Novel Ontology Model for Communicating Heterogeneous Negotiation Mobile-Agent in a Transport Environment

Sawsan Saad[1], **Hayfa Zgaya**[1], **Slim Hammadi**[2]

[1]LAGIS UMR 8146

Ecole Central de Lille, France

sawsan.saad@ec-lille.fr, hayfa.zgaya@ec-lille.fr

[2]LAGIS UMR 9146

Ecole Central de Lille, France

slim.hammadi@ec-lille.fr

Abstract: In this paper, we address the problem of negotiation process in a multi-agents system by using ontologies. Therefore, we present an ontology solution based on the knowledge management system for semantic heterogeneity. The proposed solution prevents the misunderstanding during the negotiation process through the agents' communications. Our approach aims to enable agents able to understand each other when using these ontologies. Thus, we propose a general architecture for Negotiation process which uses Ontology-based Knowledge Management System (NOKMS). This architecture consists of three layers: the Negotiation Layer (NL) that describes the negotiation process between the Initiator Static Agents (ISAs) and the Participant Mobile Agents (PMAs) by using suitable ontologies, the Semantic Layer (SEL) contains the semantic translator which uses in the case of misunderstanding of the sent messages between the agents, and the last one is the Knowledge Management Systems Layer (KMSL) which bases on the Intelligent Knowledge Base (IKB) to give the flexibility to our negotiation ontology. In addition, we will illustrate an agent architecture which helps our architecture on applying the different operations in the different layers. Finally, we present a case study which applies our architecture on the *Multimodal Transport Information System (MTIS)* project where we will show two scenarios applicable: the first uses our negotiation ontology architecture in one transport system, and the second applies this architecture on the multi-transport systems. These case studies show that the proposed NOKMS improves the execution of negotiation process in multi-agents systems in order to satisfy the transport customers.

Keywords: Multi-Agents Systems, Negotiation, Ontology, Knowledge Management System, Transport Information System.

Sawsan Saad is currently a doctorate in LAGIS laboratory at the high France School, Ecole Centrale de Lille (EC-Lille) in the Optimization, the Artificial Intelligence and the Logistic field. Born in *Salamieh* (Syria) in 1979. She received the BSc degree in Computer Engineering from Al-Baath University (Syria) in 2003. She obtained her MSc degree in Information Systems in 2007 from INSA de Lyon (France). She was member of the Organizing Committees of international congress on "Logistic and Transport" LT'2007.

Dr. Hayfa Zgaya is teacher-researcher in Computing Science in LAGIS laboratory within the high France School Ecole Centrale de Lille (EC-Lille) where she obtained her PhD degree in July 2007. Her main research areas are the Optimization, the Artificial Intelligence and the Logistic field. Born in Tunis in 1978, she obtained her master's degree in computer science in November 2001 from the France polytechnic high school of Nantes University. She takes an active part in the national working group ORT of GDR MACS and she was responsible of the Organizing Committees of international workshops MHOSI' 2005, LT'2006 and LT'2007.

Slim Hammadi is a full Professor of production planning and control at the Ecole Centrale de Lille (French «Grande Ecole»). Born in Gafsa (Tunisia) in 1962, he has obtained by 1988 the Master degree in Computer science from the University of Lille (France). Pr Hammadi obtained a P.h.D degree in job-shop scheduling and control in 1991 at Ecole Centrale de Lille. He is a senior member of IEEE/SMC and has served as a referee for numerous journals including the IEEE Transactions on SMC. Pr. S. Hammadi was Co-Organizer of a Symposium (IMS) of the IMACS/IEEE SMC Multi conference CESA'98 held in Hammamet (Tunisia) in April 1998. He has organized several invited sessions in different SMC conferences where he was session chairman. He was chairman of the International congress on "Logistic and Transport" LT'2004, MHOSI'2005, LT'2006 and LT'2007. His teaching and research interests focus on the areas of production control, production planning, computer science, discrete and dynamic programming and computer integrated manufacturing.

1. Introduction

This work belongs to the French national project VIATIC.MOBILITE from the industrial cluster I-TRANS «France highlights leading-edge technology in rail systems and innovative transport»[*]. In fact; it has become much more important for public transportation companies to provide an adequate Level Of Service to their customers. This is due to the growing competition in the public transport market and also to the privatization of the transport companies. The great difficulty related to the traffic management in such systems is allied to the respect of the planned departure and arrival times of the vehicles at the different stops of the network. In fact, many incidents can occur and force a customer to wait longer, which decreases the level of service. Hence, operational decisions have to be taken in real-time by human decision makers, called regulators. However, these operators are overloaded with information that they have to treat immediately in order to find the relevant decisions that result in new vehicle schedules. Decision Support Systems (DSS) are computer technology solutions that can be used to support complex decision making and problem solving. Decision making is the study of how decisions are actually taken and how they can be better or more successfully taken. In order to realize the necessity of a DSS for a Transportation System, it is important to grasp the problems related to a real-time management of the traffic [20].

In our *Multimodal Transport Information Syste (MTIS)* [1] project, we presented a Multi-Agent Decision Support Systems that provides the regulators with the relevant decisions to undertake in case of disturbances. In fact, scheduling can be defined as a problem of finding the optimal sequence for executing a finite set of operations under a certain set of constraints which must be satisfied. A scheduler usually attempts to maximize the utilization of individuals and / or machinery and minimize the time required to complete the entire process being scheduled. Therefore the scheduling problem is very hard to solve [27]. A Genetic Algorithms (GAs) have been used to solve this problems in our *MTIS* . The proposed multi-agent system is based on metaheuristics for the research and the composition of the services; services research is based on the Mobile Agent paradigm (MA) using this dynamic optimization algorithm for the MA Workplans design. The first step of optimization prepares the MA routes, taking into account the network state. The services composition uses evolutionary algorithms to optimize the responses in terms of costs and delays, knowing that a response to a user request must respect a fixed due date with a reasonable cost. We also designed and optimized the management of the data flow of the users' requests, which can be simultaneous and numerous. We developed also a negotiation protocol intended for the transport area which permits the agents to negotiate when perturbations may exist and as a result the system needs to reassign news nodes. The negotiation protocol uses messages to exchange the information. Those messages are exchanged between the *Scheduler Agents (SAs)*, representing the initiators of the negotiation, and the *Intelligen*t *Collector Agents (ICAs),* representing the participants of the negotiation*.* This protocol has studied before only the cases of the simple messages and it proposed ontology without illustrating it, and this later didn't include the solutions when the agents ICAs did not understand the messages sent from the SA agent In this paper, we propose an approach that aims to improve the protocol of the negotiation of the multi-agents systems which has been proposed in the previous work [2]; we present an ontology solution based on the knowledge management system for semantic heterogeneity. The proposed solution prevents the misunderstanding during the negotiation through the agents' communications. Our approach aims to make the agents able to understand each other when using these ontologies.

The rest of this paper is organized as follows: firstly, we discuss some related work (Section 2). Then, we discuss the role of the ontology in multi-agent systems (Section 3), ontologies and their combination problems will present in (Section 4). And we will present a negotiation ontology based on knowledge management system proposal (Section 5). An agent architecture based knowledge model (Section 6), and a real transport case study is illustrated in (Section 7); finally conclusion and future work are presented (Section 8).

