



HAL
open science

Adaptive organization for cooperative systems

Nadia Abchiche-Mimouni

► **To cite this version:**

Nadia Abchiche-Mimouni. Adaptive organization for cooperative systems. 16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2012), Sep 2012, San Sebastian, Spain. pp.128-138, 10.3233/978-1-61499-105-2-128 . hal-00763641

HAL Id: hal-00763641

<https://hal.science/hal-00763641v1>

Submitted on 18 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive organization for cooperative systems

Nadia Abchiche-Mimouni

IBISC E.A. University of Evry

523 Place des Terrasses 91000 Evry, France

Tel: 00 (0) 33 1 60 87 39 10

Fax: 00 (0) 33 1 60 87 37 89

nadia.abchiche@ibisc.univ-evry.fr

Abstract

This paper presents an original approach for dynamic organization in multi-agent systems to deal with the problem of integrating several heterogeneous reasoning methods (RMs) in a single system. Our approach is based on two observations: (1) several RMs may be usable for solving the same problem, and (2) there is no deterministic way to find the most adequate (RM) no a way allowing to combine the RMs. Some heuristics can guide the problem-solving process in combining the RMs and if necessary, switching from one RM to another. These heuristics depend on: the context and the description of the problem itself, and on the constraints of the desired solution. The real problem is that the context depends on dynamic and unpredictable knowledge. An adaptive approach is implemented in a multi-agent system (MAS) to build cooperative scenarios in a dynamic way. In such a system, each agent reifies a particular RM. The ways to combine the RMs are decided in a decentralized way according to cooperative knowledge embedded in the agents. The organization of the agents around a pivotal agent role helps the scenarios to finish and to propose one solution to the user.

Categories and Subject Descriptors

I.2.11 [Computing Methodologies]: Distributed Artificial Intelligence, Multiagent systems, Coherence and coordination.

General Terms

Algorithms.

Keywords

Distributed problem solving, Agent Reasoning, Collective decision making, Agent Cooperation.

1. INTRODUCTION

The integration of reasoning models (RMs) which is addressed in this work, is in the scope of cooperative problem-solving ([5]) because it required integration various ways of solving and various kinds of knowledge to solve a whole problem. Many integrating approaches exist. But the means they use are ad-hoc. This make them difficult to use for solve new problems. Thus, they are difficult to extend with new RMs ([2], [3], [6], [10]). However, these woeks offer a whole expertise on the contexts which require of such integration. The advantages of such approaches are as well at the methodological level as at the computational level because of the difficulty of finding an algorithm effective able to choose an adequate RM according to specific situation.

In this paper, an original approach is proposed to deal with the integrating reasoning models problem. For this purpose, multi-agent systems paradigms are used as a mean to design a distributed approach in an adaptive way.

The next section gives an overview of the problem of the reasoning models integration. The followed methodology is explained and illustrated with a diagnosis example. The multi-agent approach will be detailed in the third section. Two versions of the system for dynamic organization are presentedand

compared. First evaluations show that negotiation mechanisms improve the cooperative scenario. Before the conclusion, some related works are mentioned.

2. PROBLEM DESCRIPTION

The reasoning models integration problem is complex. It must be addressed by mean of a methodology then it will be illustrated on a case-study.

2.1 Reasoning models integration

A reasoning model (RM) includes a way of reasoning such as Modus Ponens, analogy or qualitative reasoning which can be used to make some deductions from any corpus knowledge representation in a particular application domain. This concept is similar to the reasoning mode described in [12] where knowledge representation is clearly distinguished from the way we intend to use the same knowledge. Each RM can be described by a set of characteristics which makes it strongly relevant in certain contexts, simply useful in the others, or sometimes unusable. For example, a qualitative reasoning is a RM for which it is enough to know the membership of system parameters in an interval to be able to predict the behavior of the (modeled) system ([4]). The drawback of this way of reasoning is that it is necessary to have information about all the system parameters to obtain an explanation of its behavior. Another example concerns the use of the induction process to infer a new solution from past encountered and solved problems. This implies having a way to measure the similarity between stored cases and the current problem description, which the system has to solve.

The integration of RMs consists in implementing several RMs in the same system so that there is a synergy between them for the resolution of a problem that the RMs are not capable to solve individually, at least, with acceptable performances. For example, a range for a missing parameter can be induced from past situations so that a qualitative reasoning becomes usable.

