Acta Cybernetica **20** (2012) 515–537.

Joint Perception in Agent Communication[*]

László Z. Varga[†]

Abstract

Correctness of agent communication requires that the communicating agents share a common ontology. Most of the ontology merging approaches assume that there is a global, "god's eye" view which is a combination of the concepts in the ontology of the individual agents. These approaches admit that the agents may have different views and try to resolve the differences within the limits of the global view which contains only the concepts based on the individual perceptions of the agents. In this paper we introduce the notion of *joint perception* in order to enrich the available concepts in the global view and we introduce the notion of *conceptualization based on joint perception* in order to enable the agents to resolve the differences of their views by introducing new concepts. We propose an *incremental ontology negotiation protocol for the conceptualization based on joint perception* and demonstrate it in a blocks world. With this work we develop new insights into ontology merging and negotiation for agent communication by defining a formal realization proposal for emergent semantics.

Keywords: distributed artificial intelligence, correctness of agent communication, ontology merging and negotiation, perception and conceptualization, joint perception, conceptualization based on joint perception, ontology negotiation protocol

1 Introduction

Agents can communicate with each other only if they have a common language. This means in computing terms that agents must merge their ontology into a common ontology. Ever since more than one computer system existed, ontology merging has been a fundamental issue. In the beginning the problem to be solved was the migration of data from one system to another, then the interoperability of computer systems was in focus and recently researchers started to investigate automated ontology negotiation. Most of the approaches assume that there is a global view which contains the concepts of the systems under investigation and the goal of ontology

[*] This work was supported by TÁMOP-4.2.2/B-10/1-2010-0030 and TÁMOP-4.2.1./B-09/1/KMR-2010-0003. This work is part of COST Action IC0801: Agreement Technologies. Many thanks to the anonymous reviewers for their valuable comments that improved the paper.

[†] Faculty of Informatics, Eötvös Loránd University, E-mail: `lzvarga@inf.elte.hu`

DOI: 10.14232/actacyb.20.4.2012.4

merging and negotiation is to discover and learn the concepts not present in one agent but present in the other. If the concepts in the ontologies of the agents are not completely compatible, then the merging methods try to resolve the contradiction to achieve a consistent global view. The approaches assume that if the ontologies of the agents are merged in this way and the agents perceive the concepts correctly, then the agents are able to communicate and work together using the merged ontology. The perception of the agents is a critical point in the above reasoning and has not been studied in the same detail as the other points of ontology merging and negotiation. In this paper we are going to investigate how perception of agents influences the agents' ability to merge and negotiate their ontologies. We propose an ontology negotiation protocol to discover new concepts with the help of joint perception in ontology merging and negotiation. The proposed ontology negotiation protocol may help agent communication, however the main goal of the paper is to better understand the role of perception in ontology merging and negotiation.

1.1 Semantics in Agent Communication

In order to be able to discuss the role of perception in agent communication, we are going to use the knowledge formalization approach by Genesereth and Nilsson in [6]. Formalization of knowledge consists of a conceptualization and an ontology. The instances of the real world are first conceptualized and then formally encoded in an ontology used by the agents in their communication as shown in Figure 1.

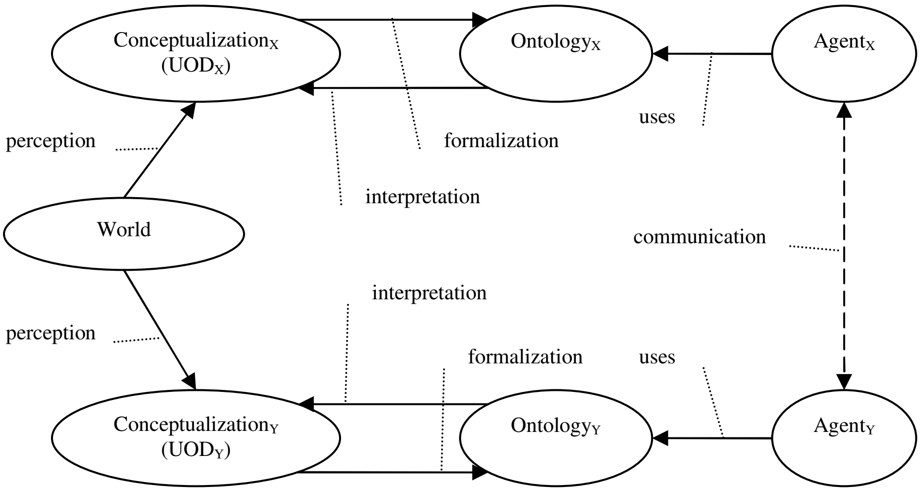

Figure 1: Semantic meaning in agent communication

When agents communicate with each other, they want to send to each other statements about the real world which is shown on the left hand side of Figure 1. The relevant instances perceived by each agent are conceptualized in the corresponding conceptualizations: Conceptualization$_X$ for Agent$_X$ and Conceptualization$_Y$ for Agent$_Y$. In accordance with Genesereth and Nilsson [6], the conceptualization consists of the universe of discourse (UOD), the functional basis set and the relational basis set. The conceptualization is informal and its elements are not formally named. There can be different conceptualizations for the same world. An example of Genesereth and Nilsson [6] for agent specific conceptualization is the wave and particle conceptualization of light, where the different conceptualizations explain different aspects of the behavior of light. In that example we can observe that a conceptualization depends not only on the interest of the agent, but also on the perception capability of the agent.

In addition to the dependence on perception capability, conceptualization is context dependent as well. Context dependence can be observed in different domains and it is expressively described in the domain of image databases by Santini et al. [10]:

> "The full meaning of an image depends not only on the image data, but on a complex of cultural and social conventions in use at the time and location of the query, as well as on other contigiencies of the context in which the user interacts with the database. This leads us to reject the somewhat Aristotelean view that the meaning of an image is an immanent property of the image data. Rather, the meaning arises from a process of interpretation and is the result of the interaction between the image data and the interpreter."

While the conceptualization differences due to agent interest differences are usually mentioned in research papers that deal with ontology merging and negotiation, the conceptualization differences due to context dependence and the limitation of the perception capabilities of the agents are not in focus. We are going to focus on this perception dependence aspect of conceptualization in this paper.

Once the agent has its conceptualization, the conceptualization is formalized in an ontology that names the objects, functions and relations of the world as perceived by the agent. The ontology is represented in a formal language. The interpretation of an ontology is a mapping between the elements of the language and the elements of the conceptualization. An ontology can be mapped to the same conceptualization in several ways and an ontology can be mapped to different conceptualizations, therefore an ontology may have several interpretations. The intended interpretation is the one that the ontology developer had in mind when he/she created the ontology. Functions and relations of an ontology are satisfied by an interpretation if they are true in the corresponding conceptualization. Although several conceptualizations may satisfy an ontology, the conceptualization designated by the intended interpretation is meant to be the conceptualization to be used when agents communicate.

