ON APPLYING MULTIAGENT SYSTEMS TO A COLABORATIVE ASSEMBLY ROBOT SYSTEM: A COMPARISON OF TWO COORDINATION SCHEMES

Doru Pănescu, Marius Şutu, Ştefan Dumbravă

"Gh. Asachi" Technical University of Iaşi,
Faculty of Automatic Control and Computer Engineering
Department of Automatic Control and Applied Informatics
Bd. D. Mangeron 53 A, Iaşi, Romania, dorup@ac.tuiasi.ro

Abstract: A comparison of two multiagent schemes is presented. One is a blackboard based system, the other uses a message exchange coordination mechanism. They are compared regarding the possibility to plan and control an assembly manufacturing system with two industrial robots working in cooperation. The coordination and communication issues are discussed, the advantages for each of the proposed approaches being pointed out. The aim of the research is to increase the autonomy and flexibility of robot based manufacturing systems.

Keywords: multi-agent systems, planning, blackboard systems, cooperative robotics, robot assembly.

1. INTRODUCTION

A main feature needed for the present manufacturing systems is flexibility, which is to be related with the important actors of the respective systems, the industrial robots. These are still among the most important exponents of the Robotics technologies, gathering several improvements; in spite of their implemented abilities, some features as their deployment and autonomy are not as expected. The control methods, as well as the classical programming approaches seem to allow no further spectacular progress, so other methods and techniques have to be considered. Such an attempt is presented in this contribution, namely a possibility offered by a field of Artificial Intelligence (AI) - the Multi Agent Systems (MAS).

The MAS domain is a topical and evolving one. This explains the lack of standardization and validation for most of its techniques. In this regard, the aim of this contribution is twofold: to determine certain agent based schemes that should be appropriate for the use in collaborative Robotics, and to make a comparison between them in order to determine the strong and weak points. Considering these goals, the following paragraph notices certain specific features needed for the application of MAS in assembly based manufacturing. Then the focus is on two main issues for MAS: planning and coordination mechanisms. Two different architectures are proposed and then compared. This contribution refers to an on-going research and thus the proposed schemes are only partially implemented. Even so, some conclusions could be already drawn in the final paragraph.

2. SOME FEATURES SPECIFIC FOR THE APPLICATION OF MAS IN ASSEMBLING

One important application area for MAS is manufacturing. When considering such an approach, several issues have to be decided, and specific aspects appear when the respective systems are to be used in an assembly process. This kind of application relies on a planning phase: the assembling of a specific product supposes a well determined sequence of actions, which can be characterized as both goal and environment dependent. The respective stage can be related with AI, as planning has been a main preoccupation for this field from very beginning. Its evolution can be summarized as starting with a research on finding a general planning scheme, and then being more devoted to obtaining specific planning algorithms. This new trend applies when thinking to the MAS planning phase for the assembly manufacturing processes, too.

The respective application area conducts to some specific characteristics, as:

- The assembling plan is known for each product, meaning that it can be already present into the agents' knowledge bases, as a priori knowledge.
- The actions may not be always performed in the same order, as a robot solving sequence can depend on the available resources and other environment state conditions; such variations are even more frequent in a cooperative approach, when two or more robots are involved.
- To obtain improved solutions (faster, with an optimum resource use, easy to face the goal changes) distributed approaches become a must, leading to MAS and robot cooperative assembly.

In the manufacturing system used for experiments there are two flexible cells organized around two industrial robots (Panescu and Dorin, 2006). Each robot is able to solve manipulation and assembling tasks, being connected with some other devices; one robot is capable to serve and command a machine tool, while the other can work with a computer vision system. Both robots receive information about the parts existing in the storage devices from their own working area and can control the transfer of the parts from one cell to the other via a closed loop conveyor. The two robots can be involved in assembling operations too, as an assembly table is placed in an area reachable by both of them. It results the main role the two robots have in the manufacturing system, explaining why the considered architectures have been developed for two agents dedicated to the robots (see the Figs. 2 and 3). Fig. 1 shows a product used in experiments, which supposes the assembly of four parts, named as A, B_1, B_2, and C; some further details can be found in (Panescu *et al*., 2006).

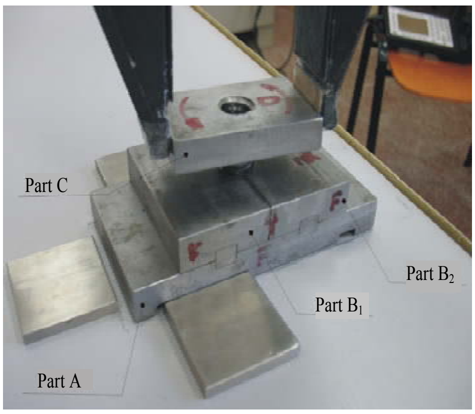

Fig. 1. The product considered for assembling

As about MAS planning, the first point is to clarify what distributed nature refers to (Durfee, 2001); it may denote only the execution phase, when several actors are able to perform the plan actions, even simultaneously, or the proper planning process, i.e. the plan is devised by several entities (agents). A final possibility is when both the planning and execution phases are distributed. In the considered case, the following points can be revealed:

- The execution stage is a distributed one; the robots may act simultaneously in the assembling process in order to minimize the manufacturing time.
- The planning phase can be solved both in a centralized and distributed manner. As usually a single input exists for the goals of the manufacturing system, the respective goals can be processed and their solving plans determined by a central unit/agent, which should assign tasks to the other agents. A distributed planning approach is also possible, when all the agents contribute to both plan development and execution. The advantages and disadvantages of these two methodologies are well known (Durfee, 2001) and some further comments are revealed by the present research.

The planning approach must be related with the MAS coordination method. Because the agents' decisions are based on local knowledge (in the proposed architectures each agent is connected with certain sensors, which means it can get a partial view on the environment), and the agents' actions can be done simultaneously, a coordination mechanism is needed; this should apply starting from the planning stage. From this point of view, various schemes have been considered in MAS, without any final standardization or definite conclusion. That is why it is worth comparing the different possibilities and determining their performance in specific applications. Two methods have been considered: a blackboard based approach and respectively a direct negotiation technique based on message exchange.

For both discussed schemes an important issue is the decision making process. This can vary from a logical deductive method, based on a symbolic internal model of the environment, to a purely reactive approach, when the decision means a mapping of the sensorial information into a predefined behaviour. It is easy to observe that a combination of these should provide the optimum result for the MAS involved in manufacturing. Indeed, such a system has some a priori knowledge on the way various tasks can be fulfilled in a state based approach, which conducts towards deliberative agents, but it also needs feedback from the environment, resulting in the presence of a reactive component. This conclusion was applied for both solutions presented below, being materialized by an interleaving of the planning and execution phases.

3. A BLACKBOARD BASED MAS FOR ASSEMBLY

The first presented solution is mainly a distributed one, as the plans are supposed to be discovered by the contributions of two agents, but it still has a centralized component, namely a blackboard (see

Fig. 2). Blackboard systems can be considered a paradigm for distributed problem solving (Huhns and Stephens, 2001). Starting from the brainstorming experts' analogy, the respective method principle of operation can be easily described. There is a knowledge base – the blackboard, which is used as a common resource by several modules (experts, knowledge sources or agents) in an incremental solution development. Each agent should interact with the blackboard when appropriate, namely when it is able to make a contribution or when the blackboard contains a piece of information that the agent is supposed to read.

The connections between the various components are shown in Fig. 2, for the experimental system briefly described in the previous paragraph. The agents (*Agent*$_1$ and *Agent*$_2$ in Fig. 2) are connected with the controllers of the two robots, receiving sensorial information via these devices (Panescu and Dorin, 2006; Panescu *et al*., 2006). Each agent operates in a loop containing 3 main phases: writing to the blackboard, reading the blackboard, and sending the control information to the operational part. The first phase supposes that the respective agent has carried out a planning operation and its result is sent to the blackboard. The second phase is meant to allow each agent the receiving from the blackboard specific information (mainly data on the planning state) permitting it to decide about the next operation to fulfil. This can be a proper command sent to the operational part for an action to be carried out, or the decision can be about the necessity of continuing planning. Some further details are needed.

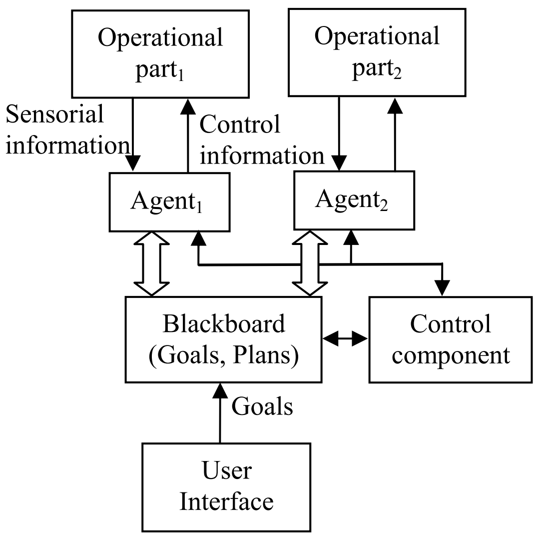

Fig. 2. The blackboard based MAS architecture

Each agent begins its activity with an initiation stage, when it should fill in its own knowledge base with the facts corresponding to the information provided by the sensors that the respective agent is connected to. This starting procedure is completed with a first blackboard reading in order to get the goals that have already been placed there by the user.

After that, the above mentioned cycle takes place. Each agent tries to find a plan for the goals present in the blackboard. These are ordered, as they have a priority level attached; in the considered manufacturing process, the user establishes the priorities according to the deadline that each product has to comply to. So, an agent should work out to find the plan for solving the goal with the highest priority. As already mentioned, three main points are guiding the planning operation: the distributed approach, the interleaving between planning and execution, and the use of some predefined plan schemes. Thus, an agent is not supposed to find an entire plan (maybe it even cannot do this), but the first action or actions that he can do in order to solve the highest priority goal, according to its knowledge on how that goal can be solved and the information on the environment state. The respective partial plan is sent to the blackboard.

As known from the blackboard systems' principle of operation, the interaction of various entities with the blackboard is mediated by a control component. When an agent intends to write to the blackboard it will send a request to the control component. This decides the right moment for the information transfer, as explained later. The types of information an agent may place in the blackboard are:

- A partial plan the agent can achieve and that can be already started (it means the respective plan begins with an action that can be performed by the agent in the present environment state).
- A partial plan the agent can contribute with in solving one of the present goals, but that cannot be already run, as there are some preceding actions that the respective agent is not able to do.
- A goal (in fact a sub-goal) the agent discovered in its planning activity and that is beyond its operational abilities; this means it is a goal that the agent is placing in the blackboard in order to receive help for its solution from the other agent.
- Information on the result of an action fulfilment, in accordance with the necessary reactive behaviour. Depending on the action result, detected by the agent sensorial system, the MAS can continue the plan execution relying on the previous result, or be obliged to enter a re-planning phase when the result of an action is not the expected one.

As about the blackboard reading operation, this is also done under the control component supervision. Namely, this is in charge with detecting the moment when a blackboard piece of information is necessary for an agent and should be read by this. It refers to one of the following cases:

- A plan (most often partial plan) that an agent proposed is to be executed, and the respective agent should be informed. This means it is a plan that was chosen as having the best cost.
- A new goal appeared in the blackboard and an agent should be informed about this, as it is a potential contributor to the respective goal solution.
- There is no plan available for the present goals and the agents should be informed in order to try other planning solutions or to detect new sensorial information or even to ask for the user help.

As about the last phase of the agent cycle, namely when it transmits commands towards its operational system, this meant in the considered system certain information sent to the robot controllers. Without going into details, for this coupling (physical robot – software agent) a rearrangement of robot programs was needed, under the form of several routines; the information sent by a software agent is the one starting such a routine. In our system, examples of such routines are: placing of a certain part on the assembly table and its fixing, machine tool loading, part transfer via conveyor.

The coordination process for this solution is tuned by the control component. Its principle of operation is according to the following guiding points.

- When the blackboard contains no plan to be put into execution, the writing operations of the agents are preferred. When more than one blackboard writing operation is asked, there is an established order of agent preference, which is dependent on the goals present in the blackboard. For example, for the goals that imply part identification tasks the agent of the robot connected with the computer vision system is preferred, while for the part processing tasks the other agent is chosen to first write the blackboard, as the corresponding robot is able to work with the machine tool.
- When the blackboard contains some solutions for a goal, but not all that are known as possible for the respective goal, a writing operation is favoured, too. This situation supposes that a priori knowledge is incorporated in the control component, specifying that several agents are potential contributors for a certain goal working out.
- When the blackboard contains all the agents' plan contributions for a certain goal, the agent reading operations are preferred. Actually, this supposes that first the control component should inform an agent about the fact that the blackboard contains a piece of information for it, about the type of information, and then the proper agent blackboard reading should follow. About this case, it is to notice the coordination role of the control component, as it is the one to choose from two or more partial plans the one with the lowest cost.
- When the blackboard contains goals not read by the agents, the reading operations are preferred, too. It means a goal driven operation is favoured.

4. A MESSAGE EXCHANGE BASED MAS ARCHITECTURE

The second scheme for the MAS is presented in Fig. 3. This was designed as a system with two agents, considering the manufacturing system described above, but its functionality can be extended to systems with more agents. The two approaches contain some common points. Thus, in both cases the agents are dedicated to the main actors of the manufacturing process (in our case, the two robots). The way the agents are interfaced with the manufacturing system is also the same, as well as the MAS cycle (planning, coordination, execution). The main difference with the previous scheme concerns the coordination and implicitly the communication mechanism. In this case there is no control part to mediate between agents, so they have to coordinate by themselves with an appropriate negotiation protocol, based on exchange of messages.

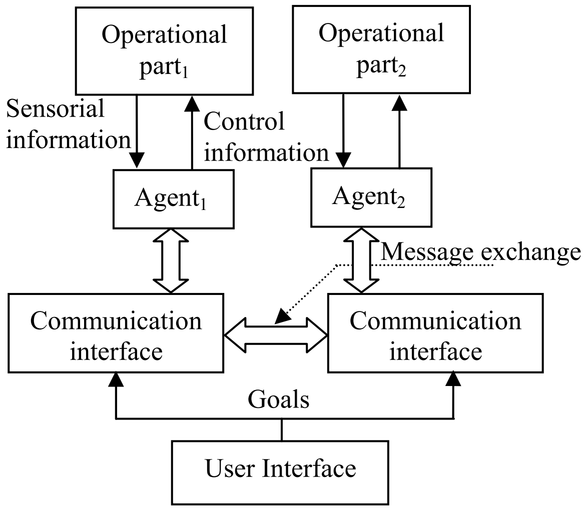

Fig. 3. A message exchange based architecture

The considered application is about cooperative agents, i.e. the agents' goals are not antagonistic. Even so, a negotiation is necessary because the agents might work out the same goals with some resources used in common. A negotiation protocol for MAS is defined by certain issues, namely: efficiency, stability, simplicity, distribution and symmetry (Huhns and Stephens, 2001). In the considered architecture these were established as follows.

Regarding the efficiency, each agent calculates the cost of its actions and sends the respective result to the other agent; thus, the decision is simple to choose: the current solution with the minimum cost should be preferred. In this way, nor a waste of resources may appear, neither instability – there is no possibility for an agent to deviate from the above rule. The distributive character is also provided, as when the agents know each other their proposals, they can come up with the solution, without a central unit help. Concerning the simplicity and symmetry, these result from the way the communication process supports the agent negotiation.

First, all the agents inform themselves about the assembly goals. As in the blackboard approach, these are already ordered according to the user provided information. Each agent is trying to determine a partial plan for the goal that must be solved first. If it finds such a plan, a message is sent to the other agent. The communication is solved by appropriate interfaces, which depend on the agents' physical connection. As an important issue, it was considered that a synchronous communication type is appropriate. Indeed, after an agent decided about a partial plan, it has to know whether the other agent possesses a better plan or not. This means the agent will wait the reply that can be:

- *"Your plan is the best, you can execute it".*
- *"I have a better plan, so your plan should not be used".*

An intermediate case may appear when the costs of an agent's actions for part of the plan are the best and for the rest of the plan the other agent has better costs. In this case, the agent receiving the message should modify the plan, changing it for those actions that it can perform better (with lower costs). The respective modified plan is sent back, as the message answer, according to the synchronous protocol.

A particular case appears when one or both agents have no plan for a goal. If only one agent cannot contribute, it should reply validating the other's proposal. If both agents have no plan, the following communication procedure is used. When an agent performs no operation (this means no plan is devised, no message is read) for a certain interval, though unsolved goals exist, it will send a message to the other agent with the content: *"No plan available for the highest priority unsolved goal"*. If the agent receiving the message is in the same situation, its answer will be with the same content, and then the agents have the following possibilities (otherwise the same as in the previous architecture):

- To make more detailed goal decomposition according to the a priori plan schemes the agent possesses in its knowledge base; if a solution is found, the message exchange mechanism should be restarted in accordance with the above mentioned procedure. For example, such a situation appears when no processed part is available in the robot storage devices, but certain raw parts and the machine tool are available, and the required part can be obtained by processing. This case necessitates a more detailed plan in comparison with the situation when the processed part is already available.
- To ask the user help; for example, the user is supposed to supply some further parts in the storage devices.
- To wait for a change in the environment that might make possible a planning solution (for example, the computer vision system is running and it will identify a new part in the storage device).

The above possibilities are considered in the respective order; this means that first an agent tries to solve the impasse by a further planning operation, done at a finer level of granularity, then it asks the user contribution and finally enters a waiting state.

5. A COMPARISON BETWEEN THE TWO PROPOSED SCHEMES

As already mentioned the two proposed approaches have several common points. To notice the differences, it is worth commenting their application on the considered case study. The example used in experiments refers to assembling a product composed of four parts, whose structure can be observed in Fig. 1. The partial ordered (non-linear) plan for solving the goal of assembling such a product is represented in Fig. 4; this should be understood as the action of placing part A is to be executed the first, then the placement of parts B_1 and B_2 can be done in any order (even simultaneously), and finally the action for assembling the part C is to be carried out.

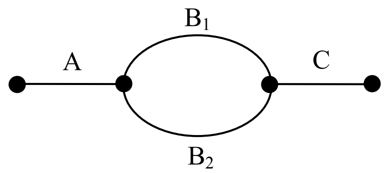

Fig. 4. An example of non-linear plan

Let us analyse the corresponding planning phase for the two schemes introduced in the previous paragraphs. In the blackboard approach, to decide on a part of the plan that is linear, for example that referring to the action of placing part A, the following steps might appear: one agent is proposing action A, the other agent is proposing the same action with a different cost or specifies that it is not able to execute part A assembling. Then the control unit is making the decision that is communicated to the chosen agent, and after the respective agent reads the blackboard it can launch the proper action of part A handling and assembling. The implementation taken into account is a rule based one; this supposes that each agent, as well as the control component should be materialized as rule based programs, in CLIPS (CLIPS, 2006). In this case, as a measure of efficiency, one may count the number of rules' run. In the above mentioned sequence this is four: two rules for the agents' proposals (the blackboard writing operations are thought that cannot be done simultaneously!), one for the control unit decision and one for the chosen agent information. If the same planning problem is considered for the message exchange approach, only three rules are involved. Thus, one agent is sending a message to the other with its plan proposal of executing the action A (this is the first rule); then a rule is fired by the agent receiving the message to decide whether it has the possibility to carry out the respective action with a better cost or not, and finally a rule is necessary to send the answer to the agent that initiated the plan.

The advantage for the message exchange based architecture is kept when a non-linear plan is involved and some concurrent actions can be performed. Indeed, for the case of the two actions B_1 and B_2, in the blackboard scheme the run of five rules is needed: the two agents are sending their contributions to the blackboard, then the control component is composing the plan from the two proposals (in the considered case this means the decision for the two actions to be executed simultaneously by the two agents), and finally two rules are fired when the two agents read the respective decision from the blackboard. Instead of this, in the message exchange approach only three rules are needed: two result from the two messages sent between the agents (the information message and the answer) and one rule is required for the comparison and decision made by the replying agent.

This difference in efficiency in favour of the second architecture is easy to explain. The blackboard architecture supposes an indirect interaction between agents, by the blackboard and the control unit. These centralized components determine a bottleneck and the above remarked temporal inefficiency. Meanwhile the blackboard architecture has as an advantage the easy way the MAS communication is put into practice. Independent of the number of agents, the same principle applies: for a new agent only the communication channel with the blackboard must be created. In the above analysed case, only two agents were involved, which explains the simplicity of the communication part in Fig. 3. When more agents would be involved a kind of broadcasting mechanism has to be created and some complications appear for the planning phase of the respective architecture: a protocol for the plan validation must be set up, to detect the moment when the agreement of all the agents has been reached. A sketch for such a protocol is:

- All the agents send messages in a broadcast manner for their plan proposals.
- When all the proposals have been received, and a solution can be decided based on the best cost, this is put into practice by agents without any other communication operation.
- When the above step cannot reach a decision (for example, there is a goal for which no agent has proposed a solution), a message exchange between certain agents should be necessary, following the procedure considered for the case of two agents. In such a situation the blackboard approach can be more efficient, as it can be easy to embed heuristics for the doubt cases into the control component.

From the advantages determined by a blackboard based MAS one can find some as being important in manufacturing control. Thus, as several independent knowledge sources may be used, that can possess their own representation and inference mechanisms, modularity and easy integration result for the respective approach. This can be worthy for the manufacturing systems, when different modules may be developed independently and must be integrated.

6. A FEW DETAILS OF THE CLIPS IMPLEMENTATION. CONCLUSIONS.

For both presented approaches a CLIPS implementation is under development. This rule based environment was preferred for certain reasons. It offers good programming facilities for the components of the designed schemes: the agents, the blackboard and its attached control component. It also possesses certain interfacing facilities, as it can be embedded into a Visual C application (Şutu and Panescu, 2007). Meanwhile a rule based program is well suited to implement planning schemes as the ones considered for the proposed architectures, by means of the opportunistic way the rules are fired. The agents' knowledge bases (namely their working memories) contain facts on the present environment state; the rules' left hand side holds the planning patterns (mainly the actions' preconditions) and so, through the pattern matching process, an action is included into a plan when its preconditions are satisfied. A further CLIPS option that has been used is the modular development and execution of knowledge bases by means of a special construct, named "*defmodule*" (CLIPS, 2006). This is useful both for the agent and blackboard construction, facilitating the specification of certain constraints on facts' visibility from one module to the other. The coordination and communication protocols can be supported by this modular organization, for example allowing the facts to be changed between two agents to be grouped.

A conclusion of the current research is the necessity to find the adequate coordination and communication mechanisms for the MAS used to plan and control a manufacturing system. Usually the planning phase involved in such a process does not imply complex decisional procedures or difficulties in finding the right cost functions; as an example, in the presented system the costs of actions were considered to be proportional with the robots' movements.

Though the use of AI in Robotics is not new, several issues are open problems even now. The application of AI based planning systems is still limited and the presented results show that new possibilities are offered by MAS. These systems provide well suited methods and tools, due to their distributed approach that makes tractable even the most complex manufacturing systems. By comparing several methods of distributed AI and choosing the most appropriate MAS architecture, the design and implementation stages can be made easier.

REFERENCES

CLIPS Ref. Manual (2006). www.ghg.net/ clips/

Durfee E. (2001). Distributed Problem Solving and Planning. In: *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence* (Weiss, G. (Ed.)), pp. 121 - 157. The MIT Press, Cambridge.

Huhns M. and L. Stephens (2001). Multiagent Systems and Societies of Agents. In: *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence* (Weiss, G. (Ed.)), pp. 79 - 114. The MIT Press, Cambridge.

Panescu D. and V. Dorin (2006). A Case Study on Robot Use in a Flexible Manufacturing System, *Bulletin of the Polytechnic Institute of Iaşi*, Tom **LII (LVI)**, Fasc. 7B, pp. 117 – 126.

Panescu, D., G. Varvara, A. Pricop, St. Dumbrava and M. Postolache (2006). On multi-robot systems design and implementation; a comparison based on three case studies. *Memories of Romanian Academy Scientific Sections*, IV, Vol. **XXIX**, (in press).

Şutu M. and D. Panescu (2007). A Rule Based Implementation of a Multiagent System for Manufacturing, *The 9th Intern. Symposium on Automatic Control and Computer Science*, Iasi, Romania (to be published).