. Not all the articles, for instance, have a 'DOI', which would seem to be a perfect identifier. The second strategy involves the construction of undetermined representations. In order to catch all the possible properties that something may have over encounters, the identification criteria are lost or too vague to be applied. Consequently, the puzzle of understanding if something is the same or is different from something else still remain unsolved. The representation provided by Schema.org is a perfect example. The things that can potentially be captured by all the properties associated to the schema 'article' are very different. See for instance the ambiguity in identifying 'news article' and 'scholarly article'. The third strategy offers criteria that can be used for identifying things only in a given context, which are very difficult to be used outside that context. Just think about the property 'location' provided by the Osikat schema. It is very difficult to find this information about an article, outside the context of a library service.

[27] http://schema.org/
[28] http://wiki.dbpedia.org
[29] http://oskicat.berkeley.edu/search/X
[30] http://bibliographic-ontology.org/

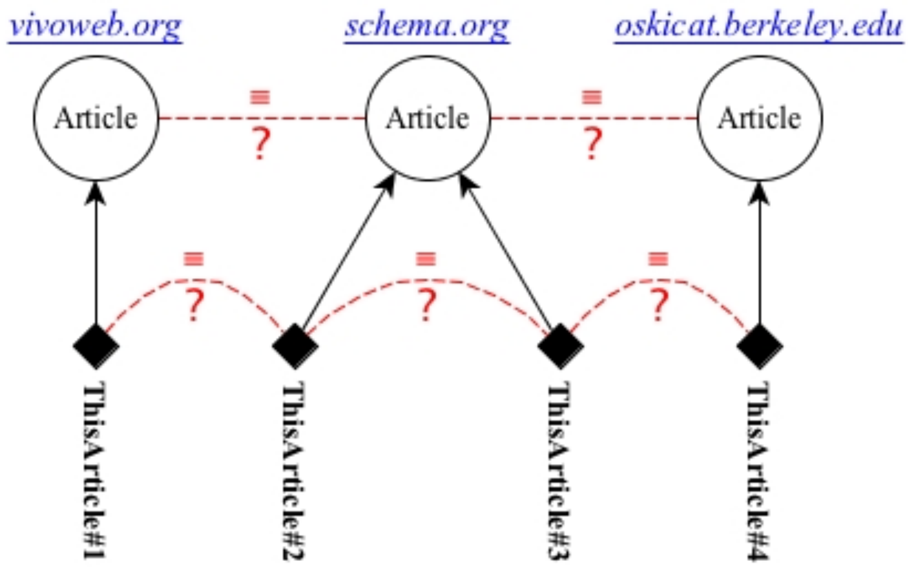

Figure 16: Diversity in knowledge and sameness puzzle

Figure 1 shows how the major implications of the above described drawbacks are: *i)* the generation of multiple and diverse representations (i.e., conventionally defined descriptions) for the same things; *ii)* the generation of multiple identifiers, i.e., *pointers by which we identify something* (e.g., URI), that can be related to the same resources, i.e., *any physical or virtual thing of limited availability within a system* [**14**] (e.g., a web page, an Amazon product, an article, and so forth). Just think how many new classes, properties, instances and corresponding URIs are created with ontology editors like Protégé whenever a new ontology is constructed. Following our example resumed by Figure 1, we can come out with puzzling situations where the same article is described and pointed as two articles just because we are selecting two different formats (i.e., paper and digital). Moreover, we may say that the article that I have on my desk (i.e., the draft of a research work that my advisor gave me one year ago) is the same article that now is stored in the library.

The issues of knowledge diversity and unique identifiers for resources that correspond to "things" are clearly central issues. They need to be addressed for achieving knowledge integration and semantic interoperability. This is why, in the last few years, most of the research effort has been devoted on designing the best methods for constructing reusable ontologies, and aligning and integrating multiple heterogeneous representations. All these attempts are motivated by the effect produced by the above described approaches to the sameness question, which, in turn, is the main reason for the difficulties in reusing and managing schemas, ontologies and identifiers of resources that correspond to thing.

In our opinion there is a lack of work on devising a new KR approach for addressing the puzzle of sameness without generating the above mentioned difficulties. This new reliable methodology would be a promising solution in supporting the current existing knowledge integration methodologies. It would help indeed in understanding whether two representations are about the same thing, thus exploiting (and not eliminating) the value of diversity in knowledge and enabling the reuse of identifiers.

5.1. Specifying the causal model: PC model

Where the underlying assumption of the state-of-the-art approaches is that two things are the same when they *share some specific properties*, the underlying assumption of our approach is that two things are the same when they *are part of the same causal history*, where we take a causal history to be a sequence of elements related by causal dependencies. We call *causal model* the information object that we use to represent and codify a causal history at different levels of complexity. A causal model is a relational structure, its *relata* are *Objects*, i.e., things of any particular sorts (e.g., animals, artifacts, stuff, and so forth), that may cause or be caused. These objects may stand in various relations, for instance, spatiotemporal relations and relations of part and whole (for more details see chapter on how to model t*eleologies*). Moreover, they are linked by the *Actions* they perform (e.g., running, eating, informing, and so forth). But it is the objects *Functions* that make a causal model. Following the teleosemantics approach, we take *Producer* and *Consumer* functions as the functions that are used for modeling the causal dependencies in a causal model. For instance, 'writer' is the producer function of a person who writes something, 'reader' is the consumer function of a person who read something, and so forth. This way of modeling the causal chain of every causal model is what allows us to describe a causal model as a *Producer-Consumer (PC) Model*.