Communicating agents are shown on the right hand side of Figure 1. When Agent$_X$ sends a message to Agent$_Y$, then it formalizes its message using Ontology$_X$, and the message sent from Agent$_X$ to Agent$_Y$ uses the concept names used in Ontology$_X$. Agent$_Y$ decodes the message using Ontology$_Y$. This decoding can be completed if Agent$_Y$ can find the corresponding names in Ontology$_Y$. Successful decoding of the message does not necessarily mean that the communication is semantically correct.

The communication is semantically correct only if the elements of Figure 1 match correctly, which means that if a concept c in the real world is conceptualized in Conceptualization$_X$ and formalized as c_{FX} in Ontology$_X$, then there is in Ontology$_Y$ a c_{FY} that can be mapped to c_{FX} and the intended interpretation of c_{FY} in Conceptualization$_Y$ is the conceptualization of the same c real world concept.

1.2 Ensuring Correct Semantics in Agent Communication

In order to ensure semantically correct agent communication, both agents must have a concept of the real world concept that they want to talk about. Basically this is the goal of all the ontology matching and negotiation research. The individual research reports usually focus on some of the elements of Figure 1 and assume that the others are correct.

The most studied part of Figure 1 is the right hand side, where the focus is on the formally represented ontologies. Ontology$_X$ and Ontology$_Y$ may be different, because there may be different names for the same concept in the two ontologies, or there may be different concepts in the ontologies. Semantic correctness needs that the ontologies are matched or aligned. Rahm and Bernstein [9] present a taxonomy that explains the common features of the different ontology[1] matching techniques developed in the context of ontology translation and integration, knowledge representation, machine learning, and information retrieval. An important feature of these ontology matching approaches is that they investigate the formally represented ontology on schema and instance level and try to find similarities in the formal representations based on such properties as name, description, data type, relationship types, constraints, and structure. Although the investigation of the similarities of the formal representations may indicate similarities in the concepts, the semantic correctness is not guaranteed and needs to be verified by humans. The problem with human verification is that a human is just another agent with his/her own conceptualization based on his/her own perception and we cannot be sure that this third conceptualization correctly covers and integrates the conceptualizations of Agent$_X$ and Agent$_Y$.[2]

Uschold and Gruninger [11] and Gruninger [7] propose the idea of ontological stance as a standard for semantic correctness. They go deeper than the formal representation of the ontologies and say that two ontologies are equivalent if their intended models are equivalent. Proving the equivalence of intended models is done by proving that the logical theories captured in each ontology are logically equiva-

[1] Rahm and Berstein write about schema matching, but their work can be applied to ontology matching as well.

[2] This must be one of the reasons why any attempt to create a "global ontology" of the world have failed. The ontology of Agent$_X$ and Agent$_Y$ were also created by humans, so when a third human verifies the merging of Ontology$_X$ and Ontology$_Y$, the verification is basically the same problem as merging the ontologies.

lent. Logical equivalence is verified if all statements and inferences that hold for one agent, also hold when translated into the other ontology. If the inference does not hold in the translated ontology, then the intended models are not equivalent. Unfortunately there is no procedure for generating and verifying all possible inferences for any given pair of ontologies, therefore we cannot prove semantic correctness of two ontologies, we can only prove the incorrectness if we find a conflicting inference.

The recent ontology negotiation approach includes all the elements of Figure 1. Truszkowski and Bailin [2] initiated the term ontology negotiation and recently Diggelen et al. [5] proposed an implementation of ontology negotiation for agent communication. As Williams [12] writes, a basic assumption is that both agents are able to point at instances in the real world and this can be known for both of them. Agent$_X$ points at an instance in the real world and sends the formal representation of this instance in Ontology$_X$ to Agent$_Y$. Agent$_Y$ can see the instance in the real world and find its formal representation in Onology$_Y$, and thus can create a mapping between Ontology$_X$ and Ontology$_Y$. The mapping method can be supported by a learning method as Williams [12] proposes or explanation based as Diggelen et al. [5] propose. Because agents point at instances in the real world, the ontology negotiation approach includes the perception part of Figure 1 as well. A basic assumption of ontology negotiation is that the agents do not have any errors in their perception of the world although their perceptions may differ. This is necessary for a successful ontology negotiation, but is it enough? Can we be sure that agents can negotiate successfully if they do not have error in their perception? In the next sections of this paper we will investigate this as well.

In the case of the above three ontology merging and negotiation approaches, the differences in the ontologies are due to the different categorizations by the agents and the goal is to match these categorizations. If the perceptions of the agents may have errors, then the agents have to eliminate the conflicting facts as well. In the example of Cholvy [3] one witness saw a dark blue car on the crime scene, while the other saw a dark green car and there were two men in it. In this case a unique consistent view of the world can be achieved by dropping some of the perceptions and keeping other perceptions. The selection is often helped by preference relations like in the work of Amgoud and Kaci [1]. In the rest of the paper we will assume that the perceptions of the agents are correct according to their conceptualization of the world.

2 The Perception Problem in a Blocks World

The above overview of ontology merging and negotiation indicates that merged ontologies and error free perceptions are needed for correct agent communication. Now we are going to investigate this in a blocks world example. Although the blocks world example below in this section has some kind of image processing flavor, we are not focusing on image processing. We are using images only because they are expressive. At the end of section 2 we will show that the blocks world example has similarities with other domains as well.

2.1 Perception Capabilities in a Blocks World

When we assume that the perception of the agent is error free, then we assume that there is an unambiguous mapping from the real world to the perception of the agent. This means that if the real world instance is in the sensing range of the agent, then the agent perceives the instance and the same real world instance always maps to the same perception of the agent.

Definition 1. *Error Free Perception: the perception of the agent is error free, if the agent perceives every real world instance that gets in the sensing range of the agent, and the same real world instance always maps to the same perception of the agent.*

We are going to investigate agents in a blocks world example, where the perception of the agents is through an image caption sensor. The sensor is able to make camera images of the real world. Figure 2 shows the perception of the real world by Agent$_X$. The axes x and z are not part of the perception, they are on the figure just to show the orientation. The conceptualization of this world by Agent$_X$ consists of three blocks, no functions and two relations (square and circle) corresponding to the shape of the blocks. The formalization of this conceptualization is Ontology$_X$ and in accordance with the formalization approach by Genesereth and Nilsson [6] it is the following[3]:

$$< \{a, b, c\}, \{\}, \{square_X, circle_X\} > \tag{1}$$

Agent$_X$ has the following representation of the current state of the world:

$$square_X(a), circle_X(b), square_X(c) \tag{2}$$

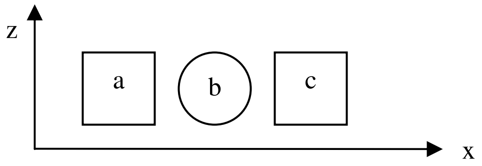

Figure 2: Perception of the blocks world scene by Agent$_X$

Figure 3 shows the perception of the same scene of the real world by Agent$_Y$. Agent$_Y$ has exactly the same type of sensors and sensing capabilities as Agent$_X$, but Agent$_Y$ has a different view. The axes y and z are not part of the perception, they are on the figure just to show the orientation. The conceptualization of Agent$_Y$ is the same as that of Agent$_X$ and its formalization is the same (except that the symbols are differentiated with the y index):

$$< \{a, b, c\}, \{\}, \{square_Y, circle_Y\} > \tag{3}$$

[3] We call this formal representation as Ontology$_X$, although it is not a complex sophisticated ontology.