---

[*] http://www.i-trans.org

2. Related Work

Negotiation has been done by different research works; Bravo et al. [10] has presented a semantic proposition for manipulating the lack of understanding massages between the seller and buyer agents during the exchange of messages in a negotiation process. Otherwise, Zgaya et al. [2] have provided a negotiation protocol for the transport area where they have proposed a flexible ontology to facilitate the communications between the agents. A generic negotiation model for multi-agent systems has been proposed by [18], built on three levels: a communication level, a negotiation level and a strategic level and the later is the only level reserved for the application. In addition, they have illustrated their negotiation protocol which based on a contract which in turn based on negotiation too. This protocol enables many- to-many negotiations between the agents.

The idea of negotiation ontology was proposed by [5] based on the ontology framework which provides the terminology to reason in terms of negotiation protocol.

In another work on ontology researches, Razmerita et al. [17] have presented generic ontology-based user modeling architecture where they illustrated the usage of knowledge management systems in the generic framework based on ontology. Sridharan et al. [13] have developed a framework for an ontology based knowledge management system to facilitate the use of the knowledge in a web based learning environment. Recently, Abou Assali et al. [3] presented an ontology-based knowledge management system for indexing and retrieving the internal resources of an enterprise, in their work they have used tow ontologies (domain ontology and application ontology).

We have found that the different previous approaches treat the negotiation, negotiation ontology, the ontology-based management system. Our work is based on the treatment of the Negotiation Ontology-based Knowledge Management System (NOKMS).Through this work; we aim to give the flexibility of the multi agents systems and to improve at same time the negotiation process between the agents which use different ontologies.

3. Ontology in Multi-Agent Systems

In this section we will firstly introduce the different definitions of the ontology in different domains.

Ontology is the branch of philosophy which considers the nature and essence of things. From the point of view of Artificial intelligence, it deals with reasoning about models of the world. A commonly agreed definition of ontology is: 'ontology is an explicit and formal specification of a conceptualisation of a domain of interest' According to [30]. In this definition, a *conceptualisation* refers to an abstract model of some phenomenon in the world which identifies the concepts that are relevant to the phenomenon; *explicit* means that the type of concepts used, and that the constraints on their use are explicitly defined; *formal* refers to the fact that an ontology should be machine-readable and finally *shared* reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but not accepted by a group[26].Finally, the ontology is a set of definitions of the content-specific knowledge representation primitives (concepts (also known as classes), relations(properties), functions, instances, and axioms) [15].This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. The main motivation of the ontologies is that they allow sharing and reuse of the of formally represented knowledge bodies in computational form.

When we talk about agents we mean agents in a Multi-Agent system, where more than one agent is present, and where agents can interact. The term "agent" can be understood differently depending on the focus of research in our work we will use two types of the agent. The first type is FIPA based agents follow FIPA [33] standards. Interaction among agents is done by the FIPA Agent Communication Language (FIPA ACL) [6]. Agents communicate using ontology and a content language. The agent platform needs to support a directory facilitator where all agents can register [8].The second type is Mobile Agents which are software agents which can move from one place to another. There are 2 kinds of mobility: strong and weak. The strong mobility means a migration of an agent with its execution state and its variables values from one computer to another. The weak mobility is when an agent migrates and caries only the code and variables values.

Indeed, within a multi-agent system, agents are characterized by different views of the world that are explicitly defined by ontologies, that is views of what the agent recognises to be the concepts describing the application domain which is associated with the agent together with their relationships and constraints [28]. Interoperability between agents is achieved through the reconciliation of these views of the world by a commitment to common ontologies that permit agents to interoperate and cooperate while maintaining their autonomy. In open systems, agents are associated with knowledge sources which are diverse in nature and have been developed for different purposes. Knowledge sources embedded in a dynamic environment can join and leave the system at any time [22]. Agents sharing the same ontology can exchange their knowledge fluently as their knowledge representations are compatible with respect to the concepts regarded as relevant and with respect to the names given to these concepts [4]. The most straightforward way to realize a common ontology would be to develop a standardized ontology which is used by all agents.

4. Ontologies and Combinations Problems

In previous section we talked about the role of ontology in the muti-agent systems. Where in open multi-agent systems, communication problems that arise from heterogeneous ontologies should be solved, rather than avoided. We will summarize those heterogeneous ontologies problems as follow:

4.1 Ontology Mismatching

What are the types of differences between the ontologies? According to Klein [25], there are different categorizes of mismatches:

- *Language level or meta-model level:* In this level, mismatches occur when ontologies written in different ontology languages are combined. The languages can differ in their: 1) *Syntax*: this mismatch occurs when different ontology languages use different syntax, but, more important , constructs available in one language(e.g: stating that classes are disjoint) are not available in another. 2) *Semantics of primitives*: the more possible different at the meta-model level is the semantics of language constructs. Although sometimes the same name is used for a language construct in two languages. The semantics may differ. 3) *Logical representation*: the mismatches of this level are the difference in representation of logical notions. Also, notice that this mismatch is not about the representation of the concepts, but about the representation of logical notions (e.g.: by giving translations rules from one logical representation to another). 4) *Language expressivity:* the mismatch at the metamodel level is the difference in expressivity between two languages. (e.g. some languages have construct to express negation while anther languages have not that express). Indeed, If the ontologies are not represented in the same languages, a translation between sources ontologies to the same language is required.
- *Ontology level*: Even for ontologies expressed in the same languages, it is possible to appear ontology level mismatch. These mismatches may occur as well as when they use different languages. The two basic types of ontology mismatches are: (1) *Conceptualization mismatches*, which are mismatches of different conceptualizations of the same domain and (2) *Explication mismatches*, which are mismatches in the way a conceptualization is specified [7].

4.2 Ontology Mapping, Merging and Alignment

We distinguish the principled kinds of Ontology Mediation. In the context of semantic knowledge management, Ontology Mediation is especially important to enable sharing of data between heterogeneous knowledge bases and to allow applications to reuse data from different knowledge bases[7]. Where *Ontology Mapping* is mostly concerned with the representation of correspondences between ontologies*; Ontology Alignment* is concerned with the discovery of these correspondences; and *Ontology Merging* is concerned with creating the union of ontologies, based on correspondences between the ontologies.

5. A Negotiation Ontologies based on Knowledge Management Systems (NOKMS)

We propose a general architecture for negotiation process which uses ontology-based knowledge management system (Figure1).

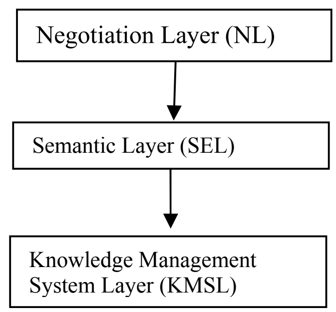

Figure Management System Layers

Figure 1. A Negotiation Ontology - based Knowledge

We organize our architecture as follow: the first layer contains the Negotiation Layer (NL) where the Initiator Static Agents (ISAs) send the first massage to the Participant Mobile Agents (PMAs) to start the negotiation process. The second layer represents the Semantic Layer (SEL), in the case of not understanding the negotiation messages; the SEL uses a translator semantic in order to help it to translate automatically the various types of exchanges between the agents, because the agents don't have suitable ontologies which contain the suitable vocabulary for their communications and their negotiations. The third layer is the Knowledge Management Systems Layer (KMSL) uses ontology in purpose of automatic classifying and using of the news ontologies and meta-ontologies.

