

Combining normal communication with ontology alignment

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. This paper considers the combination of agent communication and ontology alignment within a group of heterogeneous agents. The agents align their ontologies by constructing a shared communication vocabulary. Because ontology alignment is not a goal in itself, the agents refrain from it unless they believe it to be inevitable. We discuss three protocols that all implement lazy ontology alignment, although they give rise to different communication vocabularies.

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, 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. Moreover, a common ontology is disadvantageous for the problem solving capacities of the agents as different tasks typically require different ontologies [2].

Over the last decade, much research has been conducted on the alignment of heterogeneous ontologies. Most approaches that deal with these issues require some form of human intervention. Prompt [9] and Chimaera [7] are examples of tools that assist humans in merging and aligning ontologies. However, in open multi agent systems, ontologies have to be aligned on such a large scale, that human involvement in this task is no longer feasible. Recently, a few approaches have been proposed that address the problem of heterogeneous ontologies in a fully automatic way [10, 12, 14]. The primary focus of these approaches is on concept learning: making the meaning of a concept clear to another agent. These techniques might seem to make normal communication protocols applicable in systems with heterogeneous ontologies: before the agents start cooperating, they teach the concepts in their ontologies to each other. This way, each agent would know every other agent's ontology, which would solve their incomprehension. However, such an approach is highly unpractical in an open multi agent system

as it requires an agent to learn a vast amount of foreign concepts before it is able to communicate even the smallest piece of information. To make matters worse, this must be done not only once, but continuously as new agents enter and leave the system all the time.

In this paper, we explore ways to *efficiently* combine ontology alignment techniques with normal agent communication protocols. To make our results as generally applicable as possible, we deal with these issues in a highly abstract and formal way. Our approach allows the agents to preserve their private ontologies for knowledge representation and reasoning. To communicate, the agents build an intermediate ontology (or interlingua [13]). This ontology is shared among all agents and indirectly aligns their ontologies. Because the intermediate ontology is only used for communication purposes, we refer to it as *communication vocabulary* (or cv). Initially, the communication vocabulary is empty. The agents enable themselves to communicate by adding concepts to the cv. This way, the agents gradually extend the cv whenever they consider this necessary. Hence, the ontology alignment technique boils down to adding concepts to the cv at the proper moments. Normal communication proceeds by translating a concept from the speaker's private ontology to the communication vocabulary which the hearer translates back again to its own private ontology.

The communication protocols we propose all implement a combination of normal communication and ontology alignment. We evaluate these protocols according to the following criteria:

- Does the combination give rise to a small, yet sufficiently expressive communication vocabulary?
- Does the combination implement lazy ontology alignment?

The first question is of particular importance as the communication vocabulary should not simply become the union of every agent's private ontology, as often occurs in practice. This way, every agent would have to learn every other agent's ontology. Not only is this very resource-consuming, the situation only gets worse as more agents enter the system. In an open system, it would give rise to a forever growing communication vocabulary rendering itself useless in the long run. As is shown in [3], a *minimal* communication vocabulary may already fully align the agents ontologies.

The second question is of particular importance because the agents should be able to communicate even if their ontologies are not fully aligned. Otherwise the agents would have to learn a large number of concepts to align their complete ontologies, before they can start communicating the matter at hand. Usually, a limited number of shared concepts suffices for the successful communication of a particular matter. Therefore, the ontology alignment protocol should be *lazy*, providing local solutions for communication problems as they arise.

Section 2 describes the conceptual framework which provides the formal underpinning to study the different ontologies in a MAS. Using these notions, we define what qualifies as successful communication. Section 3 describes the operational framework dealing with the implementation of ontologies. Section 4 concerns communication. We briefly describe a formal abstraction of a concept

learning technique. On the operational level, we describe three protocols which combine ontology alignment with normal communication. They all implement lazy ontology alignment, but differ in quality w.r.t. minimal cv construction.

2 Conceptual framework

We assume that the agents have access to the same elements in the universe of discourse (Δ), and use the same symbols to refer to these individuals (given by the set IND). We will focus on the 1-ary relations between these individuals, namely sets of individuals. A conceptualization is defined as a set of 1-ary relations; it is thus a subset of 2^Δ . In our framework, the agents may conceptualize their world differently and are therefore allowed to adopt different conceptualizations. Note that, at this level, the elements in the conceptualization are not yet named. This is done by the ontology, which *specifies* the conceptualization [5]. 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 in [4]). We will refer to the intended interpretation function with \mathcal{I}^{INT} .

Because we are mainly interested in the relations between concepts, an ontology is defined as a preorder $\mathcal{O} = \langle \mathcal{C}, \leq \rangle$ where $\leq \subseteq \mathcal{C} \times \mathcal{C}$ is a relation 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 preorder which is conforming to the subset ordering on the intended interpretations of the concepts. We will write $x \equiv y$ as a shorthand for $x \leq y \wedge y \leq x$, and $x < y$ as a shorthand for $x \leq y \wedge \neg(y \leq x)$.

