Achieving Semantic Interoperability in Multi-agent Systems

A Dialogue-based Approach

SIKS Dissertation Series No. 2007-4

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Printed by Gildeprint drukkerijen B.V., Enschede

ISBN 978-90-9021559-4

Achieving Semantic Interoperability in Multi-agent Systems

A Dialogue-based Approach

Het tot stand brengen van semantische interoperabiliteit in multi-agent systemen

Een dialoog gebaseerde benadering

(met een samenvatting in het Nederlands)

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Universiteit Utrecht
op gezag van de rector magnificus, prof.dr. W.H. Gispen,
ingevolge het besluit van het college voor promoties
in het openbaar te verdedigen
op woensdag 21 maart 2007 des middags te 2.30 uur

door

Jurriaan van Diggelen

geboren op 7 december 1976, te Leiden

Promotor:	Prof. dr. J.-J.Ch. Meyer
Co-promotoren:	Dr. ir. R.J. Beun
	Dr. F.P.M. Dignum
	Dr. R.M. van Eijk

Contents

Preface **ix**

1 Introduction **1**
1.1 Problem statement . . . 2
1.2 Research Aim . . . 3
1.3 Methodology . . . 4
1.4 Overview . . . 7

2 Agents and Ontologies **9**
2.1 Intelligent Agents . . . 9
2.1.1 Agents . . . 11
2.1.2 Multi-agent Systems . . . 12
2.2 Agent Communication . . . 13
2.2.1 Language as Action . . . 14
2.2.2 Agent Communication Languages . . . 15
2.3 Ontologies . . . 17
2.3.1 Ontology in Traditional Philosophy . . . 17
2.3.2 Ontology in Modern Philosophy . . . 20
2.3.3 Ontologies in Information Systems . . . 22
2.3.4 Task-Dependency of IS Ontology . . . 25
2.3.5 Ontologies in Multi-agent Systems . . . 28
2.4 Ontology Reconciliation . . . 29
2.4.1 Ontology Integration . . . 29
2.4.2 Ontology Alignment . . . 31
2.4.3 Ontology Mediation Services . . . 33
2.4.4 Ontology Negotiation . . . 34
2.4.5 Hybrid Approaches . . . 36
2.5 Conclusion . . . 37

3 Foundations of Ontology Reconciliation 39
3.1 Meaning . . . 39
3.1.1 The Peircean Model . . . 39
3.1.2 Sinn und Bedeutung . . . 41
3.1.3 Conventionality . . . 42
3.2 Meaning Representation . . . 42
3.2.1 World . . . 42
3.2.2 Description Logic . . . 43
3.2.3 Indistinguishable Models . . . 47
3.2.4 Symbol Grounding Problem . . . 50
3.2.5 Grounded Semantics . . . 51
3.3 Ontology Reconciliation . . . 54
3.3.1 Meaning Conveyance . . . 54
3.3.2 Alignment Criteria . . . 56
3.3.3 Comparison with Other Approaches . . . 57
3.4 Conclusion . . . 58

4 Ontology Reconciliation in ANEMONE 61
4.1 Assumptions . . . 61
4.1.1 Agent Components . . . 61
4.1.2 Degree of Heterogeneity . . . 64
4.1.3 Degree of Sophistication . . . 66
4.2 Approach . . . 67
4.2.1 Ontologies . . . 67
4.2.2 Overview of the Protocol . . . 71
4.2.3 Design Objectives . . . 72
4.3 Protocols . . . 73
4.3.1 Normal Communication Protocol . . . 73
4.3.2 Concept Definition Protocol . . . 83
4.3.3 Concept Explication Protocol . . . 85
4.3.4 Message Protocol . . . 89
4.4 Using ANEMONE in Larger Communities . . . 90
4.4.1 Acquired concept explication . . . 91
4.4.2 Unknown concept relations . . . 92
4.5 Discussion . . . 94
4.5.1 The Role of Symbolic Communication . . . 94
4.5.2 ANEMONE for Hybrid Ontology Reconciliation . . . 95
4.6 Conclusion . . . 96
4.6.1 Future Work . . . 97

5 Formal Protocol Analysis 99
5.1 Ontologies . . . 100
5.1.1 Knowledge and Dynamics . . . 104
5.1.2 Ontology Implementation . . . 106
5.2 Requirements . . . 108

5.2.1 Subjective Equivalence 108
5.2.2 Sound and Lossless Communication 109
5.2.3 Optimal Communication Vocabularies 110
5.2.4 Communication Vocabularies for Multiple Agents 111
5.3 Protocols 114
5.3.1 Implementing Automatic Ontology Mapping 114
5.3.2 Ontology Negotiation Protocols 116
5.4 Conclusion 127
5.4.1 Future Work 127

6 Simulation-based Strategy Analysis 129
6.1 Ontology Negotiation Strategies 130
6.2 Related Work 131
6.3 Model 132
6.3.1 Modeling Agents and Ontologies 132
6.3.2 Communication 135
6.3.3 Interoperability Measures 136
6.4 Word Selection Strategies 137
6.4.1 Description of Strategies 137
6.4.2 Experiments 138
6.5 Meaning Selection Strategies 144
6.5.1 Model 144
6.5.2 Description of Strategies 145
6.5.3 Experiments 147
6.6 Conclusion 153
6.6.1 Future work 153

7 Case Study of ANEMONE 155
7.1 Domain 155
7.1.1 RSS Aggregators 156
7.1.2 Personal News Agents 157
7.2 Implementation 158
7.2.1 Download News 159
7.2.2 Train Classifiers 159
7.2.3 Communication 161
7.3 Results 161
7.4 Conclusion 165

8 Conclusions 167
8.1 Results 167
8.2 Future work 169
8.2.1 Agents and the Semantic Web 169
8.2.2 Towards a Wider Scope of Applicability 171

Bibliography 173

Samenvatting **187**

Index **191**

SIKS Dissertation Series **195**

Preface

A modern DVD-video contains, besides the main film, a documentary on how the film was made. Following this practice, I would like to start by presenting "the making of ANEMONE". In the course of the last four years, I began to notice that doing a research project resembles in many respects the making of a film. A well-known lesson in film making is that the outcome not only relies on what is shot, but also on how it is shot. It is well-worth the effort to try shooting the same scene with different lighting techniques, camera angles and distances.

The subject of my "film" became clear to me soon after starting my PhD project. The ontologies of agents should not be like rigid trees with stiff branches, but more flexible like a sea anemone waving in the current. I knew that filming a sea anemone could produce a colorful picture. The task that lay ahead of me was to find the proper lighting and to prepare for all close-up and overview shots. Fortunately, I was not alone in this. In fact, it was rather crowded on the film set. Let me introduce you to my supervisors.

First (in alphabetic order), there was Robbert-Jan Beun. He was very good at directing the light from an unexpected angle. I recall him once saying "something *creaks* in your work". A long discussion followed in which we discussed the foundations of meaning, representation and the nature of being. The upshot of these discussions can be found in Chapter 3. I know I can still have lively discussions with him on these issues, but I hope the creaking sounds have disappeared.

My second supervisor was Frank Dignum. He liked to have a clear view: bright light on the front side. Besides his scientific help, he allowed me to take part in the organization of scientific events, which were valuable experiences. In 2005, the AAMAS conference was held in Utrecht in which I participated in the local organization team. It was a week's hard work doing things that I could never have imagined doing when I started as a PhD student. Thanking me for the effort, Frank gave me a box with Lion candy bars that were left over from the conference. I have never eaten a Lion since.

My third supervisor was Rogier van Eijk. He helped me with many close-up shots. We have had long discussions on many theoretical aspects of the thesis. He was also very good at finding illustrative examples. He once encouraged me to use two professors from our department in a cartoon example. One of these was my promotor, and the other a member of my reading committee. At that time, I had only been in the department for two months and was not sure whether it would be a good idea to make jokes about my employers. Fortunately, they all took it well, which Rogier, of course, already knew. I believe it is characteristic for the relaxed and friendly atmosphere in our department.

The fourth person who supervised me was John-Jules Meyer, my promotor. John-Jules enjoys a lot of light. Usually, light was shining from all directions during the discussions with my four-headed supervision team. That was when John-Jules was at his best. He encouraged me to conduct research on a broad range of issues, which I fully enjoyed. Sometimes the light blinded me. John-Jules was understanding at these moments and provided me with sunglasses. For all his friendliness and tolerant scientific views, John-Jules sometimes said "I smell a rat". That was when there were flaws in my work. I must say that, over the last four years, my research has benefited considerably from John-Jules' well developed olfactory sense.

For four years we have worked on the ANEMONE system. I cannot deny that I have sometimes been stung by the anemone, but overall, working on this project has been an experience I wouldn't have missed for anything. I am particularly grateful to my supervisors, Robbert-Jan, Frank, Rogier and John-Jules. Thank you all for all those pleasant meetings and interesting discussions, for the excellent steersmanship, for the support, and for hearing the creaks and smelling the rats.

I would also like to thank the members of my reading committee: Jörgen van den Berg, Les Gasser, Frank van Harmelen, Walt Truszkowski and Albert Visser. It was an honor for me to have my thesis read by such experts in the field.

I also thank the people from my department. Many people have been a continuous source of inspiration. I thank Marco Wiering for his help on the machine learning aspects of the thesis. I thank Davide Grossi for his help on some logical aspects of the thesis. I thank Virginia Dignum for many pleasant teaching experiences and for organizing scientific events with me. I also thank all the other people with whom I have coöperated or whose ideas I impertinently stole.

Besides the work aspect, the people of my department have been pleasant company. A couple of memorable events include snowboarding in France with Birna, Geert, Davide and Henk-Jan, bicycle riding in Japan with Mehdi and Birna and checking out Chinese restaurants in New York with Henk-Jan. Also in our home department, there always was a pleasant atmosphere. This was due to my roommates Huib, Davide and Bob as well as to the other PhD students: Cees, Paul and Susan. I also thank Mehdi, Jan and Martin for the many heated discussions we had during our lunch breaks as well as Henry and Gerard for staying rather relaxed at these moments. Thanks also to Richard for all the SIKS courses I took. During the last year, our department has experienced an enormous growth, but the pleasant atmosphere has remained. I would like to thank the newcomers for that.

Finally, I would like to thank my friends and family. Special thanks go to my brother Marinus, for helping me with the cover and to my other brother Carel for sharing his experiences as a filmmaker with me. Thanks to my parents for their love and support and, of course, for their *genes and memes* without which this thesis would not be. My final thanks and love go to my girlfriend Laura. She has followed "the making of" from day one and has always been there for me behind the scenes.

Chapter 1

Introduction

> *Behold, the people is one, and they have all one language; and this they begin to do: and now nothing will be restrained from them, which they have imagined to do. Go to, let us go down, and there confound their language, that they may not understand one another's speech.*
>
> Genesis 11, 6-7

A future scenario presented by Luck et al. (2003) bodes well for those who like to travel. It describes a software-based travel agent assisting a European customer in planning a holiday trip to the United States. The travel agent takes a number of tasks upon itself, such as finding a cheap flight, arranging transport in the US and booking accommodation. The agent does not blindly follow orders. It suggests other possibilities when appropriate and revises previous plans when unexpected situations arise.

The recent technological developments that have brought this scenario within reach have also revealed some obstacles along the way. One of these obstacles is the problem of enabling different software agents to cooperate with each other. The travel agent is situated in a multi-lingual environment. In order to collaborate with other agents, it should be able to deal with language differences. For example, the travel agent might encounter another agent representing a French airline company that refers to a *flight* as a *voyage en avion*. If the travel agent does not know that a *voyage en avion* means the same as a *flight*, misunderstandings could arise which would obstruct their cooperation. But there are even deeper language problems lying in wait for the travel agent. For example, it might encounter an agent representing an airplane technician speaking of a delay caused by problems with the *strapdown-inertial-sensor*. In this case, the technician is using a word that means something completely strange to the travel agent and that is not even translatable to its own vocabulary. For successful collaboration, agents must be able to deal with unknown words and unfamiliar meanings. This is the problem that is investigated in this thesis.

The aim of this chapter is to acquaint the reader with this problem and our approach towards it. Section 1.1 presents the problem statement. Section 1.2 describes our research aims. The methodology is described in Section 1.3. An overview of the thesis is presented in Section 1.4.

1.1 Problem statement

Two concepts that have played a prominent role in artificial intelligence research over the last decade are that of an *agent* and that of an *ontology*, both of which are briefly described below.

An agent is usually defined as an autonomous software entity, situated in some environment, that exhibits reactive and pro-active behavior (Wooldridge and Jennings, 1995). Furthermore, an agent is typically part of a larger community, i.e. a *multi-agent system* (MAS). Such an agent should be able to socially interact with the other agents to coordinate its actions and share its knowledge, i.e. the agents should be *semantically interoperable*. Agents have been advocated as key components of *complex systems*, i.e. systems with a large and unpredictable domain, or systems whose constituents are not known beforehand and might change over time (*open systems*). The travel agent scenario is a typical example of a multi-agent system possessing these properties.

An ontology is usually defined as a specification of a conceptualization (Gruber, 1993b). For example, it may specify a taxonomy stating that the concept *flight* is a special kind of the concept *means-of-transport* and that the concepts *long-distance-flight* and *short-distance-flight* are special kinds of the concept *flight*. Ontologies have been advocated as the key to semantic interoperability in distributed systems. Different system components sharing the same ontology can exchange their knowledge fluently as their knowledge representations are compatible with respect to the concepts regarded as relevant and with respect to the names given to these concepts.

Since semantic interoperability is of crucial importance to agents, and shared ontologies are viewed as the technology enabling semantic interoperability, it is not surprising that ontologies have been embraced by the agent community. However, a shared ontology is difficult to realize in a MAS. Because MAS's are typically open and have no centralized designer (Huhns and Stephens, 1999), it is practically impossible to guarantee that every agent in the system will use the same ontology. In open systems, new agents may enter the community at any time. Such newcomers may very well use an ontology different from the rest. Moreover, without a centralized designer, it is very unlikely that different system developers equip their agents with exactly the same ontology.

If the use of an ontology is not the ready solution for semantic interoperability in multi-agent systems, some further measures are required. As indicated by the thesis' title, the problem dealt with in this thesis is as follows.

Problem statement:

How to achieve semantic interoperability in multi-agent systems.

Heterogeneous ontologies are a natural consequence of the way in which multi-agent systems are built up. The problems that arise are of the kind illustrated in the travel agent scenario. One agent speaks of a *flight* which is not understood by the other agent, which calls it a *voyage en avion*. In this case, the agents use different names for the same concepts. As another example, consider the problem arising when one agent speaks of a *strapdown-inertial-sensor*, which refers to a concept unknown to the other agent. In this case, the agents have different areas of expertise which is reflected by a different set of concepts in their ontologies.

Such problems might be avoidable in a very simple MAS by enforcing one common ontology. However, in complex or open systems, where the advantages of the multi-agent approach most strikingly appear, problems with heterogeneous ontologies are bound to occur.

1.2 Research Aim

Because our work concerns multi-agent systems, the solution for semantic interoperability problems must conform to the characteristics of agents. These characteristics place some important restrictions on the kind of techniques that are applicable. More optimistically however, they allow us to limit the scope of our investigation and set up an achievable research aim for this thesis. Three of such characteristics are described below.

Firstly, the solution must be applicable to a *decentralized* multi-agent architecture. Because an agent is assumed to be autonomous and social, every agent in a MAS should be able to solve its own communication problems and enable itself to socially interact when necessary. This excludes the possibility of involving humans in the process of achieving semantic interoperability. Furthermore, it disqualifies centralized approaches (e.g. specialized services that mediate between agents with interoperability problems) as these do not enable the agents to solve their interoperability problems autonomously.

Secondly, the solution must be applicable in *dynamically changing* systems. This excludes the possibility to solve all problems at design-time because new problems could arise at run time when new agents enter the system. The solution must be applicable at agent interaction time.

Thirdly, the solution must be *computationally attractive*. Agents are assumed to respond in a timely fashion to their environment and to other agents. Therefore, they should not spend too much time on heavy computations while deciding what to do next. One of the measures to improve computational efficiency is to solve semantic interoperability problems on an as-need basis. In this way, the agents only spend computational resources on those semantic interoperability deficiencies that cause problems. Another measure is to ensure that, when agents decide that problem solving is unavoidable, the solution is materialized in a form that demands a minimal amount of storage and computational resources.

Now, the initial directions for our investigations can be set out. Firstly, we will follow an ontology-based approach towards achieving semantic interoperability.

As argued, an ontology is usually insufficient to provide for semantically interoperable agents. Our approach aims at repairing these shortcomings of an ontology. Secondly, we will follow a dialogue-based approach. The reason for this is that it conforms nicely with the characteristics of agents discussed above. A dialogue-based approach is distributable in a multi-agent system, as every agent possesses skills to participate in dialogues. Furthermore, the solution using dialogues is applicable at agent interaction time. In this way, the agents preserve their heterogeneous ontologies, but are equipped with the right conversational skills to overcome semantic interoperability problems.

Agents do not spontaneously have dialogues. They require carefully designed dialogue mechanisms that prescribe how the agents should react under which circumstances. The research aim of this thesis can thus be formulated as follows:

Research aim:

> *To develop computationally attractive dialogue mechanisms for achieving ontology-based semantic interoperability in multi-agent systems.*

In the travel agent scenario, on could think of the following dialogues. The travel agent meets a French airline agent, and begins a dialogue by asking for a particular *flight*. The French airline agent does not understand the word *flight* and makes this known to the travel agent. The travel agent continues the dialogue by explaining the meaning of *flight*, after which the agents resume their original conversation. On another occasion, the travel agent misunderstands the airplane technician agent speaking of a delay caused by a *strapdown-inertial-sensor* problem. The airplane technician chooses not to clarify this particular concept, but to rephrase its original message as a *technology* problem, which is understood by the travel agent.

The first example is remarkable, as language itself becomes the topic of conversation and is not only used as a means to exchange information. This could solve semantic interoperability problems. The second example shows that agents can steer their dialogue in such a way that the relevant information is conveyed differently, using more understandable terms. This could make the approach computationally attractive. In agent communication, not all information that is possessed by one agent must necessarily be conveyed to the other agent. By distinguishing between relevant and irrelevant information, the agents can decide which semantic interoperability problems are in need of a solution and which are not.

1.3 Methodology

Besides the *designing* of dialogue mechanisms, our research also concerns the *evaluation* of the proposed mechanisms. For example, if the travel agent decides that a *flight* means the same as a *voyage en avion*, we must be able to analyze whether the agent is justified in this decision. If the travel agent regards learning the meaning of *strapdown-inertial-sensor* as needless, we must be able to evaluate whether this concept indeed contains information unnecessary to the travel agent. In this thesis, we adopt a multidisciplinary research methodology to investigate such issues.

The prime motivation for this is the complex nature of the problems involved. One of these problems is to automatically find translations between ontologies, for example the translation of *voyage en avion* to *flight*. As argued by Dou, McDermott and Qi (2004), this problem is "AI-complete", meaning that a program that solved it would have to be as intelligent as a person. Clearly, there is no uniform approach towards solving this problem. It therefore requires investigation from multiple perspectives. In addition, the dialogue mechanisms themselves have many aspects. For example, consider the dialogue in which the agents circumvent the need to learn the concept *strapdown-inertial-sensor*. The dialogue demands a philosophical investigation, since philosophers traditionally devote much attention to meaning and language use. In addition, the workings of the dialogue mechanism should be stated mathematically, to enable a more precise analysis. Other insights can be gained by performing simulation experiments in which multiple agents use the dialogue mechanism to communicate with each other. Ultimately, the proposed mechanism should be validated by implementing it in order to find out if it works in the "real world".

The research described in this thesis has been conducted in both the context of *computer science* and of *information science*. This has also led to a multidisciplinary approach, providing different perspectives on the subject. From a computer science perspective, we have focused on a technological solution and investigated its formal properties. From an information science perspective, we have focused on how information is used and shared within an organization (in this case a multi-agent system).

In this thesis, we will use each of these methods to validate our results. This allows for a better understanding of the problem, its possible solutions and the limitations of the proposed solutions. The disciplines and research methodologies which have inspired our research are discussed in more depth below.

Philosophy

The topic of ontology has a long history in philosophy. Because ontologies play such an important role in our work, we will briefly discuss philosophical views on ontologies and argue how these insights can be applied to ontologies in computer systems.

Furthermore, we will consider the philosophy of *symbolic representations* with which, at its most fundamental level, our research is concerned. This involves thorny issues such as what a symbol is, how it can represent something else, and what it can be made to represent. These kinds of problems have a long philosophical tradition. Most artificial intelligence research disregards this tradition and assumes that such problems are solved in the knowledge representation language adopted. However, in our research philosophical ideas on representation are highly relevant. These enable us to obtain a clear understanding of the limitations of knowledge representation languages, the nature of the problem and its possible solutions.

Formal analysis

Some properties of dialogue mechanisms can best be analyzed *formally*. By introducing a formal framework, we obtain the means to state *precisely* the workings of the communication mechanism and its desirable properties. Furthermore, it allows us to give a solid proof that the proposed mechanism actually possesses the properties we have claimed.

Simulation-based analysis

Another method we will employ to analyze our proposals is based on simulation. When multiple agents are involved that resolve their communication problems in a pairwise fashion, the situation may become too complex to be formally analyzable. For these cases, the use of simulation experiments is a proper method of investigation. In this thesis, we describe simulation experiments carried out to investigate the effects of a single agent's communicative behavior on the overall semantic interoperability of the system.

A simulation experiment usually adopts a simplified model of the real situation to make the experiment computationally feasible. By running the experiment several times, statistically valid results can be obtained. However, the critic might claim that these results only apply to the simplified situation. This is indeed a methodological danger of this investigation method. We must be careful to ensure that all aspects that are relevant for the properties under investigation are included in the model. Of course, it may be debatable as to what qualifies as a *relevant* aspect. Nonetheless, the simulation-based approach is the only way to make progress in understanding many complex problems in artificial intelligence (AI).

Case study

An old proverb says, "the proof of the pudding is in the eating". We serve our pudding in the form of a software implementation. This enables us to test our dialogue mechanisms in the "real world". The software implementation focuses on a specific case. However, in our vision this case contains many aspects that are typical for multi-agent systems. Therefore, the findings of this case study can be generalized to many other multi-agent applications.

Using appropriate analogies

Agent research is strongly inspired by analogies. For example, an agent is viewed as analogous to a human, and a multi-agent system is viewed as analogous to an organization. These analogies are useful to developers of multi-agent systems as they provide an intuitive understanding of the way in which the system is built up. In addition, the analogies are useful to researchers as many of the problems that exist in multi-agent systems have their counterparts in human society. Researchers can use the solutions encountered in human society as an inspiration for multi-agent systems.

Throughout this thesis, we use a strong analogy. Agents with heterogeneous ontologies are like people speaking different languages. Of course, an argument from analogy may not always be valid and should never be held as scientific proof on its own. The analogy primarily serves an explanatory purpose, i.e. to provide the reader with an intuitive understanding of the proposed solution. However, a close analogy with human communication might also support a given solution because human communication contains many ingenious solutions to interoperability problems.

1.4 Overview

The chapters in this thesis are roughly organized according to research methodology. The philosophical perspective is provided in Chapter 3, the formal analysis is given in Chapter 5, the simulation-based analysis in Chapter 6 and the case study in Chapter 7. Appropriate analogies are used throughout the thesis. An overview is as follows:

Chapter 2 discusses background information and related work on agents and ontologies. It serves to deepen the understanding of the concepts touched upon in Chapter 1 and introduces additional notions that are used throughout the thesis.

Chapter 3 discusses the foundations of ontology reconciliation. We introduce some main ideas from philosophy and logic fundamental to our research. By interpreting and combining these ideas, we describe some limitations of symbolic representations and suggest how these can be overcome.

Chapter 4 introduces, in a formula-free style, our proposed solution to the research problem, called ANEMONE. Furthermore, it describes motivations for the major design decisions behind ANEMONE. The chapter is based on published work (van Diggelen et al., 2006a).

Chapter 5 provides a formal analysis of the solution proposed in the preceding chapter. It introduces a formal framework which is used to prove some of the main properties of the communication mechanism. The chapter combines published work (van Diggelen et al., 2004, 2006b, 2007).

Chapter 6 presents a simulation-based analysis of the communication mechanism. In particular, we investigate which choices, available to a single agent, are most beneficial to the whole community of agents. The chapter combines published work (van Diggelen et al., 2005, 2006c).

Chapter 7 presents a case study. It describes the application of ANEMONE to a particular domain, discusses the implementation details and presents the results.

Chapter 8 concludes the thesis and reflects the results in the light of the initial research aims presented in Chapter 1. Furthermore, it suggests directions for future research.

Chapter 2

Agents and Ontologies

Nomina sunt consequentia rerum.
(Names are the consequence of things.)

Dante Alighieri (La Vita Nuova, 1293)

Whereas the problem addressed in this thesis is "hot" on the current AI research agenda, many of the issues involved have a long scientific history. To understand which direction research should take, it is important to know where we stand and where we came from. Both issues are addressed in this chapter and aim at providing the reader with a basic understanding of the central concepts and ideas dealt with in this thesis.

The chapter is organized as follows. We start very broadly by discussing intelligent agents in Section 2.1. We narrow down on agent communication in Section 2.2, and on ontologies in Section 2.3. In these sections, it is not our intention to give a complete overview of the field. Rather, we will restrict ourselves to those aspects that are relevant for the current investigation. Section 2.4 presents a literature overview of the study area that is closest to the subject of this thesis, namely ontology reconciliation. Section 2.5 concludes the chapter.

2.1 Intelligent Agents

Since the beginning of computers, there has been a continuous struggle to design computer systems that are comprehensible, maintainable and adjustable. A weapon in this struggle has always been *modularization*, i.e. dividing the system into multiple relatively independent components. By giving a system the right modular structure, system developers can focus on smaller parts of the problem that are easier to comprehend. The result is also known as a *separation of concerns*(Ghezzi et al., 1991).

The success of modularization to achieve separation of concerns is demonstrated by the evolution of computer programming languages. One of the early developments in programming languages was the emergence of structured programming around 1970. Computer programmers became aware of the limitations of unstructured programs consisting of a long, cluttered list of instructions and "goto" statements that redirect the flow of control from one part of the program to the other. The structured programming paradigm involves modularization of a large structure of instructions into smaller substructures (subroutines), which contain fewer instructions and can be better understood.

The structured programming paradigm was overtaken by the object-oriented programming paradigm in the mid 1980's. This involved a further modularization by introducing *objects*, i.e. packages of subroutines and data. In structured programming, the data of a program can be seen to be present at a central location. In object-oriented programming, the data is *decentralized*, i.e. distributed over the objects in the program. This modularization turned out to be successful. For example, it has facilitated software to be built by a *development team*, enabling different persons to implement different objects that later form the whole application.

Up to now, finding the right modularization has been one of the primary driving forces in software engineering in general. Many people agree that also the current software development techniques have reached their limits. As computer systems are applied to more and more complex domains, it becomes increasingly difficult to provide for a comprehensive solution that can be realized in practice. The measures that are proposed to solve these problems can be seen as a continuation of earlier developments in software engineering, i.e. achieving separation of concerns by modularization. Viewed from a programming perspective, this new paradigm holds that not only *data* is decentralized (as in object-oriented systems), but also that the system's *execution* is decentralized. The different components run in parallel. Furthermore, the entities into which the system divides are relatively independent of each other, allowing system developers to focus on one thing without having to worry about how this will fit in the overall system. A key notion in such a software system is that of an autonomous *agent*. A system consisting of a number of agents, simultaneously performing their actions, is called a *multi-agent system* (MAS).

Whereas this next step in technology, as introduced above, might seem practicable with relative ease, the contrary is the case. Multi-agent systems can be regarded as a revolution in computer system architecture. The division of a system into autonomous agents creates problems that are new to the software engineering discipline and which have previously only been addressed by researchers in artificial intelligence. In addition to artificial intelligence, the MAS discipline also draws from research in philosophy, sociology, economics and cognitive psychology.

Whereas MAS's have been a very popular research area over the last decade, critics have hinted at the disappointing number of real world applications that have been produced so far. Most likely, this should not be attributed to the MAS paradigm itself, but to the difficult nature of the subject matter. The MAS paradigm advocates a necessary step in software engineering: modularization into autonomous entities to achieve separation of concerns. It is undisputed that modularization to achieve separation of concerns is key to successful software engineering. Furthermore, modularization into autonomous entities is evidently a *useful* modularization: the entire human society works on that principle.

In the following two sections, we will discuss the notions of agents and multi-agent systems in more depth.

2.1.1 Agents

As argued by Wooldridge and Jennings (1995), an agent can be described as a software system that enjoys the following properties:

- autonomous
- social ability
- reactivity
- pro-activeness

An agent possesses *autonomy* if it is capable of performing its own actions without intervention of humans or other components in the software system. This is perhaps the most fundamental property of an agent. We have emphasized the application of agents to achieve a separation of concerns. This is only achieved when the individual agents are independent of each other, i.e. when they are autonomous.

Having assumed that agents are autonomous, the idea of adopting a team of agents that collectively approach a problem requires further attention. Whereas each agent is capable of autonomously solving parts of a larger problem, some coordination mechanisms are required to guide the overall process. For example, the task of building a house requires the bricklayer to coordinate its actions with the roofer so that the roof is built only after the sidewalls are finished. Such coordination mechanisms require *social ability* from agents. This important property of agents is discussed in more depth in Section 2.2.

Reactivity refers to the agent's ability to perceive its environment and respond to it in a timely fashion. This introduces the idea that, besides other agents, there is something external to an agent, namely an environment in which it is situated. This property can also be seen as a straightforward consequence of our earlier discussion. We argued that agents are used to solve a certain problem. Clearly, the problem does not exist in the agents themselves, as these are the *means* to solve the problem. Hence, there must be something external to the agents in which the problem resides, i.e. their environment. This environment may be the physical world, the internet or a user via a graphical user interface. To be able to solve problems, the agents must be capable of interacting with their environment: they perceive it and are capable of performing actions in it.

Pro-activeness means that agents pursue their goals not only by reacting to their environment, but also by taking the *initiative*. This property is closely related to an agent's autonomy. An agent does not need instructions on when and how it should act. It is capable of acting on its own.

Cognitive agents

Agents are often characterized in terms of *mental* notions, such as knowledge, belief, desire, obligation and intention. Such agents are called cognitive agents. The idea of describing a system in terms of mental notions is inspired by the *intentional stance* (Dennett, 1989). Dennett argues that a complete and accurate low-level description of how a system works, is not always practicable to explain and predict the system's behavior. Such a low-level description may simply be too complex to see the forest for the trees. By adopting the intentional stance, the system is viewed at a higher level of abstraction, in terms of mental properties. Such mental properties provide a useful and valid abstraction tool to understand a complex system, such as a multi-agent system.

A famous architecture for cognitive agents is the BDI architecture (Rao and Georgeff, 1991), which characterizes agents in terms of beliefs, desires and intentions. An example of a programming language based on that architecture is 3APL (Hindriks et al., 1999).

2.1.2 Multi-agent Systems

As argued, modularization into autonomous agents greatly facilitates the development and maintenance of a software system. This benefit is most noticeable when a team of developers is involved in the realization of a MAS. Ideally, every developer focuses on a few agents without worrying on how these will fit in the overall system. This reduces the development time considerably as the people involved in the project are not required to spend time to tune their design decisions.

In the maintenance phase, the modular structure of a MAS is also beneficial. It enables a developer to adjust one component of the system, namely the agent, while leaving the other components unaltered. Furthermore, new agents can be easily added to the system to keep the system up to date.

Besides these improvements for development and maintenance, the MAS approach allows a software system to be realized in a way that is unprecedented in the history of software engineering. These are so-called *open multi-agent systems*. An open MAS allows software agents to dynamically enter and leave the system. It is not known in advance which agents will constitute the system. Every developer can contribute by adding his or her agent to the system. As the internet has demonstrated, such open architectures might produce undreamed-of results. Examples of open multi-agent systems are electronic institutions, e.g. electronic auction houses (Rodriguez, 2001), and the travel agent scenario described at the beginning of Chapter 1.

A consequence of a MAS being open is, most likely, *heterogeneity*. In a system that is accessible for every agent regardless of their background, the differences between agents may be considerable. For example, agents may have conflicting goals, which demands for negotiation protocols (Rosenschein and Zlotkin, 1994). The agents may also be heterogeneous with respect to their beliefs, as discussed by

Lebbink (2006). Moreover, the agents' ontologies may be heterogeneous. This is investigated in this thesis.

2.2 Agent Communication

The step from a collection of individual agents to a system of interacting agents is expected to proceed fluently. After the agents are assembled, they should automatically form a cooperating unity. Of course, this is not something that can be taken for granted. For this purpose, an interaction mechanism has been proposed, known as an *agent communication language*, which is another aspect of agents that can rightly be called revolutionary. Below, we will argue why it is needed and what is so new about it.

Consider the on/off button of a vacuum cleaner. The button can be viewed as an interface between the person using it and the motor of the vacuum cleaner. This is an example of a very simple interface. There is no sense in which messages are used in this process. Therefore, we would not speak of communication between the user and the vacuum cleaner. Furthermore, the button *directly* connects the input (the user pushing the button) to the output (the motor switching on and off). There is no freedom of interpretation by the vacuum cleaner. We say that the interface establishes a *deterministic connection*.

Now, consider a computer that is connected to a printer. The computer instructs the printer to start a new line by sending a control character for carriage return (CR) through the printer cable to the printer. According to Shannon's theory of communication, this process qualifies as communication as we can identify a sender, a channel through which information is transmitted, and a receiver (Shannon, 1948). We say that the printer is made *syntactically interoperable* with the computer if the message (the CR character) triggers the right process in the printer, namely to start a new line. We call this kind of interoperability syntactic, as the reaction of the receiver is determined purely by the *form* of the message.

On the one hand, the printer's interface is considerably more complex than that of an on/off button, as it involves a communication process. On the other hand, the interface does not differ so much from the on/off button as it still establishes a deterministic connection. The printer has no freedom how to interpret the characters that it receives. The communication involved here is very elementary, as the messages are restricted to imperative orders. In fact, the CR character sent down the printer cable is analogous to a finger pushing the on/off button of the vacuum cleaner. In both cases, the device receiving the input has no freedom on how to react.

Interfaces that establish a deterministic connection are ubiquitous in computer science. Examples are graphical user interfaces (GUI's), objects interacting through method calls and SQL-based interfaces for databases. For most applications, such a deterministic connection is fine and even desirable. For example, a button in a GUI that responds one time with OK and another time with Cancel would generally be considered bad user interface design.

Nevertheless, a deterministic connection is not suitable for agents for which autonomy is essential. In order to keep agents in control of their own actions, their communication should not solely consist of imperative orders. A more flexible interface is required that allows agents to coordinate their actions, but that leaves the final decision on how to react to a message to the agent. Generally, the agent's reaction to a message will be guided by what the agent considers to be in its own interest under the current circumstances. This principle of agent communication is nicely summarized by the following slogan: "objects do it for free, agents do it for money" (Jennings et al., 1998).

This idea entails that the meaning of a message can no longer be identified with the actions that it causes the receiver to perform. In order to preserve the autonomy of agents, the meanings of the messages exchanged must be defined *independently* of the specific internal workings of agents. Therefore, contrary to the other forms of communication we have discussed so far, agent communication can be viewed to proceed via a *language* that has its own semantics. In order for an agent communication language (ACL) to provide for interoperability, all agents must ascribe the same meaning to the language. In other words, they must be *semantically interoperable*.

Semantically interoperable agents are capable of understanding each other's messages, but are free to react to them as they please. Therefore, an ACL establishes a non-deterministic connection between agents as it does not fully specify how it determines the agents' behavior. This type of interface can rightly be called revolutionary in software engineering. Over the last decade, agent communication has drawn much interest from the research community. Much of this research has been inspired by human communication, which is the topic of the next section.

2.2.1 Language as Action

Traditionally, philosophers have studied human language by focusing on the *truth* of an utterance. Over the last fifty years, there has been a shift of attention towards the *effects* of an utterance. As pointed out by Austin (1962), language can best be understood as a kind of *action*. Speakers do not simply utter sentences that are true or false, but their words function as actions that change the world in some sense. The theory that views language as action is known as *speech act* theory.

Consider the following utterances: "I now pronounce you husband and wife", "You're fired" or "The meeting is adjourned". Each of these utterances performs a speech act. The effects of these speech acts can be understood as changing some institutional state of affairs. However, speech acts are not restricted to these matters. Also questions, apologies and assertions can be regarded as speech acts.

The elementary form of a speech act is *F(p)*, where *F* is the *illocutionary force* and *p* the propositional content (Searle and Vanderveken, 1985). The illocutionary force of an utterance is the speaker's intention in producing that utterance. The propositional content refers to what the utterance is about. For example, the sentences "The shop is closed", "Is the shop closed?" and "I apologize for the shop being closed" express the same propositional content, but with a different illocutionary force.

Many speech acts are performed *indirectly* (Bach, 2006). For example, one may utter, "The shop will be open tomorrow at 9:00" to indirectly convey that the shop is closed now. Such hidden meanings can be explained using the theory of conversational implicatures proposed by Grice (1975). According to Grice, cooperative communication is guided by four maxims:

1. *Maxim of Quantity*: Make your contribution as informative as required (for the current purposes of the conversation), but do not make your contribution more informative than required.

2. *Maxim of Quality*: Do not say things you believe to be false or for which you lack adequate evidence.

3. *Maxim of Relevance*: Be relevant, i.e. say things related to the current topic of the conversation.

4. *Maxim of Manner*: Avoid obscurity of expression; avoid ambiguity; be brief (avoid using too many words); be orderly.

These maxims should not be regarded as sociological generalizations, but rather as presumptions made in the course of strategic inference (Bach, 2006). The presumption that the speaker communicates according to Grice's maxims, enables the hearer to derive more information than is actually contained in the utterance. This is called a *conversational implicature*. For example, by assuming that the sentence "The shop will be open tomorrow at 9:00" obeys the maxim of relevance, the person wanting to enter the shop may derive that the shop is not open now, because otherwise that piece of information would be irrelevant to the current situation.

Another example of a conversational implicature is the interpretation of the sentence "John has three children" as John has *exactly* three children. By only regarding its literal meaning, the sentence "John has three children" would leave open the possibility that John has four children or more. By assuming that the speaker has obeyed the maxim of quantity, the hearer can derive that the speaker's contribution is as informative as required which excludes the possibility of John having four or more children.

As will appear in Chapter 4, conversational implicatures can be usefully applied to recognize semantic interoperability problems.

2.2.2 Agent Communication Languages

An agent communication language provides a number of speech acts that allows agents to flexibly interact. Much research has focused on the standardization of an ACL. Of particular concern has been how their formal semantics is to be specified, and how the formal semantics is to be linked to a theory of agency (Dignum, 2000). Two well-known agent communication languages are KQML (Finin et al., 1994) and FIPA ACL (FIPA, 2002a). The two languages are similar in syntax and semantics. An example of a KQML message is given below.

```
(tell
       :sender    Ag-1
       :receiver  Ag-2
       :language  prolog
       :ontology  travel
       :content   price(hotel23,100))
```

The first line in the message is the *performative*, which specifies the illocutionary force, in this case *tell*. Other examples of performatives in KQML are *advertise*, *ask-if* and *sorry*. The last line describes the actual content of the message. The distinction between performative and content corresponds to the elementary form of a speech act, which, as has been argued, can be described as *F(p)*. The other lines in a KQML message specify the sender and the receiver, the language that is used to describe the content and the ontology that underlies it.

Much effort has been invested to specify formal semantics of the performatives of an ACL. Formal semantics of a performative are specified according to the speech act philosophy, i.e. by stating their *effects*. In case of KQML and FIPA ACL, a speech act is assumed to affect the mental states of the participating agents. It thus embraces the idea of *cognitive agents*. In the above example, the semantics of the *tell* performative specifies that after the speech act is finished, Ag-2 knows that Ag-1 believes that *price(hotel23,100)* is true.

Usually, agent communication involves more than the exchange of just one speech act. *Communication protocols* are used to govern the exchange of a series of speech acts. For example, they may be used to specify which agents are allowed to perform which speech acts at which moments. Typically, an agent communication protocol leaves the agent some freedom as to which particular message it chooses to exchange. In other words, the protocol is non-deterministic. This non-determinism is to be resolved by the agent's *communication strategy*, which makes a decision when different options are available.

By standardizing an ACL, the agents are enabled to understand the *structure* of the exchanged messages and to understand the *performatives*. To achieve semantic interoperability, the agents must also be capable of understanding the *content* of a message, which is not directly supported by the ACL. In fact, an ACL should be viewed as wrapping content in a *structure* that can be understood by agents (Huhns and Stephens, 1999). To understand the content itself, the receiver must understand the language and the ontology.

Most effort on agent communication is spent on developing ACL's. This is not surprising, as this aspect provides a groundbreaking field for agent researchers, whereas languages for knowledge representation and ontologies have been studied in the AI community for decades. It is commonly believed that some knowledge representation language will eventually become standard for describing content in ACL messages. For example, KIF (Ginsberg, 1991) has been put forward as a standard language for use in KQML messages.

Likewise, it is thought that one or a few ontologies will become standardized and ubiquitously known among agents. Both KQML and FIPA ACL have reserved a field for an identifier of the ontology of the message's content. Most researchers assume that the sender and receiver use the same ontology. In this case, the ontology identifier serves to assure the receiver of this fact. It enables the receiving agent to be certain that it uses the correct ontology to interpret the content. As will be argued, this idea is based on misplaced optimism rather than profound insight.

Inherent to the nature of ontologies and multi-agent systems is the occurrence of ontologies that are *not* shared. The lack of shared ontologies in MAS's is as much a restraining factor for semantic interoperability as the lack of a standardized ACL. To get a better idea of how ontology problems arise in MAS's, we will discuss the (long) history of ontologies and argue how they are used in modern information systems. This is the topic of the next section.

2.3 Ontologies

Whereas ontology has been the territory of philosophy for a very long time, the term has now also become widespread in the information systems (IS) community. Although there seems to be consensus in the IS community that their usage of the word "ontology" differs from the philosophical usage, no consensus has yet been reached on an alternative definition. Recent articles on IS have brought about an abundance of definitions for the term "ontology", e.g. (Gruber, 1993a; Wielinga and Schreiber, 1993; Guarino, 1998; Sowa, 2000). This, in turn, has generated a large number of papers that aimed at resolving the terminological confusion, e.g. (Guarino, 1997; Smith and Welty, 2001; Zúñiga, 2001; Øhrstrøm et al., 2005). Many of these papers, again, proposed a new definition of "ontology".

We do not intend to resolve this terminological confusion. Rather, we will cover some characteristics which are commonly ascribed to ontologies in IS and discuss some issues of disagreement. In particular, we will characterize the role of ontologies in open multi-agent systems, which is needed to gain a better understanding of the central problems dealt with in this thesis.

Before the topic of ontology in IS is discussed, some attention will be paid to its philosophical heritage.

2.3.1 Ontology in Traditional Philosophy

The word "ontology" (or ontologia) literally means *the study of being*. It is composed of the two Greek words ὄντος (meaning *being*) and λογία (meaning *the study of*). As reported by Øhrstrøm et al. (2005), the word was first used in 1606 by a German philosopher called Jacob Lorhard in his book *Ogdoas scholastica*. However, ontology as a discipline is much older and can be traced back to the ancient Greeks. With the following words, the Greek philosopher Aristotle (384-322 BC.) stated the rationale of this discipline:

> THERE is a science which investigates being as being and the attributes which belong to this in virtue of its own nature. Now this is not the same

> as any of the so-called special sciences; for none of these others treats universally of being as being. They cut off a part of being and investigate the attribute of this part; this is what the mathematical sciences for instance do. Now since we are seeking the first principles and the highest causes, clearly there must be some thing to which these belong in virtue of its own nature.
>
> (Aristotle, 384-322 BC, Metaphysics, book IV)

To understand this fragment, it should be borne in mind that Aristotle regarded *being* as an aspect from which things can be studied. Just as there is a special science which investigates beings from the aspect of movement and change (the science presently known as physics), Aristotle argues that there is a science which investigates beings from the aspect of being. He called this science *first philosophy*[1], as it deals with the "first principles and highest causes" of beings. In fact, it deals with those attributes of beings that make them beings (in virtue of their own nature).

A substantial part of the study of being as being is preoccupied with a system of categories of things that exist. In his treatise *categories*, Aristotle enumerates the following ten categories into which entities in the world divide: Substance, Quantity, Quality, Relation, Place, Date, Posture, State, Action, Passion. Throughout the centuries, philosophers have proposed numerous other category systems that make different distinctions on different dimensions. Instead of giving an overview of these category systems, we will outline the different positions that philosophers have had on the *aims* of a category system.

Aristotle intended his category system to have objective validity in the world. This approach is called *categorical realism* (Carr, 1987), or *Aristotelian essentialism*. According to categorical realism, some category systems are better than others, simply because they match up better with the *real structure of the world*. Categories are defined by stating their *essential* characteristics, i.e. those characteristics that an entity *must* have in order to belong to that category. A member of a specific category may also possess other characteristics that are not required to establish its membership nor preclude its membership. These are called *accidental* characteristics. The position of Aristotle is in line with that of his contemporary philosophers, such as Plato (427-347 BC.), who used the famous metaphor that science should *carve nature at its joints* (Plato, 427-347 BC, 265e1-3). According to Plato, there is a perfect realm, which is populated by *ideas* that have eternal existence. What appears to our senses is only an erroneous and imperfect reflection of these *ideas*. In Plato's terminology, the essence of a category is the *idea*. A good category system contains the *ideas* itself, and is not misguided by the imperfect reflections that appear to our senses. Because the category system is a description of the perfect and eternal realm, it has objective and eternal validity, which is independent of the eye of the observer.

Other philosophers have rejected this robustly realist approach to categories. One of them is John Locke (1632-1704) who held that that there are many ways

[1] A better known term for "first philosophy" is "metaphysics" which was coined by a first century editor who assembled the works of Aristotle.

to classify the world, each of which may be useful, depending on one's aims and purposes. In the following text, he motivates this position:

> That every particular thing should have a name for itself is impossible. [...] It is beyond the power of human capacity to frame and retain distinct ideas of all the particular things we meet with: every bird and beast men saw; every tree and plant that affected the senses, could not find a place in the most capacious understanding. [...] If it were possible, it would yet be useless; because it would not serve to the chief end of language. Men would in vain heap up names of particular things, that would not serve them to communicate their thoughts. Men learn names, and use them in talk with others, only that they may be understood: which is then only done when, by use or consent. [...] yet a distinct name for every particular thing would not be of any great use for the improvement of knowledge: which, though founded in particular things, enlarges itself by general views.
>
> (Locke, 1689)

John Locke argues that particular things are all things that exist, and that it would be impossible (or undesirable) to give a name to everything that exists. He gives three reasons for this. The first reason is that so many particular things exist in the world, that it would be impossible to store and process names for them all. The last two reasons are more pragmatic. He argues that, even if it were possible, such a large naming scheme would be useless for communication. Because it is too large to make everyone acquainted with the naming convention, it would not contribute to mutual understanding. Furthermore, he argues that knowledge should not be represented at a too fine-grained level (the level of particular things), but that knowledge "enlarges itself by general views". Locke uses this as a justification for the use of general terms (or categories) which do not have a direct correspondence to things that exist, but are merely *useful*. Locke's position is known as *conceptual relativism* or *conventionalism*. Conceptual relativists deny that any system of categories is correct in the sense of corresponding to the structure of the world (as in Aristotelian essentialism). There are simply no categories that "carve nature at its joints". When two category systems conflict, there is no question as to which is wrong; it is a matter of convenience which one we choose.

Relating these ancient ideas to our discussion on multi-agent systems, the dichotomy can still be regarded as relevant when developing ontologies for software agents. According to Aristotle, only the *environment* of the agents should be taken into account while devising an ontology, simply because that is what the ontology is about. Locke's position would be that also the agents themselves are relevant in this endeavor, as the ontology constitutes the world view of an agent. They *use* the ontology for some *pragmatic* purpose.

2.3.2 Ontology in Modern Philosophy

Philosophical progress does not necessarily lead to solutions, but rather to detailed exposure of different positions. This also holds for ontology, where the essentialism-conventionalism debate has remained lively until this day. Nevertheless, scientific triumphs throughout the centuries have more than once shed new light on these issues.

Newtonian physics has involved the application of precise and almost perfect mathematics to the material world. To predict the motion of an object in a gravitational field, color and shape can be regarded as irrelevant, but it is essential to know the mass of the object. An Aristotelian essentialist would say that mass forms the *essence* of the object, and that color and shape are its *accidental* properties. Also the periodic table of chemical elements, developed in the 19th century, appears to reveal a categorization that "matches the real structure of the world", in the Aristotelian sense. Also, many modern-day physicians are pursuing a theory of everything, i.e. one coherent theory that fully explains all physical phenomena. If such a theory were to be found, it would be hard to deny that the underlying ontology of this theory would not reveal something about the "real structure of the world".

In other sciences, such as biology, essentialism has not been fruitful. Before Darwin (1808-1882), biology was essentialistic in nature. Biologists were preoccupied with the search for an eternal classification scheme of animals. This involved giving clear-cut definitions of animal species to distinguish between members of a species, their imperfect variations, and their non-members. After Darwin proposed his theory of evolution, essentialism quickly became less popular in biology. The evolution theory points out that an eternal animal classification scheme simply cannot exist, because animals evolve over time. Furthermore, the boundary between two animal species is in the eye of the observer[2], rather than an intrinsic and immutable property. In modern biology, it is no longer pretended that animal classification schemes are timeless and have objective validity in the Aristotelian sense. Scientists admit that multiple classification schemes are possible depending on one's perspective and purposes. In modern biology, several different animal classification schemes coexist which are used for different purposes. The most widely known taxonomy stems from Linnaeus (1707-1778) which contains the animal kingdom and the plant kingdom as the two top categories. Modern versions of this taxonomy have added three extra top categories: Fungi, Bacteria and Protists. Recently, a taxonomy has been proposed by Carl Woese, which is based on genetic relationships, rather than morphological similarities (Woese and Fox, 1977). His taxonomy contains 23 main divisions, which for the largest part are occupied by unicellular beings. Although Woese's taxonomy is popular among microbiologists, many botanists and zoologists prefer the old classification scheme.

[2]Since Darwin, many definitions of "species" have been proposed. The most famous one is by Ernst Mayr whose definition is based on the capability to sexually reproduce. However, this leaves the term undefined for species that reproduce asexually. No consensus has been reached yet on an alternative definition.

Also among philosophers, the disagreements in the essentialism-conventionalism debate are considerable. Grossmann (1983) characterized his category system as an attempt "to bring Aristotle's *Categories* up-to-date". His system contains eight highest categories: Individuals, Properties, Relations, Classes, Structures, Quantifiers, Facts, Negation. It appears that much of the "update" of Aristotle's system that Grossman considered necessary has been inspired by modern predicate logic. A recent attempt that is clearly in the Aristotelian essentialist tradition has been performed by Chisholm (1996) who presented his category system as being "about the ultimate categories of reality". The debate on essentialism was intensified by Putnam (1975) when he presented his "twin earth argument" to demonstrate that the real essence of water is to be found in its chemical formula.

Aristotelian essentialism also encounters opposition. In the following text fragment, Daniel Dennett ridicules this approach by regarding the question "what is an island?":

> Suppose you see an island half a mile off shore at high tide. If you can walk to it at low tide without getting your feet wet, is it still an island? If you build a bridge to it, does it cease to be an island? What if you build a solid causeway? If you cut a canal across a peninsula (like the Cape Cod Canal), do you turn it into an island? What if a hurricane does the excavation work? This sort of inquiry is familiar to philosophers. It is the Socratic activity of definition-mongering or essence-hunting: looking for the "necessary and sufficient" conditions of being-an-X.
>
> (Dennett, 1995, p.95)

Apparently, Dennett prefers conceptual relativism to essentialism, at least when it comes to islands.

Ontological commitment

One of the most influential philosophers of the twentieth century is Willard van Orman Quine (1908-2000). His work on ontology is neutral with respect to whether categories have real existence or not. In his essay "On what there is" (Quine, 1953), the typical ontology question "What is there?" is disregarded. Instead, the question "When are we committed to the existence of certain entities?" is addressed. The theory that results, i.e. a theory of *ontological commitment*, may fairly be called the received view. Quine starts his essay by regarding the problem of nonbeing: something that does not exist must in some sense exist, otherwise what is it that does not exist? He proposes to solve this problem using Russell's theory of description. Instead of stating "Pegasus does not exist", which would commit one to the existence of Pegasus, one should paraphrase this as "nothing Pegasizes" (where "Pegasizes" is that what one does when one is a Pegasus). The paraphrase of the sentence does not commit one to the existence of Pegasus. According to Quine, because "Pegasize" is not a singular term, it does not involve an ontological commitment. When one states that "something Pegasizes", one is committed to the existence of Pegasus, because a *value* is needed for the *bound variable* to make the statement true. This has led Quine to his famous dictum: "to be is to be the value of a bound variable".

Quine's theory of ontological commitment enables one to move from sentences that are accepted as true to claims regarding what exists in the world. It does *not* say anything about what constitutes a *good* ontology. At the end of "On what there is", Quine devotes a few words to this issue: "the question what ontology actually to adopt still stands open, and the obvious counsel is tolerance and an experimental spirit". It seems that Quine favors a conceptual relativistic approach to ontology.

2.3.3 Ontologies in Information Systems

It is widely acknowledged that ontology means something different in information systems than in philosophy. However, there is little agreement so far on what these differences are. We will start by describing one obvious difference, which has not been subject to controversy. In philosophy, the word "ontology" refers both to the discipline (the study of what exists) and to the product of such a study (an account of what exists). In the context of information systems, the word is only used in the latter sense. In fact, an ontology in IS refers to something even more specific than that. Whereas a philosophical account of what exists usually contains a theory on what it means to exist and how to settle existential disputes, such examinations are absent in an IS ontology. An ontology in IS confines itself to a category system: a description of categories into which entities can be said to divide.

Apart from this very basic conception, the views on ontologies in IS are highly divergent. Of course, the inability of researchers to properly define their area of study is not unique to people working on ontologies. Agent researchers are uncomfortable with the question "what is an agent?", and artificial intelligence researchers with the question "what is intelligence?". As serious as it might seem, the lack of definitions does not necessarily hamper progress in these fields. It may be just as harmless as the difficulties one encounters when trying to give a definition of an island (see page 21). However, the meanings attached to the word "ontology" in the IS community have become so diverse and contradictory, that it created a terminological impasse. Ironically enough, the study area of these researchers concerns the very topic of avoiding terminological confusion itself.

To deal with this problem, we will adopt a rather general definition of an ontology and stay away from these controversies as much as possible. For those controversies regarding the word "ontology" that may cause misunderstandings, we will state our position.

The following definition of an ontology is adopted in this thesis (Gruber, 1993a):

> *An ontology is an explicit specification of a conceptualization.*

This definition is probably the most popular definition of an ontology. It marked the start of an era in which ontologies became a hot issue on the IS research agenda. At the time, ontologies were used to support *knowledge sharing and reuse* (Neches et al., 1991). They were envisioned as key components to realize the goals of the Knowledge Sharing Effort (KSE, 1994), an initiative to enable libraries of reusable knowledge components that can be invoked over networks. Using these libraries, developers would no longer have to start their knowledge bases from scratch. Instead, they could assemble a number of reusable components (such as reasoning modules and ontologies), and focus on the specialized task which would be new to their specific system. We elaborate on the role of ontologies in supporting knowledge sharing and reuse below.

Because many different knowledge bases require the same pieces of domain knowledge, an ontology can be said to enable reuse of knowledge. For example, many knowledge bases need to store information about time. *Time* involves notions such as *TemporalEntity*, *Interval*, *begins*, *ends*, etc. Once this kind of knowledge is represented in an ontology of time, e.g. (Hobbs and Pan, 2004), this piece of domain knowledge can be reused in another knowledge base that also needs to represent time.