5.1.1 PC model nodes

In the design of the PC Model we began with the core of the *PC Patterns* modeling effort provided in the previous chapter. In a PC Model, "object" is a cover term for those concepts that represent what is perceived across encounters. Every object can be thought as *the set of all the representations of how the same thing "fills" space, any time we encounter it*. The specification of `Object` is shown below:

```
Object = Oid Type [Comment] ;
Oid = ObjectID ;
ObjectID = "o" Integer ;
Type = "kind" | "individual" ;
Comment = Text ;
```

The `Type` nonterminal captures distinctions about the two main possible types of objects. Examples of each of these object types are:

1. **Kind**: *person*, *cat*, *car*, *article*, …
2. **Individual**: *Barak Obama*, *Garfield*, *Fiat Punto*, ...

The `comment` nonterminal is a part of all the PC Model specifications, and exists to allow for annotations, adding clarifications and other observations about the element being modeled.

"Action" is a cover term for those concepts that represent how objects change in time. Actions represent *an abstraction over multiple occurrence of changes in time of an object*. The specification of `Action` is shown below:

```
Action = Aid Type [Comment] ;
Aid = ActionID ;
ActionID = "a" Integer ;
```

```
Type = "conceive" | "realize" | "destroy"
     | "conceive" ;
Comment = Text ;
```

The `Type` nonterminal captures distinctions about the four main possible types of actions. Examples of each of these action types are:

1. **Conceive**: *devise*, *invent*, *create*, ...
2. **Realize**: *give-birth*, *build*, *assemble*, ...
3. **Destroy**: *eat*, *burn*, *wreck*, …
4. **Serve**: *feed*, *transport*, *inform*, …

`Conceive` represents the process by which an object, which was not existing before, is conceived. `Realize` represents the activity of building something with or from something else, i.e., making it recognizable. `Destroy` is the inverse of "realize" and represents the activity by which an object "disappears". `Serve` represents the activity by which any two objects may affect one another.

"Function" is a cover term for those concepts that represent *the behavior that an object is expected to have in a causal chain*. The specification of `Function` is shown below:

```
Function = Fid Type [Comment] ;
Fid = FunctionID ;
FunctionID = "f" Integer ;
Type = "producer" | "consumer" ;
Comment = Text ;
```

The `Type` nonterminal captures distinctions about the two main possible types of objects. Examples of each of these function types are:

1. **Producer**: *writer*, *mother*, *vehicle*, ...
2. **Consumer**: *reader*, *son*, *passenger*, ...

Every object that performs an action plays a specific function. The object that affects another object, possibly itself, plays the function of a `Producer`. The object being affected by the action of another object plays the function of a `Consumer`. The fact that objects affect each other by actions and can play the functions of a producer or a consumer involves the definition of the notion of *State*.

"State" is a cover term for those concepts that represent how objects, actions and functions change in a causal chain. States describe how objects, actions and functions fill any given causal model. The specification of `State` is shown below:

```
State = Sid Type [Comment] ;
Sid = StateID ;
StateID = "s" Integer ;
Type  =  "spatial_state"  |  "temporal_state"  |
     "functional_state" ;
Comment = Text ;
```

The `Type` nonterminal captures distinctions about the three main possible types of states. Examples of each of these state types are:

1. **Spatial state**: *weight*, *heavy*, *color*, *position*, ...
2. **Temporal state**: *duration*, *date*, *before*, ...
3. **Functional state**: *active*, *idle*, *in love*, *sick*, …

5.1.2 PC model links

We capture the fact that an object may perform one or more actions, may play one or more functions, and a function may involve one or more actions, by using the "by*" link. This link encodes all the possible relations that exist between objects, actions and functions, thus forming the backbone structure of every PC model. The specification of `By*` is shown below:

```
By* = Bid (FromObject | FromAction |
      FromFunction)    (ToObject    |
      ToAction   |   ToFunction)   Bond
      [Comment] ;
Bid = ByID ;
ByID = "b" Integer ;
FromObject = IDREF ;
IDREF = ObjectID ;
FromAction = IDREF ;
IDREF = ActionID ;
FromFunction = IDREF ;
IDREF = FunctionID ;
ToObject = IDREF ;
IDREF = ObjectID ;
ToAction = IDREF ;
IDREF = ActionID ;
ToFunction = IDREF ;
IDREF = FunctionID ;
Bond   =   "admissible"   |   "proper"
     {default,    if    absent,    is
     "admissible"} ;
Comment = Text ;
```

Besides having its own ID, `By*` has three `From*` optional IDs and three `To*` optional IDs, this is to allow the selection of the *relata* of the links. For instance, if the pair of optional IDs is `FromObject` and `ToAction` it means that the given `By*` is providing the action performed by an object. Notice that we are admitting the case in which `By*` can be used for modeling reflexive relations (e.g., `FromObject` and `ToObject`). This can be used for modeling parthood relations (e.g., an object that is a part of another object)[31]. The major function of the Bond nonterminal is to capture the fact that some elements are "expected" to be associated to other elements. For instance, the fact that an article has the function of informing someone is usually expected. On the contrary, the fact that an article is used as scrapbook is just possible, but usually not expected. This can be modeled by saying that "being information" is a proper function of an article and 'being a scrapbook' is just an admissible function of an article.

Together with by* link we provide the st* link. This link encodes all the possible relations between objects, actions, functions and their states, thus capturing how the elements of a PC model affect each other and how they change over a causal chain. The specification of St* is shown below:

```
St*    =    Stid    FromState
     (ToObject | ToAction |
     ToFunction) [Comment] ;
Stid = StID ;
StID = "st" Integer ;
FromState = IDREF ;
IDREF = StateID ;
ToObject = IDREF ;
IDREF = ObjectID ;
ToAction = IDREF ;
IDREF = ActionID ;
ToFunction = IDREF ;
IDREF = FunctionID ;
Comment = Text ;
```

[31] We are willing to address this issue in our future work.