Given a subset X of \mathcal{C} , we define the following:

- an element $x \in X$ is minimal iff $\neg \exists y \in X. y < x$
- an element $x \in X$ is maximal iff $\neg \exists y \in X. x < y$

Let us now consider a simple multi agent system consisting of 2 agents α_1 and α_2 . We use 1 or 2 in subscript notation whenever we need to stress that something belongs to α_1 or α_2 . In the system, the following ontologies are important ($i \in \{1, 2\}$):

- $\mathcal{O}_i = \langle \mathcal{C}_i, \leq_i \rangle$: the private ontology of α_i .
- $\mathcal{O}_{cv} = \langle \mathcal{C}_{cv}, \leq_{cv} \rangle$: the communication vocabulary.
- $\mathcal{O}_{i.cv} = \langle \mathcal{C}_{i.cv}, \leq_{i.cv} \rangle$, where $\mathcal{C}_{i.cv} = \mathcal{C}_i \cup \mathcal{C}_{cv}$: the private ontology of α_i , the cv and their interrelations.
- $\mathcal{O}_{1.2} = \langle \mathcal{C}_{1.2}, \leq_{1.2} \rangle$, where $\mathcal{C}_{1.2} = \mathcal{C}_1 \cup \mathcal{C}_2$: a God's eye view of the private ontologies of α_1 and α_2 .
- $\mathcal{O}_{1.2.cv} = \langle \mathcal{C}_{1.2.cv}, \leq_{1.2.cv} \rangle$, where $\mathcal{C}_{1.2.cv} = \mathcal{C}_{1.2} \cup \mathcal{C}_{cv}$: a God's eye view of all ontologies in the MAS.

Example: Consider a travel-agent α_1 and an internet travel service α_2 (conforming to the scenario envisioned in [6]). Figure 1 graphically represents the ontologies in the system using Euler diagrams. \mathcal{O}_1 and \mathcal{O}_2 are the agent's private ontologies; when combined, these ontologies form $\mathcal{O}_{1.2}$.

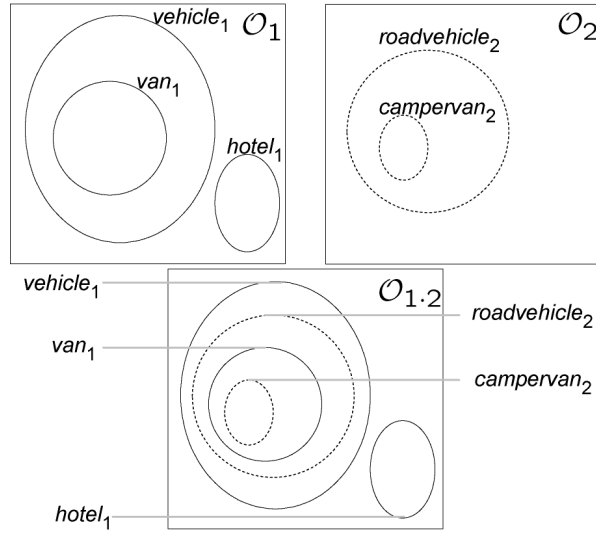


Fig. 1. Example ontologies

2.1 Knowledge distribution

Not every ontology is known by the agents. For example, \mathcal{O}_2 is unknown to α_1 (the agents can not “look inside each others head”). \mathcal{O}_{cv} on the other hand, is known by every agent, whereas $\mathcal{O}_{1.2,cv}$ is only partially known by the agents. We distinguish between local knowledge, common knowledge and implicit knowledge [8]. Local knowledge refers to the knowledge of an individual agent which is not accessible to other agents. Something is common knowledge if it is known by every agent and every agent knows that every agent knows it, which is again known by every agent etc. Something is implicit knowledge within a group, if someone in the group knows it, or the knowledge is distributed over the members of the group. By means of communication, the agents can only acquire knowledge which was already implicit in the group.

Assumption 1

- \mathcal{O}_i is local knowledge of α_i .
- \mathcal{O}_{cv} is common knowledge of the group.
- $\mathcal{O}_{i,cv}$ is local knowledge of α_i .
- $\mathcal{O}_{1.2}$ is implicit knowledge of the group.

Note that this assumption implies that also $\mathcal{O}_{1.2,cv}$ is implicit knowledge within the group. In section 3 it is shown how these characteristics are implemented in the system.

