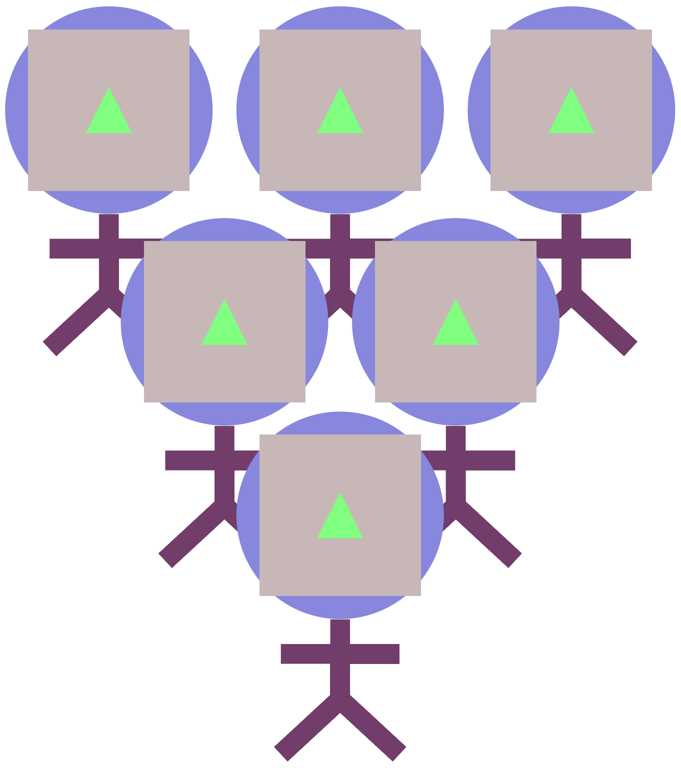


Connecting conceptual agents in Semantic Web

Connecting conceptual agents in Semantic Web

Aris van Dijk

October 10, 2008

Master's thesis
Cognitive Artificial Intelligence
at the University of Utrecht

Supervised by
Dr. H.M. Aldewereld
Dr. M.V. Dignum

Preface

Dear reader! Laying before you is a piece of work I have enjoyed very much; the exploration of which was as interesting as the result. After I planned to visit this area, many practical issues led me to new insights, new journeys, new destinations. This paper is meant to give a close experience of the peculiarities of the country, its impressive sights, its inhabitants, all its sounds and the images which I have tried to describe as objectively as possible.

Along the way the reader may find definitions; they are not meant as results or destinations, but as posts in the landscape to get a first sight on what is coming. The reader may stumble upon stories, as real as reality; upon examples, as curious as curiosity; upon quotes, footnotes, schemata, algorithms that describe stepwise programs, and everything that was gathered to involve the reader in the subject. All to travel from each research question to its answer and the conclusion.

At some point of your journey the reader may get stuck. Do not despair; other parts of the country may be far easier to grasp, and you don't share the background knowledge I have gathered as an experienced traveler in these areas. Follow the signs; return to the point where you lost the way, reading the text in reverse order; or skip to the next picture.

It may also be at some point of your journey that you feel something is missing. Do not hesitate to contribute to the discovery of the country. At `http://sourceforge.net/projects/croc` the project is maintained under an open source license.

Finally, I want to thank everyone who helped me; especially Huib Aldewereld and Virginia Dignum, my master's project supervisors, both working on the ALIVE project (`http://www.ist-alive.eu`).

The digital version of this thesis is available at `http://croc.sourceforge.net/thesis.pdf`, which contains links, colors, and pretty zoom functions. At `http://croc.sourceforge.net/` one may also find other publications on the subject of this thesis.

Contents

Part I

Introduction

Introduction

Once upon a time, a long time after the first industrial revolution, people decided to build a computer; a machine that would not yield movement, like a steam machine — that would not yield time representation, like a clock — but that would yield outcomes to mathematical problems (problems of thought!) on demand. Several years later (1948), a computer (the hardware) was made to store its own program (the software) in *memory*. After much years of improvement on this machine, a new technical concept was formulated: the development of "[a] self-contained, interactive and concurrently-executing object, possessing internal state and communication capability" (Hewitt, 1977), a *software agent*.[1]

Definition 1 *An* agent *is a computer system that is* situated *in some* environment*, and that is capable of* autonomous action *in this environment in order to meet its design objectives. [Woolridge and Jennings, 1995]*

Artificial agents are not standing alone; there are multiple agents, serving different goals, having different plans, and different beliefs. To function well, they have to be able to process information from their environment, and understand the content of that information. And software agents, together with other software services, currently are situated in the *web*. There are a lot of different software pieces out there, designed for specific tasks and knowledge domains.

The *Semantic Web* is the total of the web providing content cognitive artificial agents can understand, and connecting these agents and other services for exchanging content (communication).

Definition 2 *The* Semantic Web *provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. [W3C Semantic Web Activity]*

But, currently (2008), the Semantic Web is not there yet: it is under construction. The signs of construction are various technologies: tagging (providing labels for things), microformats (placing small units of data (e.g., vCards, calendar events) in recognisable HTML representations), rich data format (RDF). The last (RDF) technology developed further under the names RDFS and OWL to become a *Web Ontology Language*.

'Ontologies' are world-views, classifications that give representations a place. As agents (and their designers) are autonomous, they developed their own specific ontologies to express what they liked. However, when communicating with other agents (using representations) it became apparent that shared elements in ontologies of both agents were indispensable[2]. This lead to some hard work of mapping, aligning or merging ontologies.

But let us focus on the underlying mechanisms of classification and representation and see if they provide a solid basis for artificial intelligence.

[1]For some readers, it might not be clear how to identify a software agent. The important difference with other software is its *'autonomy'*. Think for example of the Word assistant ("paperclip"): doing things you don't suspect or desire... and compare this to, e.g., Word itself, which only reacts on explicit commands given. Or compare a robotic lawn mower with a simple microwave oven.

[2]This was called "the interoperability problem" [Wache et al., 2001].

1.1 The mechanism of classification

We will start with the use of *representation.* Processing a representation is a process of *identification.* An example of a representation is natural language. We process natural language by *reidentifying* things by the words or phrases. For example, processing the word "fish" leads to a fish-reidentification.

Millikan [2000] describes how reidentification is the central process of cognition.[3] To be able to recognize, we use *concepts.* Millikan defines a concept as *an ability to reidentify for a purpose.*

Definition 3 *A* concept *is an ability to reidentify for a purpose. [Millikan, 2000]*

For example, a frog might have a concept for "something frog-food", and its biological purpose is to get food (by extruding its tongue when it identifies "something frog-food"). Notice that the frogs ability is fallible: it may also eat other moving small things, such as thrown dust.

Because processing a representation is a process of *identification*, Semantic Web 'ontologies' should serve this ability to reidentify. These 'ontologies' currently define *classes*, in a hierarchical way. But, *classification* is a different task than *identification* [Millikan, 2000, Chapter 3].

Classification as a world view Classification is placing something from a certain domain in a class. Classification can be used for reference: "red book on the table in my study" [Millikan, 2000, p.35] uses, e.g., the word "red" to efficiently narrow the class of books that can be meant. Classification can be used for efficient *knowledge representation.* This way of referring involves choosing the right words to classify efficiently. However, the words that a speaker uses for classification are words that have to be *identified* by the hearer before they can participate in that classification.

Classification can also be seen as a restriction on what a system is able to represent. In a way, a system can be described with (a) a classification system and (b) a set of rules to apply to each class in that system. In that perspective, the classification system is the world-view of the system. Semantic Web 'ontologies' seem to try to capture such world-views for Semantic Web systems. (Each class is a kind of sign.)

Definition 4 Semantic Web 'ontologies' *define the [class] concepts and relationships used to describe and represent an area of knowledge. 'Ontologies' are used to classify the terms used in a particular application, characterize possible relationships [i.e. class properties], and define possible constraints [i.e. class restrictions] on using those relationships. [W3C Semantic Web Activity]*

But as Semantic Web 'ontologies' are not common knowledge (there are many different 'ontologies' on the same domains), there is an 'interoperability problem' there.

[3]We use concepts to gather information about subjects that is likely to persist. For example, if we have a concept of Oscar, the next time we identify Oscar we may already know he has red hair, speaks Dutch, and always can be trusted. If we have a concept of tigers, the next time we identify a tiger we may already know we have to run.

World views are not common knowledge Would one big classification system ('ontology'), which all agents would use to represent the world with, be a solution for this interoperability problem? There is a Semantic Web project that tries to develop such a world-covering 'ontology': CYC (pronounce as *psyche*; `http://www.cyc.com`). However, for such a project we should realize:

> There are multitudes of entwined substances, very very many more, surely, than we have ideas of. The ones that are picked up by thought and by language are only those that have properties of interest to us. [Millikan, 2000, p.30]

A *world of senses* [Frege, 1892] is very broad. Such a shared classification system or 'ontology' should represent all not referencing subjects as well, e.g., 'Odysseus', 'Alice' (from the story Alice in Wonderland), and 'the mad hatter'.

Moreover, agents have own interests, thus will pick up other ideas. An *enCYClopaedia* will not soon or never be complete.[4] Also, a shared ontology restricts the autonomy of the agent. "A common ontology forces an agent to abandon its own world-view and adopt one that is not specifically designed for its task" [van Diggelen et al., 2006].

Finally, our project is directed to the abilities of identifying. *Conceptions* (the ways we identify) are different from person to person [Millikan, 2000].

Mapping world views Would mapping different classifications be a solution? There has been quite some research to the alignment of 'ontologies'; e.g., Klein [2001]. We consider three serious problems with the mapping of different classifications.

The first problem is that different 'ontologies' may depart from completely different world views. See Semy et al. [2004] for an analysis. Mapping may be impossible.

Second, classes adapted to particular interests of the agent draw somewhat artificial boundaries, and therefore classes may cover approximately the same instances, but still be not equal.

Third, (van Diggelen [2007] calls this the "symbol grounding problem" for knowledge representation) expressing relations between classes does not give us abilities to reidentify, because every class symbol on itself is void of meaning.

For these reasons we do not consider the *mapping of classifications* to be a serious solution. Instead, the interoperability problem should be solved by proper *identification* of represented content.

1.2 Towards identification

Because one big encyclopaedia is impossible, the different agent world-views are not common knowledge; and therefore *identification* is to play the key role for solving the 'interoperability problem'. A classification system can only be used for semantical reference if we are able to reidentify the classes that are used (see Story 1 for an example).

[4] Still, an encyclopaedia is useful for looking up detailed and comprehensive descriptions, and a lot can be learned from them (finding new concepts as well as sharpening existing concepts).

Story 1 V. coming to know the class 514–Topology.

A stranger, Mrs V. visited the library of the town. Librarian Mr L. helped her: "In this shelf you will find your book, madam!" Looking to the colorful row of books before her, V. asked: "What is this shelf about?" "Oh, this is the collection 514–Topology, which is part of the collection 500-Natural sciences & mathematics, which is part of the collection of everything." spoke the librarian proudly. "Em. . . could you then give me information about 500-Natural sciences & mathematics?" "Yes, madam," said L., and he handed her the thick handbook of science.

So the main thing is not to specify a classification system (or world-view) that the agent uses, but to enable the other agent to reidentify the meaning of the representations the agent might use.

Expression of concepts The subjects an agent is referring to have to be described to other agents. But how to express these descriptions?

We will call the whole of concepts of an agent, the "conceptuology" of that agent, analogous to the word "ontology" in Semantic Web.

Definition 5 *The whole of concepts of an agent is called the* conceptuology *of that agent.*

If the expression is again relative to a conceptuology, with which conceptuology should these descriptions be expressed? Take for example a bee which has a concept HONEY. If we were to describe this concept such that a bee agent can communicate with other agents, we could describe it in the conceptuology of a brown bear, using his concepts FOOD and *take out with pawn from bee holes*. But then agents that do not know the bear conceptuology have a problem: they may know FOOD, but not the concept *take out with pawn from bee holes*; although they may have a concept BEE HOLES, the bear conceptuology doesn't need to have a separate concept for that. And is the bear conceptuology rich enough to describe the concept *bee dance* as well?

Language Obviously, these descriptions cannot be expressed in a universal conceptuology, because a universal conceptuology does not exist. However, there does exist something quite universal *we* use to express ourselves: *natural language*. *Words* and their use are shared. So Semantic Web 'ontologies' can write descriptions using structures of words. Using a natural language lexicon (such as English) offers a large vocabulary and a direction to shape the concepts. (An artificial, evolutionary language could also be used (see Steels [1997a]), but then Semantic Web agents first have to conceptualise their world.)

Millikan [2000, Chapter 6] argues that language can be just as rich as perception to learn us a concept:

> My claim is that having a concept grounded only through language is no different than having a concept grounded only through, say, vision. [Millikan, 2000, p.90]

This is a clear motivation for us to investigate ways of providing artificial agents in Semantic Web with conceptuologies. Our project is the research of how a conceptuology with conceptions (abilities to reidentify) based on language-like representations can be provided for Semantic Web agents.

1.3 Research questions

The main research question we mean to answer: Is the use of a conceptuology with conceptions based on language-like representations a realizable and valuable alternative, for artificial agents, to the use of shared classifications in 'ontologies'?

We can formulate the following subquestions for this research question.

Description of a conceptuology First, if we want to use a conceptuology, we need to know what kinds of concepts there are that we use.

Question 1 *What different kinds of concepts (abilities to reidentify) are there?*

Second, to investigate more properties of concepts, we will investigate more about concept ontology.

Question 2 *What are the properties of, and relations between concepts?*

After this, we will compare our starting point with the starting point of 'ontologies', consisting of class definitions.

Question 3 *Can class definitions as in 'ontologies' give us concepts?*

We show there are several problems with using class definitions for identification.

Realisation of a conceptuology Artificial agents may only have representational information to get concepts from, not presentational information (they do not see, hear, feel, etc.). Therefore,

Question 4 *Is it possible to have concepts through representational statements alone?*

We will show that this is possible, comparing how humans are able to use representations, and investigate which representations can be used for concepts.

But before answering this question, we will take some time to explain how representations can be formed.

Question 5 *How do we represent things? What does representation involve?*

Evaluation of using the conceptuology Concepts are abilities that may fail. We will analyse at which places a conceptuology based on representations in particular may fail.

Question 6 *How fallible are concepts?*

Finally, we will touch on the use of the conceptuology in a broader perspective of agent communication.

Question 7 *How can agents use conceptuologies?*

Implementation of the conceptuology After having evaluated and analysed realisation of these conceptuologies, we will give a proof of concept by the system CROC[5]: a Representational Ontology for Concepts. That will answer the implementation question:

Question 8 *How can such a conceptuology actually be implemented?*

1.4 Overview

In Part II we will answer our main theoretical questions one by one. It contains:

1. A sketch of concept kinds, in Chapter 2;
2. A more detailed ontology of concepts, especially substance concepts, in Chapter 3;
3. A review of class definitions as in 'ontologies', in Chapter 4;
4. How representations are related to concepts, in Chapter 5;
5. A solid theoretical and practical foundation for grounding concepts using representations, in Chapter 6;
6. How fallible concepts based on representations are, in Chapter 7.
7. Collaboration of agents in Semantic Web using conceptuology, in Chapter 8.

In Part III we discuss the details of the practical implementation called CROC: a Representational Ontology for Concepts, in Chapter 9. Some practical examples are presented in Chapter 10.

In Part IV we will summarise our findings to answer the main research question.

1.5 Relevance for cognitive artificial intelligence

To position the research presented in this thesis in the domain of Cognitive Artificial Intelligence (CKI), we mention its most important roots.

Our research is about communication in *multi-agent systems*. Woolridge [2002] sums five keywords that explain the idea behind multi-agent systems: 'ubiquity' (computers and parts of intelligence everywhere), 'interconnection', 'intelligence' (more complex tasks), 'delegation' (because of *distributed* capacities) and 'human-orientation'. To cite about the last keyword: "The fifth and final trend is the steady move away from machine-oriented view of programming toward concepts and metaphors that more closely reflect the way in which we ourselves understand the world."

The perspective of *concepts* (from Millikan [2000]) from which we start, is really 'human-oriented'. The key notion of Semantic Web is that content, services and agents are *distributed*, and that is exactly what is the case. In

[5]The name "CROC" is an acronym, and it stands for "CROC: a Representational Ontology for Concepts". The 'C' stands for itself.

that perspective, Semantic Web focuses on practice. Semantic Web is directed to *interconnection*, and from that perspective our research question is directly derived.

As mentioned, reidentification is the central process of cognition. Therefore incorporating conceptuologies for intelligent agents in the Semantic Web is highly relevant for the field of cognitive artificial intelligence.

We believe that the notion of concepts as central to cognition has been largely ignored by artificial intelligence systems. Classical logic used classes for subject descriptors, or rather one-place predicates to express membership of a subject class, and n-ary relations, or rather more-place predicates for describing happenings, completely ignoring happening concepts and subject concepts. Reidentification was described as a static *judgement* instead of an ability *for a purpose*. Perhaps our use of language expressions may shed new light on natural language processing.

1.6 Related work

Within the field of Semantic Web, a lot of research has been done to the problem of combining and relating 'ontologies', and the solutions of mapping, aligning and merging (Klein [2001]; de Bruijn et al. [2006]; Wache et al. [2001]). We have already argued that merging will not be considered, because conceptuologies are taken to be decentralized. The techniques used for automatic ontology mapping are interesting, and possible to relate to our approach: there have been extensional methods (comparing the occurrences of a type; e.g., van Diggelen et al. [2006]), intensional methods (comparing properties of a type) and other semantics-based methods (e.g., Sabou et al. [2006]), but also methods that use, e.g., WordNet to map terms.

Objections to the current approach of Semantic Web 'ontologies' have also been made by, e.g., Shirky [2003], and Gärdenfors [2004]. The latter also comes up with a different 'conceptual' approach, which originates from the prototype-view of concepts (see Section 6.3.3 for a discussion).

From a very different perspective, Steels [1997a] describes an approach to language formation and adaptation between artificial agents. Agents in the same environment invent new words for property (e.g., colour) values of the things they encounter, and try to adapt their language with the other agents. We aim to reuse existing natural language lexicons instead of arbitrary new words, but the matching technique may be similar.