Ontologies can be said to enable knowledge sharing as it provides a way to specify content-specific agreements. Once the ground rules for modeling a domain are specified in an ontology, multiple parties can agree to adhere to that specification. If they do that, it would enable them to share knowledge, as their knowledge bases would all have the same underlying structure. For example, two components that share the same ontology about time have no difficulties in exchanging knowledge about something taking place in the afternoon. The time ontology prescribes a uniform way to model the concept afternoon, namely as an *Interval* which *begins* and *ends* at certain moments. Because both components have agreed to adhere to this specification, the description of afternoon given by one component can be readily processed by another component.

Ontologies are useful for knowledge sharing and reuse because their application involves a distinction between more or less changeless knowledge expressed in the ontology and other types of knowledge that are more changeable. This distinction was certainly not new at the time. In an influential paper (Brachman et al., 1983), the authors argue that a knowledge base should consist of two parts: a TBox and an ABox. The TBox stores terms and their definitions (like an ontology), and the ABox stores sentences constructed using these terms. The TBox represents general knowledge about a problem domain, which is not subject to changes. The ABox represents problem-specific knowledge that is subject to occasional or even continuous change (Nardi and Brachman, 2003). Originally, the authors did not apply this distinction with the purpose of enabling knowledge sharing and reuse. Their goal was to improve computational efficiency of reasoning. Furthermore, they considered it sensible from a modeler's perspective to first describe concepts and their analytic interrelations, and then describe contingent facts about what is true of the world. Nowadays, the TBox is advocated as a proper way to implement ontologies (Baader et al., 2004). To some extent, the lines of research on ontologies and on TBoxes have converged.

As argued before, there is presently no uniform conception in the research community of what an ontology is. One reason for this is that over the last decade the range of ontology applications has become much wider. Besides knowledge sharing and reuse, ontologies are used for search, reliability, communication between people, specification, maintenance and knowledge acquisition (Jasper and Uschold, 1999). Furthermore, ontologies have been embraced by the semantic web community as one of the key components (Berners-Lee et al., 2001). As a result, much of the current ontology research is conducted from an internet perspective. Below, we will clarify how ontologies are construed in this thesis by stating our position on a few controversial issues.

Specification of naming

A debatable issue is whether an ontology should be seen only as a specification of the conceptualization itself, or that it also specifies which *names* should be attached to the concepts in the conceptualization. Clearly, the first position is in line with the philosophical reading. If someone translates Aristotle's ontology from Greek to English, this is not considered as changing anything to the ontology. Some authors, e.g. Steels (1998a), also treat ontologies in AI in this way, i.e. as independent of the names that are used to describe the conceptualization.

However, as noted by Guarino (1998), most researchers in the IS community view an ontology as an engineering artifact, constituted by a specific *vocabulary* and a set of explicit assumptions regarding the intended meaning of the words in the vocabulary. In this reading, an ontology is dependent on the *names* that are used to describe the conceptualization. Translating an ontology from one language to another would make a different ontology. This position makes sense from an IS point of view. To enable knowledge sharing, different components should not only use the same conceptualization but also attach the same names to the concepts.

This is also the position we adopt in this thesis. An ontology in philosophy corresponds to what we will call a *conceptualization*. Our use of the word ontology denotes a *specification of a conceptualization*, i.e. including the names that are used for describing the concepts.

Shared or non-shared ontologies

Many authors have stressed that ontologies should be shared. The original definition (Gruber, 1993a) does not explicitly mention that ontologies are shared. However, Gruber has given another definition in the SRKB mailing list, which defines an ontology as an *agreement about a shared conceptualization* (reported by Uschold and Grüninger (1996)). Studer, Benjamins and Fensel (1998) propose a modified version of Gruber's definition, which states that an ontology is a formal, explicit specification of a shared conceptualization. They thereby stress that an ontology captures consensual knowledge, which is not private to an individual, but accepted by a group. According to the authors, sharedness is an intrinsic property of an ontology. Consequently, they consider the description "a shared ontology" to be a pleonasm (Benjamins and Fensel, 1998).

Indeed, an ontology is most commonly used to specify a shared conceptualization. However, in the context of multi-agent systems, the ontologies of the agents are often not shared. We would still like to speak of these as ontologies. We therefore do not regard sharedness as an *intrinsic* property of ontologies, but rather as a *useful* property. Besides, the property of being shared in a group is rather ill defined. How large should the group be? Does a group of size one qualify, in which case the agent shares the ontology with itself? Suppose that a group of agents share the same ontology, and that another agent enters the group which does not know that ontology. Does the ontology suddenly stop to be an ontology as it is not shared by the new coming agent?

2.3.4 Task-Dependency of IS Ontology

The philosophical controversy regarding Aristotelian essentialism and conceptual relativism (described in Section 2.3.1) has also manifested itself in information systems. We will first describe some approaches that are influenced by the essentialist approach. After that, we will consider some relativistic approaches.

All-purpose ontologies

Recently, an increased interest in so-called top-level or upper-level ontologies, has led many researchers towards the Aristotelian essentialist paradigm. A top-level ontology describes very general concepts that are the same for every domain and are not dependent on task and purpose. The aim is to use this ontology as a ground for more specialized (task-specific) ontologies. This already creates some degree of interoperability between different components that are based on the same upper ontology. Examples of top-level ontologies are the upper-level ontology by Sowa (2000), SUMO (Niles and Pease, 2001) and BFO (Grenon et al., 2004).

A very ambitious approach to develop an all-purpose ontology is pursued by the Cyc project (Lenat and Guha, 1990). The Cyc ontology aims at formalizing a vast quantity of background knowledge, which is not tied to a specific domain. The knowledge in the Cyc ontology is supposed to be independent of context and to be usable for all purposes. It has proven extremely difficult to implement a *complete* set of rules and facts that represent everything there is to know in every possible situation. The Cyc approach of dealing with incompleteness has always been: try harder and add more rules. In a recent article (Matuszek et al., 2006), the Cyc development team is reported to have spent approximately 900 person-years of labor in developing a knowledge base containing more than 2.2 million assertions and more than 250,000 terms.

The idea of an all-purpose ontology has been severely criticized by several authors. According to Smith (2003), it is not feasible to build a single ontology (not even a top-level ontology) for a broad population of information systems communities. In the dynamic world of current IS practice, the requirements that are placed on information systems change at such a rapid rate, that it is not possible to keep a single ontology up to date. Another source of criticism against all-purpose ontologies like Cyc, stems from Dreyfus (1992). He argues that an all-purpose ontology hopelessly aims to be independent of any context including every day skills and cultural savoir-faire. Therefore, all these types of knowledge are to be included *as objective facts* in the ontology, which is simply too much to be feasible. As an example, he considers *a gift*. A complete representation of *a gift* in an all-purpose ontology must also include the circumstances in which gift-giving is considered appropriate. According to Dreyfus, this would not only be very hard to realize, but also unnecessary. There is no need to spell out the supposed objective features of gift-giving. Members of a culture who have acquired the necessary social skills have absolutely no difficulties in giving appropriate gifts in the appropriate circumstances. This is not because they have an exhaustive list of gift-giving situations and their features, which they use to objectively recognize a situation as a gift-giving situation. Rather, they have the *practical skills* which enable them to act appropriately. Therefore, an ontology does not stand on its own, but is dependent on context. A proper understanding of an ontology demands practical information, cultural savoir-faire, and a purpose for which it is used.

Task-dependent ontologies

These kinds of arguments have supported the idea that ontologies in IS are task-dependent, in accordance with the philosophical tradition of conceptual relativism. Many authors have stressed the task-dependency of ontologies, e.g. (Bylander and Chandrasekaran, 1987; van Heijst et al., 1997). A motivation for this position is wonderfully exposed by Davis et al. (1993) who address the question: What is a knowledge representation? According to the authors, it is a misconception that a knowledge representation (KR) should include as much information as possible. In the following quote, they defend this position:

> [Ontological commitments] are in effect a strong pair of glasses that determine what we can see, bringing some part of the world into sharp focus, at the expense of blurring other parts. These commitments and their focusing/blurring effect are not an incidental side effect of a representation choice; they are of the essence: A KR *is* a set of ontological commitments. It is *unavoidably* so because of the inevitable imperfections of representations. It is *usefully* so because judicious selection of commitments provides the opportunity to focus attention on aspects of the world we believe to be relevant.
>
> (Davis et al., 1993)

It is argued that reality contains too much detail to be represented with perfect fidelity. Simplifications are therefore unavoidable. Which simplifications are made in the knowledge representation is determined by the ontological commitments that are described by the ontology. Besides being unavoidable, simplifications are also very useful for a KR as they provide a way to deal with the overwhelming complexity of the real world. Aspects of reality that are relevant for a certain task should be clearly represented in the KR. Aspects that are irrelevant for the task should be left out, in order not to burden the KR with useless details. This idea is also expressed by the proposition: "A perfect copy is generally a useless model" (Harrenstein, 2004).

The view that ontologies are task-dependent is supported by an important finding from AI research, namely that the success of a problem-solving method is highly dependent on the representation of the problem (Fink, 2003). A common way to represent an AI-problem is as a search for a sequence of actions that lead from an initial state to a goal state. There are different ways to formulate the same problem in a state-based representation. Two states may be very near to each other in one representation, and very far from each other in another representation. Consequently, a goal state may be very easy to reach in one representation, but almost impossible to reach in another representation.

Consider an example taken from Russel and Norvig (2003, p.66), regarding the classic 8-queens problem. The goal is to place eight queens on a chessboard such that no queen attacks any other queen[3]. The states in this problem correspond to *chess-positions* and the actions to *moves*. The initial chess-position is the empty chessboard, and the goal chess-position is a board with eight queens with no queen attacking any other. The moves and chess-positions that are processed by the search algorithm can be defined in an ontology. One possibility for this is as follows (ontology A):

- A *chess-position* is any arrangement of 0 to 8 queens on the board.
- A *move* is to add a queen to an empty square.

Using ontology A, the number of chess-positions that can be investigated by the search algorithm equals $\frac{64!}{(64-8)!} \approx 3\times10^{14}$. In other words, this representation is highly inefficient for solving the 8-queens problem. Another possible representation for the 8-queens problem adopts the following ontology (ontology B):

- A *chess-position* is an arrangement of n queens ($0 \leq n \leq 8$), one per column, in the leftmost n columns, with no queen attacking any other queen.
- A *move* is to add a queen to the leftmost empty column, such that it is not attacked by any other queen.

Using ontology B, the number of chess-positions that can be investigated is only 2057, which makes this representation much more efficient for solving the 8-queens problem than ontology A.

Whereas ontology B is more suitable for the task of solving the 8-queens problem, it is not as widely applicable as ontology A. For example, consider the 8-queen-duel-problem, which is to place eight queens on the board such that every queen is attacked by exactly one other queen. Ontology B is not suitable for this problem because its definitions of *chess-position* and *move* are too narrow. Ontology A, on the other hand, can be used for this problem, although it would be very inefficient. To solve the 8-queen-duel-problem efficiently, one should devise a third ontology which is tailored to that particular problem.

[3] A queen attacks any other queen which is in the same row, column or diagonal.

This example shows that computational efficiency might demand a task-specific ontology, instead of an ontology that serves a wide range of purposes. One might object to this conclusion by arguing that ontology A, together with an algorithm that limits the search space, can very well be used for solving the 8-queens problem. The objection would hold that efficiency measures should be encoded as procedural knowledge in the search algorithm but not as domain knowledge in the ontology. This appears to achieve the best of both worlds: a general ontology and efficiency of reasoning. We will argue below why this objection is misguided. Suppose that, as suggested by the objection, ontology A would be adopted as the underlying domain model, and that the search algorithm would only consider chess-positions of the type described in ontology B. In this scenario, the *chess-position* definition in the ontology would be more general than the chess-positions that are actually processed by the search algorithm. As a result of the attempt to separate the procedural knowledge on how to limit the search space from the domain knowledge about chess, the ontology no longer accurately describes what a chess-position is. However, the way that a chess-position is defined in ontology A is still not independent of tasks and purpose. For example, it cannot be applied to 9-queen problems, or general chess problems, or problems on a checkerboard with 10×10 squares. There seems to be no limit to how general a chess position should be defined in order to make it completely independent of task and purpose. This line of reasoning leads to an ontology that is so general, that it makes itself obsolete.

Although we cannot prove that an all-purpose ontology is unfeasible, we can at least say that there are strong arguments against it. Furthermore, since 2500 years of Aristotelian essentialism has not been able to come up with an objective task-independent ontology, it is unlikely that such an ontology will be available to the IS-community within the near future. Therefore, in this thesis, we will adhere to the position that an ontology is task-dependent.

2.3.5 Ontologies in Multi-agent Systems

As has been argued in Section 2.2.2, ontologies are embraced by the agent community to establish content-specific agreements between agents. Heterogeneous ontologies are highly problematic for the agents' interoperability. Despite considerable disagreements in the research community on what an ontology is, we have identified some important properties of ontologies in this section. Firstly, an ontology is task-dependent. Secondly, an ontology is an artifact constructed by humans. In the light of these properties, we will argue why heterogeneous ontologies are a natural phenomenon in multi-agent systems.

In multi-agent systems, there is usually a task division between agents. Typically, a complex task is divided into sub-tasks, each of which is fulfilled by a particular agent (Durfee, 1999). Bearing in mind that ontologies are task-dependent, it follows that such agents must use different ontologies that best suit the tasks they are designed to accomplish.

A second cause for heterogeneous ontologies is related with the way that MAS's are developed. As has been argued, a MAS may be developed and maintained by different teams that are not burdened with the responsibility of reaching agreement on common design decisions. Part of what is developed and maintained by these teams are the agents' ontologies. In such situations, it is highly unlikely that the different system developers will all come up with the same ontology. Especially in open multi-agent systems, where the agent-developers may be completely unknown to each other, the idea of all agents possessing the same ontology is inconceivable.

2.4 Ontology Reconciliation

The process of overcoming difficulties caused by heterogeneous ontologies is called ontology reconciliation. In this section, we will discuss techniques for ontology reconciliation, mainly from an agent perspective. Since many techniques have their origins in more conventional systems such as distributed databases and knowledge systems, we will adopt other perspectives when appropriate.

The problems that arise in agent systems with heterogeneous ontologies exhibit a strong resemblance to the problems that arise in a multilingual society. Also, the solutions that are applied to overcome these problems are, to a large extent, alike. To provide the reader with an intuitive understanding, we will illustrate each ontology reconciliation technique with an analogy from human society.

2.4.1 Ontology Integration

Ontology integration is the most straightforward way to solve interoperability problems in distributed systems. This approach aims at overcoming difficulties with heterogeneous ontologies by replacing them with one common ontology. The common ontology is obtained by integrating every distinct ontology in the system so that it satisfies the needs of all users. Usually, this process involves more than simply putting the distinct ontologies together. This is because of two reasons. Firstly, the ontologies may contain overlapping areas. The same concept in one ontology may be modeled differently in another ontology. In this case, the integration of two ontologies demands one uniform way to model the concept. Secondly, putting two ontologies together does not make explicit the relations between the different ontologies, as these are not described in any of the two ontologies. Therefore, for every two concepts that stem from different ontologies, an additional concept relation must be specified.

Ontology integration has its roots in the database community, where the ontology reconciliation problem is known as the schema integration problem(Batini et al., 1986). The most widely used schema integration technique resolves conflicts by modifying the initial database schemas, analogous to ontology integration. Because ontologies are more complex structures than database schemas, the techniques that are developed for schema integration require some revision to become applicable to ontology integration. However, the main idea underlying the two approaches is the same.

In the IS community, ontology integration has become a popular technique to deal with heterogeneous ontologies. The main reason for this is that it gets rid of all ontology related interoperability problems at once. The heterogeneous ontologies are simply no longer present. Ontology integration is one of the steps in the On-To-Knowledge Methodology (OTKM) (Sure et al., 2004). OTKM is a methodology for designing and maintaining ontology-based knowledge management applications. In the early modeling stages, the domain experts and potential users propose a set of *concept islands*, i.e. ontologies that model a subdomain of the corresponding domain. Later, these concept islands are integrated to form one uniform ontology. An alternative methodology for constructing shared domain ontologies has been proposed by Aschoff et al. (2004). The ONIONS methodology (Steve et al., 1997) is an example of the ontology integration approach applied to a medical domain.

Because the integration of ontologies is often a laborious task, tools have been developed which assist people in merging ontologies. Examples of such tools are Chimaera (McGuinness et al., 2000), Prompt (Noy and Musen, 2000) and FCA-Merge (Stumme and Maedche, 2001). These tools merge ontologies semi-automatically, i.e. they propose suggestions but leave the ultimate decisions to the system developer. An overview of semi-automatic tools for matching database schemas is given by Rahm and Bernstein (2001).

Despite its popularity in the IS community, it is unlikely that ontology integration provides a suitable solution for ontology reconciliation among agents. This claim is motivated by the following two arguments.

The first argument is basically the same as the argument against standardized all-purpose ontologies. One multi-purpose ontology, which perfectly suits the needs of all agents is not always realizable in practice. An agent that is specialized in a specific task needs an ontology that is tailored to its task. Therefore, agents with different tasks require different ontologies. Ontology integration forces an agent to make concessions in which world view to adopt which could be disadvantageous for its performance.

Our second argument concerns system development. As argued at the beginning of this chapter, the division of a system in autonomous agents enables a distributed development. System developers that do not even know each other can independently develop autonomous agents without worrying about how this fits in a larger architecture. This advantage does not hold when the system developers must reach consensus on one integrated ontology.

To compare ontology integration in MAS's with a human language policy, consider a suppressing regime that enforces one language upon its population. An example of such a policy can be found in the Spanish colonization of the Americas, whose primary aim was to eliminate the native American languages. Of course, the analogy of ontology integration with such a harsh language policy breaks down on a number of points. It is not used to demonstrate the "immorality" of ontology integration. However, the analogy *does* demonstrate that ontology integration is in conflict with the autonomy of agents. A strong dictator is needed to make every component use the same ontology. This is not necessarily a problem for distributed databases or for relatively simple information systems. For agents, however, auton-

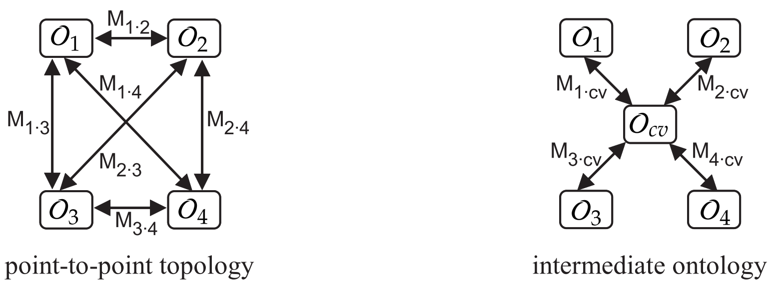


Figure 2.1: Different topologies for ontology alignment.

omy is fundamental and restricting the agent's freedom to adopt its own subjective world view is not acceptable.

2.4.2 Ontology Alignment

Ontology alignment allows agents to preserve their individual ontologies, leaving their autonomy unaffected. Communication is enabled by a set of pre-defined *mappings*, which specify the relations between the agents' ontologies. Similar to ontology integration, it is assumed that the work of creating ontology mappings is done by humans. Most ontology integration tools (discussed in the previous section) can also be used to create ontology mappings.

In the context of databases, this approach corresponds to global-as-view schema integration (Calí et al., 2001). In the global-as-view approach, the independently developed schemas persist and are integrated by constructing a global view, i.e. a virtual schema that makes the mappings between the individual schemas explicit.

Two different topologies to implement ontology mappings are sketched in Figure 2.1. The left-hand side of this figure shows point-to-point mappings. In the point-to-point topology, a mapping is defined for every pair of ontologies. The agents exploit these mappings to translate to and from each other's ontologies. Suppose, for example, that agent 1 (with ontology O_1) wishes to speak to agent 2 (with ontology O_2). Agent 1 first composes a message stated in its own ontology O_1. The mapping $M_{1\cdot 2}$ is used to translate the message from O_1 to O_2 to create a message that is understandable for agent 2.

The advantage of point-to-point mappings is that, for a pair of interacting agents, only one of them must know the ontology mappings for their ontologies to be aligned. In this way, one agent can be made interoperable with another agent, without having to implement any changes in the other agent. The disadvantage of the point-to-point topology is that many mappings have to be established in order to align the agents' ontologies. To reach complete interoperability in a system of n agents, $\frac{1}{2}n^2 - n$ mappings must be established (equal to the number of non-reflexive, non-directed arches in a fully connected graph).

Point-to-point ontology alignment is analogous to a country whose population is taught all the languages of its neighboring countries. An example of such a country is the Netherlands, where English, French and German is taught in schools. The advantage is that the Dutch people are prepared to communicate with their neighbors regardless of whether their neighbors also learn any foreign languages. A disadvantage is that it requires people to invest much time in learning foreign languages.

A way to reduce the number of mappings is to use an *intermediate ontology* or *interlingua* (Ciocoiu et al., 2001). This approach is illustrated at the right in Figure 2.1. The intermediate ontology indirectly aligns the agents' ontologies. Because the intermediate ontology is only used for communication purposes, we also refer to it as *communication vocabulary* (cv). Suppose, for example, that agent 1 (with ontology O_1) wishes to speak to agent 2 (with ontology O_2). Agent 1 first composes a message stated in its own ontology O_1; it then uses $M_{1\cdot cv}$ to translate the message to O_{cv}. Agent 2 receives this message and uses $M_{2\cdot cv}$ to translate the message from O_{cv} to O_2.

The advantage of using an intermediate ontology instead of point-to-point mappings, is that it drastically reduces the number of mappings needed. Using an intermediate ontology in a system of n agents only requires n mappings to be defined in order to establish complete interoperability. A disadvantage of this approach is that, among a pair of interacting agents, both agents must know a mapping from their own ontology to the intermediate ontology. Therefore, the approach only works when every agent cooperates.

A serious attempt to solve language misunderstandings by using an intermediate language, has been performed in Europe in the late nineteenth century. The project aimed at introducing one artificial language, namely Esperanto. The language was taught as widely as possible to citizens in different countries. Esperanto was not intended to replace the native languages of these countries. Rather, it would serve as an auxiliary language to enable communication between people from different countries. The introduction of this system demanded a relatively small effort by people: everybody should learn one extra language: Esperanto. Despite its initial popularity, the Esperanto movement did not succeed in achieving its goals. The main reason for this has been the unwillingness of people to cooperate with the project.

Regardless of whether a point-to-point or an intermediate topology is adopted, the use of ontology alignment in multi-agent systems suffers a major disadvantage: it presumes that the mappings can be established at design-time, before the agents start interacting. This assumption is not very realistic in multi-agent systems because of two reasons. Firstly, in an *open* system, it is not known beforehand which agents will constitute the system. It is therefore impossible to state at design-time which ontology mappings are needed at agent interaction time. Secondly, it is not clear when design-time stops and interaction time starts. As many development methodologies point out (Schreiber et al., 2000), system development should proceed in a number of iterative cycles. According to the so-called *development cycle*, improvements are implemented in each cycle based on the findings from the previous cycle. This means that agents, and their ontologies in particular, also evolve at interaction time (Noy and Klein, 2004). Ontology alignment does not provide the flexibility to deal with changing ontologies, as it assumes that one moment exists after which the agents' ontologies no longer change.

2.4.3 Ontology Mediation Services

A more flexible approach to ontology reconciliation is based on *mediation services* (Wiederhold and Genesereth, 1997). In the agent literature, such services are usually referred to as *ontology agents*. According to the FIPA Ontology Service Specification (FIPA, 2000), an ontology agent is expected to facilitate agent communication by registering ontologies, generating mappings between ontologies, and by making translations between ontologies. In other words, an ontology agent provides a central point which can be consulted by agents with communication problems. This approach opens the possibility to reconcile heterogeneous ontologies at agent interaction time, thus being more flexible than ontology alignment.

Mediation services are analogous to interpreters that translate between human languages. The European parliament is an example of an organization that has chosen this approach to solve their language problems. This organization contains people from more than ten different language communities. The European parliament currently spends over 150 million Euro each year to hire interpreters that translate documents and speeches into all the different languages. The reason why this expensive solution has been chosen is clear. The parties involved could not reach agreement on a common language for the organization (i.e. ontology integration was too far-reaching). Furthermore, the interpreter solution could be adopted without asking the members of the parliament to follow time consuming language courses (cf. ontology alignment demanded too many prior investments).

The KRAFT architecture (Preece et al., 2000) is an example of the ontology mediation approach. In this architecture, knowledge sources, knowledge fusion entities and users are all represented by independent agents. The heterogeneity of the distributed knowledge sources is solved by specialized agents that, among other things, resolve ontology mismatches. OBSERVER (Mena et al., 2000) is another example of a distributed architecture that is based on the ontology mediation approach. This approach makes use of so-called *Interontology Relationships Managers* to process queries in multi-ontology environments.

Different approaches for ontology agents or mediation services differ greatly in how these components are positioned and implemented. Invariable in these approaches is the idea that a mediator is capable of translating between the different ontologies in the system. This establishes that other components are relieved from the necessity of having this capability themselves. The ontology mediation approach can therefore be viewed as a *centralized solution* to semantic interoperability problems.

Although the mediation approach might appear as an attractive solution for interoperability problems in multi-agent systems, it does not solve the problem of how to establish these mappings themselves. Basically, it relocates the problem from the individual agents to the ontology agent. One might argue that an advantage of this relocation is that it is less work to establish these mappings at one location (at the ontology agent), than to establish ontology mappings for every agent. However, the contrary may also hold because at the time the mappings are created, it is not known who will communicate with whom about what. It is therefore also unknown which mappings are actually needed and which are not, which may lead to the establishment of superfluous mappings.

An additional disadvantage of this approach is that it deprives agents of control over establishment and use of ontology mappings. This need not be a problem in systems with a carefully designed structure where every agent trusts the ontology agent and knows how to reach it. However, in open heterogeneous multi-agent systems, where the need for semantic integration is most pressing, such a well-defined structure is usually absent. It is not self-evident that agents are willing to delegate the task of finding ontology mappings to this ontology agent. An ontology translation that is considered sufficiently accurate by the ontology agent, may be regarded as unacceptable by the individual agents. Furthermore, in open and heterogeneous systems it cannot be guaranteed that the ontology agent does not have malicious intentions. The ontology agent might have an agenda of its own, and deliberately make the wrong translations between ontologies. Basically, these problems arise because a *centralized* solution for ontology reconciliation is applied to an inherently *decentralized* multi-agent system.

2.4.4 Ontology Negotiation

Ontology negotiation (ON) tackles the ontology reconciliation problem in a fully decentralized way. This approach can best be understood by observing how two people communicate that do not speak each other's language and do not have an interpreter at their disposal. For example, consider an English customer who wishes to order a cup of coffee in a cafe in Japan. First, he speaks the word "coffee" a number of times, but finds out that he is not understood by the waitress. Eventually, he points to a cup of coffee that is standing at another table. The waitress understands him, their conversation finishes, and the customer gets his cup of coffee. The next time the customer visits the cafe to order a cup of coffee, the dialogue with the waitress proceeds more fluently. The customer speaks the word "coffee", the waitress remembers from the last time what he meant by this word, and brings him the cup of coffee. This conversation is made possible because the waitress and the customer share one word "coffee", which is sufficient for them to cooperate on the task of ordering coffee. If the customer would want to order something else, his words would be incomprehensible again for the waitress, and he would have to use pointing again to get his message across. Again, this would give rise to a larger shared vocabulary, which would enable them to cooperate on a wider range of tasks.

This analogy illustrates that ontology negotiation does not demand any preparations by agents before they can communicate. Unlike ontology alignment, no pre-defined mappings are required. Unlike the ontology mediator approach, no intervention of third agents is required. The agents solve the problem between themselves, i.e. the ontology reconciliation problem is tackled decentralized. Another remarkable difference is that this solution does not aim at solving all ontology mismatches at once. Instead, an *incremental* solution is established which deals with ontology problems when they arise. In this way, only those mappings are established which are actually needed. Ontology negotiation thus overcomes the two main objections raised in the previous section against the mediator approach.

Whereas ON is a promising approach, it is also a very ambitious approach. According to Uschold and Grüninger (2002), ontology negotiation can be regarded as *the Holy Grail of semantic integration*. This is mainly because the technique requires agents to be capable of both *detecting* ontology mismatches and of *resolving* them. Whereas other approaches for ontology reconciliation keep the possibilities for human involvement open, detecting and resolving mismatches must proceed fully automatically in ON. Another complication in ON is that the reconciliation process must proceed in a distributed way. The previously described approaches for ontology reconciliation proceeded from a god's eye view over the heterogeneous ontologies. In ON, such a god's eye view does not exist, i.e. two agents that wish to align their ontologies do not have access to each other's ontologies. Agents cannot "look inside each other's head". Because the area is also relatively new, much work remains to be done.

The term *ontology negotiation* was coined by Bailin and Truszkowski (2002). In their paper, the authors present a communication mechanism that enables agents to exchange parts of their ontology in a pattern of successive clarifications. The approach has been applied in the context of scientific archives. Many approaches that investigate ontology reconciliation from an agent perspective, have a strong focus on the automatic generation of ontology mappings. As this is a necessary component of ON, we will review some of these techniques below. The DOGGIE approach (Williams, 2004) focuses on machine learning techniques that can be used to make the meaning of a concept clear to another agent. Burnstein et al. (2003) propose a technique for meaning exchange between agents that is based on the formal semantics of ontologies, defined in terms of a neutral topic domain. The approach described by Wiesman and Roos (2004) proposes to find ontology mappings based on corresponding instances of concepts and joint attention. Furthermore, many of the automatic ontology mapping techniques that have been proposed outside the agent community are relevant for ON (Maynard et al. (2004) present an extensive survey). These techniques are discussed in more depth in the next chapter.

Automatic ontology mapping is not the only component that is required for successful ON. Other relevant work that focuses on the use and properties of ontology mappings in agent communication is reported by Doherty et al. (2003) and Stuckenschmidt and Timm (2002). A formal treatment of ontology mappings in agent communication from a programming perspective is reported by van Eijk et al. (2001). Soh and Chen (2005) have proposed a framework which aims at minimizing ontology exchange during ON. Their approach is based on the idea that ontology exchange is only useful when it improves operational efficiency. The work by Wang and Gasser (2002) describes how mutual ontology alignment in a group of agents can be of benefit for the whole group.

Another important component of ON is a detection mechanism for ontology mismatches. An overview of possible ontology mismatches is described by Visser et al. (1997). A study on how ontology mismatches can be detected is reported by Beun et al. (2004). The main focus of this work is on homonym problems that arise when multiple ontologies are used. For example, a word in one ontology might mean something different than the same word in another ontology. The approach describes a way to detect such mismatches.

Other related work for ON can be found in the literature on negotiation, of which, as its name suggests, ontology negotiation is a special kind. Negotiation protocols are well-studied interaction mechanisms that enable agents with different interests to cooperate (Rosenschein and Zlotkin, 1994). For example, they may be used to make a buyer and a seller agree on prices for certain goods, or in air traffic control, to decide which airplanes are allowed to land first. What ontology negotiation protocols have in common with these protocols is their distributed nature. As this is one of the most striking differences with other ontology reconciliation techniques, *ontology negotiation* is an appropriate name to characterize this approach. There is no central coordinating entity that manages the interactions between the agents, but the agents reach an agreement among themselves. In ontology negotiation the agreement is about a (piece of) shared ontology. Similar to other negotiation protocols, the agents' interests may be conflicting. It is easiest for an agent to make other agents adapt to its own ontology, thereby saving the costs of learning foreign ontologies. Of course, not every agent should maintain that policy. The ON protocol serves to resolve such issues.

2.4.5 Hybrid Approaches

The approaches described in this section are not mutually exclusive alternatives. They can be combined in several ways to achieve semantic interoperability among agents. Just as humans have multiple ways at their disposal to overcome language problems, so the same should probably also hold for agents.

Consider the scenario of the travel agent described in Chapter 1. Suppose a group of system developers take the first steps towards such a scenario by creating an integrated multi-agent system that represents all travel agencies in the Netherlands. They start their development by agreeing on a standard (upper) ontology which specifies the basic concepts in the problem domain. For example, this ontology contains concepts such as *country*, *customer*, *activity*, etc. The standard ontology is implemented in every agent. As the development of the system progresses, the standardized ontologies of the agents turn out to be insufficiently tailored to their specific tasks and extra concepts are added to the individual ontologies to solve these problems. For example, the agency for adventure travels might add a concept *bungee-jumping* to their agent's ontology and the travel agency for business travels might add a concept such as *OPEC-country*. As a result, the ontologies of the agents start to diverge. The system developers notice this, and apply ontology alignment to provide every agent with the right mappings that they need to communicate with other agents. For example, they make the mapping that *OPEC-country* is a *country* publicly available. After the development of the system has finished, the system is put to use in the Netherlands. Then the development team decides that it would be a good idea to integrate the Dutch system with the one that is used in Germany. Because the German agents use other ontologies than the Dutch agents, some form of ontology reconciliation is required. The development team decides to add an ontology mediator to the system that can translate between the ontologies of the Dutch and the German agents. This solution can be added to the system without making changes to the existing agents. However, despite the efforts that were invested into ontology integration, ontology alignment and adding mediator agents, some ontology problems remain. Sometimes a Dutch agent's ontology is changed, after which the ontology alignment between the Dutch agents is left outdated. Sometimes changes have occurred in one of the German ontologies without the ontology agent knowing it. It may also happen that foreign agents enter the system using ontologies that are unknown to the other agents and the ontology mediators. In a dynamic and heterogeneous system as this one, such unforeseen problems are bound to occur. In these cases, the agents are left to their own devices and ontology negotiation is the proper technique to solve their problems. For this reason, every agent should be enabled to participate in ontology negotiation.

The example shows the use of a hybrid approach to ontology reconciliation. The ontology integration efforts that were invested in the early development stages still prove themselves useful. Many messages between the Dutch agents are composed using the standard ontology. The ontology alignment is used when Dutch agents with different areas of expertise communicate to each other. When Dutch agents communicate with German agents, they consult the ontology mediator. The ontology problems that remain are solved using ontology negotiation.

To return to the analogy, the approach that is used by humans is hybrid. People employ different solutions to communicate with people in other language communities. Sometimes, one person has learned the native language of the other person (corresponding to point-to-point ontology alignment). Sometimes two persons speak a language other than their native language but one which is known by both persons (corresponding to intermediate ontology alignment). Sometimes people use a language interpreter (corresponding to an ontology agent). Nevertheless, it often occurs that people elucidate their language during their conversations (corresponding to ontology negotiation). For example, this happens when someone tries to make himself clear in a country of which he does not speak the language and when there are no interpreters around. It also occurs within the same language community, for example when the jargon that is spoken by an expert is not properly understood by another person.

2.5 Conclusion

In this chapter, we have introduced multi-agent systems as a technology offering unprecedented opportunities for software engineering. This is mainly because an agent is equipped with a flexible interaction mechanism that enables it to cooperate with another agent even if it has not previously been designed to do so. The interaction between agents proceeds via an agent communication language which, if successful, makes them *semantically interoperable*.

An ontology has been advocated as a key component to achieve semantic interoperability among agents. However, a careful study on the nature of ontologies and MAS's reveals that specifying an ontology alone is generally not sufficient for this purpose. We have argued that an ontology should not be regarded as a description of the structure of the world, but rather as a description of how agents choose to *view* the world. Autonomous agents may be heterogeneous and may view their world differently. Therefore, they may use different ontologies.

To solve the resulting semantic interoperability problems, some form of ontology reconciliation is required. Due to the dynamic and distributed nature of MAS's, the agents should be capable of establishing ontology mappings themselves without the involvement of humans or mediating agents. This leads us to an approach called *ontology negotiation*. In ON, the agents detect and resolve ontology problems while they engage in conversation.

A necessary component of ON is a technique to automatically establish an ontology mapping. This requires the agents to decide when two concepts can be regarded as related. The following chapter deals with the issues involved in this matter.

Chapter 3

Foundations of Ontology Reconciliation

Concepts without percepts are empty, percepts without concepts are blind.

Immanuel Kant (A critique of pure reason, 1781)

This chapter aims at providing a theoretical basis for ontology reconciliation. This leads to philosophical topics such as what meaning is and how it can be represented in an ontology. As innocent as these issues might appear, they open a Pandora's Box of problems. Our discussion is limited to those issues that are relevant for the current investigation.

Section 3.1 presents a brief analysis of meaning from a semiotic perspective. Section 3.2 discusses how meaning can be satisfactorily represented. In particular, we describe Tarskian semantics, its shortcomings, and a way to overcome these problems. Section 3.3 elaborates on these ideas and argues how automatic ontology mapping can be established. Furthermore, we compare our approach to ontology reconciliation with other approaches. We conclude in Section 3.4.

3.1 Meaning

We start our discussion with an analysis of meaning in the context of semiotics, i.e. the study of signs, symbols and meanings. One of the founders of semiotics is Charles Sanders Peirce (1839-1914). His account of meaning is discussed below.

3.1.1 The Peircean Model

In the model of Peirce, a sign is assumed to consist of three parts, yielding a *triadic model* (Chandler, 2002). Whereas many authors have accepted the presence of three

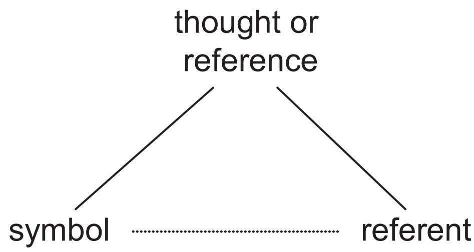


Figure 3.1: The Peircean model of a sign

components of a sign, there is no consensus on the names of these components and their exact details. In our discussion, we adopt the names that are used by Ogden and Richards (1923) and try to stay away from as many controversial issues as possible.

Peirce's triadic model is often represented as a triangle with the three components at its corners, as depicted in Figure 3.1. The first part of the sign is the *symbol* and regards the form which the sign takes. The second part is the *thought or reference*, which regards the sense made of a sign. The third part is the *referent*, i.e. that to which the sign refers. Each of the three components is discussed in more depth below.

Symbol

A symbol can be many things, such as a word, an image or a sound. When someone interprets a symbol as standing for something else, the symbol can be viewed as one of the three components that make up a sign. Many semioticians, among whom de Saussure and Peirce, have argued that symbols may be completely arbitrary, i.e. no natural connection may exist between a symbol and what it stands for. For this reason, Figure 3.1 shows the link between the symbol and the referent as a dashed line.

According to Peirce, signs may be arbitrary in different degrees. An example of a sign that is not completely arbitrary is an *icon*. An icon is a sign in which the symbol resembles what it stands for in some way. An example of an icon is ✈, commonly used for representing an airport. This sign qualifies as an icon for airports because it resembles, in some sense, the shape of an airplane. Other examples of icons are onomatopoeia, i.e. words that sound similar to what they refer to, such as "cuckoo" and "boom".

Nevertheless, most symbols in a language are not iconic, but, as Peirce calls it, symbolic. A word such as "airplane" bears no natural relation to a real airplane, not in shape, neither in sound. Also, the concept-names in an ontology and the bit strings in a computer should be regarded as completely symbolic, i.e. as completely *arbitrary* signs.

Thought or reference

If the relation between the symbol and its referent is not established by resemblance, how else is it established? According to Peirce, the symbol and referent are related because they are mediated by the thought or reference (or *interpretant* in his terminology). The thought or reference corresponds to the sense that is made of the sign. Therefore, the theory presumes the presence of a sense-maker, or an *interpreter*. Thus, the symbol is connected to its referent by the thought of its interpreter.

Meaning arises in the Peircean model, when the interpreter makes sense of the symbol. This process is called *semiosis*. The meaning that arises during semiosis can be viewed as a relation between the symbol and the referent, i.e. corresponding to thought or reference. Therefore, other versions of the semiotic triangle have used the word "meaning" to refer to what we call "thought or reference"(Vogt, 2002).

Referent

The referent is the object, in some kind of external and objective reality, to which the symbol refers. Objects are not restricted to concrete objects, but may be interpreted rather broadly. Referents may be abstract, fictional or refer to a state of affairs. For example, a traffic light sign for stop consists of: a red light facing traffic at an intersection (the symbol), the idea that a red light indicates that vehicles must stop (the thought or reference) and vehicles halting (the referent) (Chandler, 2002).

3.1.2 Sinn und Bedeutung

Gottlob Frege (1848-1925) was a German philosopher who has been very influential in the philosophy of language. In his seminal paper "Über Sinn und Bedeutung" (Frege, 1892), Frege argued that the referent of a term alone is insufficient to provide a complete account of meaning. His argument is as follows. Consider the terms *morning star* and *evening star* which both have the same referent (or *bedeutung* in his terminology), namely Venus. If the meaning of a term were fully determined by its referent, then the *morning star* would mean exactly the same as the *evening star*. A consequence of this would be that the sentence "*the morning star is the evening star*" would mean exactly the same as the sentence "*the morning star is the morning star*". However, the first sentence describes a major astronomical discovery, whereas the last sentence is trivial, i.e. it is true on the basis of its constituents. Frege proposed to solve this problem by adding an extra component to the analysis of meaning, namely *sinn*. Whereas, the *morning star* and the *evening star* have the same referent (*bedeutung*), they differ in *sinn*. Consequently, the sentence "*the morning star is the evening star*" can be explained to differ in meaning from the sentence "*the morning star is the morning star*".

Frege's distinction between *sinn* and *bedeutung* matches well with the Peircean triad (Sowa, 2000, p.192). Frege's *sinn* corresponds to Peirce's *thought or reference* and Frege's *bedeutung* corresponds to Peirce's *referent*. In Peirce's model, the terms *morning star* and *evening star* have the same referent. The thought or reference differs as the first term evokes a thought of being visible in the morning and the latter term evokes a thought of being visible in the evening.

3.1.3 Conventionality

The semiotic analysis described in this section has stressed once more the need for conventions in symbolic systems. The more arbitrary a sign is, the more important the role of conventions. Whereas iconic signs involve some degree of conventionality, communication using symbolic signs is *based* on convention. This social dimension has been recognized by most semioticians.

Whereas humans usually interpret signs by unconsciously relating them to a familiar system of conventions (Chandler, 2002), computer systems must be provided with an explicit account of conventions. Ontologies have been proposed to make such conventions explicit. Thus, the *purpose* of ontologies in IS can be given a justification from semiotics. The next section discusses how ontologies can be *realized* in a way that is semiotically justified. In particular, we will discuss how meaning can be represented in a computer.

3.2 Meaning Representation

The following quotation characterizes the problem that is currently encountered by AI researchers doing knowledge representation.

> The notation we use must be understandable to those using it and reading it; so it must have a semantics; so it must have a Tarskian semantics, because there is no other candidate.
>
> (McDermott, 1987)

As the above quotation indicates, Tarskian semantics is the most common way to provide a formal account of meaning. This also appears from formalisms that have been proposed to develop ontologies, which mostly follow a Tarskian approach. Tarskian semantics is discussed in Section 3.2.1 and 3.2.2.

Despite its popularity in the KR community, Tarskian semantics is not ideal for all purposes. In fact, it is difficult to justify from a semiotic perspective. We will discuss two problems of Tarskian semantics, namely the problem of indistinguishable models in Section 3.2.3 and the symbol grounding problem in Section 3.2.4. A way to overcome these problems is discussed in Section 3.2.5.

3.2.1 World

As described in a well-known textbook on AI (Genesereth and Nilsson, 1987), the formalization of knowledge begins with determining which objects are assumed to exist and their interrelationships. The set of objects in the world that is inhabited by the agent is called the *domain of discourse*, denoted by $\Delta^{\mathcal{I}}$. There is usually not one unique way to determine which objects exist in the world. Basically, an object may be anything about which we want to say something.

For an agent to be able to store knowledge about $\Delta^{\mathcal{I}}$, it needs a conceptualization, denoted by ρ. The conceptualization consists of sets of objects that serve to specify the meanings of concepts, and n-tuples that serve to specify the meanings of n-ary relations between objects. Note that, at this level, the elements in the conceptualization are not yet named. The conceptualization only contains the meanings themselves.

Example 3.1. As a running example, we use a very simple scenario in which we want to represent knowledge about the even and odd numbers between 1 and 4. The domain of discourse and conceptualization are as follows:

- $\Delta^{\mathcal{I}} = \{1, 2, 3, 4\}$: the domain of discourse consists of the natural numbers between 1 and 4.
- $\rho = \{\{2, 4\}, \{1, 3\}, \{1, 2, 3, 4\}, \emptyset\}$: the conceptualization consists of the set of even numbers and odd numbers in this domain. Furthermore, we included the set of all numbers in the domain, and the empty set as elements in the conceptualization.

By specifying the domain of discourse and the conceptualization, the referent component in the Peircean model is given a formal basis. The next section describes how the other components of the sign can be formalized.

3.2.2 Description Logic

The next step in the formalization of knowledge is to introduce a language that is appropriate to the meanings one wishes to represent. We have chosen for description logic (DL) (Baader and Nutt, 2003), as this language is particularly popular in the context of ontologies and the semantic web. However, for the purpose of demonstrating Tarskian semantics, we could also have chosen another language, such as predicate logic.

Syntax

Firstly, the language must contain *names* for the elements in the domain of discourse, also referred to as *individuals*. The names for the elements in $\Delta^{\mathcal{I}}$ are given by the set Δ. Secondly, the language must contain names for the elements in ρ. The names for the unary relations in ρ are given by the set of *concepts* C that are definable using a *concept language*. The names for the binary relations in ρ are given by the set of *roles* R. Conforming to the description logic paradigm, the conceptualization is assumed to contain no n-ary relations for $n > 2$. In this section, we adopt the description logic $\mathcal{ALC}$ as a concept language.

Let C^a be a set of atomic concepts with typical elements c, d and let R be a set of roles with typical elements r, s. The syntax of the concept language $\mathcal{ALC}$, with typical elements C, D is defined by the following BNF:

$C ::=$	c	$\mid$	(atomic concept)
	$\top$	$\mid$	(top concept)
	$\bot$	$\mid$	(bottom concept)
	$\neg C$	$\mid$	(negation)
	$C \sqcap D$	$\mid$	(intersection)
	$C \sqcup D$	$\mid$	(union)
	$\forall r.C$	$\mid$	(value restriction)
	$\exists r.C$		(existential quantification)

Given a concept C and an individual a, a *concept assertion* is of the form $C(a)$. The intuitive meaning of this assertion is that individual a is member of concept C. Given a role r and individuals a, b, a *role assertion* is of the form $r(a, b)$. The intuitive meaning of this assertion is that individual a stands in relation r with individual b. A set of concept and role assertions constitute an *ABox*, referred to as $\mathcal{A}$.

The relations between concepts are defined by *terminological axioms*. Given two concepts C, D, a terminological axiom is of the form $C \equiv D$ or $C \sqsubseteq D$. The intuitive meaning of the former axiom is that C is equivalent to D. The intuitive meaning of the latter axiom is that C is either equivalent to, or a subconcept of D. A set of terminological axioms constitute a *TBox*, referred to as $\mathcal{T}$. We regard a TBox as an ontology as it specifies the conceptualization.

So far, we have discussed the syntax of description logic. This corresponds to the symbol component of the Peircean model. Having treated the referent component in Section 3.2.1, this leaves us to specify the third component of the sign which relates the symbol to its referent. This is done by specifying the *semantics* of $\mathcal{ALC}$.

Semantics

The semantics of the concept language is defined using an interpretation function[1] $\mathcal{I}$ that maps individuals in Δ to elements in $\Delta^{\mathcal{I}}$, concepts in C to subsets of $\Delta^{\mathcal{I}}$ and roles in R to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Given the interpretation of atomic concepts and roles, the semantics of the concept language $\mathcal{ALC}$ is defined by restricting the interpretation function as follows:

- $\mathcal{I}(\top) = \Delta^{\mathcal{I}}$
- $\mathcal{I}(\bot) = \emptyset$
- $\mathcal{I}(\neg C) = \Delta^{\mathcal{I}} \backslash \mathcal{I}(C)$
- $\mathcal{I}(C \sqcap D) = \mathcal{I}(C) \cap \mathcal{I}(D)$
- $\mathcal{I}(C \sqcup D) = \mathcal{I}(C) \cup \mathcal{I}(D)$
- $\mathcal{I}(\forall r.C) = \{a \in \Delta^{\mathcal{I}} | \forall b.\langle a, b\rangle \in \mathcal{I}(r) \rightarrow b \in \mathcal{I}(C)\}$

[1] The description logic community usually refers to the interpretation function with $\cdot^{\mathcal{I}}$ and refers to the model $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ with $\mathcal{I}$. Our notation differs by writing $\mathcal{I}$ to refer to the interpretation function, conforming to the notation that is most common in logic.

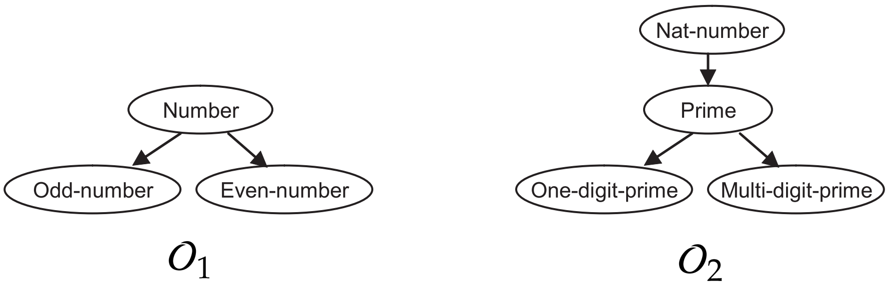


Figure 3.2: Example of number ontologies

- $\mathcal{I}(\exists r.C) = \{a \in \Delta^{\mathcal{I}} | \exists b.\langle a, b\rangle \in \mathcal{I}(r)\}$

The semantics of concept assertions is defined as $\models_{\mathcal{I}} C(a)$ iff $\mathcal{I}(a) \in \mathcal{I}(C)$. The semantics of role assertions is defined as $\models_{\mathcal{I}} r(a, b)$ iff $\langle \mathcal{I}(a), \mathcal{I}(b)\rangle \in \mathcal{I}(r)$.

An interpretation $\mathcal{I}$ satisfies a terminological axiom of the form $C \equiv D$, written $\models_{\mathcal{I}} C \equiv D$ iff $\mathcal{I}(C) = \mathcal{I}(D)$. An interpretation $\mathcal{I}$ satisfies a terminological axiom of the form $C \sqsubseteq D$, written $\models_{\mathcal{I}} C \sqsubseteq D$ iff $\mathcal{I}(C) \subseteq \mathcal{I}(D)$.

For a set of statements Γ, we write that $\models_{\mathcal{I}} \Gamma$ iff for every $\gamma \in \Gamma$, it holds that $\models_{\mathcal{I}} \gamma$. We write that $\Gamma \models \Gamma'$ iff for all $\mathcal{I} : \models_{\mathcal{I}} \Gamma$ implies $\models_{\mathcal{I}} \Gamma'$.

Example 3.2. This example elaborates on Example 3.1. We give names to the elements in the domain of discourse by adopting the set $\Delta = \{one, two, three, four\}$. Now, suppose that we want to specify the conceptualization consisting of odd and even numbers, i.e. we want to define the ontology $\mathcal{O}_1$ depicted in Figure 3.2. The TBox is specified as follows:

$\mathcal{T} = \{$ *Number* $\equiv \top$
Odd-number $\equiv \neg$ *Even-number* $\}$

The first terminological axiom states that *Number* contains all numbers in the domain of discourse. The second axiom states that *Odd-number* is the complement of *Even-number*. This TBox already contains a reasonable amount of implicit knowledge that can be made explicit by following the semantics of description logic. This kind of reasoning is also called *terminological reasoning*. Examples are:

$\mathcal{T} \models$ *Odd-number* $\sqsubseteq$ *Number* : an odd-number is a number.
$\mathcal{T} \models$ *Odd-number* $\sqcap$ *Even-number* $\equiv \neg$*Number* : no number is both odd and even.
$\mathcal{T} \models$ *Odd-number* $\sqcup$ *Even-number* $\equiv$ *Number* : all numbers are either odd or even.

Now, suppose that we want to define assertional knowledge using this ontology. This can be done using an ABox, for example:

$\mathcal{A} = \{$ *Even-number(two)*
Odd-number(three) $\}$

The combination of this ABox, together with the TBox presented earlier enables the following information to be derived:

$\mathcal{T}, \mathcal{A} \models \neg$*Odd-number(two)* : Two is not an odd number
$\mathcal{T}, \mathcal{A} \models$ *Number(three)* : Three is a number

The previous example shows what description logics are good at. By formally describing concepts and the relations between them, we obtain a system that not only possesses knowledge about what is explicitly represented, but is also capable of deriving new knowledge by taking the formal semantics of the statements into account. Many textbooks on AI take this as a crucial aspect of a knowledge system (Darlington, 2000). This reasoning capability can be regarded as a success of Tarskian semantics.

Nevertheless, it should be borne in mind that this reasoning occurs *within the system*. This property holds for almost every AI reasoning technique, such as automated theorem provers, DL reasoners, inference engines and logic programming. Without going into details, we will briefly characterize this kind of reasoning. On an abstract level, reasoning within the system solves a search problem consisting of an initial state, a goal state and actions that can be used to traverse through the search space (Russel and Norvig, 2003). The initial state corresponds to the premises; the goal state corresponds to the conclusion; the actions correspond to reasoning rules. For example, consider a simple propositional reasoner. Suppose the initial state contains p, $p \rightarrow q$, $q \rightarrow r$, the goal state contains r, and the available reasoning rule is modus ponens. By finding a path containing two times modus ponens, the system connects the initial state to the goal state and is justified to conclude r. A DL reasoner works analogously by assuming the TBox and ABox as the initial state and the formula to be derived as the goal state. The actions are usually reasoning rules from some kind of tableau algorithm (Baader and Nutt, 2003). If the reasoner can find a path from the initial state to the goal state, it is justified to conclude the formula in the goal state.

When we are interested in deriving a concept relation *between two systems*, the situation is different. In this case, a path of reasoning rules that connects the premises to the conclusion is typically absent, as it is broken by the boundary between the two systems. For example, consider an initial state $p, p \rightarrow q, q \rightarrow r$ that stems from system A and a goal state s that stems from system B. In this case, it is impossible to use modus ponens (or any other rule) to establish a path from the initial state to the goal state. The path of reasoning rules simply fails to bridge the gap between states with propositions from A and states with propositions from B. Likewise, the relation between *Even-number* in O_1 and *Prime* in O_2 (Figure 3.2) is difficult to derive, as no explicit statements exist that relate concepts in O_1 to concepts in O_2.

Whereas Tarskian semantics has provided the basis for reasoning techniques *within* a system, it is not necessarily a proper basis for reasoning techniques *between* systems, such as ontology alignment techniques. Because applying reasoning rules alone is insufficient to derive a relation between two concepts in different ontologies, we must compare the *meanings* of the concepts as specified by Tarskian semantics. In order to derive that two concepts are equivalent, we must check whether their interpretations are equivalent. Therefore, the question becomes relevant whether we can devise a TBox that specifies an interpretation in a way that is suited for this purpose. Unfortunately, this is not the case as will be demonstrated in the next section.

3.2.3 Indistinguishable Models

Returning to the conceptualization in Example 3.1 and the TBox in Example 3.2, we might wonder how well this TBox actually models this conceptualization. Ideally, the TBox allows for one unique interpretation, i.e. the interpretation that maps concepts to elements in the conceptualization. We call this interpretation the *intended interpretation*, denoted by $\mathcal{I}^{INT}$. In our example: $\mathcal{I}^{INT}(\textit{Even-number}) = \{2, 4\}$; $\mathcal{I}^{INT}(\textit{Odd-number}) = \{1, 3\}$; $\mathcal{I}^{INT}(\textit{Number}) = \{1, 2, 3, 4\}$; $\mathcal{I}^{INT}(\perp) = \emptyset$.

Although $\mathcal{I}^{INT}$ is an interpretation that is allowed by the TBox $\mathcal{T}$, it is not the *only* interpretation that is permitted by $\mathcal{T}$. For example, another permitted (non-intended) interpretation is $\mathcal{I}^2$: $\mathcal{I}^2(\textit{Even-number}) = \{1, 3\}$; $\mathcal{I}^2(\textit{Odd-number}) = \{2, 4\}$. Nevertheless, the terminological axioms in $\mathcal{T}$ did succeed in excluding some non-intended interpretations, such as $\mathcal{I}^3$: $\mathcal{I}^3(\textit{Even-number}) = \{1, 2, 3\}$; $\mathcal{I}^3(\textit{Odd-number}) = \{2, 3, 4\}$. $\mathcal{I}^3$ is excluded as it does not satisfy the axiom that states that Even-number and Odd-number are disjoint.