5.1 Negotiation Layer (NL)

In this section we illustrate the negotiation layer where the negotiation is defined as a process whose transitions and states are described by the negotiation protocol that agents have to follow for interaction [10].

The agents participate in the negotiation by using their languages for formulating negotiation messages in order to interact and to take the decision. The language used by the agent to interact and execute the exchange of the messages and knowledge is called Agent Communication Language (ACL), as [31], the standard of ACL defined by the Foundation for Intelligent Physical Agents (FIPA).FIPA ACL consists of a set of primitives which allow the communication between the agents and based on the communication acts. In our proposal, (Figure 2), NL contains the initiators (ISAs), agent communication language (ACL) and the participants (PMAs) in the negotiation process.

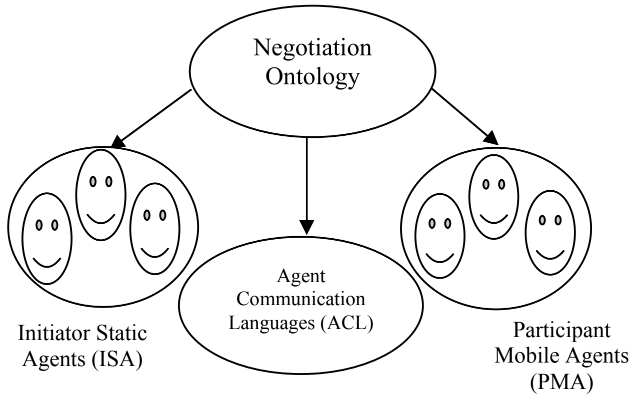

Figure 2. Negotiation Layer (NL)

This layer the initiators start the negotiation process by sending the ACL messages to the participants. The problem will tack place when the participants don't understand the communication messages, or when the new agent wants to participate in a negotiation process and it must understand the protocol and the communication language messages, in this case the agents need an interoperable language between themselves for understanding each other. We find that the best solution is to use ontology. We'll illustrate this problem in the suit of this paper. Indeed, our negotiation layer represent the negotiation process as illustrated in our previous work [2], where we have presented the negotiation protocol (Figure 3) which uses a flexible ontology and this protocol allows a partial agreement from each PMA agent, to be confirmed partially or totally by the initiator of the negotiation (ISA agent), and it allows the renegotiation process, if necessary for the rest of tasks which need to be reassigned.

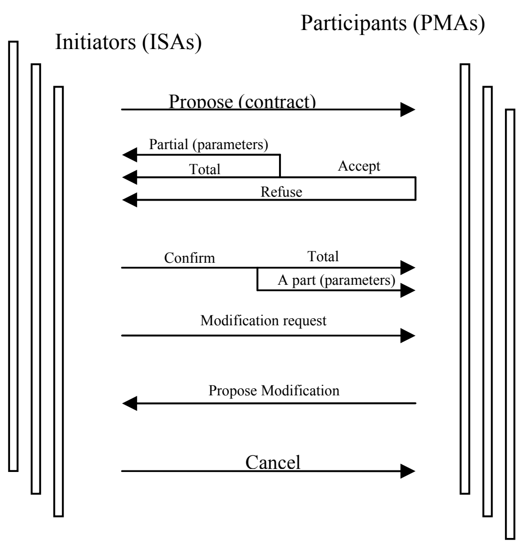

Figure 3. Proposed Protocol

The formula of the ACL messages is as follow [31]:

<Sender, Receiver, Services, Performative, Contents, Language, Ontology>

- Sender: the identity of the sender of the message, that is to say, the name of the agent of the communicative act.
- Receiver: the identity of the intended recipients of the message.
- Performative: the type of the communicative act of the ACL message. The performative

parameter is a required parameter of all ACL messages.

Performative = {Propose, Agree (total, Partial), Confirm, Cancel, Call for Proposal, Not Understood…}

- Services: the "yellow pages" proposed by the recipient of the message
- Content: the content of the message. The meaning of the content of any ACL message is intended to be interpreted by the receiver of the message.
- Language: the language in which the content parameter is expressed.
- Ontology: the ontology(s) is used to give a meaning to the symbols in the content expression (vocabulary, terms, relations…).

The usage of this formula is very easy when the agents interact by exchanging the messages which contain the same ontology. But the semantic interoperability problems take place when the sharing information and knowledge use different ontologies, or when there are multiple ontologies which resemble a universal ontology.

5.2 Semantic Layer (SEL)

As we have presented in the previous paragraph, the negotiation process will be easier handled when we use the ontology. Ontology [10] can be regarded as a vocabulary of terms and relationships between those terms in a given domain. Ontologies have been studied in different researches domains because they facilitate the communication among the negotiation agents where ontology is used as an Interlingua. Our purpose is to find a solution especially in the case of misunderstanding of the negotiation messages among the agents.

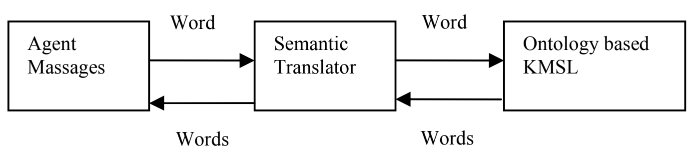

Figure 4. Semantic Layer (SEL)

For this reason, SEL, (figure 4), helps the system in its research to find the best solution, where SEL uses the semantic translator which, in turn, translates the massages sent from the initiators (Propose, Call for Proposal, Confirm…). These messages were not understood by the participants and vice versa for the messages of the participants (Agree (total, Partial), refuse,…). The formula which must be used to solve this problem is:

<Sender, Receiver, Ontology1, Ontology2, P>

P: a predicate which used to determine the relationship among the ontologies and decide the level of transibility between the initiator ontologies (languages) and the participant ontologies (languages).

P= (ontol-relationship ?Ontology1 ?Ontology2 ?Level)

Where Level:{ Weakly-Translatable, Strongly-Translatable, Approx-Translatable}.

In SEL, the translator semantic examines the level of transibility among the ontologies by sending a word to the ontology-based KMSL which resend the set of semantically equivalences words. In fact, the ontology-base KSML connect with KMSL to answer the query of the semantic translator which determines the level of transibility to facilitate the translation process.

5.3 Knowledge Management Systems Layer (KMSL)

Basically; the role of Ontology in the Knowledge Systems is to facilitate the construction of domain model. A meta-ontology and knowledge model, which is necessary for this construction and usage, describe the primitives used by a knowledge representation language, like (concepts, parameters, relations, etc).

