
Ontology negotiation: goals, requirements and implementation

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

Institute of Information and Computing Sciences

Utrecht University, the Netherlands

E-mail: jurriaan@cs.uu.nl

E-mail: rj@cs.uu.nl

E-mail: dignum@cs.uu.nl

E-mail: rogier@cs.uu.nl

E-mail: jj@cs.uu.nl

*Corresponding author

Abstract: Communication in heterogeneous Multi-Agent Systems (MASs) 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.

Keywords: Multi-Agent Systems; MASs; semantic interoperability; ontologies; ontology negotiation; description logic.

Reference to this paper should be made as follows: van Diggelen, J., Beun, R.-J., Dignum, F., van Eijk, R.M. and Meyer, J.-J. (2007) 'Ontology negotiation: goals, requirements and implementation', *Int. J. Agent-Oriented Software Engineering*, Vol. 1, No. 1, pp.63–90.

Biographical notes: Jurriaan van Diggelen studied Cognitive Artificial Intelligence. He has written his PhD thesis on achieving semantic interoperability in multi-agent systems at the Intelligence Systems Group at the Utrecht University. Since 2007, he has been doing postdoctoral work on context awareness and crisis management. His main interests are agent communication, ontologies, context awareness and the semantic web.

Robbert-Jan Beun studied Electrical Engineering at the Eindhoven University of Technology and worked for many years at the Institute for Perception Research and Tilburg University, where he combined teaching and research in the field of AI and pragmatics. Since 2000, he has been a Researcher in the Information Science Group of the Utrecht University. He is mainly interested in dialogue modelling, dialogue systems, speech act theory and non-verbal communicative behaviour.

Frank Dignum received a PhD from the Free University of Amsterdam, the Netherlands in 1989. After receiving his PhD, he has set up the Computing Science Department of the University of Swaziland. From 1993 until 2000, he worked at the Eindhoven University of Technology and started working on agents and electronic commerce. From 2000, he has been an Associate Professor at the Utrecht University in the area of agent technology. He has published numerous papers and books. He has also organised several workshops and was the local Chair of AAMAS 2005.

Rogier M. van Eijk studied Computer Science. In 2000, he received his PhD on a dissertation on programming languages for multi-agent systems. Since then he has been a Teacher and Researcher in the Information Science Group of the Utrecht University, where he started combining research on multi-agent systems and human-computer communication. His main interests are in formal and computational models of communication, dialogue systems, agent communication and embodied conversational agents.

Professor Dr. John-Jules Ch. Meyer obtained his PhD from the Vrije Universiteit in Amsterdam in 1985. Since 1993, he has been a Professor of Computer Science at the Utrecht University. At the moment, he is the Head of the Intelligent Systems Group. He is a member of the IFAAMAS board steering the international AAMAS conferences, and of the editorial boards of the *Journal of Applied Non-Classical Logics*, *Data and Knowledge Engineering* and the *Journal of Intelligent Agents & Multi-Agent Systems*. His current research interests include artificial intelligence and intelligent agents in particular. In 2005, he was appointed as a Fellow of the European Coordinating Committee for Artificial Intelligence.

1 Introduction

Most protocols that 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 realise consensus among all involved system developers on which ontology to use (Hameed *et al.*, 2004). This has motivated researchers to develop tools that assist people in creating generally shared ontologies. Chimaera (McGuinness *et al.*, 2000) and FCA-Merge (Stumme and Maedche, 2001) 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 (Lenat and Guha, 1990) and Sensus (Swartout *et al.*, 1996). 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 (Bylander and Chandrasekaran, 1987), this may be disadvantageous for the problem-solving capacity of an agent.

Ontology alignment (Noy and Musen, 2000) 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., Bouquet *et al.*, 2004; Stuckenschmidt *et al.*, 2004) 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 (MAS), it assumes that the mappings are predefined 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 (Wiederhold and Genesereth, 1997) or *ontology agents*.¹ An ontology agent provides a central that can be consulted by agents with communication problems. This approach opens the possibility to reconcile heterogeneous ontologies at agent interaction time, making it 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 (Preece *et al.*, 2000)), or mediation services (e.g., OBSERVER (Mena *et al.*, 2000)) assume that the mappings between ontologies are established manually. For large open MASs, 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.

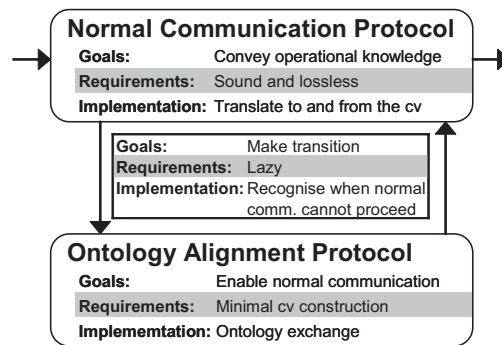
These problems have led agent researchers to investigate even more flexible approaches for the semantic integration problem. Recently, a few approaches appeared in literature, which tackled the problem in a fully *decentralised* way. There is no central coordinating ontology agent; the agents solve their communication problems at interaction time by exchanging parts of their ontologies. Truszkowski and Bailin (2002) have coined the term *Ontology Negotiation* to refer to such approaches. 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 DOGGIE approach (Williams, 2004) 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 (2005), 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 (Uschold and Gruninger, 2002). 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 (Rahm and Bernstein, 2001). Increased interest in ontologies has given rise to matching techniques that are specialised in the rich representations of ontologies (Mitra *et al.*, 2004). Research on the detection of ontology mismatches in agent systems is reported in Beun *et al.* (2004). Ontology negotiation follows an integrated approach that 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.

