
Ontology Negotiation: Goals, Requirements and Implementation

Jurriaan van Diggelen, Robbert-Jan
Beun, Frank Dignum, Rogier M. van
Eijk, John-Jules Meyer

Institute of Information and Computing Sciences
Utrecht University, the Netherlands
{jurriaan, rj, dignum, roger, jj}@cs.uu.nl

Abstract

Communication in heterogeneous multi agent systems is hampered by the lack of shared ontologies. Ontology negotiation offers an integrated approach that enables agents to gradually build towards a semantically integrated system by sharing parts of their ontologies. This solution involves a combination of a normal agent communication protocol with an ontology alignment protocol. For such a combination to be successful, it must satisfy several criteria. This paper discusses the goals and requirements that are important for any ontology negotiation protocol. Furthermore, we will propose some implementations that are constructed according to these criteria.

1 Introduction

Most protocols which are studied in the agent communication community build on the assumption that the agents share a common ontology (we refer to these as *normal* communication protocols). However, normal communication protocols are difficult to apply in open multi agent systems, such as those on the internet, in which common ontologies are typically *not* available. In these systems, it is difficult to realize consensus between all involved system developers on which ontology to use [15]. This has motivated researchers to develop tools that assist people in creating generally shared ontologies. Chimaera [20] and FCA-Merge [33] are examples of tools that assist people in merging ontologies. Other approaches aim at developing one all-purpose ontology which is to be used as a “golden standard” by everyone; examples of such large scale ontologies are Cyc [18] and Sensus [34]. However, the common ontology paradigm forces every agent to use the same ontology and thereby give up their own way of viewing the world. Because an ontology

is task-dependent [8], this may be disadvantageous for the problem solving capacity of an agent.

Ontology alignment [24] has been proposed as a technique that enables agents to keep their individual ontologies by making use of *mappings* between the different ontologies. To communicate, the agents exploit the mapping and translate to and from each other's ontology. Most techniques for reconciling heterogeneous ontologies in the semantic web (e.g. [6],[31]) also adhere to the alignment approach, i.e. the original ontologies are linked by semantic mappings. Although ontology alignment is a step in the right direction to achieve a semantically integrated multi agent system, it assumes that the mappings are pre-defined before the agents start interacting. In an open system, agents may enter and leave the system at any moment. Therefore, it should also be possible to align ontologies at agent interaction time.

One way to solve this problem may be to use mediation services [37] or *ontology agents* [1]. 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, thereby being more flexible than ontology alignment. However, this argument only holds when every agent trusts and knows how to find the ontology agent. Furthermore, the ontology agent should be capable of finding the correct mappings between the agent's ontologies. This problem was not addressed by the FIPA Ontology Service Specification [1] which was only intended to be a *specification* of an ontology agent. Most implementations that make use of ontology agents (e.g. KRAFT [25]), or mediation services (e.g. OBSERVER [21]) assume that the mappings between ontologies are established manually. For large open MAS's, this is not an option because inter ontology mappings have to be established on such a large scale that human involvement in this task is not feasible. For an ontology agent to perform this task automatically, it would be a very resource-consuming task as the agent should be capable of generating mappings between *every* ontology in the system.

These problems have led agent researchers to investigate even more flexible approaches for the semantic integration problem. Recently, a few approaches appeared in literature which tackled the problem in a fully *decentralized* way. There is no central coordinating ontology agent; the agents solve their communication problems at interaction time by exchanging parts of their ontologies. W. Truszkowski and S. Bailin have coined the term *Ontology Negotiation* to refer to such approaches [3]. In their paper, the authors present a communication mechanism which enables agents to exchange parts of their ontology in a pattern of successive clarifications. The DOGGIE approach [38] makes similar assumptions, but focuses mainly on the machine learning aspects of ontology exchange, namely the problem of teaching the meaning of a concept to another agent. Another approach is proposed by Soh and Chen [28], where agents exchange ontological knowledge when they believe it would improve operational efficiency.

Whereas ontology negotiation is a promising approach, it is also regarded as the most ambitious approach [35]. This is mainly because it requires fully automatic ontology matching and because the agents should be able to detect when their ontologies are insufficiently aligned for successful communication to proceed. Related work on fully automatic ontology matching has been conducted for over ten



years, starting with automatic database schema matching techniques [26]. Increased interest in ontologies has given rise to matching techniques that are specialized in the rich representations of ontologies [23]. Research on the detection of ontology mismatches in agent systems is reported in [4]. Ontology negotiation follows an integrated approach which combines agent communication protocols, automatic ontology matching techniques and automatic detection of ontology mismatches. This combination raises many questions which are not satisfactorily answered by the individual areas of research. The purpose of this paper is to clarify the goals and requirements of ontology negotiation and to propose some implementations according to these requirements.

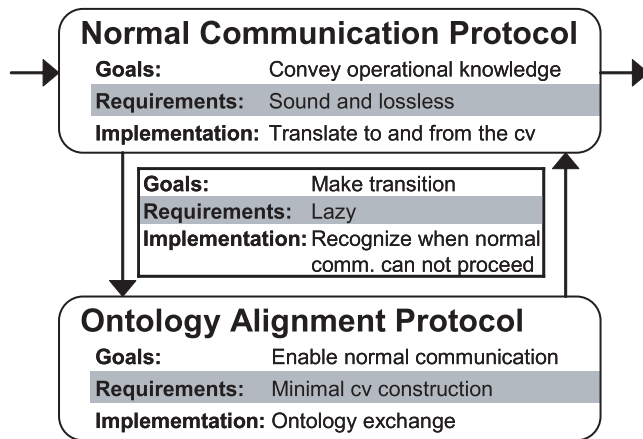


Figure 1 Overview of the ontology negotiation protocol

As depicted in Figure 1, communication involving ontology negotiation is established by two protocols: the normal communication protocol and the ontology alignment protocol. The goal of the normal communication protocol is to convey assertional knowledge, i.e. knowledge relevant to a particular problem or task. Normal communication proceeds by making use of an intermediate shared ontology which indirectly aligns the agent’s local ontologies (cf. *interlingua* [36]). Because the intermediate ontology is used only for communication purposes, we refer to it as communication vocabulary (cv). The agents participate in normal communication by translating to and from the communication vocabulary. When the cv insufficiently aligns the agent’s local ontologies to enable normal communication, the agents make a transition to the ontology alignment protocol. This protocol aims at enabling normal communication by adding concepts to the communication vocabulary. This is done using ontology exchange: one agent teaches ontological information to the other agent. After the agents have solved the problem, they return to the normal communication protocol.

There are different ways to realize a communication mechanism of the type described above. To be able to judge the quality of an ontology negotiation protocol, we will characterize the requirements for normal communication, ontology alignment and the transition between them. We will do so by drawing on the general framework of *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 [27]. 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 semantic integration techniques, *ontology negotiation* is an appropriate name to characterize this approach. There is no central coordinating entity which 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. Similarly 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 can maintain that policy. The negotiation protocol serves to resolve that issue. Negotiation protocols also have other characteristics, such as *efficiency* and *simplicity* [27]. These characteristics are usually ignored by ontology negotiation protocols (such as [3] and [38]).

Efficiency states that the agents should not waste resources in coming to an agreement. This is an important issue in ontology negotiation, as agents should efficiently establish their communication vocabulary. As agents negotiate the cv they want it to be sufficiently large such that they are capable of conveying what they want to convey. Agents may easily achieve this by adding every concept in their local ontology to the communication vocabulary. However, this would require agents to learn more concepts than necessary which would be a waste of resources. In an open system, it would lead to a forever growing ontology, which would burden the agents with large and slowly processable ontologies, and which would be difficult to learn for newcomers. Therefore, in the ontology alignment protocol, agents should come to agree on a cv that is an acceptable solution to the communication problem and which is somehow *minimal* in size. We call this requirement *minimal cv construction* (see Ontology Alignment Protocol in Figure 1).