Here, besides the `St*` ID, we have a `FromState` nonterminal that is not optional, providing the reference state ID, and three optional nonterminal elements, like for the by* link i.e., `ToObject`, `ToAction` and `ToFunction`, used for providing the related ID. We would like to point out that, at this level of specification, we left implicit a possible existing constraint. The type of the source state (spatial, temporal and functional) may indeed force the selection of a specific target element. If this is the case, it would mean that, for instance, all the temporal states can be only associated to actions and all the functional states can be only associated to function.

To illustrate a possible application of `By*` and `St*` links, along with the elements specified in the previous sub-section, let us consider the two following examples:

(1) a person wrote something one month ago

```
<OBJECT Oid="o1" Type="kind">
person
</OBJECT>
<ACTION Aid="a1" Type="realize">
wrote
</ACTION>
<STATE Sid="s1" Type="temporal_state">
one month ago
</STATE>
<By* FromObject="o1" ToAction="a1" bond="proper"/>
<St* FromState="s1" ToAction="a1"/>
```

(2) the author of this article is famous

```
<FUNCTION Fid="f1" Type="producer">
author
</FUNCTION>
<OBJECT Oid="O2" Type="kind">
article
```

```
</OBJECT>
<STATE Sid="s2" Type="functional_state">
famous
</STATE>
<By* FromFunction="f1" ToObject="o2" bond="proper"/>
<St* FromState="s2" ToFunction="f1"/>
```

5.2. Tracing and exploiting the causal model

Saying that two things are the same if they are part of the same causal history is like saying that something that is happening right now is the same of something that was happening some moments ago. We can say that they are the same writing (or the same reading) because they are part of the same causal history: it is *me* (object-producer) *writing* (action) this *article* (object-consumer). Now, according to this new strategy of approaching the sameness puzzle, change can be easily explained through the parts of a causal history. For instance, there would be no problems at claiming that the article that yesterday was on the online repository is the same article that today is on the table of my apartment. This just means focusing on two spatiotemporal parts of that article: the first that yesterday was on the online repository, the second that today is in the apartment. Similarly, it would be perfectly fine to state that this article and the same article without the title page (and maybe without some other pages) are two parts of the same causal history. Moreover, there would be no inconsistencies in admitting that we are talking about the same article even after it has been cut into small pieces. It is just a matter of considering a longer or a shorter causal history for that article.

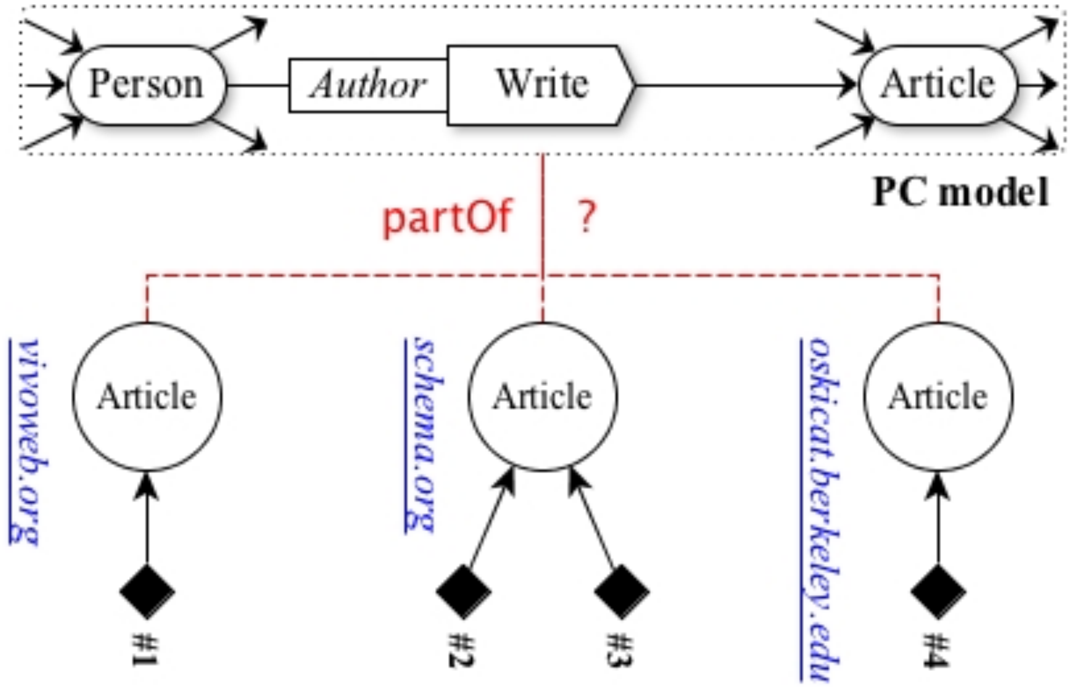

Figure 17: The new way of approaching the sameness puzzle

Figure 2 shows how the major implications of the above described strategy shift are: *i)* enabling to trace the multiple and diverse representations of the same thing via a single representation (i.e., the reference causal history); *ii)* mitigating the generation of multiple identifiers, considering each single representation (class, property or instance) just as a *part of* the same thing (i.e., the reference causal history).

5.2.1 From KRs to PC model

The PC model we described in the previous section plays a central role in the computational implementation of this new approach. However, a full implementation can be provided only by specifying the process that allows to reduce any number of input KRs to a reference PC Model and to keep-track of them by exploiting their diversity. We call this process *Concretization operation* and we describe it as follows.