The integration of RMs is based on the fact that to make reasoning with the aim of the resolution of a complex problem, it is often necessary to call upon more than one type of reasoning. The problem-solving process of such a problem is then seen as an interlacing of several known types of reasoning. For example, during a deductive reasoning to do a mathematical theorem proving, we often have to use examples or counter-examples to make certain steps of the demonstration. So, a “reductio ad absurdum” will be used to demonstrate a part of the theorem.

In intelligent systems, the use of several kinds of knowledge (heuristics, qualitative, behavioral, etc.) and of various modes of exploitation of these knowledge in the same system is more and more spread ([1], [5], [9]). Such systems propose algorithms or means to combine several methods of resolution of problems.

On the other side, it's well admitted, in Artificial Intelligence, that the diversity of the problem solving methods are due to the limitation of each of them. In fact, the same process can be solved using different RMs and there is no best RM, but just (perhaps) a preferred one to exhibit a particular aspect (or a part) of a problem to solve.

Some experiences in cognitive science have showed that human reasoning process is neither done in a monolithic way nor uses a single way of reasoning for solving problems [7].

In the he next section, we will describes the originality of our approach and we will explain the applied methodology to achieve our goal. Some arguments for a new integrating approach will be also presented.

2.2 Methodology

To deal with the problem of RMs integration, one argue for a distributed and a dynamic approach. Indeed, existing approaches are either based on an

ad-hoc algorithm or on an exhaustive exploration of the different integration possibilities. In the first case, we have systems in which adding a new RM is too difficult (requires to modify the design of the system itself). In the second case, there is often a risk of a combinatorial explosion of computation due to the algorithmic complexity. These weaknesses come mostly from:

- (1) the tendency which consists in palliating disadvantages of the RMs by abolishing them and by replacing them with ad-hoc means and,
- (2) the difficulty in determining, in priori, the way to adapt the integration mechanism to every new situation to solve.

That being said, existing approaches are a precious inspiration source, since they allow us to work out corpus of knowledge constituting whole expertise on the integration problem itself and to design generic integrating heuristics. The weaknesses of the different RMs are expressly represented, within the RMs, as being part of knowledge the RMs have on themselves. These knowledge are exploited by the RMs in order to:

- (1) Initiate collaboration with other RMs.
- (2) Distinguish the situations where it is necessary to improve their own processing of the situations where they can go without this improvement.
- (3) Spontaneously offer help to other RMs.

Thereby, the RMs collaborate intentionally by choosing the moment and the strategy, not only according to their own capabilities and the evolution of the problem-solving process, but also according to other RMs capabilities so that they can interact with them.

RMs need to have a representation of skills of other RMs and to know their availability for taking into account requests. The solution of the problem is constructed thanks to the interactions between the different RMs.

2.3 A case-study

To illustrate the integration of several RM and the followed methodology, a simple application in a troubleshooting domain is considered: an example based on a diagnosis of faults in an electrical car circuit; so, the problem to solve is: *finding an explanation for a malfunction of the considered electrical circuit*.

Let's consider three RMs that are representative of those used in the application domain. A case-based reasoning (CBR), a model-based reasoning (MBR) and a failure mode and effects analysis (FMEA). A CBR is a RM which solves new problems by using or adapting solutions that were used to solve past problems. It manipulates a base of cases which is accessed with indexes and similarity methods. The cases representing descriptions of old solved problems witch can of various kinds. Such a RM can give rapid results if relevant indexes are available. But, because it's based on experience, it lacks of reliability. A MBR refers to a RM which is based on a model of the physical world. With this approach, the main focus of the application to develop is designing a correct model. Then at run time, an "engine" combines this model knowledge with observed data to derive conclusions such as a diagnosis or a prediction by looking at the symptoms to then determine the possible (hypothesis) causes. Such RM reflect physical laws rather than observed coincidences, that might only be true under certain conditions. So, they are supposed to have a high level of reliability but the processing can be very expensive in time.

A FMEA is a procedure for analysis of potential failure modes within a system for the classification by severity or determination of the failure's effect upon the system.

Several heuristics can illustrate the synergy between the three mentioned RMs (CBR, MBR and FMEA). For example, a FMEA can be used to reduce the combinatory explosion of hypothesis generated by a MBR, by considering the most critical ones. A CBR can be used to infer a solution for a part of the problem which has not been modeled in the MBR. This is because the cases represented in a CBR are

various. Because of its high level of reliability, a MBR can be used to confirm (or infirm) a particular result for a CBR and so on.