However Agent$_Y$ has the following representation of the current state of the world:

$$square_Y(a), square_Y(b), circle_Y(c) \tag{4}$$

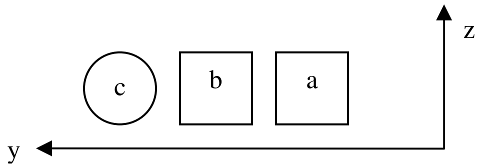

Figure 3: Perception of the blocks world scene by Agent$_Y$

We assume that both agents are able to point at the blocks in the real world and this can be perceived by both of them. Technically this could be implemented for example by sending a radio signal to the selected block and if the selected block receives the signal, then it emits light which is perceivable by both agents. So if Agent$_X$ wants to point at block a, then it sends a signal to block a and Agent$_Y$ perceives that block a is glowing. In this blocks world example the letters denote the same blocks, i.e. block a perceived by Agent$_X$ on Figure 2 is the same as block a perceived by Agent$_Y$ on Figure 3, block b perceived by Agent$_X$ is the same as block b perceived by Agent$_Y$, and block c perceived by Agent$_X$ is the same as block c perceived by Agent$_Y$.

2.2 Ontology Merging and Agent Communication in the Blocks World

We are now investigating how the different ontology merging techniques cope with the above blocks world example. In section 1.2 we have seen that there are three major approaches: the merging technique based on the formally represented ontologies (schema level matching), the verification method based on the logical equivalence of the logical theories captured in each ontology, and the ontology negotiation.

Claim 1. *Schema level matching and error free perception are not enough for conflict free agent communication.*

Proof. The merging technique based on the formally represented ontologies and systems (see Rahm and Bernstein [9]) can be on the schema or on the instance level.

The schema level matching in our blocks world example would result in stating that the ontologies of Agent$_X$ and Agent$_Y$ match each other, because their formal representation (1) and (3) have the same structure. After investigating the technical capabilities of the sensors of the agents and their processing software, the ontology merger would say that *square*$_X$ maps to *square*$_Y$ and *circle*$_X$ maps to *circle*$_Y$, because the agents have exactly the same type of sensors. Although both agents have error free perception and their ontologies are matched, the agents would have problems in their communication, because if Agent$_X$ sends the message *circle$_X$(b)* to Agent$_Y$, then it would be mapped to *circle$_Y$(b)* and it would conflict with the *square$_Y$(b)* information of Agent$_Y$. So this example shows that schema level matching and error free perception are not enough for conflict free agent communication.

If schema merging is combined with instance level merging, then the above conflict on the shape of block b can be found already at the ontology merging phase. The instance level merger would not be able to find out how to map *square$_X$* to the concepts of Agent$_Y$, because in the case of block a the symbol *square$_X$* maps to *square$_Y$*, but in the case of block c the symbol *square$_X$* maps to *circle$_Y$*. The same way, the instance level merger would not be able to find out how to map *square$_Y$* to the concepts of Agent$_X$, because in the case of block a the symbol *square$_Y$* maps to *square$_X$*, but in the case of block b the symbol *square$_Y$* maps to *circle$_X$*. Therefore instance level ontology merging would fail. □

Claim 2. *The verification method based on the logical equivalence of the logical theories captured in each ontology and error free perception are not enough for conflict free agent communication.*

Proof. The ontology merging verification approach of Gruninger [7] is based on logical theories captured in the ontologies. In the case of the above blocks world example there is no complex theory captured in the formal representation of the conceptualizations, because there are no inference rules for the blocks. Therefore we can use basic statements about the state of the blocks world and general logic to verify the equivalence of the ontologies of Agent$_X$ and Agent$_Y$. Let us investigate the following expressions:

$$square_X(a) \wedge square_X(c) \tag{5}$$

$$square_Y(a) \wedge square_Y(c) \tag{6}$$

$$circle_Y(a) \wedge circle_Y(c) \tag{7}$$

Expression (5) states that blocks a and c are both squares. Agents are able to point at the blocks, so they can identify blocks a and c. Statement (5) is evaluated true by Agent$_X$. The *square$_X$* symbol can be translated either to *square$_Y$* or *circle$_Y$*. If *square$_X$* is translated to *square$_Y$*, then we get expression (6) which is evaluated false by Agent$_Y$. If *square$_X$* is translated to *circle$_Y$*, then we get expression (7) which is again evaluated false by Agent$_Y$. So expression (5) holds for Agent$_X$, but it does not hold in any translation into the formalization of Agent$_Y$. This means that the intended models of Agent$_X$ and Agent$_Y$ are not equivalent, the symbols of Agent$_X$ cannot be mapped to Agent$_Y$ and their ontologies cannot be merged. □

Claim 3. *Ontology negotiation and error free perception are not enough for conflict free agent communication.*

Proof. The ontology negotiation approach to ontology merging is somewhat similar to the instance level merging of the formally represented ontologies, but the merging is done at runtime by the agents instead of offline investigation of the formal representations. The agents point at an instance in the real world and send the representation of the instance to the other agent. In the case of the above blocks world example, Agent$_X$ would not be able to negotiate how to map $square_X$ to the concepts of Agent$_Y$, because when it points at block a and sends the symbol $square_X$ to Agent$_Y$, then Agent$_Y$ would map $square_X$ to $square_Y$, but when Agent$_X$ points at block c and sends the symbol $square_X$ to Agent$_Y$, then Agent$_Y$ would map $square_X$ to $circle_Y$. The same way, Agent$_Y$ would not be able to negotiate how to map $square_Y$ to the concepts of Agent$_X$, because in the case of block a the symbol $square_Y$ would map to $square_X$, but in the case of block b the symbol $square_Y$ would map to $circle_X$. Therefore ontology negotiation would fail. □

Remark 1. In our simple blocks world example there are only few instances and we could easily find a mismatch in the instances in instance level schema matching or ontology negotiation, as well as conflicting statements in logic based verification. However in a complex application there may be too many instances and these types of mismatches may remain undiscovered until there is a conflict in the communication of the agents.

2.3 Why Ontology Merging Fails in the Blocks World

In the previous section we have two identical agents with two different views of the same blocks world, and the merging of their ontologies fails and the agents are not able to correctly communicate. The schema level ontology merging based on the formally represented ontologies and systems succeeds, but agent communication will not be correct. The other ontology merging approaches do not succeed in merging the ontologies, although the agents are identical. How can this be, and how can the two agents have so different perception of the same blocks world? The explanation is in the limited perception capabilities of the agents.

