COORDINATION IN MULTI-AGENT DECISION SUPPORT SYSTEM

Application to a Boiler Combustion Management System (GLZ)

Noria Taghezout[1], Abdelkader Adla[2] and Pascale Zaraté[2]

[1] *Department of Computer Science, University of Es-Senia Oran, BP 1524, El-M' Naouer, 31000, Oran, Algeria*

[2]*IRIT, INPT – ENSIACET, 118 route de Narbonne 31062 Toulouse Cedex 9, France*

Keywords: Multi-agent system, Group Decision Support System (GDSS), MADS (Multi-agent Decision support system), Coordination agent.

Abstract: In Multi-agents systems, the cognitive capability present in an agent can be deployed to realize effective problem-solving by the combined effort of the system and the user. It offers the potential to automate a far wider part of the problem solving task than was possible with classical DSS. In this paper, we propose to integrate agents in a group decision support system. The resulting system, MADS is designed to support operators during contingencies. We experiment our system on a case of boiler breakdown to detect a functioning defect of the boiler (GLZ: Gas Liquefying Zone) to diagnose the defect and to suggest one or several appropriate cure actions. In MADS the communication support enhances communication and coordination capabilities of participants. A simple scenario is given, to illustrate the feasibility of the proposal.

1 INTRODUCTION

In Distributed Artificial Intelligence (DAI) systems, problem solving agents cooperate to achieve the goals of the individuals and of the system as a whole. Each individual is capable of a range of identifiable problem solving activities, has its own aims and objectives and can communicate with others (Jennings, 1993). Typically agents within a given system have problem solving expertise which is related, but distinct, and which has to be coordinated when solving problems. Such interactions are needed because of the dependencies between agents' actions, the necessity to meet global constraints and because often no one individual has sufficient competence to solve the entire problem.

In this paper, we propose to integrate agents in a group decision support system. The use and the integration of software agents in the decision support systems provide an automated, cost-effective means for making decisions. The agents in the system autonomously plan and pursue their actions and sub-goals to cooperate, coordinate, and negotiate with others, and to respond flexibly and intelligently to dynamic and unpredictable situations.

We argue that an agent-oriented approach is the most natural and appropriate mean to achieve better support for the group decisions, and we propose to improve the coordination protocol.

The manufacturing process of the oil plant (GLZ), selected as application domain in this study, is split into two subdivisions: Utility subdivision and Process subdivision. The Utility subdivision is constituted of Pumps, Desalination Unit, boilers, Turbo-generators and Air Compressors while the Process subdivision concerns the tasks of manufacturing of liquefied Gas. This subdivision is composed of 6 strings where a string is group of equipments. Every string contains 10 sections which are going to be used to liquefy gas. The management system of the boiler combustion is one of the most critical systems for the good functioning of the plant and has a high impact on the methods of cogitation and apprehension of various problems related to maintenance (see Figure 2). The exploiting staff is often confronted with situations that impose a quick reaction of decision-making. This requires consequent human and material resources and adapted skills (for more details see (Adla, 2007)), to diagnose the defect and to suggest one or several

Taghezout N., Adla A. and Zaraté P. (2009).
COORDINATION IN MULTI-AGENT DECISION SUPPORT SYSTEM - Application to a Boiler Combustion Management System (GLZ).
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Artificial Intelligence and Decision Support Systems*, pages 194-199
DOI: 10.5220/0001999301940199

appropriate cure actions.

The reminder of the paper is organized as follows: Section 2 describes our contribution. It is followed by section 3 which covers the integration of agent technology into a DSS. The multi-agent architecture for group decision support systems and the corresponding coordination protocol are described in section 4 and section 5. We also present an example of a scenario in section 6. Finally, Section 7 gives some concluding remarks.

2 CONTRIBUTION

Research that studied group decision support systems in the existing literature used mainly face-to-face facilitated GDSS. Some of its results may not apply to distributed teams (Chen, 2002) that, it is difficult for distributed teams to arrange face-to-face meetings or to meet at the same time virtually. Moreover, although most presented GDSS environments try to solve problems in the real world, the lack of an integrated procedure, from decision identification, basic information acquiring, to final decision proposed, makes the systems only partially supportive or even needful of outside assistance. Still, despite the existence as well as the extensive use of numerous general-purpose commercial systems, it is our belief that these systems do not readily fulfill the needs or operational usages of specialists or experts in different organizations to render their expertise in GDM processes.

In our study we consider another gap: the coordination problems when they occur have several causes. Most of them are a consequence of limitations in both the decision making processes and the technological support for communication. For this reason, the information and tasks related to the decisions made in GDDS have to be visible to other organizations to keep the relief effort coordinated between the agents.

In addition, the quality of support received during the decision making processes is the key to reaching optimal decisions. Decisional guidance mechanism provides the decision makers with step-by-step guidance throughout the decision-making process and allows them to evaluate more alternatives. As a result, DSS users with decisional guidance can easily come up with better decisions than those with no decisional guidance.

Mahoney et al. (Mahoney, 2003) pointed out that when faced with Complexities in a decision situation, decisional guidance helps users to choose among and interact with a system's capabilities. They argued that in less structured tasks that deal with uncertainty and risk, users need more guidance to choose among competing solution techniques or among alternative methods of processing information to structure an appropriate decision-making process using the GDSS.

3 AGENT INTEGRATION IN DSS

We got inspired by two main research works. Firstly, the main ideas resumed in the table described in (Forth, 2006) were very interesting for our study (see Table 1). It defines how the capability of an agent may be utilised in a DSS application, and also identifies alternative agent design architectures suitable to underpin this. As a constituent part of problem-solving in the domain, an agent may choose particular sources of information to use. Data Gathering may be a function within an agent (sensing), or a dedicated activity of a specialised information agent if the task is complex.
Secondly, the approach developed by Zamfirescu (Zamfirescu, 2003) addressed the problem of self-facilitation in GDSS by establishing a common and meaningful high-level collaboration pattern among the group members inspired from the SP theory. In his GDSS approach, the main entities have been defined: the personal assistant agents, the resource agents and the plan agents.

Table 1: Mapping DSS functions to agent capabilities.

DSS Function	Agent Function
Data collection	Knowledge acquisition and assimilation
Model creation	Perception and knowledge representation
Alternatives case creation	Planning and reactivity
Choice	Action selection
Implementation	Action execution

4 THE MULTI-AGENT SYSTEM

Agents are used to collect information outside of the organisation and to generate decision-making alternatives that would allow the user to focus on solutions that were found to be significant. According to this a set of agents is integrated to the system and placed in the DSS components (as shown in Figure 1), we distinguish:

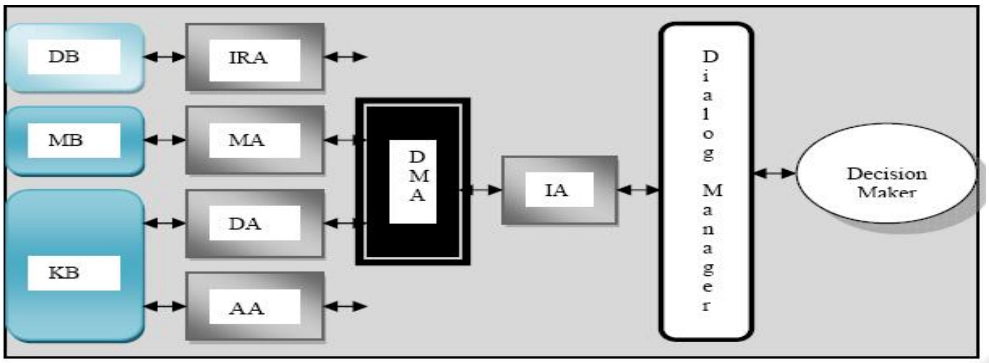

Figure 1: Agent architecture for Individual DSS (Adla et al., 2007; Jennings, 1993).

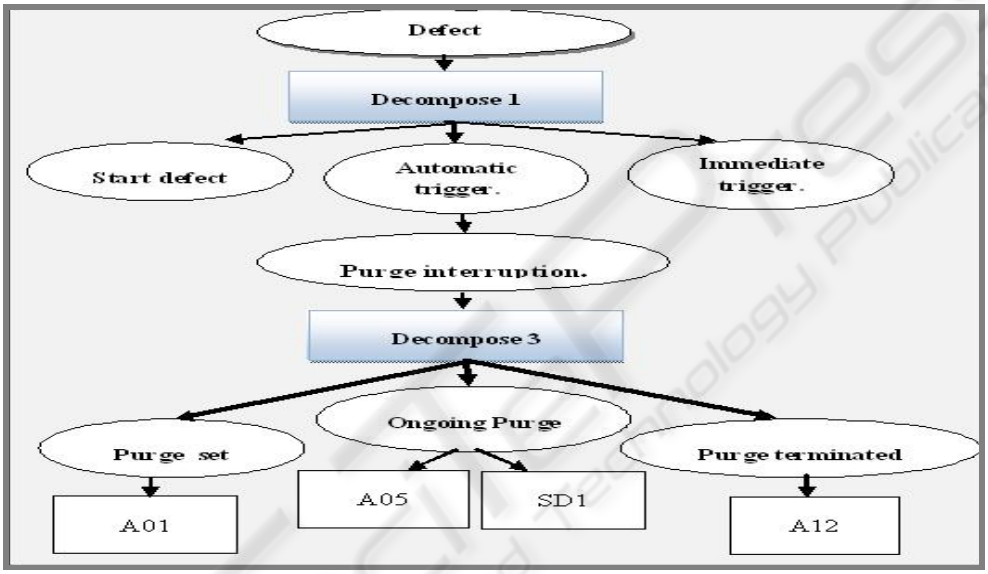

Figure 2: A partial hierarchy of tasks and methods of the application (A01, A05 , SD1 and A12: feasible methods; Decompose1, 2: Decomposition methods).

The Interface Agent (IA) Continuously receives data from the process – e.g. alarm messages about unusual events and status information about the process components.

A Decision Maker Agent (DMA) performs most of the autonomous problem solving.

An Information Retrieval Agent (IRA) primarily provides intelligent information services.

A Diagnosis Agent (DA) is activated by the receipt of information from DMA which indicates that there might be a fault.

Knowledge Management Agent (KMA) comprises, manage and update knowledge base;

The Action Agent (AA) generates a plan of action which can be used to repair the process once the cause and location of the fault have been determined. As described in Figure 3, a refined representation of DA, AA and KMA agents is given.

The Coordinator Agent provides two services to task agents: (i) it computes summary information for hierarchical plans submitted by the task agents, and, (ii) coordinates hierarchical plans using summary information.

5 A COORDINATION PROTOCOL

The problem solving mechanism is based on a set of cycles until the entire problem is solved. Each cycle consists of the following steps :(1) identifying candidate methods; (2) identifying triggered methods ;(3) selecting a method; (4)assigning the method to an agent; (5) executing the method; and (6) Evaluating the task state.

5.1 Agents Structure

Clearly, all the modules representing the inner structure of an agent may depend on each other.
This is especially true for the local problem solver and the coordination module (as shown in Figure 3.) which do not only exchange real time information but, in addition, must coordinate their decision rules and performance criteria. If we consider the relationship between the coordination module, the problem solver, and the knowledge base. We found that they have to make sure the data needed available. The communication between the agents may roughly be described by the coordination module and the interface component. They are describing the way how agents may communicate.

5.2 Agent Communication Language

The agents needs are formalized as a set R of requests $r \in R$, each of which expresses a question about agent's data or an action to perform given a set of parameters. These requests appear both in human-agent and in interagent dialogues.
The request performative rp can take the following values, for example see Figure 4, Figure 5 and Figure 6):

- Ask-for requests that represent questions about the agent's data.
- Assert-is for assertions about agent's data values (to answer to Ask-for requests).
- Order for requests that represent actions to perform.
- Affirm for acknowledgment when an action has been performed (in answer to an Order request).
- Assert-cannot to express that the agent is unable to perform a command.
- Assert-can to express that the agent can perform a command, but misses a field value to be able to do so.
- Unknown for asserting that the agent doesn't know a field or the overall action to perform.

5.3 Communication between Agents

The requests structure presented above is also used by agents in their communications. Indeed, these requests formalize the contents of the messages exchanged by agents, each of which is structured as follows: ***m* = [id, *C*, sender, receiver, agenda]**.
Where id is the message id, C is the message content, sender and receiver are the sender and receiver agents' ids and the agenda term for indicating the memory of the message associated to the initial agent request.

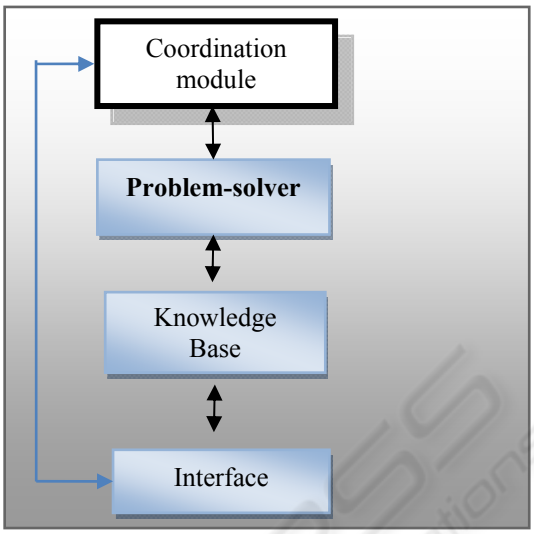

Figure 3: Representation of DA, AA, and KMA agents.

5.4 Discussion

When an agent receives:

- An Order request, and if it misses a value to perform it, it builds an Assert-can answer.
- An Assert-can request, it replaced the required field stated in the assert-can answer by its value if owned.
- An unknown request, it looks in the corresponding field in the Ask-for request.
- An assert-is request, when answering an Ask-for request, it can trigger it by using the information received trough the Assert-is answer.

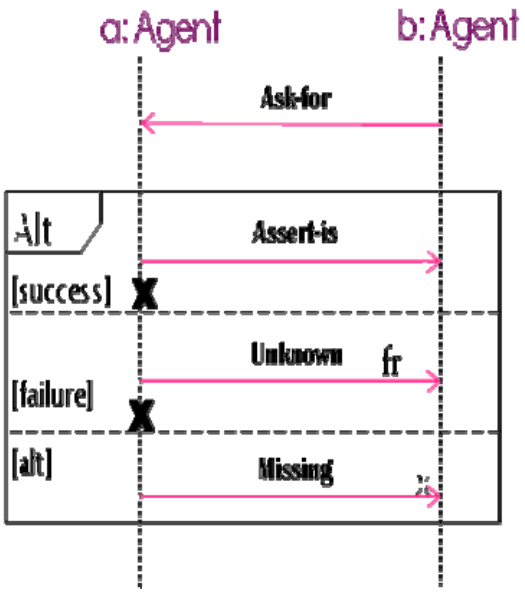

Figure 4: Ask-for protocol.

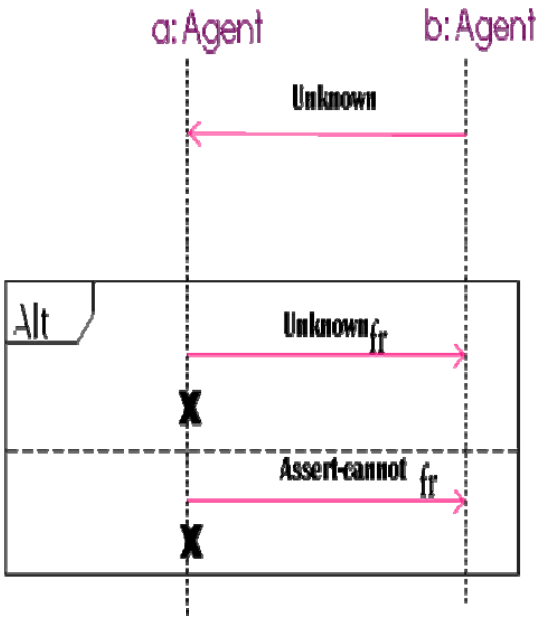

Figure 5: Unknown protocol.

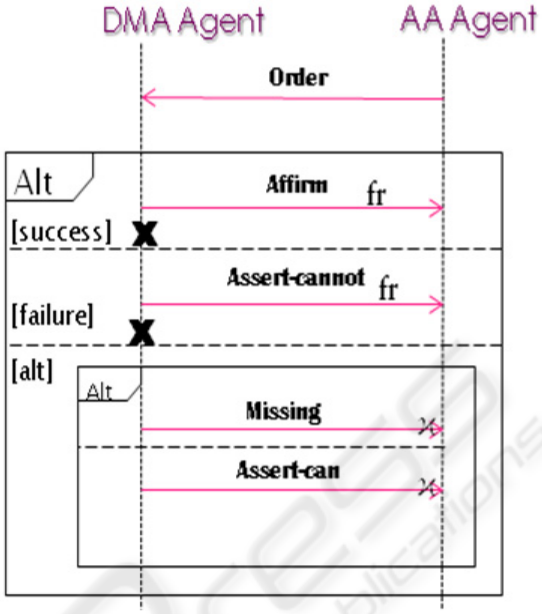

Figure 6: Order protocol (fr: final response).

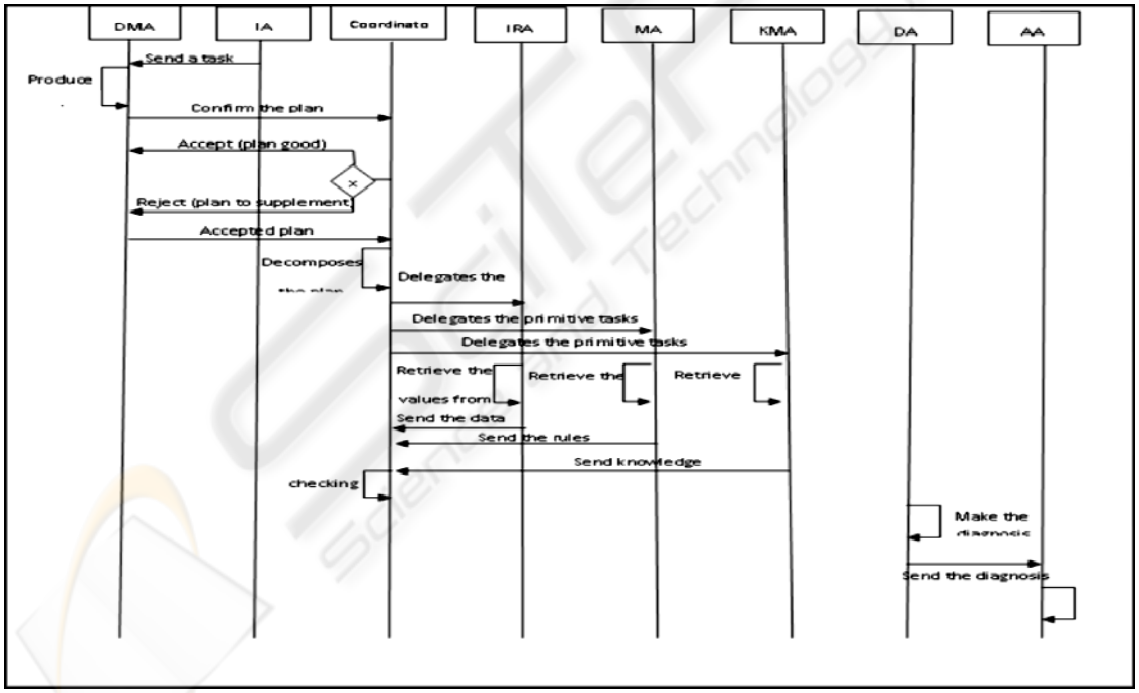

Figure 7: A coordination scenario diagram (AUML).

6 A COORDINATION SCENARIO

When the task management agent (DMA) receives a task from an interface agent (IA), it decomposes the task based on the domain knowledge it has and then delegates the primitive tasks to the other agents (IRA, MA, KMA, DA or AA). The task management agent will take responsibility for retrieving data, modelling, diagnosing fault, planning action, resolving conflicts, coordinating among the related agents and finally reporting to the interface agent which conveys the results to the user.

As described in Figure 7, the DMA agent first gets input data through the interface agent. Next, the modelling agent searches for rules to select a suitable model and to execute the model to get analytical results. Additionally, all the parameters values needed by the models are retrieved from the database via the information retrieval agent. After finishing model analysis, the diagnosing and the action agents use the results of the model analysis to identify the fault causes and to perform a suggested action plan.

7 CONCLUSIONS

MAS paradigm offers a new dimension with respect to GDSS integration with complementary services, making it easier to build complex and flexible architectures suitable to organizational settings (Zamfirescu, 2003).

In this paper, we have integrated agents into GDSS for the purpose of automating more tasks for the decision maker, enabling more indirect management, and requiring less direct manipulation of the DSS. In particular, agents were used to collect information and generate alternatives that would allow the user to focus on solutions found to be significant. We expect to finish the system implementation that supposes all the decisional tools and validation with other oil plants which are geographically dispersed.

REFERENCES

Adla, A., Soubie, J-L., and Zarate, P., 2007. A Co-operative Intelligent Decision Support System for Boilers Combustion Management based on a Distributed Architecture. In *Journal of Decision Systems*, Lavoisier, Vol. 16, pp.241-263.

Chen, M., 2002. TeamSpirit: Design, implementation, and evaluation of Web-based group decision support system. In *Decision Support System*, Lavoisier B. V., 43. pp. 1186-1202.

Filip, F.G, 2008. Decision support and control for large-scale complex systems. Annu Rev Control, doi: 10.1016/j.arcontrol.03.002.

Forth, J., Statis, K., and Toni, F., 2006. Decision Making with a KGP Agent System. In *Journal* of Decision Systems, Lavoisier, vol. 15, pp. 241- 266.

Jennings, N. R., 1993. Coordination Through Joint Intentions In Industrial Multi-Agent Systems. *AI MagaZine*, 14 (4). pp. 79-80.

Mahoney, L. S., Roush, P. B. and Bandy, D., 2003. An investigation of the effect of decisional guidance and cognitive ability on decision making involving uncertainty data. In *Information and Organization* 13 (2), pp. 85–110.

Zamfirescu, C.B. , 2003. An agent – oriented approach for Supporting Self-Facilitation in Group Decisions. In *Studies in informatics and Control*, 12(2), pp. 137-148