

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**4,800**

Open access books available

**122,000**

International authors and editors

**135M**

Downloads

Our authors are among the

**154**

Countries delivered to

**TOP 1%**

most cited scientists

**12.2%**

Contributors from top 500 universities



**WEB OF SCIENCE™**

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.

For more information visit [www.intechopen.com](http://www.intechopen.com)



# Towards the Optimization of Client Transport Services: Negotiating by Ontology Mapping Approach between Mobile Agents

Sawsan Saad, Hayfa Zgaya and Slim Hammadi  
*LAGIS UMR 8146, Central School of Lille  
France*

## 1. Introduction

This work belongs to the national French project VIATIC.MOBILITE from the industrial cluster I-trans\*, which is an initiative bringing together major French players in rail technology and innovative transport systems. In fact, Transport users require relevant, interactive and instantaneous information during their travels. A Transport Multimodal Information System (TMIS) can offer a support tool to response and help network customers to make good decisions when they are travelling providing them all needed information in any existent and chosen format (text, multimedia...), in addition, through different handheld wireless devices such as PDAs, laptops, cell phones, etc. So in a previous work (Zgaya, 2007a), we proposed a Multi Information System (MIS) based on a special kind of software agent called Mobile Agent (MA) (Carzaniga et al., 1997). The realization was successful, thanks to a two-level optimization approach (Zgaya et al., 2007b), where the system optimizes the selection of nodes to answer the different requests. Our customer is satisfied if he obtains rapidly a response to his request, with a suitable cost.

But in the case of network errors, the MAs begin the negotiation process which allows new assignments to cancelled services to available network nodes. For this purpose, we designed a negotiation protocol intended for the transport area which permits to the agents to negotiate when perturbations may exist (Zgaya et al., 2007c). Our protocol uses messages to exchange the information. Those messages are exchanged between initiators and the participants in the negotiation process. Indeed, this protocol has studied before only the cases of the simple messages without using ontology and did not include the solutions when the participant agents did not understand the messages sent from the initiators agent. Thus, we propose an approach that will improve the negotiation protocol through the multi-agent systems by adding ontology in the negotiation process. Our solution bases on the knowledge management system to facilitate automatically the management of the

---

\* <http://www.i-trans.org>

negotiation messages and to solve the semantic heterogeneity. In our proposal, we incorporate architecture for negotiation process with that uses an Ontology-based Knowledge Management System (NOKMS) (Saad et al., 2008c). The architecture consists of three layers: (Negotiation Layer (NL), Semantic Layer (SEL) and Knowledge Management System Layer (KMSL)). But in this work we talked about only (NL and SEL) that describes the negotiation process as well as illustrates the different messages types by using the different ontologies. Our proposed NOKMS improves the communications between heterogeneous negotiation mobile agents and the the quality of service (QoS) response time with the best cost in order to satisfy the transport customers

This paper is organized in six parts, as follow: in the second section, we discuss some related work. Then, we illustrate the ontology mapping idea. We present in section 4 the global system architecture describing its general functioning. In section 5, we illustrate our negotiation protocol with using the ontology approach. A case study will discuss in (Section 6). Finally, conclusion and prospects are mentioned in last section.

## 2. Related Work

Negotiation is a process by which two or more parties make a joint decision (Zhang et al., 2005). Negotiation has been done by different research works; (Bravo et al. 2005) presented a semantic proposition for manipulating the lack of understanding messages between the seller and buyer agents during the exchange of messages in a negotiation process. Otherwise, (Zgaya et al., 2007c) provided a negotiation protocol for the transport area to facilitate the communications between the agents. A generic negotiation model for multi-agent systems has been proposed by (Verrons et al., 2004), 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. Negotiations can be used to resolve conflicts in a wide variety of multi-agent domains. In (Jennings et al., 2000), an application include conflicts illustrated the usage of joint resources or task assignments, conflicts concerning document allocation in multi-server environments and conflicts between a buyer and a seller in electronic commerce.

For ontology approach, it has an important role in the multi-agent systems. In fact, there are many of definitions of the ontology according to the different domains where we use it. Firstly, 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 conceptualization of a domain of interest*' (Gruber, 1993). In this definition, a *conceptualization* 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 *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 (Studer et al., 1998), (Obitko et al., 2004).

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 recognizes to be the concepts describing the application domain which is associated with the agent together with

their relationships and constraints (Falasconi et al., 1996). 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

### 3. Ontology Mapping

Ontology mapping process aims to define a mapping between terms of source ontology and terms of target ontology. The mapping result can be used for ontology merging, agent communication, query answering, or for navigation on the Semantic Web.

The approach for ontology mapping varies from lexical to semantic and structural levels. Moreover, the mapping process can be grouped into data layer, ontology structure, or context layer. The process of ontology mapping has five steps: information ontology, obtaining similarity, semantic mapping execution and mapping post-processing (Maedche and Motik, 2003). The most important step of ontology mapping is the computation of conceptual similarity. First define similarity:

Sim:  $w_1 w_2 o_1 o_2 \rightarrow [0, 1]$ , the similar value from 0 to 1.

Sim (A, B) denote the similarity of A and B.  $w_1$  and  $w_2$  are two term sets.  $O_1$  and  $O_2$  are two ontologies.

Sim (e, f) = 1: denote concept e and concept f are completely sameness.

Sim (e, f) = 0: denote concept e and concept f are completely dissimilar.

## 4. The Proposal Architecture

### 4.1 General System

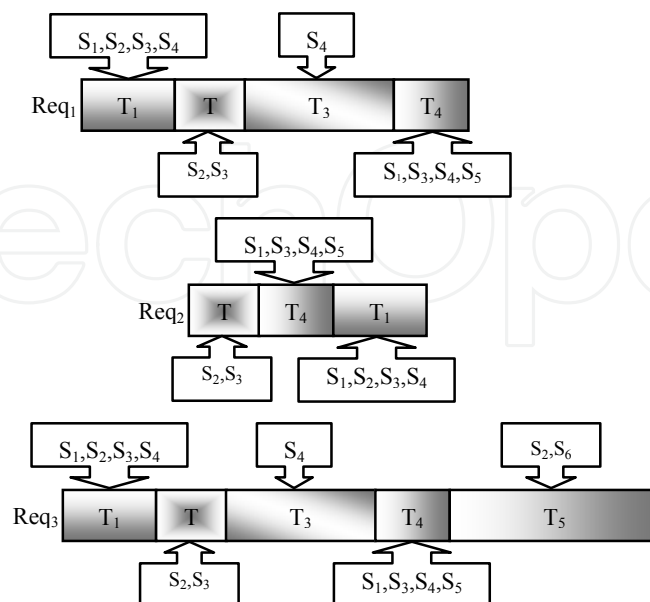


Fig. 1. Nodes identification

Firstly, we will illustrate the problem by which our TMIS bases. From general point of view, our system has a two-step assignment problem: firstly the assignments of network nodes to MAs to build their initial Workplans and then, a sub-set of these nodes are selected to assign tasks. A task is an independent sub-request which belongs to one or several requests formulated simultaneously by different customers. So, information providers which propose services corresponding to identify tasks are recognized (figure 1). Consequently, nodes must be assigned to tasks in order to satisfy all connected users and respecting delays of responses and minimizing their cost (QoS).

To resolve the described problem, we have proposed a system based on the coordination of five kinds of software agents (Zgaya et al., 2007b, 2007c) (figure 2):

- 1) **Interface Agents (IA):** These agents interact with system users, allowing them to choose appropriate form of responses to their demands so IA agents manage requests and then display results. When a multimodal network (MN) customer access to the MIS, an agent IA deals with the formulation of his request and then sends it to an available identifier agent. This one relates to the same platform to which several users can be simultaneously connected, thus it can receive several requests formulated at the same time.
- 2) **Identifier agents (IdA):** This agent manages the decomposition of the requests which were formulated through a same short period of time  $\mathcal{E}^*$  ( $\mathcal{E}$ -simultaneous requests). The decomposition process generates a set of sub-requests corresponding, for example, to sub-routes or to well-known geographical zones. Sub-requests are elementary independent tasks to be performed by the available set of distributed nodes (information providers) through the Transport Multimodal Network (ETMN). Each node must login to the system registering all proposed services. A service corresponds to the response to a defined task with fixed cost, processing time and data size. Therefore, an agent IdA decomposes the set of existing simultaneous requests into a set of independent tasks, recognizing possible similarities in order to avoid a redundant search. The decomposition process occurs during the identification of the information providers. Finally, the agent IdA transmits cyclically all generated data to available scheduler agents. These ones must optimize the selection of providers, taking into account some system constraints
- 3) **Scheduler Agents (SA):** Several nodes may propose the same service with different cost and processing time and data size. The agent SA has to assign nodes to tasks minimizing total cost and processing time in order to respect due dates (data constraint). Selected set of nodes corresponds to the sequence of nodes building Workplans (routes) of the data collector agents. The agent SA has firstly to find an effective number of collector agents then he has to optimize the assignments of nodes to different tasks. This behaviour will be developed later.
- 4) **Intelligent Collector agents (ICA):** An agent ICA is a mobile software agent which can move from a node to another through a network in order to collect needed

---

\* Fixed by the programmer

data. This special kind of agent is composed of data, code and a state. Collected data should not exceed a capacity threshold in order to avoid overloading the MA. Therefore, the agent SA must take into account this aspect when assigning nodes to tasks. When they come back to the system, the agents ICA must transmit collected data to available fusion agents.