Definition 2. *Limited Perception Capability: given an agent that can perceive an application domain from different contexts and the perception of the agent is error free in each context, then an agent has limited perception capability if there is at least one application domain instance which maps to different perceptions of the agent in different contexts.*

Claim 4. *Agents with error free perception capabilities may not be able to resolve conflicting perceptions by choosing one of their already existing concepts, if the perception capabilities of the agents are limited.*

Proof. Figure 4 shows how the agents view the same blocks world. There are two cylinders and a cube in the blocks world. Agent$_X$ perceives this blocks world's projection on the x-z plane (Figure 2) and Agent$_Y$ perceives this blocks world's projection on the y-z plane (Figure 3). Agent$_X$ perceives cylinder b as $circle_X(b)$

and Agent$_Y$ perceives cylinder b as $square_Y(b)$. The concepts perceived by the agents are in line with the concepts of their perception devices, but not with the concepts of the real world as seen by the humans. The cylinder may be perceived by the perception devices either as a circle or a square depending on the position of the agents, and the agents are not aware of the three dimensional nature of the blocks. This limited perception capability is the root of the problems in the discussed ontology merging approaches and agent communication. □

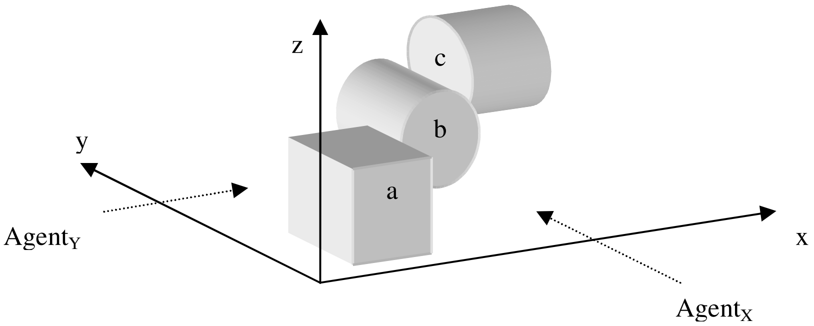

Figure 4: The blocks world scene in 3 dimensions

As long as the agents keep to the concepts of their perceptions, they will have conflicts. If they try to resolve the conflict with an argumentation framework which is based on the existing concepts of the agents, like that of Amgoud and Kaci [1], then one of the agents will be regarded more reliable than the other and the perception of the more reliable agent will win. However in this case none of the perceptions are better than the other, therefore eliminating the conflict between the agents by dropping one of the statements will not help to have a better perception of the blocks world.

2.4 The Blocks World and other Domains

We intentionally used the toy example of the blocks world in this paper, because our goal here is to have a fundamental understanding of the role of perception in ontology merging and negotiation. Once we have a clear understanding, then later ontology merging and negotiation techniques can be improved to handle more complex situations and huge amount of data.

One may think that the above blocks world example is too specific and not realistic. This is not the case, because we can easily create similar examples for the semantic concept learning application of Williams [12] in the World Wide Web domain: Agent$_X$ is a historian expert and Agent$_Y$ is a computer virus expert. They both have the concept of "*web page of professional interest*" and the concept of "*professionally non interesting web page*". The "*web page of professional interest*" concept corresponds to the *square* concept and the "*professionally non interesting* *web page*" concept corresponds to the *circle* concept in the blocks world. A web page with title "The Trojan Horse" is similar to the *cube* in the blocks world, because both the historian and the computer expert may classify this page as "*web page of professional interest*". A web page with the title "History of Ancient Greece" is similar to the *cylinder* in the blocks world, because the historian may classify this page as "*web page of professional interest*", while the computer expert classifies this page as "*professionally non interesting web page*".

Apart from the blocks world example and its analogy above, the described phenomenon is in the heart of almost every data integration project where different representations must be merged. The views of the different developers may be different, therefore the developers of one system may identify the relevant features of a concept in a different way from the developers of the other system. This means that there may be a mismatch between the perceptions of the different developers and the concepts in the real world. For example, if there are two systems (X and Y) and the developers of both systems want to represent people and houses. The developers of system X find that the relevant features of a person are its name and social security number, while the relevant features of a house are the name of its owner, its address and the date when it was built, so they perceive the person as a *(name, number)* pair and the house as a *(name, address, date)* triple. The developers of system Y find that the relevant features of a person are its name, its address and its date of birth, while the relevant features of a house are the name of its owner and its topographical number, so they perceive the person as a *(name, address, date)* triple and the house as a *(name, number)* pair. We can see that this is similar to the blocks world example: the house corresponds to cylinder *b* and the person corresponds to cylinder *c*. The limited two dimensional perception capability is the internal representations in systems X and Y in the following way: *circle* corresponds to the *(name, address, date)* triple and *square* corresponds to the *(name, number)* pair.

Obviously the developers of system X and Y can easily understand the above toy problem and explain the differences of the concepts and their representations to each other, then add the necessary new representations and create the necessary mappings between the two systems. If there are more complex concepts and internal representations, then the developers may have difficulties in understanding and explaining the differences, therefore they need automated methods.

As we said before, the perception capability depends on the context as well, like in the case of image databases in Santini et al. [10]. If the image of a painted portrait is placed among images of other paintings (some of which are portraits and some of which are not), then an automated tool would label the images, among them the portrait, with "painting". If the image of the portrait is placed among photos and paintings of faces, then the automated tool would label the images with "face". Both perceptions are good in their context, however if we want to resolve the difference of the labellings, then the best result can be achieved if we take into account both perceptions. In the following we are going to discuss this kind of joint perception.

3 Conceptualization Based on Joint Perception

The above blocks world example clearly demonstrates that the success of ontology merging greatly depends on the perception of the agents. If the conceptualization of an agent does not describe the real world in a way that includes all the aspects necessary for the successful communication between the agents, then ontology merging fails. Although the conceptualization may be enough for a single agent to execute its own tasks, the pair of agents will not understand each other. Because the concepts in the conceptualizations of the individual agents cannot describe the real world in this case, a new conceptualization is needed. The new conceptualization may contain the concepts of the individual agents, but it should contain additional concepts as well. The new concepts are developed by combining the different views of the agents, which we call conceptualization based on joint perception.

Wooldridge [13] defines perception as the agent's capability to observe its E environment with the help of the *see* function and map it to a set of Per perceptions:

$$see : E \rightarrow Per \tag{8}$$

In accordance with Genesereth and Nilsson [6], the perception without formalization is the conceptualization of the agent. The formal representation of the perception follows the formalism of the ontology of the agent. Based on this, we define joint perception as two agents' capability to jointly observe their shared environment:

Definition 3. *Joint Perception: Given the E environment in which two agents $Agent_X$ and $Agent_Y$ observe the environment with the help of their see_X and see_Y functions and map the environment to two sets of perceptions Per_X and Per_Y:*

$$see_X : E \rightarrow Per_X \tag{9}$$

$$see_Y : E \rightarrow Per_Y \tag{10}$$

and the agents can communicate to each other the formalization of their perceptions with the $send_X$ and $send_Y$ functions,

then we define joint perception as the agents' capability to observe the E environment with the help of their modified see_{Xjoint} and see_{Yjoint} functions and map the environment to the Cartesian product of their own perception and the communicated perception of the other agent:

$$see_{Xjoint} : E \rightarrow Per_X X send_Y(Per_Y) \tag{11}$$

$$see_{Yjoint} : E \rightarrow Per_Y X send_X(Per_X) \tag{12}$$

The Cartesian product of the agent's own perception and the communicated perception of the other agent is called the **conceptualization based on joint perception**.