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* Uschold and Gruninger, 1996). 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.

Figure 1 Overview of the ontology negotiation protocol



There are different ways to realise a communication mechanism of the type described above. To be able to judge the quality of an ontology negotiation protocol, we will characterise 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 (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 semantic integration techniques, *ontology negotiation* is an appropriate name to characterise 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 can maintain that policy. The negotiation protocol serves to resolve that issue. Negotiation protocols also have other characteristics, such as *efficiency* and *simplicity* (Rosenschein and Zlotkin, 1994). These characteristics are usually ignored by ontology negotiation protocols (such as Bailin and Truszkowski, 2002; Williams, 2004).

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 (van Diggelen *et al.*, 2004). 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 van Diggelen *et al.* (2006), who present 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 that assists a customer in planning a holiday trip to the USA (a scenario envisioned in Luck *et al.* (2003)). The travel agent performs services such as finding a cheap flight, investigating prices for car rental, suggesting other transport possibilities, and finding out which licences are required for campsites on the way. We consider communication between a travel agent α_1 with ontology O_1 and a car rental service α_2 with ontology O_2 . O_1 shows the expertise of α_1 on different sorts of accommodation and O_2 shows the expertise of α_2 on cars. Besides O_1 and O_2 , other ontologies can also be distinguished in this system: the communication vocabulary (O_{cv}), the mapping between α_1 's ontology and the cv ($O_{1\cdot cv}$), the mapping between α_2 's ontology and the cv ($O_{2\cdot cv}$), and the ontology that would arise if we would combine O_1 and O_2 ($O_{1\cdot 2}$). Figure 2 shows these six ontologies in the initial situation when the cv is empty. The diagrams represent different conceptualisations (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 . 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 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 $O_{1\cdot cv}$.

Figure 2 Ontologies of agents in the initial situation

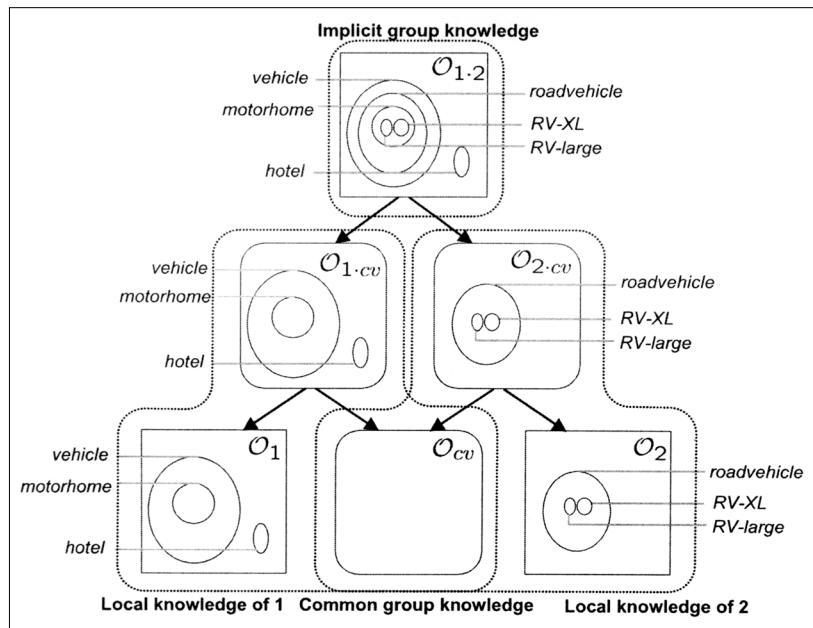
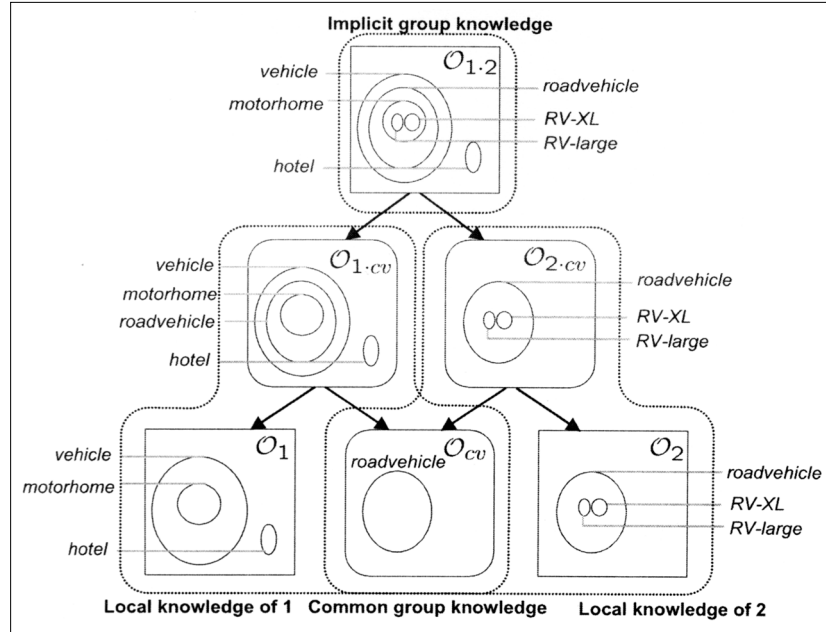