Finally, within the field of conceptual semantics (called I-semantics by Jackendoff [1989]), there have been different researches that investigate categories of concepts. Especially Jackendoff [1989] describes conceptual categories, comparing to natural language, in much detail. However, the analyses of concepts have largely been from the definitional perspective on concepts, which we reject (see Section 6.3.2).

Part II

The concept

Concepts

Chapter 2

Sketch of concept kinds

In the introduction, we defined the word ‘conceptuology’ as the body of concepts (abilities to reidentify) possessed by an agent.

To consider how such a conceptuology can be realized for artificial agents, we first need to analyse which different kinds of concepts (abilities to reidentify) there are. We will answer this question first by classifying *presented things*, and the concepts that may reidentify them; then we will see how these different concepts relate, and how they form the basis for our *re*presentations.

2.1 Presented things

What kinds of concepts are there? We will try to give a brief sketch of the world of concepts, starting with substance concepts as Millikan [2000], relating happening concepts with them, and continuing with properties and predicates, and a final word about other subject concepts.

In the literature there have been many other sketches of ‘categories’ of concepts, starting with Aristotle. A recent example is Jackendoff [1989]. Our purpose here is to place the categories we mention in the framework of Millikan [2000], which describes (substance) concepts as *abilities to reidentify for a purpose.*

2.1.1 Substances

There are substance concepts. Millikan [2000] describes these, and references to Aristotle. I can see, hear a bird. I can eat, taste, and smell fruits.

There are different types of substances. Millikan [2000] distinguishes kinds (like a car, an Audi), individuals (like my car), and stuffs (like water). A kind is a type of token individuals. A stuff does not have a clear marker for individualism.

In the following, we will display substance concepts like this: MY SUBSTANCE.

2.1.2 Happenings

Events and actions

How do I see an action? An action seems to involve a complex of things-to-learn-about. If I see a man hitting his car, I see the man, I see the car, I see the man's hand, and (this makes me) see and hear the man hitting the car. What is different in seeing the man hitting the car from seeing the man is that there is *time* involved. Without time, a man cannot hit, a chair cannot stand, and nobody can own his car.

Actives, passives, to be and to have (which situations) all fall under the notion of happenings (which Aristotle distinguished).

I can reidentify time-things just like I can reidentify substances. I can see and hear the man hitting his car. If he would hit me, I would feel it. I can see and feel a vibration or movement.

There are happening kinds (like a coronation), individuals (like the coronation of the Queen), and stuffs (like movement).

In the following, we will display happening concepts like this: *my happening*.

Representing properties or time-things always has its limits. It is a *description* of the relation the things have, a putting-things-in-script.

States of affairs

What about states of affairs? We have concepts for them, which are models for worlds. Of course we have, because they are much more easy than happening descriptions in a lot of cases. Take the happening description "The flower is in the vase on the table.": we regard this as a state of affairs, not just as a temporal description (we create our model for it). But most of the states of affairs cannot be frozen forever — they are soon *translated* to happenings, temporal statements. That is why we communicate in happenings; creating models for states of affairs is a efficient reasoning tool. We will focus on the implementation of happening concepts instead, the more generic way.

2.1.3 Properties

I cannot see a property. The properties looks-as, sounds-as, feels-as, tastes-as, smells-as are very near to us. (But electrical conduction is a basic property we cannot sense in a basic way.) Looks-as can be described in terms of colour and shape and size and place; feels-as in terms of texture and mass and temperature and humidity and stickiness, and so on. Properties are all things that simply come with things-to-learn-about.

In the following we will display property concepts like this: *my property*.

2.1.4 Predicates

All values for these properties, such as white, yellow, triangular, rounded, wet, cold are qualifications about how they look, or how they feel. These qualifications about subjects are *predicates*. This term is (like the term *substances*) borrowed from Aristotle (Aristotle, §2).

Other predicates may involve a more complex judgement, like armed, empty, grammatical, poor, eager, fake, and may involve subjectivity, like beautiful, fast, big.[1]

To come to these descriptions, we need to be *able to apply* these predicates. We cannot track properties and property values as we track substances or happenings (by their properties); but we needed to learn the words (by tracking) to be able to use them, to derive them or use them in derivations.

Predicate-application is a mechanism that also allows use of complex predicates (relation predicates (Section 2.1.5) and happening predicates (Section 5.4)). Therefore we will distinguish 'atom predicates' from other (complex) predicates.

In the following, we will display atom predicate concepts like this: *my predicate*.

2.1.5 Relations

There are many relation words which are used in complex predicates. Take for example "between ... and ...", "like ...", "on top of ...". These relation words are used in combination with subject arguments to form predicates.

Just like atom predicates, relation predicates are qualifications; unlike atom predicates, they involve more subjects. But we should have abilities to reidentify relations between subjects: we have *relation concepts*.

In the following, we will display atom predicate concepts like this: *my relation*.

2.1.6 Other subject concepts

Returning to subject concepts, there are quite some subject concepts that are not substances. Take for example unit concepts, number concepts, symbol concepts, but also the concept of a car brand, the concept of a class, the concept of a colour, and so on. These are often abstractions that are used to classify (kinds of) substances (see Section 3.3.3).

We also have time concepts and place concepts, we have time interval and path concepts. These are also subjects, we can reidentify, e.g., last year and home, but they are very different from substances. However, here again there are kinds (like a year), individuals (like past year).

In the following, we will display other subject concepts like this: MY SUBJECT.

Relational subject concepts

Sometimes subject concepts have their intension in a relation: for example, the concept for "relative". We do not have a real subject kind concept for "a relative"; we have a concept for "relatives *of* ...". We might however have abilities to reidentify "directors" or "mothers" independently as well; although their real intension is for use in relations, they may be regarded as concepts on their own.

[1]Note the difference between these predicates and basic predicates. Something cannot look armed, or feel empty. Moreover, the class we call "things" shares all basic properties: all 'things' have a colour, a shape, can make a sound, have a taste, a size, a temperature, and so on. Not everything can be empty or rich, these predicates are only applicable to certain substances.

Concept class	**Notation**	**Examples**
Subject concept	MY SUBJECT	A CAR BRAND, A COLOUR, A YEAR, MY BIRTHDAY
Substance concept	MY SUBSTANCE	A CAR, HONEY, OSCAR
Happening concept	*my happening*	*is*, *throws*, *her birth*, *hill climbing*
Property concept	*my property*	*colour*, *shape*, *temperature*
Atom predicate concept	*my predicate*	*poor*, *foolish*, *loose*
Relation concept	*my relation*	*from*, *inside*, *above*

Figure 2.1: Concept kinds.

2.2 Relating presentations

How does this all relate? When we start with perceiving, these are basic properties. From these we *are able to identify* substances, and happenings. A diversity of predicates can be applied to everything we identify.[2]

2.3 Conclusion

In this chapter, we answered the question 'What different kinds of concepts (abilities to reidentify) are there?'

Concepts for the things presented to us can be categorised in a set of basic classes. We have considered substance concepts, happening concepts, property, atom predicate, and relation concepts. Also, we mentioned that many other possible abstract concepts fall under subject concepts.

In the following chapter we will investigate properties of and relations between different kinds of concepts in more detail.

[2]See `http://croc.sourceforge.net/crocodile.svg` for a nice picture of the concepts for presented things.

Chapter 3

Concept ontology

Now we have a general overview of concept kinds, we will turn to our next question: What are the relations that can be learned between concepts? How do these relate to the relations between classes in a classification?

In this Chapter we will first recall what has been written by Millikan [2000] about *substance concepts*. Our discussion of the relations between subject concepts will be worked out in more detail. Most of what is said about substances (like a car, a mouse) also holds for other subjects (like the species Gliridiae, a car brand).

3.1 (Most) substances are not classes

Most kinds are not classes [Millikan, 2000] [Millikan, 2005, Chapter 6]. A class has a *definition* that states *common properties* of its members. A kind like the species dog is not a class (nor a fuzzy class), because it is not true that each instance resembles (or has same properties with) *all* other instances. "There are no properties that *every dog* has in common with *every other dog*." [Millikan, 2000]

Instead, what holds the group together is that its instances are *causally related*. This causal relation may have the *result* that certain properties are probable to be shared among all instances. "There *is* a *good explanation* of why *one* is likely to be *like the next*." [Millikan, 2000]

But, it is not possible to give a *class definition* for (most) substances. We will come back later on (class) definitions in Chapter 4 and Section 6.3.2.

3.2 Analysing substance properties

3.2.1 Substance templates

Still, we as humans use some property descriptions for efficiently describing concepts. Millikan [2000] mentions "substance templates": "Having a concept of a substance requires (. . .) understanding what sorts of predicates will remain stable over encounters with the substance, that is, what some of the meaningful *questions* are that can be asked about the substance. You can ask how tall Mama is, but not how tall gold is."

Substance templates accompany substance kind concepts, e.g., A PERSON, A SPECIES, A CHEMICAL ELEMENT. They bring understanding of the concepts that are instances of them.

All (most) properties of substances that are specified are probable and associated (not constitutive for the concept), and they can be clarified by substance templates. Substance templates are tools for better induction, they describe which properties are likely to persist.

There indeed are many substance templates:

> Huge numbers of substances are not merely substances, but bring with them templates for more concrete substances falling under them. For example, the ability to identify cats is easily applied to discovering what sorts of questions can be asked about individual cats. What colour is this cat (it will not change as with chameleons), is it tame or untamed (not applicable to flies), and does it have feline leukemia (not applicable to dogs) or a loud purr? [Millikan, 2000, p.83]

For concepts that fall under multiple concepts, multiple substance templates apply. For example, MR. HIPPOCRATES can be A MEMBER OF THE SOCIETY, A DOCTOR, and A PHILOSOPHER all together, in which case questions from each of these can be asked about MR. HIPPOCRATES.

Because templates of questions do not only apply to substances, but also to (more abstract) subjects (like CULTIVAR), we will instead of "substance templates" speak of "subject templates".

3.2.2 Other properties

Properties of some substance that do *not* support induction for this substance are not useful for describing concepts we might form about this substance, but can still be used for representation of this substance. For example, a car can be represented using the properties *brand* (persistent) and *place* (not persistent): "the Peugeot here".

Subjects have different properties, thus they need to be described. The concept GOOD does not have a *place*, nor has the concept PEUGEOT BRAND or SPECIES CAT, but A CAR or THE UNIVERSITY OF UTRECHT has a *place*. All substances have a place.

3.3 The 'is-a' relation

Having a concept A BOOK also includes the ability to reidentify a book every time the ability to reidentify the book 'On Clear and Confused Ideas', or a book of mine, succeeds. The same way, having a concept A HUMAN BEING would include the ability to reidentify a human being every time the ability to reidentify Mr. Hippocrates succeeds, but also every time the ability to reidentify a patient succeeds. This is an 'is-a' relation. "Patient" is called a hyponym of "human", or "human" a hypernym of "patient".

3.3.1 The ‘is-a’ relation and substance templates

Substance templates inherit via the ‘is-a’ relation. If certain properties of animals support induction, these properties are also good for induction on mice and cows. That is, the substance templates for a mouse and a cow *inherit* from the substance template for an animal, because a mouse *is* an animal and a cow *is* an animal.

3.3.2 The ‘is-a’ relation as induction

We know a doctor is a human. However, we might at some time meet a robot doctor, which is not a human.[1] In this case the ‘is-a’ relation is inductive.

This may be more clear if we reflect on the way we form subconcepts, which is by induction. A child going to the zoo the first time might never have seen or heard of a gibbon, a chimpanzee, a bonobo. But the child identifies them as an ape. When the child however learns how to discriminate the different kinds of apes, why should it lose its ability to reidentify for example gibbons as apes? That is why an ‘is-a’ relation is learnt. It can also be the other way around: we can learn the concept a person after we learned Papa, Mama, and me.

We should distinguish another case where the ‘is-a’ relation fails, which is not because of inductiveness. We might be confronted with a stone lion statue at the zoo, which is a lion but is not an animal (the example is from Kamp and Partee [1995]). Our concept for lions *fails* if we identify it as just a lion; stone lions are not dangerous and no reason to run away for. On the other hand, we identify it as something of stone and combine it with our identification of the lion shape. Stone lions are not lions; they are *representations* of lions. The predicate “stone” in “stone lion” is therefore not separable from it — in Section 5.5 we will come back on this kind of composite intensions.

3.3.3 Abstractions from substances

An important occurrence of the ‘is-a’ relation is in *abstraction*. For example, consider the difference between the *substance* concept an Audi and the *subject* concept Audi (brand). *an* Audi is a car; Audi is a car brand (with, e.g., the property *has chairman*, a property a car doesn’t have). There is a difference between “Audi again” and “an Audi again”. However, “an Audi again” immediately leads to “Audi again”!

We can describe *an* Audi as a car which *has brand* Audi. Here, the concepts a car, *has brand*, and Audi together give background information about the concept *an* Audi. However, someone that has the concept *an* Audi does *not necessarily* have the concept Audi and *has brand*, just like I don’t need the concepts Elstar and *has cultivar* to have the concept an Elstar apple. There are a lot of nice examples which can illustrate the same. I don’t need a concept of me to have a concept of my books. I don’t need a concept of the

[1] In the following we will also discuss the difference between a doctor (a person) and doctor (an occupation). This difference should not lead us to believe a robot doctor is not a doctor (but only is identified as a robot which *has the occupation* doctor), because we do not need to have this abstraction from a doctor. It is really the ‘is-a’ relation that a doctor is a human which is inductive.

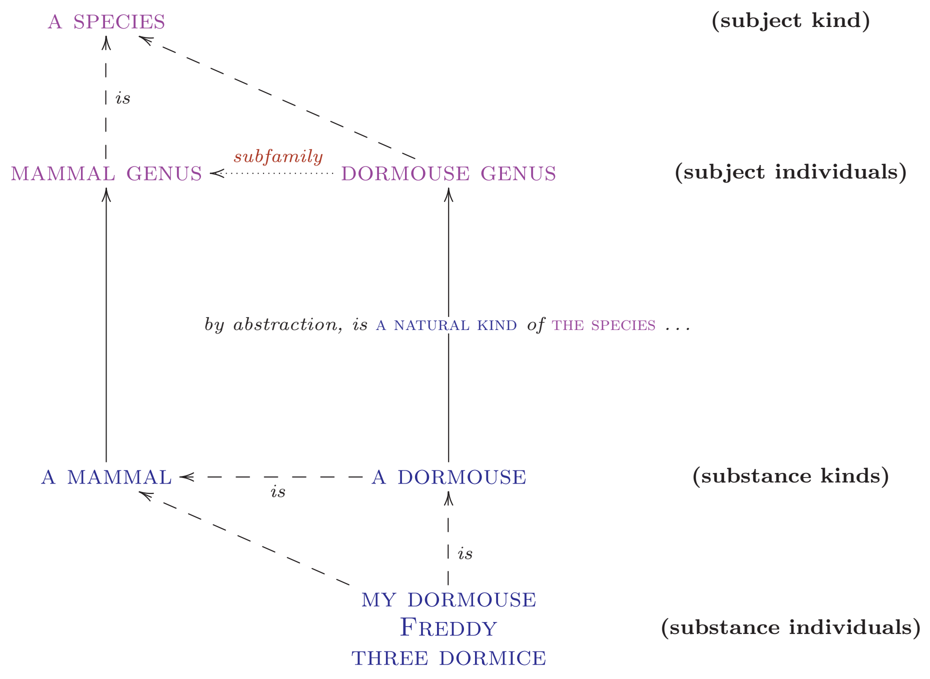


Figure 3.1: Different ‘is-a’ relations between concepts. Notice the transitivity of the hyponymy relation.

second edition of ‘Alice in Wonderland’ (with, e.g., the property *print run*, thus it is not a subconcept of a book!) to have a concept of a second edition of ‘Alice in Wonderland’. I don’t need a concept of the species mouse (with, e.g., the property *has population*) and *to be of a species* to have a concept of a mouse.

Trying to give this kind of background-information with concepts is called *“abstraction”*. Abstraction is the process of creating concepts of properties (e.g., *has brand*) and substances (e.g., Audi) as relations.

Abstraction is only possible when there is a more abstract concept available: we can only abstract from an Audi if we have, e.g., a concept a car, together with concepts *has brand* and the Audi brand. We can go on like this: a car is an object of *type* car.

There are multiple abstractions available. A car might also be described as a transport vehicle which *number of wheels* is 4.

We humans often abstract. Perhaps this has to do with creating good classification systems. Abstraction also is a tool for making more and more precise what you mean, because you link to other concepts. But there is not always consensus about the right abstractions. . . ! — they are a matter of science.

Abstractions are often tried to be used as *definitions*, but the abilities to reidentify abstract subjects are often harder to use than abilities to reidentify concrete subjects. For example, reidentifying a mouse by reidentifying it as a natural kind, and then using a universal way of determining its genus (e.g., by gene analysis), is harder than just reidentifying a mouse. It is generally not a good idea to *substitute* concrete concepts by abstract concepts, even though science may have consensus about the definition.

3.4 Happening roles

Subject concepts may interact with concepts for happenings: they may be formed *derivatively* from happening concepts, determined by the *role* in the happening.

Happening concepts relate sets of substance concepts (as subjects, objects, etc.). For example, *writing* involves:

- an actor ('agent role') which is (often – inductive) a human being;
- an story as object ('patient role');
- (often – inductive) using a pen, ink, paper, etc.

Jackendoff [1989] enumerates the role types ("thematic roles") that may be involved in happenings.

Implicit roles Notice that some roles may be implicit in sentences. Take for example *measuring*, which does not only involve the measured property in the 'patient role', and the measuring agent in the 'agent role', but also the *measurement* in a 'result role'. For a concept of A MEASUREMENT, the concept of a 'result role' is indispensable.

