A System for the Control of a Manufacturing Cell Based on Pattern Recognition and Intelligent Agents

Leonard Iurescu, St.Gh. Pentiuc, Adrian Graur

Research Center in Computer Science
University "Stefan cel Mare"of Suceava, Romania
leonardiurescu@yahoo.com, pentiuc@eed.usv.ro, adriang@eed.usv.ro

Abstract

The aim of the paper is to present a solution based on a Multiagent System for a flexible manufacturing cell. The cell is supervised by a multi-cam system. An image processing subsystem and a pattern recognition module are used in order to detect the various types of pieces that will be handled by the robot. The control is assured by a multi Agents System. The intelligent agents cooperate with all information sources and give commands to a conveyor band, a stoking area, and a robot arm that takes pieces from the conveyor and put them in the stocking area.

1. Introduction

The paper presents a solution for controlling a flexible manufacturing cell destined to sort and store work pieces in the storage area of machinery in a machine tool manufacturing factory. This cell is fed with workpieces by a conveyor band which is monitored by a digital video camera connected to a computer. By processing the images, the computer determines the work piece type and its position relative to the conveyor band. Based on that information the computer will direct the robotic arm to take the work piece and place it in the storage area or the machinery. The storage area is monitored by a digital video camera connected to a computer, which controls the robotic arm, as well as providing more precise information about the location where the work piece has to be dropped in the storage area. All the activities of the cell are controlled by an intelligent multiagent system.

The multiagent systems (MAS) comprise of intelligent agents and their relationships with the environment as well as with other agents. Flexibility is the main characteristic which distinguishes an intelligent agent among other agents. Thus, if an agent is defined, "as a computer system capable of autonomous action in some environment" [5], an intelligent agent (IA) is "a computer system capable of flexible autonomous action in some environment" [6]. These definitions are the most known and most generic, but because the intelligent agent domain is a new domain and continuously developing, the definitions can differ on a case by case basis, based on the IAs used.

In this definition as in the majority of definitions, the IAs are not specifically defined as such, but rather through their characteristics. As we have seen in the cited definition, as well as in the work of Vidal [4], one major characteristic of an agent is the autonomy, which means that the agent can operate as an independent process, with no human intervention and has control over its actions and over its internal state. The other major characteristic is flexibility, which implies to be reactive, proactive and to have social abilities.

The reactive behavior is determined by the continuous interaction of the IA with the environment and the fact that the IA reacts to the changes in the environment. The proactive behaviors implies that the IA not only reacts to the changes in the environment but also accomplish goals, thus making decisions themselves. The social ability assumes that the IAs are capable of interacting with each other, in general accomplished through message exchange [1]. This ability is used by the IAs to cooperate, negotiate and coordinate one with each other such that the final task is accomplished.

Taking into considerations the IA characteristics described above, the MAS are the ideal systems to be used in the control of a flexible manufacturing cell. This flexible manufacturing cell presented in the article is controlled by a MAS with three intelligent agents. Those three agents are associated with three components of the manufacturing cell: the conveyor band, the robotic arm and the machinery as shown in figure figure 1. The three agents, through message exchange, cooperate in order to move the work piece from the conveyor band to the machinery storage area, and coordinate their behavior in such a way that they resolve the conflicts raised from the asynchronous working mode of the MAS.

2. The control system

The control system of the manufacturing cell using the multiagent system is composed of the following elements as shown in figure Fig. 1 and described below.

- Intelligent agent (A1, A2, A3);
- Digital video cameras (C1, C2);
- Conveyor band (B1) transporting bins arranged sequentially;
- Conveyor band (B2) feeding the next manufacturing cell with workpieces arranged in bins;
- Infrared sensors (S1, S2) detecting the position of conveyor band B1 bins;
- Robotic arm (R);
- Machinery (M).

The conveyor band B1 feeds the manufacturing cell with workpieces. Some of the workpieces are retained and others are sent to the conveyor band B2 which feeds other manufacturing cells. The conveyor band B1 brings workpieces which are sorted by the agent A1, and moved into the storage bin of the machinery M by the robotic arm R. The storage bin of the machinery is monitored by the agent A2. The agent A1 controls the conveyor band B1, and sends the information about the workpieces to agent A3 which with the help of the robotic arm R moves the work piece from the band into the machinery storage area. The agent A2 dictates when and where the robotic arm R drops the work piece picked up from the conveyor band.