- 5) **Fusion Agents (FA):** These agents have to fusion correctly collected data in order to compose responses to simultaneous requests. The fusion procedure progresses according to the collected data availability. Each new answer component must be complementary to the already merged ones. Providers are already selected and tasks are supposed independent. Therefore, there is no possible conflict. A response to a request may be complete if a full answer is ready because all concerned components are available. It can be partial if at least a task composing the request was not treated, for example, because of an unavailable service. Finally, a response can be null if no component is available. If an answer is partial, the correspondent result is transmitted to the concerned user through the agent IA which deals with request reformulation, with or without the intervention of the user.

To respond the tasks, needed data is available through the ETMN and their collect corresponds to the jobs of ICA agents. Then, it must search the optimizing solution to solve the problem of the assignment process. This optimization is the topic of the SA behaviour explicit in the next section.

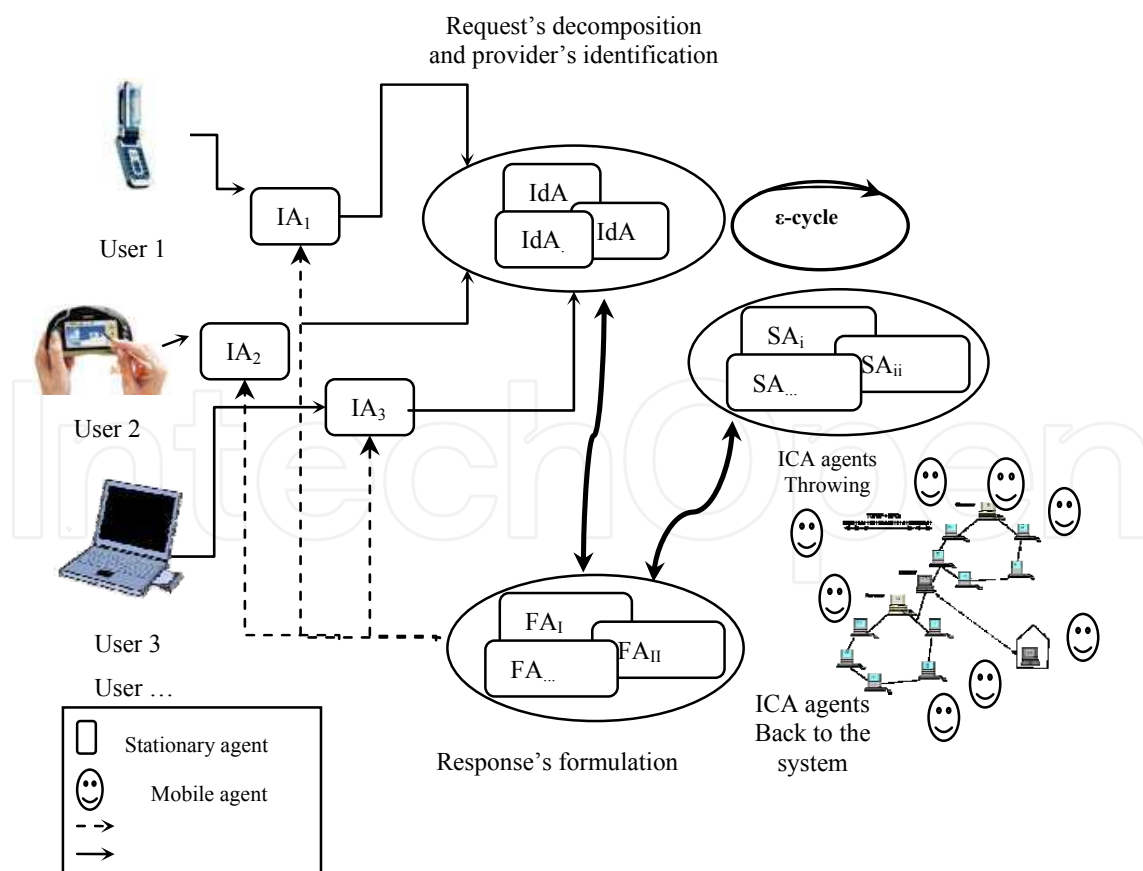


Fig. 2. Multi-Agent Approach



### 3.2 The Optimizing Solution by Scheduler Agents SA Behavior

Since his creation, the SA agent calculates an actual number of ICA agents that created at the same time, and then he gives everyone an Initial Workplan (IWp) which updates whenever the network status varies considerably. When the IdA agent, from the same society (we call agents IdA, SA, FA and ICA created at the instant  $t$  the agents society), gives him a number of tasks thus the SA agent has to begin the optimization process (Figure 3).

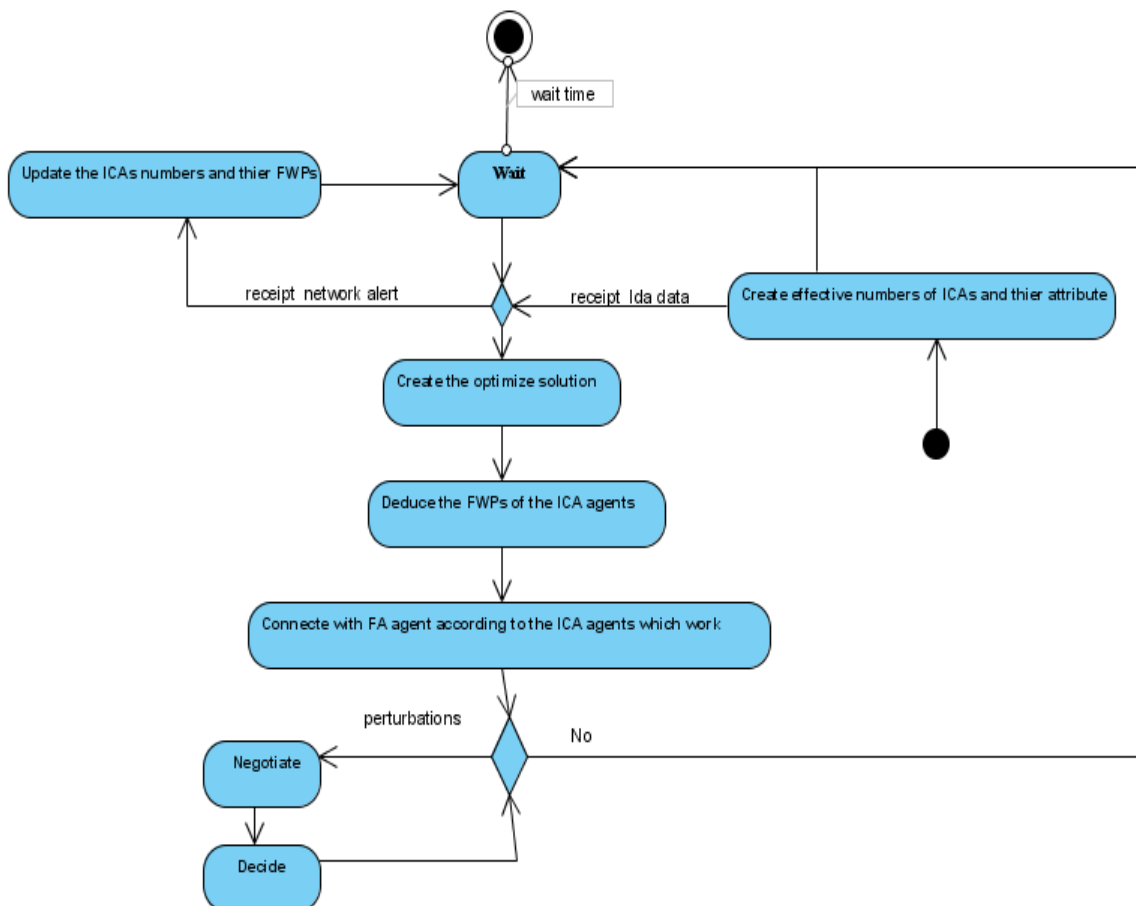


Fig. 3. SA Behaviour

The SA agent has to optimize the assignments of nodes to the exiting tasks, by minimizing total cost and processing time to respect due dates. To solve this assignment problem, we proposed a two level optimization solution, expressing the complex behaviour of an agent SA, which was already studied and implemented in previous works (Zgaya et al., 2007b, 2007c). The first level aims to find an effective number of ICA agents, building their initial Workplans in order to explore the ETMN completely (Zgaya et al., 2007b). The second level represents the data flow optimization corresponding to the nodes selection in order to increase the number of satisfied users (Zgaya et al., 2007c). This last step deduces final Workplans of ICA agents from initial ones, by using Evolutionary Algorithms (EA). So we have designed an efficient coding for a chromosome (the solution) respecting the problem constraints (Zgaya, 2007a). A possible solution is an instance of a flexible representation of the chromosome, called Flexible Tasks Assignment Representation (FeTAR). The chromosome is a matrix  $CH(I' \times J')$  where rows represent independent identified tasks