3.5 Conclusion

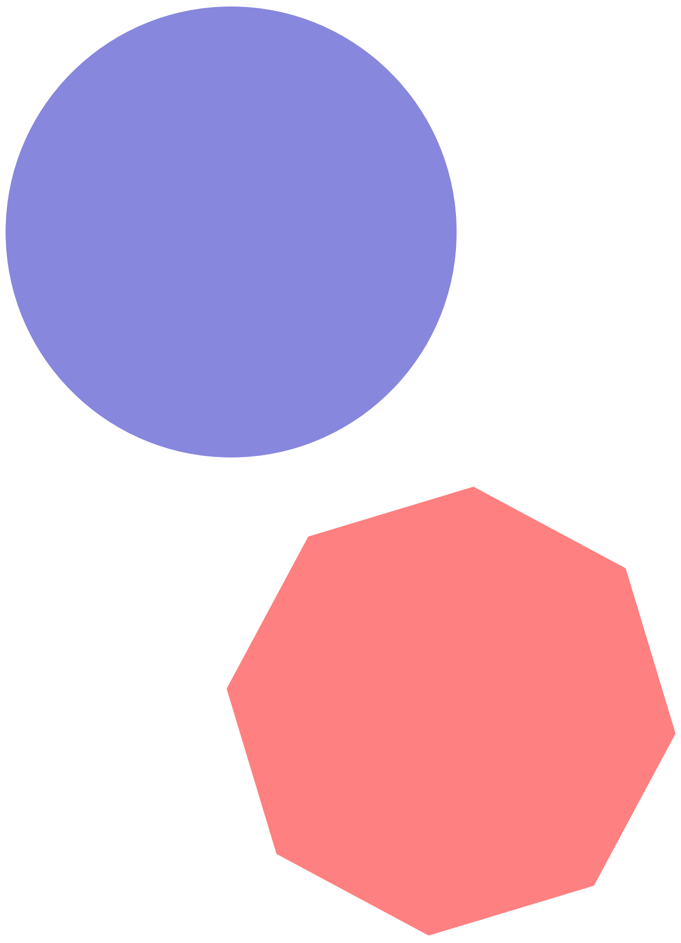
In this chapter, we have considered the question 'What are the properties of, and relations between concepts?'

We discussed properties of subject concepts. (Most) substances are not classes. For a subject kind, there are properties that support induction: the subject template.

The 'is-a' relation is an important relation between subject concepts, but it may be inductive. Abstractions that make use of the 'is-a' relation are a widely used mechanism for giving background, but it is generally not a good idea to substitute concrete concepts by their abstract descriptions.

Finally, subjects may be related with happenings by thematic roles.

We will show in the next chapter how the class definitions in Semantic Web 'ontologies' differ from concepts as described in this and the previous chapter. After that, we will give the outlines of a conceptuology for agents using representations, which is built on the basic insights about concept ontology that have been explained.


Classes compared to concepts

Chapter 4

Classes compared to concepts

Semantic Web 'ontologies' are world-views, classifications that give representations a place. In the introduction we explained why the use of different classifications will lead to the 'interoperability problem'. And a classification of the world of senses is difficult to share: it would be a classification system with as domain every thing (including nothing. . .)!

But a different perspective is that 'ontologies' were designed not for classification, but for identification, by use of the definitions of the classes they contain. That would be subscribing to the classical conceptuological philosophy that concepts have *definitions* (see also Section 6.3.2), and that concepts cover shared 'properties' and 'property value restrictions' as classes.

We have already argued in Section 3.1 that (most) substances *are* not classes (because they do not have common properties). We will here proceed to show that treating substances as classes does not only fail to cover natural substances, but also poses serious problems for identification.

In short, we will answer the question: can class definitions as in 'ontologies' give us concepts? We will therefor compare 'ontologies' with 'conceptuologies'.

Similar criticisms of the current Semantic Web 'ontology' approach can be found (at more length) in, e.g., Shirky [2003], also referenced by Gärdenfors [2004].

4.1 Ontology versus conceptuology

The philosophical use of the term ontology refers to the study of things that are, conceptuology refers to the study of our abilities to reidentify things (for a purpose). Differences are apparent:

- conceptuology will freely describe things that might not exist (like Odysseus), ontology treats all non-existing things as equal;
- concepts of, e.g., "fake book" are not ontological statements (but they have a purpose);
- concepts for kinds cover a set of ontological references (for a good reason);

- there may be missing or confused concepts.

We argue Semantic Web agents should rather describe conceptuologically than ontologically. The term 'ontology' in computer science is a bit confusing, because it has conceptuological purposes.

But it is a matter of view. If we agree with Frege (Frege [1892]) that there is a 'world of senses', perhaps ontology can describe senses as well.

Definition 6 *The* reference *or* extension *of an expression is what objects that expression denotes. The* sense *or* intension *of an expression is how the mind grasps the meaning of the expression.*

However, the project of using conceptuology descriptions for agents is not an ontological approach. It is directed to the *abilities* senses give us to identify subjects. *Conceptions*, the ways we identify (using intensions), differ from person to person [Millikan, 2000].

4.2 Subjects: classes instead of concepts

Current 'ontologies' contain class descriptions, definitions. Classification systems have the advantage that they can mention a class and immediately by class deduction derive superclasses; that they can classify by pointing which way to go in the classification tree and reach a unit class in a few steps. But to get these efficient properties, there are restrictions that have to be applied to the kind of class descriptions.

4.2.1 Classes as more artificial, and therefore difficult to match

From Chapter 3 it should already be clear that class descriptions are different from substance concept descriptions, because most substances are not classes.

Classes have an artificial flavour compared to substance concepts, because they use *property restrictions* instead of inductive properties. They are artificial in their boundaries:

> Where substance boundaries are vague in nature, the purposes of *classification* are sometimes served by drawing artificial boundaries around the extensions of these substances. [Millikan, 2000, p.38]

They focus on a particular set of properties, which identification does not require:

> Unlike a task of classifying, the task of identifying a substance doesn't require that any one particular set of the substance's properties be known or manifest to one, or that different people should use the same properties of the substance in order to identify it. [Millikan, 2000, p.38]

Therefore: classes are difficult to match. It even makes it impossible to have *same* classes when the whole classification tree is not the same!

Argument 1 *Classes have artificial boundaries, and focus on a particular set of properties, which make them very hard to match.*

4.2.2 Class hierarchy

Substances are not generally organised into tree or grid structures [Millikan, 2000, p.38]. Millikan [2000, §2.6] describes the hierarchy among substance concepts as "a logical mess":

> Aristotle thought there was a hierarchical ordering among all substances. According to the doctrine of "real definition" or of natural ordering by genus and differentia, substances were supposed to form a logical tree. I think this doctrine was seriously wrong. The structure of the domain of substances is frankly a logical mess, a mare's nest of overlappings and crisscrossings. There are multitudes of entwined substances, very very many more, surely, than we have ideas of. [Millikan, 2000, p.30]

Although substances always can be forced to be placed in a hierarchy, this is a way of "projection" that does not completely do right to the substance:

> Consider, for example, the substance *chair*. I have argued that this is a historical substance, but what substance template do chairs fit under? It is clear that one would not want to project a science of furniture, for example, for although there may be one or two questions pretty certain to have answers for each kind of furniture (what was it designed to be used for?), there are not nearly enough to delimit in advance all or most of the determinables that are relatively reliably determined for most chairs. [Millikan, 2000, p.30]

Classes pushing definitions from abstractions

Classes are pushed to use abstractions, and treating such as the class *definition*. A classification tree cannot do without using the 'is-a' relation. An abstraction is the definition in an 'is-a' substance concept (e.g., A CAR), property concept (e.g., *has brand*), and substance concept (e.g., AUDI).

However, it was argued (3.3.3) that to have a *concept* of AN AUDI, we don't need a concept of, e.g., *has brand*. But when we want to define a class as part of a classification tree, we are forced to use abstractions. The definition (or description) of abstraction lays an extra burden for intelligent agents. Every agent that wants to place the substance concept AN AUDI in a good classification tree, *is forced* to have a class definition of, e.g., *has brand* and THE AUDI BRAND.

Argument 2 *Ontology definition (defining a class as part of a classification tree) lays an extra burden for intelligent agents.*

Compared to concepts: concepts do not make part of on a classification tree. The 'is-a' relation is useful for having richer concepts, but not necessary. (Instead, a concept may just be a name or an image.)

Sharing a class hierarchy doesn't share concepts

Class definitions often rest on concepts. To continue the example, the classification of AN AUDI rests on the more abstract concepts *has brand* and AUDI. Will a classification therefore always have concepts as its building blocks?

Yes, that is inevitable; but in ‘ontologies’, the classification ends in *very abstract* classes: e.g., `owl:Thing`, `owl:DatatypeProperty` or `xsd:string`. For example, AUDI again can be abstracted to CAR BRAND with the *name* “Audi”. The current approach of ‘ontologies’ is to continue with definitions until some very basic concept like `owl:Thing`, `owl:DatatypeProperty` or `xsd:string` is reached.

The problem is that during such abstraction, our abilities to reidentify for such classes will almost disappear. The only concepts that remain are concepts for, e.g., `owl:Thing`, `owl:DatatypeProperty` and `xsd:string`. Sharing a class hierarchy does not share concepts for the classes that fall below the most abstract classes.

Sharing such ‘ontologies’ will not really give us concepts to share, that is, unless we agree about all abstractions that were made in the classification tree (that is, when we both classify exactly the same way), which we know from the real world often not is the case.

Argument 3 *Sharing ‘ontologies’ doesn’t really give us concepts to share.*

4.2.3 The context problem

Classifications, that depend on *deductive* mechanisms, are sensitive to context changes. For example, when the class of DOGS is defined to have two EARS, a dog will change into a non-dog when it loses one ear. Concepts are more stable because they are abilities to reidentify, and merely use inductive knowledge.

The changing classifications per context need to be mapped to non-changing concepts. An approach that tries to do this with classification is using metamodels (abstract classifications), and specify transformation rules for different contexts to translate to concrete classifications. Using concepts as a basis instead provides more simple and flexible solutions.

Argument 4 *Class deduction is sensitive to context changes.*

4.2.4 The symbol grounding problem

Grounding is about translating representations (classifications) to their extension. A substance concept, an ability to reidentify something for a purpose, is already grounded.

If representations are not mapped to concepts, there is a symbol grounding problem. A symbol in itself has no meaning. Symbol manipulation via some syntactic rules is void of meaning.

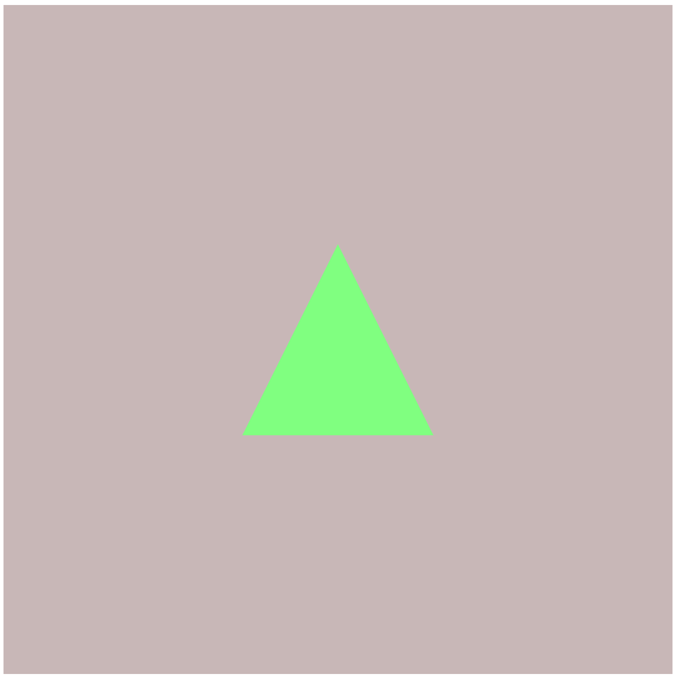
Argument 5 *Concepts are needed for grounding symbols.*

4.3 Conclusion

In this chapter, we have compared our starting point with the starting point of ‘ontologies’, consisting of class definitions, answering the question ‘Can class definitions as in ‘ontologies’ give us concepts?’

We argued that treating substances as classes will pose serious problems for identification. First, if ‘ontologies’ have to serve abilities to reidentify, they cannot have an ontological approach to senses, because conceptions differ from person to person. Second, class definitions are more artificial (and therefore difficult to match); class definitions push definitions from abstractions, thereby placing an extra burden for intelligent agents; sharing the class hierarchy doesn't share the concepts that shaped that hierarchy; class deduction is sensitive to context changes; classes without concepts are not grounded.

From these arguments we conclude it is time for a very different approach to provide agents with a conceptuology. The basis for this approach is set out in Chapter 6. Preceding this chapter we will first investigate an important mechanism that is closely related to conceptuology, and that will provide background for the rest of our thesis: representation.



Representations

Chapter 5

Lexical representations

Having the concepts for what is presented to us, we use mechanisms to *re-present* this information.

The purpose of representations is to have a intermediate that applies to the purposes of the concepts that are used for presentations. A representation can be descriptive (describing what we believe), directive (describing what we intend) or can be both (see Millikan [2005, Chapter 9]). Questions also use representations: they are like descriptives with holes to be filled.

Representations *have semantic mappings.* They contain information *about* something.

The content of a representation is determined by its purpose; for a descriptive, e.g., we choose to represent the identified in which we have the most confidence.

Concepts give the ability to handle representations.

As presentations in the real world involve complexes of identifications, representations will involve complexes of identifications. *Names* are the atoms of lexical representations: the ability to identify by name *is* a concept. Names may consist of more words; for example, in the name "dead metaphor", there are two words (which are in other contexts names on their own!); in the name "Alice in Wonderland" there are three.

This chapter will continue to investigate different complex lexical representations that can be formed with names.

5.1 Representing subjects

Subject or substance concepts can be used both in representations of subject intension ("yellow flower") as in representations of subject extension ("Oscar", "a flower", "gold", "this flower"). Referring to subject extension uses mechanisms as *quantification*, as in "a flower", and *determination*, as in "this flower".

Subject intensions can be complex. We may use combinators like in "friend and colleague" or "professor or doctor".

Reference to kinds Representation may also directly refer to the subject intension.

There is a difference between quantification and direct reference to kinds (see Carlson [1977]). Compare “all frogs are green” with “frogs are green”. In the case of reference to kinds, bare singulars (“a frog is green”) or bare plurals (“frogs are green”) are used. Equivalent sentences for other subject types are, e.g., “gold melts at 1337 Kelvin” or “Henry talks”. Notice that in “a frog is green” the word “a” can be read as a existential quantifier as well, for which the meaning becomes different. One can easily see if reference to kind is meant if substitution of the bare plural for the bare singular and vice versa doesn’t change the meaning.

To avoid the ambiguity in expressions like “a frog”, technically we may represent reference to kinds with directly refering to the subject intension: “frog is green”, “car has wheels and a steering-wheel”.

Yet another reference to kinds For the ‘is-a’ relation, we cannot use direct reference to kinds. In the sentence “cars are vehicles” with “cars” we have a direct reference to the kind. With “vehicles” we do not mean *a set of vehicles*, because cars may not exist. With “vehicles” we also do not mean to refer to the kind; otherwise sentences like “cars are motorbikes” and “cars are lorries” become true as well, because motorbikes and lorries fall under the subject kind vehicles.

Therefore we will represent the ‘is-a’ relation with “cars are ‘vehicles’”: we use a quoted form of “vehicles”.

5.2 Representing predicates

We have lexical representations of predicates; basic predicates, such as red, sticky, sweet, has name; more subjective ones, such as lazy; relations, such as family-of. Inside our brain, we however are likely to represent the basic properties more directly instead of representing predicates. Most computers do not have a sense of the basic properties we have a sense of; but they can of course represent these lexically.

When communicating about or reflecting on properties, they seem to involve happenings. We use the happening *is* to represent predicate statements: “frogs *are* green”, “the swan *was* white”.

5.3 Representing happenings

Happenings (such as the man hitting the car, the chair standing on the floor, the dog barking) have involved subjects, tied to roles as described in Section 3.4. Happening representations may have involved predicates as well, at least necessary for representing predicate statements (see the previous section).

Representations can be descriptive, they can also be directive, or contain both directive and descriptive elements. Happening representations are the basis of all complete representations: directives, descriptives, as well as interrogatives involve happenings as their basis. Statements on itself have truth-values, and therefore may be combined with the usual logical connectives: they can be concatenated with “and”, “or”, “if . . . then . . . ”, etcetera.

Time-slices of happenings Carlson [1977] makes a distinction between reference to kinds, reference to individuals, and reference to *instances (occurrences) of individuals*. We believe this last distinction is a mistake, and has to do with ways of representing happenings instead.

The meaning of "Henry talks" is ambiguous between the meaning of "[an instance of Henry] talks" and of "Henry talks". In our opinion, this is an ambiguity that arises from the interpretation of "talks", referring to a happening, not from the interpretation of "Henry", the entity the expression is referring to. By counting the readings of "Tweetie makes her nest in my tree" we see that only "[an instance of Tweetie] makes a nest in [an instance of my tree]" and "Tweetie makes her nest in my tree" are possible readings, not "[an instance of Tweetie] makes a nest in my tree" or "Tweetie makes a nest in [an instance of my tree]". By the way, in this example it is already hard to think about which time-slice of Tweetie is exactly referred to: that is as well determined by the happening. Making a nest involves more time than talking. The last thing is that we can also use it for kinds: "frogs are green" can be read as "today, frogs are green", "some chameleons are green" as "some chameleons are green now". However, in the past tense, e.g., in "dinosaurs ruled the world", the time-instantiation of the happening is clear: the happening is not continuous.

In short, the time-scope of the happening has to be represented as well: if the happening is continuous, or holds over a finite time distance.

Temporal logic Happenings involve tense and aspect. Happening statements can be placed in a more global framework of temporal logic of that also includes possible worlds, and happening patterns. Happening concepts often imply changing properties. To *reason* about happenings, one would need a temporal logic to specify what happens when.

Happenings can be seen as taking place at some interval of time. Knowledge of happening descriptives includes knowledge about the relative position of their time intervals (i.e., if they overlap, etc.). The time scale that can be constructed from this knowledge is continuous: a happening may be described with a sequence of other happenings, and vice versa. (There however also happen to be concepts about absolute intervals in the time space: we know about "yesterday", "last year", "1 september 1986".)

For temporal logic we need a notion of "before", "at", and "after". We can state a happening *pattern* with "if h_0(at some time:x), then h_1(at x), h_1(before x) and h_2(after x)", where we use "h_n" for a happening representation, and ":x" for a time variable assignment. An example pattern representation: "Before a door is closed, it is open". We can state a happening *sequence* with "h_1(at some time:x) and h_2(after x:y) and h_3(after y:z)". An example sequence representation: "Close the door, and open the windows afterwards".