From these examples, one can understand the interest of combining several RMs in a single system. It's easy to argue for a distributed methodology so that adding a new RM could be possible without modifying the system design. This is why we the following questions are asked:

1. Is it possible to provide each RM by means to determine its own relevance dynamically depending on context, which itself is changing dynamically?
2. Why not make explicit the characteristics of the RMs locally?
3. At a given moment, what would do RM when its methods are not relevant?
4. Is it possible to provide RM by means, allowing them to overcome these shortcomings?
5. At what level of abstraction is it necessary to represent knowledge mentioned above?

The next section describes a system based on a multi-agent approach where each RM is represented by an agent. Cooperation strategies allow the agents to reason about the RMs' capabilities. By this way, cooperation reasoning allows finding a compromise between strength and weaknesses of the different RMs.

3. MULTI-AGENT APPROACH

The designed approach consists in using an adaptive multi-agent system (MAS) to build in a dynamic and a distributed way, the manner of combining several RMs for solving a particular problem ([8], [13]). More precisely, it is to design a MAS. in which RMs cooperate and interact to exchange tasks and results. In this approach, RMs intertwine according to their relevance. The result is a scenario (cf. 3.2.) consisting of the interactions that lead to the solution. These interactions are due to an individual reasoning process about the way to cooperate. The principal question the agents ask is: *do they use their own methods to solve their own tasks or do they ask other agents to do it, or do they use their methods to handle the requests of other agents?*

The MAS will thus be able to solve a problem which, a priori, is outside the capabilities of each individual agent.

3.1 System description

Each agent representing a RM has a generic architecture to manipulate the domain knowledge (diagnosis) and cooperation knowledge which are used for choosing the most adequate cooperative attitude at each execution cycle.

Domain and cooperation knowledge are decomposed according to three axis (see FIG.1): tasks (the goals to achieve), methods (the different ways to achieve the goals) and domain model (needed knowledge to execute the methods). Criteria and characteristics are assigned to the methods in order to reflect their performances (reliability, complexity, efficiency, completeness, timeliness, etc.).

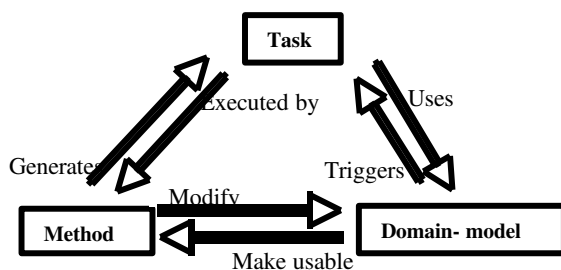


FIG.1: Task, methods and domain-model

The methods can generate sub-tasks if they are not terminal. The above figure (FIG. 2) shows some examples.

	Diagnosis	Cooperation
Tasks	generate-Hypothesis, discriminate-Hypothesis	allocate-Task, delagate-Task, solve-Conflict
Methods	perform-index, Depth-first	negociate, constraint-Ralex
Domain model	symptoms, base of cases	competencies, evolving-Resolution

FIG. 2: Examples of tasks decomposition

Agent architecture summarizes the components of each agent (see FIG.3).

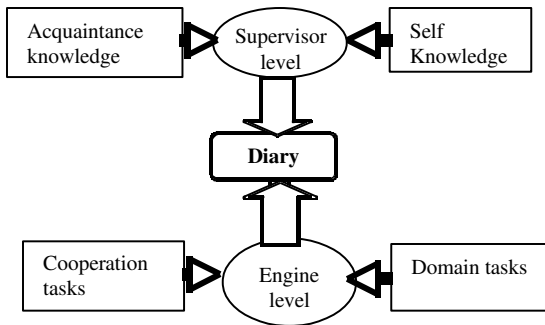


FIG.3: Internal agent architecture

The supervisor observes, evaluates and guides the local activity of the agent. Its main objective is to manage the tasks candidates of the diary, either by insertions, or by suppressions, and to fix criteria on these tasks so that the engine can choose a method to carry them out. It also establishes a cooperation attitude that the agent adopt with respect to the other agents, concerning the execution of the tasks. This can guide the agent in answering to the asked question in the previous section. The engine is in charge of performing the planned tasks by choosing an adequate domain or cooperation method.