(1) Concretization()

```
for each kr_c and pc_n
    if kr_c = < Class > and pc_n = < Object >
        get kr_c
        return pc_n
    end if
    if kr_c = < ObjectProperty_N > and pc_n = < Function >
        get kr_c
        return pc_n
        determine By* as Function-Object link
    end if
    if kr_c = < ObjectProperty_V > and pc_n = < Action >
        get kr_c
        return pc_n
        determine By* as Action-Object link
        determine By* as Action-Function link
    end if
    if kr_c = < DataProperty > and pc_n = < State >
        get kr_c
        return pc_n
        determine St* as State-Object or State-Action
        or State-Function link
    end if
end for
```

The inputs of **Concretization()** are always: *i)* a *knowledge representation construct*, i.e., kr_c (as represented in the table above), taken from a given KR (e.g., an ontology with classes and properties), and *ii)* a *PC model node,* i.e., pc_n (as represented in the table above), taken from a given PC model (e.g., *Person-Author-Write-WrittenWork-Article*). Here we consider the possible knowledge representation constructs as the constructs used in OWL[32], but the set of input reference representation languages can be easily extended, for instance, to RDF[33] and others. Thus, in this case, on the one hand, a kr_c can be a *class*, like 'Article', an *object property* like 'Author', with *range* 'Person', and a *data property* like 'NumberOfPages', with *datatype* 'Integer'. Notice that we make a central distinction between object properties denoted by nouns and object properties denoted by verbs ($ObjectProperty_N$ and $ObjectProperty_V$ in the table above). This is in order to make a clear distinction among those properties that can be functions and those properties that can be actions. On the other hand, a pc_n is one of the nodes that we specified in Section 3. Through the concretization process each given kr_c is converted into a given pc_n. Moreover, after each conversion step a by* link or a st* link is determined. The final expected output is a PC Model evoked by the input knowledge representation.

Now, there are two major observations. Firstly, in the example we provided, we assumed that all the knowledge representation constructs have an already existing PC model input to which they can be associated. However, there may be the case in which, given a kr_c, no pc_n can be returned. This situation is handled by evaluating a possible extension of the given PC model. This can be done by adding a new object, action, function, or state, or by determining a new by* or a new st*. For instance, 'booking' and 'availability' as knowledge representation inputs may involve an evaluation for the enrichment of the given PC model with a new state and a new action. If this is the case, given the input object property 'booking' and the input data property 'availability', the concretization process returns the action 'booking' and the state 'availability'. It must be noticed that the given PC model may have also been left unchanged. This would have meant that the given knowledge representation inputs are not taken as part of that causal history representation. The second observation is that the output PC model produced by means of the concretization operation can be used as input with a new KR, thus producing a new extended PC model, including both the given input KRs. If this is the case, it is possible to state that: *i)* the two input KRs are parts of the same causal history, represented by the

[32] https://www.w3.org/OWL/
[33] https://www.w3.org/RDF/

given extended PC model and *ii)* both the input KRs can be considered as the same by integrating them into a unified KR.

5.2.2 From PC model to KR

The generation of a unified KR from an extended PC model (including two inputs KRs) happens according to another central process, i.e., what we call *Abstraction operation*. This operation can be seen as the inverse of concretization and we describe it as follows.

```
(2) Abstraction()
for each pc_n and kr_c
                  if pc_n = < Object > and kr_c = < Class >
      get pc_n
      return kr_c
   end if
   if pc_n = < Function > and kr_c = < Class >
      get pc_n
      return kr_c
   end if
   if pc_n = < Function > and kr_c = < ObjectProperty_N >
      get pc_n
      return kr_c
      determine class expression (Class-
      ObjectProperty)
   end if
   if pc_n = < Action > and kr_c = < Class >
      get pc_n
      return kr_c
   end if
   if pc_n = < Action > and kr_c = < ObjectProperty_V >
      get pc_n
      return kr_c
```

determine *class expression (Class-ObjectProperty)*
end if
if pc_n = < *State* > and kr_c = < *Class* >
get pc_n
return kr_c
end if
if pc_n = < *State* > and kr_c = < *DataProperty* >
get pc_n
return kr_c
determine *class expression (Class-DataProperty)*
end if
end for

The inputs of **Abstraction()** are always: *i)* a *PC model node,* i.e., pc_n taken from a given PC model that may be the result of the integration of more given KRs (see previous section); *ii)* a *knowledge representation construct*, i.e., kr_c, taken from two or more input KRs. As showed in the table above, every given PC model input object is converted into a given class. Every given PC model input function may be converted into a given class or a given object property. Every given PC model input action may be converted into a given class or a given object property. Every given PC model input state may be converted into a given class or a given data property. Moreover, whenever a property is returned, a class expression for defining the KR classes is determined. The expected output of the whole operation is a KR that is an integration of the two (or more) given KRs. This is possible because the input KRs are taken as parts of the same causal history and can be considered as different descriptions of the same thing.

5.3. A cause study

We ground the explanation of our approach in the following concrete example. Let us take the three reference knowledge representations introduced in Section 2, i.e., the Vivoweb.org, the Schema.org and the Oskikat representations of 'Article'. We consider them as KR_1, KR_2 and KR_2 respectively and we represent (a portion of) the corresponding schemas (with some instantiation examples) as follows.

Article – *KR₁*				
DOI	abstract	pageNo	subject	date
#123	*Text*	*56*	*ai*	*01-01-2017*

Article – KR_2				
author	award	comment	editor	rating
J.	*Award 'a'*	*4*	*K.*	*3 stars*
J. and M.	*Award 'a'*	*13*	*W.*	*4 stars*

Article – KR_3				
title	author	location	language	availability
Text	*J. and M.*	*Data Lab*	*Eng*	*Boolean*

In this example we take the table labels as classes and the field labels as either object properties or data properties. Suppose that there is a need to check whether these are three representations of the same thing. We solve this problem via **Concretization()**. Let us assume that the reference PC model, i.e., PC_0, has been already constructed and that a representation of it, with the concepts that are relevant to this example, is as reported in Tab. 1 (the terms in "*" are the result of the iterative application of the concretization operation, as described below).