5.4 Happening predicates

Aristotle used substance concept types like MAN as predicates as well. We certainly can use subject words like "man" in predicates, but then we mean the 'is-a' relation: "is a man" is a predicate (see also Section 3.3 about the 'is-a' relation, which corresponds to Aristotle, §3).

The predicate "is a man" is one example where a happening statement is used as a predicate. There are many other examples, for example "runs" (also mentioned by Aristotle), "talks with John", "scared by these words".

So, happenings may be used as predicates in a representation. The subject to which the happening predicate is applied often has a specific role in that happening; e.g., in "Alice, talking to John" and "Alice, talked to by John", the subject 'Alice' has agent role and patient role, respectively. But it does not need to be so; e.g., think of the happening predicate "the parents of which were lead to believe that . . . ", in which a mere reference is used to express the relation.

5.5 Predicate-subject composite intensions

Classification Besides stating knowledge about subjects, predicates can be used as *modifiers on subject intensions* in composite expressions. Take for example "robot doctor".

Predicate-applications may modify inductive knowledge. A substance concept A SWAN may have in its description that it has WHITE *colour*. This is an inductive property: there may occur swans that are not white. Applying the predicate BLACK (*colour*) to A SWAN gives a subject intension that overrides this inductive knowledge.

Deriving the mix of the predicate and the subject intension is a complicated ability (see also Section 7.3): it often involves more than a simple combination of properties.[1] For "red book", e.g., we do not think of a book that is completely red, but of e.g. a book with a red cover. We have a concept for "red", and a concept for "book", and use these concepts to *classify* which things can be described as "red books" and which cannot. More interesting combinations are, e.g., "midget giant", "stone lion" (see Kamp and Partee [1995]).

We can also misuse predicates; e.g., "blue mood", "thick voice", "lazy stone". Here no substance templates are applicable, no properties correspond; but we identify the predicate.

Idioms Sometimes predicate-subject complexes can become 'dead metaphors' [Millikan, 2000] or 'idioms' [Fodor and Lepore, 1996]. Take for example the expression "striped apple" (see Kamp and Partee [1995]), for which we develop a new ability to reidentify, because what we call "striped apples" are not what we typically take for "striped".

5.6 Conclusion

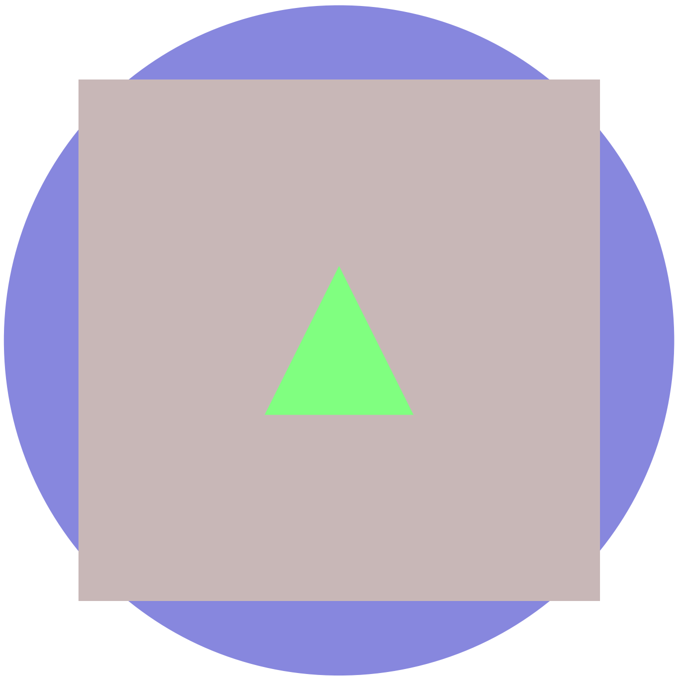
In this chapter, we answered the question 'How do we represent things? What does representation involve?'

The mechanism of lexical representation makes use of names for concepts, and is able to combine them in different uniform ways, which we analysed in this chapter.

Lexical representations therefore can be used well to state (or question) properties of, and relations between its involved subjects. We will use this

[1] We might know which properties are meant by the substance template; we expect each representation to use mostly inductive properties.

background of lexical representations in the next chapter that is about *grounding concepts in representations*.



Grounding concepts in representations

Chapter 6

Grounding concepts in representations

Now our motivation for using concepts for identification instead of classification is clear, we want to describe how concepts can be grounded in representation. The fundamental notion is that a concept is an *ability*, not a representation (or a prototype), but this ability to reidentify can be grounded in representations.

6.1 Representation

6.1.1 Concepts through language

Artificial agents are not (all) grounded in the *presentation which we humans have*, i.e. they (mostly) can't smell, taste, see, hear, touch.

Millikan [2000, Chapter 6] argues that language is an representation in which concepts can be grounded, just like our concepts can be grounded in picture representations [Millikan, 2000, §6.1]. Language may be a very important source for our concepts; small children learn a new word every hour.

> It sounds a bit queer to speak of learning a word for a substance as learning a way to identify that substance. But just as the relation of one part of the pattern on the TV screen to another part can manifest the relation of one part of Bill Clinton to another, the relation of a word to other words in a sentence can manifest the configuration of a substance in relation to other substances and properties in the world. (...)
>
> So if learning what a substance looks like can be learning how to identify it, similarly, learning a word for a substance can be learning a way to identify it. (...)
>
> It is even possible, indeed it is common, to have a substance concept *entirely* through the medium of language. It is possible to have it, that is, while lacking any ability to recognize the substance in the flesh. For most of us, that is how we have a concept of Aristotle, of molybdenum and, say, of African dormice. — There, I just handed you a concept of African dormice, in case you had none before. Now you can think of them nights if you like, wondering

> what they are like – on the assumption, of course, that you gathered from their name what sorts of questions you might reasonably ask about them (animal questions, not vegetable or mineral or social artifact questions). (. . .)
>
> [Language] enables one conceptually to track these substances and easily to discover under what sorts of substance templates they fall. [Millikan, 2000, §6.1]

So through language we can get an ability to reidentify. Millikan compares abilities to reidentify by language with other abilities to reidentify.

> Learning a language is, in part, just learning more ways to pick up information through the senses and put it away in the right boxes. A difference, of course, is that this way of picking up information is much more fallible than in the case of ordinary perception. But no human ability is infallible. Furthermore, just as substances are sometimes look-alikes in the flesh (twin brothers), many substances are sound-alikes in words (John$_{(Doe)}$ and John$_{(Roe)}$). But substances are tracked through the medium of words not merely by means of the same words manifesting the same substances. Like more direct manifestations of substances, words and sentences occur in context, allowing methods of tracking to be used. . . [Millikan, 2000, §6.1]

A concept grounded in language can consist of a single word. But what about richer concepts? Our project is to enable agents to represent through language *as much as possible* of what they know about, or of what constitutes their concepts (see Story 2).

What should agents be able to express with language to be helpful in communication with other agents? We will answer this question in Section 6.2 and the following sections.

Story 2 A concept by language alone, by a single word or by descriptions.

"What is an African dormouse?"
"Well, a kind of dormice with bushy tails that lives in Africa. Dormice are particularly known for their long periods of hibernation, . . . "
"But what is a mouse?"
"Well, a small gray animal that likes cheese, . . . "
"But what is an animal? And what is cheese?"
"Oh, I'm afraid I don't know. Someone told me so. I must have forgotten what it meant. . . I thought you might perhaps know. . . "

6.1.2 Representation of *what*?

In the previous section it was argued representing with language can be just like presenting.

If I say "it was walking on the rocks towards its nest" we all see something before us. If I add "at sundown", this picture might change. Representation lets us make a picture *using things we have seen before*, but the result picture (trying to capture all information about the things that were said) might be completely new and strange.

If we read “the bloodcurdling cry of the mountain lion has been described as the scream of a woman both terrified and in pain”, we can almost hear it, and shiver from it.

Here I described two examples that use senses that are quite important to us. But seeing- and hearing-information is not so interesting and available to the computer (yet). So what information or presentation are we talking about in the case of computers? Of course, the information we have provided them with, so the information which is relevant for its purposes. Still, representations work the same way: the computers task is trying to capture all information about the things that were said in one ‘big picture’.

The key notion is that new concepts are linked to concepts for which the agent already *has purposes*.

6.2 Providing the conceptuology

Now we have seen that concepts can be based on representations, explained what representations are, the crucial question is: how exactly will we provide a conceptuology for agents?

In this chapter we will further cover the what are the building stones of such a conceptuology. Cases of fallibility of these concepts will be covered in the next chapter.

6.2.1 Names

At the basis of lexical representations are *names*. A name is an identifier. The ability to reidentify by name exactly is the basis of a conceptuology.

A name gives the agent an *rough indication* that it is the same concept. And a *rough indication* is exactly what we need. Although the names are the same, concepts can have a little different flavours, different descriptions.

When the ability to reidentify by name fails (e.g., if the name is unknown), we need other mechanisms to reidentify. Using representational statements in the related knowledge of a subject, one can apply *reasoning* about these statements.

6.2.2 Purposes

Concepts are abilities *for a purpose*. The notion of a purpose is crucial for concepts. An artificial agent needs to have built-in purposes for at least some concepts it possesses. For example, an artificial agent may be built with desires or plans in which the happening *flying* occurs.

If an agent learns new words, the purposes of these words may be derived from the purposes of the concepts they are linked to. An agent may use *representations that link names* for having a concept with a purpose. For example, the statement “Green elephants can fly” gives a purpose to an ability to reidentify “green elephants”, if we have purposes associated with “flying”.

But an agent may also learn new practical purposes. If we want to enable others to learn a concept, why not give a description of their purpose? We can represent purposes using happening concepts with roles for agents filled in: an agent’s *actions* can be described in relation to those names. An agent’s purpose of its concept ENEMY may be that it has to destroy it. Our purpose of our concept CHAIR includes that we are able to sit on them. An agent's purpose of its concept PET FISH may be that we can sell them for such and such prices.

Not all purposes are easy. What is our purpose of our concept RED? Our purposes for concepts are (often) multiple and largely undefinable. For artificial agents, purposes might be more easy and well-defined.

6.2.3 Reasoning with representations

Represented *statements* that are in the related knowledge of a concept may include *predicates*, logical combinators, *quantification*, and *determination* of subjects.

In the following, we will examine some uses of these reasoning mechanisms for concepts in more detail.

To representational statements one can apply *reasoning* to derive new statements. Logical reasoning comes in three forms: deduction, induction, and abduction.

Definition 7 *Given a* precondition*, a* conclusion*, and a* rule *that the precondition implies the conclusion, [the logical reasoning forms] can be explained in the following way:*

Deduction *means determining the* conclusion*. It is using the* rule *and its* precondition *to make a* conclusion*. Example: "When it rains, the grass gets wet. It rains. Thus, the grass is wet."* Mathematicians *are commonly associated with this style of reasoning.*

Induction *means determining the* rule*. It is learning the* rule *after numerous examples of the* conclusion *following the* precondition*. Example: "The grass has been wet every time it has rained. Thus, when it rains, the grass gets wet."* Scientists *are commonly associated with this style of reasoning.*

Abduction *means determining the* precondition*. It is using the* conclusion *and the* rule *to support that the* precondition *could explain the* conclusion*. Example: "When it rains, the grass gets wet. The grass is wet, it must have rained."* Diagnosticians and detectives *are commonly associated with this style of reasoning.*

[Wikipedia]

Criticism to "math done with words" Shirky [2003] (which Gärdenfors [2004] cites) exhibits a very critical approach to Semantic Web as "a world where language is merely math done with words". The rhetoric is however directed not directly against *using lexical representations* and reasoning about them, but rather against the *applicability of universal affirmatives or negatives* (and their use in syllogisms and deduction). We completely agree universal affirmatives or negatives about substances mostly do not occur deductively. Most universally quantifying statements are instead inductive. Shirky [2003] does not only provide examples of deduction with bad universals, but also goes on to provide examples of induction that fail. Indeed: induction is fallible; and an inductive conclusion based on a specific set of knowledge may very well fail with other reasons. But fallibility of induction of course is not a problem, as long as the outcome of induction is not treated as a *deductive* fact.

words	[African dormouse, instance of genus *Graphiurus*, . . .]
'is-a' relations	is a dormouse, is a mouse, is a small animal
substance templates	[small animal, mouse]
related knowledge	[Fred is an * , An adult * has a body length of between 6 and 19 centimetres, There is a fictional character of an * in 'Alice's Adventures in Wonderland', . . .]
purposes	—

Figure 6.1: Representations belonging to the concept of the substance kind African dormouse.

In fact, our approach to ground concepts in representations is not at all based on treating induction as deduction, or concepts as having deductive definitions; on the contrary, we will argue against this in Section 6.3.2.

6.3 Learning substance concepts

In Figure 6.1 there is a general impression about what to represent about a substance. As example we took the substance kind African dormouse[1].

Related knowledge may include definitions/descriptions, predicates, (typical) instances. In this section, we will analyse further which kind of representations are particularly useful for giving us abilities to reidentify. We will do this for:

- representing 'common sense' knowledge;
- representing definitions;
- representing inductive properties: prototypes, quality spaces, and *substance templates*;
- representing extension.

We will first focus on how these representations are *not* meant to function; and second how they may be used for a concept.

6.3.1 'Common sense' knowledge

Negative

How much 'common sense' world knowledge needs to be involved in our knowledge about concepts? Of how much use is it for matching or learning concepts?

World knowledge is knowledge *we* are able to gain by our senses: by eye, by ear, by smell, by taste, by touch. We know fruits are sweet, that you won't find telephones in the refrigerator, but will find milk, that ammonia has a characteristic pungent odor, that dogs bark, that paper has a smooth surface, that textile is soft, and so on. But computers are deaf, blind, taste-, smell-, and touch less.

[1]In this case, 'is-a' relations may be inferred from the name "African dormouse": dormouse, mouse, African thing. 'Is-a' relations may as well be represented by statement representations.

Do they, however, need to represent what can be seen, heard, smelled, tasted, felt from substances they want to learn about?

Of course they don't need to; most computers do not use this knowledge either. The only purpose of specifying such common sense world knowledge would be to match or learn concepts more easily, *when the information we get contains common sense.*

Positive

Of course our descriptions need to be helpful. If we are dealing with intelligent enough agents, our descriptions should be such that intelligent enough agents can understand the words we use.

Agents that use extensive 'common sense' world-knowledge may be helped a lot with 'common sense' knowledge of other agents, when communicating. For human-computer interaction 'common sense' definitely plays a great role; for computer-computer interaction, it may play a role, but is not necessary.

The best way to use world knowledge like visual and auditory information, is not the way of lexical representation. Other conceptual abilities for visual and auditory senses may very well be used rather as extensions of the lexical concepts that can be provided.

6.3.2 Definitions

Negative

We disagree that substance concepts all can be grounded by giving a definition, against the 'classical theory of concepts' (see Laurence and Margolis [1999]).

There are things that have a definition. Take for example a description of MAXIMUM DAY TEMPERATURE, which is built of MAXIMUM, DAY and TEMPERATURE. When we have a description that is a definition, coordination of same concepts reduces to coordination of the underlying concepts of the definition.

Describing substances But not everything has a definition. We saw in 3.1 that most substances are not classes: thus that properties are associated, and not definitional.

The associated properties of these substances may be ascribed by the substance template. Matching the substance template and the properties of the substance template is of great help. For example, a concept of a PERSON, MR. HIPPOCRATES, can be matched by the (functional) property *name* that is described by the substance template for PERSON.

Problems for definitions Still, whether we can describe associated properties of a substance (using a substance template or not), there are problems for taking these as *definitions*.

A problem for definitions are that they are too strict. For example, take an incomplete definition of DOG like "a *barking* CREATURE with 4 LEGs". Some dogs may have 3 legs. This could be solved by a typicality description, containing the typical values of its properties: *typical dogs* have 4 legs, but dogs with 3 legs still can be considered as dogs.

No definition But a more fundamental problem is that there may not be a definition for everything (or it is very hard to find one). A lot of descriptions in Semantic Web 'ontologies' will just be descriptions, not definitions. Take for example a description of *teach*: "a HAPPENING involving a TEACHER and a PUPIL, both AGENTs". It references to HAPPENING and AGENT, but these references do not *define* what teaching is. It is even a question if a concept like *teach can be defined.* The classical view (described as such by Laurence and Margolis [1999]) on conceptuology thought it could, but that view has faced serious criticism (see Laurence and Margolis [1999]).

Positive

If there is a definition for the extension of a concept, we will be able to represent for that concept in terms of others without losing meaning. The problem for identification can be delegated to the concepts in terms of which the definition is stated. Having more definitions is even better.

6.3.3 Inductive properties: prototypes, quality spaces, and *substance templates*

Negative

We disagree that substance concepts could be grounded by just summing up a number of predicates that apply to all or most instances. That is against the 'prototype theory of concepts' (see Laurence and Margolis [1999]) that there is a prototypical set of properties as basis for each substance concept. See Section 3.1 why: most substances are not classes, and do not have *common properties.*

Gärdenfors [2004] advocates the view that concepts are *regions in conceptual space*, where the *conceptual space* is made up of quality dimensions. For that view, the central thesis is that we distinguish concepts by their properties. But concepts are not like that. Our concept of MAMA is not founded on a set of properties: it is founded on our *purposes* instead. Our ability to reidentify fails if we say "Mama" to someone who looks just like her, and has the same qualities. (Also for Putnam's famous example: twin earth water is not water [Putnam, 1975].)

Positive

But knowledge, predicates that have been derived *by induction* can be relevant for a concept. Although agents may easily have different prototypes for the same concepts, it is quite useful to compare characteristic sets of properties.

Substance templates As mentioned before, *substance templates* describing the properties that support induction are highly relevant for the negotiation of substance concepts: by the substance template, we know which properties are relevant and which are not. Relevant, inductive properties are much more important to represent.

Representational statements may include 'is-a' relations to subject kind concepts. We can use *subject templates* to learn about concepts that fall under *a subject kind we have a concept for.* A subject may then be identified by its inductive properties (see Story 3).

Story 3 How to identify a duck by description.

"Let's see... You say it has white feathers, and you found it in the 'pond'. That must be a swan or a duck. Was it small or big? Small? Than it must have been a duck."