Figure 3 Ontologies of agents that share one concept


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 conceptualisation (ρ). We will focus on conceptualisations that consist of sets of individuals, *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 (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 conceptualisation are not yet named. Rather, the conceptualisation 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 conceptualisation as this depends on the agent's task. In our framework, different agents are allowed to adopt different conceptualisations that best suit their needs.

To be able to formalise knowledge about the domain of discourse, the meanings in the conceptualisation must carry a name. This is done by the ontology, which *specifies* the conceptualisation (Gruber, 1993). The ontology introduces a set of symbols C which, when interpreted under their intended interpretation, refer to the elements in the conceptualisation (conforming to the treatment by Genesereth and Nilsson (1987)). We will refer to the intended interpretation function with $I^{INT}: C \rightarrow \rho$. I^{INT} is a surjective function. This means that for every element $y \in \rho$, an element $x \in C$ exists for which

$I^{INT}(x) = y$, i.e., every element in the conceptualisation is named. In the example, I^{INT} is represented by the grey horizontal lines that connect concept names to circles (their meanings).

For an ontology to *fully* specify the conceptualisation, it would have to spell out the exact value of I^{INT} , which would be unfeasible. Therefore, the ontology only specifies that aspect of I^{INT} that 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 $\forall x, y \in C. x \leq y \Leftrightarrow I^{INT}(x) \subseteq I^{INT}(y)$. This states that an ontology specifies a conceptualisation as a reflexive, transitive relation, which conforms to the subset ordering on the intended interpretations of the concepts. Note that, although the conceptualisation has the antisymmetry property, this property does not necessarily hold for the ontology. An ontology may specify multiple ways to refer to the same element in the conceptualisation (e.g., 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 1 Given the following ontologies:

- $O_i = \langle C_i, \leq_i \rangle$ (for $i \in \{1, 2\}$) – the local ontology of α_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:

- $O_{1.2} = \langle C_{1.2}, \leq_{1.2} \rangle$ – a god's eye view over the ontologies in the system, where $C_{1.2} = C_1 \cup C_2$. $\leq_{1.2}$ conforms to the subset ordering on the intended interpretations of the elements in $C_{1.2}$.
- $O_{i-cv} = \langle C_{i-cv}, \leq_{i-cv} \rangle$ (for $i \in \{1, 2\}$) – the local ontology of α_i and the communication vocabulary, where $C_{i-cv} = C_i \cup C_{cv}$. \leq_{i-cv} conforms to the subset ordering on the intended interpretations of the elements in C_{i-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. 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 Definition 1 in further depth.

2.1.1 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 A_i , which consists of a set of membership statements that state which individuals belong to which concepts. A membership statement in A_i is of the form $c(a)$, where $c \in C_i$ and $a \in \text{IND}$. We avoid naming conflicts between two local 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.

2.1.2 God's eye view ontology

$O_{1,2}$ is the ontology that would arise if the local ontologies of the agents were combined. $O_{1,2}$ is a virtual ontology, *i.e.*, it is not materialised 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.*, $C_i, C_{cv}, C_{i,cv} \subseteq C_{1,2}$. 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 C_1 or C_2 is empty). This is because, as argued before, $\leq_{1,2}$ conforms 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, $\leq_{1,2}$ 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 $O_{1,2}$.

2.1.3 Ontologies for alignment

The communication vocabulary 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 $O_{1,cv}$ and $O_{2,cv}$, respectively. 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 $C_{cv} = C_{1,cv} \cap C_{2,cv}$. By adopting the ontology $O_{i,cv}$ to define mappings between the communication vocabulary and the local ontology of α_i , we avoid the introduction of special mapping operators (proposed in Stuckenschmidt and Timm (2002) and Borgida and Serafini (2002)).

2.2 Knowledge and dynamics

2.2.1 Knowledge distribution

Not every ontology is known by the agents. For example, O_2 is unknown to α_1 (the agents do not have access to each other's local ontologies). O_{cv} on the other hand, is known by both agents, whereas $O_{1,2}$ is neither known by α_1 nor α_2 . We distinguish between local knowledge, common knowledge and implicit group 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 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 O_i is local knowledge of α_i
- 2 O_{cv} is common knowledge of α_1 and α_2
- 3 $O_{i,cv}$ is local knowledge of α_i
- 4 $O_{1,2}$ 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 O_{cv} is common knowledge makes this ontology appropriate for communication. The assumption that $O_{1,2}$ 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 cannot learn something from each other, which was not already implicitly present beforehand.