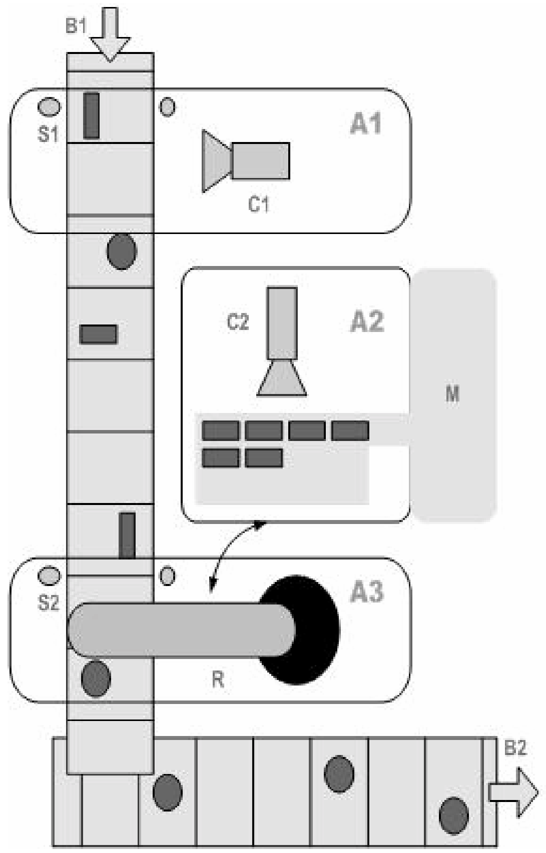

Figure 1: Flexible manufacturing cell controlled by a multiagent system.

2.1. Agent A1

The conveyor band B1 has length_B1+1 bins. In each bin there is only one work piece (object). Each bin is numbered between 0 and length_B1. Each bin represents a state of the environment which is detected by the function read_image_C1() which acquires images of the conveyor band monitored by the camera C1. The image is stored in memory at the address img1, along with the bin number bin1 in the queue1 queue using the function image(bin1, img1).

The A1 agent processes images of the conveyor band B1 placing information into the queue1 queue such as if in front of the agent A3 there are work pieces that are needed or not. The "raw" images having the addresses placed into the queue1 queue, are processed first by the detect_object(img1) procedure which detects if in the bin1 from img1 there is a work piece. This kind of detection can be done very fast using the binary image with a threshold obtained from a histogram analysis [2], [3]. Based on this detection, agent A1 will perform a reactive action against the environment by executing plan0() procedure, or it will jump to an internal level which does not imply not only reactive action but also proactive action by sending information to agent A3 to pick-up the work piece from the conveyor band if the work piece is the one that the agent is looking for. Thus, the intelligent agent has a vertically layered architecture, with the first layer being reactive – plan0(), and the second layer being the planning layer – plan1().

The control system procedures are executed in a multithreading environment. Thus, procedure prog2() executes quasi-simultaneously with procedure plan0() or immediately after plan1(). The A1 agent, through prog2() procedure, controls the conveyor band B1 . Processing a "raw" image is done by prog2() as well, by calling read_image_C1() function. The image is stored in memory and can be retrieved for later processing from the address img2.

A work piece is picked-up from the conveyor band if the first element from queue2 has the work piece characteristics that one is looking for. If the work piece is not the one that is needed, then a new image is retrieved from the queue.

The next procedure is called only when there is no work piece on the conveyor band.

```
procedure plan0()
  add(queue2,object(bin1,
   object_not_found));
end /* procedure plan0() */
```

The objective of plan1() procedure is to recognize the work piece from the image acquired and processed, and, depending on the result, to determine the behavior of A3.

```
procedure plan1()
 (bin,img) <- first(queue1);
 (bin,object_R,[+other
  characteristics])<-
  <- recognize_object(img);
 remove(queue1);
 /* wp1 – workpiece1 which is used by
 the machinery */
 if ( object_R != wp1 ) then
   *) execute plan0();
 else
   add(queue2, object_found);
   add(queue3, (bin,onject_R, …));
 end if
end /* procedure plan1() */
```

2.2. Agent A2

The agent processes the signals from the digital video camera C2 and controls the time and location of the work piece placement, into the machinery's storage area, by the robotic arm.