(services), composing globally simultaneous requests and columns represent recognized distributed nodes (providers). Each element of the matrix specifies the assignment of a node  $S_j$  to the task  $T_i$  as follows:

$$CH [i, j]= \begin{cases} 1: \text{if } S_j \text{ is assigned to } T_i ; 1 \leq i \leq I' \text{ and } 1 \leq j \leq J' \\ * : \text{if } S_j \text{ may be assigned to } T_i \\ X: \text{if } S_j \text{ cannot be assigned to } T_i \end{cases}$$

We notice that each task must be assigned, so we assume that each task must be performed at least by a node, selected from a set of nodes proposing the service which corresponds to a response to the concerned task where this is the first selection step. After that, we apply the second selection step which is one of the most important aspects of all EA. It determines which individuals in the population will have all or some of its genetic material passed on the next generations. We have used random technique, to give chance to weak individuals to survey: parents are selected randomly from current population to crossover with some probability  $p_c$  ( $0 < p_c < 1$ ).

In our case, we use the fitness function where a chromosome is firstly evaluated according to the number of responses which respect due dates, namely responses minimizing correspondent ending dates and respecting correspondent due dates. Then a solution is evaluated according to its cost. Therefore, a chromosome has to express ending responses date and the information cost. As we mentioned, a request  $req_w$  is decomposed into  $I_{t,w}$  tasks. Therefore, the total processing time  $EndReq_w$  for each  $req_w$  is computed by the means of the algorithm *fitness\_1* below. This time includes only the effective processing time on the MN. We assume that, the ending date  $D_w$  corresponding to the total execution time of a request  $req_w$ , includes also the average navigation time of ICA agents. This is expressed by:

$$\gamma = \delta - \frac{\sum_{j=1}^J CT_j}{J} \quad (1)$$

$$\Rightarrow \forall 1 \leq w \leq R, D_w = EndReq_w + \gamma \quad (2)$$



**Fitness\_1 algorithm**

Step 1:

$m'$  is the ICA agents number so

$\forall k$  with  $1 \leq k \leq m'$ , initialize :

- The set of tasks  $U_{ck}$  to  $\emptyset$
- Total time  $EndU_{ck}$  to perform  $U_{ck}$  to 0

Step 2:

Look for the set of tasks  $U_{ck}$  performed by each  $ICA_{ck}$  and their processing time  $EndU_{ck}$  as follows:

for  $k := 1$  to  $m'$

  for  $j := 1$  to  $J'$

    for  $i := 1$  to  $I'$

      if  $S_{cj}$  belongs to the Workplan of  $ICA_{ck}$  and  $S_{cj}$  is assigned to  $T_{ci}$

        {  
          $U_{ck} := U_{ck} \cup \{T_{ci}\};$   
          $EndU[ck] := EndU[ck] + P_{cicj};$   
         }

Step 3:

Compute processing time of each request require the identification of ICA agents which perform tasks composing the request. Total processing time of a request is the maximum processing times of all ICA agents which perform tasks composing this request. This is calculated as follow:

for  $w := 1$  to  $R$

{

  for  $k := 1$  to  $m'$

$treatedAC[ck] := false;$

$EndReq[w] := 0;$

$i := 1;$

      while  $i \leq I'$  and  $\exists k_1 / 1 \leq k_1 \leq m'$  and

$treatedAC[ck_1] = false$

      {

        if  $T_{ci} \in req_w$

        {

$ck := 1;$

          while  $k \leq m'$  and  $T_i \notin U_k$

$ck := ck + 1; // end while$

          if  $\neg TreatedAC[ck]$

          {

$EndReq[w] := \max(EndReq[w], EndU[ck]);$

$TreatedAC[ck] := true;$

```

        } //end if
    } //end if
} //end while
} //end outer for-loop
    
```

Form the other side, total cost of a request  $req_w$  is  $CostReq[w]$  expressed by  $C_w$ , is given by the mean of the algorithm below:

**Fitness\_2 algorithm**

Repeat steps 1 and 2 for each request  $req_w$  ( $1 \leq w \leq R$ )

Step 1:

$CostReq[w] := 0$

Step 2:

for  $i := 1$  to  $I'$

{

if  $T_{ci} \in req_w$  {

find the node  $S_{cj}$  ( $1 \leq j \leq J'$ ) assigned to  $T_{ci}$   
 in FeTAR instance

$CostRe[w] := CostRe[w] + Co_{cicj}$

} //end if

} // end for

Knowing that by using expression (1), we can deduce ending date from fitness\_1 algorithm, the new FeTAR representation of the chromosome express for each request  $req_w$   $1 \leq w \leq R$ , its ending date and its cost.

An example of a generated FeTAR instance with  $I'=8$  and  $J'=10$ , where the evaluation of this chromosome is illustrated by a evaluation vector which explicit: for each  $req_w$ , its total cost ( $C_w$ ) and the total time required for his response ( $D_w$ ). The average cost of all requests and the response time can be deducted from generated vector, can be illustrated as follows:

$w$	$d_w$	$C_w$	$D_w$
1	10	5	6
2	5	1	1
3	10	4	2
4	5	3	2
4	3	2	1
6	5	3	2

CH	$S_1$	$S_{13}$	$S_{24}$	$S_{55}$	$S_{68}$	$S_{70}$	$S_{71}$	$S_{78}$	$S_{79}$	$S_{93}$
$T_8$	*	*	*	*	1	*	*	*	*	*
$T_{12}$	*	*	*	*	x	*	*	1	*	*

T <sub>18</sub>	*	1	*	*	x	*	*	*	*	*
T <sub>22</sub>	*	*	*	*	x	*	1	*	*	*
T <sub>35</sub>	x	*	1	x	x	x	*	*	x	x
T <sub>51</sub>	x	x	*	*	x	x	x	1	*	*
T <sub>52</sub>	*	*	*	1	x	x	x	x	*	*
T <sub>58</sub>	*	*	*	1	x	x	*	*	*	*

Table. 1. Example of a FeTAR instance

In this work, we are interested into the interaction between SA agents and ICA agents, especially in case of some network disturbances. In that case, these two kinds of agents have to negotiate the reassignment of tasks which still need providers. We will illustrate that in the rest.

## 4. The Negotiation Ontology Protocol

### 4.1 Problem Description

Some perturbations can occur during the mobile agents moving through the distant network nodes (bottleneck, failure, crash...). In this case, the ICA agents have to avoid the unavailable nodes in their remained final Workplans. In addition, they have to change their itineraries in order to take into account the cancelled tasks which still need assignment because of the conflicts. Therefore, a new assignment process has to occur to find suitable new available providers (Saad et al., 2008a) To do this, we have to benefit of active ICA agents who are still travelling through the network and to exploit new ones otherwise. So ICA agents have to interact with SA agents in order to find suitable solution to the current situation. Thus, in (Zgaya et al., 2007c) we proposed a negotiation process inspired from the well-known contract net protocol (CNP) between the ICA agents who represent the participants of the negotiation and SA agents who are the initiators.

This protocol has studied before only the cases of the simple messages and it proposed ontology without illustrating it, and this later didn't illustrate the problem which will take place when the participants don't understand the communication messages, or when the new agent wants to participate in a negotiation process Thus agent must to understand the protocol and the communication language messages, in this case the agents need an interoperable language between themselves for understanding each other. But as we know in open and dynamic environments (such as the Web and its extension the Semantic Web) are by nature distributed and heterogeneous. In these environments ontologies are expected to complement agreed communication protocols in order to facilitate mutual understanding and interactive behaviour between such agents. Thus, agents may differ in their view of the world, creation, representation and exploration of domain ontologies they commit to. Because, for each common domain ontology; people may store their data in different structures (structural heterogeneity) (Malucelli and Oliveira, 2004). And they use different terms to represent the same concept (semantic heterogeneity). Moreover there is no formal mapping between ontologies.