Table 3: *An example of PC Model*

Object	**Function**	**Action**	**State**
Person	{Author, *Editor*}	{Write, *Edit*}	{Date}
Article	{Realization, *Edition*}	{*null*}	{DOI, Abstract, PageNo, *Subject*, Award, Rating, Title, Location, Language, *Availability*}

The concretization operation is processed sequentially. The first application of the operation takes KR_1 and PC_0 as given inputs and return PC_1 as expected output. The second application of the operation takes KR_2 and PC_1 as given inputs and return PC_2 as expected output. The third application of the operation takes KR_3 and PC_2 as given inputs and return PC_3 as expected output. Every time we run the operation we may encounter the following situations.

1. a *KR* input is returned into a given *PC* input. In the example this may be the case of the 'DOI' property, which is returned into a corresponding state (see Tab. 1);
2. a KR property input of a KR class input that is returned as a PC Model object, is returned as a state that in the PC model is associated to an action. In the example this is the case for 'Date' that in KR_1 is a property of 'Article' but in PC_0 it is a temporal state either of 'Writing' or 'Editing'. This situation is handled just by returning the associations as they are modeled in the PC model;
3. a *KR* input (i.e., a class or a property) cannot be associated to any *PC* input (i.e., the corresponding term does not occur in the given PC model). In the example this is the case for 'Subject' (in KR_1), 'Editor' (in KR_2) and 'Availability' (in KR_3). This situation can be handled by:
    a. enriching the reference PC model with one corresponding element. For instance, 'Date' and 'Subject' will be present in PC_1 and 'Availability' in PC_3 (see term marked with '*' in Tab. 1);
    b. enriching the reference PC model by adding more elements. For instance, the producer function 'Editor' will be present in PC_2 along with the consumer

function ‘Edition’ and the action ‘Editing’ (see term marked with ‘*’ in Tab. 1). This solution is usually provided whenever the causal chain encoded by the PC model is extended with a new core element (i.e., object, function, action).

Once the concretization process is finished, the final PC model output, i.e., PC_3, according to the given example, can be used as input of **Abstraction()**. This can be done for generating:

1. one of the previous knowledge representation inputs, KR_1, KR_2 or KR_3. For instance, the representation of ‘Article’ provided by KR_1;
2. a new representation for the previous input KRs elements. For instance, we may have an article with just ‘DOI’, ‘rating’, ‘title’ and ‘author’ (collapsing all the information provided by ‘Author’, i.e., ‘J.’ and ‘J. and M.’ are taken as ‘J. and M.’);
3. a new representation for a new KR element. For instance, we may run abstraction for creating a new class labeled as ‘Editor’;
4. a new representation that is the result of the integration of the previous input KRs. For instance, we may have a KR that is the result of the integration between KR_1 with KR_2, or between KR_1, KR_2 and KR_3.

The example process described above highlights various key features of our approach. Let us analyze them. Firstly, it seems plausible to assume that the PC model can be used for modelling a huge varieties of concepts. As a cross-check we can take any online vocabulary and see whether its concepts can be modeled according to our approach. Let us take, for instance, some of the Schema.org “commonly used types”. The ‘CreativeWork’ type can be easily seen as a kind of object. The type ‘Event’ can be seen as the result of an abstraction operation that combines information derived from objects, actions and functions together (just think about the reading of an article). We have the same situation for the concept ‘Person’. Similarly, the type ‘Place’ can be easily captured by abstracting from an object just some specific ‘SpatialStates’. A similar check can be performed for all the Schema.org concepts that are commonly used as properties. For instance, a property like ‘Creator’ can be mapped into a function. The same happens for properties like ‘Read’, ‘Write’, and so forth, that can be mapped into actions, or properties like PageNo, Title, Language, etc., which can be easily mapped into states. The possible domains and codomains for each property is simply captured by the by* and st* relations codified in the PC model. For instance, the domain and codomain of the property 'Creator' can be derived from the by* used for linking it to 'Person' and 'Article'. Secondly, it is interesting to consider in depth the concretization operation and the role played by the reference PC model. Every concretization operation output is indeed a PC model that is built from the integration of multiple KR inputs. This underline how every PC model is a highly flexible conceptual structure and can guarantee for the interoperability of different KRs. As we have showed above, a PC model can be continuously adapted and tuned over time with more information. The upshot is that, projecting the knowledge representations over a causal chain, the different conceptualizations seem to be always compatible and do not generate inconsistencies. This flexibility is essential in supporting central tasks like knowledge acquisition and adaptation, and is required primary for tracing and exploiting knowledge diversity. Finally, our approach may also provide interesting insights on how to integrate different computational approaches to the representation of concepts. A PC model, depending on the perspective adopted, can be seen, for instance, as a grounding for both a symbolic and connectionist view of concepts. By means of abstraction, the constructs composing a PC model can be easily mapped, as we have showed above, into a class-relational structure. Similarly, by means of concretization, the basic constructs of a class-relational structure can be transformed into a representation that is very similar to the formalism used for representing a neural network. Objects, functions and actions can be taken, indeed, for representing the (internal) units composing the hidden layer of a connectionist network. States and state types, can be taken for representing the external (input/output) units. The pivotal point here is that it seems possible to provide an account for different conceptual modelling formalisms by using a single reference conceptual modeling frame.

Part III

Conclusion

Chapter 6

6. Conclusions and future work

In this thesis, besides providing a shared terminology for the characterization of concepts and for their computational representation, we have provided a characterization of concepts as (recognition) abilities. We showed how this new characterization of concepts can be mapped to the notion of concepts studied so far in KR, what we call classification concepts. We exploited this characterization to develop RAO, a first version of an ontology of concepts as recognition abilities and we showed how it can be used to characterize which classification concepts are only nominals or also substance concepts.