2.2 Communication

Consider communication between agent α_i (the speaker) and α_j (the hearer). α_i translates its private concept $\in \mathcal{C}_i$ to a concept in the communication vocabulary $\in \mathcal{C}_{cv}$, which α_j translates back again to a concept $\in \mathcal{C}_j$. We refer to these concepts as follows:

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

Using these three concepts, we define the requirements of successful communication. The first requirement deals with the *quality* of information exchange, i.e. soundness. The definition of soundness states 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).

Definition 1. *Sound Communication*

Let C_i be the transferendum, and C_j be the translatum. Communication is sound iff $C_i \leq C_j$.

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 superset 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. For this reason, a second requirement is needed which also takes the *quantity* of information exchange into account.

The lossless requirement states that the translatum should not only be a consequence of the transferendum, but should also be the strongest consequence. This ensures that as much information as possible is preserved in the communication process.

Definition 2. *Lossless communication*

Let C_i be the transferendum and C_j the translatum. Communication is lossless iff C_j is minimal among the set $\{C'_j | C_i \leq C'_j\}$.

Note that in definition 1 and 2 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.

3 Operational framework

This section discusses the data-structures and actions that can be used to implement the conceptual framework. Note that this is only one of many possible

implementations. The properties of the components describe the circumstances under which the requirements of assumption 1 are met.

We first discuss how \mathcal{O}_i can be implemented as local knowledge of α_i . As standard in description logic knowledge bases, the agent's knowledge is represented by a tuple $\langle \mathcal{T}_i, \mathcal{A}_i \rangle$, containing a TBox and an ABox [1]. The TBox \mathcal{T}_i contains a set of terminological axioms which specify the inclusion relations between the concepts; it represents the agent's ontology. The ABox \mathcal{A}_i contains a set of membership statements which specify which individuals belong to which concepts; it represents the agent's operational knowledge. \mathcal{T}_i and \mathcal{A}_i are further explained below.

The set of concepts \mathcal{C}_i is defined conforming to the description logic \mathcal{ALC} without roles. Given a set of atomic concepts \mathcal{C}_i^a with typical elements c_i, d_i , the set \mathcal{C}_i with typical elements C_i, D_i , is defined by the BNF:

$$\mathcal{C}_i ::= c_i \mid \perp \mid \top \mid \neg C_i \mid C_i \sqcap D_i \mid C_i \sqcup D_i.$$

The semantics of the elements in \mathcal{C}_i is defined using an interpretation function \mathcal{I} which maps concepts to sets of individuals. \mathcal{I} is such that $\mathcal{I}(\top) = \Delta$, $\mathcal{I}(\perp) = \emptyset$, $\mathcal{I}(\neg C) = \Delta \setminus \mathcal{I}(C)$, $\mathcal{I}(C \sqcap D) = \mathcal{I}(C) \cap \mathcal{I}(D)$ and $\mathcal{I}(C \sqcup D) = \mathcal{I}(C) \cup \mathcal{I}(D)$. A terminological axiom is a statement of the form $C \sqsubseteq D$. A TBox \mathcal{T} is a set of terminological axioms. 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 $\models_{\mathcal{I}} \Gamma'$. We assume that enough terminological axioms are contained in \mathcal{T}_i , s.t. it fully implements the local knowledge of agent α_i over \leq_i , i.e.

Property 1. For $i \in \{1, 2\}$: $\mathcal{T}_i \models C_i \sqsubseteq D_i$ iff $C_i \leq D_i$.

Given a TBox, the relation \sqsubseteq can be computed efficiently using standard DL reasoning techniques.

A membership statement is defined as a statement of the form $C(a)$, where C is a concept and a an individual name ($\in \text{IND}$). IND refers to the set of individual names; we assume that the part of \mathcal{I} which maps elements of IND to elements of Δ is common knowledge. An interpretation function \mathcal{I} satisfies a membership statement $C(a)$ iff $\mathcal{I}(a) \in \mathcal{I}(C)$. The ABox \mathcal{A} is a set of membership statements. We assume that the ABox is sound w.r.t. to the intended representation, i.e. $\models_{\mathcal{I}^{\text{INT}}} \mathcal{A}$. 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. The assumption of a complete ABox is, nevertheless, unrealistic. Typically, the domain of discourse will be of such size that it is unfeasible to enumerate all membership statements.

Let us now focus on \mathcal{O}_{cv} . For the purposes of this paper, it suffices to define the set \mathcal{C}_{cv} as a simpler language than \mathcal{C}_i ; in particular, we leave out the \sqcup and \sqcap constructors. Given a set of atomic concepts \mathcal{C}_{cv}^a , the elements in \mathcal{C}_{cv} are defined as $\mathcal{C}_{cv} ::= c_{cv} \mid \perp \mid \top \mid \neg C_{cv}$. The omission of the \sqcup and \sqcap constructors makes it easier for an agent to achieve local knowledge of $\mathcal{O}_{i,cv}$; an extension of the framework to deal with a cv that includes these constructors is straightforward. The agents store their knowledge about \leq_{cv} in a TBox, $\mathcal{T}_{i,cv}$. The protocols

described in 4 are such that every agent knows the ordering between the concepts in \mathcal{C}_{cv} :

Property 2. For $i \in \{1, 2\}$: $\mathcal{T}_{i-cv} \models C_{cv} \sqsubseteq D_{cv}$ iff $C_{cv} \leq_{cv} D_{cv}$.

