Agent-based Semantic Negotiation Protocol for Semantic Heterogeneity Solving in Multi-agent System

Dhouha Ben Noureddine[1,2], Atef Gharbi[1] and Samir Ben Ahmed[2]
[1]*LISI, National Institute of Applied Sciences and Technology, INSAT, University of Carthage, Tunis, Tunisia*
[2]*FST, University of El Manar, Tunis, Tunisia*

Keywords: Multi-agent System, Ontology, Semantic Heterogeneity, Semantic Negotiation Protocol, Ontology Alignment.

Abstract: In this article, we propose an interactive agent model in an open and heterogeneous multi-agent system (MAS). Our model allows agents to autonomously communicate between each other through semantic heterogeneity. The communication problem can be expressed by the calculation based on the abilities acquired in the receiver agent, compared to the message sent by the sender agent. Hence, the semantic heterogeneity should be resolved in the message processing. The agent can autonomously enrich its own ontology by using semantic negotiation approach in several steps. We develop firstly, a model using an ontology alignment framework. Then, we enhance a similarity measure to select the most similar pairs by combining a psychological knowledge of the relevance, the resemblance, and the non-symmetry of similarity. At the end, we suggest a protocol for supporting semantic negotiation. In order to explain our approach, we implement a simple benchmark production system on JADE.

1 INTRODUCTION

In a MAS, agents are required to communicate to solve tasks and accomplish their goals that are assigned. The features and behavior of agents make the system comply with a set of external constraints on the system. An open MAS means that new agents enter the system and bring with them new features, and other agents take with them when they go out of the system the capabilities required by MAS, making these actions not known as a priori by other agents of the system.
Therefore, in the communication phase, the receiver agent handling different data models should understand formulated demands from another semantically heterogeneous sender agent. The problem of managing heterogeneity among various information resources is increasing in the interactive MAS requiring the adaptation to communication protocols. A standard approach to the resulting problem lies in the use of ontologies for data description. As a consequence, various solutions have been proposed to facilitate dealing with this situation.
That leads us to propose a reflective agent model to solve the semantic heterogeneity problem using two techniques: the calculation of similarity measure and the semantic negotiation protocol. Our agents are communicating with each other, when an agent asks another agent about his capabilities, it will be able to understand the answer from the definitions of the system. In our approach, each agent must have its own ontology in which it's defined.
In this paper we focus on a kind of semantic technologies, named ontology alignment. It is supposed to be accessible by agents and proposed by (Shvaiko and Euzenat, 2013). As the alignment between ontologies is incomplete, agents must then treat queries including non-defined terms in their respective ontologies, and the semantic heterogeneity should be resolved in the message processing. So, solving this problem are no longer in alignment level, it's necessary to define a higher level of messages interpretation and appropriate communication protocols mechanism. Once the translation is done, the receiver agent evaluate the understanding degree of the translated query using a thresholds system such as those defined by (Maes, 1994). This assessment focuses on the reflection thinking capabilities of agents to be able to analyse their own code and to be conscious of the capabilities they have at a given time.
Not to forget that our agent model is based on semantic negotiation technique of (Morge and Routier, 2007), our protocol can be seen as an extension of Foundation for Intelligent Physical Agents *FIPA*-

Noureddine, D., Gharbi, A. and Ahmed, S.
Agent-based Semantic Negotiation Protocol for Semantic Heterogeneity Solving in Multi-agent System.
DOI: 10.5220/0006344602470254
In *Proceedings of the 12th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2017)*, pages 247-254
ISBN: 978-989-758-250-9

request[1]. In fact, in a semantic negotiation context we have situations similar to those of the human discussions, where human beings try to solve those situations in which the involved terms are not mutually understandable, by negotiating the semantics of these terms (Comi et al., 2015).
This paper introduces a simple benchmark production system that will be used throughout this article to illustrate our contribution which is developed as robot-based application. We implement the benchmark production system in a free platform which is JADE (JavaTM Agent DEvelopment)[2] Framework (Bellifemine et al., 2007). JADE is a platform to develop MAS in compliance with the FIPA specifications (Salvatore and Vincenzo, 2009), (Chuan-Jun, 2011), (Bordini, 2006).
The remainder of this paper is organized as follows: we present in the section 2 our agent model, we describe the ontology model, the alignment service used and our semantic similarity measure. Section 3 outlines the negotiation strategy, the speech acts in FIPA-ACL and the communication protocol. A benchmark is used to explain the message exchange between agents to clarify our contribution in section 4. Section 5 provides the major conclusion.