Moreover, we have shown how the world can be modeled in terms of three concepts at three increasing levels of abstractions: objects which represent the result of the perception of substances, actions which represent how substances change in time, and functions which represent the expected behaviour of objects. These three notions have allowed us to introduce PC patterns and then teleologies as a first step towards a general solution of the knowledge and data integration problem. To this extent, we have briefly described how teleologies can be tuned to the specific problem and later adapted as needed, following a precise methodology.

As a final outcome, we described how tracing diversity via a causal model can represent a first foundational step towards the implementation of an adaptive KR system. Because of the use of what we call PC model and the sharp distinction between the "causal layer" and the "representational layer", it is possible to address the lack of generality that affect current KR approaches. We exploited some of the Millikan's main findings to design an architecture which integrate perception, knowledge acquisition and knowledge (re-)use, and we showed how any representation can be seen as the result of an adaptive process.

6.1. Future work

A number of paths with opportunities to extend the scope of this thesis were left for future work, either for lack of time or limitation of resources. In what follows we describe some of these paths. The first and most obvious is to analyze and implement possible improvements of the proposed approach, as well as its evaluation. For instance, we would like to further refine RAO and its use in the identification of classification concepts which are also substance concepts. We are also interested to exploit these ideas in the implementation of an integrated system that deploys both a recognition and a knowledge representation function. This work will consist in the development of large scale teleologies (including a full formalization of schema.org) and a reference top-level teleology. Another aspect that we are interested in is the development of a detailed methodology for the construction of teleologies (and corresponding PC models) and related KRs, together with the implementation of an *ad hoc* editor for the creation of teleologies, and the exporting and the importing of reference knowledge representations.

By last, we mentioned throughout the thesis that our work takes into consideration main biosemantics principles, proposing a first computational view of the biosemantics frame that facilitate the distinction between substance and classification concepts. As part of the future work, we are interested in extending the proposed formalization. This requires a more deeply study of biosemantics in the context of artificial intelligence, which may in turn require a re-design of some of the proposed formalizations. Although we are aware of the challenges of this line of work, we also believe it represents the most interesting path for a future work.

Bibliography

1. Adams, B., M. Raubal (2009). Conceptual space markup language (CSML): Towards the cognitive semantic web. In *2009 IEEE International Conference on Semantic Computing*, 253-260.
2. Adams, F., & Aizawa, K. (2010). *Causal theories of mental content*, https://stanford.library.sydney.edu.au/entries/content-causal/.
3. Baader, F. (Ed.). (2003). *The description logic handbook: Theory, implementation and applications*. Cambridge university press.
4. Barasalou, L. W. (1999). Perceptual symbol system. *Behavioral and Brain Science*, 22(4), 577-609.
5. Barker, Phil, and Lorna M. Campbell. What is schema.org? *LRMI*. *Retrieved April* 21 (2014): 2015
6. Barsalou, L. W. (2012). The human conceptual system. *The Cambridge handbook of psycholinguistics*, 239-258.
7. Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems* (pp. 153-160).
8. Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), 28-37.
9. Besnard, P. (2013). *An introduction to default logic*. Springer Science & Business Media.
10. Bird, Graham. *Kant's Theory of Knowledge: An Outline of One Central Argument in the 'Critique of Pure Reason'*. Vol. 1. Routledge, 2016
11. Bottazzi, E., R. Ferrario, C. Masolo. The mysterious appearance of objects, in M. Donnelly, G. Guizzardi (Eds.) (2012). *FOIS 2012*, *Formal Ontology in Information Systems*, Frontiers in Artificial Intelligence and Applications, vol. 239, IOS Press, 59-72.
12. Bouquet, P., et al., C-owl: Contextualizing ontologies. In *The Semantic Web-ISWC 2003*, Springer Berlin Heidelberg, 164-179, 2003.

13. Bouquet, P., et al., Contexts, locality and generality. In *Mathware & soft computing*, Vol. 3 Núm. 1 [-2], 47-57, 1996.
14. Bouquet, Paolo, Heiko Stoermer, Michele Mancioppi, and Daniel Giacomuzzi. "OkkaM: Towards a Solution to the" Identity Crisis" on the Semantic Web." In *SWAP*, vol. 201. 2006.
15. Brachman, R. J. (1979). On the epistemological status of semantic networks. *Associative networks: Representation and use of knowledge by computers*, 3-50.
16. Brachman, R. J., & Schmolze, J. G. (1985). An overview of the KL-ONE knowledge representation system. *Cognitive science*, 9(2), 171-216.
17. Brachman, R., H. Levesque (1985). *Readings in knowledge representation*. Morgan Kaufmann.
18. Braddon-Mitchell, D. (2007). *The philosophy of mind and cognition*. Blackwell.
19. Brooks, R. A. (1991). Intelligence without representation. *Artificial intelligence*, 47(1-3), 139-159.
20. Brooks, Rodney A. (1999). Cambrian Intelligence: The Early History of the New AI. *Cambridge MA*: The MIT Press.
21. Broomhead, D. S., & Lowe, D. (1988). *Radial basis functions, multi-variable functional interpolation and adaptive networks* (No. RSRE-MEMO-4148). Royal Signals and Radar Establishment Malvern (United Kingdom).
22. Cangelosi, A., Coventry, K. R., Rajapakse, R., Joyce, D., Bacon, A., Richards, L., & Newstead, S. N. (2005). Grounding language in perception: A connectionist model of spatial terms and vague quantifiers. *Progress in Neural Processing*, 16, 47.
23. Cangelosi, A., Greco, A., & Harnad, S. (2000). From robotic toil to symbolic theft: Grounding transfer from entry-level to higher-level categories. *Connection Science*, 12(2), 143-162.
24. Chen, Liming, Chris D. Nugent, and Hui Wang. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering* 24, no. 6 (2012): 961-974
25. Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, Alessandro Oltramari, Ro Oltramari, Luc Schneider, Lead Partner Istc-cnr, Ian Horrocks. *WonderWeb Deliverable D17. The WonderWeb Library of Foundational Ontologies and the DOLCE ontology*. 2002