Agents could quickly agree that they share a concept with a same word, say "book", when they check the associated properties or relations (using words (*names* for them). A book has an author, for example. Humans would also quickly agree on the word "book", without producing complete descriptions about what a book is, if they just check the simple associated property that books can be discriminated by authors they have.

Learning the substance template Widely recognised substance templates could be shared by a single 'ontology' (like for PERSON, CHEMICAL ELEMENT, SPECIES, ...).

Millikan [2000, Chapter 5 and Chapter 6] is about learning of substance concepts by perceptual tracking and tracking through language, and the development of substance templates.

Westera [2008] develops a nice practical approach to discover such substance templates by text mining: substance templates may be learned from frequency of combinations of predicates with subjects.

6.3.4 Extension

Negative

Distinguishing a concept by its reference (extension) is undoable most of the time. For instance, it is not possible to list all individual DOGS. For a fictional character, e.g. SHERLOCK HOLMES, it is even impossible to establish a semantic mapping.

But even for subjects where it may be possible, it is not always *desired.* If for a long period of time all vehicles are mobots, that doesn't mean agents living in that time are better off to merge their ability to reidentify vehicles with their ability to reidentify mobots: there may come a time in which other things are vehicles as well.

Positive

But an identical extension may be an *indication* that two concepts are the concepts for the same. A extensional degree of sameness can be calculated if two agents operate on the same instances (semantic mappings have to prove they do!).

6.4 Learning predicate concepts

Descriptions that link names may include properties, exemplars, logical relations between properties.

Story 4 Q. and W. agreeing on sameness of concepts by mentioning instances.

W. and Q. were discussing the politics. Q. liked to use the expression "bread and plays". W. was repetitively bringing "football-matches" into attention. After some time W. confessed he didn't know what "plays" meant. "Nor do I know what "football-matches" mean", said Q. "But we are watching one!" said W. "That's a *play*", said Q. But after discussing several soccer events, Q. and W. agreed to treat the expressions "football-matches" and "plays" as used for politically equal subjects.

6.4.1 Properties

Properties can be like Kantian categories (like *mass*, *place*, *time*, but also *temperature*, *colour*). These property concepts should be in a shared 'ontology'.

Following from these basic properties are also some classes of *subject* concepts: concepts for place (like for "in Athens"), concepts for time (like for "last year"); concepts for mass, concepts for temperature, which are expressed in numbers and units, like for "13 kilogrammes", "15 degrees Celsius".

When the property is known, one can match the concepts for the property predicates. Steels [1997b] developed such an matching of terms for property predicates as well. Steels [1997b] uses agents which have artificial words for the different things they are able to identify by their sensors. With communication, the agents learn the words of the other agent and relate them to their own predicates for the same property.

Basic property predicates, like sticky, dry, smooth, could perhaps be in a shared conceptuology as well. The only other way to learn these predicates is to negotiate about to which instances it does and which instances it does not apply.

6.4.2 Descriptions

Predicates may be complex. We mentioned happening predicates (Section 5.4) and relation predicates (Section 2.1.5).

Predicates that are based on happenings are easy to describe. For example, A BOOK may be *written in* a LANGUAGE. We here use the concept *to write*.

As predicates are qualifications, predicates may also be *defined* in terms of other predicates. But some predicates, one-place (like poor, eager) as well as more-place (friend of, parent of), are not that easy to define.

6.4.3 Derivations between properties

There are often inductive (or even deductive) derivations from properties to other properties. For example, (A DOCTOR) *has practice at* → *lives at*: a logical implication.

That different properties are related may also be learnt from the extension they cover, or from particular obvious cases in this extension (exemplars). That is where inductional or abductional reasoning may play a significant role.

6.4.4 Other, non-lexical mechanisms

Besides lexical representation, there are a lot of other possibilities for concepts. Other ways may be used to come to a better concept; our lexical approach can very well be *extended* with such mechanisms. Because our focus is on communication, we will only mention some references.

The perspective of Gärdenfors (Gärdenfors [2004]) on properties as *quality dimensions* is quite useful for negotiation of property predicates. In that perspective there is, e.g., a (continuous) colour space in which colours are regions. Humans try to capture the different colours as effectively as possible; the partitioning of the space for our purposes can be thought of as a Voronoi tessellation, in which the centres of the regions may be perceived prototypical for a colour.

That is useful for *inner representation* of, e.g., colours. For *outer* representation (used for communication) we use the mechanism of language; language provided us with *shared* regions for colours for our purposes, we believe, although there might be physical reasons as well.

6.5 Identity between concepts

Having learned how concepts may be matched using representational information, how do we handle an established identity between concepts?

There are more 'senses' for a 'reference' possible. The classical example (Frege [1892]) is that both "Abendstern" and "Morgenstern" can refer to Venus.

To a classification, saying that classes originate from senses that have an identical reference is just relating the symbols: one can simply state the classes cover the same (`owl:equivalentClass`). For the definition of the two classes it is important the properties do not conflict, however.

For concepts, if us is being told that "The evening star is the morning star", different substance concepts THE EVENING STAR and THE MORNING STAR need to be *merged* into one (see Millikan [2000, Chapter 12]). It may be our concepts for the substance are mistaken: we could have thought that "it is only visible in the evening", which is an error of our substance concept[2]: the same substance can be tracked in the morning as well.

It is of course not the case that two concepts and related representations remain for two 'modes of presentation' of Venus (see Millikan [2000, Chapter 11]). Tracking Venus, the evening star, and the morning star has become one task, one ability. (If one *would like* to have an ability to reidentify *the appearance of Venus in the evening* with "the evening star" and *the appearance of Venus in the morning* with "the morning star", these are not *substance* concepts.)

One should distinguish these cases of *identical concepts* from cases of contingent identity between classified instances. In expressions like "the president of America" we use a classification and quantification by the word "the". "Bush is the president of America" will only merge concepts if we have a substance

[2]For the concepts HESPERUS and PHOSPHORUS, delivered by Greek mythology, it is even more complex. We could have thought "it is the son of Astraeos" (which the myth told from Hesperus), but that plainly conflicts with "it is the son of Cephalus" (which the myth told from Phosphorus). If we still want to believe that the substance is a son of Eos, we only can keep one of both. We could also say our concept of both sons of Eos were *confused* with the heavenly body (Venus), and therefore rigidly revise HESPERUS and PHOSPHORUS, such that both are not identified with the heavenly body anymore. . .

individual concept THE PRESIDENT OF AMERICA as well. The concepts A PRESIDENT OF AMERICA and BUSH can perfectly be kept separate. Another such example is "all vehicles are mobots" (which may hold in the future): we can perfectly have different concepts for VEHICLES and MOBOTS, because they are different substance kinds, and the statement "all vehicles are mobots" only applies to the current classification of vehicles.

One should also distinguish these cases from cases of *necessary* identity between representations, as in "a bachelor is an unmarried man" and "one's uncle is the brother of one's parent" and "the author of a book is who has written it", which give definitional statements (instead of just 'is-a' relations).

6.6 Conclusion

In this chapter, we answered the question 'Is it possible to have concepts through representational statements alone?'

Artificial agents can have concepts through language representations alone. Language-like representations, based on lexical concepts, plus reasoning, will be able to solve the interoperability problem to a large extent.

Chapter 7

The fallibility of concepts

Having set out how concepts can be grounded in representations, the next question is: how fallible (see Millikan [2000]) are these concepts? In this chapter we will pass on particular cases of fallibility that may occur for concepts grounded in representations.

7.1 Ambiguity

Ambiguity may arise when a name has different meanings. The classical example is the word "bank", of which the meaning is ambiguous between money bank and river bank.

How can we avoid ambiguity? When humans interact, they solve ambiguities by using the context and perhaps by trial-and-error. Ambiguity problems are to be handled by context and trials. However, to avoid reoccurring ambiguity problems, two communicating agents could choose to use different words instead: "bank_1" and "bank_2". (In Section 10.1, we present how this can be implemented.)

7.2 Different words

There could be different names that accidentally refer to the same subject. An example is different languages: "horse" in English is "paard" in Dutch.[1] Another example is "morning star" and "evening star". In these cases, matching should merge the concepts (as described in Section 6.5).

7.3 Predicate-subject composite intensions

Combining concepts (e.g., applying a predicate to a subject) is a difficult ability. Kamp and Partee [1995] mention some good examples: "stone lion", "sharp

[1]For our research, we have used one natural language (English) for implementing conceptuologies. Problems that may arise when different natural language words are used for different Semantic Web agents, are comparable to problems that arise when humans from different languages have to communicate: they are quite hard to solve, the best solution may be to use an interpreter that knows both languages well.

knife", "striped apple", "midget giant" or "giant midget" (see also Section 5.5). In these cases, our 'common sense' takes inductive steps to come to a composite intension: we know that a striped apple will not be typically striped like a zebra, that a stone lion is not a growling stone animal but a statue. It is obvious that intelligent agents that lack 'common sense' knowledge, will have more artificial, less 'natural' interpretations. Notice that "not natural" exactly means that one lacks world-knowledge in some aspects. Such confusions may in some cases be avoided by using new names for different concepts, instead of combining old ones; that is, avoiding the use of combinations as "striped apple", "giant midget" by using a different name to refer to "striped apples" and "giant midgets".

7.4 Fuzzy reference

Another difficulty about representing for concepts is that we sometimes use the same name to cover more references or senses.

Take for example book kinds. We may say "This book is written by Lewis Carroll" and "This book is on my desk". The term "this book" is first used to refer to the story, then to the real book instance. Other examples are "This dormouse is extinct" and "The dormouse is of the family *Gliridae*", where the word "species" or "genus" is not used at all (leaving some work for the hearer).

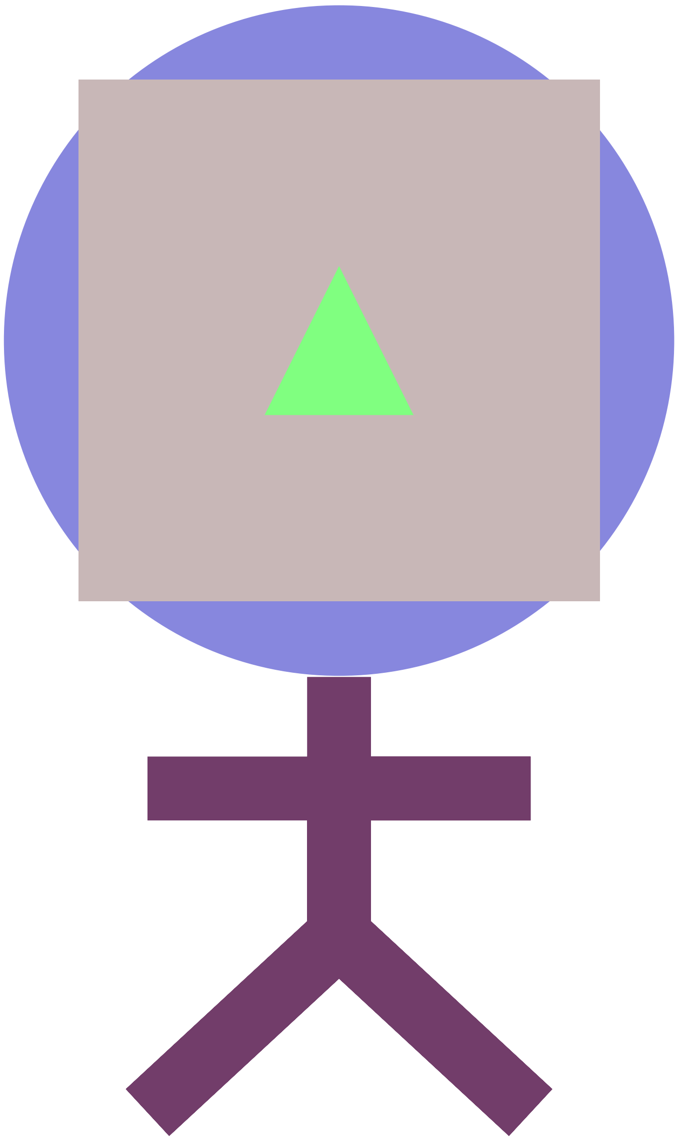
Typical cases in which reference errors may occur are part/whole (totum pro parte, pars pro toto) and type/token (abstractum pro concreto or concretum pro abstracto) distinctions. Designers of expert systems should be aware of these mistakes.

7.5 Conclusion

In this chapter, we answered the question 'How fallible are concepts?'.

We analysed at which places a conceptuology based on representations in particular may fail. Concepts based on representations are fallible, but these problems are not comparable with the interoperability problem that arises with different 'ontologies'.

Practical implementation has to minimize the mentioned cases of fallibility. In Part III, we will analyse how artificial agents may cope with it.



Conceptual agents

Chapter 8

Agents using concepts

Now we have answered the question how a conceptuology can be based on lexical representations – on which artificial intelligent agents can rely –, we continue with placing the agent with its conceptuology in the broader perspective of agent communication: connecting different conceptual agents.

8.1 Natural language

In the introduction, we already argued that agents should use a natural language lexicon for shared words. A natural language is full of shared words for concepts.

Creating an agent conceptuology based on natural language has another advantage: the agent shares the words with *us*, humans, as well. That may prove helpful in human-agent interaction.

With natural language, we also import the fallibility of it, as already mentioned in the previous chapter: ambiguity, and different words for the same. But that is not a mayor drawback; a conceptuology should be able to cope with it anyway.

8.2 Experts and laymen

Organisation The Semantic Web consists of many services. Most services will not have a lot of concepts, but will have a restricted conceptuology. For example, thermometers, radiators, and coolers will not have many concepts. They cannot understand complex representations.

However, what if these services are to be used in an organisation of distributed software systems? This project is contained in the ALIVE project (`http://www.ist-alive.eu`). The ALIVE project aims at development of a new approach in establishing *organisations* between distributed software systems. The keyword in the ALIVE approach is *adaptation*: the structure of an organisation may change, while still obeying its norms; or the actual implementation, by agents that fulfill roles in the organisation, may change.

It is of course in such contexts, where new agents or services may become part of an organisation, that the interoperability problem exactly occurs. Agents or services have to rely on their concepts when they encounter new agents or available services and have to share content or communicate.

Know-how and knowing-of Solving the interoperability problem between the 'laymen', the services, can be done by delegating the interoperability task to intelligent agents, 'experts'. Agents that *do* understand complex representations, domain experts, should be there to connect these services. These agents should have a larger conceptuology, bound by the domain they have to operate for. They should have knowledge-how to carry out tasks, but also knowledge-of to communicate with other services around in the Semantic Web.

We will analyse the methodology with an example. Suppose the task is to keep the temperature of a building at a comfortable temperature. The task is handed to a domain agent that knows-of *temperature*, BUILDINGS, and *comfortable*. It knows-that comfortable means between 18 and 20 degrees Celsius.

The agent should know-how to keep a temperature of a building some temperature. For example, it knows it can achieve such a thing by following these *work flow* steps:

- measure the temperature of the building each hour;
- warm the building if the temperature is too low, cool the building if the temperature is too high.

The agent does not have the ability to measure the temperature, but should find services that have the know-how to perform these tasks. It will contact a thermometer service, a radiator service, a cooling service, and match its concepts with their concepts.

The thermometer service knows-of DEGREES CELSIUS and *temperature*. The radiator service knows-of *warming*, the cooling service knows-of *cooling*. The domain expert agent should have all these concepts.

Learning In practice, different types of agents can be used in the Semantic Web. The current scenario is often that an agent doesn't learn new concepts at all. When agents however have capabilities to handle lexical representations in a universal way, different learning perspectives open.

- An agent can learn eager from, e.g., OpenMind, or other networks that store a lot of common world knowledge.
- An agent can learn lazy or eager from the agents it meets.

8.3 Division of linguistic labour

A view advocated by Putnam is that people using the words should refer to experts for the meaning, the "division of linguistic labor". We want to argue against that view that everyone who uses the word needs a concept (ability to reidentify) for that word himself, and that references to experts are only secondary. That is defended by Millikan [2007]:

> We know enough about weasels to know, perhaps, that the experts on weasels are biologists, perhaps, more exactly, field biologists, indeed, quite exactly, the ones who know a lot about weasels. But how are we to tell which ones know a lot about weasels rather than, say, mink or stoats if we do not ourselves know what weasels

> are? In practice, I suppose, we ask, and they tell us it's 'weasels' they know about. They are self-proclaimed keepers of the correct rules for the English term 'weasel'. But why should we trust them? Indeed, what are we trusting them about? [Millikan, 2007, p.3]

We have to possess the concept ourselves, which is already an ability which is "in no way secondary" [Millikan, 2000, p.90]. Knowing services that know a lot related to the same concept may help us a lot, however, because they have a richer, and perhaps less fallible concept.

8.4 The picture of communication

8.4.1 "Meaning, meaning and meaning"

Sending representations to another agent involves communication.

Communication involves three layers of meaning, describes Millikan [2005, Chapter 3: 'Meaning, meaning and meaning']:

- stabilizing functions, "conventional linguistic cooperative functions", i.e. the message type and structure;
- truth- and other kinds of satisfaction conditions, "semantic mapping functions", i.e. the mappings from the content parts of the message;
- private conceptions, "methods of identifying that govern individual speakers' grasps of referents and of truth- or satisfaction-conditions".

Story 5 Failing of the stabilizing functions of shouting to secretaries.

Police officer P. leaned back and shouted: "A report on the boy who was here!" After ten minutes his secretary appeared: "There is no such report, sir." "No, but I meant there should be one!" P. grunted.

Stabilizing functions In Story 5 the stabilizing function of shouting about reports to secretaries fails. Instead, a satisfaction condition of the message failed for the secretary (there was no such report).

Agent Communication Languages (ACL's) are well designed to overcome these problems with stabilizing functions. Today's ACL's, e.g., KQML, FIPA-ACL, have a list of *performatives* (see Searle [1989]), such as "ask-if", "tell", "achieve", that exactly tell what the sending agent means with, and the receiving agent should do with the representation that was sent.

Semantic mapping functions Content parts of a message are (often) mapping to a semantic model. Happening statements map to intervals in time. Substances map to entities in space. Agents are to be provided with logical models to reason using these semantic mapping functions.

Private conceptions But before any of the functions of a representation can be used, a representation has to be identified.

8.4.2 The "language-thought partnership"

Millikan [2005, Chapter 5] analyses the "language-thought partnership". Communication does not stand on its own. Communication has *purposes*. See for example Story 6.