2 AGENT REPRESENTATION

It is a reflective agent model in an open and heterogeneous MAS allowing dynamic interacting in an environment during run-time. Our model is able to modify messages at the run-time in the communication phase, and the agent is able to produce the list of those capabilities at the current time. In this context, when an agent *A* wants to communicate with an agent *B*, it will use its own ontology to build its messages, *B* will receive a formulated message compatible with the terms of agent's *A* ontology, which does not allow him to interpret this message. In fact, after receiving the message from *A*, *B* compares the request received with its own capacities at the current time. We use as technique: the similarity measure proposed by (Shvaiko and Euzenat, 2013) to calculate if two concepts are semantically similar, i.e., they share common properties and attributes, the interest of this measure is the leveraging of all ontologies aspects and holding the maximum similarity. It therefore offers immediately a secure basis for a distance measure. We improve this measure by optimizing it as an asymmetric similarity in order to enhance the performance of capturing human judgements and produce better nearest neighbors.
To make an agent reflective, we need to represent the agent's state during its own execution and manipulate it. To do so, we adopt the Alignment API [3] of (David et al., 2011) to align ontologies. The API implementation itself carries little overhead: alignments of thousands of terms (from large thesauri) are been able to be handled without a slack manner. Furthermore, the API is used to deal with instances of larger ontologies. The primary purpose of this API is that it may be used as a specific tool based directly on secondary memory storage and indexing for dealing with instances and dropping that support from the API.
We work on an alignment in semantic heterogeneous environment and we completely neglect the problem of syntactical, terminological and lexical heterogeneity. We assume that all agents use the same syntax for messages.
In this section, we will present our ontology model, then we will describe the alignment API used and its role for our agent model.

2.1 Ontology Model

An ontology O is described formally as a 6-tuple: $\{C, P, H_c, H_p, A, I\}$ where C a set of concepts, P a set of properties, H_c a set of hierarchical relationships between concepts and sub-concepts, H_p a set of hierarchical relations between properties and sub-properties, A a set of axioms and I a set of instances of concepts C and of properties P.

2.2 Ontology Alignment API

The ontology alignment requests the semantic similarity measure of the ontologies' concepts and the alignments among them. It aims to identify concepts that can be considered similar, regardless the use of the type of alignment: it can include tasks like queries interpreting, translation of messages or obtaining passage axioms between two ontologies.
The ontology alignment problem can be described in one sentence as defined (David et al., 2011): "Given two ontologies O_A and O_B each describing a set of discrete entities (which can be classes, properties, rules, predicates, etc.), find the relationships (e.g., equivalence or subsumption) holding between these entities." In the API description, (David et al., 2011) defined other parameters such as the alignment level, the arity and the set of correspondences.

[1] http://www.fipa.org/specs/fipa00026/SC00026H.html
[2] http://jade.tilab.com/
[3] is a *Java API* for manipulating alignments in the alignment format and *EDOAL* (Expressive and Declarative Ontology Alignment Language).

We can define the ontology alignment between two concepts of two different ontologies O_A and O_B as a 4-tuple $align = \{e_1, e_2, n, R\}$:

- e_1 an entity (class, relationship or instance) of the ontology O_A that should be aligned ($e_1 \in O_A$);
- e_2 an entity (class, relationship or instance) of the ontology O_B that should be aligned ($e_2 \in O_B$);
- R the correspondence relationship (e.g. equivalence, etc.) between e_1 and e_1;
- $n \in [0, 1]$ is the validity degree of this correspondence.

We integrate the Alignment API (David et al., 2011) in our model to ease our contribution. In the balance of this article, we will note $P(S)$ the set of subsets S. We define $M(O_A, O_B)$ the set of mappings between the ontology O_A and ontology O_B. By extension, if S is a set of entities (class, relationship or instance) of the ontology O_A, then we define $M(S, O_B)$ the set of mappings corresponding to entities S of ontology O_A in ontology O_B.