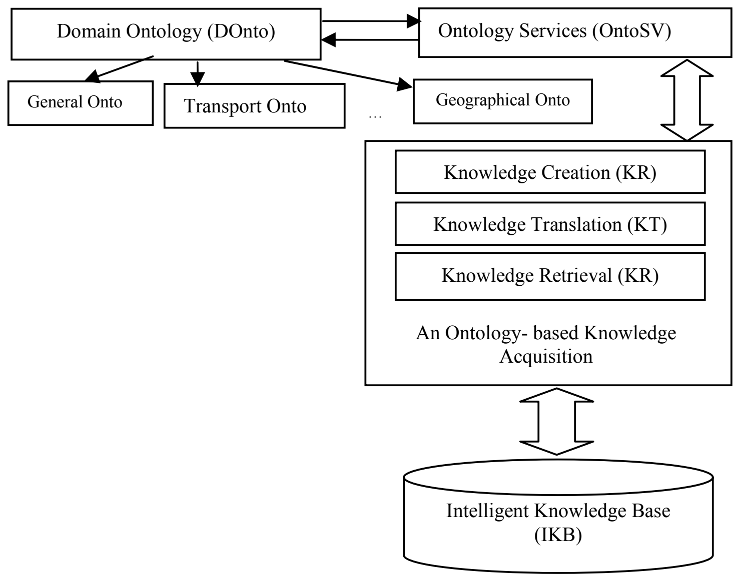

Figure 5. Knowledge Management System Layer (LKMS)

In this section we introduce the Knowledge Management Systems Layer (KMSL).Our architecture in this layer, as it represented in figure 5, consists of:

- Domain Ontology (DOnto): DOnto contains the list of application domains in which the ontology is applicable. By using this domain, the agents communicate with each other through common domain knowledge, in other words as mention in [24]: a common ontology can serve as a knowledge-level specification of the ontological commitments of a set of participating agents. A common ontology defines the vocabulary with which quires and assertions are exchange between agents). DOnto gives the flexibility to the negotiation ontologies which can capture the valid knowledge for different domains (e.g. Transport domain, Geographic domain, etc).
- Ontology Services (OntoSV): The task of OntoSV is to define the semantics of ontologies (actions, predicates used in the content of the conversation with the Ontology Agents (AOs)) which the agents use to interact with each other and support the knowledge acquisition operations (Creation, Translation, Retrieval).OntoSV adopts Open Knowledge Base Connectivity (OKBC) knowledge model as fipa-meta-ontology (an ontology used to access the OAs). The syntax of translation process used in OntoSV is defined as follow:

  <Sender, Receiver, Ontology1, Ontology2, F>

  F: the translation process service. This service is applied to translate expressions (terms, sentences) among translatable ontologies (.i.e. before using this translation action, the SEL must check whether the ontologies are translatable or not by using the predicate P which illustrated in the previous section)

  F= (translate <expression> <translate-description>)
  <translate-description> = (translate <From A> <To B> <Level>)

(A, B) could be an ontology or a language. Level is the level of transibility among the ontologies.

- Knowledge Acquisitions: very important part in the ontology process because they are used to create a new DOnto or languages, to perform the translation among ontologies and to retrieve the knowledge from the Intelligent Knowledge Base
    - ➢ Knowledge Creation (KC): this operation is used to create a new ontology with a new Donto when a new agent wants to participate in the negotiation and when it have not the appropriate ontology ,or when the PMAs don't understand the ontology because they have not this ontology, in these cases the KMSL executes KC process.
    - ➢ Knowledge Translation (KT): translates the terms and sentences among ontologies. But before that, it uses the results of the SEL to verify whether the ontologies are translatable or not.
    - ➢ Knowledge Retrieval (KR): agents can access to the meta-ontology through a query interface, and they use their ontology to view the knowledge items in the intelligent knowledge base.
- Intelligent Knowledge Base (IKB): Ontology is defined as terminological component of a Knowledge Base (KB) [10]. In addition, an agent uses its knowledge base which contains theorems to reason about the application domain. Combining these two approaches, each agent of Multi-Agent System (MAS) holds a KB which based on the domain ontology (application ontology). In our Intelligent KB (IKB); ontology together a set of individual instances of classes constitutes an Intelligent KB. IKB uses the OKBC (Open Knowledge Base Connectivity) [29], which in turn, connects to a wide verity of IKBs servers where these IKBs are applied the Knowledge Acquisitions.

6. Agent Architecture Based Knowledge Model

6.1 Agent Architecture

As we mentioned above in our architecture there are many of the operations in each layer. In the NL, the negotiation process is done by the ISA agents and PMA agents. But for the other layers, we think that best solution is to design un agent architecture based knowledge model look to help our NOKMS architecture in applying the operations on the negotiation ontology in those layers like (Create, Translate, etc…) which applied on the Agent Ontology (AO) in each transport operator in our multi-transport operators. The Agent architecture idea comes from the Pellucid project [21], and we extended and use it in our MTIS project.

There are three types *(Belief Desire Intention (BDI), Reactive, and Behavioural)* to design the agent architecture. Indeed, most of the agent architectures are combinations of basic architecture Types, for that it called hybrid architectures [8]. In this paper we use the behavioural architecture because we interest to agent memory model which use to implement the agent behaviours. In this graph we try to apply the different operations on the heterogeneous ontologies by using the agent behaviour.

Our architecture is modeled as a workflow of basic agent behavior (our model is based on events) (figure 6). The idea is taken partially from the JADE ontology model (predicates and concept), where this graph can be understand as a formal representation of ontology described by graph. Black boxes represent ontology classes, black arrows stand for relations between classes, mostly inheritance relations, expressed by words "is a". Property relations are represented by blue arrows with name of property and cardinality that is mostly multiple "*". Red boxes denote ontology individuals and red arrows relations of an individual to ontology class with associated letters "io". Such graphs can be generated using Ontoviz plug-in [14] for Protégé.

Our model is based on: Actions, Actors, DOnto and Events. In all cases Events are generated based on performed communication, and on received event Actor's model. Figure 2 shows a graph representation this model where using the same terms and it is compatible with OWL-DL [15]. We use OWL-DL to be able to integrate with web browsers, so the application could be translated into a web service in the future. In our proposal, we expect that all agents share same ontology which is General Ontology. The later uses the Communication vocabularies (*Cv*). *Cv* defined as the set of concepts to be used in communication and is specified as an ontology *Ocv* which is shared by agents [9]. General Ontology defines the *Cv* with which queries and assertions are exchange between agents. DOnto gives the flexibility to the negotiation ontologies which can capture the valid knowledge for different domains (e.g. Transport domain, Geographic domain, etc).

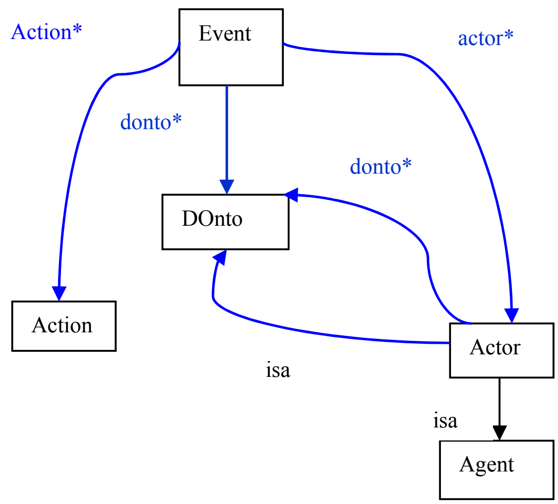

Figure 6. Agent Architecture

6.2 Formal Description of the Agent Architecture and Required Tools

We describe the knowledge model using description logic. [19]. *Event* class represents events in the system. *Event* individual *{event}* is *{action}* taken by *{actor}* on particular *{DOnto}*. Properties of *Event* class are *donto.Event, action.Event, actor.Event.*

$$Event \subseteq$$

$$action.Event(\{action\}) \cap$$

$$DOnto.Event(\{donto\}) \cap$$

$$actor.Event(\{actor\})$$

$$\{event\} \in Event$$

The *DOnto* class stands for all the domains in the agent environment (figure 7). A subclass of *DOnto* is *Actor.*