Our next focus is \mathcal{O}_{i-cv} . Because the local knowledge of \mathcal{O}_i , and the common knowledge of \mathcal{O}_{cv} has already been discussed above, we only need to focus on the relations between the concepts in \mathcal{C}_i and those in \mathcal{C}_{cv} . This knowledge of agent α_i is stored using terminological axioms of the form $C_i \sqsubseteq C_{cv}$ or $C_{cv} \sqsubseteq C_i$. These terminological axioms are added to the TBox \mathcal{T}_{i-cv} . The communication protocols implement local knowledge of \mathcal{O}_{i-cv} by giving rise to \mathcal{T}_{i-cv} with the property:

Property 3. For $i \in \{1, 2\}$:

- $\mathcal{T}_{i-cv} \models C_{cv} \sqsubseteq C_i$ iff $C_{cv} \leq_{i-cv} C_i$
- $\mathcal{T}_{i-cv} \models C_i \sqsubseteq C_{cv}$ iff $C_i \leq_{i-cv} C_{cv}$

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. The data structures as described until now do not give rise to implicit knowledge of the relations between two different agent's private concepts. 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. To solve this, we build on the assumption that an agent not only knows the ordering between its private concepts, but also has access to the intended interpretation of its private 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. This underlies Luc Steels' approach to language creation [11]. In a scenario where the domain of discourse consists of text corpora, the action **Classify** can be implemented using a text classification technique like those used in spam filters.

4 Communication

The communicative abilities of the agents are specified as actions. During the execution of actions, messages are sent through the instruction $\text{send}(\alpha_j, \langle \text{topic}, p_1, \dots, p_n \rangle)$, 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**($\alpha_i, \langle \text{topic}, x_1, \dots, x_n \rangle$)

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.

We first describe how concept learning is established in our framework. Then, we describe how this concept learning technique can be used in combination with normal communication to establish lazy ontology alignment.

4.1 Concept learning

The agents extend the communication vocabulary using the action **AddConcept**. We first describe the changes in ontologies from the conceptual level. In describing these changes we use the notion of projection:

Definition 3. Let $\mathcal{O} = \langle \mathcal{C}, \leq \rangle$ be an ontology. For $\mathcal{C}' \subseteq \mathcal{C}$, we define $\mathcal{O} \upharpoonright \mathcal{C}'$ to be $\langle \mathcal{C}', \{ \langle x, y \rangle \mid \langle x, y \rangle \in \leq \wedge x, y \in \mathcal{C}' \} \rangle$

Suppose α_i performs the action **AddConcept**(α_j, C_i, c_{cv}). As a result, the knowledge in the system changes. Let \mathcal{O} be the ontology before the action, and \mathcal{O}^+ be the ontology after the action. The change in ontologies is described as follows ($i \in \{1, 2\}$):

1. $\mathcal{O}_{1..2..cv}^+ = \langle \mathcal{C}_{1..2..cv}^+, \leq_{1..2..cv}^+ \rangle$, where $\mathcal{C}_{1..2..cv}^+ = \mathcal{C}_{1..2..cv} \cup \{c_{cv}\}$ and $\leq_{1..2..cv}^+$ is the reflexive, transitive closure of $\leq_{1..2..cv} \cup \{ \langle C_i, c_{cv} \rangle, \langle c_{cv}, C_i \rangle \}$.
2. $\mathcal{O}_{cv}^+ = \mathcal{O}_{1..2..cv}^+ \upharpoonright (\mathcal{C}_{cv} \cup \{c_{cv}\})$
3. $\mathcal{O}_{i..cv}^+ = \mathcal{O}_{1..2..cv}^+ \upharpoonright (\mathcal{C}_{i..cv} \cup \{c_{cv}\})$
4. $\mathcal{O}_{j..cv}^+ = \mathcal{O}_{1..2..cv}^+ \upharpoonright (\mathcal{C}_{j..cv} \cup \{c_{cv}\})$

We now describe the changes in ontologies from the operational level. In doing so, it suffices to regard only items two, three and four, as the first item follows from these.