2.3 Translation Data

In order to use an ontology instead of another, we must find it first. A translation program must allow an agent to locally transform a message expressed as a function of an ontology O_A to a new message expressed according to an ontology O_B. That means, when an agent A sends a request to an agent B, B must first translate the request with its own capacity existing in its ontology O_B. We choose to consider (work (Laera et al., 2007)) that MAS has access to an ontology alignment service (subsection 2.2): It should help to reformulate the propositional content of message, i.e., translate it into terms of another ontology (of the receiver agent). In this section, we explain how agent B uses this ontologies alignment service to translate the content of the request received from agent A in terms of its ontology O_B. We consider S_A the message sent by A to B. By nature, $S_A \subset P(O_A)$ (i.e. S_A is a set of concepts of ontology A). The alignment service builds then a set of mappings $M(S_A, O_B)$. The set S_B which is the translation of S_A in O_B, is defined as the set of concepts of O_B as it exists an alignment $align \in M(S_A, O_B)$ connecting to one of the concepts of S_A.

2.4 Semantic Similarity Measure

We speak about semantic similarity, when the calculated measure between two concepts are semantically similar, i.e. when they share common properties and attributes. For example, ”aircraft” and ”car” are similar because they both have the attributes of ”transportation”. Semantic matching score specifies a similarity function in the form of a semantic relation (hypernym, hyponym, meronym, part-of, etc) between the intention of the sender agent's message and the concepts of the receiver agent. This measure is a real number $\in$ [0,1] where 0 (1) stands for completely different (similar) entities (Shvaiko and Euzenat, 2013). So, we can say that the approach followed here consists of assigning each entity category, (e.g. a class), to a specific measure which is defined as a function of the results computed in the related categories of the entity. We apply this following equation (1) to compute the similarity measure between the receiver agent capabilities and the received query from the sender agent.

$$sim_c(A,B) = \frac{\sum_{(c_1,c_2)\in M_{(A,B)}}(sim_c(c_1,c_2))}{\max(|A|,|B|)} \quad (1)$$

Where $M_{(A,B)}$ is a mapping from elements of A to elements of B which maximises $Sim_c(A,B)$. The similarity between the sets is the average of the values of matched pairs. $M_{(A,B)}$ is a function returning the set of pairs of concepts resulting from the possible permutations between A and B, for instance, $M(\{x,y\},\{1,2,3\})$ returns the set of permutations $\{\{x,1\},\{x,2\},\{x,3\},\{y,1\},\{y,2\},\{y,3\}\}$.
In the alignment API (David et al., 2011), the authors ignore the cognitive sense, for instance if the concepts have more common attributes, they are more similar, and if there are more differences, they are less similar or dissimilar. An important psychological idea is that the similarity is non-symmetric. Nevertheless, (Tversky, 1977) proved that the similarity measure between concepts could not be symmetrical, human judgements have not been too. For example, we say more easily “John looks like his father” than “His father looks like John”, or in the relation is-a: “a borzoi resembles a carnivore”, than “a carnivore resembles to a borzoi”. Building on that, we apply the best average (BA) approach which doesn't face any of the pre-mentioned restraints, and takes into consideration both similar and dissimilar concepts as would be expected.
Because of above reasons, we introduce the psychological theory in our similarity measurement methods by optimizing the approach of (Shvaiko and Euzenat, 2013) through adding the non-symmetry property of similarity. To do so, we slightly shift the formulas of (Euzenat, 2013) to serve our purposes. We combine the average (obtained through the API) with the BA one (our upturn), where each average confidence (similarity measure calculated by (Shvaiko and Euzenat, 2013)) of the first ontology is paired only

with the most similar concept of the second one and vice-versa. We propose a completely new method for the computation of the similarity measurement, which has the ability to not ignore the non-symmetry of similarity, and the skill required to generate a best matching average.