### 4.2 Initiators

An initiator of a negotiation is a SA agent who never knows the exact position of each

travelling ICA agent. However, he knows all initial Workplans schemes and the final assignments of the servers (final effective Workplans). SA agent does not need to wait for all answers to make a decision, since he can accept a subset of responses to make pressing sub-decisions; urgent actions must be made according to the current positions of ICA agents. Consequently, SA agent can make decisions every short period of time. In that case, he must update the set of services which need to be reassigned by providers through the confirmation step. After that, he has to propose a new contract according to the updated services set and to the different capabilities of the participants of the negotiation. We suppose that errors on the network are identified before that an ICA agent leaves one functioning node towards a crashed one.

### 4.3 Participants

For a given task, the participants may respond with a proposal or a refusal to negotiate. In our protocol we have two types of participants in negotiation process according to the SA agent propose.

#### 4.3.1 Intelligent Collector Agents (ICAs)

A participant of a negotiation is Intelligent Collector Agents ICAs who never knows anything about the other participants of the same negotiation process. Obviously, he knows his own initial Workplan scheme and his final assignments of servers (final effective Workplan). In addition, each ICA agent has his own priorities, preferences and constraints which are dynamic, depending on the network state and on his current position in the already defined final Workplan. He has own ontology too.

- *Constraints* of an ICA agent express the tasks which he can't perform or the servers which he can't visit because they cause problems (overloading, time consuming, high latency...).
- *Priorities* express servers where the ICA agent prefers visit because they are already programmed in his remained final Workplan.
- *Preferences* express servers which are already programmed in the remained initial Workplan but not in the final one.
- *Ontology*, if 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 (Diggelen et al., 2005). General Ontology defines the Cv with which queries and assertions are exchange between agents. But one of the big problems to communication-based agents is that each one uses different terms with the same meaning or the same term for different meanings. Once we took this problem as a challenge, representing these differences in a common ontology becomes essential. The ontology includes the entire domain's knowledge, which is made available to all the components active in an information system. The use of a common ontology guarantees the consistency (an expression has the same meaning for all the agents) and the compatibility (a concept is designed, for the same expression, for any agent) of the information present in the system. However, it is

not sure that all the agents will use a common ontology. Usually, each agent has its heterogeneous private ontology and it cannot fully understand other agent's ontology. In our system, each time an ICA agent receives a new contract; it analyzes it to make a decision (refusal or total/partial acceptance) according to its ontology.

#### 4.3.2 Translation Agents (TAs)

Another participant of a negotiation is a Translation Agents TAs. TA responsible for providing the translation services that support the negotiation agents (i.e. SA agents and ICT agents). Thus, it helps solving the interoperability problems. TA uses a dictionary (or a lexical database, in our system, we use EuroWordNet) to obtain the set of synonyms terms of each term from the source ontology. The task of TA consists of applying methodology to detect semantic similarities between two concepts in the conversion between different ontologies. Once the TA has established the similarity between a pair of terms from different ontologies, this knowledge is stored in Knowledge Management System Layer (KMSL) (Saad et al., 2008b) in order to be available for future negotiation rounds. The intelligent of this system is improved occurs with time, because the matched terms is memorized. When the number of negotiations rounds increases; we aim that our system by using TA provides the following services:

- *Mapping Terms Service (MTS)*: where in common domain ontology, people may store their data in different terms to represent the same concept (semantic heterogeneity). for example if we use English Transport Ontology where we defines the "Concept" (e.g., "destination"). There is the possibility when we do the negotiation process the receiver of the message don't understand this concept because it is not listed in its ontology. The correspondent concept is defined as (e.g., "arrived-City") in its private English Transport Ontology.
- *Translation Services (TS)*: here we discuss the translating ontologies in the context of Multilingual Ontology Mapping. We exemplified the negotiation between two transport systems that use two different ontologies (English and French) languages, respectively. We represent as the terms "Destination" in the source ontology is mapped to the term "Arrivée" in the target ontology. These terms represent the destination areas related to client travel.

#### 4.4 Negotiation Ontology based on Knowledge Management Systems Model (NOKMS)

Our general architecture tries to improve the work of the negotiation protocol to facilitate the communication through the agents and to solve the semantic heterogeneity by adding the Semantic Layer (SEL) and Knowledge Management Systems Layer (KMSL). Based on these changes, (Figure 4) presents the new system architecture for negotiation process which uses ontology-based knowledge management system (Saad et al., 2008c) .

We organized our architecture as follow: the first layer contains the Negotiation Layer (NL) where the SA agent sends the first message to the ICA agents to start the negotiation process.

The second layer represents the Semantic Layer (SEL); our purpose is to find a solution especially in the case of misunderstanding of the negotiation messages among the agents.

The SEL uses a translator semantic which is Translator Agent (TA) in order to help it to translate automatically the various types of exchanges between the agents. In SEL, the translator semantic examines the level of transibility among the ontologies by sending a word to the KMSL layer which resends the set of semantically equivalences words.

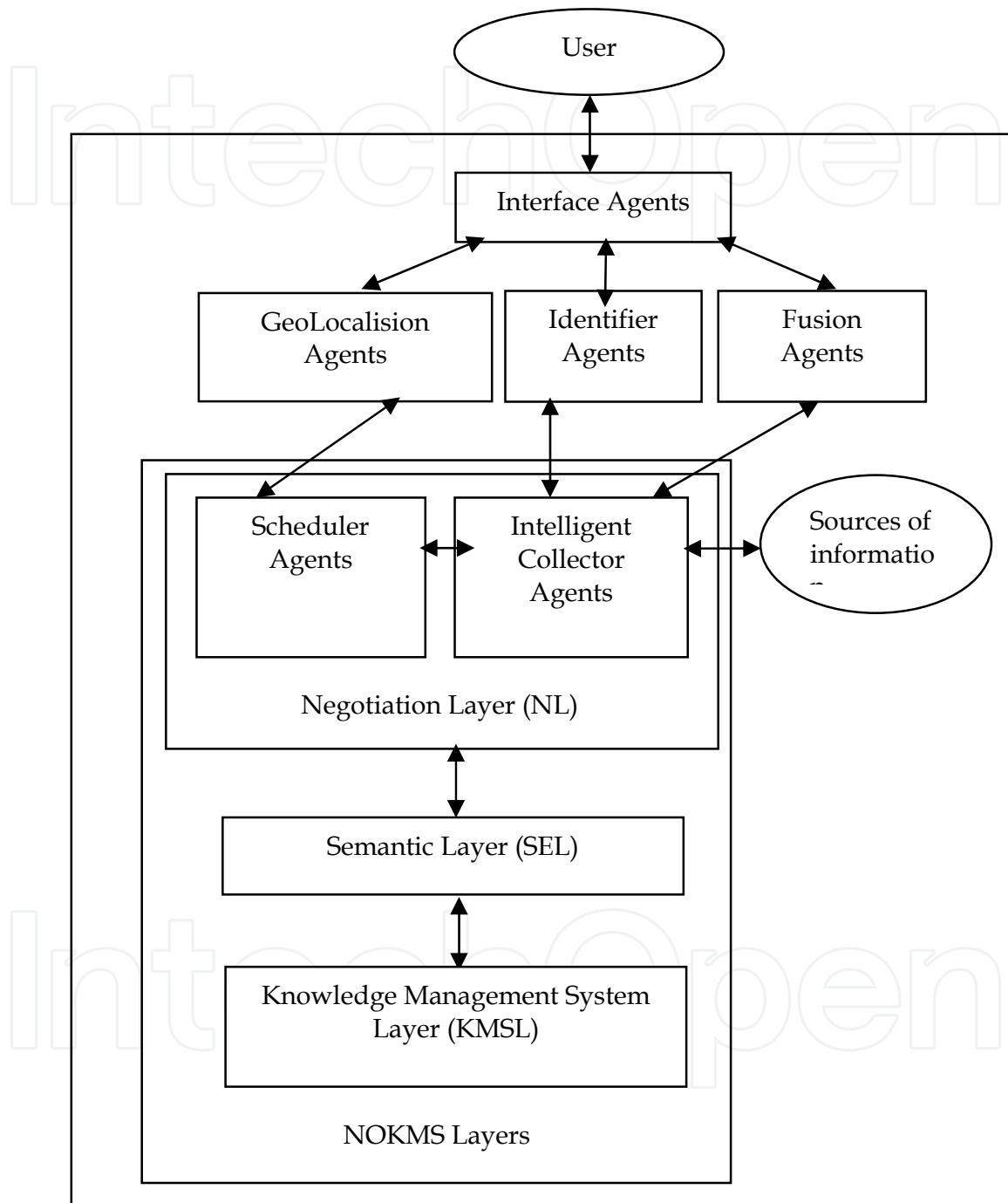


Fig. 4. Multi-agents Structure

The third one is the Knowledge Management Systems Layer (KMSL): this layer uses ontology in purpose of automatic classifying and using of the news ontologies and meta-ontologies. The architecture in this layer consists of:



- 1) *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 (Diggelen et al., 2005): a common ontology can serve as a knowledge-level specification of the ontological commitments of a set of participating agents.
- 2) *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 Agents Ontologies (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 AOs).Where ,Open Knowledge Base Connectivity (OKBC) is an application programming interface (API) for knowledge representation system (KRSs) that has been developed to address the problem of KB tools reusability. The name OKBC was chosen to be analogous to ODBC (Open Database Connectivity), as used in the database community (Geiger, 1995).
- 3) *Knowledge Acquisitions*: are a very important part in the ontology process and it applies different operations like (Knowledge Creation, Knowledge Translation, Knowledge Retrieval ), we have illustrated how we can apply those operations on the shared ontologies ( languages) in (Saad et al., 2008b).
- 4) *Intelligent Knowledge Base (IKB)*: each agent of Multi-Agent System (MAS) holds a KB which based on the domain ontology. In our IKB uses the OKBC, which in turn, connects to a wide verity of IKBs servers where these IKBs are applied the Knowledge Acquisitions.

#### 4.5 Ontology Negotiation Process

Negotiation defines as a process whose transitions and states are described by the negotiation mechanism. From the ontology point of view, this means that modelling domain factual knowledge, that is, knowledge concerning the objective realities in the domain of interest (Chandrasekaran et al., 1998). The implementation of our negotiation process combines the Ontology Negotiation Protocol (ONP) which will interact with an additional protocol called Ontology Mapping Protocol (OMP). We will explain the two protocols later. We adopt the formula of the Agent Communication Language ACL messages is as follow (FIPA0081):

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

- **Sender:** the identity of the sender of the message.
- **Receiver:** the identity of the intended recipients of the message.
- **Services:** the "yellow pages" proposed by the recipient 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...}

- **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...).
- **Protocol:** the protocol which is used to described by the negotiation mechanism

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. How can we use the message formula? We will illustrate that in the rest of paper.

#### 4.5.1 Ontology Negotiation Protocol (ONP)

The first protocol is Ontology Negotiation Protocol (ONP) represents the general scenario of agents where the SAs agents start the negotiation process by sending the messages to the ICAs agents. As we illustrated previously, we search to find the solution when there are some network errors and the agents search to find suitable new available providers for new assignment process. Here, the ICA agents participate in the negotiation by using their languages for formulating negotiation messages in order to interact and to take the decision. Our Ontology Negotiation protocol (ONP) (figure 5) is characterized by successive messages exchanges between SA agents and ICA agents. We designed our protocol so that a negotiation process can occur between several initiators and participants, it can be, for example, the case of simultaneous requests overlapping, but it is not the purpose of this paper. Here, we describe the ONP between a unique initiator and several participants. In our ONP, we allowed a partial agreement of the proposed contract from each ICA agent, to be confirmed partially or totally by the initiator of the negotiation (SA agent).

A renegotiation process is necessary while there are still tasks which need to be reassigned. The purpose of this solution is to allow the ICA agents to cooperate and coordinate their actions in order to find globally near-optimal robust schedules according to their priorities, preference, constraints and ontologies which depend on their current positions in their correspondent Workplans. Through the negotiation process tours, SA agents must assure reasonable total cost and time. We will detail the different exchanged messages between initiators and participants in next paragraph.

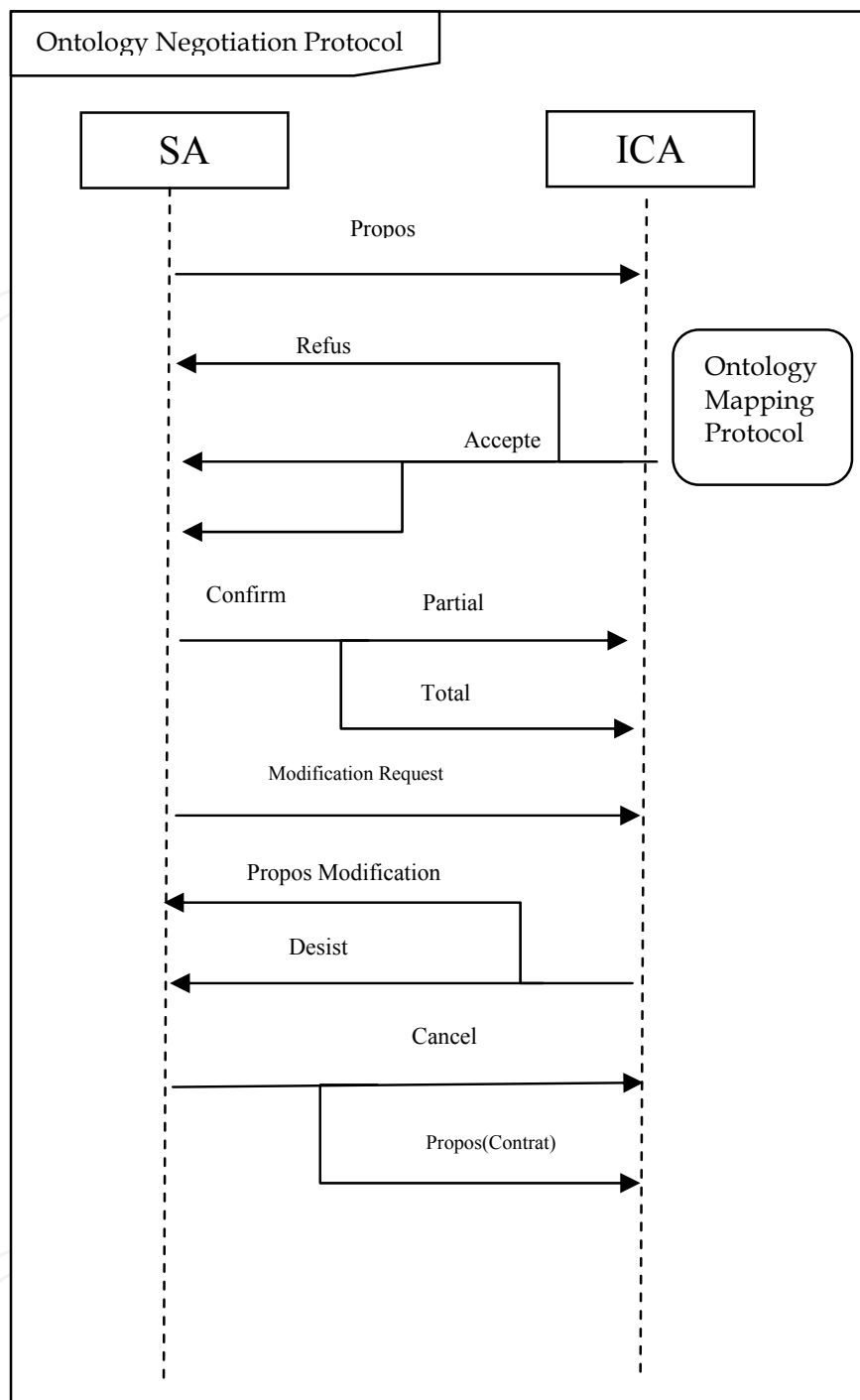


Fig. 5. ONP

#### 4.5.3 Ontology Mapping Protocol (OMP)

As we mentioned previously that another problem may take place when the participants don't understand the communication messages, or when the new agent wants to participate in a negotiation process then he has to understand the protocol and the communication language messages.