The work of Grice (Grice [1975]) is well-known for addressing issues about the purposes of communication.

Story 6 How language-thought partnership may fail.

W: "My coffee cup is empty." (Pause.) W: "My coffee cup is empty." R: "You already said that, didn't you?" W: "Couldn't you bother to fill it?" R: "Yes, I could." W: "*Do!* Please."

We will not further expand on the "language-thought" partnership, as it is well beyond the scope of this paper; but we mention it to place the topic of it in a broader perspective.

8.5 Conclusion

In this chapter, we answered the question 'How can agents use conceptuologies?', touching on the use of the conceptuology in a broader perspective of agent communication.

Natural language can provide a rich basis for sharing words mutually between agents and between agents and humans.

We showed how different agents with different knowledge may communicate with private conceptuologies. Each agent can manage it on its own: there is no need for division of linguistic labour (where only experts 'own' the concepts), private concepts and conceptions are welcome. What agents share is the language-like way of representing using words.

In the following chapter, we will present the implementation of CROC: a Representational Ontology for Concepts. We will investigate the different agent components that are touched, and show how a conceptuology can be implemented.

Part III

The CROC system

Chapter 9

Conceptuology implementation

For the design goals stated in Section 6.2 web 'ontology' languages are inadequate. The aim to use representations likewise in the state-of-art Web Ontology Language (OWL) encounters the following problems:

1. OWL does not use names. (It uses URI's.) Although classes could have names, that would not give us substance concepts but class concepts — and substances mostly are not classes.

2. OWL has a (class) hierarchy, describes (inverse) functional properties[1] — but does not really care about induction, because *class definitions are deductive.*

3. OWL has a universal subject-predicate (or rather subject-property) structure. It does not have actions as a special representational item. (A happening *is not a thing*, it is no *class*. Nor is it a predicate or property (it may be *used in a predicate*, however).)

4. OWL has *class restrictions.* (Minor: OWL does not support one-place predicates — it has boolean datatype properties instead. See Section C.2 for illustration.) Reasoner statements (these may include induction!) are *apart from the 'ontology'.*

Therefore, our representations for concepts will be placed in a new framework: CROC, a Representational Ontology for Concepts.[2] CROC will consist of these main components:

[1] Properties of substance concepts might be specified as class properties. It however is not specified which properties support induction and which do not, which makes it hard to learn new subclasses.

Some about the inductive value of a property may be derived from the specified property *type.* In OWL a *functional* property can be defined, meaning that the property is unique, i.e. for any x, there cannot be two distinct values y and y' such that the pairs (x, y) and (x, y') are both instances of this property. *Inverse* functional properties are the other way around, i.e. for any y, there cannot be two distinct values x and x' such that the pairs (x, y) and (x', y) are both instances of the property. (Inverse) functional properties are therefore likely to support induction in the real world well.

[2] In Appendix C we have made a comparison of different representations about the same subject in OWL and CROC, which is largely meant to be illustrative for OWL users.

1. An ontology of concepts, specified in OWL. We focus on practice by only making the most relevant distinctions between concept kinds, as sketched in Chapter 2.

2. CROC uses XML for transferring the representations. In Appendix B the XML Schema used for representation is sketched, which is directly based on our discussion of representations in Section 5.

3. A reasoner to process representations.

9.1 Agent components

From van Diggelen [2007], we copied a nice way of depicting agent components and their dependencies. See Figure 9.1: "Two components are placed next to each other to illustrate that the component on the left is constituted by the component(s) on the right. Two components are placed on top of each other to illustrate that the component on top cannot exist independently, but is in some sense based on its underlying component."[3]

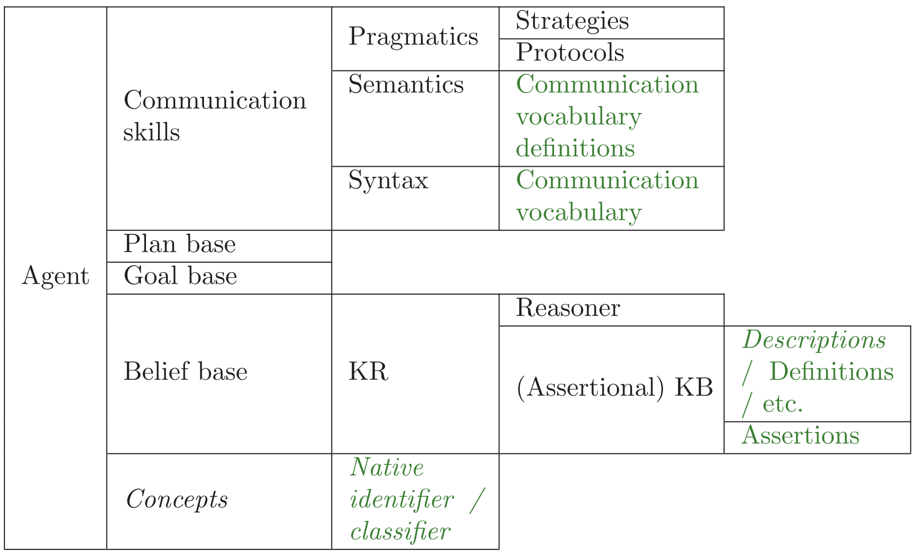


Figure 9.1: Agent components. Derived from van Diggelen [2007, Figure 4.1, p.62].

Notice the 'Native identifier and classifier' we added as a fundamental agent component, which constitutes our concepts: abilities to reidentify.

All beliefs are representations that are in some sense based on concepts. In our case the concepts (abilities to identify) and abilities to classify have to come from the representational ontology. This cannot be depicted in the same manner because it is a recursive dependence — see Figure 9.2.

[3]Unlike van Diggelen we do not have a distinction between symbolic and non-symbolic communication, because class explication like in van Diggelen [2007] is not our main interest for the system.

In the representational ontology for concepts the lexicon that is used for communication (communication vocabulary) is stored, attached to the entities (communication vocabulary definition).

Agent	*Concepts*	*Native identifier / classifier*
	Representational ontology for concepts	*Descriptions* / Definitions / etc. from KB

Figure 9.2: The dependency on the representational ontology for concepts.

The interdependence between represented knowledge and concepts, and the close connection between the communication vocabulary and concepts are the interesting properties of CROC. They show how CROC takes language and language representation as a basis for learning and matching concepts.

In the remainder of this chapter, we will describe how each of these components is actually implemented.

9.2 Knowledge representation

9.2.1 Statements

Reasoning and normal form

Our knowledge base is a 'set of beliefs' $(F_1 \wedge F_2 \wedge F_3 \wedge \ldots)$. It would be possible to convert the set of beliefs to conjunctive normal form, clauses that can be used for, e.g., resolution. The disadvantage is that the computation of a CNF has some costs, and it can lead to an exponential explosion of the set of beliefs. That is, formulas like $(X \wedge Y) \vee (X_1 \wedge Y_1)$ would rewrite to $(X \vee X_1) \wedge (X \vee Y_1) \wedge (Y \vee X_1) \wedge (Y \vee Y_1)$.

We can however use a variant of CNF to rewrite only to expressions of shorter or equal length. We may use negation normal form partly as well.

Our motivation to not convert formulas to normal form directly is because more difficult formulas will less often occur, and will be used less (and when they are used it may take some extra time). We therefore simply use strongly analytic semantical tableaux (SAKE) for computing the proofs.

Finding matches for statements

Strongly analytic semantical tableaux 'Semantical tableaux' are reasoning trees that for each node keep track of a set of positive expressions P, and negative expressions N, usually notated as $P \circ N$. When an item of both sets can be unified, the tableau node is 'closed': a contradiction is detected. A *tableau* is called 'closed' if all its branches lead to closure. The tree splits for distinguishing cases; e.g., a positive expression $p \vee q$ may cause a branch with the positive expression p and another branch with the positive expression q.

With a semantical tableau we can prove *satisfiability* of p if a tableau starting with p added to the right (the set of negative expressions) of the root node, closes. *Inconsistency* of p can be proved if a tableau starting with p added to

```
1: Apply $\beta$- and $\gamma$-reduction exhaustively.
2: for all not closed nodes do
3:    for all atom statements right and left do
4:       Search for (superficially) matching statements (respectively positive or
         negative), and add them to the left (lazy or exhaustively).
5:       Repeat from the beginning.
6:    end for
7: end for
8: Compose the answer from the results at the tableau tree leafs.
```

Figure 9.3: Iterative tableau closure.

the left (the set of positive expressions) of the root node, closes. Both mechanisms are useful for, e.g., answering a question: satisfiability accounts for 'yes', inconsistency accounts for 'no'.

For developing the tableau to prove satisfiability or consistency of a query statement, we apply the following algorithm (see Figure 9.3). Iteratively (superficially) matching statements in the knowledge base, from the subject concepts that are involved in the to be matched statement, are provided to close the tableau.

Marking statements from the knowledge base When adding new statements from the knowledge base, the reasoner does not have to search for matches for these new statements: we assume the knowledge base is consistent. Therefore we mark statements that were directly added from the knowledge base.

When complex statements from the knowledge base are β-reduced, the mark is erased. For example, when $p \vee q$ is added from the knowledge base, we do have to match $\neg p$ or $\neg q$ in the different branches of the tableau for purposes of closing it.

Related knowledge A step of the algorithm in Figure 9.3 involves the search of matching statements. The search looks up the related knowledge of the subjects that are involved in the query statement.

Related knowledge of the concept entities in the CROC ontology (see Appendix A) stores *statements*, using the predicate `croc:relatedKnowledge`.

To derive all predicates applicable to a subject, we have to find all happening complexes that involve this subject. All subjects involved in a happening should have pointers to that happening. For example, if the statement in our knowledge base is "John loves Sara", this gives us both the predicate "loves Sara" for John and the predicate "is loved by John" for Sara. Moreover, we should be aware of 'is-a' relations; that is, that more reidentifications apply to the same. For example, the predicate "Humans are mortal" is relevant for the human Socrates. Collecting all knowledge about something is resolving all pointers to the statements in which it is involved, and recursively so following the 'is-a' relations that apply.

Matching statements For closing the tableau, only statements in the related knowledge that are *superficially matching* are added (see Figure 9.3).

Symbol	Meaning	Examples
i	subject intension	“parrot”, “yellow car”, “oak or birch”, “philosopher and mathematician”
e	subject extension	“that car”, “all oaks”, “three parrots”, “John”
s	subject set	“John and Mary together”, “this car”, “only one parrot”, “yellow cars” (reference to kind)

Figure 9.4: The basic symbols used.

We count a complex statement c as superficially matching a positive or negative atom statement q when c contains a negative, respectively positive leaf statement p matching q. We count an atom statement p as matching q if the happening in p is the happening in q; and p has as much, or more involved subjects for the same roles as in q.

9.2.2 Reasoning

In this section, we will list how happening representations are to be treated in reasoning. We note that in some cases, the reasoning is not implemented in the first version of the CROC reasoner; but we will try to sketch how it could be implemented.

Happenings

Happening representation complexes describe what is and is not the case, in a world w (may be not the real world).

We first want to have a relation $\vdash_{happening}$, where $p \vdash_{happening} q$ means: in all worlds where p is true, q is true.

$$p \vdash_{\text{happening}} q \text{ if and only if in all worlds where } p \text{ is true, } q \text{ is true} \tag{9.1}$$

For logical connectives over happenings, like “and”, “or”, “not”, “if . . . then . . . ”, “ȯr” (exclusive or), we can just use the $\vdash$ of proposition logic. (We implemented reasoning by using semantic tableaux.)

Furthermore, we introduce the notation $p^{\{r_1 \to s_1, \ldots, r_n \to s_n\}}$ for a happening p with involved subjects $s_1, \ldots, s_n$ for happening roles $r_1, \ldots, r_n$. Less generic happenings make generic happenings true:

$$p^{R \cap \{r \to s\}} \vdash_{\text{happening}} p^R \tag{9.2}$$

For example, in worlds where “John fights Bill” is true, “John fights” and “Bill is fight” are also true. (We implemented this form of reasoning at the leaves of the tableau; the satisfaction relation between two happenings is computed by checking if all the involved subjects in the right hand side satisfy the involved subjects in the left hand side.)

Subject extension

The specification of a subject extension consists of description of the set of subjects that are together included in the role. (Quantified or determined subject intensions, or subject individuals.) The description may or may not be complete; it is given by specifying which subject extensions are identifiably included and which subjects are identifiably excluded. We take here the notation $\mathcal{P}$ for the inclusion set and $\mathcal{N}$ for the exclusion set, which are expected to be mutually exclusive.

Subject extension connectives are used for that. For example, consider the use of "together" and "alone" in "John and Bill together are / John alone is able to lift this stone" or "John and Bill together ($j, b \in \mathcal{P}$) / John alone ($j \in \mathcal{P}, \_ \in \mathcal{N}$) lifted this stone", the use of "without" in "Mary left the house without John" ($m \in \mathcal{P}, j \in \mathcal{N}$), the use of "other than" in "people other than John ($\_ \in \mathcal{P}, j \in \mathcal{N}$) thought that".

Reasoning between different subject extensions can be implemented by the following rule:

$$p^{\{\ldots,r\to s,\ldots\}} \vdash_{\text{happening}} p^{\{\ldots,r\to s',\ldots\}} \text{ if } s' \lhd s \tag{9.3}$$

We write $\langle \mathcal{P}_1, \mathcal{N}_1 \rangle \lhd \langle \mathcal{P}_2, \mathcal{N}_2 \rangle$ if for all $e_2 \in \mathcal{P}_2$, there is an $e_1 \in \mathcal{P}_1$ for which $e_2 \supseteq e_1$ (all included subjects have been mentioned more precise or as precise) and there is no $e_3 \in \mathcal{N}_1$ for which $e_3 \supseteq e_2$ (if $e_2 \equiv e_1$ we do not have to check this last condition because $\mathcal{P}_1$ and $\mathcal{N}_1$ are expected to be mutually exclusive); and for all $e_2 \in \mathcal{N}_2$, there is an $e_1 \in \mathcal{N}_1$ for which $e_2 \subseteq e_1$ (all excluded subjects have been mentioned) and there is no $e_3 \in \mathcal{P}_1$ for which $e_3 \subseteq e_2$ (if $e_2 \equiv e_1$ we do not have to check this last condition).

In the current implementation of CROC, we have not treated subject extension sets; it is assumed instead that the subject extension is a singleton.

In the following, we will write simply $p^{\{\ldots,r\to x,\ldots\}}$ to denote $x \in s : p^{\{\ldots,r\to s,\ldots\}}$.

'Is-a' relations

For 'is-a' relations, we need four reasoning rules. First, between subject extensions:

$$e \text{ is } e' \wedge p^{\{\ldots,r\to e,\ldots\}} \vdash_{\text{happening}} p^{\{\ldots,r\to e',\ldots\}} \tag{9.4}$$

For example, in worlds where "I saw Socrates" and "Socrates is a human" are true, "I saw a human" is true too.

Second, between subject intension constants:

$$i \text{ is } i' \wedge p^{\{\ldots,r\to\text{this } i,\ldots\}} \vdash_{\text{happening}} p^{\{\ldots,r\to\text{this } i',\ldots\}} \tag{9.5}$$

For example, in worlds where "I saw a frog" and "frogs are amphibians" are true, "I saw an amphibian" is true too.

Third, *inductively* between a subject intension (reference to kind) and a constant of that subject intension:

$$e \text{ is a } i \wedge p^{\{\ldots,r\to i,\ldots\}} \vdash_{\text{happening}} p^{\{\ldots,r\to e,\ldots\}} \tag{9.6}$$

For example, in worlds where "humans are mortal" and "Socrates is a human" is true, also "Socrates is mortal" is true, inductively. In contrary to universal quantification (e.g., "*all* humans are mortal"), which is handled deductively (e.g., which simply substitutes "Socrates is mortal"), reference to kind is handled inductively.[4]

[4]For implementation of induction, we have come no further than keeping track of the number of induction steps used for each statement added to the tableau.

Fourth, between subject intensions:

$$i' \text{ is } i \wedge p^{\{\ldots,r\to i,\ldots\}} \vdash_{\text{happening}} p^{\{\ldots,r\to i',\ldots\}} \tag{9.7}$$

For example, in worlds where "I like amphibians" and "frogs are amphibians" are true, "I like frogs" is true too.

Subject intension

A subject intension description is a delineation in the space of subject intensions.

Reasoning between different subject intensions can be implemented by the following rule:

$$p^{\{\ldots,r\to i,\ldots\}} \vdash_{\text{happening}} p^{\{\ldots,r\to i',\ldots\}} \text{ if } i' \subseteq i \tag{9.8}$$

For example, in worlds where "professors are always late" is true, "(professor and doctor)s are always late" is true too. The group of "(professor and doctor)s" is the intersection of the group of "professors" and the group of "doctors". The group of "(professor or doctor)s" is the union of them.

All logical connectives over subject intension ("and", "or", "not", "if ... then ...", "òr") can be resolved using straightforward semantical tableaux again for the satisfaction relation. Notice that a tableau with "(non-student or student)s" on the left hand side will close for every right hand side.

The problem is that this subreasoning about involved subjects can cause the tableau to split again. For example, when we have "students like books" and not "(student or professor)s like (book or paper)s" we leave with not "professors like (book or paper)s" or not "students like papers". In the first version of CROC, we therefore simplified the system such that no connectives between predicates are used.

Quantification

Quantification is largely a deductive reasoning mechanism.

For quantification, order is important. "Every man loves a woman" can be read as "there is a woman that every man loves" or as "for every man there is a woman which that man loves". "Every father of a son" can be read as "this son, thereof every father" (of which there is only one, of course...) or as "every father who is a father of a son". Therefore, representations containing quantifiers are annotated with an *ordered* list of quantifiers that have to be resolved before processing the representation further.

The substitution of an quantified expression by a new expression is called gamma-substitution. Entities to substitute with are gathered (1) from the discourse, and (2) from the knowledge base itself. For example: when substituting "every vehicle",

- entities from the discourse will be candidates: e.g., "this vehicle", "this car";
- CROC will lookup in the knowledge base all substances that are men: e.g. "Humphrey", "every car".