3.2 Cooperative scenarios

The cooperative reasoning for the integration of three particular RM is illustrated by the figure (FIG.4). In this example, three agents: a case-based model, a model-based diagnosis and a model based on FMEA cooperate in interaction with the user, try to find the cause of misbehavior of a wind-screen washer. They build a cooperative scenario witch will be assumed to be a way of combining the three mentioned RMs. In this figure, the focus is on the content of agents diary and the interactions between the agents. The internal individual reasoning agents will be more addressed in next sections.

The arrows represent the exchanges of messages between agents. The agents' internal decisions are due to the execution of the engine algorithm. These decisions are indicated in the squares whose corners are rounded. The mailboxes are represented by squares at the top of the agents.

The interactions between agents embody some points of synergy between the RMs.

In the beginning of the showed scenario, each agent tries to solve the problem (the task *diagnosticTask* in agents diary. The CBR agent has not enough information to browse its stored cases. So it broadcasts an *Ask-info* message. At time T2, FMEA agent and MBR agent are processing the generation of hypothesis (*generateHypothesis* in their diary). At T3, FMEA agent has distinguished the two most probable hypothesis (*water-tank empty* and *pomp out-of-order*). It decides to share these results with other agents (Information in the mailboxes of CBR and MBR agents). At T5, MBR agent has discriminated all of its

generated hypothesis and then, it infers that the problem isn't an electrical one (because it deals only with electrical problems). MBR agent sends this information to CBR agent. The CBR agent, indexing its cases by type of problems can initiate its *generatHyp* task and then, propose a solution.