2.2.2 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 2

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

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

Changing an agent's local 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 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 O_1 and O_2 are static. As a consequence $O_{1,2}$ 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 $O_{1,cv}$ and $O_{2,cv}$ are also dynamic. Remember that $C_{cv}, C_{i,cv} \subseteq C_{1,2}$. This imposes a restriction on which kind of changes may occur in O_{cv} and $O_{i,cv}$, *i.e.*, only concepts from $C_{1,2}$ may be added or removed from them.

2.2.3 Concept learning

We can now specify, from a conceptual viewpoint, how ontology exchange affects the ontologies in the system. As is apparent in Figures 2 and 3, when α_2 adds the concept *roadvehicle* to the communication vocabulary, the concept is added to C_{cv} , and becomes common knowledge. Definition 1 states that, as a consequence of this change of O_{cv} , *roadvehicle* also becomes part of $C_{1,cv}$. 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 O_1 , O_2 and $O_{1,2}$ 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 O_1) to *roadvehicle(a)* (stated in terms of O_{cv}). α_2 receives this message and translates it to its local ontology O_2 , in this case also *roadvehicle(a)*. Generally, the following three concepts can be identified in the communication process.

Definition 3

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

2.4 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 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 overgeneralisation, 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 C_j .

Note that in Definitions 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 O_1 and 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 *road-vehicle*. 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 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 (Baader *et al.*, 2003), 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 $\langle T, A \rangle$, containing a TBox and an ABox (Baader *et al.*, 2003). The TBox 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 A contains a set of membership statements that specify which individuals belong to which concepts; it implements the agent’s assertional knowledge. We use the description logic *ALC* without roles as a concept language that is used in the TBox and the ABox.

3.1.1 Syntax

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

- $\top \in L(C^a)$ (top concept)
- $\perp \in L(C^a)$ (bottom concept)

- $c \in C^a \rightarrow c \in L(C^a)$ (atomic concept)
- $c \in L(C^a) \rightarrow \neg c \in L(C^a)$ (negation)
- $c, d \in L(C^a) \rightarrow c \sqcap d \in L(C^a)$ (intersection)
- $c, d \in L(C^a) \rightarrow c \sqcup d \in L(C^a)$ (union).

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

3.1.2 Semantics

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

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

An interpretation I satisfies a terminological axiom $c \sqsubseteq d$, written $\models_I c \sqsubseteq d$ iff $I(c) \subseteq I(d)$. For a set of statements Γ , we write that $\models_I \Gamma$ iff for every $\gamma \in \Gamma$, it holds that $\models_I \gamma$. We write that $\Gamma \models \Gamma'$ iff for all I : $\models_I \Gamma$ implies $\models_I \Gamma'$. Given a TBox, the relation \sqsubseteq can be computed efficiently using standard DL reasoning techniques.

The semantics of membership statements is defined as: $\models_I c(a)$ iff $I(a) \in I(c)$. We assume that the ABox is sound wrt the intended interpretation, i.e., $\models_{INT} A$.

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 O_i and $O_{i_{cv}}$ (Assumptions 1.1 and 1.3) can be straightforwardly established using two TBoxes: T_i and $T_{i_{cv}}$. Common knowledge over O_{cv} (Assumption 1.2) is established using the TBox T_{cv} of which both agents maintain a version. Because both agents have the same version of T_{cv} and they know that of each other, O_{cv} becomes common knowledge. We do not index 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 $c, d \in C_i$: $T_i \models c \sqsubseteq d$ iff $c \leq d$.
- 2 For $i \in \{1, 2\}$, for all $c, d \in C_{i,cv}$: $T_{i,cv} \models c \sqsubseteq d$ iff $c \leq d$.
- 3 For all $c, d \in C_{cv}$: $T_{cv} \models c \sqsubseteq d$ 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 :

| T_1 | T_{cv} | $T_{1,cv}$ |
|---|--------------------------------|--|
| $motorhome \sqsubseteq vehicle$ $hotel \sqsubseteq \neg vehicle$ | $roadvehicle \sqsubseteq \top$ | $roadvehicle \sqsubseteq vehicle$ $motorhome \sqsubseteq roadvehicle$ $hotel \sqsubseteq \neg vehicle$ |

The TBoxes that are possessed by α_2 are:

| T_2 | T_{cv} | $T_{2,cv}$ |
|--|--------------------------------|--|
| $RV-large \sqsubseteq roadvehicle$ $RV-XL \sqsubseteq roadvehicle$ $RV-XL \sqsubseteq \neg RV-large$ | $roadvehicle \sqsubseteq \top$ | $RV-large \sqsubseteq roadvehicle$ $RV-XL \sqsubseteq roadvehicle$ $RV-XL \sqsubseteq \neg RV-large$ |

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.*, $O_{1,2}$ 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 agents' local concepts. For example, $T_{1,cv} \cup T_{2,cv}$ 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 $a \in I^{NT}(c)$ **then** add $c(a)$ to A

else add $\neg c(a)$ to A

For example, `Classify` can be thought of as a subsystem of a robot that recognises and classifies objects in the real world (cf. Steels' (1998) approach to language creation). 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).