The second item (\mathcal{O}_{cv}^+) concerns common knowledge. Because α_i knows the exact meaning of c_{cv} , it knows \mathcal{O}_{cv}^+ . To make it common knowledge, α_i sends the information ($\leq_{cv}^+ \setminus \leq_{cv}$) to α_j . This is done in the message with the “boundaries” topic (specified below). The third item ($\mathcal{O}_{i..cv}^+$) follows straightforward from the knowledge of α_i that $c_{cv} \equiv C_i$. The fourth item ($\mathcal{O}_{j..cv}^+$) is the most difficult one to establish. Given that $C_i \equiv c_{cv}$, neither $\mathcal{O}_{i..cv}$, nor $\mathcal{O}_{j..cv}$ gives sufficient information to establish the relations in $\mathcal{O}_{j..cv}^+$. Therefore, α_i conveys this information to α_j by sending an *ostensive definition* [15], consisting of a set of positive and negative examples of concept c_{cv} . Upon receiving these examples, α_j uses inductive inference to derive the relations of c_{cv} with the concepts in its private ontology. This is done in the message with the “explication” topic.

The action **AddConcept** is specified as follows. Remember that the agents have access to the intended interpretation of concepts using the **Classify** action described earlier.

Action AddConcept(α_j, c_{cv}, C_i)

- add $\langle c_{cv}, \langle C_i, C_i \rangle \rangle$ to $\mathcal{T}_{i.cv}$
- send $(\alpha_j, \langle \text{boundaries}, c_{cv}, \{D_{cv} | c_{cv} \leq D_{cv}\}, \{E_{cv} | E_{cv} \leq c_{cv}\} \rangle)$
- send $(\alpha_j, \langle \text{explication}, c_{cv}, \{p | \mathcal{I}(p) \in \mathcal{I}^{INT}(C_i)\}, \{n | \mathcal{I}(n) \notin \mathcal{I}^{INT}(C_i)\} \rangle)$

Action Receive($\langle \text{boundaries}, c_{cv}, \text{Sup}, \text{Sub} \rangle$)

- for every $C'_{cv} \in \text{Sup}$: add $c_{cv} \sqsubseteq C'_{cv}$ to $\mathcal{T}_{j.cv}$
- for every $C'_{cv} \in \text{Sub}$: add $C'_{cv} \sqsubseteq c_{cv}$ to $\mathcal{T}_{j.cv}$

The “boundaries” message ensures that the ordering on \mathcal{C}_{cv} is common knowledge between α_i and α_j , thereby satisfying property 2.

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

- add $c_{cv} \sqsubseteq C_j$ to $\mathcal{T}_{j.cv}$, where C_j is minimal among the set $\{C'_j | \forall p \in P. \mathcal{I}(p) \in \mathcal{I}^{INT}(C'_j)\}$
- add $D_j \sqsubseteq c_{cv}$ to $\mathcal{T}_{j.cv}$, where D_j is maximal among 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, such that all information about the concept C_i is conveyed to α_j . Under this assumption, property 3 holds.

4.2 Protocols for lazy ontology alignment

Building a communication vocabulary is not the primary goal of the agents, but only a means to achieve successful communication. Therefore, the agents should only resort to ontology alignment when their communication vocabulary falls short of successful communication, i.e. the ontology alignment protocol should be lazy. This requires the agents to know when their communication qualifies as successful. In section 2.2 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 discussions on lazy ontology alignment. Three different protocols are discussed below.

Protocol 1 In protocol 1, only messages are sent which are guaranteed to result in lossless communication. This requires the sender of a message to know whether its message will result in lossless communication or not. The sender knows that the receiver’s knowledge about a transferens is as *accurate* as possible (property 3). Therefore, the sender knows that whenever it uses a transferens which corresponds *exactly* to the transferendum, lossless communication will be established. This idea is captured in the precondition of the **InformExact** action. If this speech act cannot be performed, the agent is forced to add the term to

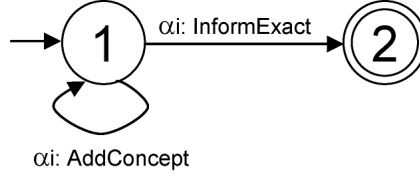


Fig. 2. Protocol P1

the communication vocabulary.

Action InformExact($\alpha_j, C_i(a)$)
if $\exists C_{cv}. C_{cv} \equiv C_i$ **then** send($\alpha_j, \langle \text{InformExact}, C_{cv}(a) \rangle$)
else fail

Action Receive($\langle \text{InformExact}, C_{cv}(a) \rangle$)
 Add $C_j(a)$ to \mathcal{A}_j , where C_j is minimal among the set $\{C'_j | C_{cv} \leq C'_j\}$

It is not difficult to prove that in protocol 1, communication proceeds in a lossless fashion as defined in definition 2. The event that is triggered upon receiving an InformExact message, produces a translatum C_j which is minimal among the set $\{C'_j | C_{cv} \leq C'_j\}$. Because the action that produces an InformExact message requires the transferendum C_i to be equivalent to C_{cv} , it follows that C_j is also minimal among the set $\{C'_j | C_i \leq C'_j\}$, thereby meeting the lossless requirement.