$$Actor \subseteq DOnto$$

$$\{actor\} \in Actor$$

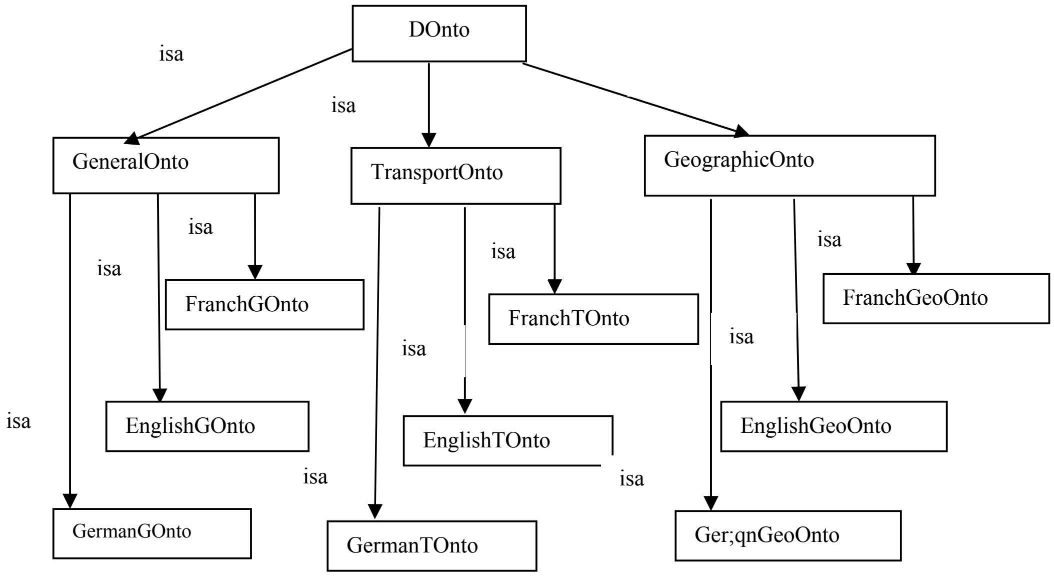

Figure 7. Domain Ontology Model

Actor class denotes actors in the environment. *Actor* individuals can take an actions *{action}* which are individuals of *Action* class.

$$\{action\} \in Action$$

Special type of *Actor* is *Agent.* Agent is used for Software agent representation in the system.

$$Agent \subseteq Actor$$

$$\{agent\} \in Agent$$

Typical actions which can be performed by software agents are defined. They represent types of inter-agent communication such as ACL (AGREE (total, Partial), PROPOS, CONFORM, CANCEL, etc…) message. When communication between agents is performed events of such kind are generated .

{aAgree, aPropos, aConform, aCancel} $\in$ *Action*

When actions (such as: creating, translating, merging or alignment) of *DOnto* are performed in the system (figure 8); Events containing of those kinds of action are stored and evaluated in the system.

{ ACreate, ATranslate, ARetriev, AMerging, AAligneme} $\in$ *Action*

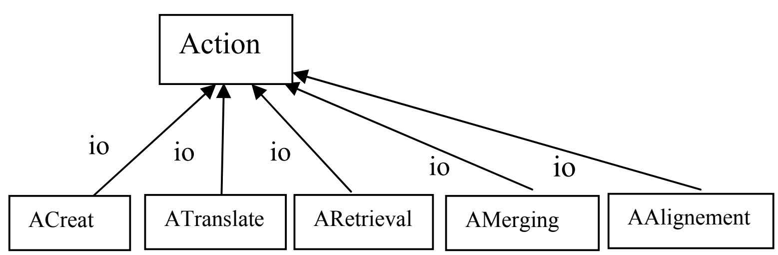

Figure 8. Action Model

For the requiered tooles, Protégé is an open-source development environment for ontologies and knowledge-based systems. We used Protégé to develop the ontology for many of reasons. Firstly, we are not requiring a large amount of time to become familiar with it because the Protégé interface was both intuitive and user-friendly. Secondly it contained a large number of plug-ins that enabled the user to extend the editor's core functionality. Some of the plug-ins that looked especially useful were the OntoViz Tab, OntoViz plg-in[9] was used for visualization of ontology to graphs. We will created all our ontology structures by using this plug-in. Result graphs are similar to UML diagrams. and the XML Tab, which enabled Protégé ontologies to be extracted from XML files and XML files to be translated into Protégé ontologies. This could facilitate the depiction of the ontology in a more presentable manner. The final deciding factor is Bean Generator plug-in[24] which can be used for exporting ontology developed in Protégé to JADE ontology model. This was used to test capabilities of ontology based on Java class representation and FIPA-SL language [34]. As we had decided to use the JADE multi-agent environment [16] for implementation of MTIS project [2].The JADE framework is also able to integrate with web browsers and Java Applets, so the application could be translated into a web service in the future, enabling greater flexibility. Similarly, due to the underlying JADE infrastructure, the prototype may be run on multipl computers with little complication.

7. Transport Case Study

In this section we illustrate a real case study applied on our MTIS project. Transport users require relevant, interactive and instantaneous information during their travels. Hence, MTIS offers a support tool to response to their demands. In the previous work of our research team [1], a detailed example has implemented negotiation process where it has illustrated the usage of negotiation protocol in the case of perturbations.

Some perturbations can occur through the network when ICAs agents follow their correspondent final Workplans, according to the generated optimal solution instance. The proposed negotiation process allowed the reassignment of the cancelled services. But this protocol has studied before only the cases of the simple messages, where it proposed ontology without illustrating it, and this later didn't include the solutions when the agents ICAs did not understand the messages sent from the SAs agent.

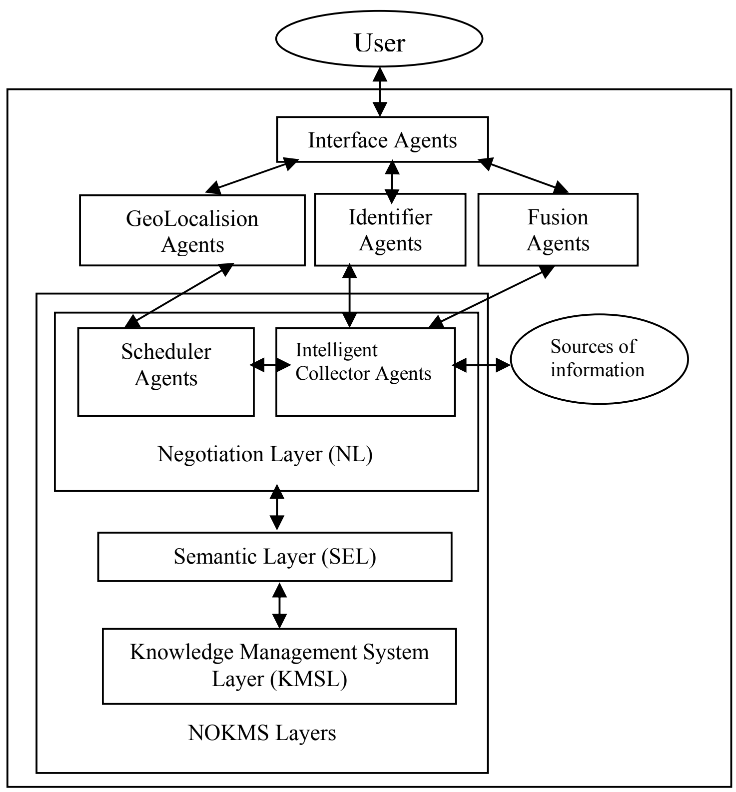

Figure 9. Multi-Agents Structure

In this paper, we propose an approach that aims to improve the protocol of the negotiation of the multi-agents systems which has been proposed in the previous work [2]; we present an ontology solution based on the knowledge management system for semantic heterogeneity. The proposed solution prevents the misunderstanding during the negotiation through the agents' communications. Our approach aims to make the agents able to understand each other when using these ontologies based on these changes, (Figure 9) presents the new system architecture.