Our approach focuses on the calculation of the average similarity between each term in O_A and its most similar term in O_B, averaged with its reciprocal to obtain a symmetric score:

$$sim_c(S_1,S_2) = \frac{sim_c(A,B) + sim_c(B,A)}{2} \quad (2)$$

We define the $A(M) \in [0,1]$ the values of a set of mappings M as the following formula shows:

$$A(M) = \frac{\sum_{(c_1,c_2,n,R)\in M} n}{|M|} \quad (3)$$

In other words, $A(M)$ represents the average of the alignment scores involved in the mapping M. Then, we consider a measure (*score*) calculating the score between two sets of concept S_1 and S_2 as follows:

$$score(S_1,S_2) = \cdot A(M) sim_c(S_1,S_2) \quad (4)$$

3 SEMANTIC NEGOTIATION APPROACH

This section shows a communication between two agents based on the semantic negotiation. Once the translation is done, the receiver agent assesses the understanding degree of the translated query using a system of thresholds. This assessment is based on the reflective capabilities of agents to be able to analyze their own codes, to be aware of the capabilities they have at a given time and to modify their own execution state or alter their own interpretation or meaning. Using this lightly understood query and the capabilities list of the agent at a given time, the receiver agent choose among our five proposed performatives how to describe its understanding of the order placed in the receiver agent so that it can, if necessary, reconsider its request.

3.1 Speech Acts in FIPA-ACL

In the literature, the majority of researches on semantic heterogeneity performs the calculation of the semantic measure without using special modeling for the content of the queries. Some authors measure the similarity between two concepts of the same ontology, others compute the similarity between two concepts of different ontologies. But, few authors calculate the similarity between two sets of concepts. The originality of our approach departed from this idea to compute the similarity between sets of concepts (request, capacity) from a concept-to-concept, especially to calculate the distance between two ontologies to optimize future alignment.

We consider that the message exchange described in the subsection 3.2 uses and respects the identified message information specified by the control FIPA-ACL performatives. We can put forward a few hypothesis to specify the response ID corresponding to the initial message to avoid any problems linked to messages intersection. Our protocol can be seen as a *FIPA-request* extension that would focus more on not-understood messages. The content of the performative will correspond to the classical performatives *request*, *agree*, etc. We will define in detail in the next subsection our new proposed performatives for the application of our communication protocol.

3.2 Communication Model Proposed

The selection of the dynamic protocol in open and heterogeneous MAS for the collaborative tasks execution during the agents' communication proves to be an important step to structure message exchange and ensure consistency of agents' behavior in the system. In order to solve possible understanding problems, two communicating agents need the contribution of other agents in the system, this is the idea of some work addressing the semantic negotiation in the literature (De Meo et al., 2012) (Garruzzo et al., 2011) (Garruzzo and Rosaci, 2008) (Messina et al., 2014). We try to use the negotiation strategy to resolve conflicts between agents.

In this section, we define the role of calculation of the *score* for the selection of candidates' capabilities, and for the communication between agents through determining the speech acts used in the response strategy in the work of (Morge and Routier, 2007). We adopt this approach because, authors assume that it's inconceivable to consider as systematically as possible the ontology alignment. The main problem they see that the alignment is unable to guarantee if it will be correct or complete. Or, if the alignment is imperfect the communication is generally impossible. So, they think that they should deal with the semantic heterogeneity problem directly during the communication, using a protocol that treats semantic negotiation. The sender agent (customer) sends requests to the receiver agent (provider). Each agent can use a number of performatives (question, request, assert, propose, refuse, reject, unknown, concede, challenge and withdraw) in a certain order (Table 1) to argue his perception of the world and his personal beliefs. We adopt this negotiation strategy because it takes into consideration the dynamicity of interactions and cognition of agents.

We consider C_p the set of capacities of an agent at a given time and S a set of concepts representing a content of the message after translation. It's possible to build from C_p, the subset $C_p(S)$ where $C_p(S)$ contains the capacities maximizing the $score(S,c)$, and $c \in C_p$. To sum up, if a subset of the maximum capabilities exists give a *score* result is close to 1, that means the capabilities of this subset are similar to the query, if the result is close to 0, that means the request and the current capabilities are different.

We define $c_p = c_p(S,c)$ the maximum value of *score* result of the subset $C_p(S)$. $C_p(S)$ is defined as follows:

$$c_p(S,c) = \max_{e \in C_p(S)} score(S,e) | c \in C_p(S) \quad (5)$$

Reciprocally, we note C_i the subset of impossible capabilities where c_i is the result of the *score*.

Our protocol is an extension of *FIPA-request*, we consider here the requesting multi-response persuasion protocol (defined *ReqMultiResPersProto*) using the following rules: $sr_{R/P}$, $sr_{A/W}$ and $sr_{A/R}$ (Morge and Routier, 2007).