This requires us to define what an *acceptable solution* is. Negotiation protocols often use formal game theoretic criteria to qualify an agreement as acceptable. For ontology negotiation, we will define a formal criterion of *sound and lossless communication* (see Normal Communication Protocol in Figure 1). Sound communication concerns the *quality of information exchange*, i.e. the receiver's interpretation of the message should follow from what the sender intended to convey. Communication is subjectively lossless (or lossless for short), if, from the perspective of the receiver, no information is lost in the process of translating to and from the cv [9]. The lossless criterium concerns the *quantity of information exchange*.

Another characteristic of negotiation protocols is *simplicity*, stating that it should impose low computational and bandwidth demands on the agents. This is also a relevant issue for ontology negotiation as concept learning is a computationally expensive process. We therefore enable the agents to learn concepts from each other on an as-need basis. This way, they find local solutions to communication problems at the time they arise. Each time they teach concepts to each other, they incrementally generate a solution for their semantic integration problem. We call this *lazy ontology alignment* (see the transition in Figure 1). This issue regards the transition between the normal communication protocol and the ontology alignment protocol.



This paper adopts a formal perspective on ontology negotiation to precisely define goals and requirements and to give solid proofs that the proposed implementations actually possess the desirable properties. Furthermore, a formal treatment of ontology negotiation is needed to clarify the relations with research on formal ontologies (such as description logics). Readers interested in a practical evaluation of our work are referred to [10] which presents an application that is based on the framework introduced in this paper.

The next section is about the requirements of ontologies and communication. It presents a formal underpinning that is needed to precisely define the requirements of ontology negotiation. Section 3 is about implementation of ontologies. It shows how ontologies can be implemented using description logics and concept classifiers such that their requirements are fulfilled. Section 4 is about the implementation of communication. It presents three ontology negotiation protocols. We evaluate these protocols according to the criteria of sound and lossless communication, laziness and minimal cv construction. We conclude and give directions for future research in section 5.

2 Conceptual framework

In this paper, we restrict ourselves to dialogues between two agents. As a running example, we use a travel-agent which assists a customer in planning a holiday trip to the United States (a scenario envisioned in [19]). The travel agent performs services such as finding a cheap flight, investigating prices for car rental, suggesting other transport possibilities, and finding out which licenses are required for campsites on the way. We consider communication between a travel-agent α_1 with ontology \mathcal{O}_1 and a car rental service α_2 with ontology \mathcal{O}_2 . \mathcal{O}_1 shows the expertise of α_1 on different sorts of accommodation and \mathcal{O}_2 shows the expertise of α_2 on cars. Besides \mathcal{O}_1 and \mathcal{O}_2 , also other ontologies can be distinguished in this system: the communication vocabulary (\mathcal{O}_{cv}), the mapping between α_1 's ontology and the cv (\mathcal{O}_{1-cv}), the mapping between α_2 's ontology and the cv (\mathcal{O}_{2-cv}), and the ontology that would arise if we would combine \mathcal{O}_1 and \mathcal{O}_2 ($\mathcal{O}_{1.2}$). Figure 2 shows these six ontologies in the initial situation when the cv is empty. The diagrams represent 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 \mathcal{O}_x to \mathcal{O}_y represents that \mathcal{O}_y is included in ontology \mathcal{O}_x . The situation after α_2 has added the concept *roadvehicle* to the cv is shown in Figure 3. This figure shows that *roadvehicle* has been added to \mathcal{O}_{cv} , which enables this concept to be used in communication. The figure also shows that α_1 has learned the meaning of *roadvehicle* by representing the relations between *roadvehicle* and the concepts in its local ontology in \mathcal{O}_{1-cv} .

The next section explains the ontologies in the system in further depth. Section 2.2 explains the agent's knowledge on these ontologies as well as their dynamics. Section 2.3 explains how normal communication is established in this framework and formalizes sound and lossless communication.

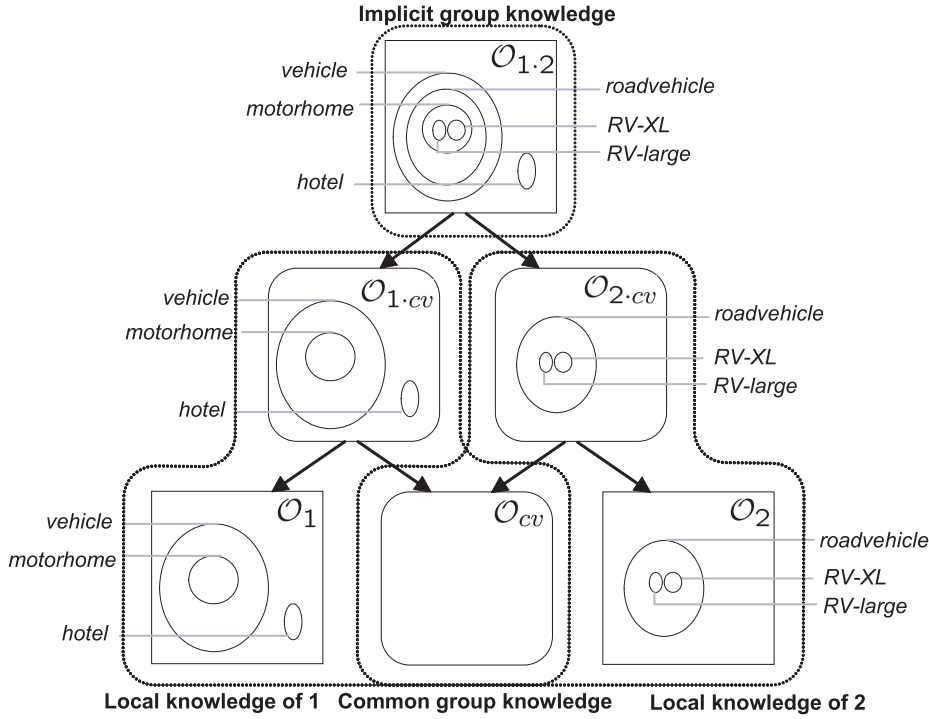


Figure 2 Ontologies of agents in the initial situation

2.1 *Ontologies*

The objects that exist in the world that the agent inhabits are given by the universe of discourse (Δ). The names of the elements in Δ are given by the set IND. For an agent to be able to store knowledge about Δ , it needs a conceptualization (ρ). We will focus on conceptualizations that consist of sets of individuals, i.e. $\rho \subseteq 2^\Delta$. Furthermore, as commonly done in AI systems ([29]), we assume that the elements in ρ form a bounded lattice structure by considering the partial ordered set (ρ, \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 $x \cup y \in \rho$. Note that, at this level, the elements in the conceptualization are not yet named. Rather, the conceptualization contains the *meanings* which the agent uses to represent knowledge about its environment. In the example Figure, the elements in ρ are represented by circles. There is no general way to decide what constitutes a good conceptualization as this depends on the agent’s task. In our framework, different agents are allowed to adopt different conceptualizations which best suit their needs.