Two case studies are applied on our NOKMS architecture; the first one uses our architecture in a transport operator, while the second one applies this architecture on the multi-transport operators. These scenarios exhibit that the proposed NOKMS improves the execution of negotiation process in MAS, in order to satisfy the transport customers. As we noted earlier, there are many problems related to this negotiation process, but in this paper, we illustrate only the case of misunderstanding of the messages sent among the agents. We make use of the proposed example [1] to illustrate the application of these scenarios.

At the instant t=11.00 and during 2 seconds ($\Delta\varepsilon = 2s$), we assume the existence of a number of users connected to our system who formulate a number of queries. Since, this set of queries, where (A, B, C and D) are 4 cities in different countries (for example: A= Lyon, B=Paris, C= London, D= Berlin), is:

- Query1: < travel at the instant t from B to C >
- Query2: < travel in the next weekend from A to B with minimum cost, and ask about weather and cultural events for the next weekend in place B >
- Query3: < travel at the instant t from A to C >
- Query4: < Ask about the perturbations of public transport circulate between B and C >
- Query5:< Look for the best service in correspond to the train X, available in A today at 12.00 to go to C >
- Query6:< travel at the instant t from A to B >
- Query7:< Looking for a hotel of a good (Quality /Cost) in D during the next weekend and make a reservation, look for the best way and time of departure to travel from B to C by the car, According to traffic >
- Etc.

The system starts its work by dividing these queries into I'=64 tasks, and it should remark that there is no direct way between A and C or A and D. these tasks are:

- T1=" Perturbations of traffic between B and C (at the instant t) "
- T2=" Ask about the weather in B (next weekend)"
- T3=" Look for a hotel of a good (Quality /Cost) in D during next weekend ,and make a reservation"
  - T6=" Find the shortest way to go by the car from B to D"
  - T9=" Find the best departure time to travel from B to D by the car according to the traffic during the next weekend"
  - T13=" Ask about cultural events in B (next weekend)"
  - T16=" travel from B to C today(at the instant t=11.00 or starting from 12.00)"
  - T19=" travel from A to B (today at the instant t=11.00 or in the next weekend, With the best price / best service related to the train X at the instant 12.00 Today)"
  - Etc.

After determining the tasks, we have searched for the best optimized solution by using the genetic algorithm. The solution generated by the system is illustrated by the chromosome [11]. This solution meets the total time for execution all queries with a maximum time Dmax=23,46s and with a total cost = 30, 5 units cost. According to this solution the following initial and final Workplans of roads of the ICAs give the following results [12]:

- m= 5, Si: transport information system operator where 1≤ i ≤ n. Then, IWps are:
    - IWp1= (S20, S15, S1, S3) ;
    - IWp2= (S18, S7, S10, S17) ;
    - IWp3= (S2, S13, S19, S6) ;
    - IWp4= (S16, S14, S5, S12, S4) ;
    - IWp5= (S11, S8, S9) ;
- m'=5, the FWps are:
    - FWp1={S20{T9,T37,T39},S15{T28,T58},S1{T19,T29,T66,T88}, S3{T3,T26,T32,T33, T38,T42,T61,T85}};
    - FWp2={S18{T1,T13,T30,T36,T41,T65,T76,T77}, S7{T34},S17{T25,T44,T60,T80}};
    - FWp3= {S2{T59,T78,T79,T84},S13{T53}, S19{T6,T16,T22,T52,T57,T67,T96}, S6{T68}};
    - FWp4= {S16{T63,T73,T74,T90},S14{T71}, S5{T20,T86,T95,T99},S12{T75,T83}, S4{T21,T64, T81}};
    - FWp5= {S11{T40,T56,T69},S8{T31,T35}, S9{T2,T82}};

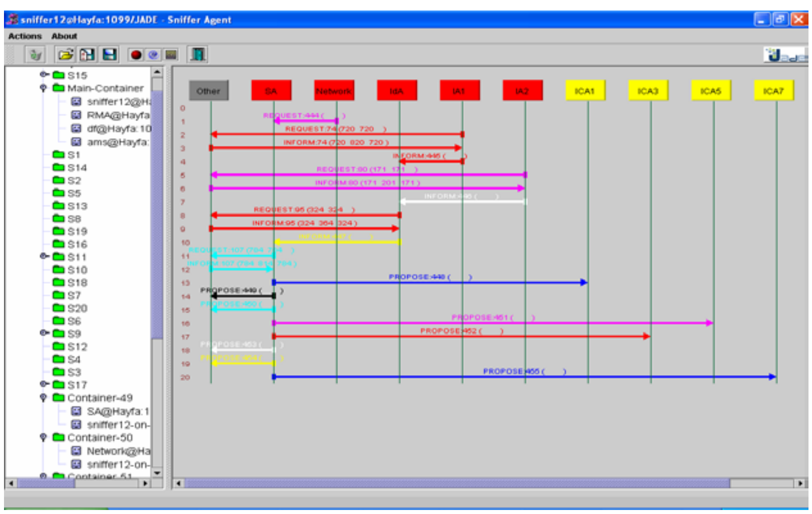

Figure 10. Message Exchange Between Agents

On this window of the Sniffer graphic tool (figure 10), we can see available servers containers on the network, where ICA agents can move in order to collect data according to the adopted contract model.

7.1 First Scenario

This scenario is applied on one transport system (for example: the French transport operator) where it uses French ontology. In this study, we try to illustrate the case when ICAs agents don't understand the messages sent from the SA agent, although, the agents use the same ontologies and languages to start the negotiation process. In this scenario, there are two possibilities of the occurrence of the misunderstanding in our system: the first possibility will take place after sending the first message (*Propose (contract)*) by the SA agent to the ICAs agents, where some of these later discover that they don't understand the contents of the message. Before sending any message to the SA, ICAs ask the SEL by using the predicate *P* to determine the level of transibility between the AS ontology and ICA ontology. According to the result of this predicate, ICAs decide what it must send to SA (i.e. they will send either: accept (partial) or refuse).The ICAs agents send their agreements if they would like to participate in the negotiation process. For this purpose, the SEL evoke the KMSL, which in turn, verifies DOnto to determine the domain ontology, then, it goes to the OntoSV which uses KT to translate the terms and the sentences among the determined ontologies. The second possibility is when the ICAs know their FWps, like our example. The agents ICAs are supposed to visit their first nodes by the order as in their FWps without problems before the declaration of all unavailable nodes. In this case, the proposed negotiation process allows us to reassign the nodes (i.e. new negotiation tour).We suppose the set of the nodes which is not available, as follow:

$$Ind = \{S1,S3,S7,S14,S5,S17,S12,S9,S13,S19\};$$

We deduce the tasks to reallocate them:

Φ={T19,T29,T66,T88,T3,T26,T32,T33,T38,T42,T61,T85,T34,T71,T20,T86,T95,T99,T25,T44, T60,T80,T75,T83,T2,T82,T53,T6,T16,T22,T52,T57,T67,T96}.