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 `send(α_j , \langle topic, p_1 , ..., 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 `Receive(α_i , \langle topic, x_1 , ..., 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 (Burnstein *et al.*, 2003; Williams, 2004; Mitra *et al.*, 2004). 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 (van Diggelen *et al.*, 2006). 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 Maynard *et al.* (2004) 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 URLs to refer to these addresses (*i.e.*, IND is the set of URLs). 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 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 realise these effects, α_j 's TBoxes $T_{j,cv}$ and T_{cv} must be updated such that Property 1.2 and 1.3 hold. To realise the effects regarding 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 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 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 $T_{j,cv}$

The effects of concept learning on $O_{j,cv}$ 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 Figures 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 ($\alpha_j, \langle \text{explication}, c, \{p | I(p) \in I^{INT}(c)\}, \{n | I(n) \notin I^{INT}(c)\} \rangle$)

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

- add $c \sqsubseteq d_j$ to $T_{j,cv}$, where d_j is most specific in the set $\{d'_j | \forall p \in P. I(p) \in I^{INT}(d'_j)\}$
- add $d_j \sqsubseteq c$ to $T_{j,cv}$, where d_j is most general in the set $\{d'_j | \forall n \in N. I(n) \notin I^{INT}(d'_j)\}$

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 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 c to C_{cv}^a , 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 $L(C_{cv}^a \cup \{c\}) \setminus L(C_{cv}^a \setminus \{c\})$. The sending agent α_i conveys the meanings of these concepts that are also in C_i . We can now define the action `AddConcept` as follows:

Action AddConcept(α_j, c)

- For all $d \in (L(C_{cv}^a \cup \{c\}) \setminus L(C_{cv}^a \setminus \{c\})) \cap C_i$:
 - a SendBoundaries(α_j, d)
 - b 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 recognise 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 (Table 1).

Table 1 Evaluation of the protocols

| <i>Protocols</i> | <i>Sound and lossless</i> | <i>Lazy</i> | <i>Minimal cv</i> |
|------------------|---------------------------|-------------|-------------------|
| P1 | + | - | - |
| P2 | + | + | - |
| P3 | + | + | + |

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 cannot 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($\alpha_j, \langle \text{InformExact}, d_{cv}(a) \rangle$)

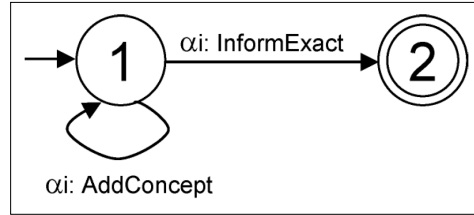
Action Receive($\alpha_i, \langle \text{InformExact}, d_{cv}(a) \rangle$)

Add $e_j(a)$ to 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 InformExact 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. 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 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 C_j , thereby meeting the lossless requirement.

Figure 4 Protocol P1



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 which are performed by the agents are described. We describe some of the instructions that are executed *within* an action; these are preceded with \perp .

$\alpha_2 : \text{AddConcept}(\alpha_1, roadvehicle)$
 $\perp \alpha_2 : \text{add } roadvehicle \sqsubseteq \top \text{ to } T_{cv}$
 $\perp \alpha_1 : \text{add } motorhome \sqsubseteq roadvehicle \sqsubseteq vehicle \text{ to } T_{1,cv}$
 $\alpha_2 : \text{InformExact}(\alpha_1, roadvehicle(a))$
 $\perp \alpha_2 : \text{send}(\alpha_1, \langle \text{InformExact}, roadvehicle(a) \rangle)$
 $\alpha_1 : \text{receive}(\alpha_2, \langle \text{InformExact}, roadvehicle(a) \rangle)$
 $\perp \alpha_1 : \text{add } vehicle(a) \text{ to } A_1$

This conversation has given rise to a communication vocabulary as shown 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:

$\alpha_1 : \text{AddConcept}(\alpha_2, hotel)$
 $\perp \alpha_1 : \text{add } hotel \sqsubseteq \neg roadvehicle \text{ to } T_{cv}$
 $\perp \alpha_2 : \text{add } hotel \sqsubseteq \neg roadvehicle \text{ to } T_{2,cv}$
 $\alpha_1 : \text{InformExact}(\alpha_2, hotel(a))$
 $\perp \alpha_1 : \text{send}(\alpha_2, \langle \text{InformExact}, hotel(a) \rangle)$
 $\alpha_2 : \text{receive}(\alpha_1, \langle \text{InformExact}, hotel(a) \rangle)$
 $\perp \alpha_2 : \text{add } \neg roadvehicle(a) \text{ to } A_2$

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 wrt minimal cv construction and laziness (cf. Table 1). 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 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 recognise when communication has been lossless and when not.