This might encourage one to try and add more terminological axioms to $\mathcal{T}$, such that non-intended interpretations such as $\mathcal{I}^2$ are also excluded. A possible attempt which makes use of roles is shown below:

$$
\begin{array}{ll}
\mathcal{T} = \{ & \textit{Number} \equiv \top \\
& \textit{Odd-number} \equiv \neg\ \textit{Even-number} \\
& \textit{Odd-number} \sqsubseteq \exists \textit{Smaller-than}.\textit{Number}\ \}
\end{array}
$$

This attempt aims at excluding a non-intended interpretation such as $\mathcal{I}^2$ by stating that an odd-number is always smaller than some other number in this domain of discourse. Indeed, if *Smaller-than* is interpreted as intended, namely as $\{\langle x, y\rangle | 1 \leq x < y \leq 4\}$, then $\mathcal{I}^2$ would be excluded. This is because $\exists \textit{Smaller-than}.\textit{Number}$ would be interpreted as $\{1, 2, 3\}$, such that the interpretation of *Odd-number* could not be $\{2, 4\}$. However, the role *Smaller-than* might also acquire a non-intended interpretation, e.g. $\{\langle x, y\rangle | 1 \leq y < x \leq 4\}$, i.e. the interpretation of the greater-than relation. In this case, $\mathcal{I}^2$ would not be excluded. Because there is no way to ensure that *Smaller-than* is interpreted as intended, this attempt did not succeed in excluding the non-intended interpretation $\mathcal{I}^2$.

Another, more desperate, attempt might be as follows. We could use the more expressive description logic $\mathcal{ALCO}$ to define *collection of individuals*, i.e. concepts that are defined as enumeration of individuals. The semantics of a collection of individuals is defined as expected: $\mathcal{I}(\{a_1, ..., a_n\}) = \{\mathcal{I}(a_1), ..., \mathcal{I}(a_n)\}$. The TBox that aims at defining even and odd numbers as collections of individuals would be:

$$\begin{aligned}\mathcal{T} = \{ \quad & \textit{Number} \equiv \top \\ & \textit{Odd-number} \equiv \{\textit{one,three}\} \\ & \textit{Even-number} \equiv \{\textit{two,four}\} \ \}\end{aligned}$$

Besides the fact that spelling out all individuals belonging to a concept is not feasible in a large domain of discourse, this approach does not succeed either in excluding the non-intended interpretation $\mathcal{I}^2$. For example, this interpretation is obtained when the individuals *one*, *two*, *three* and *four* acquire the following non-intended interpretation: $\langle one,4\rangle$, $\langle two,3\rangle$, $\langle three,2\rangle$, $\langle four,1\rangle$. Because there is no way to ensure that individuals are interpreted as intended, using collections of individuals in the TBox does not help either to capture the intended interpretation.

Bisimulation

The problem to devise a TBox containing sufficient terminological axioms to exclude all non-intended interpretations is caused by an *invariance result* (Blackburn et al., 2001). An example of an invariance result in first order logic is that satisfaction is invariant under isomorphism. This means that if two first order models are isomorphic, then the same formulae are true in both models. Consequently, first order formulae cannot distinguish between two isomorphic models.

For description logic, the invariance result is even more far-reaching, i.e. satisfaction in description logic is invariant under *bisimulation*. We will explain bisimulation in description logic below. A well-known fact is that description logic $\mathcal{ALC}$ is a notational variant of the propositional modal logic $K_{(m)}$ (Schild, 1991). Formulae in $K_{(m)}$ are interpreted using a *Kripke model* $\mathcal{M} = \langle W, R, V\rangle$, where W is a set of worlds, R is an accessibility relation between worlds and V is a truth assignment function that states which propositions are true in which worlds. The relation between description logic interpretations and Kripke models are as follows:

DL Interpretation	Kripke model
domain of discourse	set of worlds
atomic role	accessibility relation
atomic concept	set of worlds in which a proposition is true

When two Kripke models are bisimilar, they make the same formulae true. Bisimulation can be defined as follows (Blackburn et al., 2001):

Definition 3.1. *Given two models $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$, a non-empty binary relation $Z \subseteq W \times W'$ is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ if the following conditions are satisfied:*

- *If wZw' then w and w' satisfy the same proposition letters.*
- *If wZw' and Rwv, then there exists v' (in $\mathcal{M}'$) such that vZv' and $R'w'v'$*
- *If wZw' and $R'w'v'$, then there exists v (in $\mathcal{M}$) such that vZv' and Rwv*

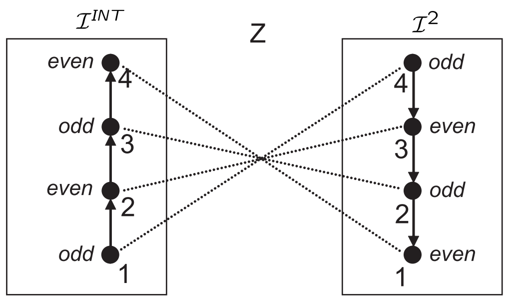


Figure 3.3: Bisimulation between intended and non-intended interpretation

Applying the notion of bisimulation to DL interpretations gives us insight on which interpretations can be excluded without excluding other interpretations. If the non-intended interpretation is bisimilar to the intended interpretation, then the same terminological axioms are satisfied by the two models. Therefore, it is useless to try to exclude the non-intended interpretation by adding extra terminological axioms. The TBox can simply not distinguish between the two interpretations.

Figure 3.3 shows a bisimulation between the intended interpretation of even and odd numbers and the non-intended interpretation $\mathcal{I}^2$. The worlds represent the objects 1,2,3 and 4; the arrows between worlds represent the *Smaller-than* role; the interpretations of even and odd numbers are indicated by placing them as propositions next to the worlds. The bisimulation points out that the attempts performed in the previous section to exclude $\mathcal{I}^2$ from the interpretations permitted by $\mathcal{T}$ were doomed to fail from the beginning.

The notion of bisimulation provides a theoretical underpinning on which models can be distinguished by description logics and which can not. For every model, there are bisimilar models. Since we are interested in the precise characterization of a model, this is problematic. In the example of Figure 3.3, the one interpretation of even and odd numbers is exactly the opposite of the other interpretation. Apparently, bisimilar models may be very dissimilar. Moreover, the example in Figure 3.3 shows an even stronger relation than bisimulation, namely isomorphism. This means that even the more expressive language predicate logic cannot distinguish between the two models. We must conclude that Tarskian semantics provides a very inaccurate way to characterize a model. For ontology mapping, such inaccuracies are not acceptable.

From a semiotic perspective, the problem can be rephrased as follows. Let Peirce's referent component be satisfactorily realized by the conceptualization and Peirce's symbol component by the language's syntax, the relation between the two remains to be specified. This must be done by an interpretation function that is described by logical formulae (in our case a TBox). The problem of indistinguishable models holds that a uniform description of the interpretation function cannot be given using Tarskian semantics. Therefore, the relation between the referent and the symbol remains underspecified.

Before we argue how these problems can be overcome, we will describe another problem of Tarskian semantics.

3.2.4 Symbol Grounding Problem

The problems that are faced by most knowledge representations are even more difficult than the ones described in the previous section. The examples with number agents discussed until now were relatively easy, as the domain of discourse of these agents could be represented using the objects themselves, i.e. numbers. Most knowledge representations are about things that exist in the world, for example hotels and flight-tickets. Because these things *themselves* cannot be included in a mathematical set, a symbolic abstraction must be used to represent the domain of discourse. This complicates things considerably.

Suppose, for example, that we choose $\Delta^{\mathcal{I}} = \{hotel1, hotel2, ticket1\}$ to model the domain of hotels and flight-tickets. Suppose that, in the language, we want to specify the intended interpretation of the concept *Hotel*, namely $\mathcal{I}^{\text{INT}}(Hotel) = \{hotel1, hotel2\}$. Even if we could model the unique intended interpretation of the concept *Hotel*, it would still be unspecified to which entities the symbols *hotel1* and *hotel2* refer to. A thorough semantics would include a formal analysis of the symbols *hotel1* and *hotel2* as well. Obviously, if we would continue along the same path of applying Tarskian semantics, we would end up in an infinite regress. However, what other symbolic means are available that are not subject to this problem? How is the meaning of a symbol to be grounded in something other than just more meaningless symbols? This problem is known as the *symbol grounding problem* (Harnad, 1990).

The symbol grounding problem is a philosophical problem about how symbols may acquire intrinsic meaning without being parasitic on the meanings in our heads. If the goal is to build a knowledge representation that is useful for humans, the human mind can bridge the gap between the symbol and its meaning. For example, a knowledge system that uses well-understandable names, such as *Hotel* and *Flight-ticket*, may be a helpful resource for humans to store and look up information about hotels and flight tickets. However, if the goal is to build a knowledge representation that provides the computer with knowledge about things, it becomes irrelevant which names are chosen. For example, we can replace a formula such as *Hotel(hotel1)* with a structurally equivalent formula with its atoms replaced by arbitrary unique strings, such as *g123(g726)*. For the computer that does not have background knowledge in understanding natural language, the two strings are equivalent in meaning. Nevertheless, the idea is wide spread that cleverly chosen names add something to what is represented for the machine. This fallacy has been called *wishful mnemonics* (McDermott, 1981).

The symbol grounding problem has been used by Searle (1980) in his Chinese room experiment to demonstrate the shortcomings of the symbol manipulation paradigm in AI. He considers a situation in which a non-Chinese speaking person answers Chinese questions by simply looking up the answers in a book containing Chinese question-answer pairs. Although the person does not understand any of the Chinese symbols that he receives and returns, he is capable of answering all questions of the Chinese questioner correctly. Searle argues that, although it might *seem* that this person speaks Chinese, we would not consider him a Chinese speaker, as no understanding is involved in answering the questions.

Besides philosophical problems, the symbol grounding problem can also be held responsible for many deep and persistent problems in symbolic AI. One of these problems is the frame problem (Harnad, 1993), i.e. the problem of efficiently determining which things remain the same in a changing world. Another problem is the common sense knowledge problem (Dreyfus, 1981), i.e. the problem of representing the vast amount of implicit knowledge that people have about the world and themselves. A symbolic knowledge representation requires all knowledge to be explicitly represented, including trivialities like "if Tony Blair is in Paris, then his left ear is also in Paris". It is unlikely that these problems will ever be solved in an ungrounded symbolic knowledge representation.

Another problem that can be added to the list of symptoms of the symbol grounding problem is the automatic ontology mapping problem. Ungrounded knowledge representations bottom out in symbolic descriptions of the world as conceived by human designers. As a result, the computer is left ignorant on the meaning of these symbolic primitives. Without any further means, we should not expect such a computer system to be able to derive inter-ontology relations. We can safely claim that, for an agent to learn the translation of another agent's concept into its own ontology, it must at least know the precise meaning of the concepts in its own ontology.

From a semiotic perspective, the symbol grounding problem can be rephrased as follows. By modeling the referent component of a sign using a symbolic representation, another semiotic triangle is created. This nesting of semiotic triangles is, according to Peirce, not a problem in itself. However, if every referent creates another sign which contains another referent, an infinite regress may occur. In semiotics, this is known as unlimited semiosis (Chandler, 2002). In AI, it is known as the symbol grounding problem.

3.2.5 Grounded Semantics

In order to find a way out of the impasse of purely symbolic representations, we return our attention to Peirce's triangle. From this perspective, a solution to the symbol grounding problem can be easily understood. A *symbol* (such as "hotel") is related to an external *referent* (hotels in the real world) via a *thought or reference*. When a human system developer is present, the semiotic triangle is complete as the system developer carries the *thought* that "hotel" relates to hotels in the real world. However, when the system developer is disregarded and the knowledge representation is considered by itself (as in an ungrounded KR), the *thought* is absent as well. As a result, an incomplete semiotic triangle remains in which only *symbols* and *referents* are present. In this amputated triangle, the *symbol* "hotel" is no longer related to its *referent* hotels in the real world, as this relation only existed due to a *thought or reference*. Peirce has formulated this as follows (Peirce, 1894, par. 7):

> The symbol is connected with its object by virtue of the idea of the symbol-using mind, without which no such connection would exist.

Stated in this way, it becomes obvious how the symbol grounding problem should be overcome. The semiotic triangle must be restored to include a *thought or reference*. The implementation of *thought or reference* in a computer system remains a difficult problem, and is called the *physical symbol grounding problem*(Vogt, 2002). For example, for a robot to solve the physical symbol grounding problem, it must know how to relate the sensory data that are received from its camera to the symbols that are used in its knowledge representation. Unlike the symbol grounding problem, which is a philosophical problem that applies to purely symbolic AI approaches, the physical symbol grounding problem is technical and can be solved, in principle.

Approaches for specifying semantics that go beyond the symbol level are called *grounded semantics*. Various approaches for grounded semantics have been proposed. Cognitive semantics provides a grounding in conceptual spaces (Gärdenfors, 2000), i.e. geometrical structures that are based on a number of quality dimensions. This approach has been advocated to open up the possibility for automatic ontology mapping by Kuhn (2004). Other approaches aim at grounding in experience (Wang, 1995), or grounding in iconic and categorical representations (Harnad, 1987). All these approaches are motivated by a dissatisfaction with Tarskian semantics for knowledge representation. A successful account of grounded semantics is immune to the problem of indistinguishable models and the symbol grounding problem.

In this thesis, we will provide a grounded semantics for agents by making use of their property of situatedness. As agents perceive the environment they inhabit, they can use their perception to ground their symbolic knowledge representations. A perceptual grounding requires an agent to recognize an object in the world as belonging to a concept in its KR.

In order to establish a perceptual grounding, an agent must be able to perceive the world about which it wishes to represent knowledge. This requirement is not always met, as the intended interpretation of a symbol may be something inaccessible for the computer. For example, it is most likely that a KR about hotels is implemented in a computer system that is incapable of perceiving a hotel in the real world. To quote Roy (2005), most current computer systems are trapped in sensory deprivation tanks, cut off from direct contact with the physical world.

Nevertheless, this requirement should not be regarded as a serious restriction to perceptually grounded semantics. Rather, it places a knowledge representation in the right context, namely that of perception (Chalmers et al., 1995). Things can only be known that are, at least some of the time, accessible to the senses. Whereas this conception has a long philosophical tradition (see the quotation from Kant at the beginning of this chapter), it turned out to be problematic for early AI systems. For situated agents, it need not be problematic. Below, we give four examples of perceptual grounding in different domains, ranging from simple to beyond the current state of technology.

We begin with a very simple example concerning the number ontologies.

Example 3.3. Grounding for number agents
Consider two agents Ag-1 and Ag-2 with ontologies O_1 and O_2 (Figure 3.2). The ontologies are specified using TBoxes. A perceptual grounding is provided by a set classifiers that recognize which numbers belong to which classes.
The classifiers of Ag-1 are implemented as follows:

- **Action** Classify(*Number,x*)
  return true
- **Action** Classify(*Odd-number,x*)
  if (x **mod** 2) = 1 **then return** true
  else return false
- **Action** Classify(*Even-number,x*)
  if (x **mod** 2) = 0 **then return** true
  else return false

The classifiers of Ag-2 are implemented as follows:

- **Action** Classify(*Nat-number,x*)
  return true
- **Action** Classify(*Prime,x*)
  if (x > 1) **and** (*for all* $y \in \mathbb{N}$ *s.t.* $1<y<x$: (x **mod** y) > 0) **then return** true
  else return false
- **Action** Classify(*One-digit-prime,x*)
  if (1 < x < 10) **and** (*for all* $y \in \mathbb{N}$ *s.t.* $1<y<x$: (x **mod** y) > 0) **then return** true
  else return false
- **Action** Classify(*Multi-digit-prime,x*)
  if (x > 9) **and** (*for all* $y \in \mathbb{N}$ *s.t.* $1<y<x$: (x **mod** y) > 0) **then return** true
  else return false

Example 3.4. Grounding on the internet
A domain that is particularly well suited for grounded semantics is the internet. Much content on the internet concerns knowledge about the internet itself. As the internet itself is accessible to software agents, a grounding is available. We will give a few examples below.

Web directories such as the open directory project[2] and the Yahoo directory[3] are specialized in linking to other web pages and categorizing those links. These category names can be given a perceptually grounded semantics by constructing a classifier that recognizes which web pages belong to which categories. Such a classifier can be implemented using text classification techniques available from the natural language processing community (Jackson and Moulinier, 2002).

[2]http://www.dmoz.org/
[3]http://dir.yahoo.com/

A growing market on the internet is that of online music stores, such as the iTunes Music Store[4]. These music stores classify their audio files into different musical genres. The semantics of the genre classification can be grounded by constructing a classifier that recognizes which audio files belong to which genres. Musical genre classifiers have been studied by Tzanetakis and Cook (2002).

Another possible application of perceptually grounded semantics on the internet is the domain of RSS news feeds. This topic is investigated in our case study discussed in Chapter 7.

Example 3.5. Grounding for mobile robots
Mobile robots equipped with sensors such as cameras and sonars provide other means for perceptual grounding. This type of grounding underlies the approach to language development described by Steels (1998b). A perceptual grounding is also used in the robot soccer domain, for example to connect the concept "ball" in the robot's internal workings to the football in the playing field. This kind of grounding can be established using pattern recognition techniques.

Example 3.6. Complex grounding scenarios
For other domains, the current state of the art falls short to enable a specification of grounded semantics. Consider, for example, the following items from the Dublin Core ontology (DCMI, 2005): *Accrual Periodicity*, *Audience Education Level* and *Provenance*. These items are difficult to analyze in terms of the Peircean model. What is their referent? What is their thought or reference? Even if they can be made to fit in the Peircean model in a satisfactory manner, the classifier would be very difficult to implement. A grounding in perception alone would probably be insufficient. Most likely, notions such as time, action and purpose are involved. These notions are not yet sufficiently understood to provide a basis for grounded semantics.

3.3 Ontology Reconciliation

Having argued for grounded semantics in knowledge representations, we will now turn our attention to ontology reconciliation. This involves the issue of how one agent can convey the meaning of a concept to another agent and how the other agent can use that information to establish an alignment. These issues are discussed in the following two sections. After that, we will compare our approach with other approaches for automatic ontology mapping.

3.3.1 Meaning Conveyance

When the meaning of a non-shared concept is expressible in terms of shared concepts, a definition can be used as a means to convey meaning. In the same way, a dictionary is usually helpful to explain the meaning of an unknown word. However, this approach breaks down when insufficiently shared concepts are available to

[4] www.apple.com/itunes/music/

compose a definition. Likewise, a Chinese-Chinese dictionary is practically useless to someone who does not speak a word of Chinese.

In the total absence of shared symbols, meanings must be conveyed non-symbolically. The most striking example of this can be found in early child language learning. The spontaneous learning of sound-meaning relations by young infants may fairly be called an astonishing achievement. Although this cognitive process is still poorly understood, we can safely claim that, in the end, it must be based on something that the infant and its language teachers *share* in some sense. This may be something in their brains or in the external world.

In our proposal, we build on the assumption that the agents have shared access to the referents in their world. They can use these referents as a basis for non-symbolic meaning conveyance. In this way, one agent conveys the meaning of a concept to another agent by pointing to examples in the external world. This kind of definition is called an *ostensive definition*, which was coined in modern philosophical literature by Ludwig Wittgenstein (1889-1951). Wittgenstein's primary purpose of introducing this notion was to contrast his own views with it. Nevertheless, he acknowledged that ostensive teaching plays an important role in child language development. He argued that it only applies to a small part of human language, i.e. a *primitive* language (Wittgenstein, 1953).

Since the expressivity of agent communication has not come close to that of human language, ostensive teaching has proven useful to teach concepts in an agent communication language, e.g. (Steels, 1998b; Obitko and Marik, 2003; Tzitzikas and Meghini, 2003). Our use of ostensive definitions is described in Chapter 4 and 5. Some philosophical limitations of the method are described below.

Limitations of ostensive definitions

We will describe the limitations of ostensive definitions in meaning conveyance from the perspective of the Peircean triangle. We will first discuss a possible problem with conveying the referent of a sign. Then, we will discuss a possible problem concerning the thought or reference of a sign.

As has been argued before, the referent in Peirce's triad is not restricted to concrete objects. This may cause problems for the application of ostensive definitions. Obviously, pointing is problematic to objects that are non-existent, e.g. *unicorn*, or processes, e.g. *halting*. In case the referent is an abstract object, e.g. *whiteness*, it becomes difficult to point to exactly that aspect (cf. the objection raised by Wittgenstein (1953)). Suppose one intends to ostensively define *whiteness* by pointing to a white sheet of paper. How does the other agent know if it is the color, shape or material that is being pointed at?

These difficulties arise when the perceptive information that is manifested by the ostensive definition is inadequate to determine the referent of the concept. This may be caused by the fact that the meaning of the concept could not be satisfactorily grounded in perception, as was the case in Example 3.6. Another reason may be that the perceptive information cannot be manifested in isolation, as was the case in the ostensive definition of *whiteness*. Nevertheless, the Examples 3.3, 3.4 and 3.5 discussed earlier are not affected by this problem.

Another problem with ostensive definitions concerns the conveyance of thought or reference. Clearly, an ostensive definition only explicates the referent of a sign. For the receiver to fully grasp the meaning of the sign, it must derive the thought or reference itself. This has a number of consequences. Subtle differences in meaning such as between the *morning star* and the *evening star* (discussed in Section 3.1.2) cannot be conveyed, as the referents of these expressions do not reveal any difference. Furthermore, it builds on the assumption that, given a referent, the receiver is capable of deriving a thought or reference. In the AI community, this problem is known as *concept learning* (Mitchell, 1997). This problem is by no means trivial. For example, it would require the number agent to derive the classifier for prime numbers (described in Example 3.3), solely on the basis of a few examples of prime numbers.

3.3.2 Alignment Criteria

As argued in the previous section, an ostensive definition only conveys a part of a concept's meaning from the sender to the receiver. In this section, we will show that this information is usually sufficient for the receiver to establish an ontological alignment.

Until now, we have discussed meaning from an objective viewpoint. When it comes to an agent performing ontology alignment, a subjective view is more appropriate. In Section 2.3.4, we have stressed the subjective and pragmatic nature of an agent's ontology. An ontology can be compared with a strong pair of glasses that determine what the agent can see. Hence, a difference in meaning that can be noticed from an objective viewpoint, may be invisible to an agent that judges from its own subjective viewpoint. Thus, we are not interested in *objective* criteria that specify when two meanings are equal, but in *subjective* criteria that specify when an agent ought to assess that two meanings are equal. The adoption of subjective alignment criteria has two important consequences that are discussed below.

Firstly, the task of meaning conveyance is facilitated because teaching a concept only requires those aspects of meaning to be conveyed that are relevant for the receiver. Having accepted that an agent's ontology fully specifies what is relevant, the teaching of a concept only requires the receiver to grasp the relations with the concepts in its own ontology. Particularly, the concepts that are acquired from other agents during ontology alignment are *not* given a perceptually grounded semantics. For example, when the number agent with O_1 is given the ostensive definition of the concept *Multi-digit-prime* it suffices to derive the relation that *Multi-digit-prime* is a subconcept of *Odd-number*. This alleviates the need to derive a classifier for *Multi-digit-prime*. Therefore, the concept-learning problem mentioned in the previous section is avoided.

Another consequence of adopting subjective alignment criteria is that two concepts may be regarded as having different meanings by one agent, and as having equal meanings by another agent. For example, the concepts *Prime* and *One-digit-*

prime have different meanings for the agent with O_2, but mean the same for the agent with O_1. This is because *Prime* and *One-digit-prime* have the same relations with the concepts in O_1, i.e. they are both subset of *Number*, and overlap with *Odd-number* and *Even-number*. How this idea can be exploited for minimal semantic integration forms an important subject in this thesis.

3.3.3 Comparison with Other Approaches

Most ontology mapping proceeds manually. Manual approaches can be justified from a semiotic perspective. The human becomes the interpreter of a sign and relates the symbol with its referent. For this reason, all components of the triadic model are present. This enables a proper comparison between two signs. In case the ontologies are well documented, manual mapping enables reliable ontology reconciliation. Otherwise, it can still be a very difficult task, even with human intervention. For example, without any documentation, it is not self-evident what *Accrual Periodicity* means (see Example 3.6).

Most techniques for automatic ontology mapping start with an ungrounded knowledge representation. Having discussed the problems with such representations, we will discuss to what extend these techniques succeed in overcoming these problems. Knowledge web (Maynard et al., 2004) has made the following classification for techniques for automatic ontology mapping:

- Extensional methods
- Terminological methods
- Structural methods
- Semantics-based methods

We will briefly discuss these approaches below.

Extensional methods compare the extensions of classes. This technique corresponds to our proposal for automatic ontology mapping. As has been discussed, using this technique with an ungrounded knowledge representation faces problems with indistinguishable models and symbol grounding, i.e. the extension of a concept is usually unknown. This could be circumvented by assuming that the agents' ABoxes exhibit a certain degree of overlap. This assumption does not necessarily hold in practice.

Terminological methods compare the strings that are used as the symbol of a sign. The methods range from finding a common substring in two strings, using Wordnet (Fellbaum, 1998), to using translation dictionaries to translate between different languages. Whereas these approaches might produce good results on some occasions, they are unsound from a Peircean perspective. In particular, they neglect the insight that signs are arbitrary. In Chapter 7, we will give some examples of terminological methods breaking down in practice. Lacking a theoretical foundation, we regard terminological methods as rules of thumb that do not promise any significant progress for the field in the future.

Structural methods compare the structures of entities in an ontology. For example, if two concepts have the same number of attributes, they are judged as being similar to a certain degree. Such methods, perhaps in combination with other methods, might help in finding the right ontology mappings. Nevertheless, they are theoretically difficult to justify. The following example is often used to demonstrate how the same piece of information can be represented in a *structurally* different way, e.g. (Rahm and Bernstein, 2001; Noy and McGuinness, 2001). Consider the concept *white-wine*. This concept can be defined as a subset of *wine*, i.e. $\textit{white} \equiv \textit{wine} \sqcap \textit{white-object}$. Another way to represent this information would be to use an attribute *color* and define *white wine* as *wine* with *white* as attribute value for *color*, i.e. $\textit{white-wine} \equiv \textit{wine} \sqcap \forall \textit{color}.\{\textit{white}\}$. In knowledge representation, this process is called *reification* as the concept *white-object* in the first representation is made into an object *white* in the second representation. The two representations are structurally different, yet represent the same information. Furthermore, nothing in the formal semantics of description logic specifies that these two representations represent the same. Clearly, structural methods have difficulties to discover such mappings.

Semantics-based methods make use of the formal semantics of ontology languages to derive ontology mappings. For example, subsumption or satisfiability tests are used, which are well-studied reasoning tasks in description logic. The limitations of these methods are that they only become usable after a preprocessing phase in which a number of concept mappings are declared. When no relations at all are known to exist between ontologies, these techniques are not usable to derive ontology mappings.

3.4 Conclusion

In this chapter, we have discussed Peirce's account of meaning and Tarski's account of meaning representation. As an example of Tarskian semantics, we have introduced description logic, a language which is particularly popular to formalize ontologies. Whereas Tarskian semantics is a successful formal basis to enable reasoning within a system, it is not a proper basis to enable reasoning between different systems. Most approaches for automatic ontology mapping accept this shortcoming, and use rules of thumb to overcome it.

In our approach, we aim at providing automatic ontology mapping with a theoretical basis. We therefore considered Peirce's theory of meaning which assumes that symbols require an interpreter in order to be meaningful. This idea is well usable in the context of multi-agent systems, where the agents themselves can act as symbol interpreters. By using a combination of Tarskian semantics and concept classifiers maintained by the agents, we obtain a so-called perceptually grounded semantics. This kind of semantics forms a solid basis for automatic ontology mapping.

A perceptually grounded knowledge representation opens the possibility to convey the meaning of a concept using an ostensive definition, i.e. by providing a set of examples of the concept to be explicated. The receiver of an ostensive definition establishes an ontological alignment based on a number of examples. This task does not necessarily require the receiver to reconstruct the exact meaning of a concept. Instead, an agent can maintain subjective alignment criteria. Therefore, the receiving agent is only required to grasp those aspects of the concept's meaning that it considers relevant.

In the following chapter we present our approach to ontology negotiation that is based on the ideas introduced until now.

Chapter 4

Ontology Reconciliation in ANEMONE

Whatever we may want to say, we probably won't say exactly that.

Marvin Minsky (1988, p.236)

This chapter presents ANEMONE: AN Effective Minimal Ontology Negotiation Environment (van Diggelen, Beun, Dignum, van Eijk and Meyer, 2006a). The chapter is organized as follows. Section 4.1 explicates the assumptions underlying this work. Furthermore, it serves as a bridge between the previous two chapters, the current chapter and the next two chapters. Section 4.2 presents an overview of the approach and states its objectives. Section 4.3 describes the communication protocols. Section 4.4 describes issues that arise in larger communities of agents and is followed by a discussion in Section 4.5. We conclude in Section 4.6

4.1 Assumptions

In the last chapters we have covered many issues, such as agents, communication, logics, ontologies and knowledge representations. Before we describe our approach to ontology reconciliation, we pay some attention to how these things fit together and describe our point of departure.

4.1.1 Agent Components

Figure 4.1 presents different parts of the technology and research that are relevant for achieving semantic interoperability in MAS's. Most of these parts have been touched upon in previous chapters in a criss-cross fashion. In this figure, they are loosely arranged as follows. Two components are placed next to each other

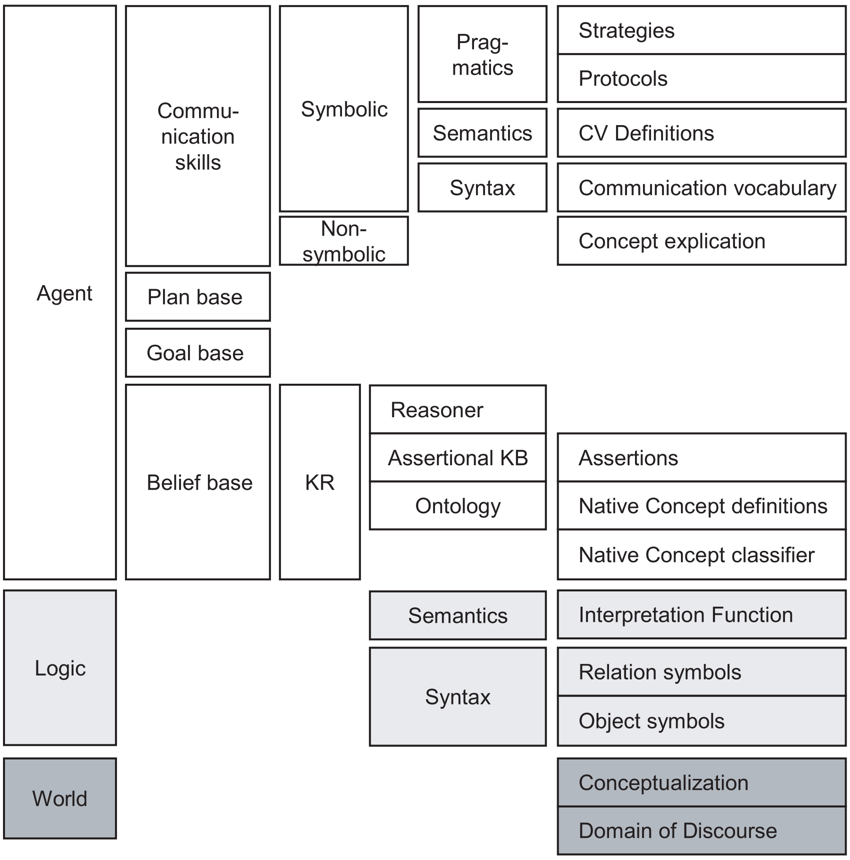


Figure 4.1: Agent components

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.

The bottom of this figure shows the world and logic components, which have been discussed in Section 3.2.1 and 3.2.2 respectively. Because these components form the foundations of the symbolic reasoning parts of an agent, they are placed at the base of the figure.

An agent consists of a belief base, goal base, plan base and communication skills. The belief base is implemented by a knowledge representation, which again consists of a reasoner, an assertional knowledge base (or ABox), and an ontology (or TBox). The ontology consists of *native* concepts, which stresses that these concepts are not learned but have been present since design-time. The native concepts are defined using native concept definitions. Native concepts are grounded by native concept classifiers.

The communication skills of an agent can be subdivided into non-symbolic communication skills and symbolic communication skills. Non-symbolic communication is required for concept explication. This is needed to make the meaning of a concept clear to another agent when there is no shared concept available to communicate the meaning symbolically. Symbolic communication is subdivided in syntax, semantics and pragmatics. The syntax is given by the communication vocabulary. The semantics of the communication vocabulary is defined in terms of the agent's native concepts. The pragmatics of communication is determined by communication protocols and strategies.

Dependencies

A number of components in Figure 4.1 are dependent on each other. These dependencies are important as they give insight into which parts of an agent can be changed without requiring revisions in the other parts. Figure 4.2 shows these dependencies, where an arrow from one component to another component represents that changes in one component must be carried through in the other component.

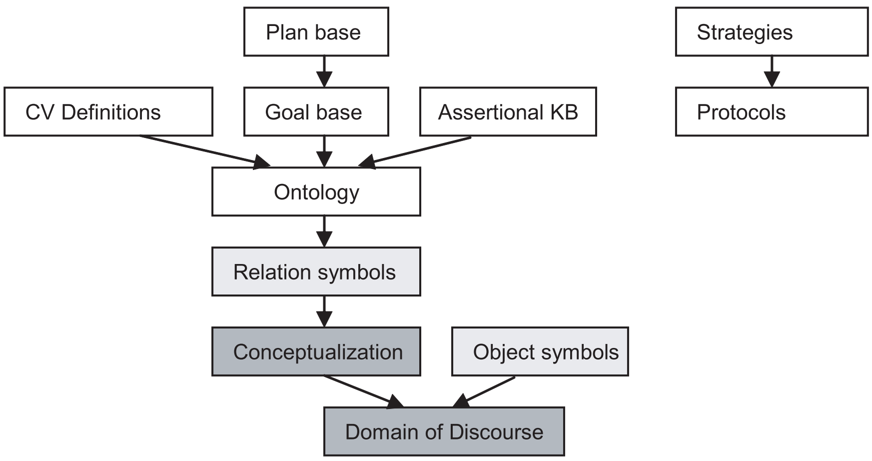


Figure 4.2: Dependencies between agent components

The dependencies are motivated below. Subsets of the *domain of discourse* constitute the *conceptualization* whose elements are referred to by *relation symbols*. Relation symbols are used in the agent's *ontology*. The *assertional knowledge base* and the definitions of the concepts in the cv (*cv definitions*) are defined in terms of the ontology. Furthermore, the *goal base* of an agent may contain declarative goals (van Riemsdijk et al., 2005), which are also defined in terms of the ontology. The *plan base* of an agent is designed to fulfil the goals. *Communication strategies* are dependent on *communication protocols*, as these resolve the non-determinism that exists in the communication protocol (see Section 2.2.2).

4.1.2 Degree of Heterogeneity

Agents may be heterogeneous in many degrees. In principal, it is not "more realistic" to assume that agents are heterogeneous in every aspect than to assume that agents share some common features. In fact, coordination between *completely* heterogeneous agents is paradoxical as the idea of coordination itself presumes some kind of organization. There is always some level of abstraction at which cooperating agents can be viewed to have something in common. Since we are faced with the task of making ontology reconciliation feasible, we must assume some homogeneity between the agents. Our goal is to make assumptions that are both realistic and enable the agents to solve ontological differences.

Below, we describe which of the agent-components in Figure 4.1 we assume to be homogeneous, starting from the bottom components.

Assumption 4.1. Homogeneous agent components

1. *Domain of discourse.* Ultimately, meaning is defined in terms of elements of the domain of discourse. To enable agents to reach semantic agreements, we assume this component to be shared.

2. *Object symbols.* The agents use the same symbols to refer to the elements in the domain of discourse. A straightforward extension of our work would allow agents to reach agreements on the meaning of object symbols (provided that the universe of discourse is shared). Because this problem is not central to ontology problems, we did not address it in this thesis.

3. *Concept Explication.* The agents share the method for explicating concepts. Therefore, non-linguistic communication does not cause any misunderstandings.

4. *Protocols.* The dialogues between agents adhere to the same communication protocol.

The following agent-components may be heterogeneous (but need not necessarily be)

Assumption 4.2. Possibly heterogeneous agent components

1. *Conceptualization*. Every agent may use a conceptualization that best suits its needs. Typically, the conceptualizations that underlie the agents' ontologies are different.

2. *Relation symbols*. Every agent uses different symbols to refer to the elements in the conceptualization.

3. *Interpretation function*. As a consequence of the preceding two heterogeneous components, interpretation functions are also different.

4. *Native concept classifier*. The agents' ontologies are heterogeneous. Therefore, they also have classifiers for different concepts. The concept classifiers may also differ with respect to the classification method.

5. *Native concept definitions*. Heterogeneous ontologies imply heterogeneous concept definitions.

6. *Assertions*. The assertional knowledge base of each agent typically contains different information. Otherwise, there would be no need for agents to communicate. Moreover, different ontologies give rise to different assertions.

7. *Goal base*. Different agents may have different goals.

8. *Plan base*. Different goals require different plans.

9. *Communication vocabulary.* The agents build up their communication vocabularies to enable communication. Therefore, it typically contains shared concepts. However, as will appear in this chapter, the communication vocabularies of agents are not always fully shared.

10. *Definitions of cv*. A concept in the communication vocabulary is defined in terms of the agent's native concepts. Because the native concepts of agents differ, the definitions of concepts in the cv are also different. As a consequence, two agents may ascribe slightly different meanings to a concept in the communication vocabulary.

11. *Strategies*. The communication strategies that are investigated in Chapter 6 are assumed to be homogeneous. However, this is not a strict requirement of Anemone. Therefore, we allow the agents to adopt heterogeneous strategies.

Since our point of departure also assumes some degree of homogeneity between the agents, one might raise the objection that nothing is actually gained compared to assuming one homogeneous ontology. This objection is misguided because, as argued in Chapter 2, a standardized ontology seriously impairs the problem solving performance of agents. This does not apply to Assumption 4.1. In fact, a common domain of discourse is likely to be already present in a community of agents that are eager to collaborate with each other. Agents with a different domain of discourse can be said to live in different worlds. There are no clear reasons why such agents should be able to interact in the first place. Agents with different ontologies can be said to view the world differently. It is for these agents that we intend to achieve semantic interoperability. The assumption we have made regarding common communication skills is not a restriction either in the way that a common ontology is. These assumptions do not affect the internal workings of the agent, but prescribe how the agents should interact with other agents. Achieving common communication skills in an open community of agents is mainly a problem of "pushing standards". These kinds of problems are beyond the scope of this thesis.

Besides the assumption that some agent components are the same for every agent, we also assume one agent component to be distinct for every agent. This assumption applies to the relation symbols, and consequently for the names of the native concepts.

Assumption 4.3. Assuredly heterogeneous agent components

1. *Relation symbols*. Every agent uses distinct names to refer to elements in the conceptualization.

2. *Native concepts*. None of the agents' native concepts have the same name.

This assumption achieves that concepts with the same name also have the same meaning. The problem of detecting ontology mismatches discussed in Section 2.4.4 is thereby facilitated. An agent regards every concept from another agent's ontology as a mismatch, unless it has previously learned the concept in concept explication. The problem that one word in one ontology might mean something different from the same word in another ontology is thus solved. This assumption can be easily realized by using namespaces: prefixing the agent's concept names with an identifier that is unique to the agent.

4.1.3 Degree of Sophistication

The different components of Figure 4.1 can be implemented with varying degrees of sophistication. In this thesis, we will focus on basic technologies that are well established in research. Below we will make these assumptions explicit.

Assumption 4.4. *The domain of discourse consists of concrete entities*[1] *perceptible to all agents.*

The motivation behind this assumption is that agents must be able to communicate the meaning of a concept by pointing to its instances. As discussed in Section 3.3, this technique is difficult to apply when the entities involved are non-existent, abstract or imperceptible.

Assumption 4.5. *The conceptualization consists of subsets of the domain of discourse.*

This assumption states that the meanings expressible by an agent's ontology are restricted to sets of objects. This excludes n-ary relations with $n \geq 2$. In the case of description logic ontologies, this means that no roles are present. There are two reasons why we have chosen for this simplification.

Firstly, it is simpler to point to examples of a concept, which are individuals, than to point to examples of a role, which are pairs of individuals. We will focus on

[1] In philosophy, the distinction between abstract and concrete objects is widely agreed to be of fundamental importance. Nevertheless, no standard account exists of how the distinction is to be explained (Rosen, 2006). In this thesis we will use the terms rather loosely, referring to their intuitive meaning.

the basics of non-symbolic communication and assume that the conceptualization contains no binary relations.

Secondly, when the ontology language contains roles, the type of mappings needed to reconcile ontologies are much more complex. We have already hinted on reification in Section 3.3.3, where we presented two ways to model the concept *white-wine*, namely as *wine* $\sqcap$ *white-object* and as *wine* $\sqcap$ $\forall$*color.{white}*. Reification is particularly important in knowledge representation when the representation language contains binary relations. This is because only in a language with roles, a reified concept (i.e. a concept that has been made into an object) can be used as an attribute value of a concept. When the ontology language contains roles, the ontology reconciliation technique should be able to deal with ontological differences in which a concept in one ontology is reified in the other ontology. Dealing with such differences is difficult, because the domain of discourse assumed by one ontology differs from the domain of discourse assumed by the other ontology (i.e. it conflicts with Assumption 4.1.1). For example, in the example of *white-wine*, the object *white* is absent in the domain of discourse assumed by the first representation, and present in the domain of discourse assumed by the second representation.

The last assumption we will describe in this section concerns the degree of sophistication of the communication protocol.

Assumption 4.6. *The agent communication protocol serves to enable one agent to inform another agent about an assertion.*

In the agent communication community, many kinds of communication protocols are investigated, for example auction protocols to make a buyer and a seller agree on a price (FIPA, 2001) or brokering protocols to support brokerage interaction in MAS's (FIPA, 2002b). An important aspect of these protocols is the exchange of assertional knowledge between agents. In this thesis, we will focus on this core aspect of agent communication and investigate communication protocols that serve the purpose of conveying an assertion from one agent to another.

4.2 Approach

4.2.1 Ontologies

Ontologies may take a variety of forms, but almost without exception they contain concepts in a subsumption hierarchy as their core elements. In our analysis, we avoid introducing unnecessary additional constructs and focus on simple ontologies consisting of sets of concepts and concept relations. A concept relation is one of the following: $\sqsubset$ (strict subconcept relation), $\sqsupset$ (strict superconcept relation), $\equiv$ (equivalence), $\perp$ (disjointness), $\oplus$ (overlap). The symbol $\perp$ is also used to denote the bottom concept. Figure 4.3 presents a graphical representation of two example ontologies. An arrow between two concepts represents a subconcept relation (and against the flow a superconcept relation). Two concepts in two different branches in the ontology are disjoint. Concepts that are equivalent or overlap are connected with a line with the ≡ or ⊕ symbol in it. For readability, we have left out concept relations that are derivable from other relations. The following terminology is useful for discussing the communication mechanisms of ANEMONE:

- γ_1 is a *particularization* w.r.t. γ_2 iff $\gamma_1 \sqsubset \gamma_2$ or $\gamma_1 \oplus \gamma_2$
- γ_1 is an *implied concept* w.r.t. γ_2 iff $\gamma_1 \sqsupset \gamma_2$ or $\gamma_1 \equiv \gamma_2$

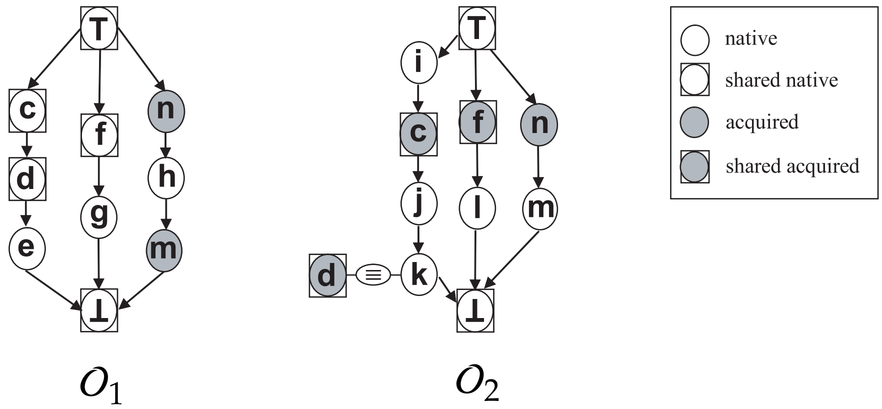


Figure 4.3: Example ontologies

In the context of ontology negotiation, some other aspects of ontologies are important. These are the distinction between *native* and *acquired* concepts and the distinction between *shared* and *non-shared* concepts. Both distinctions are explained below.

Native and Acquired concepts

In Figure 4.3, a native concept is represented as non-shaded and an acquired concept as shaded. An agent that has not yet exchanged any ontological knowledge with other agents, has only native concepts in its ontology. During ontology negotiation, the agents occasionally teach each other new concepts. These are the agent's acquired concepts and are defined additionally to its native concepts. We refer to the set of native concepts in an ontology as the native ontology. Likewise, the set of acquired concepts are referred to as the acquired ontology.

There are some important differences in the way that acquired concepts and native concepts are treated. These differences are summarized as follows:

	Native	**Acquired**
Is dynamic?	No	Yes
Is accompanied by concept classifiers?	Yes	No
May be what the speaker intends to convey?	Yes	No
May be how the hearer interprets a message?	Yes	No

Because the agents might learn new concepts at any time, the acquired ontology changes over time, i.e. it is dynamic. The native ontology, on the other hand, does not change over time, i.e. it is static. Whereas the agents have classifiers for their native concepts, the agents do not have classifiers for acquired concepts. This is because, during the teaching of a concept, the agent only learns the definition of the acquired concept in terms of other concepts. It does not learn the classifier for the acquired concept, which would be far more difficult to establish. As it turns out, it is not necessary either.

We will now focus on the last two differences, which have to do with speaking and hearing messages. A conversation starts with the intention of the speaker to convey an assertion to the hearer. This assertion is stated in terms of native concepts and not in terms of acquired concepts. Likewise, the interpretation of the message by the hearer corresponds to an assertion that is stated in terms of native concepts. The reason for this is simple. As discussed in Section 4.1, the beliefbase of the agent is stated in terms of its native ontology. Therefore, what the speaker intends to convey stems from a belief that is stated in terms of its native ontology. Furthermore, the belief that results when the hearer interprets a message, must be stated in terms of the native ontology.

One might raise the objection that an agent should also be allowed to state its beliefs in terms of acquired concepts. We will give two reasons why this objection is misguided. The first reason is that acquired concepts are learned. As argued in Section 4.1, the beliefbase of an agent is not an independent component, but is actually used by the agent to select its plans and goals. Changing the ontology of the beliefbase would also require a change in the plan base and goal base. Learning the meaning of an acquired concept in terms of other concepts is one thing, but learning how to use an acquired concept in planning and goal selection is far more difficult. A second reason not to use acquired concepts in the beliefbase is that it would probably not even be desirable. The native ontology of an agent is implemented by its system developer to enable it to perceive the world in the right categories that are optimal for fulfilling its tasks. Adding concepts of other agents to its ontology would change its world view and might reduce the agent's performance. We do not claim that it would be uninteresting if the agent's world view would be influenced by other agents, but rather that it would raise different issues that are not related to solving interoperability problems.

Shared concepts

Another important issue in ontology negotiation is keeping track of which concepts are shared with other agents. In Figure 4.3, a concept that is shared between Ag-1 and Ag-2 is represented by a box around the concept. When we speak of a concept that is shared between two agents, we actually mean that the concept is common knowledge between the agents. In ontology negotiation, a concept may also be *unknowingly shared*. In this case, both agents know the concept, but do not know this of each other. Examples of unknowingly shared concepts are **n** and **m** in Figure 4.3. This aspect becomes relevant when ontology negotiation is used in an environment with more than two agents. For example, consider three agents Ag-i, Ag-j and Ag-k. Ag-i might, in successive conversations with Ag-j and Ag-k, teach the same concept to Ag-j and Ag-k. As a result, Ag-j and Ag-k both know this concept, but do not know of each other that they know this concept. The concept has become unknowingly shared.

In a situation with more than two agents, the agents keep track with which they share which concepts. A concept may be shared between one pair of agents, and be non-shared between another pair of agents. This property follows naturally from the fact that the shared ontologies are built up decentralized. In the context of a speaker and a hearer, we use *shared ontology* to refer to the set of concepts that are shared between the speaker and the hearer. In the context of a multi-agent system, we use *communication vocabulary* to refer to the set of concepts that are knowingly or unknowingly shared between any pair of agents.

Comparison with other approaches

The distinction between acquired and native concepts serves the same goal as *mappings* in the ontology alignment approach (discussed in Section 2.4.2). For example, in Figure 4.3, Ag-2 knows that concept **d** from O_1 corresponds to concept **k** in its own ontology O_2, because its ontology contains an acquired concept **d** that is equivalent to native concept **k**. The same result could have been accomplished by using an ontology mapping between O_1 and O_2 that states that concept **d** is mapped to concept **k**.

As pointed out by Borgida and Serafini (2003), ontology mappings need not necessarily be equivalence mappings but may also state that a concept in one ontology is a sub- or superconcept of a concept in another ontology. The authors introduce a so-called *into-bridge rule* to specify that concept **c** in O_1 is a subconcept of concept **i** in O_2. They define an *onto-bridge rule* to specify that concept **c** in O_1 is a superconcept of concept **j** in O_2. Our approach also allows these kinds of relations to be defined, namely by placing concept **c** as an acquired concept in O_2 as a subconcept of **i** and as a superconcept of **j**.

The assumption discussed earlier that acquired concepts are not used for representing the beliefs of the agents is rather straightforward from the perspective that an acquired concept serves to specify an ontology mapping. Ontology mappings are used for *translations* between ontologies and do not change the way that an agent represents its beliefs. The same holds for acquired concepts.

As described in Section 2.4.2, ontology mappings can be implemented using a point-to-point topology or using an intermediate ontology. In our approach, the communication vocabulary can be viewed as analogous to an intermediate ontology. In both approaches, the speaker translates the native concept that it intends to convey to the communication vocabulary, which the hearer translates back again to its own native ontology. In another respect, however, the two approaches are different. The intermediate ontology is shared between every agent in the system. The communication vocabulary, on the other hand, may contain concepts that are shared between some pairs of agents, and not between other agents.

In Section 2.4.2, we have presented Esperanto as an analogy for an intermediate ontology. For the communication vocabulary that arises in our approach, a closer analogy would be a *pidgin* language. The American Heritage Dictionary (2000) defines a pidgin as a language with the following properties :

- It is usually a mixture of two or more languages
- It has a rudimentary grammar and vocabulary
- It is used for communication between groups speaking different languages
- It is not spoken as a first or native language

The analogy is striking because the communication vocabulary also contains native concepts stemming from different agents, i.e. it is a mixture between languages. Furthermore, it arises spontaneously serving a very practical purpose: to enable speakers of different languages to communicate.

4.2.2 Overview of the Protocol

The communication protocol of ANEMONE consists of three layers, as depicted in Figure 4.4.

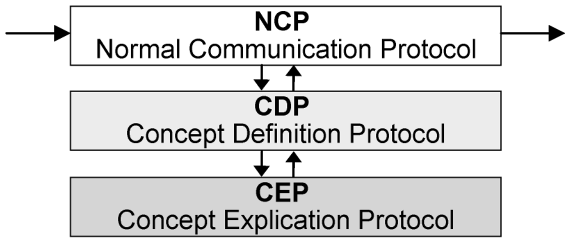


Figure 4.4: Layered communication protocol

The upper layer of the protocol (NCP) deals with normal agent communication, i.e. the kind of social interaction that agents normally exhibit when no ontology problems exist in the system. Every conversation starts in the NCP layer. If the agents fail to understand each other, the agents switch to the middle layer in the protocol. In this layer (CDP), the agents explain the meaning of a concept to each other by exchanging concept definitions. The meaning of a concept is explained in terms of other concepts. If the communication difficulties are so severe that the agents do not even understand each other's concept definitions, the agents switch to the lowest layer in the protocol (CEP). In CEP, the agents exchange the meaning of a concept using non-symbolic communication, i.e. by pointing to examples of the concept.

4.2.3 Design Objectives

The goal of this chapter is to describe the domain independent design choices of the ontology negotiation protocol. For example, how do the agents traverse through the protocol layers and what general techniques can be used in the individual layers. These design choices are motivated by a number of design objectives that state what the ontology negotiation protocol should be able to achieve. These design objectives are described below.

The goal of an ontology negotiation protocol is to enable the agents to build up something that enables them to reach a desired level of coordination. This raises questions like *what* should be built up during ontology negotiation, *where* should they build it up and *when* should they do that? The design objectives of Anemone are stated by answering these questions:

- What? *Minimal & Effective*
- When? *Lazy*
- Where? *Decentralized*

These objectives are explained in more depth below.

Minimal and effective

The ontology negotiation protocol should enable the agents to build up a shared ontology that is minimal and effective. The objective of minimality applies to both the size and the use of the shared ontology. An ontology that is minimal in size can be processed efficiently as it does not contain bulks of superfluous concepts. A case study performed by Giovannucci and Rodríguez-Aguilar (2005) points out that this is a serious issue in practice, as management of large ontologies might cause significant overload. An ontology that is minimal in use enables agents to keep their conversations short, leading to more efficient interactions.

The counterpart of minimality is effectiveness. The shared ontology should contain enough concepts to convey sufficient information in a sound manner. Particularly, communication should be *lossless*, meaning that, from the perspective of the hearer, no information is lost in the communication process. When communication is lossless, spending more effort on building the shared ontology does not contribute to the amount of information that can be conveyed between the agents.

Creating a shared ontology that is only minimal in size is trivial: adopt an empty ontology. Creating a shared ontology that is only effective is also trivial: make the shared ontology the union of all native ontologies in the system. The combination of a minimal and effective ontology, however, is non-trivial. The trade-off between the two is what makes this objective interesting.

Lazy

The lazy[2] objective states that ontology exchange should occur on an as-need basis. The agents should only put effort into building a shared ontology when strictly necessary. Ontology exchange is not a goal in itself, but a means to enable communication.

To meet the lazy objective, the agents should only leave NCP when communication fails to be effective and should try to return as soon as possible. Furthermore, the agents should only exchange ontological information in CEP when this cannot be done in CDP. This is because defining a concept in CDP is less resource-consuming than learning a concept in CEP. In other words, for the agents to be lazy, they should stay as high up as possible in the layered communication protocol.

Decentralized

In line with the ontology negotiation paradigm, no central location exists where a shared ontology is built up. Every agent increments its own ontology with the necessary concepts. In this way, they collectively address the semantic integration problem, solving communication problems between themselves when they arise. Multi-agent systems are usually thought to lack a central control. This makes a decentralized solution to interoperability problems particularly suited.

4.3 Protocols

4.3.1 Normal Communication Protocol

Normal communication protocols have been extensively studied in the agent communication literature. The focus of this section is on the adjustments that are required to deal with the partially shared ontologies described in Section 4.2.1, and the criteria upon which the agents base their decision to resort to CDP. This confronts us with the task of finding a proper way to deal with *message composition*, *message interpretation*, and the decision of when to *switch to CDP*. To discuss these issues, we assume a simple setting in which a speaker intends to convey one of its native concepts to a hearer. *Message composition* is the speaker's task of translating the native concept it intends to convey to a concept it will actually speak to the hearer. *Message interpretation* is the hearer's task of translating the concept in the message to one of its native concepts to store the information in its knowledge base. Either the speaker or the hearer may decide to *switch to CDP* when it believes that more concepts should be shared between them.