This protocol is determined by a set of sequence rules (see table 1). "Each rule specifies authorized replying moves. According to the "Request/Propose" rule ($sr_{R/P}$) is quite similar. The hearer of a request (request($\varphi(x)$)) is allowed to respond either by asserting an instantiation of this assumption (assert($\varphi(a)$)), or with a plea of ignorance (unknown($\varphi(x)$)). The respond can resist or surrender to the previous speech act." For example, the "Assert/Welcome" rule (written $sr_{A/W}$), indicates that when it sends an (assert(Φ)), surrendering acts are close to the dialogue line. A concession (concede(Φ)) surrenders to the previous proposition. Resisting acts allow the discussion; a challenge (challenge(φ)) and refuse (refuse(φ)), resist to the previous proposition. In the "Assert/Reject" (written $sr_{A/R}$) rules, the rejection of one of the assumptions previously asserted (reject(φ)) closes the dialogue line. As mentioned in his article in section 5, the same argumentative/public semantics are shared between an assertion and a proposition. Furthermore, assert($\neg\varphi$), refuse(φ) and reject(φ) are identical. But, the place of speech acts are different in the sequence of moves.

A strategy is applied to choose which communicative act to use according to a threshold system $\in$ [0,1] (Maes, 1994). The answer given by our system depends on the results of c_p and c_i using c_{min} and c_{max} [4]. We differentiate 5 different response strategies for the message of the receiver agent according to $sr_{R/P}$, $sr_{A/W}$ and $sr_{A/R}$ rules of (Morge and Routier, 2007).

Table 1: Set of speech acts and their potential answers. (Morge and Routier, 2007).

sequence rules	Speech acts	Resisting replies	Surrendering replies
$sr_{R/P}$	request($\varphi(x)$)	propose($\varphi(a)$)	unknown(φ)
$sr_{A/W}$	assert(Φ)	challenge(φ), refuse(φ), $\varphi \in \Phi$	concede(Φ)
$sr_{A/R}$	assert(Φ)	challenge(φ), $\varphi \in \Phi$	concede(Φ) reject(φ), $\varphi \in \Phi$

1. If $c_p \geq c_{max}$ and $c_i \leq c_p$ and $|C_p(S)| = 1$, the request is considered properly understood by the agent. We respond by asserting an instantiation of this assumption (assert($\varphi(a)$)).
2. If $c_{min} < c_p < c_{max}$ and $c_p < c_i$, the receiver agent believes that the received query is not possible (i.e. $c_p < c_i s$). So, it sends to the initial agent a list of closest events possible to the received command. For this, we introduce the performative propose($\varphi(a)$) indicating: 1) the initial message is not executable and 2) that the content of the message is a set of commands (request) that are acceptable and judged to be close to the original message.
3. If $c_p \leq c_i$, and ($c_{max} \leq c_p$, but $|C_p(S)| > 1$), and ($c_{min} < c_p < c_{max}$); then, impossible capabilities can be ignored, but the agent is not sure if the request is understood ($c_p < c_{max}$) or that there are too many candidate queries ($|C_p(S)| > 1$). In other words, the receiver agent has a list of possible candidate capabilities, but can not proceed with executions. Hence, the B agent makes a clarification request to the agent A by noting the set of possible capabilities most corresponding to the received query (i.e. the receiver agent sends the set $C_p(S)$ to the sender agent). That is why, we introduce the act speech (challenge(φ)).
4. If $c_p \leq c_{min}$ and $c_{min} \leq c_i$, the receiver agent thinks that the order received is understood, but it is impossible. So, the receiver agent must tell the sender agent that his command is understood, but is not currently applicable. We introduce to notify this situation, the performative ((concede (φ)).

[4]Maes proposed empirically use the values c_{min}=0.3 and c_{max}=0.8.

5. If $c_p \leq c_{min}$ and $c_i \leq c_{min}$, the receiver agent is not able to correctly interpret the request of the sender agent. We then introduce the performative refuse.