26. Coliva, A. (2006). *I Concetti, Teorie ed esercizi*. Carocci.
27. Crowley, Terry, and Claire Bowern. *An introduction to historical linguistics*. Oxford University Press, 2010
28. Davis, E., & Marcus, G. (2015). Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9), 92-103.
29. Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179-211.
30. Findler, N. V. (Ed.). (2014). *Associative networks: Representation and use of knowledge by computers*. Academic Press.
31. Fodor, J. A. (1998). *Concepts: Where cognitive science went wrong*. Oxford University Press.
32. Fodor, J. A. (2008). *LOT 2: The language of thought revisited*. Oxford University Press on Demand.
33. Fodor, J. A., & Lepore, E. (2002). *The compositionality papers*. Oxford University Press.
34. Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1), 3-71.
35. Forsyth, David A., and Jean Ponce. A modern approach. *Computer vision: a modern approach* (2003): 88-101
36. Frixione, M., & Lieto, A. (2011). Representing concepts in artificial systems: a clash of requirements. *Proc. HCP*, 75-82.
37. Frixione, M., & Lieto, A. (2012). Representing concepts in formal ontologies. Compositionality vs. typicality effects. *Logic and Logical Philosophy*, 21(4), 391-414.
38. Gärdenfors, P. (2014). *The geometry of meaning: Semantics based on conceptual spaces*. MIT Press.
39. Gibson, James J. Perceiving, acting, and knowing: Toward an ecological psychology. *The Theory of Affordances* (1977): 67-82
40. Giunchiglia, F., & Fumagalli, M. (2016, July). Concepts as (Recognition) Abilities. In *Formal Ontology in Information Systems: Proceedings of the 9th International Conference (FOIS 2016)* (Vol. 283, p. 153). IOS Press.
41. Giunchiglia, Fausto, and Mattia Fumagalli. "Teleologies: Objects, Actions and Functions." International Conference on Conceptual Modeling. Springer, Cham, 2017.

42. Giunchiglia, F., et al., Semantic Matching: Algorithms and Implementation. In *Journal on Data Semantics*, 1-38, 2007.
43. Giunchiglia, Fausto, and Toby Walsh. A theory of abstraction. *Artificial intelligence* 57.2-3 (1992): 323-389
44. Giunchiglia, Fausto. Managing Diversity in Knowledge. *Keynote talk, European Conference on Artificial Intelligence (ECAI-06)*, 2006. http://www.disi.unitn.it/~fausto/knowdive.ppt
45. Gruber, Thomas R. Toward principles for the design of ontologies used for knowledge sharing. *International journal of human-computer studies 43*, no. 5-6 (1995): 907-928
46. Guarino, N., (2009). The ontological level: Revisiting 30 years of knowledge representation. In *Conceptual modeling: Foundations and applications*, 52-67. Springer Berlin Heidelberg.
47. Guarino, N., & Welty, C. (2002). Evaluating ontological decisions with OntoClean. *Communications of the ACM*, 45(2), 61-65.
48. Guarino, N., G. Guizzardi (2015). "We Need to Discuss the Relationship": Revisiting Relationships as Modeling Constructs. In *Advanced Information Systems Engineering*, 279-294. Springer International Publishing.
49. Guarino, N., Oberle, D., & Staab, S. (2009). What is an Ontology?. In *Handbook on ontologies* (pp. 1-17). Springer Berlin Heidelberg.
50. Guizzardi, G. (2005). *Ontological foundations for structural conceptual models*. CTIT, Centre for Telematics and Information Technology.
51. Hayes, B. (2013). First links in the Markov chain. *American Scientist*, 101(2), 252.
52. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
53. Hebb, D. O. (1949). *The organization of behavior: A neuropsychological approach*. John Wiley & Sons.
54. Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8), 2554-2558.
55. Johnson-Laird, P. N. (1977). Procedural semantics. *Cognition*, 5(3), 189-214.

56. Joyce, D., Richards, L., Cangelosi, A., & Coventry, K. R. (2003). On the foundations of perceptual symbol systems: Specifying embodied representations via connectionism. In The logic of cognitive systems. *Proceedings of the fifth international conference on cognitive modeling* (pp. 147-152). Universitätsverlag Bamberg.
57. Klir, G., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic* (Vol. 4). New Jersey: Prentice hall.
58. Kuhn, W., (2009). A functional ontology of observation and measurement. In *GeoSpatial Semantics*, 26-43. Springer Berlin Heidelberg.
59. Lalumera, E., (2010). Concepts are a functional kind. *Behavioral and Brain Sciences*, *33*(2-3), 217-218.
60. Lieto, A., Minieri, A., Piana, A., & Radicioni, D. P. (2015). A knowledge-based system for prototypical reasoning. *Connection Science*, 27(2), 137-152.
61. Lieto, A., Radicioni, D. P., Rho, V., & ICAR-CNR, P. I. (2015, June). A Common-Sense Conceptual Categorization System Integrating Heterogeneous Proxytypes and the Dual Process of Reasoning. In *IJCAI* (pp. 875-881).
62. Macdonald G. and D. Papineau, Introduction: Prospects and problems for teleosemantics. In G. Macdonald and D. Papineau, eds., *Teleosemantics: New Philosophical Essays*, Oxford U. P., 2006.
63. MacLennan, B., (1998). Finding order in our world: The primacy of the concrete in neural representations and the role of invariance in substance reidentification. *Behavioral and Brain Sciences*, *21*(01).
64. Markman, E., (1989). *Categorization and Naming in Children*, Cambridge, MIT Press.
65. Martin, A., L. Chao (2001). Semantic memory and the brain: structure and processes. *Current opinion in neurobiology*, 11(2), 194-201.
66. McCarthy, J., Generality in artificial intelligence. In *Commun. ACM 30*, 12, 1987.
67. McClelland, J. L., & Rogers, T. T. (2003). The parallel distributed processing approach to semantic cognition. *Nature reviews. Neuroscience*, 4(4), 310.
68. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.
69. Millikan, R. 'Biosemantics'. In *Journal of Philosophy* 86, 281-297, 1989. Repr. in *White Queen Psychology and Other Essays for Alice*, Cambridge, Mass: MIT Press, 1993.