In this section, we will propose six drafts of NCP of increasing complexity. We begin our discussion with a trivial way to implement message composition, interpretation, and a switch to CDP. As we evaluate this protocol against the quality

[2] The term *lazy* is borrowed from the computer programming community where *lazy evaluation* refers to a technique that postpones the computation of an expression until it is known that the results of the computation are actually needed.

criteria of minimality, effectiveness, and laziness (discussed in Section 4.2.3), we detect shortcomings and adjust the protocol accordingly. Subsequent evaluations and refinements eventually give rise to a communication mechanism of considerable complexity, which is described by NCP-6 at the end of this section.

NCP-1: Equivalence mappings

Our first proposal for NCP is a simple communication mechanism that exploits some features of partially shared ontologies introduced in Section 4.2.1, namely the distinction between native and acquired concepts, equivalence mappings, and the notion of shared concepts. These features can be incorporated in a relatively straightforward manner. The communication mechanism is described in Figure 4.5.

Composition

- The agent translates what it intends to convey to an equivalent shared concept.

Interpretation

- The agent translates the concept in the message to an equivalent native concept.

Switch to CDP

- The speaker switches to CDP when it cannot compose its message, i.e. when no equivalent shared concept is available in its ontology.

Figure 4.5: NCP-1

The message composition part establishes that only *shared* concepts are communicated between the agents. The message interpretation part establishes that the hearer interprets the messages in terms of its *native* ontology. Some examples are given below.

Example 4.1. Consider the agents Ag-1 and Ag-2 with ontologies O_1 and O_2 respectively (Figure 4.3)

1. Ag-1 intends to convey **d**; in message composition, Ag-1 "translates" **d** to the equivalent shared concept **d**; in message interpretation, Ag-2 translates **d** to the equivalent native concept **k**.

2. Ag-2 intends to convey **k**; in message composition, Ag-2 translates **k** to the equivalent shared concept **d**; in message interpretation, Ag-1 "translates" **d** to the equivalent native concept **d**.

3. Ag-1 intends to convey **g**; Ag-1 cannot compose the message as there is no shared concept that is equivalent with **g**; Ag-1 switches to CDP to teach

concept **g** to Ag-2; After the agents have visited the lower layers in the protocol, Ag-2's ontology is extended with an acquired concept **g** that is equivalent with the native concept **l**; because Ag-1 now regards concept **g** as shared, it resumes message composition and "translates" **g** to the equivalent shared concept **g**; in message interpretation, Ag-2 translates **g** to the equivalent native concept **l**.

As this example points out, the first draft of NCP already enables a form of ontology negotiation between heterogeneous agents. The shared concepts enable communication in *both* directions, as shown by the first two conversations from Example 4.1. The third example shows when the speaker decides to switch to CDP. It is also clear that this communication mechanism does not give rise to information loss, i.e. it enables lossless communication.

The problem with this proposal is that it only offers a solution for ontology reconciliation problems that can be solved using equivalence mappings. For ontologies that have different levels of granularity or different scopes, this draft of NCP is too strict. For example, suppose that Ag-1 intends to convey **c** and "translates" **c** to the equivalent shared concept **c**. Using NCP-1, Ag-2 cannot interpret this message because there is no native concept in its ontology that is equivalent with **c**. The next draft of NCP aims at overcoming this problem.

NCP-2: Approximate message interpretation

The second proposal for NCP introduces approximate message interpretation. This achieves that the hearer no longer requires an equivalent native concept in its ontology to be able to interpret a message. The proposal is described in Figure 4.6.

Composition

- The agent translates what it intends to convey to an equivalent shared concept.

Interpretation

- The agent translates the concept in the message to the most specific implied native concept.

Switch to CDP

- The speaker switches to CDP when it cannot compose its message, i.e. when no equivalent shared concept is available in its ontology.

Figure 4.6: NCP-2

In this draft, message interpretation is defined less strictly than in the previous draft. If there were a native concept available that is equivalent with the concept in the message, the hearer would still adopt the same message interpretation as in NCP-1. In this case, the equivalent native concept would also be the most specific implied native concept. If there are only native concepts available that are more general than the concept in the message (implied concepts), the hearer interprets the message as the most specific implied concept. If there are multiple most specific implied native concepts, for example when an acquired concept has two superconcepts that overlap each other, the hearer interprets the message as both of these concepts. Some examples are given below.

Example 4.2. Consider the agents Ag-1 and Ag-2 with ontologies O_1 and O_2 respectively (Figure 4.3)

1. Ag-1 intends to convey **c**; in message composition, Ag-1 "translates" **c** to the equivalent shared concept **c**; in message interpretation, Ag-2 translates **c** to the most specific implied native concept **i**.
2. Ag-1 intends to convey **e**; Ag-1 cannot compose the message as there is no shared concept that is equivalent with **e**; Ag-1 switches to CDP to teach concept **e** to Ag-2; After the agents have visited the lower layers in the protocol, Ag-2's ontology is extended with an acquired concept **e** that is a subset of the native concept **k**; because Ag-1 now regards concept **e** as shared, it resumes message composition and "translates" **e** to the equivalent shared concept **e**; in message interpretation, Ag-2 translates **e** to the most specific implied native concept **k**.

The first example shows that the problem of Ag-2 to interpret concept **c** is solved. Because Ag-2 has no native concept that is equivalent with **c**, it interprets it as the most specific implied native concept **i**. From the viewpoint of Ag-2, there is no information loss, i.e. communication is lossless.

The second example reveals a problem of NCP-2. Although the agents succeed in establishing lossless communication, they fail to do this in a lazy way. Observe that Ag-2 has interpreted concept **e** in the same way as it would interpret concept **d**, namely as native concept **k**. Therefore, lossless communication of concept **e** would also be possible using shared concept **d**, which would save the agents the effort of sharing concept **e**. Consequently, this draft of NCP also fails to give rise to a minimal shared ontology, i.e. the concept **e** becomes needlessly shared.

To maintain a minimal shared ontology, the agents should maximally exploit a minimal amount of shared concepts. They should make use of a shared concept as much as possible. In particular, a shared concept should allow for communication in both directions, which is not the case in NCP-2. For example, Ag-1 can use shared concept **c** in communication with Ag-2, but Ag-2 is not allowed to use concept **c** in communication with Ag-1, for example to convey concept **j**. We deal with this problem in the next draft of NCP.

NCP-3: Approximate message composition

This draft of NCP (Figure 4.7) introduces approximate message composition. This achieves that the speaker can use concepts in a message that are more general than what it actually intends to convey. The main difference with this protocol is that message composition is defined less strictly than in the previous draft. The speaker is allowed to use shared concepts that are more general than the concept that it intends to convey. However, when more than one shared implied concept

---

Composition

- The agent translates what it intends to convey to the most specific implied shared concept.

Interpretation

- The agent translates the concept in the message to the most specific implied native concept.

Switch to CDP

- The speaker switches to CDP when it cannot compose its message, i.e. when no implied shared concept is available in its ontology.

Figure 4.7: NCP-3

is available, it must select the most specific one, thereby minimizing the amount of information loss. Because of this change, the condition when to switch to CDP is also changed. The speaker switches to CDP when no implied shared concepts are available. Some examples are given below.

Example 4.3. Consider the agents Ag-1 and Ag-2 with ontologies O_1 and O_2 respectively (Figure 4.3)

1. Ag-1 intends to convey **e**; in message composition, Ag-1 translates **e** to the most specific implied shared concept **d**; in message interpretation, Ag-2 translates **d** to the equivalent native concept **k**.
2. Ag-2 intends to convey **j**; in message composition, Ag-2 translates **j** to the most specific implied shared concept **c**; in message interpretation, Ag-1 "translates" **c** to the equivalent native concept **c**.
3. Ag-1 intends to convey **g**; in message composition, Ag-1 translates **g** to the most specific implied shared concept **f**; in message interpretation, Ag-2 translates **f** to the most specific implied native concept $\top$.

The first example shows that the problem of the previous draft in Example 4.2.2 is solved. The second example shows that NCP-3 allows shared concepts to be used for communication in both directions, as desired.

The last example reveals a problem of this communication protocol. This conversation is not lossless. Our previous example 4.1.3 shows that, when concept **g** is losslessly communicated, the interpretation of Ag-2 is concept **l**, and not $\top$ as in this conversation. The problem of this protocol is that the condition when to switch to CDP is improper. In the extreme case, the agents would only share the concept $\top$, would communicate everything using this shared concept and would never switch to CDP. Obviously, this leads to an intolerable amount of information loss and conflicts with the objective of effective communication. The next draft of NCP aims at overcoming this problem.

NCP-4: Towards effectiveness

This draft of NCP introduces two measures to prevent overgeneralization. The first measure is performed by the speaker. If it translates the concept it intends to convey to an *equivalent* shared concept, it lets this be known by sending a message with the "ExactInform" performative. If it applies approximate message composition, i.e. the concept in the message is more general than what it intends to convey, it sends a message with the "Inform" performative. In case an "ExactInform" message is sent, communication proceeds as in NCP-2, and is guaranteed to be lossless.

In case an "Inform" message is sent, it is more difficult to guarantee lossless communication. The second measure to prevent overgeneralization, regards a criterion to distinguish between the lossless communications using "Inform" and the non-lossless ones. As illustrated by Example 4.3, the lossless cases of "Inform" (conversation 1 and 2) and the non-lossless ones (conversation 3), cannot be distinguished from the speaker's perspective. Therefore, the task of recognizing lossless communication is left to the hearer.

When the hearer is not certain that communication was lossless, it requests specification. When the speaker cannot convey more specific information due to the lack of shared concepts, the agent switches to CDP. Stated in this way, the decision to switch to CDP has to do with recognizing overgeneralization. The protocol is described in Figure 4.8.

Composition

- The agent translates what it intends to convey to the most specific implied shared concept.

Interpretation

- The agent translates the concept in the message to the most specific implied native concept.

Switch to CDP

- The hearer switches to CDP when the speaker cannot convey what it intends to convey in a message which is not overgeneralized. The hearer regards an inform message which mentions a concept γ as overgeneralized if native particularizations of γ exist (other than $\perp$).

Figure 4.8: NCP-4

The hearer recognizes overgeneralization by reasoning as follows. Upon hearing that an individual belongs to some concept, it knows that the individual is a member of every implied concept and that the individual is not a member of all concepts that are disjoint with the concept. This knowledge cannot be a symptom of overgeneralization. However, the hearer remains ignorant about the particularizations of the concept. This ignorance may be a symptom of overgeneralization. The rule for recognizing overgeneralization we propose here is: if there are native particularizations of the concept used in the message (other than $\perp$), communication may not have been lossless. Some examples are given below:

Example 4.4. Consider the agents Ag-1 and Ag-2 with ontologies O_1 and O_2 respectively (Figure 4.3)

1. Ag-1 intends to convey **e**; in message composition, Ag-1 translates **e** to **d**; in message interpretation, Ag-2 translates **d** to **k**; Ag-2 determines that communication was lossless, because no native particularizations of **d** exist.

2. Ag-1 intends to convey **g**; in message composition, Ag-1 translates **g** to **f**; Ag-2 regards the message as overgeneralized as it detects a native particularization of **f**, namely **l**; Ag-2 cannot request for specification, and switches to CDP; After Ag-1 has taught the concept **g** to Ag-2, Ag-2's ontology is extended with an acquired concept **g** that is equivalent with the native concept **l**; Ag-1 resumes message composition and sends an "ExactInform" message with **g**; in message interpretation, Ag-2 translates **g** to **l** and the conversation finishes.

3. Ag-2 intends to convey **j**; in message composition, Ag-2 translates **j** to **c**; in message interpretation, Ag-1 "translates" **c** to **c**; Ag-1 detects a native particularization of **c**, namely **d**; Ag-1 requests whether **d** holds; Ag-2 denies; Ag-1 determines that communication has been lossless and the conversation finishes.

The first example shows the correct recognition of lossless communication. The second example shows the correct recognition of an overgeneralized message. The agents solve the problem by switching to the lower layers of the protocol. The problem of the previous draft (Example 4.3.3) is thus solved. The last example shows a dialogue where Ag-1 could not guarantee lossless communication at first, but ensured itself of it by requesting for specificity.

An important consequence of the changes that are introduced in this draft of NCP is that the conveyance of information may no longer proceed in a one-shot fashion. Instead, the agents participate in a *dialogue* to convey the desired amount of information. We stress this point, as it supports our dialogue-based approach to ontology reconciliation.

The problem of NCP-4 is that it is not minimal in use. Although the agents correctly determine lossless communication, the conversation mechanism may lead to unnecessarily long dialogues. As shown by Example 4.4.3, every time that Ag-2 intends to convey **j** using shared concept **c**, Ag-1 requests specification (due to the presence of **d**), upon which Ag-2 denies. Especially when larger ontologies are involved, every time the speaker intends to convey a general concept, a cumbersome dialogue follows where the hearer requests for specification many times. The following draft of NCP aims at overcoming this problem.

NCP-5: Towards efficiency

To make communication more efficient, we make the following adjustments to the communication mechanism. Note that, in earlier drafts, we have described message composition as making a translation to the *most specific* shared implied concept. The aim of this measure was to prevent the speaker from becoming more general than necessary. However, it can also be used to enable the hearer to form a belief about what the speaker intended to convey and what it did not. In philosophy of language, such a derivation is known as a *conversational implicature*, described in Section 2.2.1. In Anemone, it works as follows: upon receiving a concept, the hearer knows that the speaker did *not* intend to convey the shared subconcepts of that concept, otherwise it would have spoken differently. It therefore considers it useless to request these particularizations. Besides, the hearer can derive additional knowledge. The hearer also knows that the speaker did *not* intend to convey any non-shared subconcepts of the shared subconcept of the concept used in the message. The protocol is described in Figure 4.9.

Composition

- The agent translates what it intends to convey to the most specific implied shared concept.

Interpretation

- The agent translates the concept in the message to the most specific implied native concept.

Switch to CDP

- The hearer switches to CDP when the speaker cannot convey what it intends to convey in a message which is not overgeneralized. The hearer regards an inform message which mentions a concept γ as overgeneralized if native particularizations of γ exist that are not
  - a shared subconcept of γ, or
  - a subconcept of a shared subconcept of γ

Figure 4.9: NCP-5

In this draft, the criterion that is maintained by the hearer to recognize overgeneralization is made more strict. Therefore, communication is made more efficient, as the hearer no longer unnecessarily requests for specification as in the previous draft. An example is given below:

Example 4.5. Consider the agents Ag-1 and Ag-2 with ontologies O_1 and O_2 respectively (Figure 4.3)

1. Ag-2 intends to convey **j**; in message composition, Ag-2 translates **j** to **c**; in

message interpretation, Ag-1 "translates" **c** to **c**; Ag-1 derives that communication has been lossless and the conversation finishes.

As shown by this example, Ag-1 does not request whether **d** holds, contrary to the previous draft (Example 4.4.3). This is because Ag-1 knows that Ag-2 could not have intended to convey **d**, because otherwise Ag-2 would have used this more specific concept in the message. Ag-1 also knows that Ag-2 could not have intended to convey native concept **e**, because **e** is a subconcept of **d**. This enables Ag-1 to know that communication has been lossless.

Until now, we have discussed how agents should deal with shared concepts that are common knowledge. As argued in Section 4.2.1, in environments with more than two agents, ontology negotiation also gives rise to unknowingly shared concepts. Obviously, unknowingly shared concepts can also be used for successful communication. A minimal ontology negotiation protocol should exploit this property. The following draft aims at incorporating this feature in the communication mechanism.

NCP-6: Dealing with unknown concepts

In this draft, in message composition, we also allow a speaker to translate to a non-shared concept (which may, in fact, be an unknowingly shared concept). NCP-6 is particularly useful for MAS's with more than two agents, because especially these systems give rise to unknowingly shared concepts.

The rule that is used in NCP-6 is that the speaker may still translate what it intends to convey to the most specific shared implied concept, but if there are more specific non-shared implied concepts present, it may also choose one of these. Once an agent has used an unknowingly shared concept in the message, both agents know of each other that they know the concept, and the concept becomes a shared concept. Because an agent is not allowed to speak non-shared implied concepts that are more general than a shared implied concept, the efficiency measures that make use of conversational implicatures (introduced in NCP-5) are still applicable. The choice to translate to a shared concept with the risk of being too general, or to translate to a more specific non-shared concept, with the risk of being not understood, is left to the agent. The protocol is described in Figure 4.10.
In this draft of NCP, message composition is made less strict, providing the speaker with different alternatives to choose from. Among these alternatives are shared concepts, unknowingly shared concepts, and non-shared concepts.

Example 4.6. Consider the agents Ag-1 and Ag-2 with ontologies O_1 and O_2 respectively (Figure 4.3)

1. Ag-2 intends to convey **m**; in message composition, Ag-2 "translates" **m** to the non-shared concept **m**; in message interpretation, Ag-1 translates **m** to **h**; because the message was sent using "ExactInform", Ag-1 determines lossless communication and the conversation finishes. After this conversation has finished, both agents regard **m** as a shared concept.

Composition

- The agent translates what it intends to convey to:
  - the most specific shared implied concept, or
  - an implied concept that is more specific than the most specific shared implied concept

Interpretation

- The agent translates the concept in the message to the most specific implied native concept.

Switch to CDP

- The hearer switches to CDP when the speaker cannot convey what it intends to convey in a message that is understood by the hearer and which is not overgeneralized. The hearer regards an inform message which mentions a concept γ as overgeneralized if native particularizations of γ exist that are not
  - a shared subconcept of γ, or
  - a subconcept of a shared subconcept of γ

Figure 4.10: NCP-6

2. Ag-2 intends to convey **m**; in message composition, Ag-2 translates **m** to the shared concept $\top$; in message interpretation, Ag-1 translates $\top$ to $\top$; Ag-1 detects overgeneralization and requests for specificity; Ag-2 resumes message composition and "translates" **m** to **m**; in message interpretation, Ag-1 translates **m** to **h**; the conversation finishes.

3. Ag-1 intends to convey **e**; in message composition, Ag-1 translates **e** to the shared concept **d**; in message interpretation, Ag-2 translates **d** to **k**; Ag-2 recognizes lossless communication and the conversation finishes.

4. Ag-1 intends to convey **e**; in message composition, Ag-1 "translates" **e** to the non-shared concept **e**; Ag-2 does not know the concept and switches to CDP; after Ag-1 has taught the concept **e** to Ag-2, concept **e** becomes shared; Ag-1 resumes message composition and "translates" **e** to the shared concept **e**; in message interpretation Ag-2 translates **e** to **k**, recognizes lossless communication and the conversation finishes.

The first two conversations show that it may be beneficial to speak a relatively specific non-shared concept (as in the first conversation) rather than a more general shared concept (as in the second conversation). Although in the first conversation the same information is conveyed as in the second conversation, the first conversation is shorter and therefore more efficient.

The contrary also holds, as is revealed by the last two conversations. In the third conversation, the speaker speaks a relatively general shared concept, resulting in effective communication, as desired. In the fourth conversation, the speaker speaks a more specific non-shared concept, which is not understood by the hearer, and results in a switch to CDP. This dialogue is not desirable, as it conflicts with the objective of laziness.

Because it is not possible to decide beforehand whether it is best to speak specific non-shared concepts or general shared concepts, this choice is left open in the protocol. Communication strategies that resolve the non-determinism in the communication protocol are investigated in Chapter 6.

4.3.2 Concept Definition Protocol

In the concept definition protocol, the speaker tries to convey the meaning of a concept by stating the relations with other concepts, i.e. it speaks a number of concept definitions. If these definitions enable the hearer to derive the *complete meaning* of the concept, the hearer switches back to NCP. If there are not sufficient shared concepts available to convey the complete meaning, the agents switch to CEP. An agent regards the meaning of an acquired concept complete, if it knows the relation with every other concept in its ontology. The protocol is described in Figure 4.11.

Composition

- The agent states the relations (i.e. $\sqsubset$, $\sqsupset$, $\equiv$, $\perp$ and $\oplus$) of what it intends to define with other shared concepts in its ontology.

Interpretation

- The agent adds the defined concept as an acquired concept to its ontology.
- The agent adds the relations in the message to its ontology, and derives relations that follow from this.

Switch to NCP

- When the hearer knows the relation of the defined concept with every native concept in its ontology.

Switch to CEP

- When the speaker does not know any shared concepts to define concept relations.
- When the hearer does not know the relation of the defined concept with every native concept in its ontology.

Figure 4.11: CDP

Example 4.7. Consider the agents Ag-1 and Ag-2 with ontologies O_1 and O_2 respectively (Figure 4.3)

1. Ag-1 intends to define **e**; in message composition, Ag-1 defines "$\mathbf{e} \sqsubset \mathbf{d}$"; this definition enables Ag-2 to derive that **e** is disjoint with **f**,**l**,**n**,**m**, that **e** is subconcept of $\top$,**i**,**c**,**j**,**k**,**d**, and that **e** is a superconcept of $\perp$. It therefore regards the meaning of **e** as complete and switches back to NCP.

2. Ag-2 intends to define **j**; in message composition, Ag-2 defines "$\mathbf{d} \sqsubset \mathbf{j} \sqsubset \mathbf{c}$"; this definition enables Ag-1 to derive that **j** is disjoint with **f**,**g**,**n**,**h**,**m**, that **j** is subconcept of $\top$,**c**, and that **j** is a superconcept of **d**,**e**,$\perp$. It therefore regards the meaning of **j** as complete and switches back to NCP.

3. Ag-1 intends to define **g**; in message composition, Ag-1 defines "$\mathbf{g} \sqsubset \mathbf{f}$"; this definition enables Ag-2 to derive that **g** is disjoint with **i**,**c**,**j**,**k**,**d**,**n**,**m** and that **g** is a subconcept of **f**,$\top$, but it can not derive the relation with **l**; it therefore regards this meaning as incomplete, and switches to CEP.

The first two examples show the successful application of the concept definition protocol to establish lazy ontology reconciliation. The speaker succeeds in conveying the complete meaning of an unknown concept by giving a concept definition. Thus, the speaker has obviated the need to visit CEP. The last example shows that the concept definition protocol enables the hearer to correctly recognize an incomplete meaning. It determines that concept definitions provide insufficient means to convey the complete meaning of a concept, and it switches to CEP.

Note that the first example links up to the earlier Example 4.6.4. In this example of NCP, it was demonstrated that in message composition, a translation to a non-shared concept might turn out to be disadvantageous for the objective of laziness. In the example presented here (4.7.1), it is shown that these losses are limited by the concept definition protocol. The agents do not have to visit the CEP layer because a visit to CDP turns out to be sufficient to convey the complete meaning of the unknown concept.

Possible adjustments to CDP

A straightforward adjustment to this protocol would be along the lines of the changes carried through in NCP-6. This adjustment allows the speaker to compose a message using non-shared concepts. In the context of concept definitions, this would be a definition of an unknown concept in terms of other non-shared and possibly unknown concepts.

Whereas the purpose of the CDP and CEP layers is to resolve the misunderstanding regarding unknown concepts, this version of CDP might actually introduce additional misunderstandings regarding unknown concepts. For this reason, we have not incorporated this idea in CDP of the standard version of ANEMONE. Nevertheless, this adjustment might improve the performance of CDP, as is illustrated by the following example.

Example 4.8. Consider the agents Ag-1 and Ag-2 with ontologies O_1 and O_2 respectively (Figure 4.3)

1. Ag-1 intends to define **h**; in message composition, Ag-1 defines "$\mathbf{m} \sqsubset \mathbf{h} \sqsubset \mathbf{n}$"; this definition enables Ag-2 to derive the complete meaning of **h** and switches back to NCP.

This example shows the successful use of the non-shared concepts **m** and **n** to define the meaning of **h**. As has been argued, unknowingly shared concepts occur when ontology negotiation is used in MAS's with more than two agents. Therefore, this adjustment would be particularly suited for larger MAS's.

4.3.3 Concept Explication Protocol

The purpose of CEP is to convey the meaning of a concept non-symbolically when no satisfactory definition of the concept in terms of other concepts can be given. The protocol introduced in this section conforms to the ideas described in Section 3.3.

In CEP, the speaking agent communicates a number of positive and a number of negative examples of the concept to be explicated. The hearer classifies these examples using the concept classifiers from its own ontology. For each native concept in its ontology, the hearer creates a *confusion matrix*, containing the following information (Kohavi and Provost, 1998):

	Classified as Negative	Classified as Positive
Actual Negative	a	b
Actual Positive	c	d

In this matrix, a is the number of negative examples of the explicated concept that were classified as negative; b is the number of negative examples that were classified as positive; c is the number of positive examples that were classified as negative; d is the number of positive examples that were classified as positive. This matrix can be used to calculate the true positive rate (TPR), and the true negative rate (TNR) as follows:

- $\text{TPR} = \frac{d}{c+d}$
- $\text{TNR} = \frac{a}{a+b}$

The true positive rate represents the proportion of positive examples that have been positively classified as belonging to a concept. The true negative rate represents the proportion of negative examples that have been classified as not belonging to a concept. In case the number of positive examples is zero, we assume that the value for TPR equals 1. Likewise, when the number of negative examples is zero, the TNR value is assumed to be 1.

In the ideal situation, when the agent's classifiers are perfect and every element in the domain of discourse is used as an example, TPR and TNR offer strict criteria for determining the correct concept relation. These criteria are described next.

CEP under ideal circumstances

Consider an ideal situation, where TPR and TNR result from applying a classifier of concept γ_1 to examples of concept γ_2. The relation between concept γ_1 and γ_2 can be determined as follows:

- $\gamma_1 \equiv \gamma_2$ if TPR = 1 and TNR = 1
- $\gamma_1 \perp \gamma_2$ if TPR = 0
- $\gamma_1 \sqsubset \gamma_2$ if 0 < TPR < 1 and TNR = 1
- $\gamma_1 \sqsupset \gamma_2$ if TPR = 1 and TNR < 1
- $\gamma_1 \oplus \gamma_2$ if 0 < TPR < 1 and TNR < 1

The first condition states that γ_1 is equivalent with γ_2 if all members of γ_2 are positively classified as γ_1, and all non-members of γ_2 are negatively classified as γ_1. The second condition states that γ_1 is disjoint with γ_2 if all members of γ_2 are negatively classified as γ_1. The third condition states that γ_1 is a subconcept of γ_2 if some, but not all, members of γ_2 are positively classified as γ_1 and all non-members of γ_2 are negatively classified as γ_1. The fourth condition states that γ_1 is a superconcept of γ_2 if all members of γ_2 are positively classified as γ_1 and some non-members of γ_2 are positively classified as γ_1. The last condition states that γ_1 overlaps with γ_2 if some, but not all, members of γ_2 are negatively classified as γ_1 and some non-members of γ_2 are positively classified as γ_1.

The following example illustrates this method for concept explication under ideal circumstances.

Example 4.9. Consider the ontologies in Figure 3.2 and the number agents introduced in Example 3.3; assume that the agents' domain of discourse consists of the natural numbers between 1 and 20.

1. Ag-2 intends to explicate *Nat-number*; in message composition, Ag-2 composes a message where all numbers between 1 and 20 are given as positive examples, and no number as negative example; in message interpretation, Ag-1 uses its classifier for *Number*, its classifier for *Odd-number* and its classifier for *Even-number* on all examples.
   Since the classifier for *Number* classifies all examples positively, the value for TPR = $\frac{20}{0+20}$ = 1; the value for TNR = 1 because there are no negative examples. Ag-1 thus assesses that *Number* ≡ *Nat-number*.

2. Ag-1 intends to explicate *Odd-number*; in message composition, Ag-1 composes a message where all odd numbers between 1 and 20 are given as positive examples, and all even numbers between 1 and 20 as negative examples; in message interpretation, Ag-1 uses its classifiers for *Nat-Number*, *Prime*, *One-digit-prime* and *Multi-digit-prime* on all examples.

For the concept *Multi-digit-prime*, Ag-1 positively classifies the positive examples 11, 13, 17 and 19; therefore $d = 4$ and $c = 6$ and TPR = 0.4. Ag-1 positively classifies none of the negative examples; therefore $b = 0$ and $a = 10$, and TNR = 1. It thus derives that *Multi-digit-prime* $\sqsubset$ *Odd-number*
For the concept *Prime*, Ag-1 positively classifies positive examples 3, 5, 7, 11, 13, 17, 19; hence $d = 7$ and $c = 3$, TPR = 0.7. Ag-1 positively classifies the negative example 2; therefore $b = 1$ and $a = 9$, and TNR = 0.9. It thus derives that *Prime* $\oplus$ *Odd-number*.
In the same fashion, Ag-1 derives that *One-digit-prime* $\oplus$ *Odd-number* and *Nat-number* $\sqsupset$ *Odd-number*.

In the ideal situation, with a finite domain of discourse, and with disregard to computational costs, this concept explication protocol guarantees correct results. However, in most practical situations, the classifiers of the agents are not perfect. Furthermore, the domain of discourse may be infinite, in which case it is not possible to exchange every element in the domain of discourse. In this case, these criteria for determining concept relationships are too strict. In the next section, we show how they can be loosened.

CEP under noisy circumstances

There are actually many ways to loosen the criteria for determining concept relations. We follow a simple approach by introducing some threshold parameters θ_1, θ_2 and θ_3. Using these parameters, the TPR and TNR values can be divided into five regions that correspond to different concept relations as done in Figure 4.12. Note

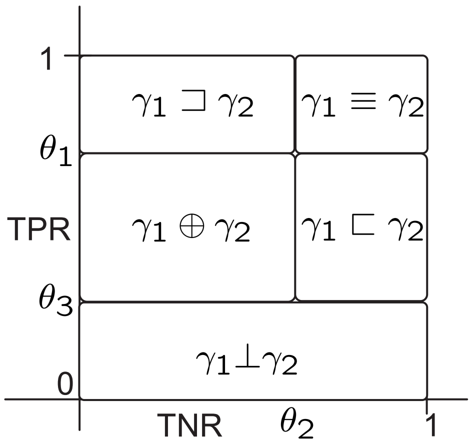


Figure 4.12: CEP boundaries

that, by using values $\theta_1 = 1$, $\theta_2 = 1$ and $\theta_3 = 0$, we obtain the same strict criteria for ideal concept explication described in the previous section. By using a looser set of values (i.e. $\theta_1 < 1, \theta_2 < 1$, $\theta_3 > 0$), the effects of noisy concept classifiers can be taken into account. Figure 4.13 describes the concept explication protocol for noisy circumstances.

Composition

- The agent sends positive and negative examples. The positive examples are individuals belonging to the concept to be explicated. The negative examples are individuals that belong to concepts that are disjoint with the concept to be explicated.

Interpretation

- For every native concept γ_1 in its ontology, the agent computes TPR and TNR using the examples in the message.
- The agent assesses the relation of the explicated concept γ_2 with every native concept γ_1 as follows:
  - $\gamma_1 \equiv \gamma_2$ if TPR $\geq \theta_1$ and TNR $\geq \theta_2$
  - $\gamma_1 \perp \gamma_2$ if TPR $\leq \theta_3$
  - $\gamma_1 \sqsubset \gamma_2$ if $\theta_3 <$ TPR $< \theta_1$ and TNR $\geq \theta_2$
  - $\gamma_1 \sqsupset \gamma_2$ if TPR $\geq \theta_1$ and TNR $< \theta_2$
  - $\gamma_1 \oplus \gamma_2$ if $\theta_3 <$ TPR $< \theta_1$ and TNR $< \theta_2$

Switch to CDP

- After the hearer has processed the examples.

Figure 4.13: CEP

The more strict the threshold parameters are set, the more often an agent determines the concept relation $\oplus$. This has a number of consequences for the future interactions between the agents. With regard to message interpretation, an acquired concept that overlaps with most of the agent's native concepts can be said to be less meaningful than an acquired concept that is subset, equivalent or disjoint with most native concepts. This is because a concept that overlaps with a native concept does not enable the agent to derive any positive or negative information about the native concept. On the other hand, an acquired concept that is subset or equivalent with a native concept, enables the agent to derive positive information about the native concept. A concept that is disjoint with a native concept enables the agents to derive negative information about the native concept.

Thus, the agents interpret messages as containing more information when they use loose threshold parameters in concept explication, than when they use strict threshold parameters. A loose set of thresholds also increases the number of incorrect message interpretations, as agents might derive too much information from their acquired concepts. In information retrieval, such a trade-off between the correctness and the quantity of information transfer is characterized by the measures *precision* and *recall* (Robertson, 2000). Precision measures how much of the information that is transferred is actually correct. Recall measures how much of the correct information is actually transferred. Stated in these terms, a loose set of threshold values leads to interactions with low precision, but with high recall. A strict set of threshold values leads to interactions with high precision, but with low recall.

Possible Adjustments to CEP

As mentioned before, the concept explication protocol described in Figure 4.13 is very elementary. Depending on the particular application, the concept explication protocol might require changes. We will discuss two possible adjustments below.

The first adjustment concerns the conditions for determining the $\perp$ and $\sqsubset$ relation. As depicted in Figure 4.12, the lower right region where TPR $\leq \theta_3$ and TNR $\geq \theta_2$ is allocated to the $\perp$ relation. The motivation for this is that the small amount of positively classified positive examples can be the result of misclassifications. One might object to this by arguing that this region should be allocated to the $\sqsubset$ relation. The motivation for this would be that the positively classified positive examples indicate a subset relation and that the small amount of positively classified negative examples are the result of misclassifications. There is no general way to decide what the correct way of handling such situations is. The choice between the two possibilities should be motivated with practical experience with the concept classifiers and the nature of the objects in the domain of discourse.

Another adjustment to CEP that may be desirable aims at preventing contradictory sets of concept mappings to be established. For example, suppose that Ag-1 with ontology O_1 (Figure 4.3) teaches concept $\mathbf{g}$ to Ag-2 with ontology O_2. Suppose that Ag-2 determines that $\mathbf{g} \equiv \mathbf{l}$ and that $\mathbf{g} \sqsubset \mathbf{m}$. These concept relations contradict each other as Ag-2's native ontology states that $\mathbf{l} \perp \mathbf{m}$. Ag-2 must solve this conflict by adopting the concept relation with the highest *confidence factor*, i.e. the relation about which it is most confident. One way to implement the confidence factor is by comparing the TPR and TNR values with the thresholds of the ideal situation ($\theta_1 = 1$, $\theta_2 = 1$, $\theta_3 = 0$). The closer these values are to the ideal thresholds, the higher the confidence factor is. For example, suppose that TPR=1 and TNR=1 for concept $\mathbf{l}$. The confidence factor for the relation $\mathbf{g} \equiv \mathbf{l}$ is high, because the values match the ideal threshold values. For concept $\mathbf{m}$, TPR=0.7 and TNR=0.7. The confidence factor for the relation $\mathbf{m} \sqsupset \mathbf{g}$ is low, because the value for TPR differs from the ideal threshold value 1 with 0.3. Thus, Ag-2 would determine that $\mathbf{g} \equiv \mathbf{l}$ and not $\mathbf{g} \sqsubset \mathbf{m}$. Of course, there is much more to be said about the confidence factor. Again, a deeper analysis of this issue would require hands-on experience in a practical domain.

4.3.4 Message Protocol

Until now, we have only discussed the general principles that underlie the communication mechanism. To establish successful agent communication, a protocol is required which specifies which messages can be sent at which times. The message protocol of ANEMONE is defined using a finite state machine, depicted in Figure 4.14. In this protocol, the Ag-i is the speaker and Ag-j is the hearer.

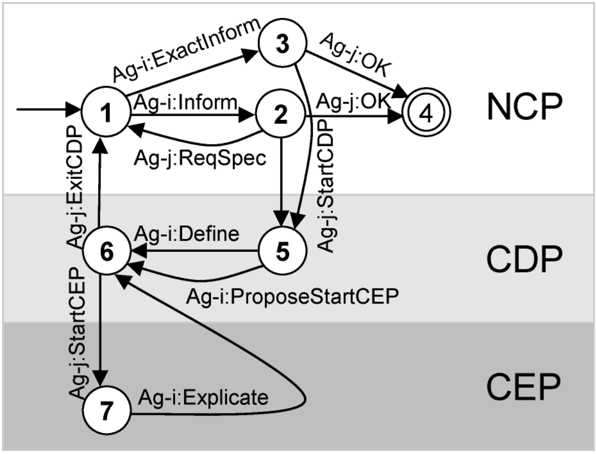


Figure 4.14: Message protocol

The agents start their conversation in state 1, where the speaker composes a message. In case the speaker translates what it intends to convey to an equivalent concept, it sends an "ExactInform" message. Otherwise, it sends an "Inform" message. The hearer interprets the message in state 2 or 3. In case it does not understand the concept used in the message, it sends a "StartCDP" message. In case it regards the message as overgeneralized, which only occurs in state 2, it requests for specification by sending a "ReqSpec" message. In case the hearer understands the message and does not regard it as overgeneralized, it answers "OK", after which the conversation finishes in state 4.

The concept definition protocol starts in state 5. In case the speaker does not have any shared concepts to compose a concept definition, it sends a "ProposeStartCEP" message. Otherwise, it sends a "Define" message. In state 6, the hearer interprets the concept definition. It sends an "ExitCDP" message when it regards the definition as complete and sends a "StartCEP" message when it regards the meaning as incomplete.

In state 7, the speaker explicates the concept by sending an "Explicate" message. After this, the agents return to the CDP layer. In state 6 the hearer sends an "ExitCDP" message and the agents return to the NCP layer.

4.4 Using ANEMONE in Larger Communities

So far, we have mainly discussed ANEMONE in the context of two agents. In principle, ANEMONE can be used in multi-agent systems of arbitrary size without requiring any changes to the protocols. Nevertheless, the application of the technique in larger communities gives rise to some phenomena we have not explicitly mentioned yet,

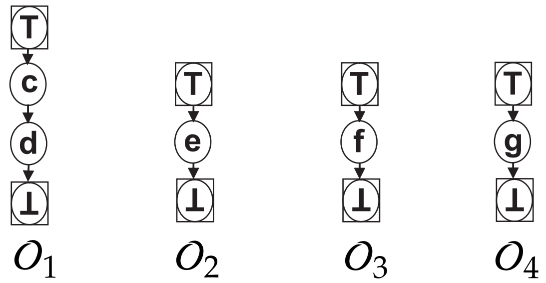


Figure 4.15: Initial ontologies

namely the explication of acquired concepts and the handling of unknown concept relations. Both phenomena are described below.

4.4.1 Acquired concept explication

Concept explication requires the speaker to provide positive and negative examples of the concept to be explicated. When explicating a native concept, these examples are available in the agent's ABox and can be obtained via the agent's concept classifiers. For acquired concepts, the situation is different. The agent's ABox contains no assertions in terms of acquired concepts and the agent possesses no acquired concept classifiers either. The problem rises how the agents should explicate acquired concepts.

In a closed system with two agents, the agents never have to explicate an acquired concept to each other. This is because the only way that a concept can be acquired by one agent is through explication by the other agent. Therefore, acquired concepts are always shared and do not need to be explicated. However, in a situation with multiple agents, it might occur that an agent is requested to explicate an acquired concept, as is shown in the following example.

Example 4.10. Consider the agents Ag-1, Ag-2, Ag-3 and Ag-4 with ontologies $\mathcal{O}_1$, $\mathcal{O}_2$, $\mathcal{O}_3$ and $\mathcal{O}_4$ (Figure 4.15)

1. Ag-3 intends to convey **f** to Ag-4; the conversation results in concept **f** being explicated to Ag-4 after which the ontology of Ag-4 is extended with the information that $\mathbf{g} \sqsubset \mathbf{f}$, as in $\mathcal{O}'_4$ (Figure 4.16)

2. Ag-4 intends to convey **g** to Ag-2; Ag-4 speaks **f**; Ag-2 does not understand **f**; the agents enter CEP.

At the end of the example, Ag-4 is requested to resolve Ag-2's ignorance of concept **f**. As we have argued, it is not straightforward for Ag-4 to find examples of the acquired concept **f** that it can use for concept explication. We describe three possibilities to deal with these issues below.

The first possibility is to avoid explication of acquired concepts. In this way, Example 4.10 would end with Ag-4 explicating **g** to Ag-2 instead of **f**. As this is a simple and effective solution to the problem, we have adopted it in our standard version of Anemone. Nevertheless, this solution may not be optimal for acquiring a minimal communication vocabulary. For example, suppose that it would turn out that **g** $\sqsubset$ **e** and that **f** $\sqsubset$ **e**. Then, it would be preferable for Ag-2 to learn concept **f** instead of **g**, because **f** is also usable for communication with Ag-3 (considerations of this type are investigated in Chapter 6).

A second possibility is to check which native concepts are related with the acquired concept and use members of these native concepts as examples of the acquired concept. In case there is a native concept which is equivalent with the acquired concept, the situation is easy. The examples that characterize the native concept are the same as those that characterize the acquired concept. Otherwise, the agent must use the positive examples of the native subconcepts as positive examples and the negative examples of the native superconcepts as negative examples. For example, for Ag-4 to explicate **f** it can use the positive examples of concept **g** and the negative examples of concept $\top$. Of course, there are no negative examples of concept $\top$. This illustrates a possible shortcoming of this approach, i.e. that it provides for too few examples to accurately convey the meaning of a concept. Nevertheless, it should be up to the hearer to decide whether the concept explication was successful or not. In some cases, the examples that are obtained by regarding native sub- and superconcepts are sufficient for the hearer to acquire the complete meaning of a concept.

A third possibility to deal with the difficulties of explicating acquired concepts, is to make an agent remember how it originally learned the concept. In this way, an agent stores the positive and negative examples of a concept that is explicated to it. Later, when it passes the acquired concept on to another agent, it uses the same examples for explicating the acquired concept. This solves the problem of finding appropriate examples of acquired concepts. A disadvantage of this method may be that it can be costly to store all examples that have once been used in concept explication. Furthermore, the individuals that were shown to the agent when it learned the concept, may not be accessible anymore at the time the agent teaches the concept to another agent.

4.4.2 Unknown concept relations

Another phenomenon that occurs in systems with multiple agents is that of unknown concept relations. The following example shows how ontologies might evolve in such a way that an agent is left ignorant about a concept relation.

Example 4.11. Consider the agents Ag-1, Ag-3 and Ag-4 with ontologies O_1, O_3 and O_4 (Figure 4.15)

1. Ag-3 intends to convey **f** to Ag-1; the conversation results in concept **f** being explicated to Ag-1

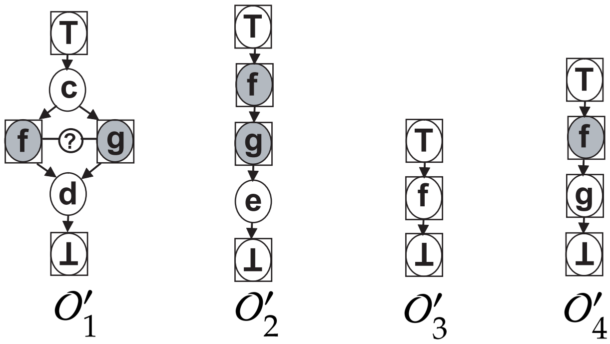


Figure 4.16: Evolved ontologies

2. Ag-4 intends to convey **g** to Ag-1; the conversation results in concept **g** being explicated to Ag-1 after which the ontology of Ag-1 becomes O'_1 (Figure 4.16)

In this example, Ag-1 is ignorant about the relation between **f** and **g**. This is because Ag-1 does not have a classifier for **f** which prevented it from assessing the correct concept relation when it was given examples of **g**.

Note that this phenomenon only occurs under specific circumstances. Usually, the relations with acquired concepts follow from the relations with native concepts. For example, suppose that Ag-2 (with O_2) explicates concept **e** to Ag-1 (with O'_1), upon which Ag-1 derives that $\mathbf{e} \sqsubset \mathbf{d}$. In this case, Ag-1 is able to derive the subconcept relation of **e** with the acquired concepts **f** and **g**.

Another way that agents might convey the relations with acquired concepts is by using the concept definition protocol. This is illustrated in the following example.

Example 4.12. Consider the agents Ag-2, Ag-3 and Ag-4 with ontologies O_2, O_3 and O_4 (Figure 4.15)

1. Ag-3 intends to convey **f** to Ag-4; the conversation results in concept **f** being explicated to Ag-4.

2. Ag-3 intends to convey **f** to Ag-2; the conversation results in concept **f** being explicated to Ag-2.

3. Ag-4 intends to convey **g** to Ag-2; Ag-2 requests for a concept definition of **g**; Ag-4 responds that $\mathbf{g} \sqsubset \mathbf{f}$; Ag-2 regards the meaning incomplete and starts CEP; Ag-4 explicates the concept; Ag-2 uses the examples to assess the concept relations other than $\mathbf{g} \sqsubset \mathbf{f}$; Ag-2's ontology becomes O'_2

This example illustrates that in systems with multiple agents, the concept definitions can be used to convey concept relations that cannot be conveyed using concept explication.

Nevertheless, Example 4.11 has revealed that agents are sometimes left ignorant about concept relations between acquired concepts. This need not be a serious problem, as agents are primarily interested in native concepts. A subtle issue rises when the agents want to use these acquired concepts in their messages. This is illustrated by the following example.

Example 4.13. Consider the agents Ag-1 and Ag-2 with ontologies O'_1 and O'_2 (Figure 4.16)

1. Ag-1 intends to convey **d**; in message composition Ag-1 translates **d** to **f**; in message interpretation, Ag-2 translates **f** to $\top$; Ag-2 derives that communication has been lossless; the conversation finishes.

In the dialogue in this example, Ag-2 has incorrectly assessed lossless communication. The message with concept **f** was overgeneralized, as **d** could be translated to **e** instead of $\top$. Ag-2 did not recognize this, because the shared concept **g** was more general than **e** and more specific than **f**. This problem occurred because Ag-1 did not know that **g** was a subconcept of **f**. In fact, Ag-1 did not know if **f** was the most specific shared implied concept (the condition for message composition in NCP-6). Therefore, the dialogue in Example 4.13 is not allowed according to NCP-6. The example does not indicate a flaw in NCP-6, but rather that its conditions should be read very carefully.

4.5 Discussion

4.5.1 The Role of Symbolic Communication

Anemone enables agents to overcome symbolic communication problems by building up a shared communication vocabulary. One may observe that the things that are symbolically communicable after the cv is built up were already non-symbolically communicable before the cv was built up. The critic could interpret this observation as evidence that the whole communication protocol is useless. The critic would raise: to inform that individual a is member of concept **c**, why not just send a to the hearer, and let the hearer classify this individual in its own ontology? Apparently, this would solve the problem that concept **c** might not be understandable to the hearer. We will give three reasons why this objection does not hold.

Firstly, the proposal of the critic requires the hearer to use its classifiers. Computationally, this is more costly than symbolic communication. For example, repeatedly using the *prime* classifier from Example 3.3 might be practically unfeasible due to high computation costs. Seen from this perspective, symbolic communication serves a purpose of computational efficiency. Secondly, non-symbolic communication as proposed by the critic requires the individual to be accessible for the hearer at the time of communication. This is not always the case. For example, a sender intends to convey that it possesses a news article, but is only willing to show the article in exchange for money. Thirdly, the proposal of the critic only applies to *inform* messages. In our discussion we have only considered *inform* messages in the NCP layer. For most applications, however, the NCP layer should be extended to deal with other performatives as well, e.g. *request*. In ANEMONE, such extensions are possible. However, it is not clear how the critic's approach can be extended to deal with *request* messages.

The persistent critic might respond to this: "In ANEMONE, symbolic communication plays a marginal role as it does not add much to the expressiveness provided by non-symbolic communication. It may have some computational benefits but it does not contribute substantially to what is already present in the system." We can only respond to this by arguing that, in ANEMONE, meanings do not miraculously enter the system. The expressiveness that is provided by symbolic communication has resulted from the expressivity of non-symbolic communication. Because agents have access to the meanings of their concepts, they can communicate at two levels: symbolically and non-symbolically. Symbolic communication proceeds in NCP by exchanging symbols with an agreed upon meaning. Non-symbolic communication proceeds in CEP by exchanging the meanings themselves. The availability of two levels of communication does not place symbolic communication in a marginal position. It places the symbol system in the *right* position with a grounding in a non-symbolic system. For ontology negotiation, and probably for most AI systems, such a non-symbolic context is necessary.

4.5.2 ANEMONE for Hybrid Ontology Reconciliation

The layered structure of ANEMONE allows for a reasonable amount of flexibility. In this section, we will discuss two hybrid approaches for ontology reconciliation in which ANEMONE is combined with other techniques.

Combination 1

As discussed in Section 2.4.5, it is always a good investment to standardize, when possible, parts of the agents' ontologies at design-time. Those parts of the ontologies that remain heterogeneous after the standardization effort can be aligned at agent interaction time using ontology negotiation. This is a hybrid approach to ontology reconciliation where standardization is combined with ontology negotiation. Such a hybrid approach also fits well within ANEMONE.

One possibility to start with a standardized ontology is to make the agents aware of this. In this case, the agents start with an ontology that partly consists of knowingly shared concepts, namely those stemming from the standardized ontology. The NCP layer of the protocol will effectively apply these concepts in communication. In case the agents intend to convey concepts that do not belong to the standardized ontology, the other layers in the protocol will prove themselves useful.

Another possibility to use standardized ontologies in ANEMONE is to namespace the concepts in these ontologies with the (unique) names of the standard ontologies. In this way, the ontologies of agents contain many unknowingly shared concepts.

The communication protocol ensures that these unknowingly shared concepts will be effectively applied in communication.

Combination 2

The concept explication method described in this chapter may not be satisfactory for all applications. The limitations of conveying the meaning of a concept by exchanging instances has been discussed in Section 3.3.1. Another concept explication method can be used by replacing the CEP layer of the communication protocol. This may involve another technique for ontology reconciliation, leading to a hybrid approach.

One possibility is to incorporate ontology mediation services in the system. Instead of switching to the CEP layer of the protocol, the agents consult the mediation service to find the appropriate ontology mapping. The mediation service may be implemented as a database of ontology mappings or may consist of human ontology engineers.

Another possibility is to use translation dictionaries in the CEP layer of the protocol. Although this approach is inadequate from a theoretical perspective (as argued in Section 3.3.3), it might achieve good results in practice.

Whichever replacement of CEP is chosen, it remains a resource-consuming activity. Therefore, the features of laziness and minimality that are incorporated in the upper layers of the protocol remain useful, even when a different concept explication method is chosen.

4.6 Conclusion

In this chapter, we have proposed Anemone as a way to solve semantic integration problems in multi-agent systems. The approach does not aim to solve all ontology problems at one stretch at design-time, neither does it presume the presence of a central coordinating agent. Anemone aims at providing agents with the tools to overcome ontology problems at agent interaction time. The agents find out themselves if their ontologies are insufficiently aligned and can resolve their misunderstandings together.

Anemone is designed as a layered communication protocol which establishes effective and minimal ontology negotiation. To enable effective communication, the agents tackle semantic integration problems when needed. Their approach to these problems is lazy, i.e. they refrain from applying resource-consuming machine learning techniques as much as possible. When necessary, the agents incrementally contribute to a shared ontology which is minimal in size and enables them to convey sufficient information.

To realize a minimal and effective ontology negotiation protocol, the agents must recognize when their ontologies are insufficiently aligned. After exploring several possibilities, we arrived at a protocol where overgeneralized messages are taken as indicators of insufficiently aligned ontologies. Furthermore, information exchange should be structured as a dialogue to enable agents to try out different ways of putting something into words.

4.6.1 Future Work

Besides proposing a technique for achieving semantic interoperability, this chapter introduces a way of thinking about ontology negotiation protocols. We have identified approximate message composition and approximate message interpretation as important aspects of an ontology negotiation protocol. This line of investigation can be pursued in several ways.

An interesting topic for future research is extending the dialogue mechanisms with other performatives, for example *query*. A simple way to implement *query* is as a request to the other agent to send an inform message. For example, Ag-2 (Figure 4.3) wants to know which individuals belong to concept **l**, and requests Ag-1 to inform about all members of concepts more specific than **f**. After this, ANEMONE establishes that Ag-1 makes inform statements that minimize information loss. However, a more direct way to implement *query* would be to judge the query messages themselves as giving rise to possible information loss. Interestingly, what qualifies as information loss in a query is the opposite of what qualifies as information loss in an inform message. In an inform message, the more general the concept is, the more information is possibly lost in the message. In a query message, the more specific the concept in the message is, the more information is possibly lost, as it restricts the number of possible answers to the query. A cleverly designed query protocol is needed to guide such dialogues.

Besides *query*, other performatives could raise similar issues. One of these is *disconfirm*. For example, an agent that disconfirms the availability of a *flight* does not necessarily disconfirm the availability of a more general concept *means-of-transport*. Another performative of interest is a *request* to another agent to perform some action. On some occasions, *request* behaves like *query*. For example, a request to *book* a *flight* can be interpreted as a request to *book* a more specific concept *Boeing-747-flight*, but not as a request to *book* a more general concept *means-of-transport*. On other occasions, however, *request* behaves like *inform*. For example, a request to *send advertisement* to every *flight-customer* can be interpreted as a request to send advertisement to every *travel-customer* (which is a more general concept), but not as a request to send advertisement to every *Boeing-747-flight-customer* (which is a more specific concept).

The informal style of presentation serves the purposes of this chapter well and provides a good starting point for future work. Nevertheless, a more rigorous treatment is required to formally verify the communication mechanism. This is the purpose of the next chapter, where we approach ontology negotiation from a formal perspective.

Chapter 5

Formal Protocol Analysis

> *"Do you then propose that we should give up mathematical logic?" said I, in fake amazement. "Quite the opposite. Logic formalizes only very few of the processes by which we actually think. The time has come to enrich formal logic by adding to it some other fundamental notions." [...] "Do not lose your faith" concluded Stan. "A mighty fortress is our mathematics. Mathematics will rise to the challenge, as it always has."*
>
> Stanislaw Ulam and Gian-Carlo Rota (Rota, 1985)

This chapter presents a formal analysis of ontology negotiation protocols. The chapter serves two purposes.

Firstly, it serves to enhance the ideas introduced in the previous chapter. By formalizing the framework, we can state the design objectives more precisely and provide a different perspective on the issues involved. By formalizing the communication protocols we can give solid proofs that the communication mechanisms possess the desirable properties. In doing so, we will restrict ourselves to the most important drafts of NCP, namely NCP-2 and NCP-5, and describe each of these in an integrated protocol with CDP and CEP.

Secondly, this chapter introduces some new ideas about ontology negotiation that could not be satisfactorily described in the informal style of Chapter 4. We will introduce a new communication protocol which enables the agents to remove superfluous concepts from their communication vocabulary. Furthermore, we will pay attention to the implementation of ontologies using description logic. This involves using an ontology language which enables the formation of composed concepts out of atomic concepts.

As a consequence of this, the workings of the communication mechanisms are sometimes formulated differently. For example, in Chapter 4 we might have written that **c** is a subconcept of **d** and **c** is disjoint with **e**, whereas in this chapter we would simply write that c is a subconcept of the composed concept $d \sqcap \neg e$. We will pay attention to such formulation differences to avoid possible confusion.

The chapter is organized as follows. Section 5.1 provides a formal framework. In Section 5.2, this framework is used to formally state the requirements of an ontology negotiation protocol. Section 5.3 proposes three protocols of increasing complexity. We will formally analyze their properties and judge them according to their requirements. We conclude in section 5.4.

The chapter is based on (van Diggelen, Beun, Dignum, van Eijk and Meyer, 2004, 2006b, 2007).

5.1 Ontologies

This section serves to deepen the treatment of ontologies in Section 4.2.1 by combining it with the ideas introduced in Chapter 3. We will first propose a formal account of ontology diagrams. Then, we will describe how different ontologies are known by the agents. Finally, we will describe how the ontologies can be implemented using description logics.

We restrict ourselves to dialogues between two agents. As a running example, we consider communication between a travel-agent Ag-1 with ontology O_1 and a car rental service Ag-2 with ontology O_2 (Figure 5.1). O_1 shows the expertise of

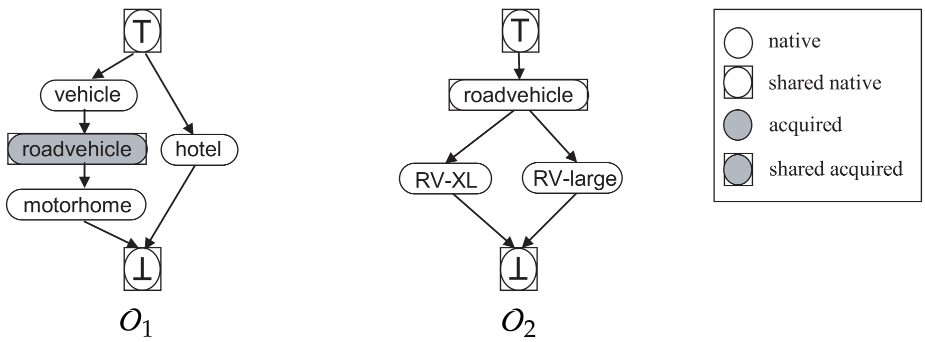


Figure 5.1: Ontologies of agents

Ag-1 on different sorts of accommodation and O_2 shows the expertise of Ag-2 on cars.