For implements the message flow which is necessary for solving the problems of interoperability, including the interaction of SA and ICA agents when requesting/receiving a service. We designed the Ontology Mapping Protocol (OMP) with the purpose of facilitating the interaction between the agents and services. (figure 6)

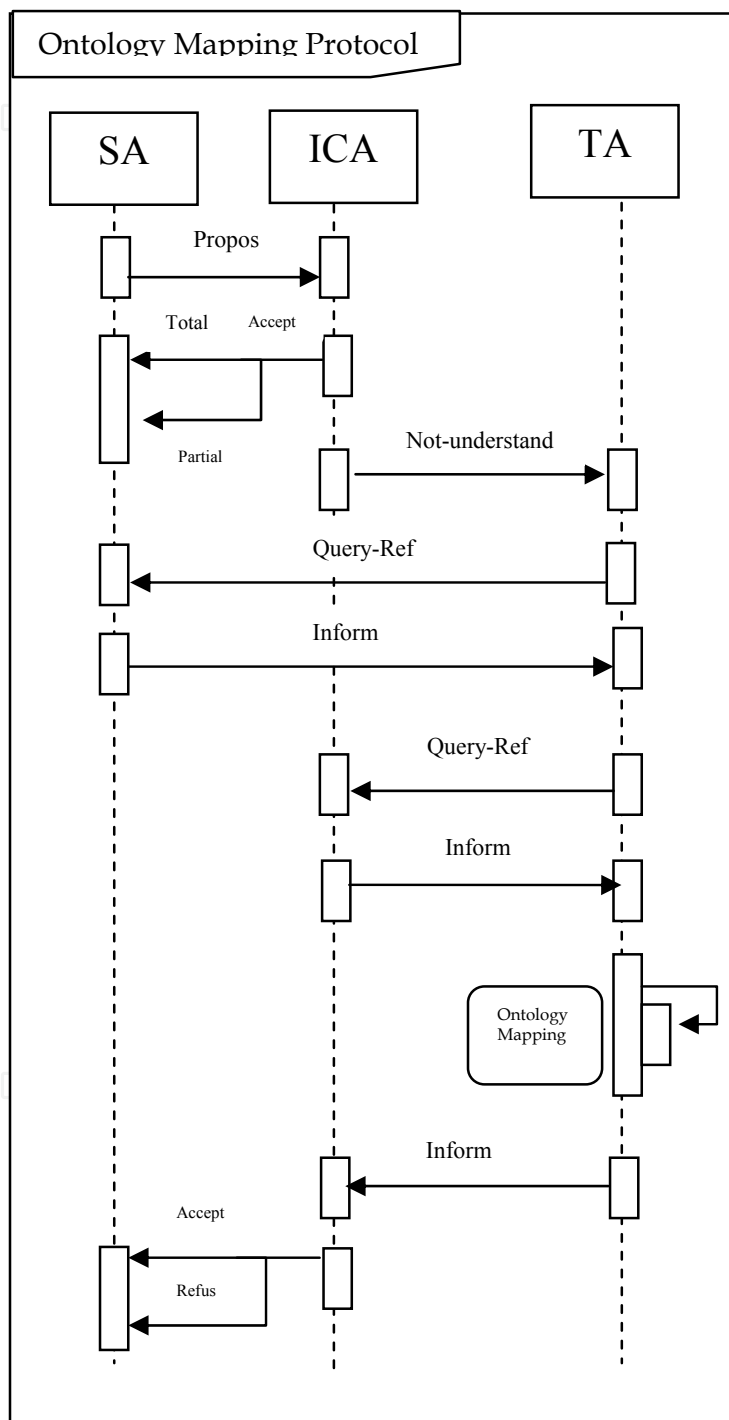


Fig. 6. OMP

After having received an ONP and not being able to interpret the requested service, the ICA sends a message with the performative NOT\_UNDERSTOOD to the TA ask him who sent

the ONP and the name of the unknown service. The TA sends the name of the service which it has just received to the SA in order to get further information about it. The SA will analyze that request and send back attributes of the concept, i.e. all the information about this service.

After having received the answer from SA, the TA knows the description, of the demanded service under negotiation and sends it to the ICA. The later selects among all service the ones whose time value is near of the received value. After the selection, the ICA answers with a list containing names of potential correspondent concepts.

After receiving all the information about the service under negotiation and a list of possible corresponding services, the TA is able to apply methods in order to match the services. In the previous work (Saad et al., 2008a); we have applied the Quick Ontology Mapping (QOM) method where this method aims to detecting semantic similarity of terms. Every term of the proposed, potential correspondent service is compared to the requested term. By using QOM method, we apply the first task of our OMP which is the Mapping Terms Service (MTS). For the second service which is Translation Services (TS), it is not in the domain of this paper.

In final step, the TA informs the ICA about the result of the comparisons delivered from the ontology mapping methods. The ICA is then able to respond to the SA, either with a ACCEPT or with a REFUS that is part of our ONP.

#### 4.5 The Agent Messages

As we have seen in the previous section, we proposed a structure for our ONP and OMP protocols. In what follows, we detail the different exchanged messages between initiator and participants.

##### 4.5.1 Proposition of the contract:

The contract message is a proposition of a new organization (the first contract) or reorganization of final Workplans to achieve tasks. If the execution of some services was cancelled because of some network perturbations, it is indeed the case of reorganization. This will be done by reassigning one more time servers to these tasks which represent the set of the Dynamic Reassigned Tasks (DRT) (Saad et al., 2008a). The initiator sends an individual contract to each active  $ICA_k$  agent who proposes the contract-reception service:

$$\langle SA_i, ICA_k, contract-reception, propose, \partial, fipa-sl, Ontology, protocol \rangle$$

With  $\partial = \partial_1$  if it acts of the first contract and  $\partial = \partial_2$  otherwise:

$\partial_1 \equiv \text{Workplan} ($ Owner : $ICA_k$ Initial : $i_1, \dots, i_{k_i}$ Final : $f_1, \dots, f_{k_i}$ )	$\partial_2 \equiv \text{FinalWk} ($ Owner : $ICA_k$ Final : $f_1, \dots, f_{k_f}$ )
--	--

With  $i_1, \dots, i_{k_i}$  represent references of nodes which belong to the initial Workplan of the ICA agent k ( $ICA_k$ ) and  $f_1, \dots, f_{k_f}$  represent references of nodes which belong to the final Workplan of the same agent. Thus we have  $k_i \leq k_f$ .

#### 4.5.2 Response to the contract:

When a participant receives the proposed contract, he studies it and answers by:

- **Total Acceptance** : if he agrees to coordinate all tasks chosen by the initiator, included in his remaining trip (remained final Workplan), according to his current position,

$\langle ICA_k, SA_i, \emptyset, \text{accept-proposal}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$

- **Partial Acceptance**: if he agrees to coordinate a subset of the tasks selected by the initiator, included in his remaining trip (remained final Workplan) or if he doesn't understand the received message sending by the initiator. Then, according to his current position, the partial-accept-proposal message content expresses the references of cancelled tasks and those of unavailable servers (the reason of the non total-acceptance):

$\langle ICA_k, SA_i, \emptyset, \text{partial-accept-proposal}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$

With  $\partial \equiv (\text{tasks: } t_1, \dots, t_n \text{ nodes: } S_1, \dots, S_m)$

- **Refusal**: if he does not agree with any task in the proposed contract (i.e. he uses the ONP for check the services only) or if he doesn't understand the received message sending by the initiator (i.e. he didn't understand the message, here he uses OMP to analyze the message). Then, the refusal message content expresses the references of unavailable servers (the reason of the refusal):

$\langle ICA_k, SA_i, \emptyset, \text{refuse}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$

With  $\partial \equiv (r_1, \dots, r_m)$

The initiator does not wait for all answers because he must act rapidly, so he just waits for some answers for a very short period of time to make a decision.

#### 4.5.3 Confirmation

An initiator has to confirm independently the agreed part of each contract  $k$  proposed to an agent  $ICA_k$  who represents an autonomous participant of the negotiation, the confirmation can be:

- **Total**: if the initiator agrees with the total response to the previous proposed contract ,

$\langle ICA_k, SA_i, \emptyset, \text{confirm}, \emptyset, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$

- **Partial**: if the initiator agrees with a partial response to the previous proposed contract, the partial-confirm-proposal message content expresses the references of agreed tasks:

$\langle ICA_k, SA_i, \emptyset, \text{partial-confirm-proposal}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$

With  $\partial \equiv (g_1, \dots, g_p)$



#### 4.5.4 Modification request

If the DRT table is not yet empty (Saad et al., 2008a); the initiator asks the participants to propose a new distribution of services assignments which are canceled, the request-modification message content expresses the DRT table:

$$\langle SA_i, ICA_k, \emptyset, \text{request-modification}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

With  $\partial \equiv (\text{DRT})$

#### 4.5.5 Modification proposition

According to our DRT algorithm, where we design a reassignment procedure strategy of servers to tasks, taking into account not only the dynamic positions of ICA agents in their Workplans, but also their constraints, priorities, preferences and ontologies, according to their respective current positions. The proposition message content expresses for each participant  $k$  the new proposition of his remained Workplan according to his current state:

$$\langle ICA_k, SA_i, \emptyset, \text{propose}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

With  $\partial \equiv \text{FinalWk} (\text{Owner: } ICA_k, \text{Final: } f_1, \dots, f_{k_f})$

Where  $f_1, \dots, f_{k_f}$  represent references of nodes which belong to the final Workplan of the agent ICA $_k$ .