70. Millikan, R., (1998). A more plausible kind of "recognitional concept". In *Philosophical Issues*, 35-41.
71. Millikan, R., (2005). *Language: A biological model*. Oxford University Press on Demand.
72. Millikan, R., (Forthcoming). Thoughts of real kinds. *The Oxford handbook of philosophy of psychology. Oxford University Press, Oxford*.
73. Millikan, R., *Language, thought, and other biological categories: New foundations for realism*, MIT press, 1984.
74. Millikan, R., *Language: a Biological Model*, Oxford University Press, 2005.
75. Millikan, R., *On Clear and Confused Ideas: An Essay About Substance Concepts*, Cambridge University Press, 2000.
76. Millikan, R., *Varieties of meaning: the 2002 Jean Nicod lectures*, MIT press, 2002.
77. Murphy, G. (2004). *The big book of concepts*. MIT press.
78. Ni, Qin, Iván Pau de la Cruz, and Ana Belén García Hernando. A foundational ontology-based model for human activity representation in smart homes. *Journal of Ambient Intelligence and Smart Environments* 8, no. 1 (2016): 47-61
79. Niles, Ian, and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pp. 2-9. ACM, 2001
80. Noy, N. F., & Klein, M. (2004). Ontology evolution: Not the same as schema evolution. *Knowledge and information systems*, 6(4), 428-440.
81. O'Reilly, R. C. (1998). Six principles for biologically based computational models of cortical cognition. *Trends in cognitive sciences*, 2(11), 455-462.
82. Ortmann, Jens, and Werner Kuhn. Affordances as Qualities. In *FOIS*, pp. 117-130. 2010.
83. Paternoster, A. (2014). *Introduzione alla filosofia della mente*. Gius. Laterza & Figli Spa.
84. Peng, B., Lu, Z., Li, H., & Wong, K. F. (2015). *Towards neural network-based reasoning*. arXiv preprint arXiv:1508.05508.
85. Pezzulo, G., Barsalou, L. W., Cangelosi, A., Fischer, M. H., McRae, K., & Spivey, M. (2013). Computational grounded cognition: a new alliance between grounded cognition and computational modeling. *Frontiers in psychology*, 3, 612.

86. Prinz, J. J. (2004). *Furnishing the mind: Concepts and their perceptual basis*. MIT press.
87. Prinz, Jesse. Beyond appearances: The content of sensation and perception. *Perceptual experience* (2006): 434-460
88. Rodríguez, Natalia Díaz, Manuel P. Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores. A survey on ontologies for human behavior recognition. *ACM Computing Surveys (CSUR)* 46, no. 4 (2014): 43
89. Rosch, E. (1975). Cognitive representations of semantic categories. *Journal of experimental psychology*: General, 104(3), 192.
90. Rosch, Eleanor, Carolyn B. Mervis, Wayne D. Gray, David M. Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive psychology* 8, no. 3 (1976): 382-439
91. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.
92. Roy, D. (2005). Grounding words in perception and action: computational insights. *Trends in cognitive sciences*, 9(8), 389-396.
93. Ryder, D., et al., eds., *Millikan and Her Critics*, John Wiley & Sons, 2013.
94. Şahin, Erol, Maya Çakmak, Mehmet R. Doğar, Emre Uğur, and Göktürk Üçoluk. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior* 15, no. 4 (2007): 447-472
95. Shea, N., Millikan's contribution to materialist philosophy of mind. In *Matière Première 1*, 127-156, 2006.
96. Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, 171(3972), 701-703.
97. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489.
98. Sloman, A. (2014). How can we reduce the gulf between artificial and natural intelligence?. In *AIC* (pp. 1-13).
99. Smeulders, A, M. Worring, S. Santini, A. Gupta, R. Jain (2000). Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1349-1380.

100. Smith, B., Prologue to Reflections and Semantics in a Procedural Language". In *Ronald Brachman and Hector J. Levesque. Readings in Knowledge Representation*. Morgan Kaufmann, 31–40, 1985.
101. Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and brain sciences*, 11(01), 1-23.
102. Sowa, John F. *Knowledge representation: logical, philosophical, and computational foundations*. Vol. 13. Pacific Grove: Brooks/Cole, 2000
103. Stöttinger, J., J. Uijlings, A. Pandey, N. Sebe, F. Giunchiglia (2012). (Unseen) event recognition via semantic compositionality. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 3061-3068.
104. Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2), 161-197.
105. Sun, R. (2006). The CLARION cognitive architecture: Extending cognitive modeling to social simulation. *Cognition and multi-agent interaction*, 79-99.
106. Tyler, L. K., Moss, H. E., Durrant-Peatfield, M. R., & Levy, J. P. (2000). Conceptual structure and the structure of concepts: A distributed account of category-specific deficits. *Brain and language*, 75(2), 195-231.
107. Vernon, D., Metta, G., & Sandini, G. (2007). A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *IEEE transactions on evolutionary computation*, 11(2), 151-180.