4 CASE STUDY

4.1 Benchmark Production System

As briefly mentioned before in (Ben Noureddine et al., 2016), we illustrate our contribution with a simple current example called *RARM* (Hruz and Zhou, 2007) (represented in Figure 1). It is composed of two inputs and one output conveyors, a servicing robot and a processing-assembling center. Workpieces to be treated come irregularly one by one. The workpieces of type A are delivered via conveyor $C1$ and workpieces of the type B via the conveyor $C2$. Only one workpiece can be on the input conveyor. A robot R transfers workpieces one after the other to the processing center. The next workpiece can be put on the input conveyor when it has been cleared by the robot. The technology of production requires that firstly an A-workpiece is inserted into the center M and treated, then a B-workpiece is added to the center, and finally the two workpieces are assembled. Afterwards, the assembled product is taken by the robot and put above the $C3$ conveyor of output. The assembled product can be transferred to $C3$ only when the output conveyor is empty and ready to receive the next produced one. We model individual robot systems as distributed agents that deal autonomously with both local task planning and conflicts that occur due to the presence of other robotic agents. The overall behavior of the *RARM* as a whole is then an emerging functionality of the individual skills and the interaction among the forklifts.
The robot-like agent connects directly to the environment via through sensors.

4.1.1 Sensing Input

The robot-like agent receives the information of the probes as follows:

1. is there an object of the type A at the extreme end of the position $p1$? (sens1)
2. is the conveyor $C1$ in its extreme left position? (sens2)
3. is the conveyor $C1$ in its extreme right position? (sens3)
4. is there an object of the type A at the treatment unit M? (sens4)

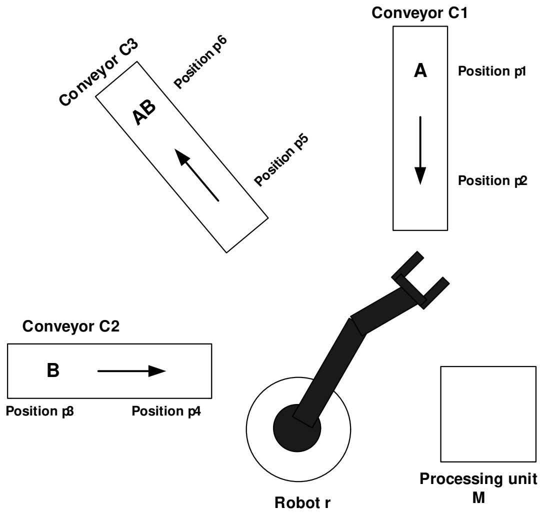

Figure 1: The benchmark production system RARM.

5. is the conveyor $C2$ in its extreme left position? (sens5)
6. is the conveyor $C2$ in its extreme right position? (sens6)
7. is there an object of the type B at the extreme end of the position $p3$? (sens7)
8. is there an object of the type B at the treatment unit M? (sens8)
9. is the conveyor $C3$ in its extreme left position? (sens9)
10. is the conveyor $C3$ in its extreme right position? (sens10)
11. is there an object of the type AB at the treatment unit M? (sens11)
12. is the agent's robot-like arm in its lower position? (sens12)
13. is the agent's robot-like arm in its highest position? (sens13)

4.1.2 Action Output

Once an adapted order, called the plan, is found; the order with elevated level must be converted to orders of low level to be sent to the releases so that the robot-like agent can really carry out the plan.
Running example
The system can be ordered using the following releases:

1. move the conveyor $C1$ (act1);
2. move the conveyor $C2$ (act2);
3. move the conveyor $C3$ (act3);
4. rotate robotic agent (act4);

5. move elevating the robotic agent arm vertically (act5);
6. pick up and drop a piece with the robotic agent arm (act6);
7. treat the workpiece (act7);
8. assembly two pieces (act8).

4.2 Preliminary Results

We prototype these ideas using the JADE agent platform. We consider two *RARM*, each one has its own ontology O_A (resp. O_B) to describe autonomous robots. We define ontology for sub-domains, sensors, perceptions, planning, actuators, decision making, etc. We assume that the agent descriptions of the world is incomplete:

- O_A has a complete description of sensing input (e.g. sens1, sens2, sens3, sens4, etc.), an action output representation (e.g. act1, act2, act3, act4, etc.), but an incomplete representation of the policy (a whole state-action installs with at most an action for each state).
- O_B has a complete description for sensing input, an incomplete representation of action output, and a complete representation of plan (policy).