Predicates

Separable predicates The case of predicates is analogous to subject intension descriptions. Predicates are applied to subject intension descriptions and used generally to classify and narrow their extension. The predicates in the quantified expressions "every green car" or "a green car" narrow the set of possible substitutions when reasoning about these expressions.

Connectives between predicates over subject intensions are straightforward, e.g.:

$$(P \wedge Q)@s \subseteq P@s \text{ and } (P \wedge Q)@s \subseteq Q@s$$

All logical connectives between predicates over subject intensions ("and", "or", "not", "if ... then ...", "òr") can be resolved using straightforward semantical tableaux again for the satisfaction relation.

The problem is that this subreasoning about involved predicates can cause the tableau to split again. In the first version of CROC, we have simplified the system such that no connectives between predicates are used.

Non-separable predicates Some predicates are not used just as separate condition, but mean to modify the subject intension. For example, "a fake car" is not simply "a car" which "is fake". This class of predicates are not simply predicates that are applied to a subject; they are predicates that cannot be separated from the subject intension they are applied to.

It is clear a different classification mechanism has to be used here, one that modifies some properties of the subject intension to which the predicate is applied, and then reidentifies which items fall under that modified intension. Predicates like "fake", "not existing", "semi-" directly have a classification purpose. These mechanisms have not been implemented in CROC yet. (It is not expected this class of predicates immediately will play a great role for a conceptuology based on lexical representations.)

9.3 Identification

9.3.1 Concepts

In Chapter 2, we made a classification of the different abilities to reidentify (concepts) that are to be included in a conceptuology. This concept ontology for CROC is specified in the ontology language OWL. We refer to Appendix A for a more detailed description of the CROC ontology for concepts, but will give some main ideas here.

For example, instances of substances will be represented by an entity of type `croc:Substance`. The distinction between subject individuals, kinds and stuffs can be represented by using the `owl:subClassOf` relation to the classes `croc:Kind`, `croc:Individual`, and `croc:Stuff`.

For each of the concepts, there is a abstract subject concept which may be used for reification. That is, there are native concepts for A SUBJECT, A SUBSTANCE, A HAPPENING, A PROPERTY. These concepts are useful to talk about subjects in general (e.g., referring to "something").

Using the annotations of OWL, we can attach names to subject descriptions with `rdfs:label`. These identifiers will be used to identify subjects by name.[5]

Using `rdf:ID`, a unique URI can be attached to a subject. (URI's are *unique* identifiers, where names still can be ambiguous; see also Section 10.1.)

9.3.2 Using classes with concepts

For knowledge representation *efficiency* purposes, it may be interesting to combine existing knowledge representation resources based on classes, such as relational databases or RDF/RDFS/OWL, with the communication and identification mechanisms provided by CROC. The CROC system does not yet contain an integration, but we will describe how it can be realized.

When relational database tables can be interpreted as sets of RDF or OWL individuals, a plug-in may be provided to use these data as if they were subject kind instances.[6] Therefore one has to start by creating a regular subject kind concept — or choose to reuse one. To the corresponding subject kind one can describe common and 'is-a' relations, treating them as merely inductive.

From the subject kind concept then a link can be created to the instances from the RDF or OWL data by reference to an RDFS class, and a converter plug-in from a data entity to a concept description. Such a converter plug-in needs a mapping that links properties to happening representation statements. For example, a property named "hasAuthor" in OWL may link to the corresponding happening representation "x has author v" in the mapping. However, one can only map a property value if the data type of it is syntactically interoperable as a CROC representation. Not syntactically interoperable property values need hand written conversion methods.

9.4 Communication

For transferring the language-like representations, as described in Chapter 5, we use XML. The XML schema (specifying which elements the XML may contain) is described in Appendix B.

Happening statements represented in XML are the content part of the messages that are sent. The purpose of the statements (e.g., descriptive, directive, or interrogative purposes) are specified by using the performatives of the agent communication language KQML (already mentioned in Section 8.4.1). For practical examples of communication, we refer to the next chapter.

9.5 Conclusion

In this chapter, we answered the question 'How can such a conceptuology (as sketched in Part II) actually be implemented?'

[5]Where multiple names may be used for the same subject (see Section 7.2), a preference for use (per communication-partner, which may have different preferences) is desirable for effective communication. We have not yet implemented this.

[6]To carry over the concepts of RDF or OWL to CROC instead, i.e. creating concepts for "class" and for "instance of" and for "property restriction" and so on, is of course not what we aim at.

We have shown the possibilities of CROC: a Representational Ontology for Concepts are extensive, and set up a framework that uses representations for developing concepts for artificial agents.

The system CROC is suitable for extension in many directions. Reasoning mechanisms may be extended, classifications may be used for effective knowledge representation (after it has been identified by concepts), other conceptual abilities may be combined with the concepts based on representations.

CROC has a lot of possibilities, but not all of it is implemented yet. We therefore list some future work what we consider important for its functioning:

- the implementation of a temporal logic for reasoning with happenings, as sketched in Section 5.3;
- the support of inductive and abductive reasoning mechanisms;
- integrating classification systems for efficient knowledge representation, as sketched in Section 9.3.2;
- higher order reasoning: reasoning about what other agents think, believe, etc., based on the context of earlier meetings with those agents.

The CROC project is open source and available on Sourceforge: `http://sourceforge.net/projects/croc/`.

In the next chapter, we will finally show some examples from practice.

Chapter 10

Practice

In this chapter we will explain the *use* of the conceptuology as described for implementation in the previous chapter, including some practical examples.

10.1 Ambiguity and the use of URI's

In Section 7.1 we described ambiguity as a case of fallibility of a conceptuology.

Ambiguity problems are to be handled by context and trials (see Section 7.1). However, to avoid reoccurring ambiguity problems, two communicating agents may establish a URI (a Uniform Resource Identifier, like `http://www.croc.name/agent1#MyConcept`) for sharing a concept, a mechanism well known in Semantic Web because it is used in 'ontologies'.

Like a name, a URI is an identifier as well. However, URI's are not shared like words in a natural language. For our purposes, URI's may be shared if two agents have *one single denotation* of a resource. That means there must be an authority or other agreement that ensures this single denotation. The CROC system uses URI's for some purposes, ensuring same interpretation for CROC agents itself.

Ambiguity is not only present in words for subjects, it is also present in relations. For example, the meaning of "Richard's book" is ambiguous between the meaning of "book that Richard owns" and of "book that is written by Richard". Here the word "of" in "book of Richard" causes the ambiguity. Because relation words like "of" are in general most ambiguous, we use URI references for relations instead of words. An alternative approach would be to derive the most applicable interpretation from the context.[1]

10.2 Communication handling processes

The CROC communication process works as follows. For each message, a message handling process is started: see Figure 10.1. This process may call different subprocesses, which will call back if ready.

[1] We can apply two different substance templates here, which have different inductive properties: the properties *has author* and *has owner*, which both have PERSONS as a range. We choose the most applicable template by the context.

```
1: Receive of some message (i.e., a directive, interrogative, or descriptive representation).
2: Internalisation of the message: named items are reidentified, and replaced by internal entities. )
3: **if** unknown names **then**
4:     Create a new concept which has that name as an identifier. Gather some from the representation: e.g., the subject type, happening roles, and a predicate's property.
5:     Start an *explication process*.
6: **end if**
7: Answer to the purpose of the message. (Start a *directive handling process*, a *interrogative handling process*, or a *descriptive handling process*.)
```

Figure 10.1: The message handling process.

```
1: Ask for explication.
2: Receive explication: as a descriptive representation. (Handle with *message handling process*.)
3: After handling explication: matching phase, to merge or link existing concepts.
```

Figure 10.2: The explication process.

Internally, CROC uses entities instead of names with representations. Each received message is first 'internalized'; each outgoing message is first 'externalized', where 'internalizing' means a fallible process of reidentification by name that yields an internal representation, and 'externalizing' means the process of representing an internal representation by name.

Dialogue 1 Call by name. An interrogative representation can be answered by a descriptive: "I am a cooling service".

Employer: I'm looking for a cooling service.
Employee: It's me.

10.2.1 Handling interrogatives, descriptives, directives

Interrogatives

Handling an interrogative is straightforward: check if the query is satisfiable, or inconsistent with the knowledge base. The answer will substitute variables bound by the outcome of the reasoning process. If the agent does not know the answer, he will reply with "sorry".

Question answering There are various possibilities for returning an answer. An answer may contain of the following parts:

1. a quick indicator "yes", "no", or "unknown" of how the reasoning process terminated;

Dialogue 2 Inductive statements.

A: Swans are white.
OWL B: OK, I'll take that into the class definition.
CROC B: OK, nice to know.
A: There is a black swan.
CROC B: OK, nice to know.
OWL B: Error in [1], or unalignable classes for "swan".

2. possible variable assignments that account for "yes" (for the wh-parts of questions);
3. a trace of the reasoning that was done, possibly containing arguments pro and contra, the evidence that was used for deriving the answer, and so on.

Parts 2 and 3 may be stated as straightforward descriptives. For our purposes we only considered implementation of 2 combined with 1.

Descriptives

Handling a descriptive first checks for inconsistency, which would mean the descriptive is not acceptable; then checks if the descriptive is satisfiable, which would mean it is already known. If none of both is the case, the descriptive is added to the knowledge base.

The agent will reply with "reject" if a descriptive is inconsistent with his knowledge base. If not, the agent will reply with "accept".

Directives

Handling a directive has to be done with *built in capabilities* of the agent. If the agent fails, it sends back "sorry".

Explication request An example directive is the explication request. An explication request is modelled by a directive representation: "describe x (to me)". We cannot do with "what is x?", because what-is questions have a different meaning: to ask about the extension of the subject. See Dialogue 3 for an example.

Dialogue 3 A what-is question compared to an explication request.

Employer: What is a car?
Employee: A Toyota is a car, and many others...
Employer: (That is only an example. I need to restate my question to reach my purpose:) Describe a car to me.
Employee: A car is a vehicle with four wheels.

10.3 Subject matching

Subject matches are lead by analysing the subject templates.

First, when a subject is explicated to another agent, the inductive properties of the subject respective to the subject templates covering the subject are described. For example, in the subject template of A SERVICE is “what it is able to do”: see Dialogue 4.

Dialogue 4 Explicating a concept.

Employer: I’m looking for a cooling service.
Employee: What is a cooling service?
Employer: A service that is able to keep the temperature of a building below some temperature.
Employee: OK. I am a cooling service.

The agent requesting the explication may have a different subject template, and will ask accordingly if necessarily. See Dialogue 5 for an example.

Dialogue 5 Further completing knowledge about the subject by the subject template.

Employer: A car is a vehicle.
Employee: Thanks. (Ah. I know a vehicle by its number of wheels. Because I am gaining knowledge about something being a vehicle, and I don’t know how many wheels this vehicle has, I will ask about it:) How many wheels does a car have?
Employer: Four wheels.
Employee: Thanks.

Furthermore, after having gathered inductive knowledge about the subject, the agent will try to merge the subject concept with concepts it already possesses, or try to relate them. A concept for subject x is considered for merging with a concept for subject y if y and x fall under a same subject kind z, and all the properties in the subject template of z match. The agent will consider to merge these concepts, and may ask the other agent.

Dialogue 6 Deriving equality by inductive properties.

Employer: A car is a vehicle with four wheels.
Employee: Thanks. (Ah. I have another concept for a subject that is a vehicle and has four wheels; perhaps they are equal:) Does a car equal an automobile?
Employer: Yes (I have one concept for them; internalising yields identical entities).
Employee: Thanks.

These mechanisms make CROC able to learn and match subject concepts.

In this first implementational phase, we have not yet implemented further specific concept matching abilities, such as matching predicates relative to a property, and matching happenings using temporal logic. The system may be usefully extended with these capabilities.

10.4 Conclusion

In this chapter we have given a proof of concept of implementing a conceptuology by the system CROC: a Representational Ontology for Concepts.

In practice CROC is able to give agents abilities to learn and match concepts by the use of names and lexical representations. That is, we have been able to prove our concept as described in Part II.

Part IV

Conclusion

We have investigated the question 'Is the use of a conceptuology with conceptions based on language-like representations a realizable and valuable alternative, for artificial agents, to the use of shared classifications in 'ontologies'?'.

Description of a conceptuology Concepts for the things presented to us can be categorised in a set of basic classes. We have considered substance concepts, happening concepts, property, atom predicate, and relation concepts. Also, we mentioned that many other possible abstract concepts fall under subject concepts.

We discussed properties of subject concepts. (Most) substances are not classes. For a subject kind, there are properties that support induction: the subject template.

The 'is-a' relation is an important relation between subject concepts, but it may be inductive. Abstractions that make use of the 'is-a' relation are a widely used mechanism for giving background, but it is generally not a good idea to substitute concrete concepts by their abstract descriptions.

Finally, subjects may be related with happenings by thematic roles.

We argued that treating substances as classes will pose serious problems for identification. First, if 'ontologies' have to serve abilities to reidentify, they cannot have an ontological approach to senses, because conceptions differ from person to person. Second, class definitions are more artificial (and therefore difficult to match); class definitions push definitions from abstractions, thereby placing an extra burden for intelligent agents; sharing the class hierarchy doesn't share the concepts that shaped that hierarchy; class deduction is sensitive to context changes; classes without concepts are not grounded.

Realisation of a conceptuology The mechanism of lexical representation makes use of names for concepts, and is able to combine them in different uniform ways.

Lexical representations therefore can be used well to state (or question) properties of, and relations between its involved subjects.

Artificial agents can have concepts through language representations alone. Language-like representations, based on lexical concepts, plus reasoning, will be able to solve the interoperability problem to a large extent.

Evaluation of using the conceptuology Concepts based on representations are fallible, but these problems are not comparable with the interoperability problem that arises with different 'ontologies'. The practical implementation has to minimize the mentioned cases of fallibility.

Natural language can provide a rich basis for sharing words mutually between agents and between agents and humans.

We showed how different agents with different knowledge may communicate with private conceptuologies. Each agent can manage it on its own: there is no need for division of linguistic labour (where only experts 'own' the concepts), private concepts and conceptions are welcome. What agents share is the language-like way of representing using words.

Implementation of the conceptuology We have shown the possibilities of CROC: a Representational Ontology for Concepts are extensive, and set up a framework that uses representations for developing concepts for artificial agents.

The system CROC is suitable for extension in many directions. Reasoning mechanisms may be extended, classifications may be used for effective knowledge representation (after it has been identified by concepts), other conceptual abilities may be combined with the concepts based on representations.

In practice CROC is able to give agents abilities to learn and match concepts by the use of names and lexical representations. That is, we have been able to prove our concept as described in Part II by implementation of CROC: a Representational Ontology for Concepts.

Conclusion Concluding, we have shown a conceptuology with conceptions based on language-like representations is realizable, both by a theoretical investigation of the concept and a practical proof of concept by the system CROC.

We have also shown that class definitions have different problems with identification; moreover, the use of different classes leads to an interoperability problem. We therefore regard concepts as a valuable alternative, for artificial agents, to the use of shared classifications in 'ontologies'.

We regard our findings as highly relevant for how the Semantic Web should develop communication between agents and services in the web. CROC could develop a standard way of expressing content on the web, and become a system that provides abilities for representing and representation-based concepts to every agent, just like humans share a natural language and natural language handling capabilities.

Our approach of using concepts for reidentifying lexical representations also touches the field of natural language processing. We have carefully distinguished different concepts for different parts of representations; there is a (almost?) one to one mapping of the representations we use and natural language representations.

The way we use representations in reasoning involves a different, more extensive logical language than usual. It may be that using these representations (instead of simple propositions, or predicates) in logic is a more understandable and easier way of modeling the actual world.

Part V

Appendix

Appendix A

The CROC Ontology

The CROC ontology in OWL format is downloadable from the location `http://croc.sourceforge.net/croc.owl`.

In Figure A.1 we have depicted the relations in the ontology. Unmarked arrows indicate the subclass relation (the subclasses all are disjoint). We have indicated with "..." that under the class of subject concepts other subclasses may be added to extend the ontology.

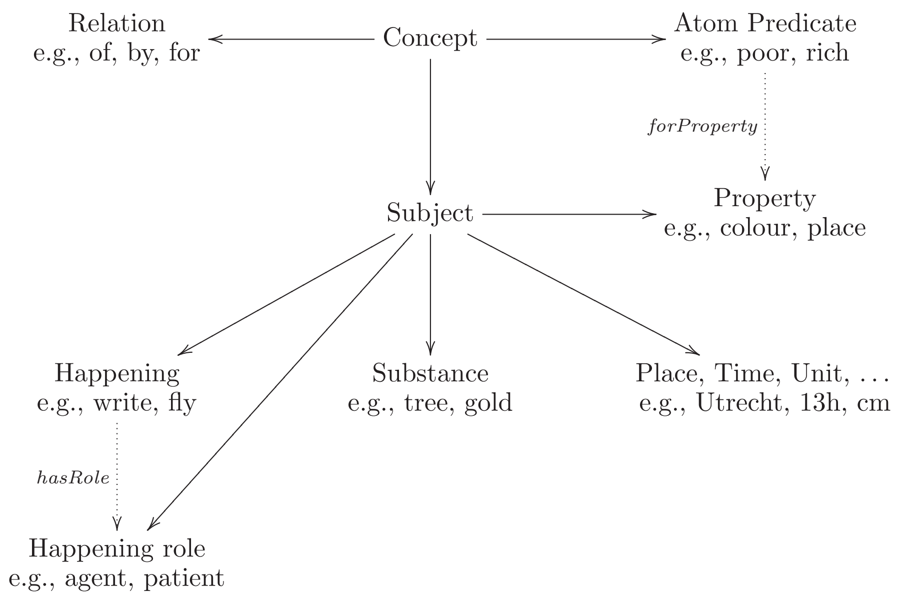


Figure A.1: Classification of concept kinds used in CROC (specified in OWL).

Every concept class is annotated with the annotation property `croc:baseConcept`, which refers the an instance of a subject concept. That is, there are concepts for A SUBSTANCE, A PROPERTY, A HAPPENING, etc. included in the ontology.

The division in subject *types* is made beneath the subject class (see Figure A.2).