Because the receiver does not know the transferendum, it cannot 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 the philosophy of language, such a derivation is known as a *conversational implicature* (Grice, 1975). In Protocol 2, it works as follows: consider the ontologies O_1 and O_2 from Figure 2, and suppose that $C_{cv}^a = \{\text{vehicle}, \text{motorhome}\}$. 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 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 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 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 assess that communication was lossless.

Action $\text{OK}(\alpha_i)$

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

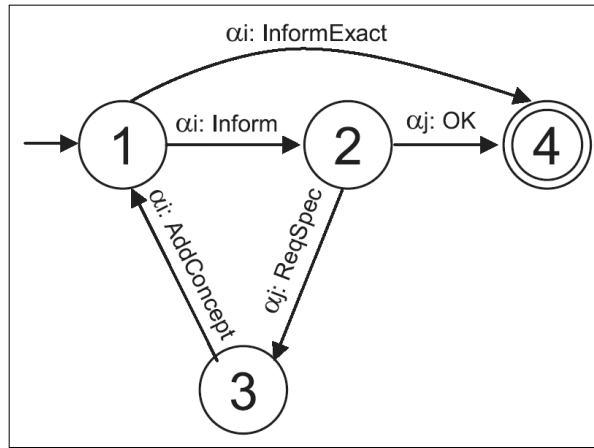
Let D be the set of concepts that are most general among the set of strict subconcepts of d_{cv} in C_{cv}

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(α_i , (OK))

Figure 5 Protocol P2



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(α_2 , $hotel(a)$)
 $\perp \alpha_1$: send(α_1 , $\langle \text{Inform}, \neg roadvehicle(a) \rangle$)
 α_2 : Receive (α_1 , $\langle \text{Inform}, \neg roadvehicle(a) \rangle$)
 $\perp \alpha_2$: add $\neg roadvehicle(a)$ to A_2
 α_2 : OK

In this example, α_2 responded with OK, because in 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 $C_{cv}^a = \{motorhome\}$. The dialogue proceeds as follows:

α_2 : Inform(α_1 , $RV\text{-large}(a)$)
 $\perp \alpha_2$: send(α_1 , $\langle \text{Inform}, motorhome(a) \rangle$)
 α_1 : Receive (α_2 , $\langle \text{Inform}, motorhome(a) \rangle$)
 $\perp \alpha_1$: add $motorhome(a)$ to A_1
 α_1 : OK

In this example, α_1 responded with OK, because in O_1 the information provided by *motorhome* is as specific as possible.

Now, suppose that α_2 wishes to communicate *RV-large(a)*, $C_{cv}^a = \{vehicle\}$.

```

 $\alpha_2$  : Inform( $\alpha_1$ , RV-large(a))
 $\perp \alpha_2$  : send( $\alpha_1$ , ⟨Inform, vehicle(a)⟩)
 $\alpha_1$  : Reqspect
 $\alpha_2$  : AddConcept( $\alpha_1$ , RV-large)
 $\alpha_2$  : InformExact( $\alpha_1$ , RV-large(a))
 $\alpha_1$  : Receive ( $\alpha_1$ , ⟨InformExact, RV-large(a)⟩)
 $\perp \alpha_2$  : add motorhome(a) to  $A_1$ 