Example: Consider the ontologies in figure 1. Initially $\mathcal{C}_{cv} = \{\top, \perp\}$. Suppose that α_1 intends to convey the assertion $van_1(a)$. 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_1 : AddConcept($\alpha_2, van_{cv}, van_1$)
 α_1 : InformExact($\alpha_2, van_1(a)$)
 $\perp \alpha_1$: send($\alpha_2, \langle \text{InformExact}, van_{cv}(a) \rangle$)
 α_2 : receive($\alpha_1, \langle \text{InformExact}, van_{cv}(a) \rangle$)
 $\perp \alpha_2$: add *roadvehicle*₂(a) to \mathcal{A}

Now, suppose that α_2 intends to convey the message *campervan*₂(a), and that $\mathcal{C}_{cv}^a = \{van_{cv}\}$. Here, and in the following examples, the meaning of concepts in \mathcal{C}_{cv}^a is as expected, e.g. $van_{cv} \equiv van_1$. The agents perform the following actions:

α_2 : AddConcept($\alpha_1, campervan_{cv}, campervan_2$)
 α_2 : InformExact($\alpha_1, campervan_2(a)$)
 $\perp \alpha_2$: send($\alpha_1, \langle \text{InformExact}, campervan_{cv}(a) \rangle$)
 ...

Although P1 always allows lossless communication, it does not give rise to a minimal cv. The condition maintained by the sender is a *sufficient* condition for lossless communication, but it is not a *necessary* condition. In the second

dialogue of the example, it was not necessary to add a new concept to the cv, as lossless communication was already enabled by the concept van_{cv} . Sometimes, the sender adds concepts to the cv that do not contribute to successful communication. 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. The following protocol attempts to overcome the problem of redundantly adding concepts to the cv.

Protocol 2 In protocol 2, the sender uses an InformExact speech act 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 accurately as possible* using a more general concept. It is upon the receiver to decide whether this approximation is accurate enough to meet the lossless criterion.

Because the receiver does not know the transferendum, it cannot directly check definition 2 for lossless communication. However, the receiver knows two things about the transferendum, which enables it, in some cases, to check the lossless condition nonetheless. Firstly, it knows that the transferendum is more specific than the transferens. Secondly, it knows that the transferens is the most accurate translation of the transferendum to the communication vocabulary. Therefore, any concept in C_{cv} which is more specific than the transferens is *not* more general than the transferendum. These ideas underly the action OK.

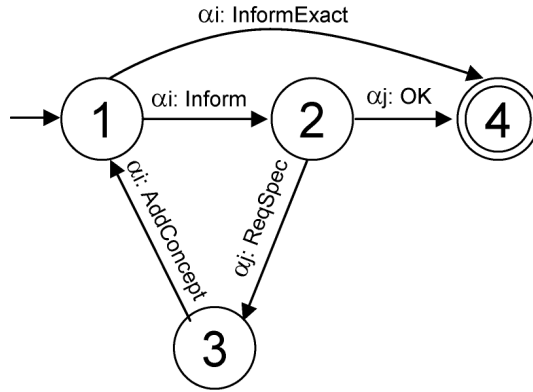


Fig. 3. Protocol P2

Action $\text{Inform}(\alpha_j, C_i(a))$
 $\text{send}(\alpha_j, \langle \text{Inform}, C_{cv}(a) \rangle)$ where C_{cv} is minimal among the set $\{C_{cv} | C_i \leq C_{cv}\}$

The event that is triggered when an inform message is received is equal to the event that is triggered when an InformExact message is received. The OK action

fails if the receiver cannot assess that communication was lossless; otherwise it responds with OK.

Action OK(α_i)

Responding to $\langle \text{inform}, (C_{cv}(a)) \rangle$

if $\neg \exists C_j$ for which

1. $C_j < D_j$, where D_j is minimal among the set $\{C'_j | C_{cv} \leq C'_j\}$ (D_j is the translatum)
2. $\neg \exists C'_{cv}. C'_{cv} < C_{cv} \wedge C_j \leq C'_{cv}$

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

else fail

If the receiver cannot respond with OK, it requests for specification (ReqSpec). After this, the sender adds a concept to the communication vocabulary.

Theorem 1. *If the receiver responds OK then communication was lossless.*

Proof: Suppose C_i is the transferendum, C_{cv} the transferens and C_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. The conditions for sending and receiving an inform speech act ensure that $C_i \leq C_{cv} \leq C_j$, and therefore $C_i \leq C_j$. Non-lossless communication means that C_j is *not* minimal among the set $\{C'_j | C_i \leq C'_j\}$. Therefore $\exists C'_j. C_i \leq C'_j < C_j$. This C'_j meets the first condition in the if-statement of OK; therefore, the second condition must be false, i.e. $\exists C'_{cv}. C'_{cv} < C_{cv} \wedge C'_j \leq C'_{cv}$. Therefore, $C_i \leq C'_{cv} \wedge C'_{cv} < C_{cv}$. This is in contradiction with the condition of **Inform** which states that C_{cv} should be minimal among the set $\{C'_{cv} | C_i \leq C'_{cv}\}$.