Related knowledge in the form of XML representations (see Appendix B) is declared as a datatype property (of parse type `rdfs:XMLLiteral`) for every

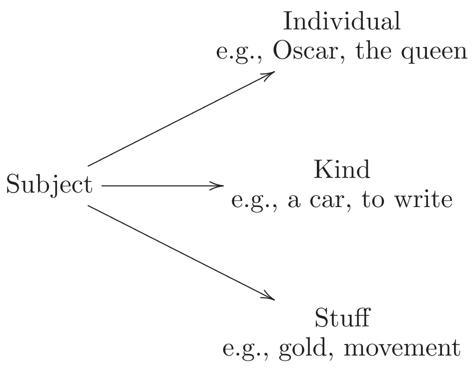


Figure A.2: The specific (disjoint) subject types, dividing the different classes of subjects in three groups.

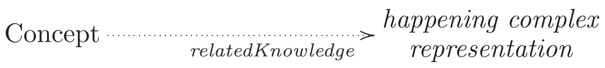


Figure A.3: The ‘related knowledge’ property.

instance of a concept (see Figure A.3). Related knowledge always has the form of a happening complex.

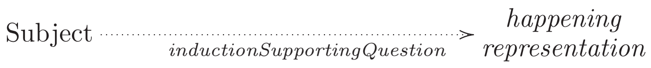


Figure A.4: The ‘induction supporting question’ property.

Subject templates are declarated using a datatype property for ‘induction supporting question’, that links to XML representations of a happenings (see Figure A.4).

The property `rdf:type` is used to refer to the specific subject type. The property `rdf:label` is used to refer to a name. The property `rdf:ID` may be used for establishing URI’s for concepts. Typical CROC XML examples can be found in Appendix C.

Appendix B

CROC XML Schema Elements

In this chapter we list the different representational elements used in XML for CROC, and shortly describe what they may represent.

The actual XML schema is downloadable from `http://croc.sourceforge.net/cr.xsd`.

B.1 Representation atoms

```
<cr:subject cr:resource="{URI}"/cr:name="{name}" />
```

A subject representation.

```
<cr:happening cr:resource="{URI}"/cr:name="{name}"
cr:tense="{tense_URI}" cr:aspect="{aspect_URI}">
   <cr:role cr:role="{role_URI}">
      <!--any (complex) subject-->
   </cr:role>
   <!--...-->
   <cr:predicate-role>
      <!--any (complex) predicate-->
   </cr:predicate-role>
</cr:happening>
```

A happening representation, which may involve one or more involved subject representations and involved predicate representations. Happening roles (see Section 3.4) are defined in CROC and can be referenced by URI. Tense and aspect likewise.

```
<cr:predicate-application>
   <cr:p><!--any (complex) predicate--></cr:p>
   <cr:s><!--any (complex) subject intension--></cr:s>
</cr:predicate-application>
```

A predicated subject representation, which falls under subject intension.

```
<cr:atom-predicate cr:resource="{URI}"/cr:name="{name}">
   (<cr:property cr:resource="{URI}"/cr:name="{name}" />)
</cr:atom-predicate>
```

An atom predicate representation. The property element is optional.

```
<cr:relation-predicate cr:resource="{relation_URI}">
   <!--any (complex) subject-->
</cr:relation-predicate>
```

A relation predicate representation.

```
<cr:happening-predicate (cr:role="{role_URI}")>
   <!--any (complex) happening-->
</cr:happening-predicate>
```

A relation predicate representation. The `cr:role` attribute may be used to specify the role of the subject that is predicated in the happening (without the `cr:role` attribute, the subject has to be referenced by ID).

```
<cr:quantified-subject>
   <cr:n><!--any numerical--></cr:n>
   <cr:s><!--any (complex) subject intension--></cr:s>
</cr:quantified-subject>
```

A numerical quantified subject representation, which falls under indeterminate subject extension.

```
<cr:quantify-over>
   <cr:n><!--any numerical--></cr:n>
   <cr:s><!--any (complex) subject extension--></cr:s>
</cr:quantify-over>
```

A numerical quantified determined-subject representation, which falls under indeterminate subject extension.

```
<cr:number cr:number="{decimal}" />
<cr:bare-number (cr:singular="singular") />
```

Numerical representations.

```
<cr:quantify cr:target="{ID␣of␣a␣numerical␣quantifying␣representation}">
   <!--any quantor-->
</cr:quantify>
```

Application of a quantor to a numerical quantifying representation. Quantors are ‘attached’ to the numerical quantifying representation. Numerical denotations can be used in determined expressions for set description as well.

```
<cr:universal />
<cr:existential />
<cr:uniqueness />
```

Quantor representations.

```
<cr:determined-subject>
   <cr:d><!--any determiner--></cr:d>
   <cr:s><!--any (complex) subject intension--></cr:s>
</cr:determined-subject>
```

A determined subject representation, which falls under determinate subject extension.

```
<cr:determiner cr:resource="{URI}" />
```

A determiner representation.

```
<cr:meta><!--any (complex) subject intension--></cr:meta>
```

A meta subject representation, which falls under subject intension.

```
<cr:variable />
```

A variable, which has meaning for the reasoner only. Any representation will be unified with the variable.

```
<x cr:ID="{ID}" />
<cr:reference cr:target="{ID}" />
```

The reference mechanism for representations. The `cr:ID` attribute introduces a reference ID for the current scope, the `cr:reference` element refers with `cr:target` to a representation in the current scope.

B.2 Representation complexes

```
<cr:and>
   <!--representation (complex) item-->
   <!--...-->
</cr:and>
<cr:or>
   <!--representation (complex) item-->
   <!--...-->
</cr:or>
<cr:xor>
   <!--representation (complex) item-->
   <!--...-->
</cr:xor>
```

And, or and exclusive or.

```
<cr:if-then>
   <cr:a><!--representation (complex) item--></cr:a>
   <cr:c><!--representation (complex) item--></cr:c>
</cr:if-then>
```

A logical implication.

```
<cr:not>
   <!--representation (complex) item-->
</cr:not>
```

Not.

```
<cr:quantifier>
   <cr:l><!--list of quantifies--></cr:l>
   <cr:c><!--representation (complex) item--></cr:c>
</cr:quantifier>
```

Attached quantifier.

Appendix C

Comparison of OWL and CROC examples

In this chapter we will compare some actual representations in CROC with similar representations in OWL. The chapter is largely meant to be illustrative for OWL users, and to easily point out the differences that have already been discussed in length in the thesis.

C.1 Example one

Listing C.1: Example 1: OWL Code

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xml:base="http://www.croc.name/my-owl-comparison-example"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#">
   <Book rdf:ID="Categories">
      <rdfs:label>Categoriae</rdfs:label>
      <rdfs:label>Categories</rdfs:label>
      <hasAuthor>
         <Author rdf:ID="Aristotle">
            <rdfs:label>Aristotle</rdfs:label>
         </Author>
      </hasAuthor>
   </Book>
   <owl:ObjectProperty rdf:ID="isWrittenBy">
      <owl:equivalentProperty>
         <owl:ObjectProperty rdf:ID="hasAuthor">
            <rdfs:domain>
               <owl:Class rdf:ID="Book">
                  <rdfs:label>book</rdfs:label>
               </owl:Class>
            </rdfs:domain>
            <rdfs:range>
               <owl:Class rdf:ID="Author">
```

```
                <rdfs:label>author</rdfs:label>
            </owl:Class>
         </rdfs:range>
      </owl:ObjectProperty>
   </owl:equivalentProperty>
  </owl:ObjectProperty>
</rdf:RDF>
```

Listing C.2: Example 1: CROC Code

```
<?xml version="1.0"?>
<!DOCTYPE RDF [
   <!ENTITY croc  "http://www.croc.name/croc" >
]>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xml:base="http://www.croc.name/my-owl-comparison-example"
xmlns:croc="&croc;"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
   <croc:Subject rdf:ID="AnAuthor">
      <rdfs:label>author</rdfs:label>
      <rdf:type rdf:resource="&croc;Kind" />
   </croc:Subject>
   <croc:Happening rdf:ID="write">
      <rdfs:label>write</rdfs:label>
      <rdf:type rdf:resource="&croc;Kind" />
      <croc:hasRole rdf:resource="&croc;Agent" />
      <croc:hasRole rdf:resource="&croc;Patient" />
   </croc:Happening>
   <croc:Substance rdf:ID="Aristotle">
      <rdfs:label>Aristotle</rdfs:label>
      <rdf:type rdf:resource="&croc;Individual" />
   </croc:Substance>
   <croc:Substance rdf:ID="ACategories">
      <rdfs:label>Categoriae</rdfs:label>
      <rdfs:label>Categories</rdfs:label>
      <rdf:type rdf:resource="&croc;Kind" />
      <croc:is>
         <croc:Substance rdf:ID="ABook">
            <rdfs:label>book</rdfs:label>
            <rdf:type rdf:resource="&croc;Kind" />
         </croc:Substance>
      </croc:is>
      <croc:relatedKnowledge rdf:parseType="Literal">
         <cr:happening cr:resource="&croc;has"
         xmlns:cr="/cr">
            <cr:role cr:role="&croc;Agent">
               <cr:subject cr:resource="~this" />
            </cr:role>
            <cr:role cr:role="&croc;RelationSubject">
               <cr:subject cr:resource="#AnAuthor" />
            </cr:role>
            <cr:role cr:role="&croc;Theme">
               <cr:subject cr:resource="#Aristotle" />
            </cr:role>
          </cr:happening>
        </croc:relatedKnowledge>
      </croc:Substance>
</rdf:RDF>
```

First, notice that CROC is less strict in the description of "author". Due to CROC, it is valid to say "this stone has Plato as author" and "this book has green as author", where "this stone" and "green" are respectively not a book and not falling under a class "author".

Second, notice that OWL places "Aristotle" in a class of authors. For CROC, no such class exists; there is a concept for an author, but that is all there is.

Third, notice that the property named "hasAuthor" in OWL is described more extensive in CROC by the verb "has" and the relation thematic role for "author".

The next thing we can do with CROC is to add knowledge about the "author" subject. We do so by adding that "something has someone as author, if and only if s/he writes it". The word "author" really is an economical expression for "the agent involved in the happening of writing", just like "sum" is for "the result involved in the happening of adding". The statement would not be placed in the related knowledge of subjects and substances, because that are the involved subjects.

Listing C.3: Example 1: CROC Code

```
<croc:relatedKnowledge rdf:parseType="Literal">
 <cr:quantifier
 xmlns:cr="http://www.croc.name/croc/cr">
  <cr:l>
   <cr:quantify cr:target="1">
    <cr:universal />
   </cr:quantify>
   <cr:quantify cr:target="2">
    <cr:universal />
   </cr:quantify>
  </cr:l>
  <cr:c>
   <cr:if-then>
    <cr:a>
     <cr:happening cr:resource="&croc;#has">
      <cr:role cr:role="&croc;#Agent">
       <cr:quantify cr:ID="2">
        <cr:n>
         <cr:bare-number cr:singular="singular" />
        </cr:n>
        <cr:s>
         <cr:subject cr:resource="&croc;#ASubstance" />
        </cr:s>
       </cr:quantify>
      </cr:role>
      <cr:role cr:role="&croc;#RelationSubject">
       <cr:subject cr:resource="~this" />
      </cr:role>
      <cr:role cr:role="&croc;#Theme">
```

```
        <cr:quantify cr:ID="1">
         <cr:n>
          <cr:bare-number cr:singular="singular" />
         </cr:n>
         <cr:s>
          <cr:subject cr:resource="&croc;#ASubject" />
         </cr:s>
        </cr:quantify>
       </cr:role>
      </cr:happening>
     </cr:a>
     <cr:c>
      <cr:happening cr:resource="#write">
       <cr:role cr:role="&croc;#Agent">
        <cr:reference cr:target="1" />
       </cr:role>
       <cr:role cr:role="&croc;#Patient">
        <cr:reference cr:target="2" />
       </cr:role>
      </cr:happening>
     </cr:c>
    </cr:if-then>
   </cr:c>
  </cr:quantifier>
</croc:relatedKnowledge>
```

For OWL, with help of `equivalentProperty` or `inverseOf` we can add the object property 'isWrittenBy'. The logical relation is there; but OWL cannot describe what "write" means.

Also notice that CROC uses a shared verb "has". In the description of it we can add "if x has y as z, then y is z of x" (e.g., "if x has y as author, then y is author of x").

C.2 Example two

Listing C.4: Example 2: OWL Code

```
<?xml version="1.0"?>
<!DOCTYPE RDF [
   <!ENTITY xsd  "http://www.w3.org/2001/XMLSchema#" >
]>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xml:base="http://www.croc.name/my-owl-comparison-example"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
   <owl:Class rdf:ID="Human">
      <rdfs:label>human</rdfs:label>
      <rdfs:subClassOf rdf:resource="#Being" />
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty>
               <owl:DatatypeProperty rdf:ID="mortal">
```

```
                <rdfs:range rdf:resource="&xsd;boolean" />
                <rdfs:domain rdf:resource="#Being" />
            </owl:DatatypeProperty>
        </owl:onProperty>
        <owl:hasValue
        rdf:datatype="&xsd;boolean">true</owl:hasValue>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Being">
    <rdfs:label>being</rdfs:label>
  </owl:Class>
</rdf:RDF>
```

Listing C.5: Example 2: CROC Code

```
<?xml version="1.0"?>
<!DOCTYPE RDF [
  <!ENTITY croc  "" >
]>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xml:base="http://www.croc.name/my-owl-comparison-example"
xmlns:croc="&croc;"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <croc:Substance rdf:ID="AHuman">
    <rdfs:label>human</rdfs:label>
    <rdf:type rdf:resource="&croc;Kind" />
    <croc:is>
      <croc:Substance rdf:ID="ABeing">
        <rdfs:label>being</rdfs:label>
        <rdf:type rdf:resource="&croc;Kind" />
      </croc:Substance>
    </croc:is>
    <croc:relatedKnowledge rdf:parseType="Literal">
      <cr:happening cr:resource="&croc;is"
      xmlns:cr="http://www.croc.name/croc/cr">
        <cr:role cr:role="&croc;Agent">
          <cr:subject cr:resource="~this" />
        </cr:role>
        <cr:predicate-role>
          <cr:predicate cr:resource="#mortal" />
        </cr:predicate-role>
      </cr:happening>
    </croc:relatedKnowledge>
  </croc:Substance>
  <croc:AtomPredicate rdf:ID="mortal">
    <rdfs:label>mortal</rdfs:label>
  </croc:AtomPredicate>
</rdf:RDF>
```

In the second example, notice that CROC describes “mortal” as an atom predicate, whereas OWL describes it as a datatype property with range boolean. For CROC, “humans are mortal” is an inductive statement, not a class restriction.

Where OWL defines humans as being a subclass of beings, CROC has an 'is-a' relation "humans are beings".

C.3 Conclusion

Of course, there is a lot more to compare between CROC and OWL. We believe these first sketches are sufficient to show the differences in functionality.

Bibliography

Aristotle. Categoriae. c. 350 B.C.

G. N. Carlson. *Reference to Kinds in English.* PhD thesis, University of Massachusetts Amherst, 1977.

J. de Bruijn, M. Ehrig, C. Feier, F. Martìn-Recuerda, F. Scharffe, and M. Weiten. *Ontology mediation, merging and aligning*, chapter 3. Wiley, UK, 2006.

J. Fodor and E. Lepore. The red herring and the pet fish: why concepts still can't be prototypes. *Cognition*, 2:253–270, 1996.

G. Frege. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, C, 1892.

P. Gärdenfors. How to make the Semantic Web more semantic. In A. C. Varzi and L. Vieu, editors, *Formal Ontology in Information Systems: proceedings of the third international conference (FOIS-2004)*, volume 114 of *Frontiers in Artificial Intelligence and Applications*, pages 17–34. IOS Press, 2004.

H. Grice. Logic and conversation. *Speech Acts. Syntax and Semantics*, 11:41–58, 1975.

R. Jackendoff. What is a concept, that a person may grasp it? *Mind and Language*, 4(1 and 2), 1989.

H. Kamp and B. Partee. Prototype theory and compositionality. *Cognition*, 57: 129–191, 1995.

M. Klein. Combining and relating ontologies: an analysis of problems and solutions. In *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, Seattle, USA, 2001.

S. Laurence and E. Margolis. *Concepts — Core Readings*, chapter Introduction. The MIT Press, 1999.

R. G. Millikan. *On Clear and Confused Ideas: An Essay about Substance Concepts.* Cambridge University Press, New York, 2000.

R. G. Millikan. On knowing the meaning, with a coda on Swampman. Originally prepared for the Jack Smart Lecture and a follow up lecture at Australian National University., 2007.

R. G. Millikan. *Language: A Biological Model.* Clarendon Press, Oxford, 2005.

H. Putnam. The meaning of 'meaning'. *Mind, Language and Reality. Philosophical Papers*, 2, 1975.

M. Sabou, M. d' Aquin, and E. Motta. Using the Semantic Web as background knowledge for ontology mapping. In *Ontology Matching*, 2006.

J. R. Searle. How performatives work. *Linguistics and Philosophy*, 12:535–558, 1989.

S. K. Semy, M. K. Pulvermacher, and L. J. Obrst. Towards the use of an upper ontology for u.s. government and u.s. military domains: An evaluation. Technical report, The MITRE Corporation, Bedford, Massachusetts, USA, 2004.

C. Shirky. The Semantic Web, syllogism, and worldview, 2003. http://www.shirky.com/writings/semantic_syllogism.html.

L. Steels. Language learning and language contact. In *Proceedings of the workshop on Empirical Approaches to Language Aquisition*, 1997a.

L. Steels. Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation. *Evolution of Human Language*, 1997b.

J. van Diggelen. *Achieving Semantic Interoperability in Multi-agent Systems — A Dialogue-based Approach.* PhD thesis, Utrecht University, 2007.

J. van Diggelen, R.-J. Beun, F. Dignum, R. M. van Eijk, and J.-J. Meyer. Anemone: An effective minimal ontology negotiation environment. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2006.

W3C Semantic Web Activity. W3C Semantic Web Frequently Asked Questions. http://www.w3.org/2001/sw/SW-FAQ.

H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based integration of information — a survey of existing approaches. In *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, pages 108–117, Seattle, USA, 2001.

M. Westera. Towards a semantic web for homo sapiens. Student paper for the course 'The Semantic Web'., 2008.

M. Woolridge. *An Introduction to MultiAgent Systems.* John Wiley & Sons, LTD, West Sussex, England, 2002.

M. Woolridge and N. Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2), 1995.