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 [14]. The ontology introduces a set of symbols \mathcal{C} which, when interpreted under their intended interpretation, refer to the elements in the conceptualization (conforming to the treatment by Genesereth and Nilsson

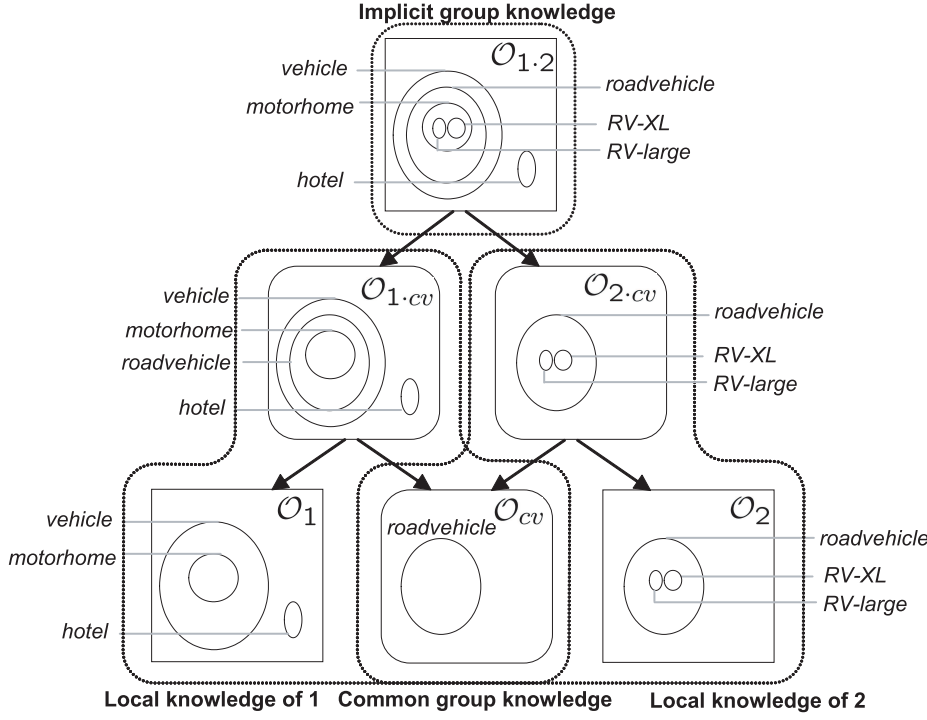


Figure 3 Ontologies of agents that share one concept

in [12]). We will refer to the intended interpretation function with $\mathcal{I}^{INT} : \mathcal{C} \rightarrow \rho$. \mathcal{I}^{INT} is a surjective function. This means that for every element $y \in \rho$, an element $x \in \mathcal{C}$ 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 $\mathcal{O} = \langle \mathcal{C}, \leq \rangle$ where $\leq \subseteq \mathcal{C} \times \mathcal{C}$ is a pre-order for which $\forall x, y \in \mathcal{C}. x \leq y \Leftrightarrow \mathcal{I}^{INT}(x) \subseteq \mathcal{I}^{INT}(y)$. 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 (e.g. synonyms). If two elements $x, y \in \mathcal{C}$ 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 1 Given the following ontologies:

- $\mathcal{O}_i = \langle \mathcal{C}_i, \leq_i \rangle$ (for $i \in \{1, 2\}$): The local ontology of α_i .

- $\mathcal{O}_{cv} = \langle \mathcal{C}_{cv}, \leq_{cv} \rangle$: The communication vocabulary of the agents, where $\mathcal{C}_{cv} \subseteq \mathcal{C}_1 \cup \mathcal{C}_2$.

We define the ontologies:

- $\mathcal{O}_{1.2} = \langle \mathcal{C}_{1.2}, \leq_{1.2} \rangle$: A god's eye view over the ontologies in the system, where $\mathcal{C}_{1.2} = \mathcal{C}_1 \cup \mathcal{C}_2$. $\leq_{1.2}$ conforms to the subset ordering on the intended interpretations of the elements in $\mathcal{C}_{1.2}$.
- $\mathcal{O}_{i-cv} = \langle \mathcal{C}_{i-cv}, \leq_{i-cv} \rangle$ (for $i \in \{1, 2\}$): The local ontology of α_i and the communication vocabulary, where $\mathcal{C}_{i-cv} = \mathcal{C}_i \cup \mathcal{C}_{cv}$. \leq_{i-cv} conforms to the subset ordering on the intended interpretations of the elements in \mathcal{C}_{i-cv} .

We use a subscript notation whenever we need to stress that something belongs to \mathcal{O}_i , \mathcal{O}_{cv} , etc. For example, a concept with the name d_{cv} is assumed to be member of \mathcal{C}_{cv} . The subscripts are omitted when no confusion arises. If a concept occurs in multiple ontologies, the intended interpretation of this concept is assumed to be unvarying in each of these ontologies. The following definition introduces some useful terminology:

Definition 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 1 in further depth.

Local ontologies

The local ontology introduces a vocabulary which the agent uses to represent and reason with assertional knowledge. The local ontology itself does not represent knowledge that is of practical use to the agent. Rather, it enables the agent to store its assertional knowledge in an efficient and useful way. An agent α_i stores this knowledge in its assertional knowledge base \mathcal{A}_i which consists of a set of membership statements that state which individuals belong to which concepts. A membership statement in \mathcal{A}_i is of the form $c(a)$, where $c \in \mathcal{C}_i$ and $a \in \text{IND}$. We avoid naming conflicts between two local ontologies by assuming that the sets \mathcal{C}_1 and \mathcal{C}_2 are disjoint. This can be easily achieved by prefixing the concept names using namespaces.

God's eye view ontology

$\mathcal{O}_{1.2}$ is the ontology that would arise if the local ontologies of the agents were combined. $\mathcal{O}_{1.2}$ 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 1,



it follows that every other ontology in the system is included in this ontology, i.e. $\mathcal{C}_i, \mathcal{C}_{cv}, \mathcal{C}_{i-cv} \subseteq \mathcal{C}_{1.2}$. Note that $\leq_{1.2}$ is not *equal* to $\leq_1 \cup \leq_2$, but that it is a *superset* (except in the hypothetical case when \mathcal{C}_1 or \mathcal{C}_2 is empty). This is because, as argued before, $\leq_{1.2}$ is conforming to the subset ordering on the intended interpretations of the concepts. It therefore also contains the relations between the elements of \mathcal{C}_1 and \mathcal{C}_2 which are not present in $\leq_1 \cup \leq_2$. For example, $\leq_{1.2}$ contains the relation *roadvehicle* \leq *vehicle*, which is present in neither \leq_1 , nor \leq_2 . In the following, when we speak of a sub- or superconcept of another concept, we do so with regard to $\mathcal{O}_{1.2}$.

Ontologies for alignment

The communication vocabulary \mathcal{O}_{cv} indirectly aligns the agent's local ontologies. α_1 and α_2 maintain a mapping from their local ontologies to the communication vocabulary in respectively \mathcal{O}_{1-cv} and \mathcal{O}_{2-cv} . This mapping states the relation between concepts in the communication vocabulary and concepts in the local ontology. It can be represented that a concept in the cv is *equivalent* to a concept in the local ontology, or that it is a *subconcept*, or a *superconcept*. From definition 1, it follows that $\mathcal{C}_{cv} = \mathcal{C}_{1-cv} \cap \mathcal{C}_{2-cv}$. By adopting the ontology \mathcal{O}_{i-cv} to define mappings between the communication vocabulary and the local ontology of α_i , we avoid the introduction of special mapping operators (proposed in [32], [5]).

2.2 Knowledge and dynamics

Knowledge distribution

Not every ontology is known by the agents. For example, \mathcal{O}_2 is unknown to α_1 (the agents do not have access to each other's local ontologies). \mathcal{O}_{cv} on the other hand, is known by both agents, whereas $\mathcal{O}_{1.2}$ is neither known by α_1 nor α_2 . We distinguish between local knowledge, common knowledge and implicit group knowledge [22]. 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 group 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 1

1. \mathcal{O}_i is local knowledge of α_i
2. \mathcal{O}_{cv} is common knowledge of α_1 and α_2
3. \mathcal{O}_{i-cv} is local knowledge of α_i
4. $\mathcal{O}_{1.2}$ is implicit group knowledge of α_1 and α_2

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

The assumption that \mathcal{O}_{cv} is common knowledge makes this ontology appropriate



for communication. The assumption that $\mathcal{O}_{1.2}$ is implicit group knowledge opens up the possibility for automatic ontology alignment. This is a necessary condition for any system where the agents must learn to share meaning. Two agents can not learn something from each other which was not already implicitly present beforehand.

Dynamics

Another aspect of ontologies is whether they are *static* or *dynamic* [16]. Static ontologies do not change over time, whereas dynamic ontologies may change over time.

Assumption 2

- \mathcal{O}_i and $\mathcal{O}_{1.2}$ are static ontologies.
- \mathcal{O}_{cv} and $\mathcal{O}_{i.cv}$ are dynamic ontologies.