Note that the modified *see* function of the agents involves communication with the other agent, therefore the mapping result of the see_{Xjoint} and see_{Yjoint} functions cannot be determined by a single agent, but by the agents together within the framework of the **ontology negotiation protocol of the conceptualization based on joint perception** that we are going to discuss in the following sections.

With the above definition we have formally defined what Cudré-Mauroux [4] writes on emergent semantics:

> "This is a novel way of providing semantics to symbols of agents relative to the symbols of other agents with which they interact."

3.1 Ontology Negotiation for Joint Perception

Diggelen et al. [5] assume that in the ontology negotiation process there is a "god's eye view" of the conceptualizations which is the union of the individual conceptualizations of the agents. However in the above blocks world example we can enable successful agent communication only by adding new concepts to the "god's eye view": the "god's eye view" is the three dimensional view which is not perceivable by any of the agents and contains the new concept of the three dimensional cylinder.

Now we are going to extend the ontology negotiation framework of Williams [12] with a modification of the ontology negotiation protocol. We assume that both agents are able to point at instances in the real world and this can be perceived by both of them, so the agents can refer to the instances with the same instance name.

The **ontology negotiation protocol of the conceptualization based on joint perception** consists of the following steps:

1. Agent$_X$ sends the name of one of its semantic concepts, the names of a set of instances of the semantic concept in the real world and points at the instances[4] in the real world. Agent$_X$ repeats this message for all its semantic concepts and the corresponding sets of the instances. In the blocks world example Agent$_X$ sends the symbol $square_X$, the names a and c, and points at blocks a and c. Then Agent$_X$ sends the symbol $circle_X$, the name b and points at block b.

2. Agent$_Y$ receives the semantic concept names, the instance names and observes the instances in the real world to find the corresponding semantic concept names in its internal representation. In the blocks world example Agent$_Y$ finds that it knows that blocks a and b are $square_Y$, and block c is $circle_Y$.

3. Agent$_Y$ builds up a joint concept name table that contains all combinations of Agent$_X$ concept names and Agent$_Y$ concept names, with observed instances. Agent$_Y$ assigns new joint concept names to each row of this table. Table 1 shows this for the blocks world example. A joint concept name can be any

[4] A conceptualization consists of an universe of discourse, a functional basis set and a relational basis set. While pointing at an object is relatively easy, ponting at a functional or relational semantic concept needs further technical details of the protocol, because the agent has to point at the tuples describing the functional or relational samples. In the case of the blocks world example it is relatively easy, because we have only unary relational concepts like $square_X\,(a)$.

unique machine generated name, but in this blocks world example we use cube and cylinder to have correspondence with the three dimensional objects.

4. Agent$_Y$ sends the joint concept table to Agent$_X$. Agent$_X$ receives the joint concept table and incorporates the new concept names into its representation by assigning the new concept names to the real world instances. As a result, the semantic names of the concepts will be changed in the local ontology of Agent$_X$. Agent$_X$ confirms this update to Agent$_Y$.

5. Agent$_Y$ receives the confirmation and incorporates the new concept names into its local ontology, too. From this point the agents can use in their communication the new semantic names, because they are unambiguous. This means that the agents collaboratively learnt new concepts and identified the instances of the new concept based on their joint perception. These new concepts were previously unknown to them.

Table 1: Joint concept table based on joint perception.

Instances	Agent$_X$concept	Agent$_Y$concept	Joint concept
a	square$_X$	square$_Y$	cube
b	circle$_X$	square$_Y$	cylinder$_X$
c	square$_X$	circle$_Y$	cylinder$_Y$
—	circle$_X$	circle$_Y$	—

As Table 1 shows, the agents in the blocks world example learn the concept of the three dimensional cylinder under two new concepts names: *cylinder$_X$* and *cylinder$_Y$* and identified the instances of these concepts: *b* and *c* correspondingly. Although for a human observer in the three dimensional world these two types of objects are the same type of objects with different orientations, the agents assign them two different semantic names, because the joint perception of the agents is not three dimensional and the agents perceive two projections of the three dimensional space. This means that the concept names *cylinder$_X$* and *cylinder$_Y$* include the shape and the orientation of the three dimensional object.

The last row of Table 1 does not have any sample, therefore a semantic name is not assigned to this row. If there were a sphere in the three dimensional space, then this row would be complete.

3.2 Complexity

The conceptualization based on joint perception has the same drawback as the instance level ontology merger approaches: in a complex application there may be too many instances to check. In addition to that, the number of concepts may increase the complexity as well, so we are going to investigate this.

If the formalization of the conceptualization of Agent$_X$ contains n semantic concept names and the formalization of the conceptualization of Agent$_Y$ contains m semantic concept names, then the number of rows in the joint concept table will be $n * m$. In order to build up this table, Agent$_X$ has to send n messages with n different semantic concept names to Agent$_Y$. Agent$_Y$ responds to Agent$_X$ with the joint concept table in one message containing all the maximum $n * m$ joint concept names. Altogether the messages of the proposed ontology negotiation protocol are proportional to n. Agent$_Y$ has to find its own semantic concept name for each sample and place the sample in the corresponding row of the joint concept name table, so the computation needed to construct the joint concept name table by Agent$_Y$ is proportional to the samples in the real world.

The ontology negotiation protocol of Williams [12] has similar complexity, because in that protocol the querying agent has to send samples for each concept name to be negotiated to the other agent, and the other agent has to decide if it can find samples for the same concept.

3.3 Ontology Negotiation as Needed

If the agents want to explore all possibilities and send to each other all concept names and their sample instances, then the joint concept table would contain all instances, as shown in Table 1. In a complex application this would be too large to send in a message, therefore we are going to modify the joint perception based ontology negotiation protocol with the lazy (or incremental) ontology alignment approach of Diggelen et. al [5]. The agents are not going to discover the whole concept space before they start communicating. Instead of that, the agents discover new concepts jointly when it is needed and they adjust their ontologies at the time when they find a mismatch in the concepts. When they discover new concepts, they incrementally solve the ontology merging problem and avoid that the reference to all instances are sent from one agent to the other.

The conceptual framework of Diggelen et. al [5] contains several ontologies for the incremental ontology alignment approach. O$_X$ and O$_Y$ are the local ontologies of the agents that want to align their ontologies in order to be able to communicate correctly. O$_{cv}$ is the communication vocabulary ontology which contains the concepts that both agents understand and use for communication. O$_{X-cv}$ is the combination[5] of O$_X$ and O$_{cv}$ and contains the mappings from the concepts of O$_{cv}$ to the concepts of O$_X$. Similarly, O$_{Y-cv}$ contains the mappings from the concepts of O$_{cv}$ to the concepts of O$_Y$. O$_{X-Y}$ is the combination of O$_X$ and O$_Y$ and contains the concepts from both agent's ontologies in a god's eye view manner. The assumption of the framework is that a) O$_{X-Y}$ contains the union of the semantic symbols of O$_X$ and O$_Y$, b) there are subset orderings of the intended interpretations of the semantic symbols in O$_{X-Y}$, O$_X$ and O$_Y$, and finally c) the subset ordering in O$_{X-Y}$ conforms to the subset ordering of O$_X$ and O$_Y$. We will refer later to these assumptions as the "subset ordering assumption".