For the purposes of this chapter, the ontology diagram depicted in Figure 5.1 is insufficiently precise. This style of ontology representation does not specify the relations between concepts that are not yet discovered by the agents. For example, it is unclear what would happen if Ag-2 would teach the concept *RV-XL* to Ag-1. We introduce a different style of ontology representation below.

When considering communication between two agents, six ontologies can be regarded as relevant. O_1 and O_2 are the native ontologies of Ag-1 and Ag-2. O_{cv} (the communication vocabulary) is their shared ontology. $O_{1\cdot cv}$ is the mapping between Ag-1's ontology and the cv; $O_{2\cdot cv}$ is the mapping between Ag-2's ontology and the cv. The ontology that would arise if we would combine O_1 and O_2 is $O_{1\cdot 2}$.

Figure 5.2 shows these six ontologies in the initial situation when the cv is empty. The situation after Ag-2 has added the concept *roadvehicle* to the cv is shown in Figure 5.3. This figure represents the same situation as Figure 5.1. Conceptualizations are represented by Euler diagrams (Shin and Lemon, 2006), showing different conceptualizations (represented by circles) on the same domain. A circle that is included in another circle represents a subconcept relation (or conversely, a superconcept relation). An arrow from ontology O_x to O_y represents that O_y is included in ontology O_x. Figure 5.3 shows that *roadvehicle* has been added to O_{cv}, which enables this concept to be used in communication. The figure also shows that Ag-1 has learned the meaning of *roadvehicle* by representing the relations between *roadvehicle* and the concepts in its native ontology in $O_{1\cdot cv}$.

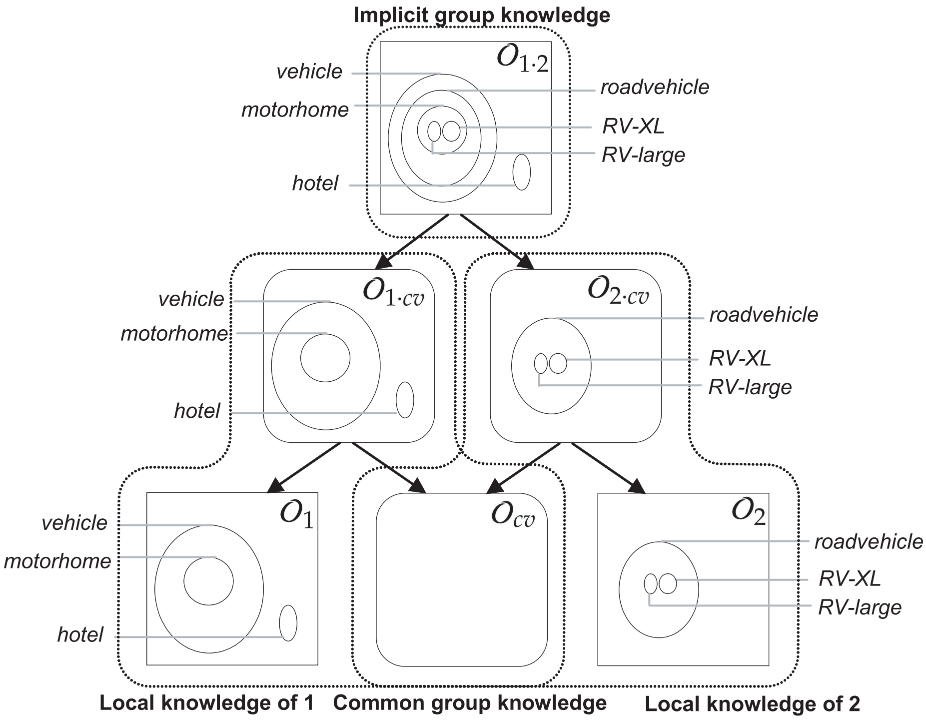


Figure 5.2: Ontologies of agents in the initial situation

As described in Section 3.2, we build our framework upon a domain of discourse ($\Delta^{\mathcal{I}}$) and a conceptualization (ρ). The names for the elements in the domain of discourse are given by the set Δ. We will focus on conceptualizations that consist

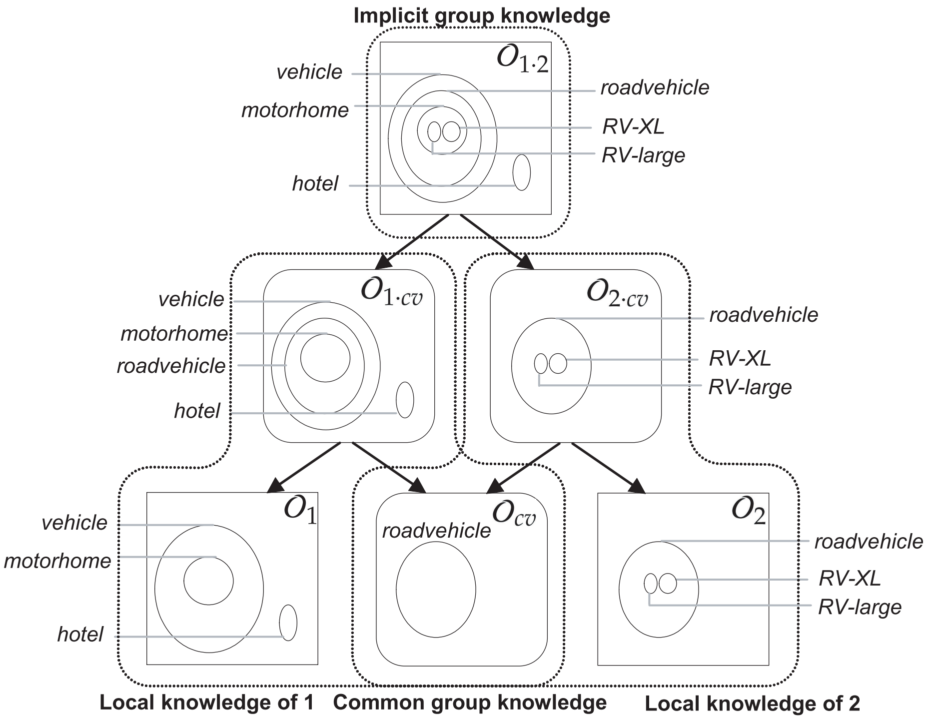


Figure 5.3: Ontologies of agents that share one concept

of sets of objects, i.e. $\rho \subseteq 2^{\Delta}$. Furthermore, as commonly done in AI systems (Sowa, 2000), we assume that the elements in ρ form a bounded lattice structure by considering the partial ordered set $(\rho, \subseteq)$. This means that $\Delta \in \rho$ (a maximal element, or top-concept), and that $\emptyset \in \rho$ (a minimal element, or bottom concept). Furthermore, for every two elements $x, y \in \rho$, $x \cap y \in \rho$ and $x \cup y \in \rho$. The elements in the conceptualization are not named, but only contain the meanings themselves. In the example figure, the elements in ρ are represented by circles. In our framework, different agents may adopt different conceptualizations.

To be able to formalize knowledge about the domain of discourse, the meanings in the conceptualization must carry a name. This is done by the ontology, which *specifies* the conceptualization (Gruber, 1993b). The ontology introduces a set of symbols C which, when interpreted under their intended interpretation, refer to the elements in the conceptualization (conforming to the treatment by Genesereth and Nilsson (1987)). We will refer to the intended interpretation function with $\mathcal{I}^{INT} : C \rightarrow \rho$. $\mathcal{I}^{INT}$ is a surjective function. This means that for every element $y \in \rho$, an element $x \in C$ exists for which $\mathcal{I}^{INT}(x) = y$, i.e. every element in the conceptualization is named. In the example, $\mathcal{I}^{INT}$ is represented by the gray horizontal lines that connect concept names to circles (their meanings).

For an ontology to *fully* specify the conceptualization, it would have to spell out the exact value of $\mathcal{I}^{INT}$ which would be unfeasible. Therefore, the ontology only specifies that aspect of $\mathcal{I}^{INT}$ which is most relevant for the agent, namely the subset ordering in ρ. An ontology is thus defined as $O = \langle C, \leq \rangle$ where $\leq \subseteq C \times C$ is a pre-order for which $\forall x, y \in C. x \leq y \Leftrightarrow \mathcal{I}^{INT}(x) \subseteq \mathcal{I}^{INT}(y)$. This states that an ontology specifies a conceptualization as a reflexive, transitive relation which is conforming to the subset ordering on the intended interpretations of the concepts. Note that, although the conceptualization has the anti-symmetry property, this property does not necessarily hold for the ontology. An ontology may specify multiple ways to refer to the same element in the conceptualization (i.e. synonyms). If two elements $x, y \in C$ have the same intended interpretation, it is not necessarily the case that $x = y$, as x may be syntactically different from y. We will write $x \equiv y$ as a shorthand for $x \leq y \wedge y \leq x$, and $x < y$ as a shorthand for $x \leq y \wedge \neg(y \leq x)$.

Definition 5.1. *Given the following ontologies:*

- $O_i = \langle C_i, \leq_i \rangle$ *(for $i \in \{1, 2\}$): The native ontology of Ag-i.*
- $O_{cv} = \langle C_{cv}, \leq_{cv} \rangle$ *: The communication vocabulary of the agents, where $C_{cv} \subseteq C_1 \cup C_2$.*

We define the ontologies:

- $O_{1\cdot 2} = \langle C_{1\cdot 2}, \leq_{1\cdot 2} \rangle$ *: A god's eye view over the ontologies in the system, where $C_{1\cdot 2} = C_1 \cup C_2$. Relation $\leq_{1\cdot 2}$ conforms to the subset ordering on the intended interpretations of the elements in $C_{1\cdot 2}$.*
- $O_{i\cdot cv} = \langle C_{i\cdot cv}, \leq_{i\cdot cv} \rangle$ *(for $i \in \{1, 2\}$): The native ontology of Ag-i and the communication vocabulary, where $C_{i\cdot cv} = C_i \cup C_{cv}$. Relation $\leq_{i\cdot cv}$ conforms to the subset ordering on the intended interpretations of the elements in $C_{i\cdot cv}$.*

We use a subscript notation whenever we need to stress that something belongs to O_i, O_{cv}, etc. For example, a concept with the name d_{cv} is assumed to be member of C_{cv}. The subscripts are omitted when no confusion arises. The following definition introduces some useful terminology:

Definition 5.2. *Given two concepts $c, d \in S$, a preorder $\leq: S \times S$ and a set $S' \subseteq S$*

- *c is a subconcept of d in S' iff $c \leq d$ and $c \in S'$*
- *c is a superconcept of d in S' iff $d \leq c$ and $c \in S'$*
- *c is a strict subconcept of d in S' iff $c < d$ and $c \in S'$*
- *c is a strict superconcept of d in S' iff $d < c$ and $c \in S'$*
- *c is most specific in the set S' if no strict subconcept of c in S' exists*
- *c is most general in the set S' if no strict superconcept of c in S' exists*

We will now discuss the ontologies introduced in 5.1 in more depth.

Native ontologies

The native ontology *by itself* does not represent any knowledge that is of practical use to the agent. Rather, it introduces a vocabulary which the agent uses to represent and reason with its assertional knowledge. An agent Ag-i stores this knowledge in its assertional knowledge base $\mathcal{A}_i$. We avoid naming conflicts between two native ontologies by assuming that the sets C_1 and C_2 are disjoint. This can be easily achieved by prefixing the concept names using namespaces.

God's eye view ontology

$O_{1\cdot 2}$ is the ontology that would arise if the native ontologies of the agents were combined. $O_{1\cdot 2}$ is a virtual ontology, i.e. it is not materialized in its totality anywhere in the system. For us, it is convenient to adopt this god's eye view over the ontologies to discuss the issues involved in ontology negotiation. From Definition 5.1, it follows that every other ontology in the system is included in this ontology, i.e. $C_i, C_{cv}, C_{i\cdot cv} \subseteq C_{1\cdot 2}$. Note that $\leq_{1\cdot 2}$ is not *equal* to $\leq_1 \cup \leq_2$, but that it is a *superset* (except in the hypothetical case when C_1 or C_2 is empty). This is because, as argued before, $\leq_{1\cdot 2}$ is conforming to the subset ordering on the intended interpretations of the concepts. It therefore also contains the relations between the elements of C_1 and C_2 which are not present in $\leq_1 \cup \leq_2$. For example, $\leq_{1\cdot 2}$ contains the relation *roadvehicle* $\leq$ *vehicle*, which is present in neither $\leq_1$, nor $\leq_2$.

Ontologies for alignment

The communication vocabulary O_{cv} indirectly aligns the agents' native ontologies. Ag-1 and Ag-2 maintain a mapping from their native ontologies to the communication vocabulary in respectively $O_{1\cdot cv}$ and $O_{2\cdot cv}$. This mapping states the relation between concepts in the communication vocabulary and concepts in the native ontology. It can be represented that a concept in the cv is *equivalent* to a concept in the native ontology, or that it is a *subconcept*, or a *superconcept*. From Definition 5.1, it follows that $C_{cv} = C_{1\cdot cv} \cap C_{2\cdot cv}$. By adopting the ontology $O_{i\cdot cv}$ to define mappings between the communication vocabulary and the native ontology of Ag-i, we avoid the introduction of special mapping operators as proposed by Stuckenschmidt and Timm (2002) and Borgida and Serafini (2003).

5.1.1 Knowledge and Dynamics

Knowledge distribution

Not every ontology is known by the agents. For example, O_2 is unknown to Ag-1 and O_1 is unknown to Ag-2 (the agents do not have access to each other's native ontologies). O_{cv} on the other hand, is known by both agents, whereas $O_{1\cdot 2}$ is neither known by Ag-1 nor Ag-2. We distinguish between local knowledge, common knowledge and implicit knowledge (Meyer and van der Hoek, 1995). Local knowledge refers to the knowledge of an individual agent which is not accessible to other agents. Something is common knowledge if it is known by every agent and every agent knows that every agent knows it, which is again known by every agent, etc. Something is implicit knowledge, if someone in the group knows it, or the knowledge is distributed over the members of the group. This includes the knowledge that would become derivable after the knowledge sources would be joined. By means of communication, the agents can only acquire knowledge that was already implicit in the group.

Assumption 5.1.

1. *O_i is local knowledge of Ag-i*
2. *O_{cv} is common knowledge of Ag-1 and Ag-2*
3. *$O_{i \cdot cv}$ is local knowledge of Ag-i*
4. *$O_{1 \cdot 2}$ is implicit knowledge of Ag-1 and Ag-2*

In the graphical representation, the different types of knowledge are indicated in the dashed boxes.

The assumption that O_{cv} is common knowledge makes this ontology appropriate for communication. The assumption that $O_{1 \cdot 2}$ is implicit knowledge opens up the possibility for automatic ontology mapping. This is a necessary condition for any system where the agents must learn to share meaning. Two agents cannot learn something from each other which was not already implicitly present beforehand.

Dynamics

Another aspect of ontologies is whether they are *static* or *dynamic* (Heflin and Hendler, 2000). Static ontologies do not change over time, whereas dynamic ontologies may change over time.

Assumption 5.2.

- *O_i and $O_{1 \cdot 2}$ are static ontologies.*
- *O_{cv} and $O_{i \cdot cv}$ are dynamic ontologies.*

In the graphical representation, dynamic ontologies are boxed by a rounded rectangle.

Changing an agent's native ontology cannot be straightforwardly established, as other components of the agents are dependent on it. For example the agent's assertional knowledge base and the agent's deliberation process (the process in which the agent reasons about what to do next) are defined using terms of the agent's native ontology. When the native ontology is changed, these other components must be adjusted as well. These issues are beyond the scope of this paper, and we therefore assume that the native ontologies O_1 and O_2 are static. As a consequence

$O_{1\cdot 2}$ is also a static ontology. O_{cv} on the other hand is a dynamic ontology. This causes no side effects as no other component of the agent is dependent on O_{cv}. In fact, it makes O_{cv} suitable as an alignment ontology as it enables agents to add concepts to it at runtime. As a consequence, the ontologies $O_{1\cdot cv}$ and $O_{2\cdot cv}$ are also dynamic.

Ontology exchange

We can now specify, from a conceptual viewpoint, how ontology exchange affects the ontologies in the system. As is apparent in Figure 5.2 and 5.3, when Ag-2 adds the concept *roadvehicle* to the communication vocabulary, the concept is added to C_{cv}, and becomes common knowledge. Definition 5.1 states that, as a consequence from this change of O_{cv}, *roadvehicle* also becomes part of $C_{1\cdot cv}$. Consequently, the relation $\leq_{1\cdot cv}$ is extended with the information that *motorhome*$\leq$*roadvehicle*$\leq$*vehicle*. As has been argued before, the static ontologies O_1, O_2 and $O_{1\cdot 2}$ remain unaffected.

5.1.2 Ontology Implementation

This section describes the data-structures and actions that can be used to implement the ontologies in the system. We show how description logic, described in Section 3.2.2, can be used to implement the ontologies of Definition 5.1 and how the requirements regarding their knowledge distribution (Assumption 5.1) can be met.

Implementing local and common knowledge

A description logic knowledge base is represented as a tuple $\langle \mathcal{T}, \mathcal{A} \rangle$, containing a TBox and an ABox. The TBox $\mathcal{T}$ is described by a set of terminological axioms which specify the inclusion relations between the concepts; it represents the agent's ontology. The ABox $\mathcal{A}$ contains a set of membership statements which specify which individuals belong to which concepts; it implements the agent's assertional knowledge. We use the description logic $\mathcal{ALC}$ (Section 3.2.2) without roles as a concept language that is used in the TBox and the ABox. From here, we will refer to this language as $\mathcal{L}$. We write C^a to refer to the atomic concepts in an ontology. We write C to refer to all concepts in an ontology, i.e. $\mathcal{L}(C^a)$.

We assume that the ABox is sound with respect to the intended interpretation, i.e. $\models_{\mathcal{I}^{INT}} \mathcal{A}$. Note that we do not assume that the ABox *completely* specifies the intended interpretation. This would make communication obsolete as the agents would already know everything. However, the assumption of a complete ABox is unrealistic as the domain of discourse will typically be of such size that it is unfeasible to enumerate all membership statements.

Local knowledge of Ag-i over O_i and $O_{i\cdot cv}$ (Assumption 5.1.1 and 5.1.3) can be straightforwardly established using two TBoxes: $\mathcal{T}_i$ and $\mathcal{T}_{i\cdot cv}$. Common knowledge over O_{cv} (Assumption 5.1.2) is established using the TBox $\mathcal{T}_{cv}$ of which both agents maintain a version. Because both agents have the same version of $\mathcal{T}_{cv}$ and they know that of each other, O_{cv} becomes common knowledge. We do not index $\mathcal{T}_{cv}$ with the agent name. The following property states that these TBoxes *fully* implement the agents' knowledge over the ontologies.

Property 5.1.

1. *For* $i \in \{1, 2\}$, *for all* $c, d \in C_i$: $\mathcal{T}_i \models c \sqsubseteq d$ *iff* $c \leq d$.
2. *For* $i \in \{1, 2\}$, *for all* $c, d \in C_{i \cdot cv}$: $\mathcal{T}_{i \cdot cv} \models c \sqsubseteq d$ *iff* $c \leq d$.
3. *For all* $c, d \in C_{cv}$: $\mathcal{T}_{cv} \models c \sqsubseteq d$ *iff* $c \leq d$.

The first item of the property should be established at design-time: the system developer should specify enough terminological axioms to completely specify the agent's native ontology. The second and third item of the property concern dynamic ontologies, and should be fulfilled by the ontology negotiation protocol.

Example 5.1. Consider the ontologies in Figure 5.3. We show how these ontologies can be implemented such that property 5.1 is fulfilled.
The following TBoxes are possessed by Ag-1

$\mathcal{T}_1$	$\mathcal{T}_{cv}$	$\mathcal{T}_{1 \cdot cv}$
motorhome $\sqsubseteq$ *vehicle* *hotel* $\sqsubseteq \neg$*vehicle*	*roadvehicle* $\sqsubseteq \top$	*roadvehicle* $\sqsubseteq$ *vehicle* *motorhome* $\sqsubseteq$ *roadvehicle* *hotel* $\sqsubseteq \neg$*vehicle*

The TBoxes that are possessed by Ag-2 are:

$\mathcal{T}_2$	$\mathcal{T}_{cv}$	$\mathcal{T}_{2 \cdot cv}$
RV-large $\sqsubseteq$ *roadvehicle* *RV-XL* $\sqsubseteq$ *roadvehicle* *RV-XL* $\sqsubseteq \neg$ *RV-large*	*roadvehicle* $\sqsubseteq \top$	*RV-large* $\sqsubseteq$ *roadvehicle* *RV-XL* $\sqsubseteq$ *roadvehicle* *RV-XL* $\sqsubseteq \neg$ *RV-large*

Implementing implicit knowledge

Until now, we have described how the first three items of Assumption 5.1 are implemented using common techniques available from description logic research. The fourth item of the assumption is not yet met, i.e. $O_{1 \cdot 2}$ is (not even) implicit knowledge. The data structures as described until now do not give rise to implicit knowledge of the relations between two different agents' local concepts. For example, $\mathcal{T}_{1 \cdot cv} \cup \mathcal{T}_{2 \cdot cv}$ does not specify the relations between Ag-1's concept *motorhome* and Ag-2's concept *RV-large*. This relation must be (at least) implicit knowledge, otherwise the agents are not capable of retrieving it. Therefore, we assume that the agents know more about their native ontologies than just the ordering between concepts, namely that they have access to the intended interpretation of their local concepts. This is done using the action `Classify`.

Action `Classify(c,a)`
Output specification:
if $a \in \mathcal{I}^{\text{INT}}(c)$ **then** add $c(a)$ to $\mathcal{A}$
else add $\neg c(a)$ to $\mathcal{A}$

For example, `Classify` can be thought of as a subsystem of a robot which recognizes and classifies objects in the real world (analogous to the approach to language creation followed by Steels (1998a)). In a scenario where the domain of discourse consists of text corpora, the action `Classify` can be implemented using text classification techniques (Jackson and Moulinier, 2002).

5.2 Requirements

This section serves to formalize the design objectives stated in Section 4.2.3. Given the framework introduced in the previous section, we will describe what qualifies as effective communication in Section 5.2.1 and 5.2.2. Then we will treat the objective of minimality by characterizing different types of ontologies in Section 5.2.3 and 5.2.4. We introduce some terminology first.

As argued in Chapter 4, the sender makes itself understandable to the receiver by translating the message stated in terms of its native ontology to a message stated in terms of the communication vocabulary. The receiver interprets this message by translating this message from the communication vocabulary to its own native ontology.

For example, consider the ontologies in Figure 5.3. Suppose that Ag-1 intends to convey the message that individual a is a *motorhome*. It translates message *motorhome(a)* (stated in terms of O_1) to *roadvehicle(a)* (stated in terms of O_{cv}). Ag-2 receives this message and translates it to its native ontology O_2, in this case also *roadvehicle(a)*. Generally, the following three concepts can be identified in the communication process.

Definition 5.3.

- *The transferendum ($c_i \in C_i$): what is to be conveyed. Ag-i (the speaker) intends to convey this concept to Ag-j.*
- *The transferens ($d_{cv} \in C_{cv}$): what conveyes. This concept functions as a vehicle to convey the transferendum to Ag-j.*
- *The translatum ($e_j \in C_j$): what has been conveyed. Ag-j (the hearer) interprets the received message as this concept.*

5.2.1 Subjective Equivalence

An ontology specifies how the agent views the world. In Section 2.3.4, we have argued that an ontology is like a strong pair of glasses that determines what the agent can see. Unavoidably, some distinctions are blurred by these glasses and are indiscernible for the agent. Consequently, two things that are different from an objective perspective may be equivalent from a subjective agent perspective. We will call such concepts *subjectively equivalent* (van Diggelen et al., 2004). Basically, two concepts are subjectively equivalent when the agent's ontology does not provide for sufficient means to express the difference. The notion can be formalized as follows:

Definition 5.4. *Two concepts c and d are subjectively equivalent for Ag-i iff*

- *the set of superconcepts of c in* C_i *is equal to the set of superconcepts of d in* C_i

This definition states that an agent regards two concepts as equivalent if they have the same superconcepts in its ontology, i.e. when the two concepts carry the same implications for its ontology.

Example 5.2. Consider the ontologies in Figure 5.2. We observe the following:

- *RV-XL* and *RV-large* are subjectively equivalent for Ag-1.
- *vehicle* and ¬ *hotel* are subjectively equivalent for Ag-2.
- *motorhome* and *roadvehicle* are subjectively equivalent for Ag-2.

5.2.2 Sound and Lossless Communication

In this section, we will state the requirements for normal communication. The first requirement concerns the *quality* of information exchange, i.e. soundness. Soundness means that the interpretation of the message by the hearer (the translatum) must follow from what the speaker intended to convey in the message (the transferendum). In ontological reasoning, when a is member of a concept c, it *follows* that a is also member of a superconcept of c. This is stated in the following definition:

Definition 5.5. ***Sound communication***
Let c_i *be the transferendum, and* e_j *be the translatum. Communication is sound iff* e_j *is a superconcept of* c_i *in* C_j*.*

An example of sound communication from Ag-1 to Ag-2 is: Ag-1 translates transferendum *motorhome* to transferens *roadvehicle* which Ag-2 "translates" to translatum *roadvehicle*. An example of non-sound communication from Ag-1 to Ag-2 is transferendum: *vehicle*, transferens: *roadvehicle*, translatum: *roadvehicle*.

It is not difficult to satisfy only the soundness requirement of communication. In the extreme case, the translatum is the top concept to which all individuals in Δ belong. This is guaranteed to be sound as this concept is a superconcept of all other concepts. However, an assertion stating that an individual belongs to the top concept, does not contain any information about the individual; it is a trivial fact. To prevent overgeneralization, a second requirement is needed which takes the *quantity* of information exchange into account.

The lossless requirement states that the translatum should not only be a superconcept of the transferendum, but that it should also be the most specific one.

From the perspective of the receiver, no information is lost in the process of translating to and translating from the communication vocabulary. From an objective viewpoint, however, information may get lost. Because this information-loss is not representable in the receiver's ontology, this loss is not present from a subjective viewpoint. For this reason, this requirement is properly called subjectively lossless communication. From now on, we shall simply refer to it as lossless communication. The definition of lossless communication is stated as follows:

Definition 5.6. ***Lossless communication***
Let c_i be the transferendum and e_j the translatum. Communication is lossless iff e_j is most specific in the set of superconcepts of c_i in C_j

Another way to state the condition for lossless communication is that the transferendum c_i is subjectively equivalent with the translatum e_j for the hearer. It can be easily shown that these two definitions are equivalent. The condition in Definition 5.6 states that e_j is most specific in the set of superconcepts of c_i. This means that that all superconcepts of c_i in C_j are either e_j or a superconcept of e_j. Hence, c_i and e_j are subjectively equivalent for Ag-j, according to Definition 5.4.

Note that in Definition 5.5 and 5.6 no mention is made of the transferens. This is because the concepts in the communication vocabulary only serve as vehicles to convey the speaker's information to the hearer. To enable sound and lossless communication, there must be sufficient vehicles available. Note that this definition defines lossless communication from the god's eye view. In Section 5.3.2, we will describe how the agents can assess lossless communication using their local knowledge.

Example 5.3. The empty communication vocabulary in the initial situation (Figure 5.2), does not enable the agents to losslessly communicate any local concept (except the top concept).

The communication vocabulary in Figure 5.3 sufficiently aligns O_1 and O_2 for Ag-1 to losslessly communicate *motorhome* to Ag-2, viz. Ag-1 translates transferendum *motorhome* to transferens *roadvehicle* which Ag-2 "translates" to translatum *roadvehicle*. This is lossless communication because the translatum *roadvehicle* is the most specific in the set of superconcepts of the transferendum *motorhome* in C_2.

O_{cv} does not sufficiently align O_1 and O_2 for Ag-1 to losslessly communicate *RV-XL* to Ag-2. Suppose Ag-2 translates *RV-XL* to *roadvehicle* which Ag-1 translates to *vehicle*. This communication process has not been lossless because *motorhome* would have been a more specific translation of *RV-XL* to Ag-1's ontology.

5.2.3 Optimal Communication Vocabularies

A communication vocabulary that enables a pair of agents to losslessly communicate every transferendum, is called a lossless communication vocabulary. It is formally defined as follows:

Definition 5.7. ***Lossless communication vocabulary***
O_{cv} is a lossless communication vocabulary for Ag-i and Ag-j iff

- *for all* $c_i \in C_i$*, there exists* $d_{cv} \in C_{cv}$*, such that* d_{cv} *is subjectively equivalent with* c_i *for Ag-j.*
- *for all* $c_j \in C_j$*, there exists* $d_{cv} \in C_{cv}$*, such that* d_{cv} *is subjectively equivalent with* c_j *for Ag-i.*

The two conditions state that for every transferendum of Ag-i and Ag-j, there is a subjectively equivalent transferens available in the communication vocabulary for the hearer. This is another way to state that lossless communication is always possible. Suppose that the hearer Ag-j translates the transferens d_{cv} to translatum e_j which is the most specific superconcept of d_{cv} in C_j. If the transferendum c_i is subjectively equivalent with d_{cv}, then e_j is also the most specific superconcept of c_i, meeting the lossless requirement.

Example 5.4. Consider the ontologies from Figure 5.2, the following are lossless communication vocabularies:

- O_{cv-1} where $C^a_{cv-1} = C_{1\cdot 2}$
- O_{cv-2} where $C^a_{cv-2} = \{RV\text{-}XL, motorhome, roadvehicle\}$
- O_{cv-3} where $C^a_{cv-3} = \{motorhome, roadvehicle\}$

An optimal communication vocabulary is not only lossless, but is also minimal in size. The following definition formalizes this.

Definition 5.8. ***Optimal communication vocabulary***
O_{cv} *is an optimal communication vocabulary for Ag-i and Ag-j iff*

- O_{cv} *is a lossless communication vocabulary for Ag-i and Ag-j*
- *no* $C'_{cv} \subseteq C_{i\cdot j}$ *exists, such that*
  - $\langle C'_{cv}, \leq'_{cv}\rangle$ *is a lossless communication vocabulary*
  - $\#C'_{cv} < \#C_{cv}$

An example of an optimal communication vocabulary is O_{cv-3} in Example 5.4. The other two communication vocabularies in this example are not optimal, because they are not minimal in size.

5.2.4 Communication Vocabularies for Multiple Agents

Until now, we have discussed communication vocabularies between two agents. The definitions 5.7 and 5.8 can be easily adapted to apply to a situation with multiple agents. A lossless communication vocabulary for MAS = {Ag-1..Ag-n} is a communication vocabulary that is lossless for every two agents in MAS. An optimal common communication vocabulary for multiple agents is the smallest communication vocabulary that is lossless for those agents.

The extensions proposed above assume that every pair of agents adopts the same communication vocabulary. In systems with more than two agents, it may be beneficial to adopt multiple communication vocabularies between different pairs of agents. As will be shown, this may lead to a smaller amount of shared concepts. We will call this a communication vocabulary distribution. In a multi-agent system MAS = {Ag-1..Ag-n}, the vocabulary distribution is characterized as a function that maps every non-directed pair of agents to a communication vocabulary:, i.e. $\mathcal{CV} : \{\{\text{Ag-i}, \text{Ag-j}\} | \text{Ag-i}, \text{Ag-j} \in MAS\} \rightarrow \{O_{cv-1}..O_{cv-m}\}$

A cv distribution is lossless if it provides every pair of agents with a lossless communication vocabulary.

Definition 5.9. ***Lossless communication vocabulary distribution***
Given a multi-agent system MAS = {Ag-1 .. Ag-n}, a cv distribution $\mathcal{CV}$ is lossless iff

- *for every $\{Ag\text{-}i, Ag\text{-}j\} \subseteq MAS$, it holds that $\mathcal{CV}(\{Ag\text{-}i, Ag\text{-}j\})$ is a lossless communication vocabulary for Ag-i and Ag-j*

Note that one agent may use different communication vocabularies with different agents. This may require the agent to learn multiple communication vocabularies. The *costs* that an agent must invest to learn a cv distribution are given by the total number of different concepts that it uses in a cv with any other agent. This is formalized as follows:

Definition 5.10. ***Costs***
Given a multi-agent system MAS = {Ag-1 .. Ag-n} and a cv distribution $\mathcal{CV}$, the costs for Ag-i to adopt $\mathcal{CV}$ equals:
$costs_{\mathcal{CV}}(Ag\text{-}i) = \sum_{Ag\text{-}j \in MAS \setminus \{Ag\text{-}i\}} size(\mathcal{CV}(\{Ag\text{-}i, Ag\text{-}j\}))$
where $size(\langle C, \leq \rangle) = \#C$

An optimal cv distribution is lossless, and minimizes the total costs for the agents to adopt it:

Definition 5.11. ***Optimal communication vocabulary distribution***
Given a multi-agent system MAS = {Ag-1 .. Ag-n}, a cv distribution $\mathcal{CV}$ is optimal iff the following holds:

- *$\mathcal{CV}$ is a lossless cv distribution.*
- *no lossless cv distribution $\mathcal{CV}'$ exists, s.t.* $\sum_{i=1}^{n} costs_{\mathcal{CV}'}(Ag\text{-}i) < \sum_{i=1}^{n} costs_{\mathcal{CV}}(Ag\text{-}i)$.

An example that illustrates the benefits of a communication vocabulary distribution is given below.

Example 5.5. Consider the four ontologies in Figure 5.4. The ontologies O_1 and O_3 exhibit expertise on the travel domain. The ontologies O_2 and O_4 exhibit expertise on the telephone domain. An optimal communication vocabulary distribution for these agents is as follows:

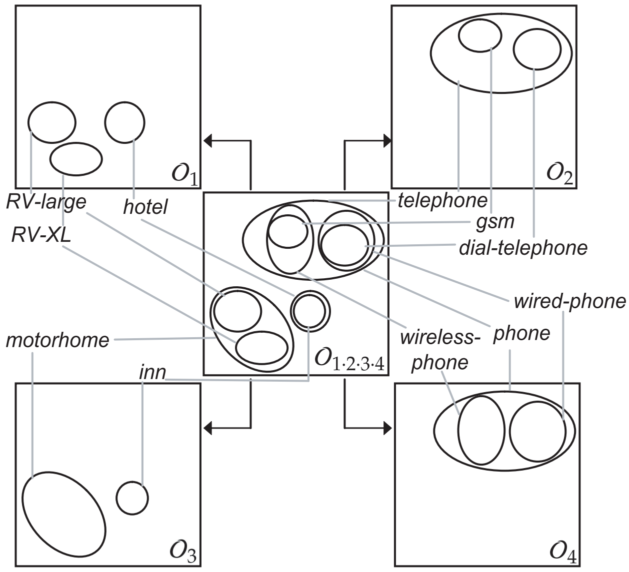


Figure 5.4: A suitable case for a communication vocabulary distribution

- $\mathcal{CV}(\{\text{Ag-1}, \text{Ag-2}\}) = \mathcal{CV}(\{\text{Ag-1}, \text{Ag-4}\}) = \mathcal{CV}(\{\text{Ag-3}, \text{Ag-2}\})$ $= \mathcal{CV}(\{\text{Ag-3}, \text{Ag-4}\}) = \mathcal{O}_{cv-1}$, where $C^a_{cv-1} = \{phone\}$
- $\mathcal{CV}(\{\text{Ag-1}, \text{Ag-3}\}) = \mathcal{O}_{cv-2}$, where $C^a_{cv-2} = \{motorhome, inn\}$
- $\mathcal{CV}(\{\text{Ag-2}, \text{Ag-4}\}) = \mathcal{O}_{cv-3}$, where $C^a_{cv-3} = \{phone, wireless\text{-}phone, wired\text{-}phone\}$

This cv distribution requires the agents to learn the following concepts:

- Ag-1: *phone*, *motorhome*, *inn*
- Ag-2: *phone*, *wireless-phone*, *wired-phone*
- Ag-3: *phone*, *motorhome*, *inn*
- Ag-4: *phone*, *wireless-phone*, *wired-phone*

Note that adopting one common optimal cv for the four agents would require every agent to learn the concepts *phone*, *motorhome*, *inn*, *wireless-phone*, *wired-phone*. Hence, the benefit of adopting a cv distribution instead of one common cv, is that each agent is only required to learn three concepts instead of five.

The example above indicates that a cv distribution may be more economical with respect to the amount of shared concepts. This benefit occurs when several groups of agents with different areas of expertise are present in the MAS. Concepts that are relevant for everyone are shared between all agents. This enables the agents to communicate on a general level. Specialized concepts that are only considered relevant by few agents, are only shared by few agents. This *jargon* enables these agents to communicate about their areas of expertise. In MAS's consisting of many heterogeneous agents, the economy of a cv distribution may be considerably large. Another advantage of a cv distribution is that it allows local adjustments of the cv when new agents join the MAS, whereas a common cv would require every agent in the MAS to adjust its cv.

5.3 Protocols

Before we propose protocols for ontology negotiation in Section 5.3.2, we will discuss how ontology exchange can be implemented in our framework. This corresponds to the CDP and CEP layer discussed in Section 4.3.2 and 4.3.3. After that, we propose some communication protocols that implement normal communication, ontology alignment and a transition between them. We evaluate these protocols using the criteria of minimal cv construction, laziness, and soundness and losslessness.

The communicative abilities of the agents are specified as actions. During the execution of actions, messages are sent through the instruction send(Ag-j, $\langle$performative, $p_1, .., p_n\rangle$), where Ag-j is the addressee of the message and $p_1..p_n$ are parameters of the message. The effect of this instruction is that Ag-j is able to perform a `Receive`(Ag-i, $\langle$performative, $x_1, .., x_n\rangle$) action, where Ag-i is the sender of the message and $x_1..x_n$ are instantiated to $p_1..p_n$. For clarity reasons, we will omit `Receive` actions from the protocols. In the specification of actions and protocols we will adopt Ag-i as the sender and Ag-j as the receiver of messages.

5.3.1 Implementing Automatic Ontology Mapping

For our automatic ontology mapping technique to work, we require that the agents have access to the same elements in the domain of discourse ($\Delta^{\mathcal{I}}$), and use the same signs to refer to these individuals (given by the set Δ). These requirements are readily met in our case study (described in Chapter 7), where every agent has access to IP-addresses (i.e. $\Delta^{\mathcal{I}}$ is the set of IP-addresses), and every agent uses URL's to refer to these addresses (i.e. Δ is the set of URL's). The ontology of news-topics that are used to classify news-articles differs from agent to agent. This is where ontology negotiation fulfils its task.

Ontology exchange is implemented using the action `AddConcept` which enables an agent to add a concept to the communication vocabulary. The effects of adding a concept were described in Section 5.1.1. To realize these effects, Ag-j's TBoxes $\mathcal{T}_{j\cdot cv}$ and $\mathcal{T}_{cv}$ must be updated such that property 5.1.2 and 5.1.3 hold. To realize the effects regarding $\mathcal{T}_{cv}$, Ag-i must communicate to Ag-j the relations of the newly added concept c with the other concepts in the communication vocabulary. It does this in the `SendBoundaries` action:

Action `SendBoundaries`(Ag-j,c)
Let mss be most specific in the set of superconcepts of c in C_{cv} and mgs be most general in the set of subconcepts of c in C_{cv}

- add $c \sqsubseteq mss$ and $mgs \sqsubseteq c$ to $\mathcal{T}_{cv}$
- send (Ag-j,⟨inform,boundaries,c, mss, mgs⟩)

Action `Receive`(⟨inform, boundaries,c, mss, mgs⟩)

- add $c \sqsubseteq mss$ and $mgs \sqsubseteq c$ to $\mathcal{T}_{j \cdot cv}$

The `SendBoundaries` action corresponds to the concept definition protocol (CDP), described in Section 4.3.2.

Realizing the effects of automatic ontology mapping on $O_{j \cdot cv}$ is more difficult to establish because neither Ag-i nor Ag-j has explicitly represented these relations in a TBox. For example, consider the ontologies in Figure 5.2 and 5.3. In the initial situation, neither of the agents has local knowledge that *motorhome≤roadvehicle≤vehicle*. Hence this information must be conveyed differently. Ag-i conveys this information to Ag-j by sending a set of positive and negative examples of concept c. Upon receiving these examples, Ag-j uses inductive inference to derive the relations of c with the concepts in its native ontology. This is done by the `Explicate` action. Remember that the agents have access to the intended interpretation of concepts using the `Classify` action described earlier.

Action `Explicate`(Ag-j, c)

- send (Ag-j,⟨inform, explication,c,$\{p | \mathcal{I}(p) \in \mathcal{I}^{INT}(c)\}$, $\{n | \mathcal{I}(n) \notin \mathcal{I}^{INT}(c)\}$⟩)

Action `Receive`(⟨inform, explication,c,P,N ⟩)

- add $c \sqsubseteq d_j$ to $\mathcal{T}_{j \cdot cv}$, *where d_j is most specific in the set* $\{d'_j | \forall p \in P. \mathcal{I}(p) \in \mathcal{I}^{INT}(d'_j)\}$
- add $d_j \sqsubseteq c$ to $\mathcal{T}_{j \cdot cv}$, *where d_j is most general in the set* $\{d'_j | \forall n \in N. \mathcal{I}(n) \notin \mathcal{I}^{INT}(d'_j)\}$

We assume that the number of examples in the sets P and N are sufficiently large, to enable Ag-j to derive every relation of c with the concepts in C_j. The `Explicate` action corresponds to the concept explication protocol (CEP) under ideal circumstances, described in Section 4.3.3.

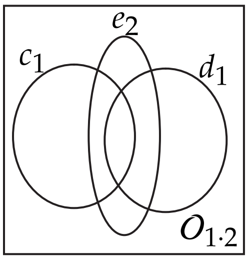
The `SendBoundaries` and `Explicate` actions are sufficient to convey the meaning of a concept to another agent. An agent that adds an atomic concept c to C^a_{cv}, may introduce more than one concept in C_{cv}, namely the concept c, and the concepts that can be composed using that concept and other concepts in the cv. The set of new concepts that are introduced in C_{cv} after an atomic concept c is added is given by $\mathcal{L}(C^a_{cv} \cup \{c\}) \backslash \mathcal{L}(C^a_{cv} \backslash \{c\})$. The sending agent Ag-i conveys the meanings of these concepts that are also in C_i. We can now define the action `AddConcept` as follows:

Action `AddConcept`(Ag-j, c)

- For all $d \in (\mathcal{L}(C^a_{cv} \cup \{c\}) \backslash \mathcal{L}(C^a_{cv})) \cap C_i$:
    - `SendBoundaries`(Ag-j,d)
    - `Explicate`(Ag-j, d)

The AddConcept action implements a *non-compositional* concept explication method. This means that, when an agent explicates a concept c, it also explicates all composed concepts that can be formed with c and the concepts in the existing cv. This is stated in the first line of the `AddConcept` action. The motivation for this is illustrated by the following example.

Example 5.6. Consider the god's eye view of the ontologies of Ag-1 and Ag-2 sketched below.



Suppose that Ag-1 adds c_1 to the communication vocabulary and subsequently explicates d_1 to Ag-2. $\mathcal{T}_{2\cdot cv}$ becomes $\{\bot \sqsubseteq c \sqsubseteq \top, \bot \sqsubseteq d \sqsubseteq \top\}$, lacking the information that $c \sqcap d \sqsubseteq e$. To resolve this issue, Ag-2 must also explicate $c \sqcap d$.

The above example shows why non-compositional concept explication is needed. However, the solution may not be very efficient, as it may require the speaker to explicate many composed concepts. Some efficiency measures can be incorporated in `AddConcept` rather straightforwardly. Many composed concepts definable in the ontology language denote $\bot$. There is no need to explicate such concepts. Furthermore, the phenomenon of Example 5.6 only occurs when overlapping concepts are involved. The speaker could restrict itself to the explication of conjunctions and disjunctions of overlapping concepts. This would reduce the number of concept explications.

5.3.2 Ontology Negotiation Protocols

In this section, we will propose three ontology negotiation protocols. The protocols differ in the way they implement normal communication, how they recognize when normal communication cannot proceed, and the communication vocabularies they give rise to. We will evaluate these protocols according to the criteria of soundness and losslessness, laziness and minimal cv construction.

In Section 5.2 we defined successful communication as being sound and lossless. Furthermore, we characterized different types of minimal communication vocabularies. Whereas these properties are defined using a God's eye view over the agents' ontologies, the agents can only use their local knowledge to assess these properties. This plays a central role in our discussion.

ONP-1

We begin with a formalization of NCP-2 (Figure 4.6). Normal communication is implemented by translating the transferendum (in the sender's native ontology) to an equivalent transferens (in the communication vocabulary). The receiver translates the transferens to the most specific superconcept in its native ontology, the translatum. This is done by the `ExactInform` action. If there is no transferens available in the communication vocabulary that is equivalent to the transferendum, the speaker decides that normal communication cannot proceed, and adds the transferendum to the communication vocabulary.

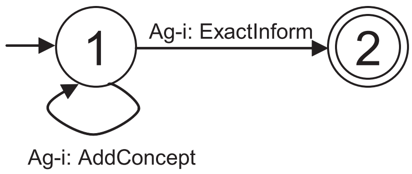


Figure 5.5: ONP-1

Action `ExactInform`(Ag-j, $c_i(a)$)
if $\exists d_{cv}.d_{cv} \equiv c_i$ **then** send(Ag-j, $\langle$ExactInform, $d_{cv}(a)\rangle$)

Action `Receive`(Ag-i,$\langle$ ExactInform,$d_{cv}(a)\rangle$)
Add $e_j(a)$ to $\mathcal{A}_j$, *where* e_j *is most specific in the set of superconcepts of* d_{cv} *in* C_j

When the condition in the if statement of `ExactInform` is not met, the agent must perform an `AddConcept` action. It is not difficult to prove that in protocol 1, communication proceeds in a lossless fashion as defined in Definition 5.6. The event that is triggered upon receiving an ExactInform message, produces a translatum e_j which is most specific in the set of superconcepts of d_{cv} in C_j. Because the action that produces an ExactInform message requires the transferendum c_i to be equivalent to d_{cv}, it follows that e_j is also most specific in the set of superconcepts of c_i in C_j, meeting the lossless requirement.

Example 5.7. Consider the initial situation in Figure 5.2, where the agents have not yet taught concepts to each other. Suppose Ag-2 intends to convey the assertion *roadvehicle(a)* to Ag-1. Below, the actions are described which are performed by the agents. We describe some of the instructions that are executed *within* an action;

these are preceded with $\llcorner$.

Ag-2 : `AddConcept`(Ag-1, *roadvehicle*)
$\llcorner$Ag-2 : add $roadvehicle \sqsubseteq \top$ to $\mathcal{T}_{cv}$
$\llcorner$Ag-1 : add $motorhome \sqsubseteq roadvehicle \sqsubseteq vehicle$ to $\mathcal{T}_{1\cdot cv}$
Ag-2 : `ExactInform`(Ag-1,*roadvehicle*(a))
$\llcorner$Ag-2 : send(Ag-1, ⟨ExactInform, *roadvehicle(a)*⟩)
Ag-1 : `receive`(Ag-2, ⟨ExactInform, *roadvehicle(a)*⟩)
$\llcorner$Ag-1 :add *vehicle(a)* to $\mathcal{A}_1$

This conversation has given rise to a communication vocabulary as in Figure 5.3.

Example 5.8. Suppose that Ag-1 intends to convey the message *hotel*(a) and that the cv contains the concept *roadvehicle* (as in Figure 5.3). The agents perform the following actions:

Ag-1 : `AddConcept`(Ag-2,*hotel*)
$\llcorner$Ag-1 : add $hotel \sqsubseteq \neg roadvehicle$ to $\mathcal{T}_{cv}$
$\llcorner$Ag-2 : add $hotel \sqsubseteq \neg roadvehicle$ to $\mathcal{T}_{2\cdot cv}$
Ag-1 : `ExactInform`(Ag-2,*hotel(a)*)
$\llcorner$Ag-1 : send(Ag-2,⟨ExactInform, *hotel(a)*⟩)
Ag-2 : `receive`(Ag-1, ⟨ExactInform, *hotel(a)*⟩)
$\llcorner$Ag-2 :add $\neg roadvehicle(a)$ to $\mathcal{A}_2$

After this conversation has finished, the communication vocabulary contains the concepts *roadvehicle* and *hotel*.

Although ONP-1 ensures sound and lossless communication, it is not lazy and does not give rise to a minimal cv. In the second dialogue of the example, it was not necessary to add the concept *hotel* to the cv, as lossless communication was already enabled by the concept ¬*roadvehicle*. If Ag-1 would have translated *hotel* to the superconcept ¬*roadvehicle*, then Ag-2 could have interpreted this as ¬*roadvehicle*, and this would have been sound and lossless communication. However, this dialogue is not allowed by ONP-1. Using ONP-1, the sender sometimes adds concepts to the cv that do not contribute to successful communication. In fact, after the agents have exchanged a number of messages, the communication vocabulary will simply consist of every transferendum that was conveyed by one of those messages. Therefore, this protocol is not satisfactory with respect to minimal cv construction and laziness. The following protocol attempts to overcome these problems.

ONP-2

ONP-2 corresponds to NCP-5 (Figure 4.9). The sender uses the `ExactInform` action when allowed. When this is not allowed, i.e. when the sender is not able to express itself *exactly* in shared concepts, it does not immediately add the concept to the communication vocabulary. Instead, it conveys the message *as specifically as possible* using a superconcept of the transferendum. This is done using an `Inform` action. It is upon the receiver to decide whether the transferens in an Inform-message is specific enough to meet the lossless criterion.

Action `Inform`(Ag-j, $c_i(a)$)
send(Ag-j, ⟨Inform, $d_{cv}(a)$⟩) *where* d_{cv} *is most specific in the set of superconcepts of* c_i *in* C_{cv}

The `Receive` action that is triggered by an Inform message is equal to the `Receive` action that is triggered when an ExactInform message is received. We will now turn our attention to the issue of how the receiver can recognize when communication has been lossless and when not.

Because the receiver does not know the transferendum, it cannot directly check Definition 5.6 for lossless communication. However, the receiver knows that the sender has obeyed the rules of the `Inform` action, and therefore that the transferens is most specific in the set of superconcepts of the transferendum. This enables the receiver, in some cases, to check the lossless condition nonetheless. In philosophy of language, such a derivation is known as a *conversational implicature* (Section 2.2.1). In ONP-2, it works as follows: consider the ontologies O_1 and O_2 from Figure 5.2, and suppose that $C^a_{cv} = \{vehicle, motorhome\}$. Suppose that the transferendum is Ag-2's concept *roadvehicle* and that Ag-2 uses *vehicle* as a transferens. Upon receiving this message, Ag-1 knows that Ag-2 did *not* intend to convey the following subconcepts in C_{cv}: *motorhome* and *vehicle* $\sqcap \neg$*motorhome*. This is because otherwise Ag-2 should have used these more specific concepts in the message. Knowing that the transferendum is more general than these concepts, Ag-1 knows that communication has been lossless.

In ONP-2, the receiver Ag-j responds `OK` when it believes that communication has been lossless. The condition of `OK` first identifies a set D that contains all concepts which are most general in C_{cv} among the set of strict subconcepts of the transferens. It knows that the sender did *not* intend to convey any information that is as specific as or more specific than any concept in D (otherwise it would have been obliged to use one of these more specific concepts). Then, it checks whether any concepts exist in C_j that are more specific than the translatum but *not* more specific than any concept in D. If there are none such concepts, it regards communication as lossless and the conversation terminates. Otherwise, it responds with `ReqSpec` (Request Specification) to start the ontology alignment protocol where Ag-i adds the transferendum to cv. The OK action can only be done if the receiver can assess that communication was lossless.

Action `OK`(Ag-i)
Responding to ⟨Inform, $(d_{cv}(a))$⟩
Let D *be the set of concepts that are most general among the set of strict subconcepts of* d_{cv} *in* C_{cv}
Let e_j *be most specific in the set of superconcepts of* d_{cv} *in* C_j *(*e_j *is the translatum)*
if *every strict subconcept of* e_j *in* C_j *is a subconcept of any of the concepts in* D
then send(Ag-i, ⟨OK⟩)

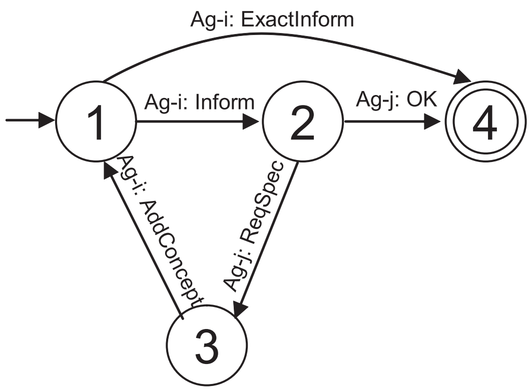


Figure 5.6: ONP-2

Example 5.9. Consider the ontologies in Figure 5.3 and suppose that Ag-1 wishes to communicate *hotel(a)* (as in Example 5.8). The dialogue proceeds as follows:

Ag-1 : `Inform`(Ag-2, *hotel*(*a*))
∟Ag-1 : send(Ag-1, ⟨Inform, ¬*roadvehicle*(*a*)⟩)
Ag-2 : `Receive` (Ag-1, ⟨Inform, ¬*roadvehicle*(*a*)⟩)
∟Ag-2 : add ¬*roadvehicle*(a) to $\mathcal{A}_2$
Ag-2 : `OK`

In this example, Ag-2 responded with OK, because in O_2 the information provided by ¬*roadvehicle* is as specific as possible.
Now, consider the ontologies in Figure 5.3 with a different cv. Suppose that Ag-2 wishes to communicate *RV-large*(*a*), and that C^a_{cv} = {*motorhome*}. The dialogue proceeds as follows:

Ag-2 : `Inform`(Ag-1, *RV-large*(*a*))
∟Ag-2 : send(Ag-1, ⟨Inform, *motorhome*(*a*)⟩)
Ag-1 : `Receive` (Ag-2, ⟨Inform, *motorhome*(*a*)⟩)
∟Ag-1 : add *motorhome*(a) to $\mathcal{A}_1$
Ag-1 : `OK`

In this example, Ag-1 responded with OK, because in O_1 the information provided by *motorhome* is as specific as possible.

Now, suppose that Ag-2 wishes to communicate *RV-large*(*a*), C^a_{cv} = {vehicle}.

Ag-2 : `Inform`(Ag-1, *RV-large(a)*)
∟Ag-2 : send(Ag-1,⟨Inform, *vehicle*(*a*)⟩)
Ag-1 : `Reqspec`
Ag-2 : `AddConcept`(Ag-1,*RV-large*)
Ag-2 : `ExactInform`(Ag-1,*RV-large*(*a*))
Ag-1 : `Receive` (Ag-1, ⟨ExactInform, *RV-large*(*a*)⟩)

$\llcorner$Ag-1 : add *motorhome*(a) to $\mathcal{A}_1$

In this example Ag-1 did not respond OK at first, because *motorhome* caused the condition to fail. Thus, Ag-1 correctly recognized non-lossless communication.

Now, suppose that Ag-2 wishes to communicate *roadvehicle*(a), and $C^a_{cv} = \{vehicle, motorhome\}$

Ag-2 : `Inform`(Ag-1, *roadvehicle*(a))
$\llcorner$Ag-2 : send(Ag-1,⟨Inform, *vehicle*(a)⟩)
Ag-1 : `Receive` (Ag-2, ⟨Inform, *vehicle*(a)⟩)
$\llcorner$Ag-1 : add *vehicle*(a) to $\mathcal{A}_1$
Ag-1 : OK

In this example, Ag-1 responded OK, because it knew that if Ag-2 had more information available about individual a, e.g. membership of *motorhome*, it would have used a more specific term, e.g. *motorhome*(a). Thus, Ag-1 correctly recognized lossless communication.