□

Example: Consider the ontologies in figure 1. Suppose that α_2 wishes to communicate $campervan_2(a)$, and that $C_{cv}^a = \{van_{cv}\}$. The dialogue proceeds as follows:

α_2 : **Inform**($\alpha_1, campervan_2(a)$)

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

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

$\perp \alpha_1$: add $van_1(a)$ to \mathcal{A}_1

α_1 : OK

In this example, α_1 responded with OK, because in \mathcal{O}_1 the information provided by van_1 is as accurate as possible.

Now, suppose that α_2 wishes to communicate $campervan_2(a)$, $C_{cv}^a = \{vehicle_{cv}\}$.

α_2 : **Inform**($\alpha_1, campervan_2$)

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

α_1 : **Reqspect**

α_2 : **AddConcept**($\alpha_1, campervan_{cv}, campervan_2$)

α_2 : **InformExact**($\alpha_1, campervan_2(a)$)

...

In this example α_1 did not respond OK at first, because van_1 caused the action to fail. Hereby, α_1 correctly recognized non-lossless communication.

Now, suppose that α_2 wishes to communicate $roadvehicle_2(a)$, and $\mathcal{C}_{cv}^a = \{vehicle_{cv}, van_{cv}, (vehicle \sqcap \neg van)_{cv}\}$ (in the extended framework, $(vehicle \sqcap \neg van)_{cv}$ can be compositionally defined in \mathcal{C}_{cv} , instead of atomic)

$\alpha_2 : \mathbf{Inform}(\alpha_1, roadvehicle_2(a))$
 $\perp \alpha_2 : \mathbf{send}(\alpha_1, \langle \mathbf{Inform}, vehicle_{cv}(a) \rangle)$
 $\alpha_1 : \mathbf{Receive}(\alpha_2, \langle \mathbf{Inform}, vehicle_{cv}(a) \rangle)$
 $\perp \alpha_1 : \mathbf{add} vehicle_1(a) \text{ to } \mathcal{A}_1$
 $\alpha_1 : \mathbf{OK}$

In this example, α_1 responded OK, because it knew that if α_2 had more information available about individual a , e.g. van_1 , it would have used a more specific term, e.g. van_{cv} . Hereby, α_1 correctly recognized lossless communication.

Protocol P2 enables the agents to communicate without having to share all their private concepts. However, the protocol may still give rise to a communication vocabulary which is unnecessary large. Protocol 3 allows the agents to remove superfluous concepts from their communication vocabulary.

Protocol 3 Concepts can be removed from the vocabulary if they are *redundant*. Redundant concepts have the property that their removal does not affect the expressiveness of the cv. We measure the expressiveness of the communication vocabulary, as the number of private concepts that can be losslessly communicated, without having to extend the cv.

Definition 4. *If c_{cv} is redundant in \mathcal{C}_{cv}^a , then $\mathcal{C}_{cv}^a \setminus \{c_{cv}\}$ allows for lossless communication of the same concepts as \mathcal{C}_{cv}^a .*

Whereas this definition can be verified from a God's eye view perspective, an agent can only indirectly check its validity. Agent α_i knows which transferendum C_i uses which transferens C_{cv} (it knows how to send an inform message). It also knows which transferens C_{cv} , is translated into which translatum C_i (it knows how to receive an inform message). This enables α_i to know that a concept C_{cv} is redundant if the following holds for c_{cv} :

- no transferendum $\in \mathcal{C}_i$ requires transferens c_{cv} . This means that α_i would never use c_{cv} in its messages.
- there is another transferens $C'_{cv} \in \mathcal{C}_{cv} \setminus \{c_{cv}\}$, which yields the same translatum as c_{cv} , and is more general than c_{cv} . This means that, as far as α_i is concerned, α_j might as well use C'_{cv} instead of c_{cv} , when α_j informs α_i about something.