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

Changing an agent's local ontology can not 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 local ontology. When the local 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 local ontologies \mathcal{O}_1 and \mathcal{O}_2 are static. As a consequence $\mathcal{O}_{1.2}$ is also a static ontology. \mathcal{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 \mathcal{O}_{cv} . In fact, it makes \mathcal{O}_{cv} suitable as an alignment ontology as it enables agents to add concepts to it at runtime. As a consequence, the ontologies $\mathcal{O}_{1.cv}$ and $\mathcal{O}_{2.cv}$ are also dynamic. Remember that $\mathcal{C}_{cv}, \mathcal{C}_{i.cv} \subseteq \mathcal{C}_{1.2}$. This imposes a restriction on which kind of changes may occur in \mathcal{O}_{cv} and $\mathcal{O}_{i.cv}$, i.e. only concepts from $\mathcal{C}_{1.2}$ may be added or removed from them.

Concept Learning

We can now specify, from a conceptual viewpoint, how ontology exchange affects the ontologies in the system. As is apparent in Figure 2 and 3, when α_2 adds the concept *roadvehicle* to the communication vocabulary, the concept is added to \mathcal{C}_{cv} , and becomes common knowledge. Definition 1 states that, as a consequence from this change of \mathcal{O}_{cv} , *roadvehicle* also becomes part of $\mathcal{C}_{1.cv}$. Consequently, the relation $\leq_{1.cv}$ is extended with the information that *motorhome* \leq *roadvehicle* \leq *vehicle*. As has been argued before, the static ontologies \mathcal{O}_1 , \mathcal{O}_2 and $\mathcal{O}_{1.2}$ remain unaffected.

2.3 Communication

In this framework, the sender makes itself understandable to the receiver by translating the message stated in terms of its local 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 local ontology.

For example, consider the ontologies in Figure 3. Suppose that α_1 intends to convey the message that individual a is a *motorhome*. It translates message *motorhome(a)* (stated in terms of \mathcal{O}_1) to *roadvehicle(a)* (stated in terms of \mathcal{O}_{cv}). α_2 receives this message and translates it to its local ontology \mathcal{O}_2 , in this case also *roadvehicle(a)*. Generally, the following three concepts can be identified in the communication process.

Definition 3

- *The transferendum ($c_i \in \mathcal{C}_i$): what is to be conveyed. α_i (the speaker) intends to convey this concept to α_j .*
- *The transferens ($d_{cv} \in \mathcal{C}_{cv}$): what conveys. This concept functions as a vehicle to convey the transferendum to α_j .*
- *The translatum ($e_j \in \mathcal{C}_j$): what has been conveyed. α_j (the hearer) interprets the received message as this concept.*

Requirements for normal communication

Using the three concepts in the definition above, we state the following 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 4 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 \mathcal{C}_j .

An example of sound communication from α_1 to α_2 is: α_1 translates transferendum *motorhome* to transferens *roadvehicle* which α_2 “translates” to translatum *roadvehicle*. An example of non-sound communication from α_1 to α_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 *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 \mathcal{C}_j

Note that in definition 4 and 5 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. Section 4 describes how the agents can assess lossless communication using their local knowledge.

Example:

The empty communication vocabulary in the initial situation (Figure 2), does not enable the agents to losslessly communicate any local concept (except the top concept).

The communication vocabulary in Figure 3 sufficiently aligns \mathcal{O}_1 and \mathcal{O}_2 for α_1 to losslessly communicate *motorhome* to α_2 , viz. α_1 translates transferendum *motorhome* to transferens *roadvehicle* which α_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 \mathcal{C}_2 .

\mathcal{O}_{cv} does not sufficiently align \mathcal{O}_1 and \mathcal{O}_2 for α_1 to losslessly communicate *RV-XL* to α_2 . Suppose α_2 translates *RV-XL* to *roadvehicle* which α_1 translates to *vehicle*. This communication process has not been lossless because *motorhome* would have been a more specific translation of *RV-XL* to α_1 ’s ontology.

3 Operational framework

This section describes the data-structures and actions that can be used to implement the ontologies in the system. One of the most widely used ontology implementation languages is description logic [2], which we will explain next. After introducing the logic, we show how this language can be used to implement the ontologies described in definition 1 and how the requirements regarding their knowledge distribution (assumption 1) can be met. Note that the description logic implementation we will introduce here is in some respects more powerful than the conceptual framework we introduced in section 2. For example, it allows specification of disjointness relations which is not strictly required for our communication mechanisms to work.

3.1 Description Logic

A description logic knowledge base is represented as a tuple $\langle \mathcal{T}, \mathcal{A} \rangle$, containing a TBox and an ABox [2]. 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} without roles as a concept language

that is used in the TBox and the ABox.

Syntax

The syntax of description logic serves to implement the set of concept names \mathcal{C} we have discussed in section 2. Given a set of atomic concepts \mathcal{C}^a , we define the language $\mathcal{L}(\mathcal{C}^a)$ as follows:

- $\top \in \mathcal{L}(\mathcal{C}^a)$ (top concept)
- $\perp \in \mathcal{L}(\mathcal{C}^a)$ (bottom concept)
- $c \in \mathcal{C}^a \rightarrow c \in \mathcal{L}(\mathcal{C}^a)$ (atomic concept)
- $c \in \mathcal{L}(\mathcal{C}^a) \rightarrow \neg c \in \mathcal{L}(\mathcal{C}^a)$ (negation)
- $c, d \in \mathcal{L}(\mathcal{C}^a) \rightarrow c \sqcap d \in \mathcal{L}(\mathcal{C}^a)$ (intersection)
- $c, d \in \mathcal{L}(\mathcal{C}^a) \rightarrow c \sqcup d \in \mathcal{L}(\mathcal{C}^a)$ (union)

The set of concepts in the ontology is given by $\mathcal{C} = \mathcal{L}(\mathcal{C}^a)$. Given two concepts $c, d \in \mathcal{C}$, a terminological axiom takes the form of $c \sqsubseteq d$. Given a concept $c \in \mathcal{C}$ and an individual $a \in \text{IND}$, a membership statement is of the form $c(a)$.

Semantics

Semantics of the concept language is defined by an interpretation function \mathcal{I} which maps every individual $a \in \text{IND}$ to an element in Δ and every atomic concept $c \in \mathcal{C}^a$ to a subset of Δ . The interpretation function is extended to non-atomic concepts as follows:

- $\mathcal{I}(\top) = \Delta$
- $\mathcal{I}(\perp) = \emptyset$
- $\mathcal{I}(\neg c) = \Delta \setminus \mathcal{I}(c)$
- $\mathcal{I}(c \sqcap d) = \mathcal{I}(c) \cap \mathcal{I}(d)$
- $\mathcal{I}(c \sqcup d) = \mathcal{I}(c) \cup \mathcal{I}(d)$

An interpretation \mathcal{I} satisfies a terminological axiom $c \sqsubseteq d$, 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 $\models_{\mathcal{I}} \Gamma'$. Given a TBox, the relation \sqsubseteq can be computed efficiently using standard DL reasoning techniques.

The semantics of membership statements is defined as: $\models_{\mathcal{I}} c(a)$ iff $\mathcal{I}(a) \in \mathcal{I}(c)$. We assume that the ABox is sound w.r.t. 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 unnecessary 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.

3.2 Implementing local and common knowledge

Local knowledge of α_i over \mathcal{O}_i and $\mathcal{O}_{i,cv}$ (assumption 1.1 and 1.3) can be straightforwardly established using two TBoxes: \mathcal{T}_i and $\mathcal{T}_{i,cv}$. Common knowledge over \mathcal{O}_{cv} (assumption 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, \mathcal{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 agent's knowledge over the ontologies.