We consider 5 plans $\{\pi_i|$i=0..4 $\}$ in disorder of the actions used in our example:

- π_0 : ($C2\_left$, $take_2$, $load_2$, $process_2$)
- π_1 : ($load_1$, put_1, $process_1$, $C1\_right$)
- π_2 : ($C1\_left$, $take_1$, $load_1$, put_1, $process_1$, $C1\_right$)
- π_3 : ($C2\_left$, $take_2$, $process_1$, $C2\_right$)
- π_4 : ($take_1$, $load_1$, put_2, $C2\_right$)
- π_5 : ($C1\_left$, $take_1$, put_2, $process_1$)

According to this kind of modeling, some action outputs explicitly designated by the robot $RARM_a$ become ambiguous to $RARM_b$. The alignment between the ontologies does not solve the lack of action outputs' representation in the robot $RARM_b$, similarly the classical request protocol does not resolve the ambiguity. We assume in this example that for every M mapping, then $A(M) = 1$. We develop a scenario between $RARM_a$ and $RARM_b$. The goal is to exploit the use of similarity measure in order to simplify the interactions among heterogeneous agents, with different sensors and different capabilities. In this scenario, the $RARM_a$ requests moving the conveyor $C3$ (i.e., act3) to the $RARM_b$ *request(1, 1,* $\{$*do, sens9, act3*$\}$*)* corresponding to the request 1 of conversation 1. The system checks the alignment service to calculate the alignment for this query, transforms the concepts in the set of concepts corresponding to the capabilities in the ontology O_B and sends a message to the $RARM_b$. Then the robot-like agent $RARM_b$ calculates its capacities.

Figure 2 shows an interaction example based on the context of $RARM_b$ executing capabilities. In our example, $RARM_b$ has 6 plans. So, $|C_p|$ = 6. Considering that Sim_c (sens9, act3) = 0.6, with c_p = (1+0.6)/2 = 0.8. $RARM_b$ applies the third strategy, it uses the performative *challenge* to inform $RARM_a$ by the ambiguity. So. it sends the answer *challenge(1, 1,* $\{\{$*do,* π_0, *act2*$\}$, $\{$*do,* π_1, *act3*$\}$, $\{$*do,* π_2, *act1*$\}$, $\{$*do,* π_3, *act5*$\}$, $\{$*do,* π_4, *act3*$\}$, $\{$*do,* π_5, *act4*$\}\}$.

The ontology of $RARM_a$ doesn't exactly allow modeling outputs, the answer of $RARM_b$ is translated into *challenge (1, 1,* $\{\{$*do,* π_1, *act3*$\}$, $\{$*do,* π_4, *act3*$\}\}$. Since the initial context is $RARM_a$, the next request to $RARM_a$ clarifies the plans: *request(1, 2,* $\{$*do,* π_4, *act3*$\}$. Now, the application finds no more difficulty and eventually is accepted by $RARM_b$ (i.e. $RARM_b$ sends a performative of confirmation *assert(1, 2))*.

5 CONCLUSION

In this paper, we propose a reflective agent model to make a negotiation in an open and heterogeneous MAS. We present a set of communicative acts allowing queries disambiguation of heterogeneous agents in incomplete alignment ontology. To do this, we use a measure similarity to compare each entity of the ontology with the other and select the most similar pairs. In fact, when an agent A sends a request to an agent B, B compares the information from the sent message with its abilities; it calculates the correspondence degree and according to this degree it chooses the corresponding performative. This model introduces a kind of process which overcomes some common problems that are encountered during the MAS development. Currently, we have been developing a benchmark production system as a case study on JADE to improve the quality of the outcome; we shift from a non-understood respond FIPA-request protocol to a multi-response to clarify the request. Finally, we provide an agent interaction model to reach an agreement over heterogeneous representations. The future planned works will deal with the implementation of the proposed model on a real multi-robot system with larger sets of data in heterogeneous ontologies.