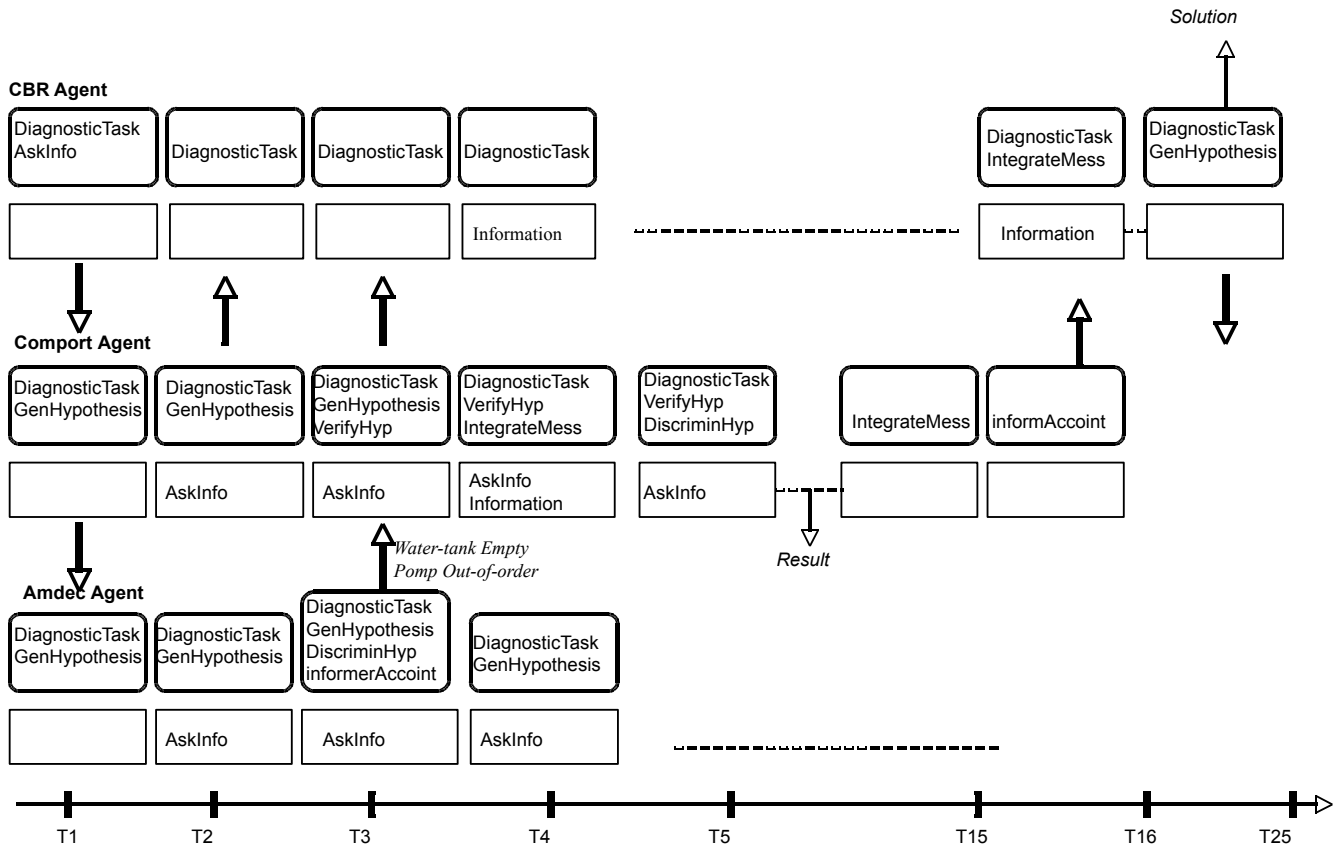


FIG.4 Cooperative scenario example

In this scenario, one can see how the **speed** of CBR agent, the **reliability** of MBR agent and the **exhaustivity** of a FMEA agent have been combined dynamically for a diagnosis of fault. The most important and original aspect of this scenario is that there is no central algorithm to guide the cooperation decisions. According to MAS paradigm, each agent has its own local control mechanism (supervisor level in FIG.3).

Over a dozen experimented scenarios, seven have particularly caught our attention. An agent seems to differ from others in a pivotal role. Such an agent seems to drive the resolution, without being able to carry alone the resolution to end. This is illustrated, for example, in the scenario of the figure FIG.4 by the CBR agent.

Based on this observation, a fundamental hypothesis has been formulated. It has been proceeded by induction to verify it.

Working hypothesis

There exists, in general, privileged reasoning model, not easily predictable automatically, for the resolution of each type of problem. However, this reasoning model requires some interventions without which under no circumstances would it give fully satisfaction.

To translate such a hypothesis in MAS. concepts, this can be represented in a particular agent role. But this is not so simple, because a preferred RM in the beginning of the problem-solving process doesn't necessary remain in this role in the rest of the scenario (ex. CBR agent in the cooperative scenario example of the figure FIG.4). It's why this mechanism has been implemented in an adaptive way. The next paragraph explains how a way to dynamically determine the pivotal role of the resolution, has been implemented.

3.3 Pivotal agent for dynamic organizations

The process of determining the pivot agent role is implemented as a part of cooperation activity of the agents. By mean of a dynamic organization, it's possible to build adaptive MAS ([8]). A first simple implementation has given promising results. So, the system has been improved it by a second version which is based on a negotiating method. The two versions are described bellow.

A-Simple pivot agent role determination

The determination of the pivotal agent role is done according to two axis. The first axis is called *communicational axis* because it concerns the exchanged messages between agents. The second one is called *computational axis* because it's related to agent contribution to resolution evolving. An agent recognizes itself as a *pivot* according to the two dimensions. That means that it is required for communication, and it's the one that drives the resolution.

The agents calculate their communicational weight Cw_c and their computational weight Cw_p . The communicational weight is measured by the disparity of received messages. This disparity consists in dividing the number of different senders on the number of acquaintances. The higher this ratio is close to one, more the diversity of messages is high, and this means that the communication is organized around the agent.

The computational weight is calculated by using a local domain dependent evaluation function. For the considered diagnosis application, the used function consists in dividing the number of assumptions processed H_p on the number of assumptions discriminated H_d (H_p/H_d). A *potential* $(Cw_c+Cw_p)/2$ is calculated. Once an agent has a ratio greater than a given threshold (0.5 for our tests), it becomes a candidate for a pivotal role.

The pivot agent role determination steps are summarized in FIG.5:

Pivot agent determination steps

1. *Eligibility for the pivot role*
2. *Broadcast a proposal*
3. *Accept or refuse the proposal*
4. *Conclude the assignment of the pivot role*

FIG.5: Pivot agent determination steps

As soon as an agent has identified itself as a candidate to the role of pivot, it broadcasts an information message to the other agents. If the reception of this message does not generate any conflict, the agent fulfills its role. It informs the other agents which will adapt their resolution or more exactly, their

cooperation attitude according to this new organization. A conflict occurs when the receiving agent is itself a candidate to the pivotal role. It accepts the proposal if its *potential* is less than the received one. Otherwise, it refuses the proposal. The initiator agent will simply cancel its proposal. In case of equality, one of the two agents is chosen randomly by the initiator agent.