```

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

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

```

 $\alpha_2$  : Inform( $\alpha_1$ , roadvehicle(a))
 $\perp \alpha_2$  : send( $\alpha_1$ , ⟨Inform, vehicle(a)⟩)
 $\alpha_1$  : Receive ( $\alpha_2$ , ⟨Inform, vehicle(a)⟩)
 $\perp \alpha_1$  : add vehicle(a) to  $A_1$ 
 $\alpha_1$  : 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 recognised 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 $\{e'_j | c_i \leq_{1,2} e'_j \wedge e'_j \in C_j\}$ (Definition 5 does not hold). This means that either e_j is not in the set $\{e'_j | c_i \leq_{1,2} e'_j \wedge e'_j \in C_j\}$ (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 $\{e'_j | c_i \leq_{1,2} e'_j \wedge e'_j \in C_j\}$, 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 $\{d''_{cv} | c_i \leq_{1,2} d''_{cv} \wedge d''_{cv} \in C_{cv}\}$. 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, wrt laziness (cf. Table 4). However, the protocol may still give rise to a communication vocabulary that 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(α_2 , *motorhome*(a))
 $\perp \alpha_1$: send(α_2 , \langle Inform, $\top(a)$ \rangle)
 α_2 : Reqspec
 α_1 : AddConcept(α_2 , *motorhome*)
 α_1 : InformExact(α_2 , *motorhome*(a))
 α_2 : Receive(α_1 , \langle InformExact, *motorhome*(a) \rangle)
 $\perp \alpha_2$: add *roadvehicle*(a) to A_2

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

The last communication vocabulary is unnecessarily large, because $\{motorhome, vehicle\}$ enables the agents to losslessly communicate the same concepts as $\{motorhome, roadvehicle, vehicle\}$. For this reason, P2 is not satisfactory wrt 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 $L(C_{cv}^a \setminus \{d\})$ allows α_i and α_j to losslessly communicate the same concepts to each other as $L(C_{cv}^a)$.

The above definition does not state how agents can recognise 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
 - a d'_{cv} is most general in the set of subconcepts of d_{cv} in the set $L(C_{cv}^a \setminus \{d_{cv}\})$
 - b 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
 - a d_{cv}'' is most specific in the set of superconcepts of d_{cv} in the set $L(C_{cv}^a \setminus \{d_{cv}\})$
 - b c_i' is most specific in the set of superconcepts of d_{cv} in C_i .

In this definition, the formula $L(C_{cv}^a \setminus \{d_{cv}\})$ 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 $C_{cv}^a = \{\text{motorhome}, \text{roadvehicle}, \text{vehicle}\}$. 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 Definition 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 : Every concept c_j that 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_i' . 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 recognises lossless communication), because Bullet 1 in Definition 7 ensures that all subconcepts in 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 C_{cv}^a
- send($\alpha_j, \langle \text{RemoveConcept}, d \rangle$)

else fail

Action Receive(\langle RemoveConcept, d \rangle)
 Remove d_{cv} from C_{cv}^a

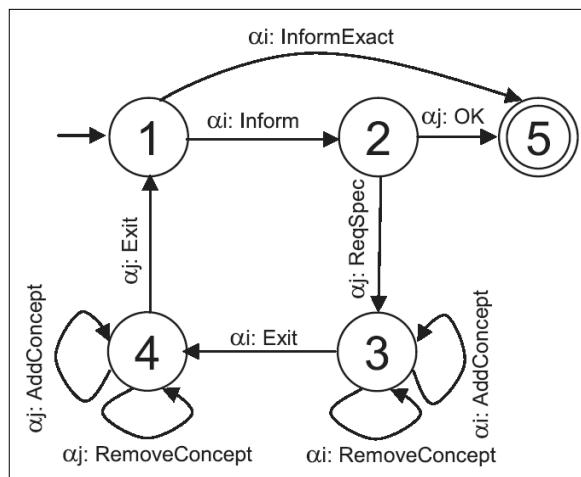
An agent performs a RemoveConcept action on a concept d_{cv} , when it considers it redundant using the criteria described in Definition 7. Concepts may become redundant after a new term is added to the communication vocabulary. Because both agents have different perspectives on the redundancy of terms, 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, the receiver α_j is also 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 $C_{cv}^a = \{motorhome, roadvehicle\}$. Suppose that α_1 wishes to communicate $vehicle(a)$:

α_1 : Inform($\alpha_2, vehicle(a)$)
 $\perp \alpha_1$: send(α_2, \langle Inform, $T(a)$ \rangle)
 α_2 : Reqspec
 α_1 : AddConcept($\alpha_2, vehicle$)
 α_1 : RemoveConcept($\alpha_1, roadvehicle$)
 α_1 : Exit
 α_2 : Exit
 α_1 : InformExact($\alpha_2, vehicle(a)$)
 ...

Figure 6 Protocol P3



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 $C_{cv}^a = \{RV\text{-large}\}$ and that α_2 intends to communicate $RV\text{-XL}(a)$:

```

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

```

Because P3 enables the agents to remove redundant concepts from the communication vocabulary, P3 scores better wrt minimal cv construction than P2. As shown in Table 1, 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 wrt 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

- Baader, F., McGuinness, D.L. and Patel-Schneider, P.F. (2003) *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge University Press.
- Bailin, S.C. and Truszkowski, W. (2002) ‘Ontology negotiation between intelligent information agents’, *Knowledge Engineering Review*, Vol. 17, No. 1, pp.7–19.

- Beun, R.J., van Eijk, R.M. and Priist, H. (2004) 'Ontological feedback in multi-agent systems', *Proceedings of Third International Conference on Autonomous Agents and Multi-agent Systems*, New York: ACM Press, pp.110–117.
- Borgida, A. and Serafini, L. (2002) 'Distributed description logics: directed domain correspondences in federated information sources', *Proceedings of the International Description Logics Workshop DL'2002*.
- Bouquet, P., Kuper, G., Scoz, M. and Zanobini, S. (2004) 'Asking and answering semantic queries', in P. Bouquet and L. Serafini (Eds.) *Proceedings of the ISWC-04 Workshop on Meaning Coordination and Negotiation (MCN-04)*.
- Burnstein, M., McDermott, D., Smith, D.R. and Westfold, S.J. (2003) 'Derivation of glue code for agent interoperation', *Autonomous Agents and Multi-agent Systems*, Vol. 6, No. 3, pp.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*, Vol. 26, No. 2, pp.231–243.
- Genesereth, M.R. and Nilsson, N.J. (1987) *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann Publishers Inc.
- Grice, H.P. (1975) 'Logic and conversation', in P. Cole and J.L. Morgan (Eds.) *Speech Acts*, New York: Academic Press, Vol. 4158.
- Gruber, T.R. (1993) 'A translation approach to portable ontology specifications', *Knowledge Acquisition*, Vol. 5, pp.199–220.
- Hameed, A., Preece, A. and Sleeman, D. (2004) 'Ontology reconciliation', in S. Staab and R. Studer (Eds.) *Handbook of Ontologies, International Handbooks on Information Systems*, Chap. 12, Berlin, DE: Springer Verlag, pp.231–250.
- Heflin, J. and Hendler, J. (2000) 'Dynamic ontologies on the web', *Proceedings of American Association for Artificial Intelligence Conference (AAAI-2000)*, Menlo Park, CA: AAAI Press, pp.443–449.
- Jackson, P. and Moulinier, I. (2002) *Natural Language Processing for Online Applications: Text Retrieval, Extraction, and Categorization*, John Benjamins Publishing.
- Lenat, D.B. and Guha, R.V. (1990) *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project*, Addison-Wesley Publishers.
- Luck, M., McBurney, P. and Preist, C. (2003) 'Agent technology: enabling next generation computing', *Agent Link Community*.
- Maynard, D., et al. (2004) '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>.
- McGuinness, D.L., Fikes, R., Rice, J. and Wilder, S. (2000) 'The chimaera ontology environment', *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000)*, Austin, Texas.
- 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)*, April, Vol. 8, No. 2, pp.223–272.
- Meyer, J-J.Ch. and Van Der Hoek, W. (1995) *Epistemic Logic for AI and Computer Science*, Cambridge University Press.
- Mitra, P., Noy, N.F. and Jaiswal, A.R. (2004) 'OMEN: a probabilistic ontology mapping tool', in P. Bouquet and L. Serafini (Eds.) *Proceedings of the ISWC-04 Workshop on Meaning Coordination and Negotiation (MCN-04)*.
- Noy, N.F. and Musen, M.A. (2000) 'Prompt: algorithm and tool for automated ontology merging and alignment', *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

- Preece, A.D., Hui, K.Y., 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*, Vol. 13, Nos. 2–3, pp.113–120.
- Rahm, E. and Bernstein, P.A. (2001) 'A survey of approaches to automatic schema matching', *The VLDB Journal*, Vol. 10, No. 4, pp.334–350.
- Rosenschein, J.S. and Zlotkin, G. (1994) *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*, Cambridge, MA: MIT Press.
- Soh, L-K. and Chen, C. (2005) 'Balancing ontological and operational factors in refining multi-agent 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*, Pacific Grove, CA: Brooks/Cole Publishing Co.
- Steels, L. (1998) *Synthesising the Origins of Language and Meaning Using Co-evolution, Self-organisation and Level Formation*, Edinburgh University Press.
- Stuckenschmidt, H. and Timm, I.J. (2002) 'Adaption communication vocabularies using shared ontologies', *Proceedings of the Second International Workshop on Ontologies in Agent Systems (OAS)*, July.
- Stuckenschmidt, H., van Harmelen, F., Serafini, L., Bouquet, P. and Giunchiglia, F. (2004) 'Using cowl for the alignment and merging of medical ontologies', in U. Hahn (Ed.) *Proceedings of the First International Workshop on Formal Biomedical Knowledge Representation (KRMed'04)*, Whistler, Colorado, June, pp.8–101.
- Stumme, G. and Maedche, A. (2001) 'FCA-MERGE: bottom-up merging of ontologies', *IJCAI*, pp.225–234.
- Swartout, B., Patil, R., Knight, K. and Russ, T. (1996) 'Toward distributed use of large-scale ontologies', *Proceedings of the Tenth Knowledge Acquisition for Knowledge-based Systems Workshop*.
- Uschold, M. and Gruninger, M (1996) 'Ontologies: principles, methods, and applications', *Knowledge Engineering Review*, Vol. 11, No. 2, pp.93–155.
- Uschold, M. and Gruninger, M. (2002) 'Creating semantically integrated communities on the world wide web', *Semantic Web Workshop Co-located with WWW 2002 Honolulu*.
- van Diggelen, J., Beun, R.J., Dignum, F., van Eijk, R.M. and Meyer, J-J.Ch. (2004) 'Optimal communication vocabularies and heterogeneous ontologies', *Developments in Agent Communication*, LNAI 3396, Springer Verlag.
- van Diggelen, J., Beun, R.J., Dignum, F., van Eijk, R.M. and Meyer, J-J.Ch. (2006) 'ANEMONE: an effective minimal ontology negotiation environment', *Proceedings of the Fifth International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*.
- Wiederhold, G. and Genesereth, M. (1997) 'The conceptual basis for mediation services', *IEEE Expert: Intelligent Systems and their Applications*, Vol. 12, No. 5, pp.38–47.
- Williams, A.B. (2004) 'Learning to share meaning in a multi-agent system', *Autonomous Agents and Multi-agent Systems*, Vol. 8, No. 2, pp.165–193.

Note

- 1 FIPA Ontology Service Specification, <http://www.fipa.org/specs/fipa00086/>.