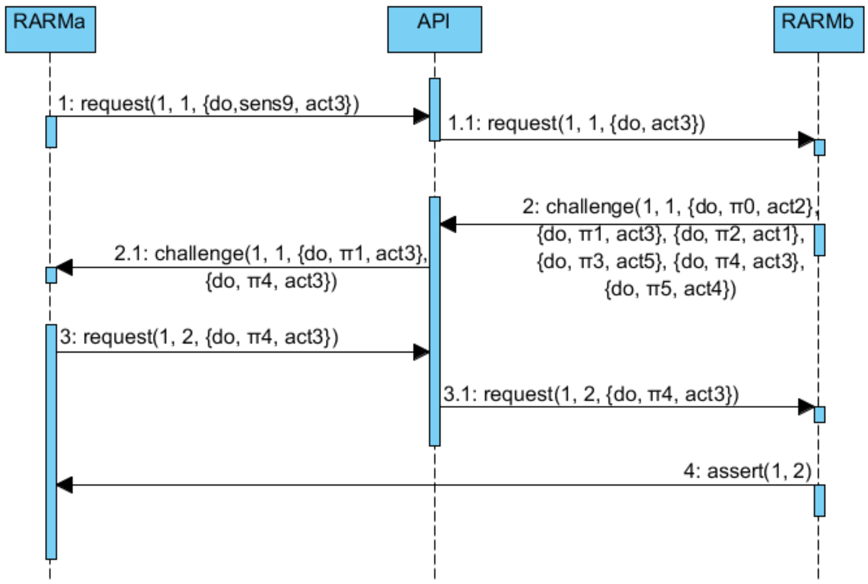

Figure 2: Interaction example between two multi-robot system. The center column doesn't refer to an agent but represents an ontology alignment service used for proof.

REFERENCES

Bellifemine, F., Caire, G., and Greenwood, D. (2007). Developing multi-agent systems with jade.

Ben Noureddine, D., Gharbi, A., and Ben Ahmed, S. (2016). An approach for multi-robot system based on agent layered architecture. In *Proc. of the 87th International Conference on Artificial Intelligence and Soft Computing (ICAISC'16), Lasbone, Portugal*, pages 1–9.

Bordini, R. e. a. (2006). A survey of programming languages and platforms for multi-agent systems. *Informatica*, 30(1):33–44.

Chuan-Jun, S. (2011). Jade implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring. *Applied Soft Computing*, 11(1):315–325.

Comi, A., Fotia, L., Messina, F., Pappalardo, G., Rosaci, D., and Sarne, G. M. (2015). Using semantic negotiation for ontology enrichment in e-learning multi agent systems. In *Ninth International Conference on Complex, Intelligent, and Software Intensive Systems*, pages 474–479.

David, J., Euzenat, J., Scharffe, F., and Trojahn dos Santos, C. (2011). The alignment api 4.0. *Semantic web journal*, 2(1):3–10.

De Meo, P., Quattrone, G., Rosaci, D., and Ursino, D. (2012). Bilateral semantic negotiation: a decentralised approach to ontology enrichment in open multi-agent systems. *International Journal of Data Mining, Modelling and Management*, 4(1):1–38.

Euzenat, J. (2013). An api for ontology alignment. *Springer-Verlag*, pages 698–712.

Garruzzo, S., Quattrone, G., Rosaci, D., and Ursino, D. (2011). Improving agent interoperability via the automatic enrichment of multi-category ontologies. *Web Intelligence and Agent Systems*, 9(4):291–318.

Garruzzo, S. and Rosaci, D. (2008). Agent clustering based on semantic negotiation. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 3(2):7.

Hruz, B. and Zhou, M. (2007). Modeling and control of discrete-event dynamic systems with petri nets and other tools. *Springer*, page 67.

Laera, L., Blacoe, I., Tamma, V., Payne, T.and Euzenat, J., and Bench-Capon, T. (2007). Argumentationove ontology correspondances in mas. In *6th international joint conference on Autonomous Agents and MultiAgent Systems (AAMAS'07)*, pages 1285–1292.

Maes, P. (1994). Agents that reduce workload and information overload. *Communications of the ACM*, 37(7):30–40.

Messina, F., Pappalardo, G., Pappalardo, C., Santoro, D., Rosaci, G., and L., S. G. M. (2014). An agent based negotiation protocol for cloud service level agreements. In *2014 IEEE 23rd International In WETICE Conference (WETICE)*, pages 161–166.

Morge, M. and Routier, J. (2007). Debating over heterogeneous descriptions. *Applied Ontology*, 2:333–349.

Salvatore, V. and Vincenzo, C. (2009). An extended jade-s based framework for developing secure multi-agent systems. *Computer Standards & Interfaces*, 31(5):913–930.

Shvaiko, P. and Euzenat, J. (2013). Ontology matching: State of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25:158–176.

Tversky, A. (1977). Features of similarity. *Psychological Review*, pages 327–352.