[5] Please note that the hyphen in O$_{X-cv}$ denotes combination and not extraction.

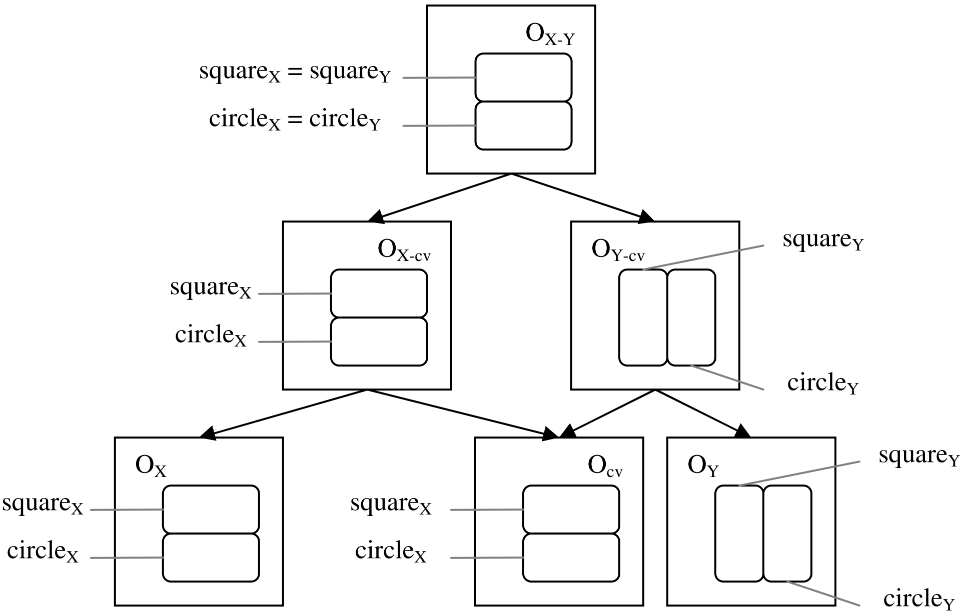

Figure 5: Ontologies according to schema level formal ontology merging in the blocks world example

Figure 5 shows these ontologies of Agent$_X$ and Agent$_Y$ in the blocks world example when their ontologies are merged with schema level formal ontology merging. As we said before, the schema level formal ontology merging would result in saying that the code of the agents are identical, therefore there are two concepts that are common to the agents: $\mathit{square}_X = \mathit{square}_Y$ and $\mathit{circle}_X = \mathit{circle}_Y$. This is the god's eye view and is in the O$_{X-Y}$ ontology. The arrow from O$_{X-Y}$ to O$_{X-cv}$ indicates that there is a mapping from the concepts in O$_{X-Y}$ to the concepts in O$_{X-cv}$: the $\mathit{square}_X = \mathit{square}_Y$ concept in O$_{X-Y}$ is mapped to the square_X concept in O$_{X-cv}$ and the $\mathit{circle}_X = \mathit{circle}_Y$ concept in O$_{X-Y}$ is mapped to the circle_X concept in O$_{X-cv}$. Similarly the $\mathit{square}_X = \mathit{square}_Y$ concept in O$_{X-Y}$ is mapped to the square_Y concept in O$_{Y-cv}$ and the $\mathit{circle}_X = \mathit{circle}_Y$ concept in O$_{X-Y}$ is mapped to the circle_Y concept in O$_{Y-cv}$.

The agents could use for example the symbols square_X and circle_X to refer to the common concepts in their communication vocabulary. This is shown in the O$_{cv}$ ontology. The arrow from O$_{X-cv}$ to O$_{cv}$ indicates that there is a mapping from the concepts in O$_{X-cv}$ to the concepts in O$_{cv}$: the square_X concept in O$_{X-cv}$ is mapped to the square_X concept in O$_{cv}$ and the circle_X concept in O$_{X-cv}$ is mapped to the circle_X concept in O$_{cv}$. The arrow from O$_{Y-cv}$ to O$_{cv}$ indicates that there is a mapping from the concepts in O$_{Y-cv}$ to the concepts in O$_{cv}$: the square_Y concept in O$_{Y-cv}$ is mapped to the square_X concept in O$_{cv}$ and the circle_Y concept in O$_{Y-cv}$ is mapped to the circle_X concept in O$_{cv}$.

Note that the schema level formal ontology merging does not take into account the instances, therefore cannot check the assumption on the subset ordering of the intended interpretation of the semantic symbols. However if we take into account the instances and the intended interpretations as described in section 2.1, then we see that although the subset ordering assumption holds for O_X and O_Y, it does not hold for O_{X-Y}, because the sets $square_X$ and $circle_X$ are disjoint in O_X, therefore the sets $square_X = square_Y$ and $circle_X = circle_Y$ should be disjoint as well, but for example block b would be a member of both sets. This is why instance level ontology merging as well as ontology negotiations, as discussed in section 2.2, do not succeed. If we keep to the subset ordering of the original ontologies, then the agents cannot put into the merged ontology new concepts that do not conform to the original subset ordering. This means that the agents cannot discover such new concepts with the help of their joint perception capability.

Now we are going to extend the ontology negotiation protocol of the conceptualization based on joint perception (described in section 3.1) to support the incremental ontology negotiation approach of Diggelen et. al [5]. Because we want to include in the extension the possibility of learning new concepts previously unknown to the agents, we cannot keep to the subset ordering assumption and cannot directly use the ontology negotiation protocol of Diggelen et. al [5]. We will assume that the negotiation protocol of Diggelen et. al [5] will be used in the first place to determine the mapping between the O_{cv} communication vocabulary and the local ontology of the agents when there is a subset ordering of the concepts of the negotiating agents. The negotiation protocol we propose here will go to a new branch to determine a new concept when the subset ordering of the concepts of the negotiating agents does not apply or a concept mismatch is detected during communication.

Basically the incremental ontology negotiation protocol works in the following way: one of the agents proposes a concept to be added to O_{cv} and then the agents negotiate the mapping between the O_{cv} and the local ontology of the other agent. This mapping is ambiguous when the individual perceptions of the agents do not describe the real world properly and a new concept needs to be discovered based on the joint perception. Let us take the blocks world example. Agent_X proposes to add the concept $square_X$ to O_{cv}. As long as Agent_X points at only block a type of samples, Agent_Y will map the concept $square_X$ to $square_Y$, because the perception of block a type of samples by Agent_Y is $square_Y$. The result will be $square_X = square_Y$ like in the case of schema level formal ontology merging on Figure 5. However if Agent_X starts to teach its $square_X$ concept with block c type of samples only, then Agent_Y will map the concept $square_X$ to $circle_Y$, because the perception of block c type of objects by Agent_Y is $circle_Y$. Both mappings may be sufficient for the communication of the agents as long as no instances of the $square_X$, $square_Y$ and $circle_Y$ concepts other than those used for the creation of the mapping appear in their communication. If another type of instance appears in the communication, then the new concept learning based on joint perception comes in.