Theorem 5.1. *If the receiver responds* `OK` *then communication has been lossless.*

Proof: Suppose c_i is the transferendum, d_{cv} the transferens and e_j the translatum. We prove the theorem by showing that the situation where the receiver responds `OK` while communication was *not* lossless leads to a contradiction. Non-lossless communication means that e_j is *not* a most specific concept in the set $\{e'_j | c_i \leq_{1\cdot 2} e'_j \wedge e'_j \in C_j\}$ (Definition 5.6 does not hold). This means that either e_j is not in the set $\{e'_j | c_i \leq_{1\cdot 2} e'_j \wedge e'_j \in C_j\}$ (option A), or that e_j is not a *most specific* element in that set (option B). We will show that both options lead to a contradiction. The conditions for sending and receiving an inform speech act ensure that $c_i \leq d_{cv} \leq e_j$, and therefore $c_i \leq e_j$; this contradicts with option A. If e_j is not *most specific* in the set $\{e'_j | c_i \leq_{1\cdot 2} e'_j \wedge e'_j \in C_j\}$, it means that some concept e''_j exists in this set for which $c_i \leq e''_j < e_j$. According to the condition in the if-statement of `OK`, it holds that some concept $d'_{cv} \in D$ exists for which $e''_j \leq d'_{cv} < d_{cv}$. Because $c_i \leq d'_{cv}$ and $d'_{cv} < d_{cv}$, it follows that d_{cv} is *not* most specific in the set $\{d''_{cv} | c_i \leq_{1\cdot 2} d''_{cv} \wedge d''_{cv} \in C_{cv}\}$. Therefore, option B is in contradiction with the condition of `Inform`.

□

The observant reader may have noted that this mechanism for recognizing overgeneralization differs slightly from the one proposed in the previous chapter. In particular, it checks for *subconcepts* of the translatum, instead of *particularizations* (sub- or overlapping concepts) of the transferens. Nevertheless, the two conditions amount to the same thing. The following property expresses that when the condition of `OK` in this chapter holds, the condition of NCP-5 stated in Figure 4.9 also holds.

Property 5.2. *Let D, e_j and d_{cv} be as defined in the action* `OK`*.*
If e'_j be a strict subconcept of e_j in C_j which is not a subconcept of any concept in D (e'_j breaks the condition of `OK`*)*
then e'_j is a particularization of d_{cv}.

Proof: To prove that e'_j is a particularization of d_{cv}, we will show that e'_j is not a superconcept, equivalent concept, or disjoint concept of d_{cv}. If $d_{cv} \sqsubseteq e'_j$, then e'_j should have been the translatum, because e'_j is a strict subconcept of e_j. Furthermore, if e'_j would be disjoint with d_{cv}, then $e_j \sqcap \neg e'_j$ should have been the translatum, because $e_j \sqcap \neg e'_j$ is a strict subconcept of e_j and a superconcept of d_{cv}. Hence, e'_j is a subconcept or overlapping concept (i.e. particularization) of d_{cv}.

□

The property stated above reveals that the analysis of ONP-2 regarding lossless communication also applies to NCP-5 presented in the previous chapter.

Because ONP-2 enables the agents to communicate without learning every concept in their native ontologies from each other, this protocol scores better than ONP-1, with respect to laziness (Figure 5.8). However, the protocol may still give rise to a communication vocabulary which is unnecessarily large, as shown by the following example.

Example 5.10. Consider the initial situation in Figure 5.2. Suppose that Ag-1 intends to convey *motorhome*.

Ag-1 : `Inform`(Ag-2, *motorhome(a)*)
∟Ag-1 : send(Ag-2,⟨Inform, $\top(a)$⟩)
Ag-2 : `Reqspec`
Ag-1 : `AddConcept`(Ag-2,*motorhome*)
Ag-1 : `ExactInform`(Ag-2,*motorhome*(*a*))
Ag-2 : `Receive` (Ag-1, ⟨ExactInform, *motorhome*(*a*)⟩)
∟Ag-2 : add *roadvehicle(a)* to $\mathcal{A}_2$

After this dialogue, the cv is {*motorhome*}. In the next dialogue, Ag-2 intends to convey *roadvehicle*. A similar dialogue follows; after this dialogue, the cv has become {*motorhome*, *roadvehicle*}. In the next dialogue, Ag-1 intends to convey *vehicle*. After this dialogue has finished, the cv has become {*motorhome*, *roadvehicle*, *vehicle*}.

The communication vocabulary resulting from Example 5.10 is unnecessarily large, because {*motorhome, vehicle*} enables the agents to losslessly communicate the same concepts as {*motorhome, roadvehicle, vehicle*}. For this reason, ONP-2 is not satisfactory with respect to minimal cv construction. The next protocol aims to overcome this problem by allowing the agents to remove superfluous concepts from their communication vocabulary.

ONP-3

This protocol concerns a communication mechanism that has not been treated in the previous section. It aims at achieving a minimal communication vocabulary by occasionally removing concepts from it. Concepts can be removed from the vocabulary if they are *mutually redundant*, i.e. redundant for both agents. Mutually redundant concepts have the property that their removal does not affect what the agents can losslessly communicate to each other during normal communication. This is stated in the following definition.

Definition 5.12. *$d \in C^a_{cv}$ is mutually redundant if $\mathcal{L}(C^a_{cv} \backslash \{d\})$ allows Ag-i and Ag-j to losslessly communicate the same concepts to each other as $\mathcal{L}(C^a_{cv})$.*

The above definition does not state how agents can recognize redundant concepts. An agent may consider a concept redundant if it determines that another concept in the cv could serve as a substitute for sending messages and that another concept in the cv could serve as a substitute for receiving messages. This is expressed in the following definition.

Definition 5.13. *Ag-i considers a concept d_{cv} redundant iff both of the following holds:*

- *d'_{cv} is a superconcept of c_i, where*
  - *d'_{cv} is most general in the set of subconcepts of d_{cv} in the set $\mathcal{L}(C^a_{cv} \backslash \{d_{cv}\})$*
  - *c_i is most general in the set of subconcepts of d_{cv} in C_i.*
- *d''_{cv} is a subconcept of c'_i, where*
  - *d''_{cv} is most specific in the set of superconcepts of d_{cv} in the set $\mathcal{L}(C^a_{cv} \backslash \{d_{cv}\})$*
  - *c'_i is most specific in the set of superconcepts of d_{cv} in C_i.*

In this definition, the formula $\mathcal{L}(C^a_{cv} \backslash \{d_{cv}\})$ denotes the communication vocabulary that remains after d_{cv} is removed. The concept d'_{cv} is the substitute for d_{cv} for sending messages. This is because the most general transferendum c_i that can be conveyed using d_{cv}, is a subconcept of d'_{cv}, and can therefore also be conveyed using d'_{cv}. The concept d''_{cv} is the substitute for d_{cv} for receiving messages. This is because d''_{cv} yields the same translatum c'_i as d_{cv}. Because d''_{cv} is more general than d_{cv} the other agent can convey its messages using d''_{cv} instead of d_{cv}.

For example, suppose that $C^a_{cv} = \{motorhome, roadvehicle, vehicle\}$. Ag-1 believes the concept *roadvehicle* to be redundant because *vehicle* satisfies the first item in the condition of Definition 5.13, and *motorhome* satisfies the second item.

Theorem 5.2. *If Ag-i considers a concept d_{cv} redundant (Definition 5.13), then concept d_{cv} is mutually redundant (Definition 5.12)*

Proof: We will prove the theorem for communication from Ag-i to Ag-j and from Ag-j to Ag-i.
From Ag-i to Ag-j: Observe that Ag-i never requires transferens d_{cv} to communicate

a transferendum c_i. Suppose that $c_i \leq d_{cv}$, which is a necessary condition for d_{cv} to qualify as a transferens. According to Definition 5.13 (1st bullet), d'_{cv} exists for which $c_i \leq d'_{cv} \leq d_{cv}$. Hence, Ag-i uses d'_{cv} as a transferens instead of d_{cv}. The same argument holds for the transferens $\neg d_{cv}$. The second bullet in 5.13 ensures that every local concept c'_i that is subconcept of $\neg d_{cv}$ is also subconcept of a subconcept of $\neg d_{cv}$, namely $\neg d''_{cv}$. Because conjunction and disjunction are compositionally defined, Ag-i would also never use d_{cv} or $\neg d_{cv}$ as a conjunct or disjunct either.
From Ag-j to Ag-i: For every concept c_j which Ag-j communicates using d_{cv}, Ag-j may also use d''_{cv}. The second bullet in Definition 5.13 ensures d''_{cv} yields the same translatum as d_{cv}, namely c'_i. Therefore, every concept c_j that is losslessly communicated using d_{cv} can also be losslessly communicated using d''_{cv}. Furthermore, Ag-i responds "OK" to messages with d''_{cv} (and thus recognizes lossless communication), because bullet 1 in Definition 5.13 ensures that all subconcepts in C_i of d_{cv} are also subconcepts of d''_{cv}. A similar argument can be made for Ag-j that communicates using $\neg d_{cv}$.

□

Theorem 5.2 states that agents can apply Definition 5.13 to *correctly* recognize redundant concepts. This raises the question whether *every* redundant concept is recognized using Definition 5.13. In other words, does Definition 5.12, in some sense, imply Definition 5.13? It appears that a redundant concept is always recognized by at least one agent. This is described in the following theorem.

Theorem 5.3. *If concept d_{cv} is mutually redundant (Definition 5.12), then either Ag-1 or Ag-2 considers it redundant (Definition 5.13)*

Proof: Let concept d_{cv} be a mutually redundant concept according to Definition 5.12. Note that, in our framework $C_{cv} \subseteq C_{1\cdot 2}$. Therefore $d \in C_1$ or $d \in C_2$. Let $j \in \{1,2\}$ be such that $d \in C_j$ and let $i \in \{1,2\}$ be unequal to j. We prove that d_{cv} is considered redundant by Ag-i according to Definition 5.13.
Because d_{cv} is mutually redundant, Ag-j can use a different concept than d_{cv} to communicate d_j to Ag-i. This concept is d''_{cv} which is most specific in the set of superconcepts of d_{cv} in the set $\mathcal{L}(C^a_{cv} \backslash \{d_{cv}\})$ (Def. 5.13, second bullet, first item). Let c'_i be the translatum that is generated by d_{cv}, i.e. c'_i is most specific in the set of superconcepts of d_{cv} in C_i (Def. 5.13, 2nd bullet, 2nd item). Because d_{cv} and d''_{cv} give rise to lossless communication, the translatum that is generated by d''_{cv} is also c'_i. Therefore d''_{cv} must be a subconcept of c'_i. This satisfies the second bullet in Definition 5.13.
Because d_{cv} is mutually redundant, Ag-j can use a different concept than $\neg d_{cv}$ to communicate $\neg d_j$ to Ag-i. This concept is $\neg d'_{cv}$ which is most specific in the set of superconcepts of $\neg d_{cv}$ in the set $\mathcal{L}(C^a_{cv} \backslash \{d_{cv}\})$. Therefore, d'_{cv} is most general in the set of subconcepts of d_{cv} in the set $\mathcal{L}(C^a_{cv} \backslash \{d_{cv}\})$ (Def. 5.13, 1st bullet, 1st item). Let $\neg c_i$ be the translatum that is generated by $\neg d_{cv}$, i.e. $\neg c_i$ is most specific in the set of superconcepts of $\neg d_{cv}$ in C_i. Therefore c_i is most general in the set of subconcepts of d_{cv} in C_i (Def. 5.13, 1st bullet, 2nd item). Because $\neg d_{cv}$ and $\neg d'_{cv}$ give rise to lossless communication, the translatum that is generated by $\neg d'_{cv}$ is also $\neg c_i$. Therefore $\neg d'_{cv}$

is a subconcept of $\neg c_i$ and d'_{cv} is a superconcept of c_i. This satisfies the first bullet in Definition 5.13.

□

Action `RemoveConcept`(Ag-j, d_{cv})
if Ag-i considers d_{cv} redundant **then**

- Remove d_{cv} from C^a_{cv}
- send(Ag-j, $\langle$RemoveConcept, $d\rangle$)

else fail

Action `Receive`($\langle$RemoveConcept, $d\rangle$)
Remove d_{cv} from C^a_{cv}

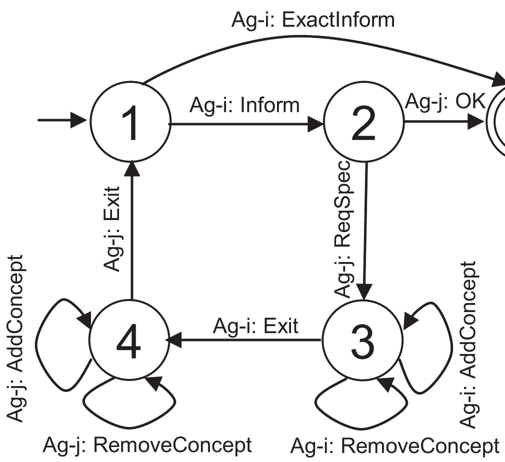


Figure 5.7: ONP-3

An agent performs a `RemoveConcept` action on a concept d_{cv}, when it considers it redundant using the criteria described in Definition 5.13. Concepts may become redundant after a new term is added to the communication vocabulary. Because both agents have different perspectives on the redundancy of terms, both agents get a chance to perform `RemoveConcept`. Sometimes, after *one* concept is added, *two* concepts can be removed from the communication vocabulary. To exploit this idea, also the receiver Ag-j is allowed to add concepts to the communication vocabulary (state 4). The second example below illustrates the application of this.

Example 5.11. Consider the ontologies in Figure 5.2, and that C^a_{cv} = {*motorhome*, *roadvehicle*}. Suppose that Ag-1 wishes to communicate *vehicle*(a)

	Sound and lossless	Lazy	Minimal cv
ONP-1	+	−	−
ONP-2	+	+	−
ONP-3	+	+	+

Figure 5.8: Evaluation of the protocols

Ag-1 : `Inform`(Ag-2, *vehicle*(*a*))
⌞Ag-1 : send(Ag-2, ⟨Inform, $\top(a)$⟩)
Ag-2 : `Reqspec`
Ag-1 : `AddConcept`(Ag-2,*vehicle*)
Ag-1 : `RemoveConcept`(Ag-1,*roadvehicle*)
Ag-1 : `Exit`
Ag-2 : `Exit`
Ag-1 : `ExactInform`(Ag-2,*vehicle*(*a*))
. . .

In this example Ag-1 considers the concept *roadvehicle* redundant after *vehicle* was added to it. As a sender, Ag-1 would never use *roadvehicle*, and as a receiver Ag-1 finds *vehicle* equally informative as *roadvehicle*.
As another example, consider the ontologies in Figure 5.3. Suppose that C^a_{cv} = {*RV-large*} and that Ag-2 intends to communicate *RV-XL(a)*

Ag-2 : `Inform`(Ag-1, *RV-XL*(*a*))
⌞Ag-2 : send(Ag-1, ⟨Inform, ¬*RV-large*(*a*)⟩)
Ag-1 : `Reqspec`
Ag-2 : `AddConcept`(Ag-1,*RV-XL*)
Ag-2 : `Exit`
Ag-1 : `AddConcept`(Ag-2,*Motorhome*)
Ag-1 : `RemoveConcept`(Ag-2,*RV-XL*)
Ag-1 : `RemoveConcept`(Ag-2,*RV-large*)
Ag-1 : `Exit`
Ag-2 : `Inform`(Ag-1, *RV-XL*(*a*))
⌞Ag-2 : send(Ag-1, ⟨Inform, *Motorhome*(*a*)⟩)
Ag-1 : `OK`

If the agents cleverly apply combinations of AddConcept and RemoveConcept (as in the above example), the protocol establishes an optimal communication vocabulary (Definition 5.8).

Because ONP-3 enables the agents to establish an optimal communication vocabulary, it scores better with respect to minimal cv construction than ONP-2. This makes ONP-3 the best protocol for ontology negotiation we have proposed in this chapter. A comparison of the three ontology negotiation protocols discussed in this section is shown in Figure 5.8.

5.4 Conclusion

In this chapter we have presented a formal analysis of ontology negotiation between two agents. The formal framework, that forms the backbone of this chapter, distinguishes six ontologies that differ in dynamics and in the way they are known by the agents. We have applied the framework for several purposes.

Firstly, we have used the framework as a specification of ontology implementation. We have shown how the specification can be met using description logics and concept classifiers. Secondly, we have formalized the design objectives for ontology negotiation introduced in Section 4.2.3. In particular, we have formalized sound and lossless communication and characterized different types of minimal ontologies. Thirdly, we have presented three formal protocols for ontology negotiation, two of which correspond to the ones presented in the previous chapter. The formal perspective adopted in this chapter enables us to guarantee sound and lossless communication. The last protocol we have discussed concerns a communication mechanism that enables agents to establish minimal ontologies by removing redundant concepts from their communication vocabularies. We have proven that their removal does not affect the expressivity of their vocabulary.

5.4.1 Future Work

The formal framework introduced in this chapter can be used for numerous other issues in ontology negotiation that require a precise and exact analysis.

One of these issues is an extension of the protocol that is more context-dependent. In this thesis, we assume that the hearer always wants as much information to be conveyed as its ontology can represent. The rationale for this is that an ontology is assumed to contain precisely those elements that are relevant for the agent's task (see Section 2.3.4). Of course, an agent may be designed to fulfil multiple tasks and its ontology might contain the sum of concepts that are relevant for any of its tasks. As a result, a concept might be relevant when the agent is involved in one task, and irrelevant when it is involved in another task. The lossless criterion may be adapted for this purpose, such that ontology negotiation becomes even more lazy. As proposed by van Rooy (2003), information can be regarded as relevant if it reduces the number of actions that an agent may choose between to solve a particular decision problem. In our work, we could abstract away from decision problems, and follow a simpler approach. By adopting a *contextual ontology language*, e.g. (Bouquet et al., 2003) or (Grossi et al., 2006), one agent might adopt multiple ontologies for different contexts (or tasks). With minimal revision, the formulae in this chapter can be adapted to take this context into account. What changes is that an agent may regard a conversation as lossless at one moment, and non-lossless at another moment. Also, ontology exchange may be altered such that an agent is only provided with those alignments that are currently relevant for the agent. What is studied in this chapter can be regarded as a special case in which there is only one context and everything is assumed to be relevant for the agent's task.

Another topic of future work is extending the ontology language with additional constructs, such as roles. In this chapter we have proposed non-compositional concept explication. Using this method, the framework can also be used when the ontology language contains roles. For example, to add the role r to a cv that contains concept c, the agent should explicate all concepts that can be formed using r and c, such as $\exists r.c$ and $\forall r.c$. As these expressions denote sets of individuals, they can be explicated using the techniques described in this chapter. However, it is not straightforward how to explicate a role itself, which denotes a set of *pairs* of individuals. Role explication might be desirable to enable the agents to exchange role assertions with each other. Furthermore, it could make the concept explication protocol more compositional. For example, having explicated the role r and the concept c, the other agent would also know the meaning of $\forall r.c$ and $\exists r.c$. This would reduce the number of concept explications.

The work described in this chapter views ontology negotiation from a formal perspective. In the next chapter, we will be concerned with issues that require an investigation from a simulation perspective, namely communication strategies.

Chapter 6

Simulation-based Strategy Analysis

> *Suppose that with practice we could adopt any language in some wide range. It matters comparatively little to anyone (in the long run) what language he adopts, so long as he and those around him adopt the same language and can communicate easily.*
>
> David Lewis (Convention, pp.7-8,1969)

This chapter presents a simulation-based analysis of ontology negotiation. The chapter serves two purposes.

Firstly, it serves to investigate the properties of ontology negotiation in larger systems. Whereas an ontology negotiation protocol provides a nice solution to incrementally establish a communication vocabulary between a *pair* of heterogeneous agents, how this solution scales to *whole* multi-agent systems is not straightforward.

Secondly, the chapter proposes communication strategies that can be used in combination with the ontology negotiation protocols discussed in the previous chapters. The communication mechanisms as discussed until now are not fully deterministic, i.e. they give an agent some freedom of choice as to which particular message it sends. The communication strategies described in this chapter enable the agents to make this choice.

In this chapter, we will describe simulation experiments carried out to investigate the properties of the proposed ontology negotiation strategies. For the purpose of performing efficient simulation experiments, we use a simplified model of the ontologies and the communication protocols. However, we only abstract away from those properties that are irrelevant to the communication strategies under investigation. Thus, the core of the ontologies and communication mechanisms of ANEMONE is left intact.

The chapter is organized as follows. Section 6.1 motivates the need for ontology negotiation strategies. Related work is discussed in Section 6.2. Section 6.3 presents the model that is used to investigate the communication strategies. Section 6.4 investigates communication strategies for word selection. Section 6.5 investigates strategies for meaning selection. We conclude in Section 6.6.

The chapter is based on (van Diggelen, Beun, Dignum, van Eijk and Meyer, 2005) and (van Diggelen, Wiering and de Jong, 2006c).

6.1 Ontology Negotiation Strategies

A decentralized approach such as ontology negotiation may give rise to a proliferation of different cv's between different pairs of agents. This would be disadvantageous, as agents would have to use different words with different agents, which would make communication unnecessarily complicated. Furthermore, agents would have to spend much effort on building cv's, as the cv that has been built up with one agent may not be useful for communication with another agent. Therefore, when two agents participate in ontology negotiation to resolve their mutual misunderstandings, they should also pursue the goal of establishing a uniform and effective cv for the benefit of the whole community.

In this chapter, we will describe communication strategies for ontology negotiation protocols that take this global goal into account. These strategies prescribe which words and meanings the agents should teach each other during ontology negotiation.

Regarding the words, we aim for a situation where every agent uses the same unique word for the same meaning. This is to be established by the agent's *word selection strategy* (WSS) which is studied by van Diggelen et al. (2005). One of the criteria of a word selection strategy is that it should give rise to a cv which is minimal in size and requires a minimal effort from the agents (see lazy and minimality requirements in Section 4.2.3). Multiple terms with the same meaning (synonyms) are therefore not preferable as they increase the size of the cv without adding anything to its expressivity. Furthermore, they needlessly require the agents to learn extra concepts.

Regarding the meanings, we aim for a communication vocabulary which enables the agents to communicate at the right level of generality (van Diggelen et al., 2006c). Agents with different areas of expertise should not communicate at an *overspecific* level, as not everything that is of interest to one agent is also of interest to another agent. To prevent the cv from becoming bulky and difficult to learn, the cv should not contain such overspecific meanings. Furthermore, to enable the agents to convey sufficient information, the meanings in the cv should also not be *overgeneralized*. In other words, we aim for an optimal communication vocabulary distribution (Definition 5.11). The right balance between specificity and generality of words must be established by the *meaning selection strategy* (MSS). We will show how a well-designed meaning selection strategy contributes to faster achievement of semantic interoperability in the group of agents.

6.2 Related Work

This chapter addresses the problem of producing order in the communication vocabulary that emerges during ontology negotiation. Because multi-agent systems usually lack a centralized control, the problem must be tackled fully decentralized. Such an organization process that relies on internal interactions between distributed entities rather than being controlled by an external component is called *self-organization*. Much research in AI is devoted to self-organization. Its application on language development is particularly relevant here. In this section, we will briefly describe related work on language development and discuss the differences and similarities with our work.

As argued by Steels (2000), a group of communicating agents can be viewed as a complex adaptive system which collectively solve the problem of establishing a shared lexicon. This problem comprises reaching agreements on a repertoire of words (concept names), a repertoire of meanings (a conceptualization) and a repertoire of word-meaning pairs (an interpretation). Steels' research addresses all three aspects. *Imitation games* are used to study the formation of speech sounds (words), *discrimination games* are used to study the formation of a conceptualization and *naming games* are used to study the formation of word-meaning pairs.

In our work, we assume that words are already provided by the agents' native concept names. Imitation games are therefore not useful to us. Because the agents' conceptualizations are also given from start, discrimination games are not required either. The naming games, on the other hand, address a problem that is similar to ours. Below, we will briefly describe the issues involved in the formation of word-meaning pairs.

Many authors have studied the issue of self-organization to achieve a globally shared set of word-meaning pairs, e.g. de Jong and Steels (2003); Barr (2004); Wang et al. (2006). A common characteristic of this research is the use of *simulation experiments* to study the influence of the agent's local behavior on the emerging communication vocabulary. Of particular interest is to see whether the communication vocabulary *converges*, i.e. whether every agent eventually uses the same word for the same meaning.

In these respects, the research described in this chapter is not different. We investigate communication strategies using simulation experiments and judge their performance according to, among other things, convergence properties. However, our research differs in other respects. The assumptions underlying the models of the above-cited articles are based on what is considered realistic from a human perspective. For example, ambiguity (one word referring to multiple meanings) plays a central role in these investigations. Because the communication vocabulary becomes ineffective when different agents ascribe different meanings to the same word, these agents must also be capable of *unlearning* a word-meaning pair. Such agents must be able to strengthen and weaken their word-meaning associations, for example, using reinforcement learning techniques (Wang et al., 2006).

Our research, on the other hand, is based on different assumptions which need not necessarily be realistic in a human environment. Our primary goal is not explanatory but constructive. We intend to provide a strategy which guides the agents which meanings and words to select when the ontology negotiation protocol leaves several options open. ANEMONE is based on the assumption that every agent possesses a unique set of native concept names (see Section 4.1.2). Therefore, ambiguous words do not occur in our framework. This alleviates the need for agents to unlearn word-meaning couplings. Therefore, the techniques for word selection strategy we propose in this chapter are simpler than reinforcement learning. For some strategies, we will show that agents using them succeed in establishing a converged communication vocabulary.

Whereas techniques with similar aims and goals to our word selection strategy have been studied more than once in the language evolution community, this is not the case for meaning selection strategies. It is well known in linguistics that people do not always speak as specific as possible, but occasionally communicate using more general words. Grice's maxim of quantity (Section 2.2.1) is a good example of this. However, our aims for meaning selection strategies are concerned with *language development*. The choice of the speaker is whether to speak a specific word which contains much information but might be unknown to the hearer, or to speak a general word which contains less information but is more likely to be known by the hearer. We believe this to be an interesting issue in language evolution and ontology negotiation. Moreover, it is a necessary component of ANEMONE, as agents using this communication mechanism simply face such choices.

6.3 Model

For the purpose of performing efficient simulation experiments, we use a simplified model that abstracts away from some issues that are irrelevant to the properties under investigation. Nevertheless, the core of the ontologies and communication mechanisms of ANEMONE is left intact. We will first briefly describe agents, ontologies and the communication protocols. Then, we will describe some measures of semantic interoperability that will be used throughout this chapter.

6.3.1 Modeling Agents and Ontologies

A multi-agent system consists of a set of agents and an interaction scheme that describes which agents are connected. A pair of connected agents is a pair that is capable of communicating.

Definition 6.1. *A multi-agent system is a network of agents, with the following elements:*

- *MAS = {Ag-1 .. Ag-n} denotes the set of agents (where n is the number of agents)*
- *CAG ⊆ MAS × MAS is the set of pairs of connected agents*

We assume that an agent's ontology is a simple taxonomic structure which can be specified using a *rooted tree*. A rooted tree is an acyclic connected graph (V, E), where V is a set of vertices, E is a set of directed edges and a particular vertex in V

is designated as the root. We adopt the following terminology from graph theory (Malik and Sen, 2004).

Definition 6.2. *Given a rooted tree* (V, E)*, we define the following:*

- *A vertex* v_j *is a child of vertex* v_i *iff* $\langle v_i, v_j \rangle \in E$
- *A vertex with no children is a leaf*
- *A vertex that is not a leaf is an internal vertex*
- *A vertex* v_j *is a descendant of vertex* v_i *iff there is a directed path from* v_i *to* v_j
- *A vertex* v_j *is a ancestor of vertex* v_i *iff* v_i *is a descendant of* v_j
- *The level number of a vertex* v_i *(denoted by* $ln(v_i)$*) equals the length of the unique directed path from* v_0 *to* v_i*, where* v_0 *is the root of the tree.*
- *The depth of a tree is the largest level number achieved by a vertex in that tree.*

The conceptualization that underlies an agent's ontology is specified as a rooted tree $\rho = \langle V, E \rangle$. The set of vertices V in the tree corresponds to the set of meanings that can be represented. Whereas in previous chapters we have modeled a meaning as a subset of the domain of discourse, here we will simply assume that a meaning is a vertex in a tree. The set of edges E in the tree specify the relations between meanings. An edge from vertex (or meaning) m_i to m_j represents that meaning m_j is more specific than m_i. This corresponds to what has been modeled using the subset relation in earlier chapters. The consequence of modeling a conceptualization as a rooted tree is that the overlap relation no longer exists. However, since most of our work involves dealing with sub- and superconcept relations, this omission is insignificant for the purposes of this chapter.

An ontology introduces names for the elements in the conceptualization. These are given by the set C. We assume that the agents know the intended interpretation of the concept names. In previous chapters, we have argued that this requires the agents to have concept classifiers. Here we simply assume that the agents know the (intended) interpretation function $\mathcal{I} : C \to V$. An ontology can thus be defined as follows:

Definition 6.3. *The ontology of agent Ag-i is specified as* $O_i = \langle C, \rho, \mathcal{I} \rangle$*, where*

- C *is the set of concept names.*
- $\rho = \langle V, E \rangle$ *is the conceptualization consisting of*
  - $V = \{m_1..m_n\}$*: the set of meanings*
  - $E \subseteq V \times V$*: a set of subset relations between meanings*
- $\mathcal{I} : C \to V$ *is a bijective mapping from* C *to* V *which specifies the intended interpretation of the concept names.*

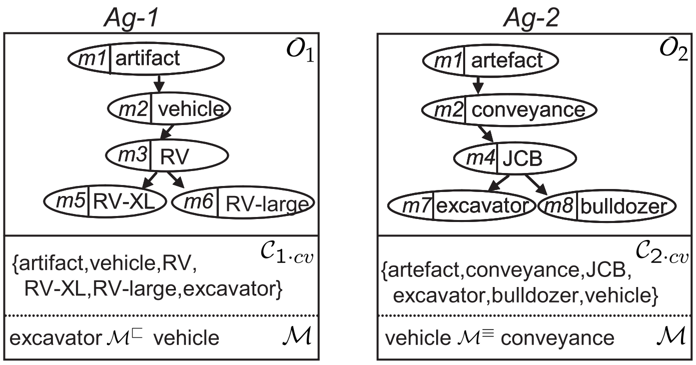


Figure 6.1: Example ontologies

A communication vocabulary consists of a set of concepts in C_{cv}. The concepts in C_{cv} are also referred to as words. Like in previous chapters, words are given semantics by specifying the relations with the concepts in the ontology. This is done by $\mathcal{M}$ that states which words are equivalent or more specific than which concepts in the ontology.

Unlike previous chapters, we assume that the agents do not keep track of which concepts are shared with other agents. This is because, in this chapter, we are interested in communication between agents that have never met before. Successful communication between these agents therefore hinges on unknowingly shared concepts (see Section 4.2.1).

Definition 6.4. *The communication vocabulary of agent Ag-i is specified as* $O_{i\cdot cv} = \langle C_{cv}, \mathcal{M}\rangle$*, where*

- C_{cv} *is a set words.*
- $\mathcal{M} = \langle \mathcal{M}^{\equiv}, \mathcal{M}^{\sqsubset}\rangle$*, where:*
  - $\mathcal{M}^{\equiv} : C_{cv} \to C$ *states which words are equivalent with which concepts.*
  - $\mathcal{M}^{\sqsubset} : C_{cv} \to C$ *states which words are more specific than which concepts.*

We will write $\mathcal{M}(c)$ to refer to $\mathcal{M}^{\equiv}(c)$ if c is in the domain of $\mathcal{M}^{\equiv}$ and to $\mathcal{M}^{\sqsubset}(c)$ if c is in the domain of $\mathcal{M}^{\sqsubset}$. Similar to what has been discussed in NCP draft 6 (Figure 4.10), the communication vocabulary of an agent includes all of its native concepts, i.e. $C \subseteq C_{cv}$ and $\{\langle c, c\rangle | c \in C\} \subseteq \mathcal{M}^{\equiv}$.

An example of two agents is given in Figure 6.1. As appears in this figure, the meanings expressible by the ontologies are matching at a general level, but differ at a more specific level. Furthermore, the ontologies attach different names to the meanings in the conceptualizations. The communication vocabulary contains all concept names in the ontology with some additional words that have been learned

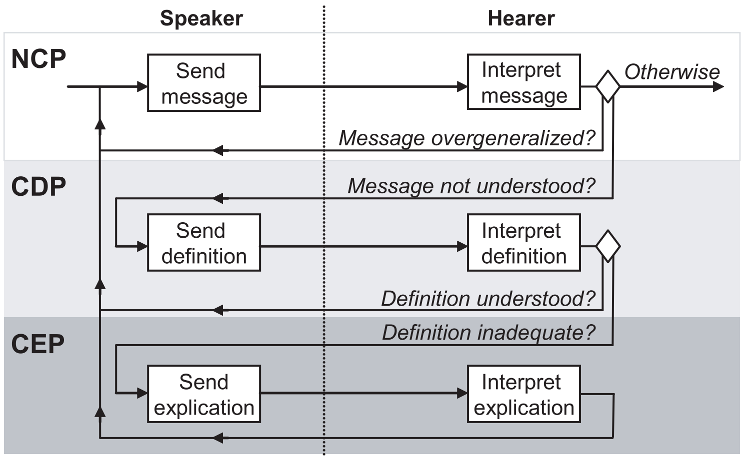


Figure 6.2: Overview of the protocol

from the other agent. The agents do not know which concepts are shared with each other, neither which words they should use to communicate with each other.

6.3.2 Communication

Figure 6.2 presents an overview of the communication protocol that is used in our experiments. The communication protocol is largely equivalent to the protocol discussed in Chapter 4. However, because we use simpler ontologies and communication vocabularies, a few simplifications can be carried through in the communication protocol as well.

The simplified NCP layer is briefly discussed below, starting with message interpretation. After the hearer has interpreted the message, it may respond in three ways. The conditions for this are as follows:

1. *Message overgeneralized*: the hearer Ag-j considers a word d overgeneralized iff $\mathcal{I}(\mathcal{M}(d))$ is an internal vertex in ρ_j and d is sent in an "Inform" message. When a message is considered overgeneralized, the hearer requests for a more specific message.

2. *Message not understood*: the hearer Ag-j does not understand a word d iff $d \notin C_{j\text{-}cv}$.

3. *Otherwise*: the hearer responds "OK", and the conversation finishes.

In message composition, the speaker should choose a meaning and a word to convey what it intends to convey. With respect to meanings, it may choose any meaning that is as general as, or more general than what it intends to convey. Formally, we define the set of meaning candidates as follows:

Definition 6.5. *MC: Meaning Candidates*
Given a concept c that must be conveyed, the set of meaning candidates is given by:
$MC = \{m | m = \mathcal{I}(c) \text{ or } m \text{ is an ancestor of } \mathcal{I}(c) \text{ in } \rho\}$

For example, the set of meaning candidates for Ag-1 to communicate *RV* (Figure 6.1), consists of *m3*, *m2* and *m1*. The meaning selection strategy can be described as: select a good meaning candidate. A good meaning selection strategy anticipates on the hearer's reaction, i.e. it selects a meaning which will likely not be regarded as overgeneralized by the hearer. For example, *m2* and *m1* would be bad choices as these would be considered overgeneralized by Ag-2.

After the speaker has decided on a meaning, it must choose a word that corresponds to this meaning.

Definition 6.6. *WC: Word Candidates*
Given a meaning m, the set of word candidates for m is given by:
$WC = \{c | c \in C_{cv} \wedge \mathcal{I}(\mathcal{M}^{\equiv}(c)) = m\}$

The word selection strategy can then by described as: select a proper word candidate. For example, the set of word candidates for Ag-2 to communicate meaning *m2* (Figure 6.1), consists of *conveyance* and *vehicle*. A good word selection strategy anticipates on the hearer's reaction, i.e. it selects a word that will most likely be understood by the hearer. In the example, *vehicle* would be a good choice, as this word is known by the hearer.

We can now summarize the process of message composition, and state precisely how the meaning and word selection strategies fit in the protocol:

1. Given that the speaker intends to convey c.
2. Use meaning selection strategy to select m from *MC*, where *MC* is the set of meaning candidates for c
3. Use word selection strategy to select d from *WC*, where *WC* is the set of word candidates for m
4. Communicate d to the hearer. Use an "ExactInform" message if $\mathcal{I}(c) = m$, use an "Inform" message otherwise.

6.3.3 Interoperability Measures

One of the measures that will be used to judge the quality of a communication strategy, is the speed at which semantic interoperability is achieved. We will now define some measures for understandings rate, i.e. measures for the degree of semantic interoperability. Suppose that Ag-i wishes to communicate a meaning m to Ag-j. If Ag-i can do this in only the NCP layer (the upper layer in the protocol of Figure 6.2), the understandings rate between Ag-i and Ag-j with respect to meaning m is 1; if the agents have to visit the CDP or CEP layer, the understandings rate is 0.

Definition 6.7. *MPUR: Meaning and Pair dependent Understandings Rate.*
$MPUR(m, \langle Ag\text{-}i, Ag\text{-}j \rangle)$ *is*

- *1 if the conversation to communicate m from Ag-i to Ag-j finishes without visiting the CDP and CEP layer*
- *else 0.*

The following measure indicates how well an agent Ag-i can communicate an average concept to Ag-j (O_i is defined as a tuple $\langle C_i, \langle V_i, E_i \rangle, \mathcal{I}_i \rangle$, conforming to Definition 6.3):

Definition 6.8. *PUR: Pair dependent understandings rate*
$PUR(\langle Ag\text{-}i, Ag\text{-}j \rangle) = \frac{1}{\#V_i} \sum_{m \in V_i} MPUR(m, \langle Ag\text{-}i, Ag\text{-}j \rangle)$

The following measure indicates how well an average agent can communicate an average meaning to an average other agent.

Definition 6.9. *UR: Understandings rate*
$UR = \frac{1}{\#CAG} \sum_{\langle Ag\text{-}i, Ag\text{-}j \rangle \in CAG} PUR(\langle Ag\text{-}i, Ag\text{-}j \rangle)$

If the understandings rate is 1, every agent can communicate everything to every other agent. We will call this situation *common understanding*.

6.4 Word Selection Strategies

We begin by investigating word selection strategies, i.e. strategies that prescribe which word an agent should choose to convey a certain meaning. Meaning selection strategies are investigated in Section 6.5.

6.4.1 Description of Strategies

The decision to be solved by the agent's word selection strategy involves choosing a concept-name among the possible word candidates when sending an inform message. We assume that `Select` always returns the concept-name with the highest score. The difference between the strategies is in the way the scores are updated after a message is received.

WSS-1 only attributes a score to its local concept names. Therefore, in a system with agents that use WSS-1 (a WSS-1 system), every agent holds on to its own native concept names when it comes to speaking. To understand other agents, they learn each other's concept names. This strategy is analogous to point-to-point ontology alignment (discussed in Section 2.4.2): every agent has a mapping to every other agent's ontology.

WSS-2 attributes a score of 1 to the most recently received concept-name; all other concept-names with the same definition are attributed a score of 0. In WSS-2 systems, an agent chooses the candidate word it has most recently received from another agent.

WSS-3 increases the score of a word each time a message with that word is received. In this way, the agents in a WSS-3 system choose the candidate word which they have most frequently received.

WSS-4 is similar to WSS-3 but also takes into account *which* agents have used the word. The score of a word is increased more when an agent with many acquaintances uses the word than when an agent with few acquaintances uses the word. We assume that the number of acquaintances of an agent can be known by the agents. In systems where every agent has an equal amount of acquaintances, WSS-4 gives the same results as WSS-3. We therefore only discuss WSS-4 at the end of Section 6.4.2, where we consider networks in which the agent's number of acquaintances differ.

The four communication strategies are specified in Figure 6.3.

6.4.2 Experiments

We evaluate the performance of these strategies using simulation experiments. To obtain a clear picture of the properties we are interested in, we abstract away from as many irrelevant aspects as possible. The agents only exchange one meaning during the experiments, but may use different words to do so. Initially, the situation is as follows:

- for all Ag-i $\in$ MAS: $\#V_i = 1$ (every ontology consists of one meaning)
- for all Ag-i, Ag-j $\in$ MAS: $V_i = V_j$ (every ontology contains the same meaning)
- for all Ag-i, Ag-j $\in$ MAS: $C_i \neq C_j$ (every ontology uses a different name for this meaning)

An experiment consists of t steps at which randomly a speaker and a receiver are selected from the set of connected agents. The agents follow the communication protocol as described in Section 6.3.2.

We evaluate the communication strategies using the following criteria:

- the speed at which it increases the understandings rate
- the number of different words that are used in the system
- the performance of the strategy in sparsely connected networks

We will investigate these issues using three experiments that are described next.

Performance regarding understandings rate

It is useful to distinguish between the words that an agent uses as a speaker to convey a certain meaning and the words that it understands as a hearer as referring to a certain meaning. Because in these experiments, the agents only exchange one meaning, we will refer to these simply as the *spoken words* (SW) and the *understandable words* (UW). The understandable words of Ag-i are exactly those in $C_{i \cdot cv}$. The

// The Select action is the same in every strategy
// Select returns the concept from Word candidates (WC) with the highest score
// If several concepts have an equal highest score, it randomly picks one from them
Action `Select`(WC)
Randomly choose y from $\{y'|y' \in WC$ and for all $z \in WC$: $\mathrm{SCR}(z) \leq \mathrm{SCR}(y')\}$
return y

WSS-1:
// Always speak your own concept name.
// The scores of the agent's private concept names are kept equal on 1.
// The score of every other word therefore remains 0.
Action `UpdateScore`(Ag-i, x)
For all $y \in \mathcal{C}_{cv} \cap \mathcal{C}$ Do $\mathrm{SCR}(y) := 1$

WSS-2:
// Speak the word which most recently another agent used
Action `UpdateScore`(Ag-i, x)
SCR(x) := 1
For all $y \neq x$ with $\mathcal{M}(y) = \mathcal{M}(x)$ Do $\mathrm{SCR}(y) := 0$

WSS-3:
// Speak the most frequently used word
Action `UpdateScore`(Ag-i, x)
$\mathrm{SCR}(x) := \mathrm{SCR}(x) + 1$

WSS-4:
// Speak the most frequently used word weighted by the user's number of acquaintances
// The number of acquaintances of Ag_i is denoted by k(Ag)
Action `UpdateScore`(Ag-i, x)
$\mathrm{SCR}(x) := \mathrm{SCR}(x) + \mathrm{k(Ag\text{-}i)}$

Figure 6.3: Implementations of different strategies

spoken words of Ag-i are those that are selected by the word selection strategy, i.e. those words in $C_{i\cdot cv}$ with the highest score. Note that in WSS-1 and WSS-2 systems, there is only one spoken word per agent, because there is only one word with a highest score. In WSS-3 and WSS-4 systems, several words may have an equal highest score, amongst which the strategy randomly chooses one. SW and UW are formally defined as follows:

- $SW_i = \{c | c \in C_{i\cdot cv} \wedge \forall d \in C_{i\cdot cv}\ \mathrm{SCR}(d) \leq \mathrm{SCR}(c)\}$
- $UW_j = C_{j\cdot cv}$

In these experiments, the agents only visit the CDP and CEP layers of the protocol when the hearer does not understand the word that is used by the speaker, i.e. when the spoken words are not included in the understandable words. Therefore, we can simplify Definition 6.7 as follows:

- $MPUR(\langle$ Ag-i, Ag-j $\rangle) = 1$ iff $SW_i \subseteq UW_j$

As there is only one meaning in the system, the PUR (Definition 6.8) is equal to MPUR (Definition 6.7). The definition of understandings rate (6.9) boils down to the probability that $SW_i \subseteq UW_j$ holds for an average connected pair of agents $\langle$Ag-i, Ag-j$\rangle$.

Eventually, all communication strategies result in an understandings rate of 1, i.e. common understanding. This is because the communication protocol prescribes that in case of an unknown word, the unknown word is taught to the ignorant agent. Because the number of different words in the system is finite, every agent will eventually acquire a set of understandable words that enables it to understand every other agent. Nevertheless, the communication strategies differ with respect to the number of steps that are required to achieve common understanding. Following strategy 1, common understanding is achieved only after *every* pair of connected agents has communicated with each other. Given that at each time step a random pair of agents is selected, we can analytically derive the expected number of steps that are required to have every pair of agents selected at least once. Probability theory predicts that the expected number of steps before UR = 1 can be calculated using the following formula (Rice, 1995, p.120):

$E(X) = q \sum_{r=1}^{q} \frac{1}{r}$, where

- X is the number of steps it requires to reach common understanding
- $q = \#\mathrm{CAG}$

Consider a WSS-1 system where n=20 and CAG = MAS × MAS, i.e. every agent speaks to every other agent. In this system, the above formula predicts that common understanding is reached after approximately 2400 turns. This prediction is confirmed by the experimental results shown in Figure 6.4.

The experiment also reveals that WSS-2 and WSS-3 require much fewer steps than WSS-1 to reach common understanding. We argue that this is because WSS-2

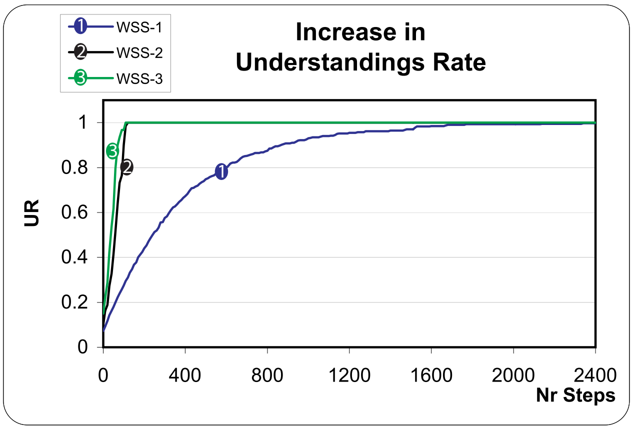


Figure 6.4: Strategy performance w.r.t. Understandings Rate

and WSS-3 incites the agents also to speak each other's words which gives rise to groups of agents that speak the same word. Therefore, two agents from the same group that never communicated before, are able to understand each other nevertheless. This explains the fast increase of UR in WSS-2 and WSS-3 systems.

Besides the speed of increase in UR , another important issue is the *size* of the communication vocabulary. WSS-1 gives rise to a communication vocabulary of size n, i.e. every agent's private concept name eventually becomes part of the communication vocabulary. Besides the fact that this is the main cause for WSS-1's slow increase in UR, another disadvantage is that newcomers in the system would have to learn a large number of words to be able to understand everyone. It is therefore desirable that the total number of different words which are spoken by the agents is as small as possible. This aspect is studied in the following section.

Performance regarding number of words

We refer to the number of words that are used in the system with NW which is defined as follows:

Definition 6.10. $NW = \# \bigcup_i SW_i$

Obviously, in a WSS-1 system the NW remains equal on n. For WSS-2, it holds that, eventually, the NW becomes 1 in which case we say that the communication vocabulary has *converged*.

Property 6.1. *In every WSS-2 system:* $\lim_{t\to\infty} P(\mathit{NW{=}1 \text{ after } t \text{ steps}}) = 1$

Proof: In every WSS-2 system, the probability that NW becomes 1 after n steps is greater than 0 (recall that n is the number of agents). This happens when n times a speaker is selected with $SW = \{x\}$, and every agent with $SW \neq \{x\}$ is selected at least one time as a hearer. Furthermore, each "trial" (the execution of n steps) is independent of the other trials: the failure of one trial to result in NW=1, does not systematically influence the probability that the next trial results in NW=1. Therefore, by definition, as the number of steps approaches infinity, the probability that NW=1 approaches 1.

□

Although the property described above is a nice theoretical result, in practice we are interested in the speed at which NW decreases. To obtain statistical significance, in the next experiment we use n=1000. Again, we set the structure as a fully connected network, i.e. CAG = MAS $\times$ MAS. Figure 6.5 shows the decrease of NW in this experiment. The results of this experiment show that the WSS-2 system gives rise

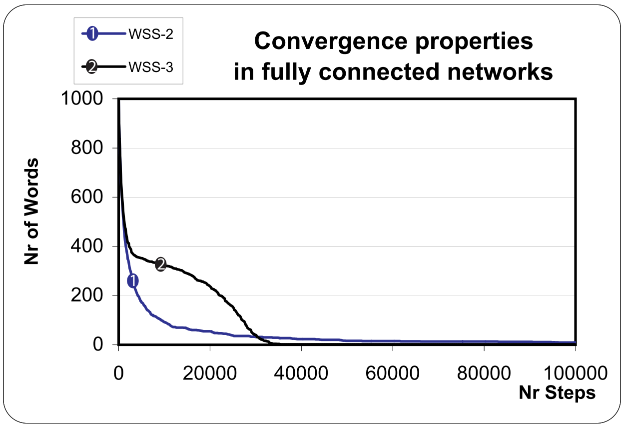


Figure 6.5: Strategy performance w.r.t. number of words

to a great decrease of NW at first, but does not lead to a fully converged cv. After 100000 steps, still 9 different words were in use, and it would have taken a very long time before the cv would have converged. WSS-3, on the other hand, performs relatively poorly at first, but gives rise to a converged cv after approximately 35000 steps. By that time, every agent spoke 35 times on average, which is not a bad result given the large size of the system.

The reason why the cv does not converge within reasonable time in a WSS-2 system, is that in these systems NW decreases only by coincidence. At the end, the probability becomes very small that one of the words "dies out", because each word is used by many agents. Agents that follow WSS-3, maintain an estimation of which words are most frequently used. Because every agent uses the word which they believe to be most frequently used, NW not only decreases by coincidence, but is guided by the beliefs of the agents.

Because WSS-3 gives rise to a stable and converged cv, WSS-3 is preferable over

WSS-2. In the next section, we discuss the strategy's performance in other network structures.

Performance regarding network structure

In the previous sections, we have evaluated the strategies in fully connected agent networks. In this section we discuss the strategy's performance in a more sparsely connected network. In doing so, we adopt some terminology from graph theory. We assume that the network is a non-directed graph, i.e. $\langle x, y\rangle \in \text{CAG} \rightarrow \langle y, x\rangle \in \text{CAG}$. Furthermore, we assume that the graph $\langle \text{MAS}, \text{CAG}\rangle$ is *connected*, i.e. it is possible to move between any pair of vertices by moving along the edges of the graph.

We call the number of acquaintances of an agent, the *degree* of an agent, denoted by k:

Definition 6.11. $k(Ag_i) = \#\{Ag_j | \langle Ag_i, Ag_j\rangle \in CAG\}$

Many networks, among which the world-wide-web, are structured as a *scale-free network* (Barabasi and Albert, 1999). Networks of this type are characterized by a large number of nodes with a relatively small k. A few nodes, however, are stars in the network and have a relatively high degree. Stated more precisely, the degree distribution follows a power law: $P(k) \sim k^{-\gamma}$, where $P(k)$ denotes the probability that a node has k edges.

The next experiment describes the results of WSS-3 and WSS-4 in a scale-free network, with n=1000, an average k of 3.11, and a maximum k of 50.

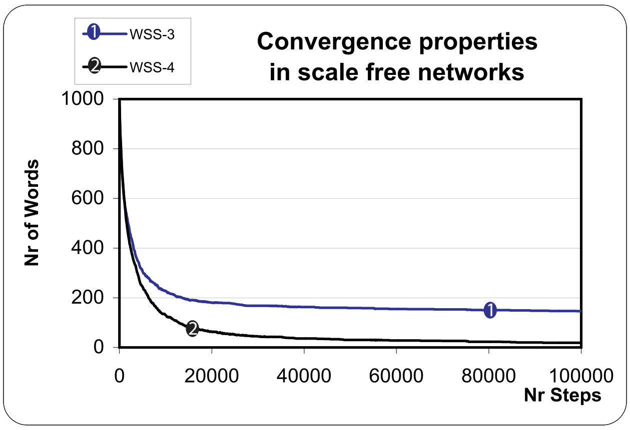


Figure 6.6: Strategy performance w.r.t. NT in a scale-free network

Although WSS-3 gave rise to a converged communication vocabulary in fully connected networks within reasonable time, the NT does not go below 140 in the experiment described above. This happens because most agents have a low degree and are not capable to form a realistic estimation of which terms are most frequently used. To overcome this problem, WSS-4 using agents take into account the degree of the speaking agent when they update their scores. Agents with a high degree have a more realistic estimation of the most frequently used sign, and are therefore taken more seriously than agents with a low degree. This explains why WSS-4 performs better than WSS-3 in this experiment, although it still does not give rise to a fully converged communication vocabulary.

6.5 Meaning Selection Strategies

In this section we investigate meaning selection strategies, i.e. strategies that prescribe at which level of generality an agent should speak.

6.5.1 Model

To be able to study the properties of a meaning selection strategy, we must provide the agents with ontologies that are randomly created within certain bounds. On the one hand, the meanings that are expressible by the agents' ontologies should exhibit some kind of overlap. Otherwise, there would be no reason for the agents to exchange information. On the other hand, the conceptualizations underlying the agents' ontologies should not be completely the same either. This would not only be unrealistic, it would also make a meaning selection strategy obsolete. Choosing a meaning at the right level of generality only becomes useful when the agents have different areas of expertise. In the following, we will introduce some notions that enable us to characterize the different ontologies in the model.

The following definition is useful to characterize the shape of a conceptualization.

Definition 6.12. *A conceptualization* $\rho = (V, E)$ *is defined according to* $B = (b_0, .., b_d)$ *if:*

- d *is the depth of the tree* ρ
- *for each* $v_i \in V$, v_i *has* $b_{ln(v_i)}$ *children*

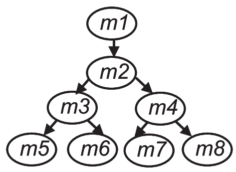


Figure 6.7: Conceptualization

For example, the conceptualization in Figure 6.7 is defined according to (1,2,2,0), because $m1$ (at level number 0) has 1 child; $m2$ (at level number 1) has 2 children;

$m3$ and $m4$ (at level number 2) have 2 children; $m5$, $m6$, $m7$ and $m8$ (at level number 3) have 0 children.

Given a conceptualization, defined as a tree ρ, we can obtain a smaller conceptualization by *pruning* subtrees from ρ (Russel and Norvig, 2003). For example, the conceptualization that underlies the ontology of Ag-1 (Figure 6.1) is the same as the conceptualization depicted in Figure 6.7, with the subtree containing $m4$,$m7$ and $m8$ pruned away. We will use this idea to obtain randomly created ontologies with conceptualizations that are different, yet not completely distinct.

The notion is formalized as follows:

Definition 6.13. *Given an ontology* $\mathcal{O} = \langle \mathcal{C}, \rho, \mathcal{I} \rangle$*, where* $\rho = (V, E)$*.* $\mathcal{O}$ *is defined according to* B *and* B_g *if*

- ρ *is defined according to* B*, and*
- $V \subseteq V'$, $E \subseteq E'$*, where*
  - $\rho' = (V', E')$ *is a conceptualization defined according to* B_g*.*

For example, the ontologies of Ag-1 and Ag-2 in Figure 6.1 are defined according to $B = (1, 1, 2, 0)$ and $B_g = (1, 2, 2, 0)$.

6.5.2 Description of Strategies

Using the different interoperability measures introduced in Section 6.3.3, we can characterize overgeneralized and overspecific concepts in more depth.

From a god's eye view

Property 6.2. *Teaching overgeneralized concepts does not increase MPUR (Definition 6.7).*

We will illustrate this property using the example where Ag-2 intends to communicate *m8* (the meaning of "bulldozer") to Ag-1. Suppose Ag-2's meaning selection strategy selects the overgeneralized meaning *m1* (the meaning of "artefact"). Before Ag-2 sends this message, $MPUR(m8, \langle \text{Ag-2}, \text{Ag-1} \rangle) = 0$ (because Ag-1 does not understand the word "artefact"). After Ag-2 has taught the concept "artefact" to Ag-1, $MPUR(m8, \langle \text{ Ag-2}, \text{Ag-1} \rangle)$ still equals 0 (because "artefact" invokes a "ReqSpec" response and Ag-2's second attempt to convey *m8* fails). Now suppose that Ag-2's meaning selection strategy selects the meaning *m4* (corresponding to the word "JCB"). This meaning is not overgeneralized, because $MPUR(m8, \langle \text{Ag-2}, \text{Ag-1} \rangle)$ becomes 1 after the concept "JCB" has been taught to Ag-1 (because "JCB" invokes an "OK" response).

Property 6.3. *Teaching overspecific concepts gives rise to little increase in PUR (Definition 6.8).*

Consider again the situation where Ag-2 intends to communicate *m8* ("bulldozer") to Ag-1. Suppose that the CV's of Ag-1 and Ag-2 are still in their initial configuration, i.e. they only contain the names of their native concepts. Suppose that Ag-2's meaning selection strategy selects the meaning *m8* (corresponding to the word "bulldozer"). Before Ag-2 sends this message, $PUR(\langle \text{Ag-2}, \text{Ag-1}\rangle) = 0$ (Ag-2 cannot communicate anything to Ag-1). After Ag-2 has taught the word "bulldozer" to Ag-1, $PUR(\langle \text{Ag-2}, \text{Ag-1}\rangle) = \frac{1}{5} \cdot MPUR(m8, \langle \text{Ag-2}, \text{Ag-1}\rangle) = \frac{1}{5}$. Now, suppose that Ag-2's meaning selection strategy would have selected "JCB". After Ag-2 has taught the word "JCB" to Ag-1, $PUR(\langle \text{Ag-2}, \text{Ag-1}\rangle) = \frac{1}{5} \cdot (MPUR(m8, \langle \text{Ag-2}, \text{Ag-1}\rangle) + \frac{1}{5} \cdot MPUR(m7, \langle \text{Ag-2}, \text{Ag-1}\rangle) + \frac{1}{5} \cdot MPUR(m4, \langle \text{Ag-2}, \text{Ag-1}\rangle)) = \frac{3}{5}$. Note that, compared to the word "JCB", the teaching of the word "bulldozer" gives rise to little increase in understandings rate between the pair (and therefore also in understandings rate in general). This is why "bulldozer" is overspecific, and "JCB" is not.

From an agent view

Property 6.2 and 6.3 characterize overgeneralized and overspecific words by describing how their teaching influences the interoperability measures. However, an agent cannot immediately use this characterization to find the right level of generality. Because one agent does not have access to the other agent's ontology, it cannot compute how the teaching of a word influences the understandings rate. Therefore, the agents follow the *expected increase in understandings rate*.

We use the notation $Exp(m', MPUR(m, \langle \text{Ag-i}, \text{Ag-j}\rangle))$ to refer to the expected value of $MPUR(m, \langle \text{Ag-i}, \text{Ag-j}\rangle)$, after a concept that means m' has been taught. Given that the current $MPUR(m, \langle \text{Ag-i}, \text{Ag-j}\rangle)$ is 0, the expected value after a concept with meaning m' is taught can be calculated as follows (ρ_i is the conceptualization of Ag-i, and ρ_j is the conceptualization of Ag-j)

- $Exp(m', MPUR(m, \langle \text{Ag-i}, \text{Ag-j}\rangle)) = 1$ if $m' = m$
- $Exp(m', MPUR(m, \langle \text{Ag-i}, \text{Ag-j}\rangle)) =$P($m'$ is not internal in ρ_j) if m is a descendant of m' in ρ_i
- $Exp(m', MPUR(m, \langle \text{Ag-i}, \text{Ag-j}\rangle)) = 0$ if the first two conditions do not hold.

The first condition states that if a concept is taught that exactly means m, then the agent is certain that teaching the concept enables communication of the meaning m. The second condition states that, if the concept that is taught means something more general than m, the expected *MPUR* equals the probability that the other agent does not consider the meaning m' overgeneralized. In our case this boils down to the probability that m' is not internal in ρ_j, i.e. the conceptualization of Ag-j does not contain more specific meanings than m'. The last condition states that, if m' is not equal or more general than m, a concept that means m' cannot be used to communicate m, and therefore the teaching of that concept will not increase the *MPUR* with respect to m.

The expected *PUR* (corresponding to Definition 6.8) after c is taught can be calculated by averaging over the expected *MPUR*'s:

- $Exp(m', PUR(\langle \text{Ag-i}, \text{Ag-j} \rangle)) = \frac{1}{\#V_i} \sum_{m \in V_i} Exp(m', MPUR(m, \langle \text{Ag-i}, \text{Ag-j} \rangle))$

Because the agents must base their decision which meaning to select on *expectations*, the agents cannot be certain that they find the right level of generality. Therefore, they must decide whether to attach more value to expected *MPUR*, or to expected *PUR*. This decision is set down in the parameters θ_1 and θ_2 which indicate the importance of respectively *MPUR*, and *PUR*. Using these parameters, the meaning selection strategy can be described as follows.

Definition 6.14. *Given that Ag-i intends to communicate a meaning m. The meaning selection strategy is described by:*
$argmax_{m' \in V_i}(\theta_1 \cdot Exp(m', MPUR(m, \langle Ag\text{-}i, Ag\text{-}j \rangle)) + \theta_2 \cdot Exp(m', PUR(\langle Ag\text{-}i, Ag\text{-}j \rangle))$,
where:

- θ_1 *is the importance factor for MPUR*
- θ_2 *is the importance factor for PUR*

In the next section, we will investigate the effects of different importance factors for *MPUR* and *PUR*.

6.5.3 Experiments

For our experiments, we adopt a group of 15 agents. An agent's ontology is randomly created according to $B_g = (3, 3, 3, 3, 3, 0)$ and $B = (2, 2, 2, 2, 1, 0)$, and contains 46 concepts. An experiment consists of t steps, where at each step a random speaker and hearer is selected from the group of agents, and a random concept from the speaker's ontology. We have prevented the same hearer-speaker-concept pair to be selected twice in the same experiment. The speaker communicates the concept to the hearer using a dialogue that conforms to the communication protocol depicted in Figure 6.2 and a word selection strategy that selects the most frequently used word (WSS-3). The speaker follows a meaning selection strategy that conforms to Definition 6.14. After each step, we measure the following:

1. UR: the understandings rate, calculated according to Definition 6.9.
2. Avg. Dialogue length : The average length of a dialogue of a randomly selected speaker-hearer-concept.
3. Avg. Nr. CDP : The average number of times that a concept is taught in (only) the CDP layer, in a dialogue of a randomly selected speaker-hearer-concept.
4. Avg. Nr. CEP : The average number of times that a concept is taught in the CEP layer, in a dialogue of a randomly selected speaker-hearer-concept.

In the next sections, we will describe the results of six different experiments that were performed using different meaning selection strategies. To obtain statistical significance, we have performed every experiment 10 times of which we will present the mean outcomes. For all results, the standard deviation was less than 5 percent of the mean.

Agents that know the ontology model

In the previous section, we have argued that the speaker can determine the expected MPUR after teaching a concept by using the probability that the hearer's ontology contains no subconcepts of that concept. In this section, we assume that the agents know the ontology model, i.e. they know that $B = (2, 2, 2, 2, 1, 0)$ and $B_g = (3, 3, 3, 3, 3, 0)$. With this knowledge, an agent Ag-i can compute the probability that a meaning m is considered (non-) overgeneralized by an agent Ag-j as follows:

- if $ln(m) <$ *the depth of* ρ then P(m is internal in ρ_j) = $\prod_{i=0}^{ln(m)} \frac{b_i}{b_i^g}$
- if $ln(m) =$ *the depth of* ρ then P(m is internal in ρ_j) = 0
- P(m is not internal in ρ_j) = 1 - P(m is internal in ρ_j)

In these formulae, $b_0, .., b_d$ are typical elements of vector B, and $b_0^g, .., b_d^g$ are typical elements of B_g.

For example, in our experiments, the probability that a meaning at layer number 0 is internal is $\frac{2}{3}$. The probability that a meaning at layer number 4 is internal is $\frac{2}{3} \cdot \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{3}$. The probability that a meaning at layer number 5 is internal is 0.

A common pattern of dialogues in ANEMONE is that the speaker speaks a relatively general concept c, after which the hearer requests for specification, after which the speaker applies its meaning selection strategy a second time and speaks a more specific concept d. When the speaker applies the meaning selection strategy for the second time, it can use extra knowledge to compute the probability that d is considered overgeneralized by the hearer, namely that concept c *is* considered overgeneralized. We incorporate this idea in the meaning selection strategy using a conditional probability. An agent Ag-i that knows that a meaning n is overgeneralized for the hearer Ag-j computes the probability that a (more specific) meaning m is considered overgeneralized as follows:

$P(e1|e2) = \frac{P(e1)}{P(e2)}$, where

- $e1$ is the event that m is internal in ρ_j
- $e2$ is the event that n is internal in ρ_j, where n is an ancestor of m.

This can be proven as follows. According to Bayes theorem (Rice, 1995), $P(e1|e2) = \frac{P(e2|e1) \cdot P(e1)}{P(e2)}$. Note that $P(e2|e1)$ is 1, because $e2$ is implied by $e1$. Hence, $P(e1|e2) = \frac{P(e1)}{P(e2)}$.

MSS-1: A short term strategy

MSS-1 uses the parameters $\theta_1 = 1$ and $\theta_2 = 0$. In other words the agents only take the expected *MPUR* into account in their meaning selection strategy. Because they are only interested in the expected increase in *MPUR* concerning the meaning that they *currently* want to convey, MSS-1 qualifies as a *short term strategy*. The results of applying MSS-1 for 10000 steps is shown in Figure 6.8.

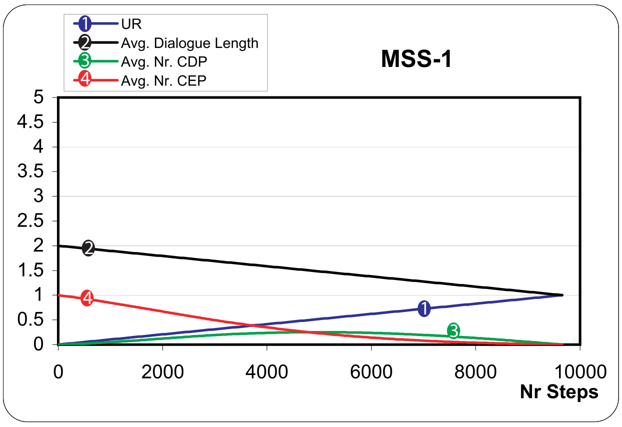


Figure 6.8: Performance of MSS-1

The situations at 0 steps can be explained as follows. Because the agents have not taught any concepts to each other, no agent understands any other agent, hence UR is 0. This means that in every dialogue, the agents have to visit the CDP or CEP layer of the protocol. Because the agents do not share any words that they can use for giving concept definitions, all teaching of new words is done using CEP (where the meaning of a word is conveyed by pointing to shared instances). Hence, Avg.Nr.CEP is 1 and Avg.Nr.CDP is 0. Because the agents visit the CEP layer every dialogue, the average dialogue length is 2.

As the number of steps increases, the agents teach concepts to each other, and the UR slowly increases. Also, the Avg.Nr.CDP increases because giving definitions becomes a viable option to teach new concepts, once a substantial amount of concepts is shared. As a result of this, there is less need for CEP, and the Avg.Nr.CEP slowly decreases. Hence, the Avg. dialogue length also decreases.

MSS-2: A long term strategy

MSS-2 uses parameters $\theta_1 = 0$ and $\theta_2 = 1$. In other words, the agents only take the expected *PUR* into account in their meaning selection strategy. Because they are interested in the expected increase in MPUR concerning any concept in their ontology, regardless whether they currently intend to convey it or not, MSS-2 qualifies as a *long term strategy*. The results of applying the long term strategy for 10000 steps is shown in Figure 6.9.

Using the long term strategy, the Avg.Nr.CEP is relatively high in the beginning. This is because the speaker may end up teaching three or four general concepts to the hearer, before it teaches the concept that is specific enough for the hearer to accept. As a result of this, the Avg. dialogue length is also relatively high. We can also observe that the strategy that aims at increasing the *PUR* indeed gives rise to a fast increase in UR. Therefore, the Avg.Nr.CEP and Avg. Dialogue length decrease

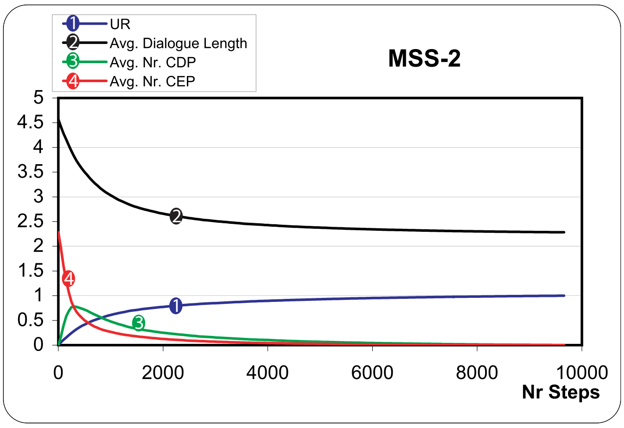


Figure 6.9: Performance of MSS-2

quickly in the beginning.

One of the reasons that MSS-2 yields a faster increase of UR than MSS-1 is that the Avg.Nr.CEP is higher in MSS-2 than in MSS-1. Another reason is that the concepts that are taught in MSS-1 are overspecific and therefore only increase UR a little (Property 6.3). To support this claim we included Figure 6.11 where the strategies MSS-1, MSS-2 (and MSS-3) are compared in a graph with the total number of CEP on the x-axis. Furthermore, this figure reveals that the total number of CEP that is required to reach an UR of 1 is around 1300 using MSS-2, and around 5000 using MSS-1. Therefore, the communication vocabulary that is produced by MSS-2 is also much smaller than the CV that is produced by MSS-1.

The following table compares the short term strategy (MSS-1) with the long term strategy (MSS-2).

	MSS-1	MSS-2
Increase in UR	-	+
Initial Avg.Nr.CEP.	+	-
Avg. Dialogue Length	+	-

With respect to a fast increase in UR, the MSS-2 performs better than MSS-1. However, the dialogues in MSS-2 are longer, and the Avg.Nr.CEP is high in the beginning. The meaning selection strategy MSS-3 aims at achieving the best of both worlds.

MSS-3: A medium term strategy

MSS-3 uses parameters $\theta_1 = 1$ and $\theta_2 = 5$, such that the agents take the expected *MPUR* and *PUR* into account. Because it combines short term goals with long term goals, this strategy is a *medium term strategy*. The results are shown in Figure 6.10.

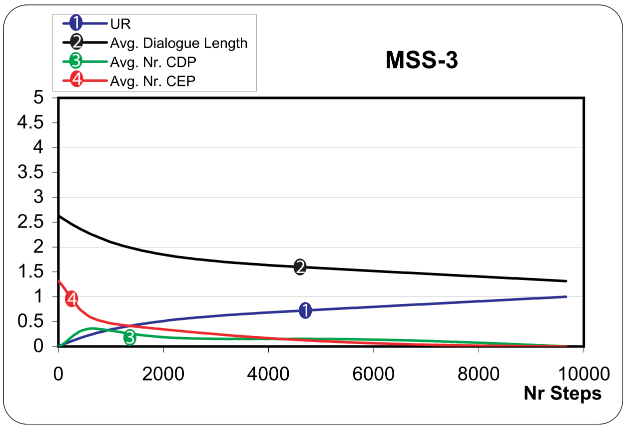


Figure 6.10: Performance of MSS-3

As this figure reveals, MSS-3 gives rise to a faster increase of UR than MSS-1, and it gives rise to shorter dialogues and initial Avg.Nr.CEP than MSS-2.

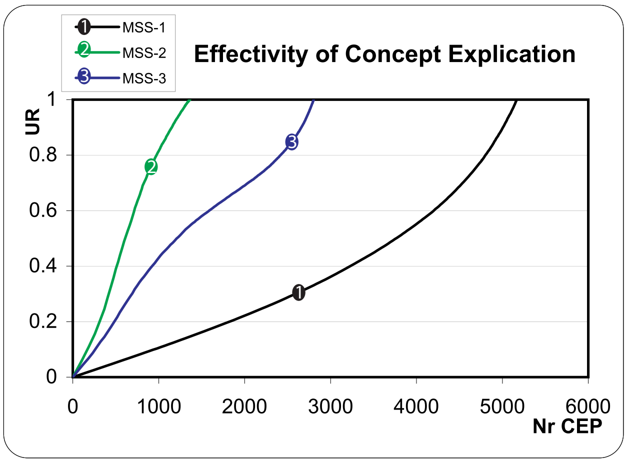


Figure 6.11: Comparison of MSS-1,MSS-2 and MSS-3

MSS-4, MSS-5, MSS-6: Learning the ontology model

The meaning selection strategies described in the previous sections build on the assumption that the agents know the ontology model. In this section, we do not make this assumption, and make the agents learn the ontology model during their conversations. This is done as follows. For every meaning in its ontology, an agent keeps track of:

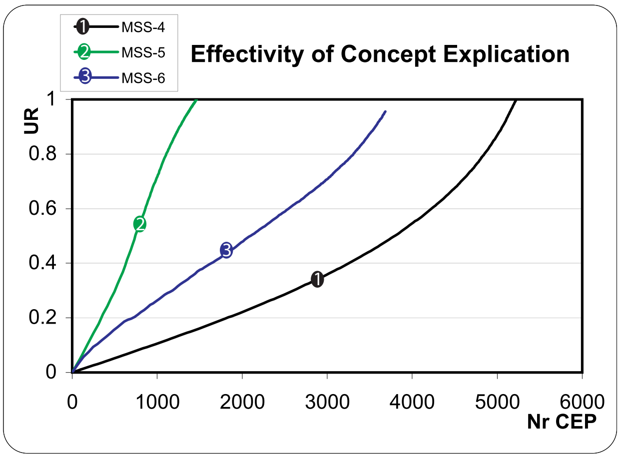


Figure 6.12: Comparison of MSS-4, MSS-5 and MSS-6

- N_1 the number of agents that regarded the meaning overgeneralized. These agents have responded "ReqSpec" to Inform-messages containing this meaning.
- N_2 the number of agents that did not regard the meaning overgeneralized. These agents have responded "OK" to inform-messages containing this meaning.

N_1 and N_2 are both initialized to 1. Using these values for meaning m, agent Ag-i can approximate the probability that m is internal in a conceptualization ρ_j as follows:

- P(m is internal in ρ_j) = $\frac{N1}{N1+N2}$

MSS-4, MSS-5 and MSS-6 adopt the same values for parameters θ_1 and θ_2 as MSS-1, MSS-2 and MSS-3 respectively. The difference is that MSS-4, MSS-5 and MSS-6 learn the ontology model as they participate in conversations. Figure 6.12 shows the performance of MSS-4, MSS-5 and MSS-6 in a similar fashion as Figure 6.11. This figure reveals that MSS-4 yields very similar results as MSS-1. This is because MSS-1 and MSS-4 incite agents to select the most specific meaning. Because the agents select the most specific meaning anyway, the results of MSS-1 and MSS-4 are the same. MSS-2 and MSS-5 incite the agents to select the most general meaning. Therefore, MSS-2 gives rise to the same results as the MSS-5. The situation is different with MSS-3 and MSS-6, which incites agents to select a meaning that is the right balance between specificity and generality. The estimation of the ontology model used in MSS-6, does influence the results, as can be seen when the results of MSS-3 are compared with MSS-6 in Figure 6.11 and 6.12. Nevertheless, MSS-6 still gives rise to a significant faster increase in UR than strategy MSS-4 and MSS-1.

6.6 Conclusion

In this chapter, we have discussed strategies for ontology negotiation. In particular, we have discussed word selection strategies for choosing the right words and meaning selection strategies for selecting the right meanings.

The experiments performed on word selection strategies demonstrate the following. Firstly, a strategy which incites the agents to adopt each other's concept names for speaking (such as WSS-2 and WSS-3) has considerable advantages over a strategy in which every agent holds on to its own concept names for speaking (such as WSS-1 or point-to-point ontology alignment). Secondly, the strategy of adopting the most frequently used concept name (WSS-3) is usable in a network with a simple interaction pattern, i.e. when everyone speaks to everyone with equal probability. Thirdly, it was demonstrated that in a scale-free network, the performance of the strategy of adopting the most frequently used concept name is improved when the agent's number of acquaintances is taken into account (as is done in strategy WSS-4).

Our investigations on meaning selection strategies have revealed that finding the right level of generality is important for ontology negotiation. We have experimentally supported this claim by comparing different communication strategies that incite the agents to convey their information at different levels of generality. An agent that conveys information using a very specific word (MSS-1), runs the risk that the other agent does not know the word. An agent that conveys information using a very general word (MSS-2), runs the risk of being too vague which would result in a lengthy dialogue. Our experiments reveal that, when the agents speak at the right level of generality, semantic interoperability of the system can be achieved fastly while the average dialogue length is kept short (as shown by MSS-3).

We have also shown that the agents can reliably assess the right level of generality themselves. They may do this by recording how many other agents do and do not consider a meaning overgeneralized. As an agent participates in conversations, it builds up a model of the other agents' ontologies, which enables it to find the right level of generality (as shown by MSS-6).

6.6.1 Future work

The model we have used for the simulation experiments in this chapter allows micro-level actions to be related to their macro-level effects. By running experiments with different communication strategies, we have obtained results that can be used for strategy analysis.

This line of research can be continued by investigating other communication strategies, such as strategies that combine the selection of words and meanings. In this way, when the optimal meaning can only be conveyed with an almost unknown word, the speaker could choose for a suboptimal meaning that can be conveyed with a generally known word.

Another topic of future research would be to apply the ontology negotiation strategies in a hybrid ontology reconciliation approach. In Section 4.5.2 we discussed the possibility to use ANEMONE in combination with standardized ontologies. In such a scenario, some famous ontologies are likely to be known by an average agent. The agent could use this knowledge in its word and meaning selection strategy. In this way, the agents do not build their estimation of word frequency and ontology model from scratch. Rather, their estimation is based on a priori knowledge of some well known standardized ontologies.

Another continuation of this work could be to make the model more complex. For example, we could study what happens if not all agents communicate with each other with equal probability. Furthermore, we could make the agents' ontologies more complex, for example by adding overlapping concept relations.

The next chapter presents a case study which puts the theory developed until now into practice.

Chapter 7

Case Study of ANEMONE

For every dollar a company spends on an application package, they spend $5 to $9 on labor and infrastructure to integrate [the application] into their existing IT infrastructure.

Nelson Mattos, director of information integration at IBM (Ryan, 2003)

This chapter demonstrates an application of the techniques discussed in the previous chapters[1]. We will focus on a case with heterogeneous RSS news feeds on the internet. The assumptions underlying ANEMONE discussed in Chapter 4 are valid in this domain. The ontologies that are involved are real news taxonomies as currently encountered on the internet. The structure of these ontologies corresponds to the ontologies discussed in Chapter 4. Thus, there is no ontology language involved. The instances are real news articles as published at the time of the case study. The concept classifiers are implemented by text classifiers. We used agents as wrappers around news feed and allowed them to communicate about their news articles. They reconciled their ontologies using ANEMONE. Although the study regards a single case, we believe that the findings are characteristic for open multi-agent systems in general.

The chapter is organized as follows. Section 7.1 discusses the domain of RSS news feeds in more depth. Section 7.2 describes the implementation details. Section 7.3 presents the results. We conclude in Section 7.4.

7.1 Domain

Over the last few years, RSS (Rich Site Summary or Really Simple Syndication) has become a popular format for the syndication of news content on the internet

[1]The prototype presented in this chapter is freely available on the author's home page.

```
<item>
   <title>Scientists urge evolution lessons</title>
   <description>The world's leading scientists call on parents and teachers
       to give children evidence-based facts on evolution.</description>
   <link>http://news.bbc.co.uk/go/rss/-/1/hi/sci/tech/5098608.stm</link>
   <guid isPermaLink="false">http://news.bbc.co.uk/1/hi/sci/tech/
       5098608.stm</guid>
   <pubDate>Wed, 21 Jun 2006 21:46:44 GMT</pubDate>
   <category>Science/Nature</category>
</item>
```

Figure 7.1: An RSS news item

(Powers, 2005). Using RSS, a news provider distributes its web content as an XML file, called an RSS feed. The advantage of using RSS is that it enables the provider to add machine-readable metadata to its news content. This metadata can be used to identify different parts of the article, such as the title, a summary, a link to the full article, and the date of publication. An example of a news item in RSS 2.0 is shown in Figure 7.1. The metadata in an RSS file are given by XML-tags, i.e. opening tags of the form `<tag-name>` and closing tags of the form `</tag-name>`. Between an opening and a closing tag, the text of the specific kind is included. In an RSS file, the tags such as *title*, *description*, *link* are nested in the tag *item*. An RSS file usually contains multiple items, containing different news articles belonging to the same topic. The overall feed also contains information about the name of the feed. For example, the news item shown in Figure 7.1 stems from an RSS feed by the BBC, called "Science/Nature".

By representing content in RSS, other internet sites or applications may use this information within their own environments. Whereas we have introduced RSS as a way to syndicate news articles, it is also widely used for web-logs, sharing comments and for educational purposes (Harrsch, 2003). It is therefore one of the first examples of machine-readable content on the web that has made its way into practice.

The major success of RSS can be attributed to the fact that it enables an *average* person to use this technology to provide and to consume content from the internet in an efficient manner. An example of an application that facilitates consuming news content from the internet is an *RSS aggregator*.

7.1.1 RSS Aggregators

An RSS aggregator is a useful tool for people that wish to stay updated on a number of favorite web sites. After the user has selected a number of RSS feeds, the RSS aggregator constantly scans the content of these feeds for updates, and presents any new items to the user. This enables the user to see his or her latest favorite news items at one glance without having to repeatedly visit several internet sites.

Usually, a news provider provides several news feeds on different categories. A selection of the various news feeds that are provided by four major news providers BBC[2], Moreover[3], Reuters[4] and Yahoo[5], is given below.

BBC	**MoreOver**	**Reuters**	**Yahoo**
News Front Page	Tennis	Top News	U.S. National
World	Boxing	Business News	Sports News
UK	Basketball	Science	NBA
Business	Law news	Bird Flu	Science News
Politics	McDonalds news	International	Asian Tsunami Disaster
Science/Nature	Jokes	Politics	Iraq
around 15 others...	*around 330 others...*	*around 10 others...*	*around 120 others...*

As appears from this overview, different news providers have adopted very different ways to classify their news articles. A proliferation of classification schemes which is typical for today's internet is also characteristic of RSS news feeds. Comparing the names *Science/Nature* by BBC, *Science* by Reuters and *Science News* by Yahoo, we observe that any naming conventions are absent. Furthermore, the news providers classify their news at differing levels of granularity. For example, Moreover subdivides sports in 20 different topics, whereas Reuters uses only one news feed to cover sports.

For the user of an RSS aggregator, this heterogeneity of classification schemes has the following consequences. Firstly, a user that is interested in, for example, science, should manually discover and sign up for the different science related news feeds, e.g. the *Science/Nature* feed at BBC, the *Science* feed at Reuters and *Science News* feed at Yahoo. Considering the amount of news feeds available, this may be a very laborious task. Secondly, after the user has discovered and selected the desired news feeds, the news feeds covering the same topic are represented in a poorly organized list. No integrated view is provided that combines news feeds belonging to the same topic. For example, we would like one topic header "Science" in which all science related articles stemming from any news feed are collected.

7.1.2 Personal News Agents

For our case study, we propose a "next generation" RSS aggregator: a personal news agent (PNA). The purpose of this application remains presenting news articles to the user in a well organized way. However, compared to an RSS aggregator, a personal news agent provides some additional functionality:

- The PNA automatically discovers new news sources.
- The PNA does not simply *aggregate* RSS news feeds but also *integrates* them.

[2]http://news.bbc.co.uk/1/hi/help/3223484.stm
[3]http://w.moreover.com/site/other/ categories_rss.html?
[4]http://today.reuters.com/rss/newsrss.aspx
[5]http://news.yahoo.com/rss

To realize these goals, we assume a multi-agent system consisting of news agents that represent different RSS news providers. The categorizations of topics used by the news providers are the agents' ontologies. A user signs up to one of the news agents which then becomes its personal news agent. The PNA not only acquires news articles by downloading them from its own RSS news source, but also communicates with other agents to obtain news articles from other sources. For example, a PNA that represents Reuters communicates with a news agent representing Yahoo to obtain articles belonging to the *Science News* feed. Thus, a PNA can be said to discover new news sources by communicating with other news agents in the system.

When a PNA acquires a news article from another news agent, it arranges this news article according to its own list of topics. For example, when Reuters' agent acquires a news article stemming from Yahoo's *Science News* feed, it does not create another topic *Science News*, but places the article under the already existing topic *Science*. In this way, the list of news topics provided to the user remains conveniently arranged. As a result, the PNA can be said to integrate news feeds, instead of simply aggregating them.

By using agents to represent news providers, a multi-agent system is obtained that is open, dynamic, and heterogeneous. The system is open because news providers may start and finish their services at any time. The system is dynamic as the agents may change from time to time, e.g. Yahoo's topic "Asian Tsunami Disaster" is clearly a temporary topic. The heterogeneous lists of news topics used by the news providers give rise to heterogeneous ontologies in the system.

To enable the news agents to communicate, some semantic integration problems crop up which are typical for open heterogeneous MAS's. The proliferation of heterogeneous ontologies in the system is inherent to the way the system is built up. Standardization efforts in this domain are almost doomed to fail, because the news providers deliberately distinguish themselves from others by using different ontologies. Invoking Wordnet, a commonly proposed technique for ontology mapping, also turns out to have its limitations in this domain. Firstly, this technique is incomplete, e.g. Yahoo's concept "NBA" is not defined in Wordnet. Secondly, the technique may be incorrect. For example, the "International" concept of the American company Reuters is really a superset of the "UK" concept of the (British) BBC. From the perspective of the BBC agent, Wordnet would suggest differently.

In the following section, we will discuss how Anemone can be applied to solve the communication problems in the system.

7.2 Implementation

We have implemented the system as a multi-threaded Delphi[6] application. A single news agent corresponds to an RSS aggregator. The main cycle of a news agent is depicted in Figure 7.2. We will discuss each step in the cycle below.

[6]http://www.borland.com/us/products/delphi/index.html

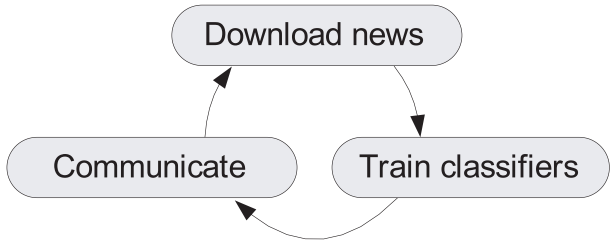


Figure 7.2: Main cycle of a news agent

7.2.1 Download News

A news agent periodically checks the internet if any new news items have arrived on one of the RSS feeds it has been subscribed to. If any new item has arrived, it downloads it and obtains a text of the kind depicted in Figure 7.1. It then uses an XML parser to extract the fields: title, description, link, and pubDate. It stores this information in a local database which constitutes the assertional knowledge base of the agent. This database can be consulted by the person who uses the PNA to stay updated on the latest news articles. Basically, this component allows for the same functionality as an RSS aggregator.

7.2.2 Train Classifiers

To obtain the native concept classifiers, the agent trains text classifiers using the items that it has downloaded from its news provider. As training examples, we use the description field of a news item. Because the agent knows to which topic these descriptions belong, we can apply a supervised learning algorithm. Before the text in the description is used as training data for the classification algorithm, some preprocessing is performed.

Preprocessing

Whether a machine learning algorithm is successful in a particular domain is often more dependent on the selection of relevant features, than on the algorithm itself. The goal of preprocessing is to eliminate all irrelevant features from the input data to make it better digestible for the machine learning algorithm.

The techniques that can be used for preprocessing are highly domain dependent. For the news descriptions, we have applied some preprocessing techniques that are common in natural language processing (Jackson and Moulinier, 2002). Firstly, all characters other than letters are removed and the string is placed in lower case. Secondly, all common words that have little or no meaning by themselves, such as "on" and "and", (called stopwords) are removed from the input string. Then, all suffixes are removed from the word by applying the Porter stemming algorithm

(Porter, 1980). Finally, most drastically of all, the input string is transformed to a term frequency vector (TFV). This builds on the assumption that only the frequency of words occurring in a document is important, regardless of word order. To transform the input samples to term frequency vectors, a word list (WL) must be built up, consisting of all words occurring in any of the input samples. The TFV indicates, in the same order, the number of times that each word occurs in the particular input sample.

The example below shows the preprocessing that is performed on the news item of Figure 7.1. As appears, the input string consists of many characters that take many values. After the preprocessing steps are performed, the string is simplified in length and number of possible values.

1. Original
   "*The world's leading scientists call on parents and teachers to give children evidence-based facts on evolution.*"
2. Character selection and lower case transformation
   "*the world s leading scientists call on parents and teachers to give children evidence based facts on evolution*"
3. Stopword removal
   "*world s leading scientists call parents teachers give children evidence based facts evolution*"
4. Stemming
   "*world s lead scientist call parent teacher give children evid base fact evolut*"
5. Transformation to term frequency vector:
   WL:⟨*base,call,children,evid,evolut,fact,give,lead,parent,s,scientist,teacher,world*⟩
   TFV:⟨1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1⟩

Classification

After the description fields are preprocessed, they can be used as training data for the classification algorithm. To train a classifier to recognize instances of a certain class, it must be provided with positive and negative examples of the class. Positive examples are the (preprocessed) descriptions of the news items belonging to the topic. Negative examples are the (preprocessed) descriptions of the news items of other topics. For example, to train the classifier for BBC's *Science/Nature* concept, the positive examples are taken from *Science/Nature* and the negative examples from *business* and *UK*.

Given that the training examples are given as vectors of size n, the examples can be viewed as points in an n-dimensional space. The learning task of the classifier is to find a hypersurface (or decision boundary) that separates the positive from the negative examples. If the hypersurface is a hyperplane, we say that the classification is *linear*. Different machine learning techniques can be used to learn the classification, such as neural networks (Mitchell, 1997) and support vector machines

(Vapnik, 1995). An in-depth discussion of these algorithms is beyond the scope of this thesis.

In our implementation, we have used a support vector machine (SVM) with a linear kernel function, i.e. corresponding to a linear classification. The motivation of using an SVM is that this algorithm has been shown to yield good results with limited amounts of training samples (Joachims, 1999b). This is a relevant issue in the news agent case as most RSS news providers only publish around four news articles per topic per day. If neural networks were used to learn the concept classifiers, it would probably take too long before the agents could participate in ontology negotiation. We have adopted the implementation SVM^{light} which has been made publicly available by Joachims (1999a).

7.2.3 Communication

Communication between the agents proceeds according to the communication mechanisms described in Chapter 4. The NCP and CDP layers can be readily implemented in the news agent system. We will focus on the implementation of the CEP layer in the communication protocol, where agents learn concepts from other agents.

Agents are capable of participating in CEP after the agents have sufficiently trained their concept classifiers on the news articles they have downloaded themselves. In CEP, the teacher sends descriptions of news articles belonging to the concept to be explicated as positive examples, and descriptions of news articles belonging to the concepts that are disjoint with the concept to be explicated as negative examples. The hearer transforms the descriptions to term frequency vectors as described in the previous section. For every concept, it applies the corresponding support vector machine to classify these examples. It computes TPR and TNR values according to the rules described in Section 4.3.3 and derives the concept relations.

Because in our application, a small amount of misclassifications is acceptable, we used relatively tolerant criteria to assess concept relations, i.e. $\theta_1 = 0.75$, $\theta_2 = 0.75$, $\theta_3 = 0.4$.

7.3 Results

We demonstrate the system using the four relatively simple agents depicted in Figure 7.3. The agents represent news publishers BBC, Moreover, Reuters and Yahoo. The agents' ontologies consist of subsets of the news feeds provided by the news publishers. After the agents have collected news articles for a period of two months, their knowledge bases were filled with approximately 200 news articles per topic. This enabled them to train their classifiers and participate in ontology negotiation. The following examples present a series of successive dialogues which were generated by the agents. For demonstration purposes, the agents make some of their internal workings public; these are preceded by a *. Due to space limitations, we abbreviated long path names of news articles with `[]`. The agents begin in a

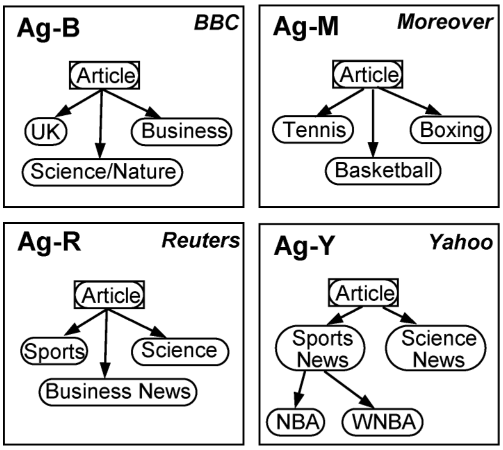


Figure 7.3: Initial situation

situation where they do not share any concept with each other (Figure 7.3).

Ag-Y intends to inform Ag-R about Science News.

1 Ag-Y:<Ag-R,<ExactInform,Science News,[...]>>
2 Ag-R:<Ag-Y,<StartCDP>>
3 Ag-Y:<Ag-R,<ProposeStartCEP>>
4 Ag-R:<Ag-Y,<StartCEP>>
5 Ag-Y:<Ag-R,<Explicate,Science News,[...],[...]>>
* Ag-R Business News: TPR=0.15 TNR=0.86
* Ag-R Science: TPR=0.76 TNR=1
* Ag-R Sports: TPR=0 TNR=0.24
* Ag-R derives equivalent(Science News,Science)
6 Ag-R:<Ag-Y,<ExitCDP>>
7 Ag-Y:<Ag-R,<ExactInform,Science News,[...]>>
* Ag-R translates to Science
8 Ag-R:<Ag-Y,<OK>>

Ag-R intends to inform Ag-Y about Science

9 Ag-R:<Ag-Y,<ExactInform,Science News,[...]>>
10 Ag-Y:<Ag-R,<OK>>

Ag-Y intends to inform Ag-M about Science News

11 Ag-Y:<Ag-M,<ExactInform,Science News,[...]>>
12 Ag-M:<Ag-Y,<StartCDP>>
13 Ag-Y:<Ag-M,<ProposeStartCEP>>
14 Ag-M:<Ag-Y,<StartCEP>>
15 Ag-Y:<Ag-M,<Explicate,Science News,[...],[...]>>
* Ag-M Basketball: TPR=0.29 TNR=0.32
* Ag-M Tennis: TPR=0.21 TNR=0.84

```
* Ag-M Boxing: TPR=0.26 TNR=1
* Ag-M derives disjoint(Science News,Basketball;Tennis;Boxing)
16 Ag-M:<Ag-Y,<ExitCDP>>
17 Ag-Y:<Ag-M,<ExactInform,Science News,[...]>>
* Ag-M translates to Article
18 Ag-M:<Ag-Y,<OK>>
```

Ag-R intends to inform Ag-M about Science

```
19 Ag-R:<Ag-M,<ExactInform,Science News,[...]>>
* Ag-R translates to article
20 Ag-M:<Ag-R,<OK>>
```

Ag-R intends to inform Ag-Y about Sports

```
21 Ag-R:<Ag-Y,<ExactInform,Sports,[...]>>
22 Ag-Y:<Ag-R,<StartCDP>>
23 Ag-R:<Ag-Y,<Define Sports,<disjoint Science News>>>
24 Ag-Y:<Ag-R,<StartCEP>>
25 Ag-R:<Ag-Y,<Explicate,Sports,[...],[...]>>
* Ag-Y NBA: TPR=0.17 TNR=0.94
* Ag-Y WNBA: TPR=0.13 TNR=0.99
* Ag-Y Science News: TPR=0.08 TNR=0.22
* Ag-Y Sports News: TPR=0.90 TNR=0.81
* Ag-Y derives equivalent(Sports,Sports News)
26 Ag-Y:<Ag-R,<ExitCDP>>
27 Ag-R:<Ag-Y,<ExactInform,Sports,[...]>>
* Ag-Y translates to Sports News
28 Ag-Y:<Ag-R,<OK>>
```

Ag-Y intends to inform Ag-R about NBA

```
29 Ag-Y:<Ag-R,<Inform,Sports,[...]>>
30 Ag-R:<Ag-Y,<OK>>
```

Ag-R intends to inform Ag-Y about Business News

```
31 Ag-R:<Ag-Y,<ExactInform,Business News,[...]>>
32 Ag-Y:<Ag-R,<StartCDP>>
33 Ag-R:<Ag-Y,<Define,Business News,<disjoint Sports,disjoint Science News>>>
34 Ag-Y:<Ag-R,<ExitCDP>>
35 Ag-R:<Ag-Y,<ExactInform,Business News,[...]>>
* Ag-Y translates to Article
36 Ag-Y:<Ag-R,<OK>>
```

Ag-Y intends to inform Ag-M about Sports News

```
37 Ag-Y:<Ag-M,<ExactInform,Sports News,[...]>>
38 Ag-M:<Ag-Y,<StartCDP>>
39 Ag-Y:<Ag-M,<ProposeStartCEP>>
40 Ag-M:<Ag-Y,<StartCEP>>
```

```
41 Ag-Y:<Ag-M,<Explicate,Sports News,[...],[...]>>
* Ag-M Basketball: TPR=0.47 TNR=0.70
* Ag-M Tennis: TPR=0.27 TNR=0.78
* Ag-M Boxing: TPR=0.03 TNR=0.73
* Ag-M derives overlaps(Sports News,Basketball)
* Ag-M derives disjoint(Sports News,Tennis;Boxing)
42 Ag-Y:<Ag-M,<ExitCDP>>
43 Ag-M:<Ag-Y,<ExactInform,Sports News,[...]>>
* Ag-Y translates to Article
44 Ag-Y:<Ag-M,<OK>>
```

Ag-Y intends to inform Ag-M about NBA

```
45 Ag-Y:<Ag-M,<Inform,Sports,[...]>>
*  Ag-M translates to Article
46 Ag-M:<Ag-Y,<ReqSpec>>
47 Ag-Y:<Ag-M,<ExactInform,NBA>>
48 Ag-M:<Ag-Y,<StartCDP>>
49 Ag-Y:<Ag-M,<Define NBA,<subset Sports News>>>
50 Ag-M:<Ag-Y,<StartCEP>>
51 Ag-Y:<Ag-M,<Explicate,NBA,[...],[...]>>
* Ag-M Basketball: TPR=1 TNR=0.51
* Ag-M Tennis: TPR=0 TNR=0.83
* Ag-M Boxing: TPR=0 TNR=0.85
* Ag-M derives subset(NBA,Basketball)
* Ag-M derives disjoint(NBA,Tennis; Boxing)
52 Ag-M:<Ag-Y,<ExitCDP>>
53 Ag-Y:<Ag-M,<ExactInform,NBA,[...]>>
* Ag-M translates to Basketball
54 Ag-M:<Ag-Y,<OK>>
```

After this conversation, the agents have built up knowledge about each other's ontologies, as shown in Figure 7.4. The agents have built up a shared ontology that enabled them to convey what they intended to convey (effectiveness), but have not made the shared ontology larger than required (minimality). The acquired concepts are not only mapped to equivalent native concepts but also to native superconcepts (e.g. Ag-M's *NBA* and *Basketball*) and to disjoint native concepts (e.g. Ag-M's *Science News*). Approaches that only deal with equivalence mappings would have failed to solve the semantic integration problems of the news agents.

The conversation shows that many of the ontology negotiation techniques that are incorporated in Anemone are useful in a realistic domain. We mention the following:

- the use of mappings other than equivalence mappings (11-18, 45-54)
- the use of generalization in message composition (29-30)
- detecting information loss (45-46)

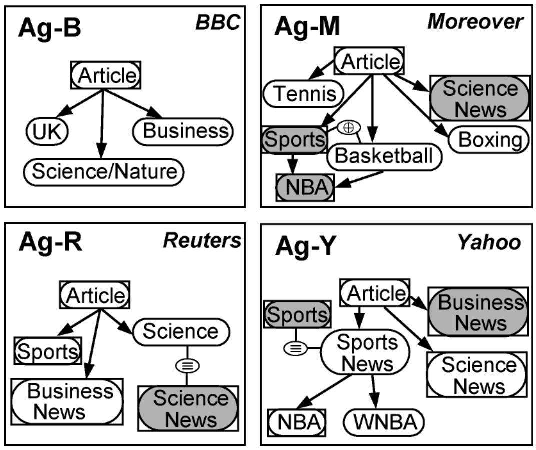


Figure 7.4: Situation after 9 conversations

- the use of speaking in *unknowingly shared* concepts (19)
- using acquired concepts for communication in *both* directions (9,29)
- the use of communicating concept definitions (31-36)
- detecting inadequate concept definitions (23-24)

In this case, it took around 10 seconds (on a Pentium 4, 2.5 GHz) for the agents to finish concept explication. Although the CEP layer worked well enough for this relatively simple case, in a scenario with more complex ontologies or where the correctness of concept mappings is of critical importance, the CEP layer needs improvements. Most likely, these improved versions will be even more time consuming. This strengthens our motivation for lazy ontology negotiation to reduce the occurrence of concept explication to a minimum.

7.4 Conclusion

In this chapter, we have applied the ontology negotiation protocols discussed in this thesis to the domain of RSS news feeds. This domain contains many features we believe to be typical for open heterogeneous MAS's. Because different news taxonomies are developed by different news providers, heterogeneous ontologies are inherent to the domain. It therefore provides a realistic environment to test the ideas introduced in the preceding chapters.

By implementing news agents and applying them on the RSS news feed domain, we have shown that agents are indeed capable of overcoming ontology problems using ANEMONE. Furthermore, we have shown that the design objectives we have pursued in this thesis are sensible in a real world domain. By participating in lazy ontology negotiation, the agents postpone the laborious task of concept explication until it is known to be needed. This feature turned out to be very useful in this domain. Furthermore, because our ontology reconciliation technique is distributed over all agents, no centralized components are needed to resolve the agents' misunderstandings.

Chapter 8

Conclusions

"When I use a word," Humpty Dumpty said, in a rather scornful tone, "it means just what I choose it to mean, neither more nor less." "The question is," said Alice, "whether you can make words mean so many different things." "The question is," said Humpty Dumpty, "which is to be master - that's all."

Through the Looking Glass (Lewis Carroll, 1871)

This thesis is about achieving semantic interoperability using a dialogue-based approach. Neither a standardized ontology, nor an ontology agent prescribes what the meaning of a concept is. Instead, the agents come to agreements among themselves.

This chapter concludes the thesis. Section 8.1 discusses the results of our research in the light of the research aim stated in Chapter 1. Section 8.2 suggests possible directions for future work.

8.1 Results

As stated in the introductory chapter, our research objective was to develop computationally attractive dialogue mechanisms for achieving ontology-based semantic interoperability in multi-agent systems. Our investigations have covered three aspects. We have translated the research objectives into more concrete design objectives, we have proposed dialogue mechanisms that satisfy these design objectives and we have evaluated the dialogue mechanisms. The main findings of each of these aspects are summarized below.

Design objectives

Achieving ontology-based semantic interoperability boils down to establishing *shared* ontologies between agents. Our approach is based on dialogues in which the participants teach concepts to each other to make parts of their ontologies shared. Concept teaching requires a profound knowledge of what a concept means. This must be realized by a *grounded knowledge representation*. To make the approach computationally attractive, the shared ontology built between agents should be *minimal*. This reduces storage load and processing time. Furthermore, the shared ontology should be *effective*, meaning that it allows *sound* and *lossless* information exchange between the agents. A third measure is to make the dialogue protocol *lazy*. This means that agents only apply computational expensive methods to exchange ontological information when strictly necessary.

Dialogue mechanisms

Our approach to lazy ontology negotiation, called ANEMONE, is built on the observation that not all information possessed by the speaker should necessarily be transferred to the hearer to establish effective communication. This enables the speaker to express itself in more general terms when no other options are available. When the speaker violates the *effectiveness* requirement, the hearer will recognize this and respond by requesting more specificity. In turn, the speaker will respond with a more specific message. In this way, information is conveyed in a *dialogue* instead of in a one-shot fashion. When effective information exchange using the current shared ontology turns out to be impossible, the agents attribute this problem to a deficient shared ontology.

After the agents have *detected* deficiencies in their shared ontology, they *resolve* the problem by exchanging parts of their ontologies. The agents use two methods for this. Firstly, they exchange a concept definition, which is computationally cheap. When this method fails, they explicate the concept by showing examples. In doing so, the grounded knowledge representations prove themselves useful, since agents use their concept classifiers both for generating examples, as well as for interpreting them. Because this is computationally relatively expensive, they will only do this when strictly necessary, being lazy.

Two agents participating in ontology negotiation not only set up a minimal shared ontology for themselves, but also for their entire community. Therefore, they adapt their word choice to what is most common to the other agents. Furthermore, they speak at a level of generality which they expect to be neither too specific, nor too general.

Evaluation

We have evaluated the dialogue mechanisms in three ways: by a formal analysis, by simulation experiments and by performing a case study.

The formal analysis enabled us to give solid proofs of the properties of the proposed communication protocols. In particular, we have proven that, from a subjective viewpoint, no information is lost in communication. Furthermore, two agents are capable of establishing a minimal communication vocabulary by removing redundant concepts.

Our simulation experiments have shown that our dialogue-based approach can be used to achieve semantic interoperability in (relatively) large groups of agents. In particular, we have demonstrated that the agents succeed in minimizing the number of synonyms in their communication vocabulary. Furthermore, they minimize the number of meanings expressible by the communication vocabulary by speaking at the right level of generality.

Our case study has shown that the dialogue mechanisms we propose are successfully applicable to real world scenarios. We have demonstrated how grounded semantics can be realized on the internet in the domain of RSS news feeds. Furthermore, we have shown that the dialogue mechanisms of ANEMONE are useful in this domain to establish lazy ontology negotiation.

Our evaluation has confirmed that, in accordance with our research aim, we have succeeded in properly embedding a technique for concept explication in an agent communication protocol. The dialogue mechanism is computationally attractive because the agents set up a minimal shared ontology in a lazy way.

However, the technique for concept explication itself, which is not based on dialogues, remains a point of concern. As our case study reveals, the technique is applicable in a non-critical domain where ontological concepts denote concrete entities. In a domain where erroneous concept mappings have disastrous consequences, or where the agents' ontologies contain abstract concepts, our current implementation of concept explication requires improvement.

8.2 Future work

In the individual chapters, we have occasionally proposed directions for future work in the line of the material discussed. In Section 8.2.2, we will capture the essence of these proposals, which is to expand the scope of applications of ANEMONE . First, however, we will sketch a more general area of future research which concerns agents on the semantic web.

8.2.1 Agents and the Semantic Web

The semantic web is a project that aims at representing the content on the internet in a form that can be more easily processed by machines (Antoniou and van Harmelen, 2004). Several technologies have resulted from this endeavor. Most essentially, markup languages have been developed to supplement web pages with machine-readable content. As with a multi-agent system, the semantic web is a distributed system for which semantic interoperability is of crucial importance. For this reason, ontology languages are also of great interest to the semantic web community. This has resulted in a standardized ontology language, called OWL (McGuinness and van Harmelen, 2004). Another fruit of the semantic web project is OWL-S (Martin et al., 2004), which is an ontology enabling advertisement of a web service, description of its operation, and description of interoperability details.

The combination of agents and the semantic web has been put forward most prominently by Hendler (2001). Basically, he envisions the semantic web as an *environment* in which agents seek information and perform actions on behalf of their users. Thus agent technology and semantic web technology are viewed as *supplementary*. The idea we propose here aims at *fusing* the ideas from the different areas, allowing some sort of cross-fertilization to occur. Semantic web research has provided solid results in ontology-based semantic interoperability, whereas agent research has mainly focused on communication mechanisms between distributed entities. We will first describe how agent research may benefit from semantic web research and then how semantic web research may profit from agent research.

A straightforward application of semantic web technology in agent research is the use of OWL for the agents' ontologies. This may be a welcome contribution, as a standardized ontology language for agent communication is currently not available. Furthermore, a technology such as OWL-S may be useful for the agent community as the problem it addresses exists just as well in multi-agent systems as on the semantic web.

The semantic web community is well aware that heterogeneous ontologies form a serious obstacle for the project. Some of the related work we have discussed in this thesis actually stems from this community. For our ontology reconciliation technique to be applicable on the semantic web, agents must be wrapped around web content, as described in Chapter 7. In this way, semantic web content is placed *within* the agents instead of in their environment, as most commonly proposed. For ontology reconciliation, this has a number of benefits which are discussed below.

As argued in Chapter 3, ungrounded knowledge representations form an unsatisfactory basis for ontology reconciliation. The internet provides many opportunities for grounded knowledge representations (see Example 3.4). Nevertheless, the semantic web, as currently envisioned, does not take advantage of these opportunities. As illustrated by the famous semantic web "layer cake" (Berners-Lee, 1999), the semantic web is founded on XML, i.e. on syntax. This causes the symbol grounding problem on the semantic web, namely how semantics may arise from a purely syntactic basis. By adding agents at the basis of the semantic web, the framework would no longer be built on syntax alone, but also on an *active* component. This would open a way to solve the symbol grounding problem, which is required for successful ontology reconciliation. In the context of our work, it would enable the ontologies to be supplemented with concept classifiers.

Another advantage of embedding semantic web content within an agent is that ontology problems can be solved using a dialogue-based approach, such as ANEMONE. As motivated in this thesis, ontology reconciliation problems become less severe when the different components exchange only the necessary pieces of information. This can only be realized when the components participate in dialogues to explore the information needs and supplies of each other. Dialogue-based information exchange is difficult to realize on the semantic web, which mainly consists of *passive* web pages. By using agents to represent web content, this problem can be overcome.

8.2.2 Towards a Wider Scope of Applicability

The work described in this thesis has proven useful in the domain of RSS news feeds on the internet. To expand the range of applications, additional research is needed. We will describe some topics of future work below.

A first way to widen the scope of applicability of ANEMONE is to use a more expressive communication language, i.e. to relax Assumption 4.6 (page 67). In this thesis, we have focused on an elementary interaction between agents, namely the exchange of inform messages. Applying our approach to more complex communication protocols requires similar adjustments to those carried through in the inform protocol. One of these adjustments is adding support for approximate information exchange, as not every meaning expressible in the ontology of one agent is also expressible in the ontology of another agent. Consequently, the agents must be able to recognize when too much information is lost, which must also be provided by the communication protocol. However, information loss in an inform message may be different from information loss in a query message, or information loss in a request message, as discussed in Section 4.6.1. This forms an interesting topic of future research.

Another way to make ANEMONE more widely applicable is to extend the ontology language with additional constructs, such as roles. This would amount to a relaxation of Assumption 4.5. In Section 5.4.1, we have argued that non-compositional concept explication can be used to avoid the explication of roles. However, this method requires each combination of roles and concepts to be explicated, which can be very resource-consuming. Furthermore, it does not enable agents to exchange role assertions. A natural way to implement role explication would be by pointing to pairs of individuals. A useful addition to the ontology language would then be role hierarchies (Baader et al., 2003) as these would enable approximate message composition and interpretation with respect to roles. It might seem that roles could be treated completely analogously to concepts. However, the introduction of roles in the ontology language raises additional difficulties. Firstly, instances of relations are more likely to be abstract or imperceptible than instances of concepts. This makes it difficult to implement role classifiers. The problem is comparable to the problem that would arise when Assumption 4.4 would be dropped. For example, upon pointing to an instance of *person* and an instance of *book*, how does the other agent know if this pair classifies as the role *owns*, *is-author-of*, *has-read*, *lends*, *hates*, *loves*, *sells* or *publishes*? A second problem of introducing roles in the ontology language concerns the issue of reification, as discussed in Section 4.1.3. The multiformity of ways in which knowledge can be represented increases when the ontology language contains roles. In particular, a concept in one ontology may be reified (represented as an individual) in another ontology. Dealing with such ontological differences is difficult as the domain of discourse is not the same for every agent, i.e. it conflicts with Assumption 4.1.1.

It is complicated to relax the assumption that every agent uses the same domain of discourse, as some form of common basis is required for the agents to perform automatic ontology mapping. Nevertheless, we could assume that only a part of the domain of discourse is shared. This part would enable the agent to perform concept explication, while the other part would be up to its individual choice. An interesting issue of future work would be to study which parts of the domain of discourse are better suited for standardization than others. Furthermore, more complex ontology mappings are needed to reconcile ontological differences when the domains of discourse are heterogeneous. The ontology mappings proposed by Borgida and Serafini (2003) provide an interesting starting point. More research is needed to incorporate such mappings in Anemone and to automatically establish them.

Finally, Anemone can be made more widely applicable by relaxing Assumption 4.4. This is probably the most difficult task to accomplish as it complicates automatic ontology mapping considerably. In Example 3.6 we have used the concept *Accrual Periodicity* as an example of an abstract meaning. Mappings between such abstract concepts are very difficult to establish automatically. In the near future, progress on such issues can be made by incorporating manual mapping techniques in Anemone, for example by using ontology agents that obtain their mappings from humans. In the long term, progress in artificial intelligence might make a more fundamental solution possible. Such a solution would involve a knowledge representation that deeply captures the semantics of abstract meanings. However, as argued by Dou, McDermott and Qi (2004), a final solution to the automatic ontology mapping problem will probably only be available by the time the whole problem of artificial intelligence is solved.

Bibliography

(2000). *The American Heritage Dictionary of the English Language*. Houghton Mifflin Company.

Antoniou, G. and van Harmelen, F. (2004). *A Semantic Web Primer*. The MIT Press.

Aristotle (384-322 BC). *Categories*. eBooks@Adelaide, The University of Adelaide Library, Adelaide. Translated by E. M. Edghill.
URL `http://etext.library.adelaide.edu.au/a/aristotle/categories/`

Aschoff, F.-R., Schmalhofer, F. and van Elst, L. (2004). Knowledge mediation: A procedure for the cooperative construction of domain ontologies. In A. Abecker, L. van Elst and V. Dignum, editors, *Proceedings of the ECAI-2004 Workshop on Agent-mediated Knowledge Management (AMKM-2004)*, pages 29–38. Valentia, Spain.

Austin, J. (1962). *How To Do Things With Words*. Harvard University Press.

Baader, F., Calvanese, D., McGuiness, D., Nardi, D. and Patel-Schneider, P., editors (2003). *The description logic handbook: Theory, implementation and applications*. Cambridge University Press.

Baader, F., Horrocks, I. and Sattler, U. (2004). Description logics. In S. Staab and R. Studer, editors, *Handbook on ontologies*, chapter 1, pages 3–28. Springer-Verlag.

Baader, F. and Nutt, W. (2003). Basic description logics. In F. Baader, D. Calvanese, D. McGuiness, D. Nardi and P. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementations and Applications*, chapter 2, pages 43–95. Cambridge University Press.

Bach, K. (2006). Speech acts and pragmatics. In M. Devitt and R. Hanley, editors, *The Blackwell Guide to the Philosophy of Language*. Blackwell Publishing.

Bailin, S. and Truszkowski, W. (2002). Ontology negotiation between intelligent information agents. *Knowledge Engineering Review*, volume 17(1):pages 7–19.

Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, volume 286:509-512.

Barr, D. J. (2004). Establishing conventional communication systems: Is common knowledge necessary? *Cognitive Science*, volume 28:pages 937–962.

Batini, C., Lenzerini, M. and Navathe, S. B. (1986). A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, volume 18(4):pages 323–364.

Benjamins, R. and Fensel, D. (1998). Community is knowledge! in KA2. In *Proceedings of the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop,KAW'98*.

Berners-Lee, T. (1999). *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper San Francisco.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001). The semantic web. *Scientific American*, volume 284(5):pages 34–43.
URL `http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html`

Beun, R., van Eijk, R. and Prüst, H. (2004). Ontological feedback in multiagent systems. In *Proceedings of Third International Conference on Autonomous Agents and Multiagent Systems*, pages 110–117. ACM Press, New York.

Blackburn, P., de Rijke, M. and Venema, Y. (2001). *Modal logic*. Cambridge University Press, New York, NY, USA.

Borgida, A. and Serafini, L. (2003). Distributed description logics: Assimilating information from peer sources. In S. Spaccapietra, editor, *Journal of Data Semantics*, volume 1 of *LNCS*, pages 153–184. Springer Verlag.

Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L. and Stuckenschmidt, H. (2003). C-OWL: Contextualizing ontologies. In K. Sekara and J. Mylopoulis, editors, *Proceedings of the Second International Semantic Web Conference*, number 2870 in Lecture Notes in Computer Science, pages 164–179. Springer Verlag.

Brachman, R. J., Fikes, R. and Levesque, H. J. (1983). Krypton: A functional approach to knowledge representation. *IEEE Computer*, volume 16(10):pages 67–73.

Burnstein, M., McDermott, D., Smith, D. and Westfold, S. (2003). Derivation of glue code for agent interoperation. *Autonomous Agents and Multi-Agent Systems*, volume 6(3):pages 265–286.

Bylander, T. and Chandrasekaran, B. (1987). Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition. *International Journal of Man-Machine Studies*, volume 26(2):pages 231–243.

Calí, A., Giacomo, G. D. and Lenzerini, M. (2001). Models for information integration: Turning local-as-view into global-as-view. In *Proceedings of International Workshop on Foundations of Models for Information Integration*.

Carr, B. (1987). *Metaphysics: An Introduction*. Humanities Press International, Atlantic Highlands, New Jersey.

Chalmers, D., French, R., and Hofstadter, D. (1995). High-level perception, representation, and analogy: A critique of artificial intelligence methodology. In D. Hofstadter, editor, *Fluid Concepts & Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. HarperCollins Publishers.

Chandler, D. (2002). *Semiotics: The Basics*. Routledge.

Chisholm, R. (1996). *A Realistic Theory of Categories*. Cambridge University Press, Cambridge.

Ciocoiu, M., Gruniger, M. and Nau, D. (2001). Ontologies for integrating engineering applications. *Journal of Computing and Information Science in Engineering*, volume 1(1):pages 12–22.

Darlington, K. (2000). *The essence of expert systems*. Prentice Hall.

Davis, R., Shrobe, H. E. and Szolovits, P. (1993). What is a knowledge representation? *AI Magazine*, volume 14(1):pages 17–33.

DCMI (2005). The dublin core metadata initiative, DCMI metadata terms.
URL `http://dublincore.org/documents/dcmi-terms/`

Dennett, D. C. (1989). *The Intentional Stance*. The MIT Press.

Dennett, D. C. (1995). *Darwin's Dangerous Idea: Evolution and the Meanings of Life*. Simon & Schuster, New York, NY.

van Diggelen, J., Beun, R., Dignum, F., van Eijk, R. and Meyer, J.-J. (2004). Optimal communication vocabularies and heterogeneous ontologies. In *Agent Communication, International Workshop on Agent Communication, AC 2004*, volume 3396 of *LNAI*, pages 76–90. Springer Verlag.

van Diggelen, J., Beun, R., Dignum, F., van Eijk, R. and Meyer, J.-J. (2005). A decentralized approach for establishing a shared communication vocabulary. *Proceedings of the International Workshop on Agent Mediated Knowledge Management (AMKM'05) held with AAMAS05*.

van Diggelen, J., Beun, R., Dignum, F., van Eijk, R. and Meyer, J.-J. (2006a). ANEMONE: An effective minimal ontology negotiation environment. In P. Stone and G. Weiss, editors, *Proceedings of the Fifth International Conference on Autonomous Agents and Multi-agent Systems (AAMAS06)*, pages 899–906. ACM.

van Diggelen, J., Beun, R., Dignum, F., van Eijk, R. and Meyer, J.-J. (2006b). Combining normal communication with ontology alignment. In *Agent communication II, International Workshops on Agent Communication, AC 2005 and AC 2006*, volume 3859 of *LNAI*, pages 181–195. Springer Verlag.

van Diggelen, J., Beun, R., Dignum, F., van Eijk, R. and Meyer, J.-J. (2007). Ontology negotiation: Goals, requirements and implementation. *International Journal of Agent-Oriented Software Engineering (IJAOSE)*. Inderscience publishers. To appear.

van Diggelen, J., Wiering, M. and de Jong, E. (2006c). Strategies for ontology negotiation: Finding the right level of generality. In *Agent Communication II, International Workshops on Agent Communication, AC 2005 and AC 2006*, volume 3859 of *LNAI*, pages 164–180. Springer Verlag.

Dignum, F. (2000). Agent communication and cooperative information agents. In M. Klusch and L. Kerschberg, editors, *Cooperative Information Agents IV - The Future of Information Agents in Cyberspace*, LNCS-1860, pages 191–207. Springer Verlag.

Doherty, P., Lukaszewicz, W. and Szalas, A. (2003). On mutual understanding among communicating agents. *Proceedings of the Workshop on Formal Approaches to Multi-Agent Systems (FAMA'03)*, pages 83–97.

Dou, D., McDermott, D. and Qi, P. (2004). Ontology translation on the semantic web. In S. Spaccapietra, editor, *Journal of Data Semantics*, volume II of *LNCS*, pages 35–57. Springer-Verlag.

Dreyfus, H. L. (1981). From Micro-Worlds to Knowledge Representation: AI at an Impasse. In J. Haugeland, editor, *Mind Design: Philosophy, Psychology, Artificial Intelligence*, pages 161–204. MIT Press, Cambridge, MA.

Dreyfus, H. L. (1992). *What computers still can't do*. MIT Press, Cambridge, Massachusetts.

Durfee, E. H. (1999). Distributed problem solving and planning. In G. Weiß, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 3, pages 121–164. MIT Press, Cambridge, MA.
URL `http://jmvidal.cse.sc.edu/library/durfee99a.pdf`

van Eijk, R., de Boer, F., van der Hoek, W. and Meyer, J.-J. (2001). On dynamically generated ontology translators in agent communication. *International Journal of Intelligent Systems*, volume 16(5):pages 587–607.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Finin, T., Fritzson, R., McKay, D. and McEntire, R. (1994). KQML as an agent communication language. In N. Adam, B. Bhargava and Y. Yesha, editors, *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463. ACM Press, Gaithersburg, MD, USA.

Fink, E. (2003). *Changes of problem representation: Theory and experiments*. Springer-Verlag, Berlin, Germany.

FIPA (2000). FIPA Ontology Service Specification.
URL `http://www.fipa.org/specs/fipa00086/`

FIPA (2001). FIPA English Auction Interaction Protocol Specification.
URL `http://www.fipa.org/specs/fipa00031/`

FIPA (2002a). FIPA ACL Message Structure Specification.
URL `http://www.fipa.org/specs/fipa00061/`

FIPA (2002b). FIPA Brokering Interaction Protocol Specification.
URL `http://www.fipa.org/specs/fipa00033/`

Frege, G. (1892). On sense and nominatum. In A. P. Martinich, editor, *The Philosophy of Language*, chapter 13. Oxford University Press.

Gärdenfors, P. (2000). *Conceptual spaces: the geometry of thought*. MIT Press.

Genesereth, M. R. and Nilsson, N. J. (1987). *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc.

Ghezzi, C., Jazayeri, M. and Mandrioli, D. (1991). *Fundamentals of software engineering*. Prentice-Hall International, London.

Ginsberg, M. L. (1991). Knowledge interchange format: the KIF of death. *AI Magazine*, volume 12(3):pages 57–63.
URL `citeseer.ist.psu.edu/ginsberg91knowledge.html`

Giovannucci, A. and Rodríguez-Aguilar, J. A. (2005). On the cost of agent-awareness for negotiation services. In B. Henderson-Sellers and M. Winikoff, editors, *Seventh International Bi-Conference Workshop on Agent-Oriented Information Systems AOIS-2005, held with AAMAS05*.

Grenon, P., Smith, B. and Goldberg, L. (2004). Biodynamic ontology: Applying BFO in the biomedical domain. In D. M. Pisanelli, editor, *Ontologies in Medicine*, pages 20–38. IOS Press, Amsterdam.

Grice, H. (1975). Logic and conversation. In P. Cole and J. Morgan, editors, *Speech Acts*, pages 41–58. Academic Press, New York.

Grossi, D., Dignum, F. and Meyer, J.-J. C. (2006). Contextual terminologies. In F. Toni and P. Torroni, editors, *Post-proceedings of CLIMA VI, 6th International Workshop on Computational Logic in Multi-Agent Systems*, number 3900 in LNAI, pages 284–302. Springer-Verlag.

Grossmann, R. (1983). *The Categorical Structure of the World*. Indiana University Press, Bloomington, Indiana.

Gruber, T. R. (1993a). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, Deventer, The Netherlands.

Gruber, T. R. (1993b). A translation approach to portable ontology specifications. *Knowledge Acquisition*, volume 5:pages 199–220.

Guarino, N. (1997). Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, volume 46(2-3):pages 293–310.

Guarino, N. (1998). Formal ontology and information systems. In N. Guarino, editor, *Formal Ontology in Information Systems. Proceedings of FOIS'98*, pages 3–15. IOS Press, Amsterdam.

Harnad, S. (1987). Category induction and representation. In S. Harnad, editor, *Categorical Perception: The Groundwork of Cognition*, chapter 18, pages 535–565. Cambridge University Press.

Harnad, S. (1990). The symbol grounding problem. *Physica D 42*, pages 335–346.

Harnad, S. (1993). Problems, problems: The frame problem as a symptom of the symbol grounding problem. *Psycoloquy*, volume 4(34).

Harrenstein, P. (2004). *Logic in Conflict. Logical Explorations in Strategic Equilibrium*. Ph.D. thesis, Utrecht University.

Harrsch, M. (2003). RSS: the next killer app for education. *The Technology Source*. URL `http://ts.mivu.org/default.asp?show=article\&id=1034`

Heflin, J. and Hendler, J. (2000). Dynamic ontologies on the web. In *Proceedings of American Association for Artificial Intelligence Conference (AAAI-2000)*, pages 443–449. AAAI Press, Menlo Park, CA, USA.

van Heijst, G., Schreiber, A. T. and Wielinga, B. J. (1997). Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, volume 46(2-3):pages 183–292.

Hendler, J. (2001). Agents and the semantic web. *IEEE Intelligent Systems*, volume 16(2):pages 30–37.

Hindriks, K. V., de Boer, F. S., van der Hoek, W. and Meyer, J.-J. C. (1999). Agent programming in 3APL. *Autonomous Agents and Multi-Agent Systems*, volume 2(4):pages 357–401.

Hobbs, J. R. and Pan, F. (2004). An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing (TALIP)*, volume 3(1):pages 66–85.

Huhns, M. N. and Stephens, L. M. (1999). Multiagent systems and societies of agents. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 2, pages 79–120. The MIT Press, Cambridge, MA, USA.

Jackson, P. and Moulinier, I. (2002). *Natural Language Processing for Online Applications: Text retrieval, extraction, and categorization*. John Benjamins Publishing.

Jasper, R. and Uschold, M. (1999). A framework for understanding and classifying ontology applications. In *Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods(KRR5)*. Stockholm, Sweden.

Jennings, N. R., Sycara, K. and Wooldridge, M. (1998). A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, volume 1(1):pages 7–38.

Joachims, T. (1999a). Making large-scale support vector machine learning practical. In *Advances in kernel methods: support vector learning*, pages 169–184. MIT Press, Cambridge, MA, USA.

Joachims, T. (1999b). Transductive inference for text classification using support vector machines. In *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

de Jong, E. D. and Steels, L. (2003). A distributed learning algorithm for communication development. *Complex Systems*, volume 14:pages 315–334.

Kohavi, R. and Provost, F. (1998). Glossary of terms. In *Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, 30, pages 271–274. Kluwer Academic Publishers, Boston.

KSE (1994). Knowledge sharing effort public library.
URL `http://www-ksl.stanford.edu/knowledge-sharing/`

Kuhn, W. (2004). Why information science needs cognitive semantics. *Workshop on the Potential of Cognitive Semantics for Ontologies (FOIS 2004), Torino, Italy*.
URL `http://musil.uni-muenster.de/documents/WhyCogLingv1.pdf`

Lebbink, H.-J. (2006). *Dialogue and Decision Games for Information Exchanging Agents*. Ph.D. thesis, Universiteit Utrecht, Utrecht, The Netherlands.

Lenat, D. B. and Guha, R. V. (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley Publishers.

Lewis, D. (2002). *Convention*. Blackwell Publishing. First published 1969.

Locke, J. (1689). *An Essay Concerning Human Understanding, book III, chapter III*. eBooks@Adelaide, The University of Adelaide Library, Adelaide.
URL `http://etext.library.adelaide.edu.au/l/locke/john/l81u/under42.html`

Luck, M., McBurney, P. and Preist, C. (2003). Agent technology: Enabling next generation computing. *Agent link community*, pages 74–75.

Malik, D. and Sen, M. (2004). *Discrete Mathematical Structures: Theory and Applications*. Thomson course technology.

Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N. and Sycara, K. (2004). Bringing semantics to web services: The OWL-S approach.
URL `http://www.w3.org/Submission/OWL-S/`

Matuszek, C., Cabral, J., Witbrock, M. and DeOliveira, J. (2006). An introduction to the syntax and content of Cyc. In *the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*.

Maynard, D., Stamou, G., Zaihrayeu, I., Barrasa, J., Euzenat, J., Hauswirth, M., Ehrig, M., Bouquet, P., Dieng-Kuntz, R., Hernández, R. L., Tessaris, S., Acker, S. V. and Bach, T.-L. (2004). D2.2.3: State of the art on current alignment techniques. Knowledge Web.
URL `http://knowledgeweb.semanticweb.org/semanticportal/servlet/download?ontology=Documentation+Ontology&concept=Deliverable&instanceSet=kweb&instance=D2.2.3%3A+State+of+the+art+on+current+alignment+techniques&attribute=On-line+PDF+Version&value=kweb-223.pdf`

McDermott, D. (1981). Artificial intelligence meets natural stupidity. In J. Haugeland, editor, *Mind Design: Philosophy, Psychology, Artificial Intelligence*, pages 143–160. MIT Press, Cambridge, MA.

McDermott, D. (1987). A critique of pure reason. *Computational Intelligence*, volume 3:pages 151–160.

McGuinness, D. L., Fikes, R., Rice, J. and Wilder, S. (2000). The chimaera ontology environment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000)*. Austin, Texas.

McGuinness, D. L. and van Harmelen, F. (2004). OWL web ontology language overview.
URL `http://www.w3.org/TR/owl-features/`

Mena, E., Illarramendi, A., Kashyap, V. and Sheth, A. (2000). OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *International journal on Distributed And Parallel Databases (DAPD)*, volume 8(2):pages 223–272.

Meyer, J.-J. C. and van der Hoek, W. (1995). *Epistemic Logic for AI and Computer Science*. Cambridge University Press.

Minsky, M. (1988). *The society of mind*. Simon & Schuster.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Nardi, D. and Brachman, R. (2003). An introduction to description logics. In F. Baader, D. Calvanese, D. McGuiness, D. Nardi and P. Patel-Schneider, editors, *The description logic handbook*, chapter 1, pages 1–40. Cambridge University Press.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. and Swartout, W. R. (1991). Enabling technology for knowledge sharing. *AI Magazine*, volume 12(3):pages 36–56.

Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 2–9. ACM Press, New York, NY, USA.

Noy, N. F. and Klein, M. (2004). Ontology evolution: Not the same as schema evolution. *Knowledge Information Systems*, volume 6(4):pages 428–440.

Noy, N. F. and McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory.
URL `http://ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html`

Noy, N. F. and Musen, M. A. (2000). Prompt: Algorithm and tool for automated ontology merging and alignment. *In Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Obitko, M. and Marik, V. (2003). Mapping between ontologies in agent communication. In *Proceedings of the CEEMAS 2003, Prague, Czech Republic*, Lecture Notes on Artificial Intelligence 2691, pages 191–203. Springer-Verlag.

Ogden, C. K. and Richards, I. A. (1923). *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. Magdalene College, University of Cambridge.

Øhrstrøm, P., Andersen, J. and Schärfe, H. (2005). What has happened to ontology. In *Conceptual Structures: Common Semantics for Sharing Knowledge: 13th International Conference on Conceptual Structures, ICCS 2005*, Lecture Notes in Computer Science, pages 425–438. Springer-Verlag, Kassel, Germany.

Peirce, C. S. (1894). What is a sign?
URL `http://www.marxists.org/reference/subject/philosophy/works/us/peirce1.htm`

Plato (427-347 BC). *Phaedrus*. eBooks@Adelaide, The University of Adelaide Library, Adelaide. Translated by Benjamin Jowett.
URL `http://etext.library.adelaide.edu.au/p/plato/p71phs/phaedrus.html`

Porter, M. (1980). An algorithm for suffix stripping. *Program*, volume 14(3):pages 130–137.

Powers, S. (2005). What are syndication feeds. O'Reilly Media.
URL `http://www.oreilly.com/catalog/syndicationfeeds/`

Preece, A. D., ying Hui, K., Gray, W. A., Marti, P., Bench-Capon, T. J. M., Jones, D. M. and Cui, Z. (2000). The KRAFT architecture for knowledge fusion and transformation. *Knowledge Based Systems*, volume 13(2-3):pages 113–120.

Putnam, H. (1975). The meaning of 'meaning'. In K. Gunderson, editor, *Language, Mind, and Knowledge*, pages 131–193. University of Minnesota Press, Minneapolis.

Quine, W. V. O. (1953). On what there is. In *From a Logical Point of View*, pages 1–19. Harvard University Press, Cambridge, Massachusetts and London, England.

Rahm, E. and Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, volume 10(4):pages 334–350.

Rao, A. S. and Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes and E. Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.
URL `citeseer.ist.psu.edu/rao91modeling.html`

Rice, J. (1995). *Mathematical Statistics and Data Analysis*. Duxbury Press.

van Riemsdijk, M. B., Dastani, M. and Meyer, J.-J. C. (2005). Semantics of declarative goals in agent programming. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 133–140. ACM Press, New York, NY, USA.

Robertson, S. (2000). Evaluation in information retrieval. In F. C. M. Agosti and G. Pasi, editors, *Lectures on Information Retrieval: Third European Summer-School, ESSIR 2000*, volume 1962 of *Lecture Notes in Computer Science*, pages 81–93. Springer Berlin / Heidelberg.

Rodriguez, J. (2001). *On the Design and Construction of Agent-mediated Electronic Institutions*. Ph.D. thesis, Inst. d'Investigació en Intelligència Artificial.

van Rooy, R. (2003). Quality and quantity of information exchange. *Journal of Logic, Language and Information*, volume 12(4):pages 423–451.

Rosen, G. (2006). Abstract objects. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*.
URL `http://plato.stanford.edu/archives/spr2006/entries/abstract-objects/`

Rosenschein, J. S. and Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, Massachusetts.

Rota, G.-C. (1985). The barrier of meaning. In *Letters in Mathematical Physics*, volume 10, pages 97–99. Springer Netherlands.

Roy, D. (2005). Semiotic schemas: a framework for grounding language in action and perception. *Artificial Intelligence, Special volume on connecting language to the world*, volume 167(1-2):pages 170–205.

Russel, S. and Norvig, P. (2003). *Artificial Intelligence, a modern approach*. Prentice Hall.

Ryan, V. (2003). Will IBM's new 'Xperanto' take over the enterprise? *Ecommerce Times*.
URL `http://www.ecommercetimes.com/story/20762.html`

Schild, K. (1991). A correspondence theory for terminological logics: preliminary report. In *Proceedings of IJCAI-91, 12th International Joint Conference on Artificial Intelligence*, pages 466–471. Sidney, AU.

Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., de Velde, W. V. and Wielinga, B. (2000). *Knowledge Engineering and Management, The CommonKADS Methodology*. MIT Press.

Searle, J. and Vanderveken, D. (1985). *Foundations of illocutionary logic*. Cambridge University Press.

Searle, J. R. (1980). Minds, brains, and programs. *The Behavioral and Brain Sciences*, volume 3.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, volume 27:pages 379–423.

Shin, S.-J. and Lemon, O. (2006). Diagrams. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*.
URL `http://plato.stanford.edu/archives/sum2006/entries/diagrams/`

Smith, B. (2003). Ontology. In L. Floridi, editor, *Blackwell Guide to the Philosophy of Computing and Information*, chapter 11, pages 155–166. Blackwell, Oxford.

Smith, B. and Welty, C. A. (2001). Ontology - towards a new synthesis. In B. Smith and C. A. Welty, editors, *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems, FOIS*, pages iii–ix. ACM Press.

Soh, L.-K. and Chen, C. (2005). Balancing ontological and operational factors in refining multiagent neighborhoods. *Proceedings of the Fourth International Conference on Autonomous Agents and Multi-Agent Systems*.

Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole Publishing Co., Pacific Grove, CA, USA.

Steels, L. (1998a). The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, volume 1(2):pages 169–194.

Steels, L. (1998b). Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation. In J. Hurford, C. Knight and M. Studdert-Kennedy, editors, *Approaches to the Evolution of Language: Social and Cognitive Bases*. Edinburgh University Press.
URL `http://arti.vub.ac.be/steels/edin.ps`

Steels, L. (2000). Language as a complex adaptive system. In M. Schoenauer, editor, *Proceedings of PPSN VI*, Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany.

Steve, G., Gangemi, A. and Pisanelli, D. (1997). Integrating medical terminologies with ONIONS methodology. In C. J. Kangassalo H, editor, *Information Modelling and Knowledge Bases*, volume VIII. IOS Press, Amsterdam.

Stuckenschmidt, H. and Timm, I. J. (2002). Adapting communication vocabularies using shared ontologies. *Proceedings of the Second International Workshop on Ontologies in Agent Systems (OAS)*.
URL `http://cis.otago.ac.nz/OAS2002/Papers/oas02-3.pdf`

Studer, R., Benjamins, V. R. and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data Knowledge Engineering*, volume 25(1-2):pages 161–197.

Stumme, G. and Maedche, A. (2001). FCA-MERGE: Bottom-up merging of ontologies. In *IJCAI*, pages 225–234.

Sure, Y., Staab, S. and Studer, R. (2004). On-to-knowledge methodology (OTKM). In S. Staab and R. Studer, editors, *Handbook on ontologies*, chapter 6, pages 117–132. Springer-Verlag.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. In *IEEE Transactions on Speech and Audio Processing*, volume 10.

Tzitzikas, Y. and Meghini, C. (2003). Ostensive automatic schema mapping for taxonomy-based peer-to-peer systems. In *Proceedings of the 7th International Workshop on Cooperative Information Agents*. Helsinki, Finland.

Uschold, M. and Grüninger, M. (1996). Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, volume 11(2):pages 93–155.

Uschold, M. and Grüninger, M. (2002). Creating semantically integrated communities on the world wide web. *Semantic Web Workshop Co-located with WWW 2002 Honolulu*.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Visser, P. R. S., Jones, D. M., Bench-Capon, T. J. M. and Shave, M. J. R. (1997). An analysis of ontological mismatches: Heterogeneity versus interoperability. In *AAAI 1997 Spring Symposium on Ontological Engineering*. Stanford, USA.

Vogt, P. (2002). The physical symbol grounding problem. *Cognitive Systems Research*, volume 3(3):pages 429–457.

Wang, J. and Gasser, L. (2002). Mutual online ontology alignment. *Proceedings of the Workshop on Ontologies in Agent Systems, held with AAMAS 2002*.

Wang, J., Gasser, L. and Houk, J. (2006). Convergence analysis for collective vocabulary development. *Proceedings of the Fifth International Conference on Autonomous Agents and Multi-agent Systems (AAMAS06)*, pages 1378–1380.

Wang, P. (1995). Grounded on experience: Semantics for intelligence. Technical Report 96, Center for Research on Concepts and Cognition, Indiana University, Bloomington, Indiana.
URL `http://www.cogsci.indiana.edu/farg/peiwang/papers.html`

Wiederhold, G. and Genesereth, M. (1997). The conceptual basis for mediation services. *IEEE Expert: Intelligent Systems and Their Applications*, volume 12(5):pages 38–47.

Wielinga, B. J. and Schreiber, A. T. (1993). Reusable and shareable knowledge bases: A european perspective. In *Proceedings of the International Conference on Building and Sharing of Very Large-Scaled Knowledge Bases*, pages 103–115. Japan Information Processing Development Center., Tokyo, Japan.

Wiesman, F. and Roos, N. (2004). Domain independent learning of ontology mappings. In *Proceedings of Third International Conference on Autonomous Agents and Multiagent Systems*, pages 844–851. ACM Press, New York.

Williams, A. (2004). Learning to share meaning in a multi-agent system. *Autonomous Agents and Multi-Agent Systems*, volume 8(2):pages 165–193.

Wittgenstein, L. (1953). *Philosophische untersuchungen - Philosophical investigations.*. Basil Blackwell, Oxford, German-English edition.

Woese, C. R. and Fox, G. E. (1977). Phylogenetic structure of the prokaryotic domain: The primary kingdoms. *Proceedings of the National Academy of Sciences of the United States of America*, volume 74(11):pages 5088–5090.

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, volume 10(2):pages 115–152.

Zúñiga, G. L. (2001). Ontology: its transformation from philosophy to information systems. In B. Smith and C. A. Welty, editors, *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems, FOIS*, pages 187–197. ACM Press.

Samenvatting

> *"Neem me niet kwalijk!", mompelde de oude, "er schijnt een fout in mijn denkraam te zijn! Ik volg u niet. Ik heb daar trouwens meer last van, van mijn denkraam bedoel ik."*
>
> Marten Toonder, 1950

Moderne computersystemen vergen een geavanceerde vorm van communicatie. Er is geen sprake meer van één centraal softwaresysteem dat instructies uitvoert. Er zijn meerdere softwaresystemen die autonoom opereren en flexibel met elkaar samenwerken. We noemen zo'n systeem een *multi-agent systeem*. Het is hierbij van cruciaal belang dat de software agenten in het systeem elkaar kunnen begrijpen. In vaktermen noemen we dit onderlinge begrip *semantische interoperabiliteit*. Deze eigenschap is niet vanzelfsprekend in multi-agent systemen. Verschillende software agenten kunnen verschillende achtergronden hebben wat tot misverstanden kan leiden. Het probleem is vergelijkbaar met mensen die verschillende talen spreken of een verschillend referentiekader hebben. In dit proefschrift gaan we deze Babylonische spraakverwarring te lijf door software agenten te ontwerpen die tijdens hun conversatie misverstanden kunnen herkennen en oplossen. Ze voeren een dialoog waarin taal niet alleen als communicatiemiddel wordt gebruikt maar waarin de taal ook zelf het onderwerp van gesprek kan worden. Bijvoorbeeld, de spreker verheldert de betekenis van zijn woorden, waarna de toehoorder aangeeft of deze toelichting voldoende is. Zoals de titel aangeeft, gaat dit proefschrift over het tot stand brengen van semantische interoperabiliteit in multi-agent systemen middels een dialoog. Voordat we beschrijven hoe zo'n dialoog eruit ziet, zullen we eerst de twee reeds genoemde begrippen verder uitdiepen, namelijk multi-agent systeem en semantische interoperabiliteit.

Een veel gebruikt voorbeeld van een multi-agent systeem is dat van de software reisagent die klanten te woord staat en helpt bij het boeken van een reis. De agent onderhandelt op het internet over goedkope vluchten met andere software agenten en probeert geschikte accommodatie te boeken. Tijdens het verkennen van de markt ontmoet de reisagent veel andersoortige agenten waarmee de communicatie niet altijd vloeiend verloopt. Zo staan *vluchtgegevens* waar de Nederlandse reisagent om vraagt, bij de Engelstalige luchtvaartagenten bekend als *flight information*. Ook gebruiken sommige agenten begrippen die volkomen vreemd zijn voor de reisagent, bijvoorbeeld een vliegtuigtechnicus die spreekt over een *traagheids-sensor*. In andere woorden, de semantische interoperabiliteit schiet tekort in deze omgeving.

Voor semantische interoperabiliteit is de notie van ontologie van belang. Ontologieën worden al onderzocht sinds de oude Grieken. Zij waren op zoek naar de fundamentele categorieën waarin alles wat bestaat valt onder te brengen. De meeste agentonderzoekers vatten het begrip ontologie wat minder filosofisch op. Voor hen is het een pragmatisch middel om te specificeren wat voor een agent van belang is. Bijvoorbeeld, de ontologie van de reisagent bestaat uit een aantal reis gerelateerde begrippen zoals *vluchtgegevens*. Ditzelfde begrip is aanwezig in de ontologie van de Engelstalige luchtvaart agent, maar dan onder de naam *flight information*. De ontologie van de vliegtuigtechnicus bevat allerlei technische begrippen zoals *traagheids-sensor*. Het feit dat de ontologieën van agenten verschillen ligt dus aan de basis van hun communicatieproblemen. De oplossing voor het semantische interoperabiliteits probleem schuilt dan ook in het op één lijn krijgen van de verschillende ontologieën. Bijvoorbeeld, de reisagent moet erachter komen dat *flight information* hetzelfde betekent als *vluchtgegevens* en dat *traagheids-sensor* een begrip is wat voor hem niet relevant is.

Software agenten converseren niet vanzelf. Ze hebben nauwkeurig uiteengezette instructies nodig over wat ze moeten zeggen in welke situaties. We noemen dit een dialoogprotocol. Behalve dat we een dialoog gebaseerde benadering volgen, stellen we nog wat andere eisen aan de oplossing. Aan de ene kant moeten de agenten alleen hun semantische interoperabiliteits problemen te lijf gaan wanneer dit strikt noodzakelijk is. Op deze manier verliezen ze hier zo min mogelijk tijd en rekenkracht mee. Aan de andere kant moet gewaarborgd zijn dat er effectief gecommuniceerd kan worden. Dat wil zeggen dat de agenten voldoende informatie uitwisselen en ze de ontvangen berichten juist interpreteren. Het evenwicht hiertussen wordt gevonden door de agenten een communicatievocabulaire te laten opbouwen wat effectief en van minimale omvang is.

De communicatieprotocollen die we in dit proefschrift voorstellen laten de agenten aanvankelijk normaal met elkaar communiceren. Alleen wanneer de toehoorder een woord niet kan begrijpen, zal de spreker de betekenis van dat woord uitleggen. In eerste instantie gebeurt dit door het geven van een definitie. De spreker legt het onbegrepen woord uit door middel van andere woorden die wel begrijpelijk zijn. Soms is dit echter ook niet mogelijk en hebben de agenten zo weinig woorden gemeenschappelijk dat ze elkaars definities ook niet kunnen begrijpen. In deze gevallen legt de spreker het woord uit door aan te wijzen wat er bedoeld wordt.

De communicatieprotocollen worden verondersteld effectieve communicatie mogelijk te maken middels een vocabulaire van minimale omvang. Natuurlijk moet zo'n bewering wetenschappelijk onderbouwd worden. In dit proefschrift doen we dat op drie manieren. Ten eerste geven we een wiskundige onderbouwing. Door het communicatieprotocol wiskundig te formuleren, kunnen we een bewijs geven dat het de gewenste eigenschappen bezit. Ten tweede voeren we simulatie-experimenten uit. Door de agenten in een simulatie-omgeving met elkaar te laten communiceren kunnen we verifiëren dat er een minimaal vocabulaire opgebouwd wordt wanneer alle agenten communiceren volgens het protocol. Ten derde beschrijven we een praktische toepassing met internet nieuwsagenten. Deze agenten wisselen nieuwsberichten uit over verschillende onderwerpen. Wanneer een agent niet begrijpt welk onderwerp bedoeld wordt met een bepaald woord, ligt de andere agent dit toe door voorbeeldnieuwsberichten over dit onderwerp te tonen. Op deze manier leren de agenten nieuwe woorden en wordt geleidelijk een effectief communicatievocabulaire opgebouwd.

In dit proefschrift hebben wij een aantal algemene principes van semantische interoperabiliteit tussen agenten geïdentificeerd. Tevens hebben we een voorstel gedaan voor een geschikt communicatieprotocol. De resultaten kunnen nu al gebruikt worden op bepaalde domeinen, zoals het verspreiden van nieuwsberichten op het internet. Ook kunnen zij dienen als basis voor vervolgonderzoek. Toekomstig onderzoek zou zich kunnen richten op het uitleggen van ingewikkeldere begrippen, zoals bijvoorbeeld het begrip *periodiciteit*. Een abstract begrip zoals deze is niet uit te leggen door een voorbeeld ervan aan te wijzen. Waarschijnlijk kunnen dergelijke problemen pas worden opgelost als de computer een *diep* taalbegrip bezit, vergelijkbaar met dat van de mens.

Index

$\mathcal{ALC}$, 43, 106

ABox, 23, 44, 62, 106
ACL, *see* agent communication language
agent, 2, 11, 62
  cognitive, 12
  environment, 11
agent communication, 13
  language, 15
  protocol, 16, 63, 64, 114
  strategy, 16, 63–65, 130
agent communication language, 13
ANEMONE, 61, 132, 155, 168
Aristotelian essentialism, *see* essentialism
assertion, 69
  concept, 44
  role, 44
assertional knowledge base, 62, 65, 104
automatic ontology mapping, 5, 35, 51, 52, 57–58, 114
autonomous, 2, 11

bisimulation, 48
bottom, 102

categorical realism, *see* essentialism
CDP, *see* concept definition protocol
CEP, *see* concept explication protocol
Chinese room experiment, 50
common knowledge, 69, 105
communication
  effective, 78, 164, 168
  lossless, 72, 77, 109, 111, 118, 121, 126, 168
  non-symbolic, 55–56, 63, 71, 85
  sound, 109, 126, 168
  symbolic, 42, 54, 63, 94–95
communication vocabulary, 32, 63, 65, 70, 100, 104, 134
  convergence, 131, 141
  lossless, 111
  lossless distribution, 112
  minimal, 72, 76, 126, 164, 168
  optimal, 111, 130
  optimal distribution, 112
complete meaning, 83
concept
  acquired, 68, 74, 91, 164
  mutually redundant, 123, 124, 168
  native, 62, 66, 68, 74, 91
  overgeneralized, 78, 80, 109, 121, 130, 135, 145, 148
  overspecific, 130, 145
  shared, 69, 74
  unknowingly shared, 69, 81, 96, 134, 165
concept classifier, 53, 63, 65, 69, 108, 159, 168
concept definition protocol, 71, 83–85
concept explication protocol, 71, 85–89, 169
  non-compositional, 116, 128
concept learning, 56
conceptual relativism, 19, 20, 22, 26

conceptualization, 24, 43, 63, 64, 101, 133, 144
confusion matrix, 85
context, 127
conventionalism, *see* conceptual relativism
conversational implicature, 15, 80, 119
cv, *see* communication vocabulary
Cyc, 25

description logic, 43–47
  semantics, 44
  syntax, 43
domain of discourse, 42, 63, 64, 101

effectiveness, *see* communication, effective
equivalence mapping, 70, 74, 164
Esperanto, 32, 71
essentialism, 18, 20, 25, 28

FIPA ACL, 15

goal base, 62, 63, 65, 69
Gricean maxims, 15, 132
grounded knowledge representation, *see* semantics, grounded

illocutionary force, 14
implicit knowledge, 105, 107
implied concept, 68
individual, 43, 78, 106, 114
intended interpretation, 47, 102, 133
interlingua, *see* ontology,intermediate
interoperability
  measures, 136–137
  semantic, 2, 14, 167
  syntactic, 13
interpretation function, 44, 65

KQML, 15

language evolution, 132
lazy, *see* ontology reconciliation, lazy, *see* ontology reconciliation, lazy
local knowledge, 104
lossless, *see* communication, lossless; communication vocabulary, lossless

meaning, 41, 133
meaning candidates, 136
meaning selection strategy, 130, 136, 144–152
message composition, 73, 135, 136
  approximate, 76, 97, 164
message interpretation, 73, 135
  approximate, 75, 97
message protocol, 89
model, 44
  indistinguishable, 47, 57
  Kripke, 48
modularization, 9
MPUR, 136
MSS, *see* meaning selection strategy
multi-agent system, 2, 12
  development, 12, 28, 30, 32
  heterogeneous, 12, 64
  maintenance, 12
  open, 2, 12

namespace, 66, 95, 104
NCP, *see* normal communication protocol
negotiation, 36
normal communication protocol, 71, 73–83
NW, 141

object, 43, 49, 50, 102
  symbol, 64
ON, *see* ontology negotiation
ontological commitment, 21
ontology, 2, 63, 102, 133
  acquired, 68
  all-purpose, 25–26, 30
  common, *see* shared
  definition, 22, 24
  dynamic, 69, 105
  god's eye view, 104
  heterogeneous, 3, 28, 145, 170
  homogeneous, *see* shared

intermediate, 32, 70
native, 68, 100, 104
philosophical, 17–22, 24
pragmatic, 19, 56
shared, 2, 17, 24, 70, 167
static, 69, 105
task-dependent, 26–28, 30
top-level, 25
ontology agent, *see* ontology mediation service
ontology alignment, 31–33
ontology integration, 29–31
ontology mapping, 31, 70, 101
ontology mediation service, 33–34, 96
ontology negotiation, 34–36
in groups, 90–94, 129
ontology reconciliation, 29, 54, 64
automatic, *see* automatic ontology mapping
hybrid, 36–37, 95
lazy, 73, 76, 84, 126, 168
manual, 31, 35, 57
ostensive definition, 55
limitations, 55–56
overgeneralization, *see* concept, overgeneralized

particularization, 68, 121
Peirce's triad, *see* Peircean model
Peircean model, 39–41, 51, 52, 54
performative, 16
personal news agent, 157
Pidgin language, 71
plan base, 62, 63, 65, 69
PNA, *see* personal news agent
point-to-point topology, 31, 70, 137
pro-active, 2, 11
protocol, *see* agent communication, protocol
PUR, 137

reactive, 2, 11
referent, 41, 43, 51
reification, 58, 67, 171
role, 43, 128
RSS, 155
aggregator, 156
feed, 156

scale-free network, 143
schema integration, 29
self-organization, 131
semantic web, 24, 43, 169
semantics
cognitive, 52
grounded, 51–54, 168
perceptually grounded, 52
Tarskian, 42, 43, 46
semiosis, 41
unlimited, 51
semiotic triangle, *see* Peircean model
semiotics, 39, 42, 51, 57
sign, 39
simulation experiment, 6, 131, 138, 147, 153
sinn und bedeutung, 41, 56
situatedness, 11, 52
social, 2, 11
speech act, 14, 15
spoken words, 140
strategy, *see* agent communication, strategy
subjectively equivalent, 109–111
support vector machine, 161
SW, *see* spoken words
symbol, 40, 51
symbol grounding problem, 50–52
physical, 52

TBox, 23, 44, 47, 49, 62, 106, 107
terminological axiom, 44, 45, 106, 107
thought or reference, 41, 51
TNR, *see* true negative rate
top, 77, 102, 109
TPR, *see* true positive rate
transferendum, 108
transferens, 108
translatum, 108
travel agent, 1, 12, 36
true negative rate, 85, 161

true positive rate, 85, 161

understandable words, 140
understandings rate, 138
ungrounded knowledge representation, 51, 57, 170
unknown concept relation, 92
UR, *see* understandings rate
UW, *see* understandable words

word candidates, 136
word selection strategy, 130, 136–144
Wordnet, 57, 158
world, 42, 62
world view, 19, 31, 69, 108
WSS, *see* word selection strategy

SIKS Dissertation Series

1998

Johan van den Akker, *DEGAS - An Active, Temporal Database of Autonomous Objects*, CWI, 1998-1

Floris Wiesman, *Information Retrieval by Graphically Browsing Meta-Information*, UM, 1998-2

Ans Steuten, *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*, TUD, 1998-3

Dennis Breuker, *Memory versus Search in Games*, UM, 1998-4

E.W. Oskamp, *Computerondersteuning bij Straftoemeting*, RUL, 1998-5

1999

Mark Sloof, *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*, VU, 1999-1

Rob Potharst, *Classification using decision trees and neural nets*, EUR, 1999-2

Don Beal, *The Nature of Minimax Search*, UM, 1999-3

Jacques Penders, *The practical Art of Moving Physical Objects*, UM, 1999-4

Aldo de Moor, *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*, KUB, 1999-5

Niek J.E. Wijngaards, *Re-design of compositional systems*, VU, 1999-6

David Spelt, *Verification support for object database design*, UT, 1999-7

Jacques H.J. Lenting, *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.*, UM, 1999-8

2000

Frank Niessink, *Perspectives on Improving Software Maintenance*, VU, 2000-1

Koen Holtman, *Prototyping of CMS Storage Management*, TUE, 2000-2

Carolien M.T. Metselaar, *Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief*, UVA, 2000-3

Geert de Haan, *ETAG, A Formal Model of Competence Knowledge for User Interface Design*, VU, 2000-4

Ruud van der Pol, *Knowledge-based Query Formulation in Information Retrieval*, UM, 2000-5

Rogier van Eijk, *Programming Languages for Agent Communication*, UU, 2000-6

Niels Peek, *Decision-theoretic Planning of Clinical Patient Management*, UU, 2000-7

Veerle Coup, *Sensitivity Analyis of Decision-Theoretic Networks*, EUR, 2000-8

Florian Waas, *Principles of Probabilistic Query Optimization*, CWI, 2000-9

Niels Nes, *Image Database Management System Design Considerations, Algorithms and Architecture*, CWI, 2000-10

Jonas Karlsson, *Scalable Distributed Data Structures for Database Management*, CWI, 2000-11

2001

Silja Renooij, *Qualitative Approaches to Quantifying Probabilistic Networks*, UU, 2001-1

Koen Hindriks, *Agent Programming Languages: Programming with Mental Models*, UU, 2001-2

Maarten van Someren, *Learning as problem solving*, UvA, 2001-3

Evgueni Smirnov, *Conjunctive and Disjunctive*

Version Spaces with Instance-Based Boundary Sets, UM, 2001-4

Jacco van Ossenbruggen, *Processing Structured Hypermedia: A Matter of Style*, VU, 2001-5

Martijn van Welie, *Task-based User Interface Design*, VU, 2001-6

Bastiaan Schonhage, *Diva: Architectural Perspectives on Information Visualization*, VU, 2001-7

Pascal van Eck, *A Compositional Semantic Structure for Multi-Agent Systems Dynamics*, VU, 2001-8

Pieter Jan 't Hoen, *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*, RUL, 2001-9

Maarten Sierhuis, *Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design*, UvA, 2001-10

Tom M. van Engers, *Knowledge Management: The Role of Mental Models in Business Systems Design*, VUA, 2001-11

2002

Nico Lassing, *Architecture-Level Modifiability Analysis*, VU, 2002-01

Roelof van Zwol, *Modelling and searching web-based document collections*, UT, 2002-02

Henk Ernst Blok, *Database Optimization Aspects for Information Retrieval*, UT, 2002-03

Juan Roberto Castelo Valdueza, *The Discrete Acyclic Digraph Markov Model in Data Mining*, UU, 2002-04

Radu Serban, *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*, VU, 2002-05

Laurens Mommers, *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*, UL, 2002-06

Peter Boncz, *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*, CWI, 2002-07

Jaap Gordijn, *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*, VU, 2002-08

Willem-Jan van den Heuvel, *Integrating Modern Business Applications with Objectified Legacy Systems*, KUB, 2002-09

Brian Sheppard, *Towards Perfect Play of Scrabble*, UM, 2002-10

Wouter C.A. Wijngaards, *Agent Based Modelling of Dynamics: Biological and Organisational Applications*, VU, 2002-11

Albrecht Schmidt, *Processing XML in Database Systems*, UVA, 2002-12

Hongjing Wu, *A Reference Architecture for Adaptive Hypermedia Applications*, TUE, 2002-13

Wieke de Vries, *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*, UU, 2002-14

Rik Eshuis, *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*, UT, 2002-15

Pieter van Langen, *The Anatomy of Design: Foundations, Models and Applications*, VU, 2002-16

Stefan Manegold, *Understanding, Modeling, and Improving Main-Memory Database Performance*, UVA, 2002-17

2003

Heiner Stuckenschmidt, *Onotology-Based Information Sharing In Weakly Structured Environments*, VU, 2003-1

Jan Broersen, *Modal Action Logics for Reasoning About Reactive Systems*, VU, 2003-02

Martijn Schuemie, *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*, TUD, 2003-03

Milan Petkovic, *Content-Based Video Retrieval Supported by Database Technology*, UT, 2003-04

Jos Lehmann, *Causation in Artificial Intelligence and Law - A modelling approach*, UVA, 2003-05

Boris van Schooten, *Development and specification of virtual environments*, UT, 2003-06

Machiel Jansen, *Formal Explorations of Knowledge Intensive Tasks*, UvA, 2003-07

Yongping Ran, *Repair Based Scheduling*, UM, 2003-08

Rens Kortmann, *The resolution of visually guided behaviour*, UM, 2003-09

Andreas Lincke, *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*, UvT, 2003-10

Simon Keizer, *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*, UT, 2003-11

Roeland Ordelman, *Dutch speech recognition in multimedia information retrieval*, UT, 2003-12

Jeroen Donkers, *Nosce Hostem - Searching with Opponent Models*, UM, 2003-13

Stijn Hoppenbrouwers, *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*, KUN, 2003-14

Mathijs de Weerdt, *Plan Merging in Multi-Agent Systems*, TUD, 2003-15

Menzo Windhouwer, *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses*, CWI, 2003-16

David Jansen, *Extensions of Statecharts with Probability, Time, and Stochastic Timing*, UT, 2003-17

Levente Kocsis, *Learning Search Decisions*, UM, 2003-18

2004

Virginia Dignum, *A Model for Organizational Interaction: Based on Agents, Founded in Logic*, UU, 2004-01

Lai Xu, *Monitoring Multi-party Contracts for E-business*, UvT, 2004-02

Perry Groot, *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*, VU, 2004-03

Chris van Aart, *Organizational Principles for Multi-Agent Architectures*, UVA, 2004-04

Viara Popova, *Knowledge discovery and monotonicity*, EUR, 2004-05

Bart-Jan Hommes, *The Evaluation of Business Process Modeling Techniques*, TUD, 2004-06

Elise Boltjes, *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*, UM, 2004-07

Joop Verbeek, *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise*, UM, 2004-08

Martin Caminada, *For the Sake of the Argument; explorations into argument-based reasoning*, VU, 2004-09

Suzanne Kabel, *Knowledge-rich indexing of learning-objects*, UVA, 2004-10

Michel Klein, *Change Management for Distributed Ontologies*, VU, 2004-11

The Duy Bui, *Creating emotions and facial expressions for embodied agents*, UT, 2004-12

Wojciech Jamroga, *Using Multiple Models of Reality: On Agents who Know how to Play*, UT, 2004-13

Paul Harrenstein, *Logic in Conflict. Logical Explorations in Strategic Equilibrium*, UU, 2004-14

Arno Knobbe, *Multi-Relational Data Mining*, UU, 2004-15

Federico Divina, *Hybrid Genetic Relational Search for Inductive Learning*, VU, 2004-16

Mark Winands, *Informed Search in Complex Games*, UM, 2004-17

Vania Bessa Machado, *Supporting the Construction of Qualitative Knowledge Models*, UvA, 2004-18

Thijs Westerveld, *Using generative probabilistic models for multimedia retrieval*, UT, 2004-19

Madelon Evers, *Learning from Design: facilitating multidisciplinary design teams*, Nyenrode, 2004-20

2005

Floor Verdenius, *Methodological Aspects of Designing Induction-Based Applications*, UVA, 2005-01

Erik van der Werf, *AI techniques for the game of Go*, UM, 2005-02

Franc Grootjen, *A Pragmatic Approach to the Conceptualisation of Language*, RUN, 2005-03

Nirvana Meratnia, *Towards Database Support for Moving Object data*, UT, 2005-04

Gabriel Infante-Lopez, *Two-Level Probabilistic Grammars for Natural Language Parsing*, UVA, 2005-05

Pieter Spronck, *Adaptive Game AI*, UM, 2005-06

Flavius Frasincar, *Hypermedia Presentation Generation for Semantic Web Information Systems*, TUE, 2005-07

Richard Vdovjak, *A Model-driven Approach for Building Distributed Ontology-based Web Applications*, TUE, 2005-08

Jeen Broekstra, *Storage, Querying and Inferencing for Semantic Web Languages*, VU, 2005-09

Anders Bouwer, *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*, UVA, 2005-10

Elth Ogston, *Agent Based Matchmaking and Clustering - A Decentralized Approach to Search*, VU, 2005-11

Csaba Boer, *Distributed Simulation in Industry*, EUR, 2005-12

Fred Hamburg, *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*, UL, 2005-13

Borys Omelayenko, *Web-Service configuration on* *the Semantic Web; Exploring how semantics meets pragmatics*, VU, 2005-14

Tibor Bosse, *Analysis of the Dynamics of Cognitive Processes*, VU, 2005-15

Joris Graaumans, *Usability of XML Query Languages*, UU, 2005-16

Boris Shishkov, *Software Specification Based on Re-usable Business Components*, TUD, 2005-17

Danielle Sent, *Test-selection strategies for probabilistic networks*, UU, 2005-18

Michel van Dartel, *Situated Representation*, UM, 2005-19

Cristina Coteanu, *Cyber Consumer Law, State of the Art and Perspectives*, UL, 2005-20

Wijnand Derks, *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*, UT, 2005-21

2006

Samuil Angelov, *Foundations of B2B Electronic Contracting*, TUE, 2006-01

Cristina Chisalita, *Contextual issues in the design and use of information technology in organizations*, VU, 2006-02

Noor Christoph, *The role of metacognitive skills in learning to solve problems*, UVA, 2006-03

Marta Sabou, *Building Web Service Ontologies*, VU, 2006-04

Cees Pierik, *Validation Techniques for Object-Oriented Proof Outlines*, UU, 2006-05

Ziv Baida, *Software-aided Service Bundling – Intelligent Methods & Tools for Graphical Service Modeling*, VU, 2006-06

Marko Smiljanic, *XML schema matching – balancing efficiency and effectiveness by means of clustering*, UT, 2006-07

Eelco Herder, *Forward, Back and Home Again – Analyzing User Behavior on the Web*, UT, 2006-08

Mohamed Wahdan, *Automatic Formulation of the Auditor's Opinion*, UM, 2006-09

Ronny Siebes, *Semantic Routing in Peer-to-Peer Systems*, VU, 2006-10

Joeri van Ruth, *Flattening Queries over Nested Data Types*, UT, 2006-11

Bert Bongers, *Interactivation – Towards an e-cology of people, our technological environment, and the arts*, VU, 2006-12

Henk-Jan Lebbink, *Dialogue and Decision Games for Information Exchanging Agents*, UU, 2006-13

Johan Hoorn, *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*, VU, 2006-14

Rainer Malik, *CONAN: Text Mining in the Biomedical Domain*, UU, 2006-15

Carsten Riggelsen, *Approximation Methods for Efficient Learning of Bayesian Networks*, UU, 2006-16

Stacey Nagata, *User Assistance for Multitasking with Interruptions on a Mobile Device*, UU, 2006-17

Valentin Zhizhkun, *Graph transformation for Natural Language Processing*, UVA, 2006-18

Birna van Riemsdijk, *Cognitive Agent Programming: A Semantic Approach*, UU, 2006-19

Marina Velikova, *Monotone models for prediction in data mining*, UvT, 2006-20

Bas van Gils, *Aptness on the Web*, RUN, 2006-21

Paul de Vrieze, *Fundaments of Adaptive Personalisation*, RUN, 2006-22

Ion Juvina, *Development of Cognitive Model for Navigating on the Web*, UU, 2006-23

Laura Hollink, *Semantic Annotation for Retrieval of Visual Resources*, VU, 2006-24

Madalina Drugan, *Conditional log-likelihood MDL and Evolutionary MCMC*, UU, 2006-25

Vojkan Mihajlovic, *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*, UT, 2006-26

Stefano Bocconi, *Vox Populi: generating video documentaries from semantically annotated media repositories*, CWI, 2006-27

Borkur Sigurbjornsson, *Focused Information Access using XML Element Retrieval*, UVA, 2006-28

2007

Kees Leune, *Access Control and Service-Oriented Architectures*, UvT, 2007-01

Wouter Teepe, *Reconciling Information Exchange and Confidentiality: A Formal Approach*, RUG, 2007-02

Peter Mika, *Social Networks and the Semantic Web*, VU, 2007-03

Jurriaan van Diggelen, *Achieving Semantic Interoperability in Multi-agent Systems: A Dialogue-based Approach*, UU, 2007-04