Then, we have 34 tasks to be reallocated in the second tour of the negotiation process.

Upon the reception of the proposed contract, each ACI responds by a partial agreement because, in this case, the perturbation has affected a subset of each FWps and each ICA agent must verifies its ontology to participate in this negotiation. By take three examples to explain this case. According to our choice of these agents, ф is as follow:

Φ={T19,T29,T66,T88,T3,T26,T32,T33,T38,T42,T61,T85,T6,
T16,T22,T52,T57,T67,T96,T2,T82,}

- The ICA1 does not visit the nodes S1, S3 and so it informs the SA that it no longer executes the tasks {T19, T29, T66, T88, T3, T26, T32, T33, T38, T42, T61, T85}.
- The ICA3 does not visit the nodes S19 and so it informs the SA that it no longer executes the tasks {T6, T16, T22, T52, T57, T67, T96}.
- The ICA5 does not visit the nodes S9 and so it informs the SA that it no longer executes the tasks {T2, T82}.

In this state, the SA asks each ICA to propose a new set of reassignment ф, according to its priorities. As follow:

- The task T3 is proposed by :
  - ICA1 in the nodes S20 and S15;
  - ICA2 in the nodes S18,S10;
- The task T2 is proposed by
  - ICA1 in the nodes S20, S15;
  - ICA5 in the nodes S8;
- No agent accepts to reassign the task T16 because the proper servers are not available (S5, S7, S19). In this case, the French transport system remark that the T16 is the travel from B=Paris to C= London today (at the instant *t*=11.00 or starting from 12.00) (i.e. it can demand reassignment of T16 from the English transport operator, as we will see in the next scenario).
- No agent accepts to reassign the task T19 because the proper servers are not available (S1, S5).

When the AS agent receives all these propositions, it decides a new contract, as follows:

- Direct reassignment of the tasks is, as follow:
  {T29,T66,T61,T85, T22,T82} (unique choice)
- According to an optimization, direct reassignment of the tasks is, as follows:
  T3 (S20), T26 (S11), T32 (S8), T33 (S16), T42 (S18), T2 (S8), and T6 (S18).
  (multiple choice)

The answers of the ICAs on the SA agent propose are:

- The ICA1 refuses the choice where the SA agent decides to reassign the task T3 to S20 because the later has left this task and accept the rest of the proposition (partial accept).
- The agents ICA3, ICA5 accept the new contract (accept total)

Thus, SA agent updates ф, as follows:

Φt= {T3, T19, T88, T38, T16, T52, T57, T67, T96}

The SA agent confirms the rested road of the ICAs agents and asks each ICA agent to propose a new set of assigned nodes ф, according to its priority.

- The ICA1 agent proposes to execute the T3 in the node S15

In this new tour of negotiation, when the SA agent receives all these propositions, it sends a new contract to the ICAs agent. Then, the ICA1 agent send an accept (total) which is confirmed by the SA agent.Thus, SA agent updates ф, as follows:

Φt= {T19, T88, T38, T16, T52, T57, T67, T96}

In this example, the SA agent remarks that it can send the T16 to anther transport operator (in our example, English transport operator) by using the Meta-System to continue the negotiation process. But for the rest of tasks which need to reassign by servers of the same transport system (French transport system) as T19. In this case, the negotiation for this set of tasks will be stopped.

7.2 Second Scenario

In the second scenario, we apply our NOKMS on multi-transport operators (for example: French transport operator, English transport operator and German transport operator), which are heterogeneous community of multi-agent systems. The French's customers want to travel to other cities out of France. The French transport system (Sys1), in this case, firstly its SA agent sends the propos (contract) message to its ICAs participant, as we noted in the first scenario. In some times, ICAs agent cannot reassign all the tasks as T16 where this task can be achieved by another system like English transport system (Sys2). The usage of another transport system comes from the flexibility of our NOKMS architecture.

In this state, the Sys1 sends their query to the Sys2 through the Meta-System which considers as the intermediate between the two systems, and which in turn, interprets the incoming ACL-Sys1 based on its NOKMS structure. The interpreted message is then converted into an 'interlingua' representation inside the Meta-System. Where, The Meta-System translates the Interlingua representation to the destination ACL-Sys2. as an example: when the Sys1 have found that it cannot reassign the task T16 and this task can be assigned by another system, then it send this task to the Meta-System using its French Transport Ontology as follow:

- T16="Voyager de l'endroit B à l'endroit C (aujourd'hui, à l'instant t/aujourd'hui, à partir de /:00)" : The Meta-System is then tries to translate this task, where firstly it verifies the level of transibility between the two ontologies in its SEL. The later evokes the KMSL which translate the proposed expression to the determined ontology (English Transport Ontology in our case), then the KSML return the following result:
- T16=" travel from B to C today(at the instant t=11.00 or starting from 12.00)"

In some cases, when KMSL verifies the IKB, and find that it has not the suitable ontology to translate the coming ontology form Sys1to Sys2. The KMSL uses the KC to create the new ontology correspond to the ontology of Sys2 according to some policies (this paper don't explain these cases). After translating the ACL-Sys1, Meta-System sends the new ACL which correspond to the ACL-Sys2 to the Sys2 to start the new tour of negotiation between the two systems. The Meta-System currently adopts FIPA semantic model which described in Semantic Language (SL) [32], as the Interlingua of the agent communications.

7.3 Negotiation Tours

The proposed negotiation process allows the reassignment of the cancelled services. The two charts (figure 11) below represent different generated optimal solutions instances assignments for the same network error scenario, where the first one represents the negotiation torus without using our NOKMS and the second represent it with our NOKMS. Remark that our proposed NOKMS give the flexibility to find new available providers out of its system where it doesn't find the suitable providers in it as in the case of the task T16. Through an agreement between its SA agents and the new ICAs agents in the new system .the two systems connect with each other by Meta-System which consider as the intermediate between the two systems. So the correspondent transport users are satisfied in spite of some network perturbations.

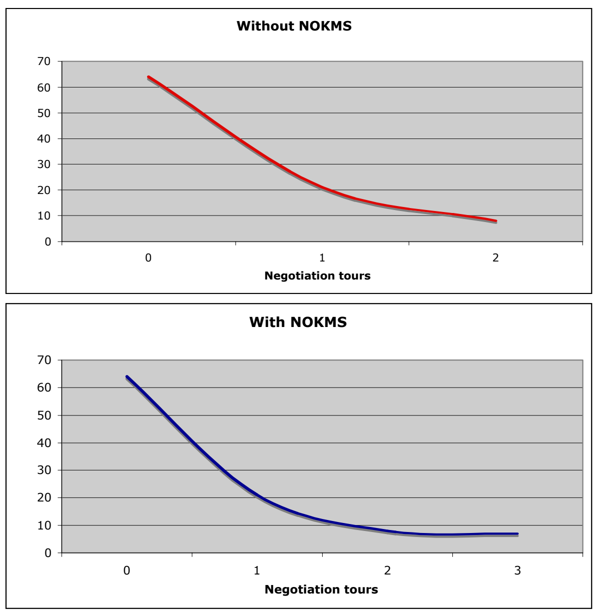

Figure 11. Negotiation Tours According to the NOKMS Application

8. Conclusion and Future Works

In this paper we have presented a new solution for the problem of language interoperability between negotiation agents, by incorporating architecture for Negotiation process which uses an Ontology-based Knowledge Management System (NOKMS). The proposed solution prevent the misunderstanding during the negotiation process though the agents' communications. The architecture consists of three layers: the Negotiation Layer (NL) that describes the negotiation process between Initiator agents and Participant Agents by using suitable ontologies, the Semantic Layer (SEL) contains the semantic translator which uses in the case of when the agents didn't understand the negotiation messages , and the last one is the Knowledge

Management Systems Layer (KMSL) which base on the Intelligent Knowledge Base (IKB) to give the flexibility to our negotiation ontology. Finally, we present a case study which applies our architecture on the *Multimodal Transport Information System (MTIS)* project where we illustrated two scenarios applicable: the first use our negotiation ontology architecture in one transport system (French transport system), and the second apply this architecture on the multi-transport systems (French and English transport systems). These scenarios presented that the proposed NOKMS improves the communications between heterogeneous of negotiation process in multi-agents systems in order to satisfy the transport customers.