This organization has improved the generation of cooperative scenarios because this allows an automatic detection of the end of the problem-solving process which become in charge of the pivotal agent. But, we were not completely satisfied because when several agents have equal scores, the selection is done randomly. Another problem is due to the oscillations of the system because the pivotal role is too dynamic and disturbs the stability of the system. This is due to the fact that several pivotal agents succeed in the same scenario. In several scenarios, the system never terminates.

B-Negotiated pivot agent role determination

The computation of the communicational weight Cw_c and the computational weight Cw_p are influenced by the knowledge that are manipulated by the agents. These knowledge can be related to the application domain (diagnosis in our case), to the cooperation process (requesting, negotiating,...) or to the control of the execution cycle of the agents.

To refine the mechanism of determining the pivot agent role and to minimize the problems mentioned in section A, a more sophisticated way for choosing the most adequate agent to this role has been implemented. For this purpose, the three dimensions of the domain, cooperation and control knowledge for each axis (communicational and computational) of the pivotal role determination have been considered.

More precisely, the third step of the pivot agent role determination process (see figure FIG.5) has been modified to deal with cooperation expertise and to negotiate before accepting or rejecting a proposal. When it receives a proposal, an agent initiates a negotiation process instead of sending an acceptance or a reject based solely on the value of the *potentials*. In the case of equality, the initiator agent begins by sending a message of request for the management of a *conflicting-pivot*. The two concerned agents consider the negotiating process according to the knowledge synthesized in figure FIG.6. Beginning from the most prior to the low prior knowledge, the negotiating agents exchange the value of their respective attributes.

(1)	Domain dimension (diagnosis)
	(1) Availability of methods associated with specific domain tasks
	(2) Rate of discriminated assumptions
	Cooperation dimension: measure rate of
	(3) Persistence of internal conflicts
	(4) Useful communications (answers and spontaneous received informations)
	(5) Successful negotiations
Control dimension: measure the rate of failures	
(6) Engine execution	
(7) Task criteria assignments	

FIG. 6: Negotiation knowledge dimensions

The figure above summarizes the most important facts identified to be relevant in determining the pivot agent role. For the domain dimension, the availability of the domain methods means that the agent is able to deal with the domain tasks. This information can be confirmed (or not) by the rate of

discriminated assumptions (cf. 3.3). The cooperation dimension measures the rate of solved conflicts and evaluates the quality of the communication by calculating the rate of useful exchanged information. It can also store success and failures of past negotiations. Control dimension considers engine failures, while trying to find a method for a task or assigning criteria to the tasks.

A value is calculated for each dimension (domain, cooperation and control) and then, it's considered during the negotiation process. The negotiation is simply a comparison between the respective values. The winner is the agent with the highest value. Such a value is an average of the components of the dimension. For example, the value of a cooperation dimension, is the average of the sum of the number of: persistent internal conflicts, useful communications, and successful negotiations. The Table 1 shows some examples of calculated values for two agents (A and B), where agent A is always the winner.

Table 1. calculated knowledge dimensions examples

Example 1	Agent A	Agent B
Domain knowledge value	0.5	0.2
Cooperation knowledge value	0.7	0.3
Control knowledge value	0.8	0.5

Example 2	Agent A	Agent B
Domain knowledge value	1	1
Cooperation knowledge value	3	1
Control knowledge value	2	5

Example 3	Agent A	Agent B
Domain knowledge value	3	3
Cooperation knowledge value	3	3
Control	3	2

knowledge value		
--------------------	--	--

The advantage of the implemented negotiation mechanism is its simplicity. The problem is that the priority is fixed a priori. As part of the future work, a more flexible negotiation model shall be implemented. This will allow exploring the situations where it's relevant to determine pivot agent role.

3.4 Results and evaluation

To evaluate the presented approach, three different cooperative scenarios have been implemented in the MAS, and then compared. In the first scenario (Scenario-1), cooperation is implemented without any explicit pivot determination mechanism. The execution of the MAS ends when a pre-defined time-out is reached for the complete scenario execution. In the second (Scenario-2) and the third (Scenario-3) scenarios, pivot agent role mechanism is implemented inside the agents, allowing them to organize the problem-solving process around such a particular agent named *pivot agent*. The two latest scenarios differ in the mechanism by which an agent is assigned by such a role. In Scenario-2, a simple method is implemented, while a negotiation method is implemented in Scenario-3.

Pivot agent determination steps are implemented as cooperation tasks (see FIG.1). Assigned methods represent the different ways to perform each task (step). The domain model (see FIG.1) of each task is concerned by knowledge that make the task activated and/or the methods usable. For instance, the main task of pivot determination process is activated when a given ratio is achieved for the problem-solving process evolving.

Let's resume to the diagnostic scenario described in Figure 4. In the beginning, CBR agent is the pivot because it "drives" the discrimination between a set of generated hypothesis. Between time t_3 and time t_5 , the pivot is the Comport agent because the communication is organized around it.

To compare the three scenarios, the oscillations of the system, the number of execution cycles of the agents and the proportion of cooperative tasks in relation to the number of domain tasks¹ are considered. For each scenario and each comparison criteria, the number of correct diagnosis is considered with the same datas (manually generated).

The ratio between cooperation and domain activity is variable for Scenrio-1, it's around 1/3 for Scenario-2 and around 2/3 for Scenrio-3. This means that the agents spent more "time" for cooperating in Scenrio-3. However, The numbers of execution cycles are nearly equal, for the two scenarios. Scenario-3 gives a little more satisfying solutions² than Scenario-2.

Even several pivot agents can succeed the same scenario and some scenarios can take place without any pivot agent has emerged, the number of execution cycles of the agents is reduced and the behavior of the system is globally improved in the second version.

Although, the MAS oscillates when no agent stands clearly for a pivotal role, it often terminates in Scenario-2 and always terminate in Scenario-3. The oscillations for Scenario-1 are avoided due the time-out method termination.

4. RELAED WORS

To our knowledge, there is no existing similar approach which can be compared to ours. However, several authors address the problem of integrating heterogeneous models. We have selected two of the most famous and recent systems implementing a integration of heterogeneous models, VLE and BIOCHAM. They illustrate respectively interoperable approaches and centralized ones.

VLE is a Virtual Laboratory Environment proposed by Duboz and his colleagues [11]). The implementation of VLE is based on the DEVS simulator ([14], [15]). This tool is dedicated to the

¹ It's a way to estimate the "time" spent in cooperating instead of solving the domain problem for which the system is mainly designed.

² The satisfiability is considered from the user point of view.

definition of the experimental frames and the analysis of simulation results. Since the integration is achieved at the operational level, the designers have to specify their models in the unified terms that are proposed by the tool. It's not the case for our system, the RMs decide in a distributed manner, when and what to communicate with other RMs. The interactions between the RMs are due to cooperation decisions. Of course, all the interactions are not optimal, in the sense that they don't always correspond to relevant coupling of the RMs. It's why the agents (implementing the RMs) in our system need to negotiate in order to manage the occurring conflicts.

BIOCHAM is a software environment for modeling biochemical systems ([3]). It combines a rule-based system for modeling biochemical systems, a numerical simulator, and query language based on temporal logic. The user writes a model by defining a set of rules with kinetic expression, a list of parameters values and initial conditions and biological properties as a list of temporal logic formulae. According to the type of study chosen by the user, the system receive different interpretations (kinetics ignored, interaction probability, etc.). The user has to deal with multi-modeling process, so he is supposed to know about all of them.

First results show that using MAS for integrating heterogeneous models raises many challenges. At methodological level, it keeps people using their preferred reasoning models. At computational level, the challenge is to design and implement cooperative reasoning as independent as possible from the application domain. The emergence of cooperative scenarios is the challenge which is related to the cooperative reasoning.

5. CONCLUSION AND PERSPECTIVES

This paper has proposed an original distributed approach for heterogeneous reasoning models integration. An adaptive multi-agent approach to generate, in a dynamic way, cooperative scenarios has been implemented. Each reasoning model is encapsulated in an agent. Constructing the cooperative scenario, reveals a way of combining the RMs so that the whole system can solve a problem which is beyond individual agents' capabilities. The dynamic observed in the system and the emergence of the pivot agent role, has been explored. So, an explicit pivot agent role determination process has been implemented in the system with two different strategies (a simple and a negotiated one).

The first obtained results are conclusive. Due to the investment necessary for conceptual modeling domain knowledge to experiment with great number of RMs, more work is necessary before providing quantitative results. Current work consist in looking for statistical laws to generate randomly large amount of data. Another way to get such data is to find a simpler application domain which would allow obtaining numerical results more easily.

Theoretical study already shows the superiority of a distributed and adaptive approach to centralized one. Indeed, centralized approaches can be used as source of inspiration to implement specific cooperative scenario.

Another current work concerns the study of the impact of the considered knowledge for the dynamic organization, on the MAS adaptivity. This aims to improve the negotiation step for the determination of the pivot agent role. Indeed, several argumentation models, based on logic formalisms, are used in MAS negotiation. Some of them use defeasible rules ([16], [17]) allowing complex argumentation and counterargument representation. Whilst others [1] argue for using classical logics. One can think that, this will supply some features to answer to questions such as: "does it interest always to have a pivotal agent role? And is it possible to differentiate the situations where such role is useful from the ones where it's not? For example, detecting that a pivot agent role will minimize the oscillations of the MAS seems to be a good prediction.

Even if the presented results are preliminary, they are very promising. Before to continue investigating, a confrontation to the multi-agent system community is essential.

A long term work aims to validate our working assumption regarding the existence of this notion of “preferred” reasoning model for each kind of problem.

6. REFERENCES

- [1] Philippe Besnard and Anthony Hunter, *Argumentation Based on Classical Logic*. I. Rahwan, G.R. Simiri (eds.), *Argumentation in Artificial Intelligence*, DOI 10.1007/978-0-387197-0_7, Springer Science+Business Media, LLC 2009.
- [2] Scott C. Bublin and R.L. Kashyap, “*CONSOLIDATE: Merging heuristic classification with causal reasoning in machine fault diagnosis*”. Proc. of the Fourth Conference on Artificial Intelligence Applications, San Diego, CA, USA, 14-18 March 1988 (Washington, DC, USA: IEEE Comput. Soc. Press 1988), p18-25.
- [3] Calzone, L. and Fages, F. and Soliman, S, “*BIOCHAM*}: An Environment for Modeling Biological Systems and Formalizing Experimental Knowledge”, *Bioinformatics*, Vol. 22, N°14, 2006.
- [4] De Kleer, J. and J. S. Brown. “Fundamentals of model-based diagnosis”. *SafeProcess 2003*.
- [5] Durfee, E.H. and Lesser, V.R. and Corkill, D.D., “Trends in Cooperative Distributed Problem Solving”, in proceedings of IEEE Transactions on Knowledge and Data Engineering, 1995.
- [6] Goldstein, I. and Papert. S. “*Artificial intelligence, Language, and the study of Knowledge*”. *Cognitive Science*, vol 1 1977.
- [7] Gorges C., *Polymorphism of human reasoning* (translated from french “Polymorphisme du raisonnement humain”), Paris, Edition PUF 1997
- [8] Jennings, N, Gibbins, N. and Kota, R., *decentralized Structural Adaptation in Agent Organizations*. Published in Book *Organized Adaptation in Multi-Agent Systems*. Springer-Verlag Berlin, 2009
- [9] Lesser, V., Zhangn C., Abdellah, S., “Integrating organizational control into multi-agent learning”. *AAMAS (2) 2009*: 757-764.
- [10] E. Masse, C. Curt, and M. Le Goc. “Développement d’une méthode de maîtrise de la sécurité des barrages : méthode basée sur le diagnostic multi modèles appliquée à un barrage en remblai”. *Actes des 25e rencontres de l’AUGC, Bordeaux, 23-25 mai (2007)*.
- [11] G. Quesnel, R. Duboz and É. Ramat, *The Virtual Laboratory Environment - An Operational Framework for Multi-Modelling, Simulation and Analysis of Complex Systems*. *Simulation Modelling Practice and Theory*, (17), 641-653, April 2009.
- [12] Gérard Sabah. *L’intelligence artificielle et le langage*. Volume 2 : processus de compréhension. Edition Hermès 1989.
- [13] M. Tambe and D V. Pynadath and C. Chauvat & A. Das and G. Kaminka, “Adaptive agent architectures for heterogeneous team members”. *ICMAS’2000*.
- [14] B. P. Zeigler and J. Lee. – *Theory of quantized systems: formal basis for devs/hla distributed simulation*. In : *SPIE*, pp. 49–58. – 1998.
- [15] B.P. Zeigler and H.S. Sarjoughian. – *Creating distributed simulation using devs m&s environment*. In : *2000 Winter Simulation Conference*. – 2000.
- [16] D.Nute, *Defeasible logic*. In D. Gabbay, C Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol 3, pages 355-395/ oxford university Press, 1994.
- [17] Alejandro J. García, Jürgen Dix and Guillermo R. Simari, *Argument-based Logic Programming*, I. Rahwan, G.R. Simiri (eds.), *Argumentation in Artificial Intelligence*, DOI 10.1007/978-0-387197-0_7, Springer Science+Business Media, LLC 2009.