#### 3.5.6 Desist

After have sending the conformation. The participants (or the initiator) don't want to continue the negotiation process. Then, he decides to desist the process. In this case, if the DRT table is not empty, the initiator can resend another contract to the participants. the desist message content is as follow:

$$\langle SA_i, ICA_k, \emptyset, \text{desist}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

With  $\partial \equiv (\text{DRT})$

#### 3.5.7 Not Understand

In our system the problem of heterogeneity may arise; when one of ICA $_k$  agents receives the message and it don't understand the concepts. Then ICA Agent will send a message to the TA, setting the performative of the ACL message to NOT UNDERSTOOD. The TA is placed in the Semantic Layer of our system (SEL) (Saad, 2008c).

The TA Agent will examine the level of transibility between the ontologies correspondent. by applying the ontology mapping method. For this proposal TA access to the services provided by the KMSL (OntoSV), which are in this case helping in the existing heterogeneity problem, trying to map concepts of ontologies and thus looking for similarities. In order to facilitate the negotiation process (i.e, reduce the number of negotiation rules), the not understood message will to be, as follow:

$$\langle ICA_k, SA_i, \emptyset, \text{not understood}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$$

With  $\partial = (c_1, \dots, c_n)$

### 3.5.8 Cancel

To avoid indefinite waiting for answers or for modifications, the initiator agent must make a decision at the end of a fixed period of time, illustrated by the last field of an agent message. Therefore he cancels the contract if there is no more solution (lack of resources, no available provider...) or he creates new ICA agents to execute the current contract:

$\langle SA_i, ICA_k, \emptyset, \text{cancel}, \partial, \text{fipa-sl}, \text{ontology}, \text{protocol} \rangle$

## 5. Case Study

As we mentioned in the previous sections, one of the big problems to communication-based agents is that each one uses different terms with the same meaning or the same term for different meanings. Once we took this problem as a challenge, representing these differences in a common ontology becomes essential. Indeed, the use of a common ontology guarantees the consistency (an expression has the same meaning for all the agents) and the compatibility (a concept is designed, for the same expression, for any agent) of the information present in the system. However, it is not sure that all the agents will use a common ontology. Usually, each agent has its heterogeneous private ontology and it cannot fully understand other agent's ontology. Problems with heterogeneity of the data are already well known within the distributed database systems community. If common domain ontology is used, it seems easier to know that people are speaking about the same subject. However, even with a common domain ontology, people may use different terms to represent the same item, the representation can be either more general, or more specific and with more details.

In our work, to market its data, an information provider must solicit the system in order to register or update the services that it offers. A service is characterized by a cost, a response time and a data size. A service is also characterized by a time relevance that allows saving information locally for a certain time to reduce the transmission of data if that is possible. For that in the previous work (Zgaya, 2007a), we have developed two databases where the first is used to register the servers which want to propose their services through our system, and the second database plays the role of "buffer zone" contain static data to a certain degree, (Figure 7)

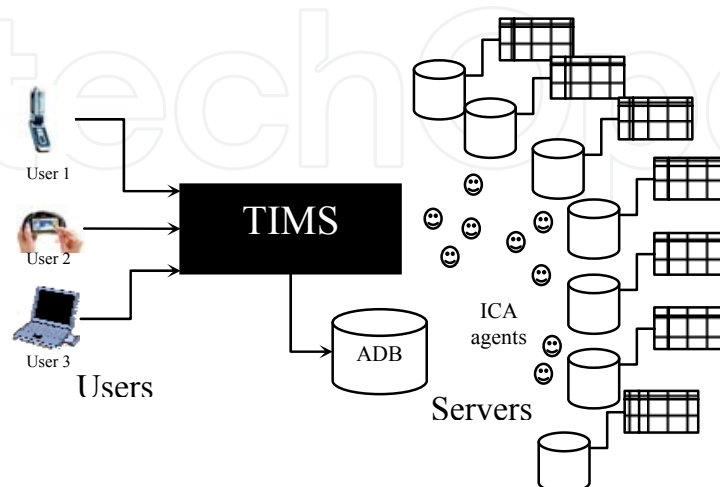


Fig. 7. Dynamic Information Archiving

We illustrated the first databases which use to register the providers of the services where each provider, wanting to offer its services through our system, must register all its services in this database. Previously, we have used the reference as the index for the services. Here, a supplier must register the label of each service proposed, its reference, the estimated response time, cost and size of data corresponding. It must also mention the address of his or its servers. The same service (same label) may be proposed by several suppliers with costs, response times and different sizes; for example when a provider  $S_{11}$  register its service ( $T_2$ ) with the  $t=0,25$ second and cost= 5 point. There is the possibility that the providers  $S_5$  and  $S_{20}$  have the same service where  $S_5$  register it as ( $T_2$ ) with the  $t=0, 15$  second and cost=5 point in the register database. May the server  $S_{20}$  register the service with the label ( $T_2'$ ) with the  $t=0, 20$  second and cost=4point. In this case, those providers use different terms with the same meaning. In this example, the simultaneous requests managed by the different IA agents are decomposed into a set of independent services which was sent to IdA agent. Thus, when the user searches service  $T_2$ , the system will create the initial Workplans which contains the initial assignment solution of servers to tasks where  $S_1, \dots, S_{20}$  represent available servers containers on the network. Then, the final assignment solution of servers to tasks is deduced from initial Workplans generation and our genetic algorithm results, in our case  $S_5$  will be in the final Workplans. The ICA agents can move in order to collect date according to the adopted contract model. Here, the move of an ICA1 agent into a server ( $S_5$ ) on the network knowing that in JADE platform, containers must be created on machines to receive agents. The DRT algorithm is implemented in the context of a negotiation process between agents SA and ICA in order to negotiate dynamically best assignments of servers to tasks according to the new set of unavailable machines. I.e. when a server ( $S_5$ ) is not available the SA begin the negotiation process where it proposes the new contract to ICA1 agent and this contract will contain the servers ( $S_{11}$  and  $S_{20}$ ) whose propos the same service. In what follow, we present an example which show the execution of this contract where ICA1 agent received a proposition of the contract from SA agent. The propose message is, as follow:

```
(Propose
:sender (agent-identifier
      : name SA@home:1099/JADE
      : addresses(sequence http://home:7778/acc))
:receiver (set
  ( agent-identifier
    : name ICA1@home:1099/JADE
    : addresses(sequence http://home:7778/acc)))
:content "((OWNS (agent-identifier
  : name
    ICA1@home:1099/JADE
  : addresses (sequence http://home:7778/acc))
(services
  : servers (sequence
    http://home:7778/acc
    http://home:2588/acc
    http://home:2590/acc
    http://home:2592/acc
    http://home:2594/acc)
```

```
: duration 120)))"  
: language fipa-sl  
: ontology English-Transport-ontolog  
: protocol Ontology-Negotiation-Protocol)
```

For  $S_{20}$  the answer will be not understand because he don't understand the message sends from SA agent although he has the same service which the user need. Indeed, problems of heterogeneity of the data are appearing here where server  $S_{20}$  has the service ( $T_2'$ ). So, the answer will be with the message not understood. For that our DRT algorithm will use the QOM algorithm to solve this problem and to do the mapping between ontologies sure according to ontologies, constraints, priorities and preferences of the ICA agents in their final Workplans.

## 6. Conclusion and Future Work

In this chapter, we proposed an optimizing approach of the data flow management, in order to satisfy, in a better manner, customers' requests. The adopted approach decreases considerably computing time because Workplans are just deduced; they are computed when network traffic varies considerably. We have presented a new solution for the problem of language interoperability between negotiation agents, by incorporating architecture for Negotiation process with that uses an Ontology-based Knowledge Management System (NOKMS). The proposed solution prevents the misunderstanding during the negotiation process through the agents' communications. The architecture consists of three layers: (NL, SEL and KMSL). But in this work we talked about the first layer only (NL) that describes the negotiation process as well as illustrates the different messages types by using the different ontologies. Our proposed NOKMS improves the communications between heterogeneous negotiation mobile agents and the QoS in order to satisfy the transport customers. Indeed, the ICA agents can to ignore crashed nodes in their remained routes, so they have to avoid visiting them. This will be done by (DRT) algorithm for reassigning substitute servers tasks which need to be reassigned. This reassignment depends on the actual positions of ICA agents in their final Workplans. It depends also on their ontologies, constraints, priorities and preferences. The new assignment constitutes a contract between ICA agents and SA agents.