The **incremental ontology negotiation protocol of the conceptualiza-**

tion based on joint perception consists of the following steps:

1. Agent$_X$ proposes to add concept name c_i to O$_{cv}$. If Agent$_Y$ is able to map concept name c_i into O$_{Y-cv}$, then the agents continue the communication (in step 3) or add other concepts to O$_{cv}$ (this step 1 is repeated).

2. If Agent$_Y$ is not able to map concept name c_i to O$_{cv}$, then the agents start a new concept discovery based on joint perception (in step 4, where c_i will be denoted by c_x).

3. The agents continuously communicate with each other. If the concepts in O$_{cv}$ correctly describe the real world for the communication, then there is no problem and normal communication goes on (this step 3 is repeated). If the agents want to extend O$_{cv}$, then they go to step 1 again. If the concepts in O$_{cv}$ do not describe correctly the real world for the communication, then at some time one of the agents, let's say Agent$_X$, sends a message to the other agent, in this case to Agent$_Y$, and the message refers to a real world instance o_x of a concept c_x, the concept name c_x is in O$_{cv}$ and mapped to c_y in O$_{Y-cv}$, however Agent$_Y$ discovers that according to its own perception o_x is not in concept c_y, rather in concept c_{y2}. In this case the agents start a new concept discovery based on joint perception (in step 4).

4. (The concept name c_x now denotes the conflicting concept: if we arrived here from step 2, then c_x denotes c_i of step 2, if we arrived here from step 3, then c_x denotes c_x of step 3.) Agent$_Y$ sends a message to Agent$_X$ and asks Agent$_X$ to show instances of concept c_x.

5. Agent$_X$ sends the names of a set of instances of the semantic concept c_x in the real world and points at the instances in the real world.

6. Agent$_Y$ receives the instance names and observes the instances in the real world to find the corresponding semantic concept names in its internal representation.

7. Agent$_Y$ builds up a joint concept name table that contains all combinations of c_x and Agent$_Y$ concept names, with instance names from Agent$_X$. If a new row is added to the joint concept name table, then Agent$_Y$ assigns new joint concept names to each new row of this table. A joint concept name can be any unique machine generated name.

8. The joint concept name table is permanently kept by each agent and updated each time a new concept discovery is completed. Each time the new concept discovery protocol is executed, only the newly added or modified rows are communicated by the agents in order to keep this table synchronized.

9. Agent$_Y$ sends the newly added rows of the joint concept table to Agent$_X$. Agent$_X$ receives the new rows of the joint concept table and incorporates the new concept names into O$_{X-cv}$. As a result some of the instances will have new semantic name. Agent$_X$ confirms this update to Agent$_Y$.

10. Agent$_Y$ receives the confirmation and incorporates the new concept names into O$_{Y-cv}$, too. This means that the agents collaboratively learnt new concepts based on their joint perception and at the same time jointly identified instances of the new concept as well, therefore the agents can refer to these instances in their future communication using the new concept name. These new concepts and the categorisation of the instances to these concepts were previously unknown to them. The agents add the new concepts to O$_{cv}$ and at the same time delete c_x from O$_{cv}$, because c_x is replaced by the new ones. From this point the agents can continue the communication using the new semantic names (step 3) and identify instances of the new semantic concepts using the joint concept name table.

As an example, let's see how the above incremental ontology negotiation protocol of the conceptualization based on joint perception works in the blocks world of section 2. A sample scenario is the following:

1. Agent$_X$ proposes to add concept name *square*$_x$ to O$_{cv}$ and points at block *a*. Agent$_Y$ maps *square*$_x$ to *square*$_y$ in O$_{Y-cv}$.

2. The agents start to communicate with each other. At some time Agent$_X$, sends a message to Agent$_Y$, and the message refers to block *c* of the concept *square*$_x$. The concept name *square*$_x$ is in O$_{cv}$ and mapped to *square*$_y$ in O$_{Y-cv}$, however Agent$_Y$ discovers that according to its own perception, block *c* is in concept *circle*$_y$.

3. Agent$_Y$ sends a message to Agent$_X$ and asks Agent$_X$ to show samples of concept *square*$_x$.

4. Agent$_X$ sends the names of block *a* and *c* in the semantic concept *square*$_x$ and points at the sample instances in the real world.

5. Agent$_Y$ receives the instance names and observes the instances in the real world to find the corresponding semantic concept names in its internal representation.

6. Agent$_Y$ builds up a joint concept name table that contains all combinations of *square*$_X$ and Agent$_Y$ concept names, with sample instance names from Agent$_X$. Agent$_Y$ assigns new joint concept names to each row of this table as shown in Table 2 below. Note that Table 2 contains the categorization of the blocks *a* and *c* as well.

7. Agent$_Y$ sends the newly added rows of the joint concept table (in this case the whole table is new) to Agent$_X$. Agent$_X$ receives the new rows of the joint concept table, stores the rows of the joint concept table in its own copy of the joint concept table and incorporates the new concept names into O$_{X-cv}$. Agent$_X$ confirms this update to Agent$_Y$.

8. Agent$_Y$ receives the confirmation and incorporates the new concept names into O$_{Y-cv}$, too. This means that the agents collaboratively learnt the new concepts *cube* and *cylinder$_Y$* together with their instances based on their joint perception and the ontologies are updated as shown in Figure 6. From this point the agents can continue the communication using the new semantic names and identify the instances of the new semantic concepts using the joint concept table.

Table 2: Joint concept table based on incremental joint perception.

Instances	Agent$_X$concept	Agent$_Y$concept	Joint concept
a	square$_X$	square$_Y$	cube
c	square$_X$	circle$_Y$	cylinder$_Y$

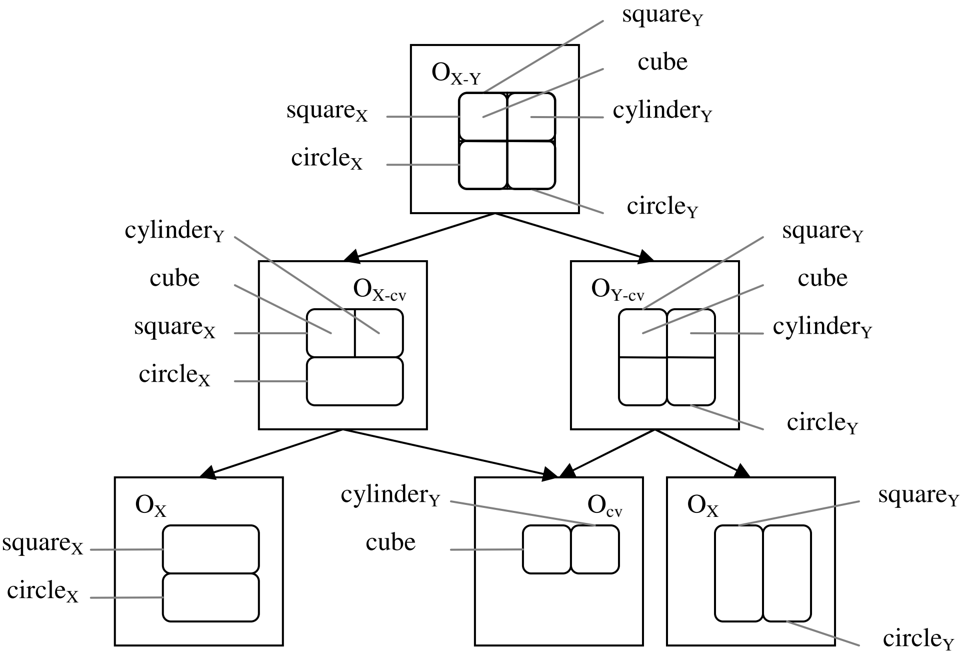

Figure 6: Ontologies of the agents after an incremental joint perception discovery cycle in the blocks world