Property 1

1. For $i \in \{1, 2\}$, for all $c, d \in \mathcal{C}_i$: $\mathcal{T}_i \models c \sqsubseteq d$ iff $c \leq d$.
2. For $i \in \{1, 2\}$, for all $c, d \in \mathcal{C}_{i,cv}$: $\mathcal{T}_{i,cv} \models c \sqsubseteq d$ iff $c \leq d$.
3. For all $c, d \in \mathcal{C}_{cv}$: $\mathcal{T}_{cv} \models c \sqsubseteq d$ 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 local ontology. The second and third item of the property concern dynamic ontologies, and should be fulfilled by the ontology alignment protocol.

Example

Consider the ontologies introduced in Figure 3. We show how these ontologies can be implemented such that property 1 is fulfilled.

The following TBoxes are possessed by α_1

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

The TBoxes that are possessed by α_2 are:

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

3.3 Implementing implicit group knowledge

Until now, we have described how the first three items of assumption 1 are implemented using common techniques available from description logic research. The fourth item of the assumption is not yet met, i.e. $\mathcal{O}_{1,2}$ is (not even) implicit group knowledge. The data structures as described until now do not give rise to implicit group knowledge of the relations between two different agent's local concepts. For example, $\mathcal{T}_{1,cv} \cup \mathcal{T}_{2,cv}$ does not specify the relations between α_1 's concept *motorhome* and α_2 's concept *RV-large*. This relation must be (at least) implicit group knowledge, otherwise the agents are not capable of retrieving it. Therefore, we assume that the agents know more about their local 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** $a \in \mathcal{I}^{INT}(c)$ **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 (cf. Luc Steels' approach to language creation [30]). In a scenario where the domain of discourse consists of text corpora, the action **Classify** can be implemented using text classification techniques [17].

4 Communication

Before we propose protocols for ontology negotiation in section 4.2, we will discuss how ontology exchange can be implemented in our framework (the lower layer in Figure 1). 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, the instruction $\text{send}(\alpha_j, \langle \text{topic}, p_1, \dots, p_n \rangle)$ may be used to send a message, where α_j is the addressee of the message, the topic specifies what the message is about, and $p_1..p_n$ are parameters of the message. The effect of this instruction is that α_j is able to perform a $\text{Receive}(\alpha_i, \langle \text{topic}, x_1, \dots, x_n \rangle)$ action, where α_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 α_i as the sender and α_j as the receiver of messages.

4.1 Implementation of concept learning

Concept learning, or automatic ontology matching, is a widely studied issue in computer science [7, 38, 23]. It is not our intention to contribute to this type of research, as the focus of this paper is on the combination of such techniques with normal agent communication protocols. Therefore, we will adopt a simple but adequate concept learning technique that is correct with respect to the theoretical framework introduced in this paper. In particular, it matches well with the semantics of description logics. Despite its simplicity, we have successfully applied this technique in the domain of internet-news in the ANEMONE system [10]. Nevertheless, we stress that this implementation of concept learning should be viewed as one possibility and that it can be replaced by other approaches, depending on the chosen domain (see [11] for an extensive survey on other possibilities).

For this approach to work, we require that the agents have access to the same elements in the universe of discourse (Δ), and use the same signs to refer to these individuals (given by the set IND). These requirements are readily met in the ANEMONE system, where every agent has access to IP-addresses (i.e. Δ is the set of IP-addresses), and every agent uses URL's to refer to these addresses (i.e. IND

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 fulfills its task in ANEMONE.

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 2.2. To realize these effects, α_j 's TBoxes $\mathcal{T}_{j.cv}$ and \mathcal{T}_{cv} must be updated such that property 1.2 and 1.3 hold. To realize the effects regarding \mathcal{T}_{cv} , α_i must communicate to α_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(α_j, c)

Let mss be most specific in the set of superconcepts of c in \mathcal{C}_{cv} and mgs be most general in the set of subconcepts of c in \mathcal{C}_{cv}

- add $c \sqsubseteq mss$ and $mgs \sqsubseteq c$ to \mathcal{T}_{cv}
- send ($\alpha_j, \langle \text{boundaries}, c, mss, mgs \rangle$)

Action Receive($\langle \text{boundaries}, c, mss, mgs \rangle$)

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

Realizing the effects of concept learning on $\mathcal{O}_{j.cv}$ is more difficult to establish because neither α_i nor α_j has explicitly represented these relations in a TBox. For example, consider the ontologies in Figure 2 and 3. In the initial situation, neither of the agents has local knowledge that $motorhome \leq roadvehicle \leq vehicle$. Hence this information must be conveyed differently. α_i conveys this information to α_j by sending a set of positive and negative examples of concept c . Upon receiving these examples, α_j uses inductive inference to derive the relations of c with the concepts in its local 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(α_j, c)

- send ($\alpha_j, \langle \text{explication}, c, \{p | \mathcal{I}(p) \in \mathcal{I}^{INT}(c)\}, \{n | \mathcal{I}(n) \notin \mathcal{I}^{INT}(c)\} \rangle$)

Action Receive($\langle \text{explication}, c, P, N \rangle$)

- add $c \sqsubseteq d_j$ to $\mathcal{T}_{j.cv}$, 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 $\mathcal{T}_{j.cv}$, 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 α_j to derive every relation of c with the concepts in \mathcal{C}_j .

Given that the agents' classifiers are free of errors, as stated in the specification of **Classify**, the **SendBoundaries** and **Explicate** actions are sufficient to convey the meaning of a concept to another agent. An agent that adds an atomic concept

	Sound and lossless	Lazy	Minimal cv
P1	+	-	-
P2	+	+	-
P3	+	+	+

Figure 4 Evaluation of the protocols

c to \mathcal{C}_{cv}^a , may introduce more than one concept in \mathcal{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 \mathcal{C}_{cv} after an atomic concept c is added is given by $\mathcal{L}(\mathcal{C}_{cv}^a \cup \{c\}) \setminus \mathcal{L}(\mathcal{C}_{cv}^a \setminus \{c\})$. The sending agent α_i conveys the meanings of these concepts that are also in \mathcal{C}_i . We can now define the action **AddConcept** as follows:

Action AddConcept(α_j, c)

- For all $d \in (\mathcal{L}(\mathcal{C}_{cv}^a \cup \{c\}) \setminus \mathcal{L}(\mathcal{C}_{cv}^a \setminus \{c\})) \cap \mathcal{C}_i$:
 - **SendBoundaries**(α_j, d)
 - **Explicate**(α_j, d)

4.2 Ontology negotiation protocols

In this section, we will propose three ontology negotiation protocols of the type depicted in Figure 1. 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 (Figure 4).

In section 2.3 we defined successful communication as being sound and lossless. 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.

Protocol 1

We begin with a very simple protocol. In protocol 1, normal communication is implemented by translating the transferendum (in the sender's local ontology) to an equivalent transferens (in the communication vocabulary). The receiver translates the transferens to the most specific superconcept in its local ontology, the *translatum*. This is done by the **InformExact** action. If there is no transferens available in the communication vocabulary that is equivalent to the transferendum, the speaker decides that normal communication can not proceed, and adds the transferendum to the communication vocabulary.

Action InformExact($\alpha_j, c_i(a)$)

if $\exists d_{cv}. d_{cv} \equiv c_i$ **then** **send**($\alpha_j, \langle \text{InformExact}, d_{cv}(a) \rangle$)

Action Receive($\alpha_i, \langle \text{InformExact}, d_{cv}(a) \rangle$)

Add $e_j(a)$ to \mathcal{A}_j , where e_j is most specific in the set of superconcepts of d_{cv} in \mathcal{C}_j

When the condition in the if statement of **InformExact** is not met, the agent

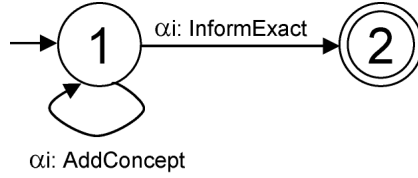


Figure 5 Protocol P1

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. The event that is triggered upon receiving an **InformExact** message, produces a translatum e_j which is most specific in the set of superconcepts of d_{cv} in \mathcal{C}_j . Because the action that produces an **InformExact** 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 \mathcal{C}_j , thereby meeting the lossless requirement.

Example:

Consider the initial situation in Figure 2, where the agents have not yet taught concepts to each other. Suppose α_2 intends to convey the assertion $roadvehicle(a)$ to α_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 \perp .

α_2 : **AddConcept**($\alpha_1, roadvehicle$)
 $\perp \alpha_2$: add $roadvehicle \sqsubseteq \top$ to \mathcal{T}_{cv}
 $\perp \alpha_1$: add $motorhome \sqsubseteq roadvehicle \sqsubseteq vehicle$ to $\mathcal{T}_{1.cv}$
 α_2 : **InformExact**($\alpha_1, roadvehicle(a)$)
 $\perp \alpha_2$: send($\alpha_1, \langle InformExact, roadvehicle(a) \rangle$)
 α_1 : **receive**($\alpha_2, \langle InformExact, roadvehicle(a) \rangle$)
 $\perp \alpha_1$: add $vehicle(a)$ to \mathcal{A}_1

This conversation has given rise to a communication vocabulary as in Figure 3. Now, suppose that α_1 intends to convey the message $hotel(a)$ and that the cv contains the concept $roadvehicle$ (as in Figure 3). The agents perform the following actions:

α_1 : **AddConcept**($\alpha_2, hotel$)
 $\perp \alpha_1$: add $hotel \sqsubseteq \neg roadvehicle$ to \mathcal{T}_{cv}
 $\perp \alpha_2$: add $hotel \sqsubseteq \neg roadvehicle$ to $\mathcal{T}_{2.cv}$
 α_1 : **InformExact**($\alpha_2, hotel(a)$)
 $\perp \alpha_1$: send($\alpha_2, \langle InformExact, hotel(a) \rangle$)
 α_2 : **receive**($\alpha_1, \langle InformExact, hotel(a) \rangle$)
 $\perp \alpha_2$: add $\neg roadvehicle(a)$ to \mathcal{A}_2

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

Although P1 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 $\neg roadvehicle$. If α_1 would have translated $hotel$ to the superconcept $\neg roadvehicle$, then α_2 could have interpreted this as $\neg roadvehicle$, and

this would have been sound and lossless communication. However, this dialogue is not allowed by P1. Using P1, 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 protocol 1 is not satisfactory w.r.t. minimal cv construction and laziness (cf. Figure 4). The following protocol attempts to overcome these problems.

Protocol 2

In protocol 2, the sender uses the **InformExact** action when allowed. When this is not allowed, i.e. 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($\alpha_j, c_i(a)$)

send($\alpha_j, \langle \text{Inform}, d_{cv}(a) \rangle$) where d_{cv} is most specific in the set of superconcepts of c_i in \mathcal{C}_{cv}

The **Receive** action that is triggered by an Inform message is equal to the **Receive** action that is triggered when an InformExact 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 can not directly check definition 5 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* [13]. In protocol 2, it works as follows: consider the ontologies \mathcal{O}_1 and \mathcal{O}_2 from Figure 2, and suppose that $\mathcal{C}_{cv}^a = \{\text{vehicle}, \text{motorhome}\}$. Suppose that the transferendum is α_2 's concept *roadvehicle* and that α_2 uses *vehicle* as a transferens. Upon receiving this message, α_1 knows that α_2 did *not* intend to convey the following subconcepts in \mathcal{C}_{cv} : *motorhome* and *vehicle* \sqcap \neg *motorhome*. This is because otherwise α_2 should have used these more specific concepts in the message. Knowing that the transferendum is more general than these concepts, α_1 knows that communication has been lossless.

In protocol 2, the receiver α_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 \mathcal{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 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 \mathcal{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 α_i

adds the transferendum to cv. The OK action can only be done if the receiver can

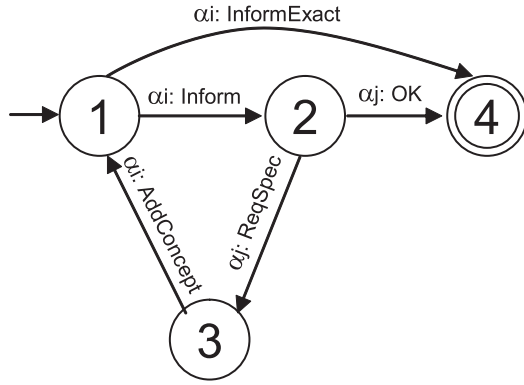


Figure 6 Protocol P2

assess that communication was lossless.

Action $OK(\alpha_i)$

Responding to $\langle \text{Inform}, (d_{cv}(a)) \rangle$

Let D *be the set of concepts that are most general among the set of strict subconcepts of* d_{cv} *in* \mathcal{C}_{cv}

Let e_j *be most specific in the set of superconcepts of* d_{cv} *in* \mathcal{C}_j *(* e_j *is the translatum)*

if every strict subconcept of e_j in \mathcal{C}_j is a subconcept of any of the concepts in D

then $\text{send}(\alpha_i, \langle \text{OK} \rangle)$

Example: Consider the ontologies in Figure 3 and suppose that α_1 wishes to communicate $hotel(a)$ (as in the last example of P1). The dialogue proceeds as follows:

α_1 : **Inform**($\alpha_2, hotel(a)$)

$\perp \alpha_1$: $\text{send}(\alpha_1, \langle \text{Inform}, \neg roadvehicle(a) \rangle)$

α_2 : **Receive** ($\alpha_1, \langle \text{Inform}, \neg roadvehicle(a) \rangle$)

$\perp \alpha_2$: add $\neg roadvehicle(a)$ to \mathcal{A}_2

α_2 : **OK**

In this example, α_2 responded with OK, because in \mathcal{O}_2 the information provided by $\neg roadvehicle$ is as specific as possible.

Now consider the ontologies in Figure 3 with a different cv. Suppose that α_2 wishes to communicate $RV\text{-large}(a)$, and that $\mathcal{C}_{cv}^a = \{motorhome\}$. The dialogue proceeds as follows:

α_2 : **Inform**($\alpha_1, RV\text{-large}(a)$)

$\perp \alpha_2$: $\text{send}(\alpha_1, \langle \text{Inform}, motorhome(a) \rangle)$

α_1 : **Receive** ($\alpha_2, \langle \text{Inform}, motorhome(a) \rangle$)

$\perp \alpha_1$: add $motorhome(a)$ to \mathcal{A}_1

α_1 : **OK**

In this example, α_1 responded with OK, because in \mathcal{O}_1 the information provided by $motorhome$ is as specific as possible.

Now, suppose that α_2 wishes to communicate $RV\text{-large}(a)$, $\mathcal{C}_{cv}^a = \{vehicle\}$.

α_2 : **Inform**($\alpha_1, RV\text{-large}(a)$)

$\perp \alpha_2$: $\text{send}(\alpha_1, \langle \text{Inform}, vehicle(a) \rangle)$

α_1 : Reqspec
 α_2 : AddConcept($\alpha_1, RV\text{-large}$)
 α_2 : InformExact($\alpha_1, RV\text{-large}(a)$)
 α_1 : Receive ($\alpha_1, \langle \text{InformExact}, RV\text{-large}(a) \rangle$)
 $\perp \alpha_1$: add *motorhome*(a) to \mathcal{A}_1

In this example α_1 did not respond OK at first, because *motorhome* caused the action the fail. Hereby, α_1 correctly recognized non-lossless communication.

Now, suppose that α_2 wishes to communicate *roadvehicle*(a), and $\mathcal{C}_{cv}^a = \{vehicle, motorhome\}$

α_2 : Inform($\alpha_1, roadvehicle(a)$)
 $\perp \alpha_2$: send($\alpha_1, \langle \text{Inform}, vehicle(a) \rangle$)
 α_1 : Receive ($\alpha_2, \langle \text{Inform}, vehicle(a) \rangle$)
 $\perp \alpha_1$: add *vehicle*(a) to \mathcal{A}_1
 α_1 : OK

In this example, α_1 responded OK, because it knew that if α_2 had more information available about individual *a*, e.g. *motorhome*, it would have used a more specific term, e.g. *motorhome*. Hereby, α_1 correctly recognized lossless communication.