In this paper, we have presented only the different ontology combinations problems but in the future, we will try to find a novel method for ontology (mepping, merging, alignment) negotiation selon our project, in which agents are able to achieve. We will try to implement this architecture by using Java Agent DEvelopment framework (JADE) which includes a proficient support for content languages and ontologies.

REFERENCES

1. ZGAYA, H., SLIM HAMMADI, C**onception et distribuée d'un système d'information d'aid à la mobilité urbaine: Une approche multi-agent pour la recherche et la composition des services liés au transport**, PHD, EC-Lille, Lille, France, July 6, 2007.
2. ZGAYA H., and S. HAMMADI, **Multi-Agent Information System Using Mobile Agent Negotiation Based on a Flexible Transport Ontology**, In Proceedings of the 2007 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS'2007). May 14-18, Honolulu, Hawaii, 2007.
3. ABOU ASSALI, A., D. LENNE and B. DEBRAY: **KoMIS: An Ontology-Based Knowledge Management System For Industrial Safety**. (Database and Expert Systems Applications DEXA'2007), Regensburg, Germany, 3-7 September, 2007.
4. DIGGELEN, J. VAN, BEUN, R.J., DIGNUM, F.P.M., EIJK, R.M. VAN, MEYER, J-J. CH. **Ontology Negotiation in Heterogeneous Multi-agent Systems: the ANEMONE System**, IOS Press 2007.
5. SH. JI, T. QIJIA, Y. LIANG, H.YANG, **An Ontology Framework for EC Automated Negotiation Protocol**, Networking, and Parallel/ Distributed Computing, 2007. SNPD 2007, July 30, 2007 - Aug. 2007.
6. SCHIEMANN, B., **OWL DL as a FIPA ACL content language**, Ulf Schreiber Artificial Intelligence Division, Department of Computer Science, 2006.
7. DAVIES, J., R. STUDER, P. WARREN, Semantic Web Technologies: Trends and Research in Ontology-based Systems, April 2006.
8. LACLAVIK, M., **Ontology and Agent based Approach for Knowledge Management PhD**, Institute of Informatics, Slovak Academy of Sciences, 2006.
9. DIGGELEN, J. VAN, BEUN, R.J., DIGNUM, F.P.M., EIJK, R.M. VAN, MEYER, J-J.CH. **Optimal Communication Vocabularies and Heterogeneous Ontologies**, IOS Press 2005.
10. BRAVO, M.C., J. PEREZ, V.J. SOSA, A. MONTES, G .REYES, **Ontology support for communicating agents in negotiation processes,** Hybrid Intelligent Systems, 6-9 November, 2005.
11. ZGAYA, H., S. HAMMADI, K. GHÉDIRA, **Evolutionary method to optimize Workplan mobile agent for the transport network application**, In Proceedings of International

Conference on Systems, Man and Cybernetics (IEEE SMC'2005), Hawaii, USA, October 10-12, 2005.

12. ZGAYA, H., S. HAMMADI, K. GHÉDIRA, **Workplan Mobile Agent for the Transport Network Application,** In Proc. of 17th IMACS World Congress Scientific Computation, Applied Mathematics and Simulation (IMACS'2005), Paris, France, July 11-15, 2005.
13. SRIDHARAN, B., A. TRETIAKOV and KINSHUK, **Application of Ontology to Knowledge Management in Web based Learning**, ICALT 2004.
14. SINTEK, M., **OntoViz Tab: Visualizing Protégé Ontologies**, 2004, http://protege.stanford.edu/plugins/ontoviz/ontoviz.html
15. OBITKO, V., MARÍK, **OWL Ontology Agent based on FIPA proposal**, Znalosti; 2004, Brno, Czech Republic, 2004.
16. JADE **- Java Agent Development Framework**, http://jade.tilab.com/, Accessed July 2004.
17. RAZMERITA, L., A. ANGEHRN and A. MAEDCHE , O**ntology-Based User Modeling for Knowledge Management Systems**, DOCIS, 2003.
18. MATHIEU, M., H. VERRONS, **ANTS: an API for creating negotiation applications,** Conf. CE2003.
19. BAADER, F., D. MCGUINNESS, D. NARDI, The Description Logic Handbook, ISBN: 0521781760, January 9, 2003.
20. BORNE; P., B. FAYECH; S. HAMMADI, S. MAOUCHE, **Decision support system for urban transportation networks,** Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on Volume 33, Issue 1, Feb. 2003, pp. 67 – 77.
21. TAMMA, V., and T.J.M., **An ontology model to facilitate knowledge sharing in multi-agent systems**, In Knowledge Engineering Review, 17(1), Bench-Capon, 2002, pp. 41-60.
22. Pellucid Consortium, Pellucid IST Project Website, http://www.sadiel.es/ Europa/ pellucid/, 2002.
23. University of Amsterdam, Ontology Bean Generator for JADE, http://www.swi.psy.uva.nl/usr/aart/beangenerator/ 2002
24. FENSEL, D., **Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce**, 2001.
25. KLEIN, M., **Combining and relating ontologies: an analysis of problems and solutions**, In IJCAI-2001 Workshop on Ontologies and Information Sharing, Seattle, WA, 2001, pp. 53–62.
26. STUDER, R., V. BENJAMINS, and D. FENSEL, **Knowledge engineering, principles and methods**, Data and Knowledge Engineering, 25(1-2):161–197, 1998.
27. MESGHOUNI, K., S. HAMMADI and P. BORNE, **On modeling genetic algorithm for flexible job-shop scheduling problem**, Studies in Informatics and Control Journal, Vol. 7, 1998, pp. 37-47.
28. FALASCONI, S., G. LANZOLA, and M. STEFANELLI, **Using ontologies in multi-agent systems**, In Proceedings of Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW), Banff, Alberta, Canada, 1996.
29. GEIGER, K., **Inside ODBC. Microsoft Press**, 1995.

30. GRUBER, TH. R., **A Translation Approch to Portable Ontology Specification**, Knowledge Acquisition, April, 1993.

31. FIPA0081: FIPA ACL Message Structure Specification, http://www.fipa.org/specs/fipa00061/index.html .

32. FIPA0008: FIPA SL Content Language Specification, http://www.fipa.org/specs/fipa00008/index.htm.

33. FIPA, Foundation for Intelligent Physical Agents website, 2002, http://www.fipa.org/

34. FIPA. FIPA0086: FIPA Ontology Service Specification, http://www.fipa.org/specs/fipa00086/index.html