2.3. Agent A3

This agent is responsible for controlling the robotic arm to pick-up the work piece from the conveyor band B1. The robotic arm R, which picks-up the workpieces from the conveyor band B1 when the agent A3 tells it, has a command panel connected to a computer. The computer sends a program to be executed by the robot after which the command panel will send back a message acknowledging the

execution of the program. The program sent to the robot contains simple movements of the robotic arm. By combining the simple arm movements, the robot can move the arm and pick-up objects with fairly accurate precision.
There are two operations that the robotic arm has to perform, first, pick-up a work piece from the conveyor band B1, and second, drop-off the work piece picked-up into the storage area of the machinery served.
The pick-up operation is controlled by the following procedure accomplished by A3 agent.

```
procedure move_pick-up_position()
  *) tell R to move to the pick-up
  position;
  *) receive acknowledgment when R
  reached the pick-up_position;
  *) receive acknowledgment from A1
  that workpiece1 is on conveyor band
  B1;
  *) compare and adjust the parameters
of workpieceCrt from binCrt depending
  on the actual position of the bin;
    *) pick-up the work piece
  workpieceCrt;
  *) tell R to move to the drop-off
  position;
  *) send message to A1 that R has
  picked-up the work piece;
  *) execute move_drop-off_position();
end
```

It can be seen that the A3 agent tells the robotic arm R to move to two fixed positions (locations), pick-up_position and drop-off_position, respectively.
The next procedure shows how the move of the robotic arm into the second position is accomplished.

```
procedure move_drop-off_position()
  *) receive acknowledgment when R
  reached the drop-off_position;
  *) send message to A2 that the work
  piece is at drop-off_position;
  *) receive message from A2 about the
  exact position where the work piece
  has to be dropped-off into the
  machinery storage area;
  *) tell the robotic arm to move and
  drop-off the work piece at the
  position (location) specified in the
  received message;
  *) receive acknowledgement from R
  that the work piece has been dropped-
  off into the machinery storage area;
  *) execute move_pick-up_position();
end
```

In both positions (locations) the robotic arm will receive messages containing information about the work piece position and orientation relative to the bin on the conveyor band and conveyor band speed, and, the position where to drop-off the work piece.
Analyzing the algorithmic complexity of the algorithms used in the control system, it may be stated that all have a polynomial time and space complexity. The image processing needs a higher computing cost. The developed algorithms have been designed to be used in a distributed and autonomous system of multiagents.

3. Conclusions

This project presents the main algorithms for a distributed solution for controlling a flexible manufacturing cell. The algorithm that forms the basis of the asynchronous multiagent system was developed first, where the agents execute asynchronously and concurrent without being globally controlled. This approach was refined later, in order to accomplish the needed performance for this kind of application. The main control procedures, implemented in each intelligent agent, were presented above.
It has been observed that the algorithm performance decreases when running on a network of computers due to the time lag introduced by the network itself. The performance degradation due to the network is compensated by the fact that the agents can be executed in parallel, thus the global performance of the system not being greatly affected. Not only that, but through careful thread handling and scheduling, the CPU cycles not utilized by the agents (e.g. when the agents are waiting for acknowledgement messages) can be used to accomplish other specific tasks.
The polynomial complexity of all designed algorithms makes them suitable to be integrated in a real-time control system of a manufacturing cell. The next step in the multiagent development is to implement compensation processes for the exception situations that may occur in a production environment, such as agent unavailability due to software or hardware failure. By building event persistence and recovery at the global control level, coupled with redundant agents should be able to avoid those kinds of situations..

4. References

[1] MAS (2004). Multiagent Systems. See http://multiagent.com

[2] Mihai Horia Zaharia, Florin Leon, „A New Approach in Workflow Modeling Using Intelligent Agents", Advances in Electrical and Computer Engineering, Suceava, Romania, ISSN 1582-7445 - No 2 pp. 57-60, 2005

[3] Pentiuc St.Gh. et al., „Mathematical Models for a Robot Arm Control System", Advances in Electrical and Computer Engineering, Suceava, Romania, ISSN 1582-7445 - No 1, pp. 91-95, 2005

[4] [4] Letia, I.A., Groza, A., "Using Agent Similarities In Business Rules For The Supply Chain", Advances in Electrical and Computer Engineering, Suceava, Romania ISSN 1582-7445 - No 1 / 2004, pp. 49-56, 2004

[5] [5] Vidal, J.M., "Distributed Artificial Intelligence and Multiagent Systems". See http://jmvidal.cse.sc.edu/ 822/winter99/ #tth_sEc1

[6] Weiß, G. (Ed.), "Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence", MIT Press, Cambridge, MA., 2000

[7] Wooldridge, M., "An Introduction to MultiAgent Systems", John Wiley & Sons, West Sussex.MAS, 2004.