Theorem 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 $\{e'_j | c_i \leq_{1.2} e'_j \wedge e'_j \in \mathcal{C}_j\}$ (definition 5 does not hold). This means that either e_j is not in the set $\{e'_j | c_i \leq_{1.2} e'_j \wedge e'_j \in \mathcal{C}_j\}$ (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 $\{e'_j | c_i \leq_{1.2} e'_j \wedge e'_j \in \mathcal{C}_j\}$, 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 $\{d''_{cv} | c_i \leq_{1.2} d''_{cv} \wedge d''_{cv} \in \mathcal{C}_{cv}\}$. Therefore, option (b) is in contradiction with the condition of Inform. □

Because P2 enables the agents to communicate without learning every concept in their local ontologies from each other, this protocol scores better than P1, w.r.t. laziness (cf. Figure 4). However, the protocol may still give rise to a communication vocabulary which is unnecessarily large, as shown by the following example:

Example: Consider the initial situation in Figure 2. Suppose that α_1 intends to convey *motorhome*.

α_1 : Inform($\alpha_2, motorhome(a)$)
 $\perp \alpha_1$: send($\alpha_2, \langle \text{Inform}, \top(a) \rangle$)
 α_2 : Reqspec
 α_1 : AddConcept($\alpha_2, motorhome$)
 α_1 : InformExact($\alpha_2, motorhome(a)$)
 α_2 : Receive ($\alpha_1, \langle \text{InformExact}, motorhome(a) \rangle$)

$\perp\alpha_2$: add *roadvehicle*(*a*) to \mathcal{A}_2

After this dialogue, the cv is $\{\textit{motorhome}\}$. In the next dialogue, α_2 intends to convey *roadvehicle*. A similar dialogue follows; after this dialogue, the cv has become $\{\textit{motorhome}, \textit{roadvehicle}\}$. In the next dialogue, α_1 intends to convey *vehicle*. After this dialogue has finished, the cv has become $\{\textit{motorhome}, \textit{roadvehicle}, \textit{vehicle}\}$.

The last communication vocabulary is unnecessarily large, because $\{\textit{motorhome}, \textit{vehicle}\}$ enables the agents to losslessly communicate the same concepts as $\{\textit{motorhome}, \textit{roadvehicle}, \textit{vehicle}\}$. For this reason, P2 is not satisfactory w.r.t. minimal cv construction. The next protocol aims to overcome this problem by allowing the agents to remove superfluous concepts from their communication vocabulary.

Protocol 3 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 6 $d \in C_{cv}^a$ is *mutually redundant* if $\mathcal{L}(C_{cv}^a \setminus \{d\})$ allows α_i and α_j to losslessly communicate the same concepts to each other as $\mathcal{L}(C_{cv}^a)$.

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 7 α_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 $\mathcal{L}(C_{cv}^a \setminus \{d_{cv}\})$
 - 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 $\mathcal{L}(C_{cv}^a \setminus \{d_{cv}\})$
 - c'_i is most specific in the set of superconcepts of d_{cv} in C_i .

In this definition, the formula $\mathcal{L}(C_{cv}^a \setminus \{d_{cv}\})$ 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 $C_{cv}^a = \{\textit{motorhome}, \textit{roadvehicle}, \textit{vehicle}\}$. α_1 believes the concept *roadvehicle* to be redundant because *vehicle* satisfies the first item in the condition of definition 7, and *motorhome* satisfies the second item.

Theorem 2 *If α_i considers a concept d_{cv} redundant (according to definition 7), then concept d_{cv} is mutually redundant (according to definition 6)*

Proof: We will prove the theorem for communication from α_i to α_j and from α_j to α_i .

From α_i to α_j : Observe that α_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 7 (first bullet), d'_{cv} exists for which $c_i \leq d'_{cv} \leq d_{cv}$. Hence, α_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 7 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, α_i would also never use d_{cv} or $\neg d_{cv}$ as a conjunct or disjunct either.

From α_j to α_i : For every concept c_j which α_j communicates using d_{cv} , α_j may also use d''_{cv} . The second bullet in definition 7 ensures d''_{cv} yields the same translatum as d_{cv} , namely c'_j . Therefore, every concept c_j that is losslessly communicated using d_{cv} can also be losslessly communicated using d''_{cv} . Furthermore, α_i responds “OK” to messages with d''_{cv} (and thereby recognizes lossless communication), because bullet 1 in definition 7 ensures that all subconcepts in \mathcal{C}_i of d_{cv} are also subconcepts of d''_{cv} . A similar argument can be made for α_j that communicates using $\neg d_{cv}$.

□

Action RemoveConcept(α_j, d_{cv})

if α_i considers d_{cv} redundant **then**

- Remove d_{cv} from \mathcal{C}_{cv}^a
- send($\alpha_j, \langle \text{RemoveConcept}, d \rangle$)

else fail

Action Receive($\langle \text{RemoveConcept}, d \rangle$)

Remove d_{cv} from \mathcal{C}_{cv}^a

An agent performs a **RemoveConcept** action on a concept d_{cv} , when it considers it redundant using the criteria described in definition 7. 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 α_j is allowed to add concepts to the communication vocabulary (state 4). The second example below illustrates the application of this.

Example: Consider the ontologies in Figure 2, and that $\mathcal{C}_{cv}^a = \{motorhome, roadvehicle\}$.

Suppose that α_1 wishes to communicate $vehicle(a)$

α_1 : **Inform**($\alpha_2, vehicle(a)$)

$\perp \alpha_1$: send($\alpha_2, \langle \text{Inform}, \top(a) \rangle$)

α_2 : **Reqspec**

α_1 : **AddConcept**($\alpha_2, vehicle$)

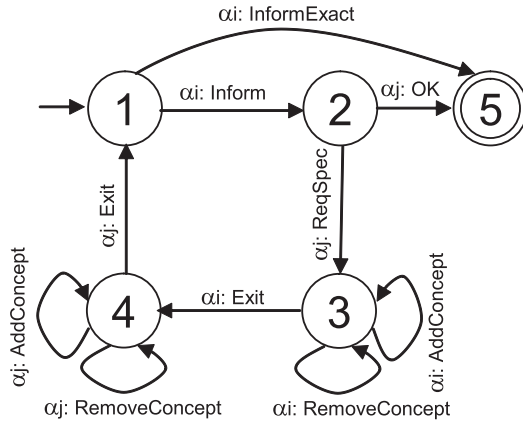


Figure 7 Protocol P3

$\alpha_1 : \text{RemoveConcept}(\alpha_1, \text{roadvehicle})$
 $\alpha_1 : \text{Exit}$
 $\alpha_2 : \text{Exit}$
 $\alpha_1 : \text{InformExact}(\alpha_2, \text{vehicle}(a))$
 \dots

In this example α_1 considers the concept *roadvehicle* redundant after *vehicle* was added to it. As a sender, α_1 would never use *roadvehicle*, and as a receiver α_1 finds *vehicle* equally informative as *roadvehicle*.

Another example: Consider the ontologies in Figure 3. Suppose that $\mathcal{C}_{c_v}^a = \{RV\text{-large}\}$ and that α_2 intends to communicate *RV-XL(a)*

$\alpha_2 : \text{Inform}(\alpha_1, RV\text{-XL}(a))$
 $\perp \alpha_2 : \text{send}(\alpha_1, \langle \text{Inform}, \neg RV\text{-large}(a) \rangle)$
 $\alpha_1 : \text{ReqsSpec}$
 $\alpha_2 : \text{AddConcept}(\alpha_1, RV\text{-XL})$
 $\alpha_2 : \text{Exit}$
 $\alpha_1 : \text{AddConcept}(\alpha_2, \text{Motorhome})$
 $\alpha_1 : \text{RemoveConcept}(\alpha_2, RV\text{-XL})$
 $\alpha_1 : \text{RemoveConcept}(\alpha_2, RV\text{-large})$
 $\alpha_1 : \text{Exit}$
 $\alpha_2 : \text{Inform}(\alpha_1, RV\text{-XL}(a))$
 $\perp \alpha_2 : \text{send}(\alpha_1, \langle \text{Inform}, \text{Motorhome}(a) \rangle)$
 $\alpha_1 : \text{OK}$

Because P3 enables the agents to remove redundant concepts from the communication vocabulary, P3 scores better w.r.t. minimal cv construction than P2. As is shown by Figure 4, this makes P3 the best protocol for ontology negotiation we have proposed in this paper.