In a future work, we will try to apply our approach to contain the different systems which can negotiate at the same time and each of these systems has their ontologies (languages) and can offer different services. This can take place when ICAs know their final Workplans. The agents ICAs are supposed to visit their first nodes by the order as in their Workplans 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) by using our DRT algorithm. But when it rest another tasks in DRT table and there is not available nodes in the same system then IS agent sends a new propose contract to a meta-system which in turn searches the suitable system to continuous the negotiation process. According to this new renegotiation process, it must to improve the DRT algorithm to adopt the novel ontology in the new system.

For the simulation part, we will create all our ontology structures by using Protégé which is an open-source development environment for ontologies and knowledge-based systems.

Protégé contains a large number of plug-ins that enabled the user to extend the editor's core functionality like the Bean Generator plug-in (JADE, 2002) 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 (FIPA0008). As we had decided to use the JADE multi-agent environment (JADE site) for implementation of MTIS project (Saad et al., 2008c). 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 multiple computers with little complication.

## 7. References

- Abou Assali, A ; Lenne,D and Debray,B (2007): KoMIS: An Ontology-Based Knowledge Management System For Industrial Safety. (DEXA'2007). Regensburg, Germany
- Bailin ,S; Truszkowski. (2002). Ontology negotiation between intelligent information agents. *The Knowledge Engineering Review*,
- Bravo, M.C; Perez, J.;Sosa,V.J; Montes, A; Reyes, G (2005): Ontology support for communicating agents in negotiation processes. *Hybrid Intelligent Systems*, 6-9 November.
- Carey, M and Johnson, D,( 1979) "Computers and Intractability: A Guide to the Theory of NP-Completeness", Freeman,.
- Carzaniga, A; Picco, G.P and Vigna, G,( 1997)"Designing distributed applications with mobile code paradigms", in Proc. of the 19th International Conference on Software Engineering (ICSE'97), Massachusetts, USA.
- Davies, J; Studer, R; Warren ,P; (2006):Semantic Web Technologies: Trends and Research in Ontology-based Systems, April
- Diggelen, J. Van; Beun, R.J.; Dignum, F.P.M.; Eijk, R.M. Van; Meyer, J-J.Ch. 2007. Ontology Negotiation in Heterogeneous Multi-agent Systems: the ANEMONE System ,IOS
- Diggelen, J. Van ; Beun, R.J ; Dignum, F.P.M. ; Eijk, R.M. Van; Meyer, J-J.Ch(2004). Optimal Communication Vocabularies and Heterogeneous Ontologies , IOS Press 2005
- Ehrig, M; Staab, S.(2004)
- Efficiency of Ontology Mapping Approaches.
- Falasconi, S; Lanzola, G; and Stefanelli, M. (1996) .Using ontologies in multiagent systems. In *Proceedings of Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW)*, Banff, Alberta, Canada,
- Geiger, K, (1995):  
Inside ODBC. Microsoft Press..
- Green, S; Hurst, L; Nangle, B; Cunningham, P; Somers, F and Evans, R. (1997). "Software agents: A review", Technical report, TCS-CS-1997-06, Trinity College Dublin, Ireland
- Gruber, Th.R. (1993): A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition Systems*.
- FIPA0081: FIPA ACL Message Structure Specification. <http://www.fipa.org/specs/fipa00061/index.html> .
- FIPA0008: FIPA SL Content Language Specification <http://www.fipa.org/specs/fipa00008/index.htm>



Java Agent DEvelopment framework.

<http://jade.titlab.com/doc>

- Jennings, N. R; Faratin ,P; A.R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated haggling, (2000): Building artificial negotiators. In Pacific Rim International Conference on Artificial Intelligenc
- Klein, M. (2001). Combining and relating ontologies: an analysis of problems and solutions. In IJCAI-2001Workshop on Ontologies and Information Sharing,pages 53–62, Seattle, WA.
- Lander, S and Lesser, V; (1993). Understanding the role of negotiation in distributed search among heterogeneous agents. In Proceedings of the International Joint Conference on Artificial Intelligence
- Maedche, A; Motik, B. (2003). Ontologies for Enterprise Knowledge Management “,IEEE Intelligent Systems,,26-33.
- Malucelli, A., Oliveira, E.( 2004). Ontology-Services Agent to Help in the Structural and Semantic Heterogeneity, In: Camarinha-Matos, L. (eds.) Virtual Enterprises and Collaborative Networks, Kluwer Academic Publishers, pp. 175-182,
- Obitko, M; Marík ,V.( 2004) OWL Ontology Agent based on FIPA proposal, Znalosti ;2004, Brno, Czech Republic,
- Saad, S; Zgaya, H and Hammadi, S (2008a), Dynamic Reassigned Tasks during the Negotiation Process by Ontology Approach between Mobile Agents, IEEE, International Conference on Intelligent Agent Technology (IAT-08).Sydney, Australia.
- Saad, S; Zgaya, H and Hammadi, S (2008b), Using Ontology to Solve the Negotiation Problems in Mobile Agent Information Systems. SMC, Singapore
- Saad, S; Zgaya, H and Hammadi, S (2008c). The Flexible Negotiation Ontology-based Knowledge Management System: The Transport Ontology Case Study, In proceedings of the IEEE, The International Conference on Information & Communication Technologies: from Theory to Applications - ICTTA'08 - April 7 - 11, 2008 in Damascus, Syria
- Sh.Ji , x; Qijia,T ; Liang,T, ;Yang,H;(2007): An Ontology Framework for EC Automated Negotiation Protocol. Networking, and Parallel/ Distributed Computing, 2007. SNPD 2007. July 30 2007-Aug.
- Studer, R; Benjamins, V and Fensel, D. (1998) : Knowledge engineering, principles and methods. Data and Knowledge Engineering, 25(1-2):161–197,
- Tamma ,V; and T.J.M. (2002a): An ontology model to facilitate knowledge sharing in multi-agent systems. In Knowledge Engineering Review. Bench-Capon,
- Tamma, V; Wooldridge, M; Blacoe, I; and Dickinson, I.( 2002b).An ontology based approach to automated negotiation. In Proceedings of the IV Workshop on Agent Mediated Electronic Commerce,
- University of Amsterdam, Ontology Bean Generator for JADE, (2002)  
<http://www.swi.psy.uva.nl/usr/aart/beangenerator/>,
- Verrons ,M. H; GeNCA. ( 2004) : un modèle général de négociation de contrats entre agents. PHD, France,
- Zgaya, H (2007a): Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine : Une approche multi-agent pour la recherche et la composition des services liés au transport PHD, EC-Lille, France



- Zgaya, H ; Hammadi, S and Ghédira, K , (2007b), Combination of mobile agent and evolutionary algorithm to optimize the client transport services, RAIRO.
- Zgaya, H and Hammadi, S. (2007c), Multi-Agent Information System Using Mobile Agent Negotiation Based on a Flexible Transport Ontology. (AAMAS'2007), Honolulu, Hawai'i.
- Zgaya, H ; Hammadi, S and Ghédira, K.(2005a) "Workplan Mobile Agent for the Transport Network Application", IMACS'2005, Paris,
- Zgaya, H ; Hammadi, S and Ghédira, K.(2005b) "Evolutionary method to optimize Workplan mobile agent for the transport network application", IEEE SMC'2005, Hawaii, USA
- Zhang, X; Lesser ,V; and Podorozhny,R (2005) Multi-dimensional, multistep negotiation for task allocation in a cooperative system. Autonomous Agents and Multi-Agent

IntechOpen



## **Knowledge Management**

Edited by Pasi Virtanen and Nina Helander

ISBN 978-953-7619-94-7

Hard cover, 272 pages

**Publisher** InTech

**Published online** 01, March, 2010

**Published in print edition** March, 2010

This book is a compilation of writings handpicked in esteemed scientific conferences that present the variety of ways to approach this multifaceted phenomenon. In this book, knowledge management is seen as an integral part of information and communications technology (ICT). The topic is first approached from the more general perspective, starting with discussing knowledge management's role as a medium towards increasing productivity in organizations. In the starting chapters of the book, the duality between technology and humans is also taken into account. In the following chapters, one may see the essence and multifaceted nature of knowledge management through branch-specific observations and studies. Towards the end of the book the ontological side of knowledge management is illuminated. The book ends with two special applications of knowledge management.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Sawsan Saad, Hayfa Zgaya and Slim Hammadi (2010). Towards the Optimization of Client Transport Services: Negotiating by Ontology Mapping Approach between Mobile Agents, Knowledge Management, Pasi Virtanen and Nina Helander (Ed.), ISBN: 978-953-7619-94-7, InTech, Available from:

<http://www.intechopen.com/books/knowledge-management/towards-the-optimization-of-client-transport-services-negotiating-by-ontology-mapping-approach-betwe>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen