ON CLIPS MULTIAGENT SYSTEM IMPLEMENTATION FOR A MULTI-ROBOT APPLICATION

Doru Pănescu, Sorin Miron

"Gh. Asachi" Technical University of Iaşi
Faculty of Automatic Control and Computer Engineering
Department of Automatic Control and Industrial Informatics
Bd. D. Mangeron 53 A, Iaşi, 700050, Romania
E-mail: dorup@ac.tuiasi.ro

Abstract: Some results of a research concerning multi-robot system planning are presented. The environment is an industrial one with mobile robots in charge with part transferring. The proposed approach is based on multiagent systems, the contract net protocol being used to solve the co-ordination problem. Some details are given regarding the design and implementation that refer to using CLIPS as software environment, with benefits got by combining rules and objects.

Keywords: mobile robots, co-ordination, multiagent systems, rule-based programming, object-oriented programming.

1. INTRODUCTION

The problem to be solved refers to planning a multi-robot system containing several mobile robots that are supposed to work in a manufacturing environment. For the present research only the high level planning was considered and thus some accepted simplifications are as follows. The working area is represented as a planar surface, with the robots being points characterized by their x and y coordinates (see Fig. 1). The number of mobile robots (these are denoted R_1 – R_5 in Fig. 1) is not a priori settled. The positions of the obstacles are known, these being mainly the machine tools, depicted as MT_{ij} in Fig. 1.

The goals for the mobile robots are issued by a manufacturing planning module considering the products to be obtained and the requirements of the processing operations, according to the Computer Integrated Manufacturing (CIM) principles (Rehg, 2000). Such goals could be for the mobile robots to transfer a part from a machine tool to another one, as the goal indicated by a dash arrow in Fig. 1. When robots have restricted working areas like in the considered environment, different solutions are possible for a goal fulfilment, obtained by co-operation. Instead of a classical scheme supposing that the tasks of the robots must be decided in a centralized manner, a multiagent approach may show some advantages, as described in this contribution.

The paper is focused on the co-ordination aspects, revealing certain issues of the design and implementation of multiagent systems (MAS) when these are used for the considered environment. The software tool that was considered is CLIPS, which was mainly used in expert system applications (Giarratano and Riley, 1989). Meanwhile, this possesses certain characteristics important in a multiagent implementation. Thus, the necessary agent behaviour, commitments and plans can be easily translated into rules and objects. The required communication mechanism can be solved by means of message passing within an object oriented approach, as CLIPS is offering the possibility to combine rules and objects. After a brief presentation of the some important notions regarding the MAS, some aspects of solving robot co-operation with the contract net protocol are presented.

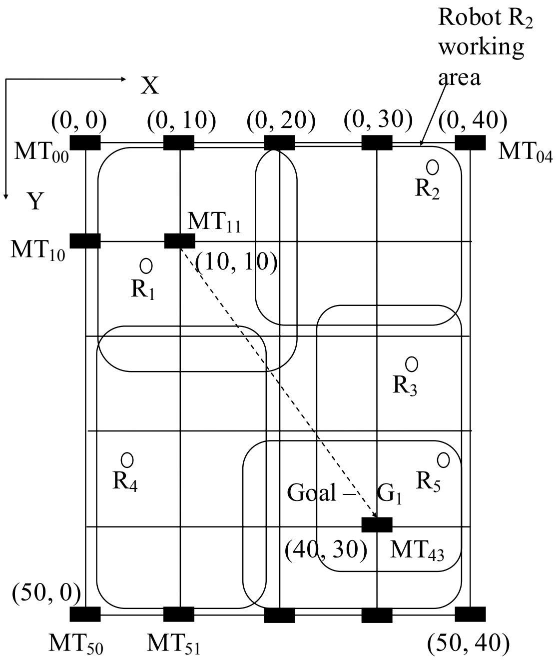

Fig. 1. The manufacturing environment with several mobile robots

2. BASIC ASPECTS OF A MULTIAGENT SYSTEM. THE IMPORTANCE OF THE CO-ORDINATION MECHANISM

A topical field of Computer Science with many implications for Robotics is Distributed Artificial Intelligence (DAI). The respective connection is more important when a multi-robot system is involved. In such a case, the respective link may determine certain advantages: complex tasks may be solved through co-operation by a team of robots and the strategies from DAI can facilitate the robot planning and control systems carrying out. As present CIM systems usually contain several robots that are supposed to work together to solve manufacturing goals, it is natural to establish an association between such a system and DAI and it becomes more valuable to study these aspects.

Multiagent systems constitute the best known and the most used approach of DAI. If one tries to make a synthesis of the large set of definitions for MAS (Jennings, 1994; Weiss, 2001; Wooldridge, 2002) the result can be: these are systems with several entities interacting with their environment and possessing those capabilities that enable a global coherent behaviour towards achieving certain specified goals, when system knowledge, sensing and acting abilities are distributed among the respective entities. The components of an MAS, namely the agents, are always considered in conjunction with their environment and must consequently have the capacity to perceive, reason and act; it is easy to notice that these are characteristics often used when defining/building a robot. In this way it is obvious to consider for planning and controlling a system with several robots the association between robots and software agents, as it is made in this contribution.

Besides the advantages of MAS (the most frequently invoked being the possibility to solve complex problems by making use of the distributed character of the tasks and of the *"divide et impera"* principle), certain issues must be solved to get an operational system: each agent has only incomplete information and specific constraints, the control is distributed being based on decentralized and asynchronous data. The solution mainly means to establish the right co-ordination mechanism between the agents. If one refers to non-antagonistic agents, then co-ordination means co-operation and several mechanisms were considered with respect to this case. All of them try to get the optimum solution by a problem decomposition approach, while avoiding agents' extraneous activities and their inappropriate resource competition. Generally these strategies combine in specific manners planning and agent synchronization by message exchange.

One such an MAS co-ordination method is the contract net protocol (Huhns and Stephens, 2001; Wooldridge, 2002), that is an interaction procedure designed to problem solving in a distributed approach. Being issued from the market mechanisms, it tries to find a solution to the so-called connection problem, which for an MAS is: finding the most appropriate agent to work on a given task. To apply it, there must be an agent issuing the goal (task to be solved) which is named the manager. All the agents that may potentially solve the goal are called contractors. Without giving the details (some points should be clarified through the examples of the following paragraphs) the manager is announcing the goal and then waits for the contractors' answers. When the agents' offers are received, the manager can decide which is the best one, and the corresponding agent will be awarded with the contract for the task solving. The flexibility of this mechanism is increased when goals can be decomposed on several levels, as illustrated in the considered multi-robot system.

3. APPLYING THE CONTRACT NET PROTOCOL FOR A MULTI-ROBOT SYSTEM

The multiagent software architecture was implemented based on the CLIPS environment. Each mobile robot is considered as an agent, together with the agent dedicated to the goal manager module. All the agents are materialized as CLIPS objects. In this way the necessary communication is got by means of the object message passing. Thus, the implementation of the contract net protocol by using objects meant the design of appropriate messages and methods. For the present stage, the user sends the goals to be fulfilled as messages to the goal manager, which is by itself an object. In the proposed solution the contracts are also objects that will be transformed by the procedures (methods) attached to the objects/agents.

The contract net protocol was adapted to the specificity of a multi-robot system and to the object oriented implementation. The two main phases, namely the manager's choice of the best contract and the execution of the chosen contract are solved by using both objects and rules. In the first phase the steps that are followed by the manager are:

1. Sending of the current goal to all possible contractors – to solve this step the message passing mechanism is combined with the possibility of the CLIPS object oriented system to handle collections of objects; there is a set of agents related to all active mobile robots and the message is sent to the elements of this set.
2. Reception and analysing of the answers from all the contractors – this supposes that the objects dedicated to various robots will send messages to the manager and the method which operates on each message makes use of the information on acceptance or declining, the cost of the goal solving and the name of the contractor.
3. Informing the chosen contractor on the decision that it must fulfil the goal according to the previously made offer.

The flexibility of the contract net protocol is gained by the possibility to shift the agents' role. So, in the considered environment it is possible that a mobile robot is not able to achieve by itself a goal, but a part of it. The respective robot should become a manager for the part of the goal that is out of its possibility, while being contractor too, as it received the initial goal from a manager. In such a case the benefit of the co-operation feature of an MAS is significant, the whole structure appearing as a consistent multi-robot system. The steps guiding an agent representing a mobile robot when receiving the announcement for a new goal are as follows:

1. Reception of the message for a goal fulfilment.
2. Evaluation of the possibility to achieve the goal. If the goal can be entirely accomplished by the agent then send a positive answer to the manager and an offer (this means that the procedure is finished), else go to the step 3.
3. Evaluation of the possibility to complete a part of the goal. If there is no such a possibility then go to step 6. Else continue with step 4.
4. Create sub-goals for all the parts of the goal that are beyond the agent capability and send messages with these new goals to all the other potential contractors.
5. Reception and analysing of the answers from the potential contractors. If there have been received positive answers for all the parts of the initial goal then calculate the overall cost of the goal achievement and send the positive answer to the manager which initiated the goal; else go to the step 6.
6. Send a negative answer to the manager.

Some remarks can be made regarding this solving strategy. Due to the way the role of manager can be transferred from one agent to the other the problem solving can entirely benefit from the "*divide et impera*" principle. Meanwhile the weak part could be the necessity of an agent to possess the ability to divide a goal in sub-goals corresponding to problems easier to solve than the initial task; there may be problems for which the sub-goals interaction complicates this decomposition (d' Inverno and Luck, 2001). But, for the planning tasks of the mobile robots in the considered conditions this is not a difficulty, as the following paragraph shows. In the second phase, the execution one, the manager is only notified about the moment when the goal is actually accomplished by the agent in charge with it.

4. CO-OPERATION IN MULTIAGENT SYSTEM DEDICATED TO MOBILE ROBOT PLANNING. A CASE STUDY

Certain issues have to be solved in order to put into practice a multiagent based methodology. Part of them were encountered in the considered case and got some specific solutions. Thus, a manager, either being the one which originated a task or one that decomposed a goal, must have a criterion to compare the received offers. As normal for such an evaluation process, a cost function is to be used, which in the considered case has the following form:

$$f(O_i) = \sum_{j=1}^{k} C_j L_j \qquad (1)$$

namely the cost of the offer O_i is the sum of the costs for the movement of the k robots that contribute in solving the respective goal. The cost for the robot R_j is easily determined by the product of the unitary cost for the movement of that robot (C_j) and the length of the robot path (L_j), which can be calculated by using the coordinates of the initial, via and final points.

An interesting development of the contract net protocol method that was carried out is the possibility for an agent to consider several goal decompositions. Such a case is presented in Fig. 2, the agent associated to robot R1 having three possibilities to partially contribute to solving the goal G_1. Besides the offer denoted as Of_1 that determines the maximum displacement towards the goal position, other two offers were also considered, namely the ones that determine a maximum closeness in the x and y axis, respectively. These additional offers may be useful when the constraints on the robot working areas prevent the choice of some offers. This is the case for the considered example, where the offer Of_1 can provide no solution, which is not the case for the other two offers. For example, when considering the offer Of_2, the robot R_1 is supposed to create a new goal (denoted as sub-goal G_2 and marked with a dash arrow in Fig. 2), which should be solved in co-operation by the robots R_2 and R_3.

A further analysis is necessary for the case when an agent becomes a sub-contractor. A hierarchy will be created in this situation that can be represented as a tree. The manager initiating the goal is placed in the root position and the contractors are placed in successive levels until the goal fulfilment. For the case of Fig. 2 part of the corresponding tree is

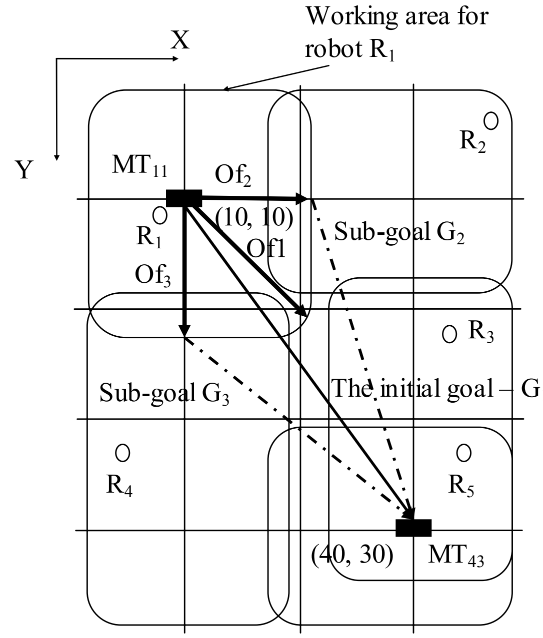

Fig. 2. An example with several offers for the same agent

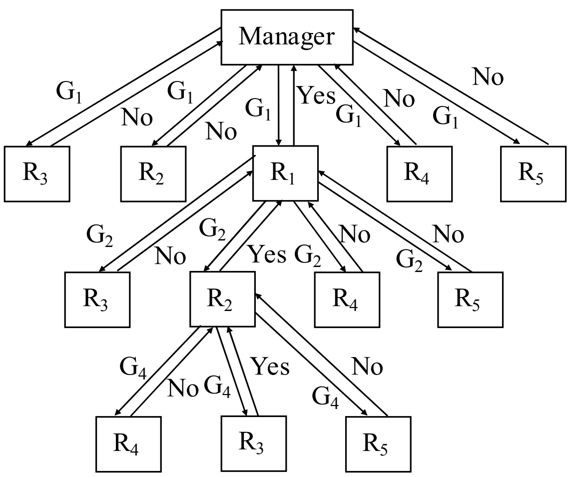

Fig. 3. A goal decomposition with a multi-robot solution

presented in Fig. 3. Though only the robot R_1 can reply with a positive answer to the manager request to solve the goal G_1 (it is the single robot that can reach the initial position), the others are involved in achieving the successive sub-goals (G_4 is the sub-goal generated by the robot R_2, not represented in Fig. 2). The tree of Fig. 3 does not represent the entire message exchange – another decomposition should be made for the sub-goal G_3.

About the transmission procedure, though the broadcast type can be used when announcing a goal, to increase the efficiency a message will not be sent to the agents that are already involved in solving part of the respective goal (it means they are predecessors of the current node). This is a specific aspect of the considered environment. It can be proven that this condition holds as long as the robots' working areas are convex and the circular paths are excluded. In order to be able to construct the solution for the various goals that could be simultaneously present, both the information on the agents that are the goal initiators and on those that respond with positive answers has to be kept. A scheme on how this information is passed for the considered example is presented in Fig. 4 (to simplify it, only the agents with a positive answer are considered). First the information flows up-down, from the goal manager to the agent corresponding to robot R_1 and this launches the new goals G_2 and G_3, getting positive answers from the robots R_2 and R_4 respectively, after a further decomposition. Then the information has a reverse flow from the last contractors towards the initial manager, which gets the complete information.

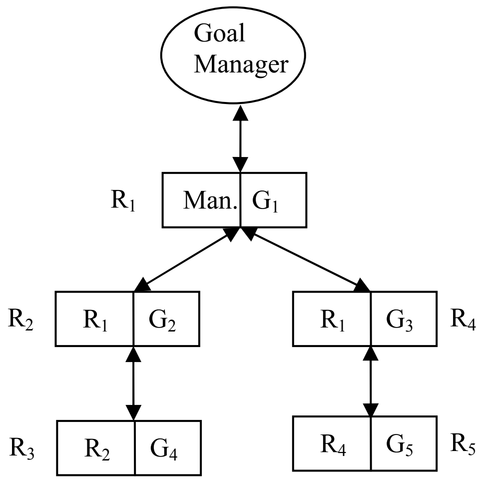

Fig. 4. A scheme of the information flow for the considered example

This analysis allowed the design of various objects necessary to implement the contract net co-operation protocol in CLIPS. The information stored for each robot is: the robot present position, the limits of its working area, the robot movement cost, the contract which is presently involved in. A contract is also materialized as an object, containing information on: the initial and final positions, its cost – determined according to expression (1), the list of nodes (agents) implicated in the respective contract, the time limit - this parameter is used by agents to order the goals in accordance with their deadlines. The combination of objects with their message exchange and rules facilitated the implementation. For example, it is obvious that a tree like the one of Fig. 3 is generated from the root towards the low level contractors by making use of the message exchanges on the goal and sub-goals announcement. On the other hand, when building a contract the tree must be passed starting with the deepest level. To get this behaviour the level for each node of the tree is stored at the tree creation and a fact with the deepest level is always present in the knowledge base; then a rule of the following form is activated:

```
(defrule  contract-building
 ?a <- (level   ?x&:(> ?x  0) )
           =>
 (do-for-all-instances ((?agent   Agent))
   (do-for-all-instances  ((?contract   Contract))
      (and (=   ?contract:level   ?x)
             (eq   ?contract:bid   yes) )
      (if  (eq   ?agent   ?contract:creator)
          then
           (send  ?contract:sender   answer  ?agent
              yes (cost  ?agent) )
      )
   )
 )
 (retract ?a)
 (assert  (level  (-  ?x  1) ) )
)
```

In the above expression the CLIPS variables are preceded by ?. In the left hand side of the rule the variable ?x is bound to the current value of the tree level and this value is decreased after each execution to obtain the tree passing towards its root. In the right hand side, the rule operates with collection of objects, namely the set of agents and contracts respectively (by making use of the CLIPS function "*do-for-all-instances")*. Through the included test, only the contracts of the agents from the current level that got a positive answer are selected, and the cost of the respective contract is sent to the agent that initiated the respective goal. In the discussed rule object encapsulation is used, as different slots of the current instance are directly accessed; thus *?contract:level*, *?contract:bid*, *?contract:creator*, and *?contract:sender* represent for the current instance of the *Contract* object, the level of the tree where it was created, its bid type (positive or negative), the contract originator and respectively the agent willing to contribute. The program run determined good solutions for various scenarios with different goals and arrangement of robots. As an example, for the situation presented in Fig. 2 the solution after the program run is shown in Fig. 5 (the cost for the movement of robots was considered the same, equal to 1).

4. CONCLUSIONS AND FUTURE WORK

This paper investigates a possibility of applying DAI in Robotics, namely in planning a multi-robot system dedicated to a manufacturing environment. A connection was made between several mobile robots that work in the same area and the agents of an MAS. Though a vast literature exists regarding the mobile robot path planning, the respective methods are still not very used in industry. This may be explained by the difficulty to have in an industrial medium mobile robots with complex sensorial and navigation systems. Instead of this, the present contribution refers to a possible industrial solution with several simple mobile robots, which should have some a priori settled working areas and paths, while the flexibility necessary for a CIM environment can be obtained by a distributed problem solving approach, which is neither difficult to implement nor intensive equipment consumer.

As about the complexity of the proposed algorithm, the first carrying out supposed an exhaustive search. Even such a strategy can be considered because the time parameter may not be critical for the planning phase, and the complexity of the problems for a well-structured industrial environment is supposed to be reduced. Meanwhile, certain precautions can reduce the method complexity. Besides the presented constraint that forbids a further considering of a node from the search tree when this node was already used and can no longer contribute to solution, some other aspects can be considered. Thus, in a distributed implementation (the present solution was tested by running on a single computer) when software agents should be materialized on different robot computers, it is clear that the real time constraint could be easily satisfied. Moreover some a priori knowledge on robots' abilities could be used in filtering the message exchange process. For the considered example, one such filter can be: a manager will not send a goal towards a robot having a working area that does not include the goal starting point. In this way the communication may be reduced, aspect important in an MAS approach.

The combination of rules and objects that CLIPS provides is a strong point in the presented research. This stands from the known fact that objects and expert systems offer good tools for an agent based implementation. Meanwhile, clear differences exist between objects, rules and agents respectively, referring to the enhanced requests on decision, autonomy and flexibility capabilities for the agents' case (Wooldridge, 2001). The presented study shows how the agent rationality and negotiation procedures can be implemented in a rule/object platform, but for the agent multi-threaded based autonomy the necessity of interfacing the CLIPS program appears, a possible solution being presented in (Pănescu *et al.*, 2004).

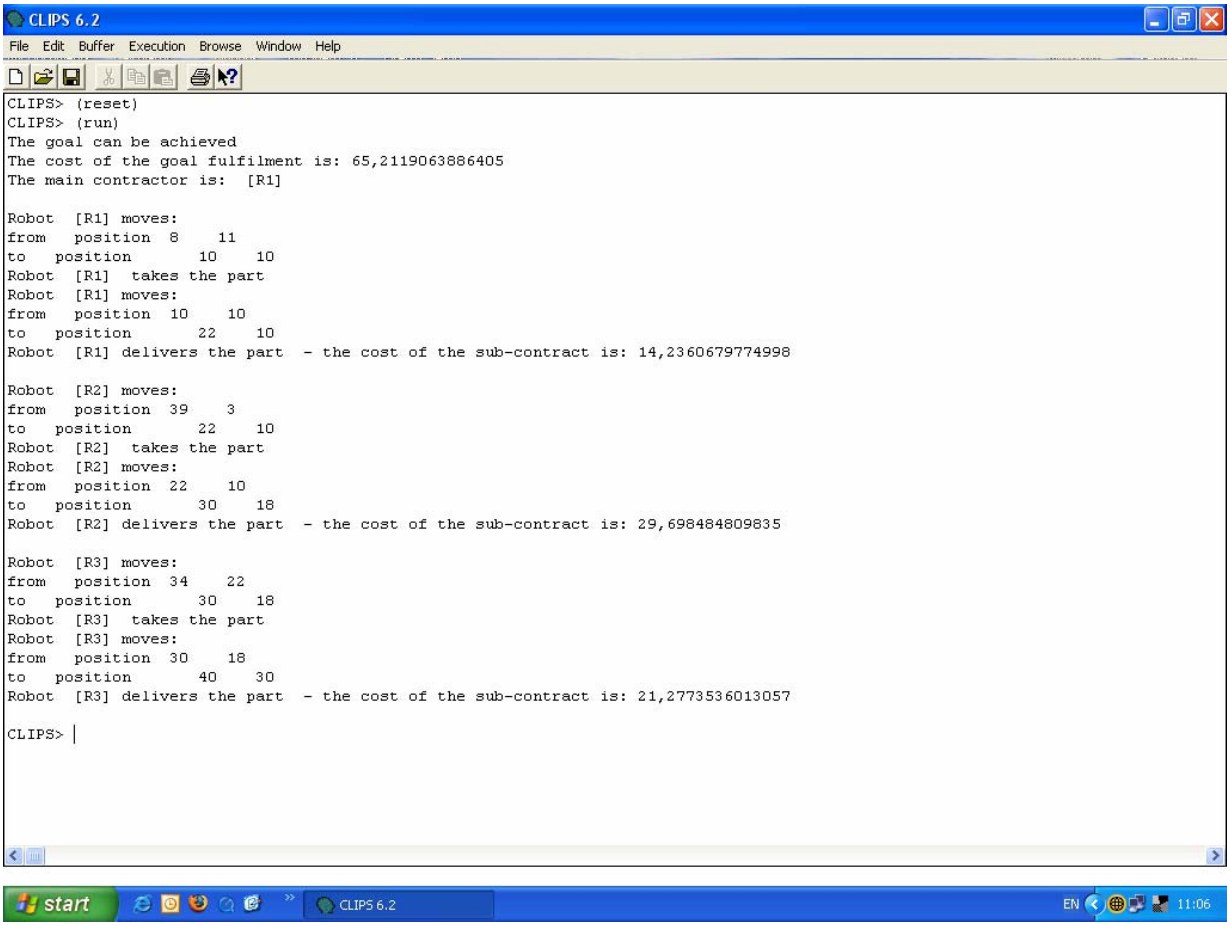

Fig. 5. The program solution for the considered example

Based on the presented method and the results of the already made experiments, some developments are planned: designing and implementation of the communication protocol for the case of the agents' operation on a computer network, development of adequate user interfaces, carrying out of tests on some miniature mobile robots. Besides these, part of the theoretical and practical results should be transferred in an MAS implementation of a multi-robot system composed of industrial manipulators, as suggested in (Varvara and Pănescu, 2005).

ACKNOWLEDGEMENT

This research was supported through the grant no. 540/2005 of the Romanian Ministry of Education (CNCSIS).

REFERENCES

D'Inverno, M., and M. Luck (2001). *Understanding Agent Systems*, pp. 42 – 59, Springer, Berlin.

Giarratano, J., and G. Riley (1989). *Expert Systems: Principles and Programming*, pp. 373 – 497, PWS-KENT, Boston.

Huhns, M., and L. Stephens (2001). Multiagent systems and societies of agents. In: *Multiagent Systems. A modern approach to Distributed Artificial Intelligence*, (Weiss, G., Edit.), pp. 99 – 103, The MIT Press, Cambridge.

Jennings, N. (1994). *Cooperation in industrial multi-agent systems*, pp. 3 - 18, World Scientific, Singapore.

Pănescu, D., V. Dorin, and St. Dumbravă (2004). On the use of rule-based programming in Robotics. *Proceedings of the 8th International Symposium on Automatic Control and Computer Science*, October 22 - 23, Iași, Romania, CD ROM.

Rehg, J. (2000). *Introduction to Robotics in CIM Systems*, pp. 110 – 132, Prentice Hall, Upper Saddle River.

Varvara, G, and D. Pănescu (2005). On the design of a multi-agent architecture for a flexible manufacturing cell. *Proceedings of the 14th International Workshop on Robotics in Alpe-Adria-Danube Region - RAAD05*, May 26-28, Bucharest, Romania, pp. 424 – 429.

Weiss, G. (2001). *Multiagent Systems. A modern approach to Distributed Artificial Intelligence*, pp. 1 – 9, The MIT Press, Cambridge.

Wooldridge, M. (2001). Intelligent Agents. In: *Multiagent Systems. A modern approach to Distributed Artificial Intelligence*, (Weiss, G., Edit.), pp. 34 – 36, The MIT Press, Cambridge.

Wooldridge, M. (2002). *An introduction to Multiagent Systems*, pp. 1 - 16, 194 – 196, John Wiley & Sons, Baffins Lane.