5 Conclusion

In this paper, we have discussed the use of ontology negotiation protocols to overcome communication problems between agents with heterogeneous ontologies. We hereby formulated the *goals* and *requirements* of an ontology negotiation protocol. An ontology negotiation protocol should enable *sound and lossless* communication between the agents. The agents should build on their solution on an as-need basis, dealing with communication problems as they arise: it should be *lazy*. Furthermore, the agents should build up a relatively *small communication vocabulary* such that it remains easy to learn and to process. We have proposed three *implementations* of protocols that all give rise to sound and lossless communication. However, they were shown to differ in quality w.r.t. laziness and minimal cv construction.

We will continue this line of research by considering situations with more than two agents. This introduces some additional complexity, as not every agent knows who has taught which concepts to whom. However, we believe that also in these situations, the principles we have described in this paper will remain valid.

References and Notes

- 1 FIPA Ontology Service Specification. <http://www.fipa.org/specs/fipa00086/>.
- 2 F. Baader, D.L. McGuinness, and P.F. Patel-Schneider. *The description logic handbook: Theory, implementation and applications*. Cambridge University Press, 2003.
- 3 S.C. Bailin and W. Truszkowski. Ontology negotiation between intelligent information agents. *Knowledge Engineering Review*, 17(1):7–19, 2002.
- 4 R.J. Beun, R.M. van Eijk, and H. Prüst. Ontological feedback in multiagent systems. In *Proceedings of Third International Conference on Autonomous Agents and Multiagent Systems*, pages 110–117, New York, 2004. ACM Press.
- 5 Alex Borgida and Luciano Serafini. Distributed description logics: Directed domain correspondences in federated information sources. *Proceedings of the International description logics Workshop DL'2002*, 2002.
- 6 P Bouquet, G Kuper, M Scoz, and S Zanobini. Asking and answering semantic queries. In Paolo Bouquet and Luciano Serafini, editors, *Proceedings of the ISWC-04 workshop on Meaning Coordination and Negotiation (MCN-04)*, 2004.
- 7 M. Burnstein, D. McDermott, D.R. Smith, and S.J. Westfold. Derivation of glue code for agent interoperation. *Autonomous Agents and Multi-Agent Systems*, 6(3):265–286, 2003.
- 8 T. Bylander and B. Chandrasekaran. Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition. *International Journal of Man-Machine Studies*, 26(2):231–243, 1987.
- 9 J. van Diggelen, R.J. Beun, F. Dignum, R.M. van Eijk, and J.-J.Ch. Meyer. Optimal communication vocabularies and heterogeneous ontologies. In *Developments in Agent Communication*, LNAI 3396. Springer Verlag, 2004.
- 10 J. van Diggelen, R.J. Beun, F. Dignum, R.M. van Eijk, and J.-J.Ch. Meyer. ANEMONE: An effective minimal ontology negotiation environment. *Proceedings of the Fifth International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, 2006.

- 11 Diana Maynard et al. D2.2.3: State of the art on current alignment techniques. <http://knowledgeweb.semanticweb.org/semanticportal/servlet/download?ontology=Documentation+Ontology&concept=Deliverable&instanceSet=kweb&instance=D2.2.3+techniques&attribute=On-line+PDF+Version&value=kweb-223.pdf>, 2004.
- 12 Michael R. Genesereth and Nils J. Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc., 1987.
- 13 H. Paul Grice. Logic and conversation. *Cole, P., and J.L. Morgan, eds. Speech Acts. New York: Academic Press, 4158*, 1975.
- 14 T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.
- 15 Adil Hameed, Alun Preece, and Derek Sleeman. Ontology reconciliation. In Steffen Staab and Rudi Studer, editors, *Handbook of ontologies*, International handbooks on information systems, chapter 12, pages 231–250. Springer Verlag, Berlin (DE), 2004.
- 16 J. Heflin and J. Hendler. Dynamic ontologies on the web. In *Proceedings of American Association for Artificial Intelligence Conference (AAAI-2000)*, pages 443–449, Menlo Park, CA, USA, 2000. AAAI Press.
- 17 Peter Jackson and Isabelle Moulinier. *Natural Language Processing for Online Applications: Text retrieval, extraction, and categorization*. John Benjamins Publishing, 2002.
- 18 Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley Publishers, 1990.
- 19 M. Luck, P. McBurney, and C. Preist. Agent technology: Enabling next generation computing. *Agent link community*, 2003.
- 20 D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. The chimaera ontology environment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000)*, Austin, Texas, 2000.
- 21 E. Mena, A. Illarramendi, V. Kashyap, and A. Sheth. 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)*, 8(2):223–272, April 2000.
- 22 J-J. Ch. Meyer and W. Van Der Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge University Press, 1995.
- 23 Prasenjit Mitra, Natalya F. Noy, and Anuj R. Jaiswal. OMEN: a probabilistic ontology mapping tool. In Paolo Bouquet and Luciano Serafini, editors, *Proceedings of the ISWC-04 workshop on Meaning Coordination and Negotiation (MCN-04)*, 2004.
- 24 N. F. Noy and M. A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2000.
- 25 Alun D. Preece, Kit ying Hui, W. A. Gray, P. Marti, Trevor J. M. Bench-Capon, D. M. Jones, and Zhan Cui. The KRAFT architecture for knowledge fusion and transformation. *Knowledge Based Systems*, 13(2-3):113–120, 2000.
- 26 Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.
- 27 Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, Massachusetts, 1994.



- 28 L-K Soh and C. Chen. Balancing ontological and operational factors in refining multi-agent neighborhoods. *Proceedings of the Fourth International Conference on Autonomous Agents and Multi-Agent Systems*, 2005.
- 29 John F. Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 2000.
- 30 L. Steels. *Synthesising the Origins of Language and Meaning Using Co-evolution, Self-organisation and Level formation*. Edinburgh University Press, 1998.
- 31 H. Stuckenschmidt, F. van Harmelen, L. Serafini, P. Bouquet, and F. Giunchiglia. Using c-owl for the alignment and merging of medical ontologies. In Udo Hahn, editor, *Proceedings of the First International Workshop on Formal Biomedical Knowledge Representation (KRMed'04)*, pages 8–101, Whistler, Colorado, June 2004.
- 32 Heiner Stuckenschmidt and Ingo J. Timm. Adaption communication vocabularies using shared ontologies. *Proceedings of the Second International Workshop on Ontologies in Agent Systems (OAS)*, July 2002.
- 33 Gerd Stumme and Alexander Maedche. FCA-MERGE: Bottom-up merging of ontologies. In *IJCAI*, pages 225–234, 2001.
- 34 B. Swartout, R. Patil, K. Knight, and T. Russ. Toward distributed use of large-scale ontologies. *Proceedings of the Tenth Knowledge Acquisition for Knowledge-based Systems Workshop*, 1996.
- 35 M. Uschold and M. Gruninger. Creating semantically integrated communities on the world wide web. *Semantic Web Workshop Co-located with WWW 2002 Honolulu*, 2002.
- 36 Mike Uschold and Michael Gruninger. Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.
- 37 Gio Wiederhold and Michael Genesereth. The conceptual basis for mediation services. *IEEE Expert: Intelligent Systems and Their Applications*, 12(5):38–47, 1997.
- 38 A.B. Williams. Learning to share meaning in a multi-agent system. *Autonomous Agents and Multi-Agent Systems*, 8(2):165–193, 2004.