An agent performs a **RemoveConcept** action on a concept c_{cv} , when it considers it redundant using the criteria described above. Concepts may become redundant after a new term is added to the communication vocabulary. Therefore, P3 allows the **RemoveConcept** action *after* **AddConcept**. Because both agents have different perspectives on the redundancy of terms, both agents get a chance to

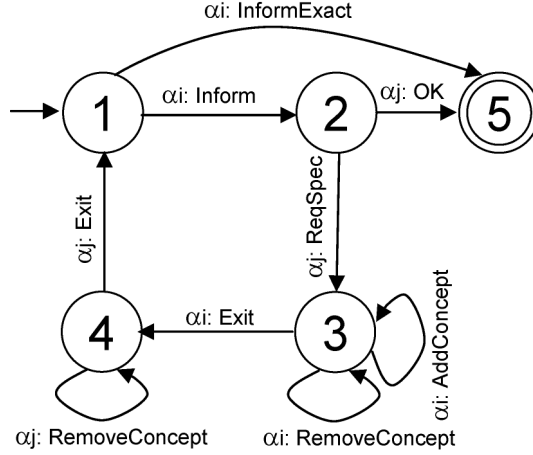


Fig. 4. Protocol P3

perform `RemoveConcept`. Due to space limitations, we will confine ourselves to this informal treatment of `RemoveConcept`.

Example: Consider the ontologies in figure 1. Suppose that α_1 wishes to communicate $vehicle_1(a)$, $\mathcal{C}_{cv}^a = \{van_{cv}, roadvehicle_{cv}\}$.

$\alpha_1 : \text{Inform}(\alpha_2, vehicle_1(a))$

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

$\alpha_2 : \text{ReqSpec}$

$\alpha_1 : \text{AddConcept}(\alpha_2, vehicle_{cv}, vehicle_1)$

$\alpha_1 : \text{RemoveConcept}(\alpha_1, roadvehicle_{cv})$

$\alpha_1 : \text{Exit}$

$\alpha_2 : \text{Exit}$

$\alpha_1 : \text{InformExact}(\alpha_2, vehicle_1(a))$

...

In this example α_1 considers the concept $roadvehicle_{cv}$ redundant after it has added $vehicle_{cv}$. As a sender, α_1 would never use $roadvehicle_{cv}$, and as a receiver α_1 finds $vehicle_{cv}$ equally accurate as $roadvehicle_{cv}$.

5 Conclusion

In this paper we have proposed some extensions to normal communication protocols that allow agents with heterogeneous ontologies to communicate. We have focussed on *lazy* ontology alignment and *minimal cv* construction. By *lazy* ontology alignment, we mean that the agents seek *local* solutions for communication problems *when they arise*. By *minimal cv* construction, we mean that the agents come up with a simple solution, i.e. the number of concepts in the communication vocabulary remains relatively *small*.

The protocols described in this paper all implement lazy ontology alignment. With respect to minimal ontology development, P3 performs best, followed by P2, followed by P1. We will continue this line of research by considering situations with more than two agents. Furthermore, we will test the framework in some real-life scenarios of collaborating personal assistants.

References

1. F. Baader, D.L. McGuinness, and P.F. Patel-Schneider. *The description logic handbook: Theory, implementation and applications*. Cambridge University Press, 2003.
2. T. Bylander and B. Chandrasekaran. Generic tasks for knowledge-based reasoning: the right level of abstraction for knowledge acquisition. *Int. J. Man-Mach. Stud.*, 26(2):231–243, 1987.
3. J. van Diggelen, R.J. Beun, F. Dignum, R.M. van Eijk, and J.-J.Ch. Meyer. Optimal communication vocabularies and heterogeneous ontologies. In R.M. van Eijk, M.-P. Huget, and F. Dignum, editors, *Developments in Agent Communication*, LNAI 3396. Springer Verlag, 2004.
4. Michael R. Genesereth and Nils J. Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc., 1987.
5. T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
6. M. Luck, P. McBurney, and C. Preist. Agent technology: Enabling next generation computing. *Agent link community*, 2003.
7. 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)*.
8. J.-J. Ch. Meyer and W. Van Der Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge University Press, 1995.
9. 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.
10. M. Obitko and V. Marik. Mapping between ontologies in agent communication. In *Proceedings of the CEEMAS 2003, Prague, Czech Republic*, Lecture Notes on Artificial Intelligence 2691, pages 191–203. Springer-Verlag, 2003.
11. L. Steels. *Synthesising the Origins of Language and Meaning Using Co-evolution, Self-organisation and Level formation*. Edinburgh University Press, 1998.
12. Y. Tzitzikas and C. Meghini. Ostensive automatic schema mapping for taxonomy-based peer-to-peer systems. In *Proceedings of the 7th International Workshop on Cooperative Information Agents*, Helsinki, Finland, 2003.
13. M. Uschold and M. Gruninger. Creating semantically integrated communities on the world wide web. *Semantic Web Workshop Co-located with WWW 2002 Honolulu*, 2002.
14. A.B. Williams. Learning to share meaning in a multi-agent system. *Autonomous Agents and Multi-Agent Systems*, 8(2):165–193, 2004.
15. L. Wittgenstein. *Philosophische untersuchungen - Philosophical investigations*. Basil Blackwell, Oxford, german-english edition, 1953.