In accordance with Table 2, in Figure 6 the O$_{cv}$ communication vocabulary ontology contains the newly discovered concepts *cube* and *cylinder$_Y$*. Both *cube* and *cylinder$_Y$* are included in the *square$_X$* concept in O$_{X-cv}$. In O$_{Y-cv}$ *cube* is included in *square$_Y$* and *cylinder$_Y$* is included in *circle$_Y$*. O$_{X-Y}$ is the merged ontology of the two agents, therefore it contains *square*$_X$ (horizontal rounded rectangle in the figure), *circle*$_X$ (horizontal rounded rectangle in the figure), *square*$_Y$ (vertical rounded rectangle in the figure), *circle*$_Y$ (vertical rounded rectangle in the figure), as well as the new concepts: *cube* as the intersection of *square*$_X$ and *square*$_Y$, *cylinder*$_Y$ as the intersection of *square*$_X$ and *circle*$_Y$.

4 Conclusions

Two agents in a multi-agent environment can communicate correctly if they share a common ontology. We can create this common ontology from the concepts perceived by the agents only if the individual perceptions of the agents correctly describe the world from both agents' view. There are two reasons why we cannot expect that the perceptions of the agents are perfect. One reason is that agents have limited perception capabilities which may be enough to perform their own tasks, but may not be correct from the point of view of the other agent. The other reason (e.g. Santini et al. [10]) is that perception is not an abstract and objective action independent from the observer, because perception depends on the complete context of the observation including the history before and after the observation, the environment of the observation, the observer and the interaction between the observer and the observed object.

So if perception is not an abstract action depending only on the perceived object, then we cannot expect that the individual perceptions of the agents always correctly describe the real world for both agents, therefore if we want to describe the world in a way that is correct from both agents' view, then we have to base the common conceptualization of the agents on the perception of both agents. This is why we introduced in this paper the notions of *joint perception* as well as *conceptualization based on joint perception.* We developed the *ontology negotiation protocol of the conceptualization based on joint perception* as an extension to the ontology negotiation framework of Williams [12]. In order to reduce instant resource usage of this ontology negotiation protocol, we developed the *incremental ontology negotiation protocol of the conceptualization based on joint perception* and showed how it fits in the incremental ontology negotiation approach of Diggelen et. al [5]. To our knowledge, this is the first work that actually describes how to create new concepts in ontology merging and negotiation for agent communication, therefore this is the first formal realization proposal for the viewpoints of Cudré-Mauroux et al. [4] on emergent semantics. In a similar way as the notion of joint intention of Jennings [8] helped to better understand cooperation in the multi-agent world, we hope that the notion of joint perception gives better insight into the role of perception in ontology merging and negotiation in multi-agent systems.

With the help of the ontology negotiation protocol of the conceptualization based on joint perception the agents can create concepts that are in line with the perceptions of both agents, therefore the ontologies of the agents can be merged into a common ontology that is suitable for both agents and the agents can correctly communicate with each other when they refer to the jointly identified concepts or the instances of the new concepts. Although we get a common ontology with the proposed ontology negotiation protocol, the disadvantage of the proposed approach may be that the concepts newly discovered by the agents and merged into the common ontology may not be "real" concepts for the human observer. Basically a concept newly discovered by the agents is "something which is viewed in a way by one agent and viewed in another way by the other agent". Another disadvantage of the proposed approach may be that if we apply this conceptualization based on joint perception in a multi-agent environment, then we may get confusingly many new concepts in every possible pairs of agents. However, the agents may not be able to discover the same "real" concepts as the human observer, because the perceptions of the agents are limited and context based, and the agents are not able to perceive the real world in its reality. Further research will have to focus on the analysis of the proposed protocols in real settings and how to apply the ontology negotiation protocol of the conceptualization based on joint perception among three or more agents in order to support the communication of the agents. In this paper we assumed that the agents benevolently participate in the joint perception, however it would be interesting to consider the cases when the agents report false perceptions either intentionally or by mistake.

References

[1] Amgoud, Leila and Kaci, Souhila. An argumentation framework for merging conflicting knowledge bases. *Int. J. Approx. Reasoning*, 45:321–340, July 2007.

[2] Bailin, Sidney C. and Truszkowski, Walt. Ontology negotiation between intelligent information agents. *Knowl. Eng. Rev.*, 17:7–19, March 2002.

[3] Cholvy, L. A general framework for reasoning about contradictory information and some of its applications. In *Proceedings of the ECAI Workshop Conflicts Among Agents*, 1998.

[4] Cudré-Mauroux, Philippe. Emergent semantics. In Liu, Ling and Özsu, M. Tamer, editors, *Encyclopedia of Database Systems*, pages 982–985. Springer US, 2009.

[5] Diggelen, Jurriaan Van, Beun, Robbert&#45;Jan, Dignum, Frank, Eijk, Rogier M. Van, and Meyer, John&#45;Jules. Ontology negotiation&#58; goals, requirements and implementation. *Int. J. Agent-Oriented Softw. Eng.*, 1:63–90, April 2007.

[6] Genesereth, M. R. and Nilsson, N. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers, San Mateo, CA, 1987.

[7] Grüninger, Michael. The ontological stance for a manufacturing scenario. In Kalfoglou, Yannis, editor, *Cases on Semantic Interoperability for Information Systems Integration: Practices and Applications*, pages 22–42. IGI Global, 2010.

[8] Jennings, Nicholas R. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artif. Intell.*, 75:195–240, June 1995.

[9] Rahm, Erhard and Bernstein, Philip A. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, December 2001.

[10] Santini, Simone, Gupta, Amarnath, and Jain, Ramesh. Emergent semantics through interaction in image databases. *IEEE Trans. on Knowl. and Data Eng.*, 13:337–351, May 2001.

[11] Uschold, M. and Gruninger, M. Creating semantically integrated communities on the world wide web. In *Proceedings of the Semantic Web Workshop Co-located with WWW 2002 Honolulu*, 2002.

[12] Williams, Andrew B. Learning to share meaning in a multi-agent system. *Autonomous Agents and Multi-Agent Systems*, 8:165–193, March 2004.

[13] Wooldridge, Michael J. *An Introduction to Multiagent Systems.* John Wiley & Sons, Inc., Chichester, England, 2009.

Received 5th April 2011