IEPM'93, International Conference on Industrial Engineering and Production Management, Mons, Belgium, 2-4 June 1993

Real time scheduling of a production cell based on a multi-agent system.

Christophe SOHIER[a] and Pierre BOURDET[a]

[a]Laboratoire Universitaire de Recherche en Production Automatisée (LURPA), Ecole Normale Supérieure de Cachan, 61 Avenue de Président Wilson - 94235 Cachan cedex - FRANCE

Abstract

In this paper, we propose a reliable and reactive cell controller for today's «adaptatif» workshop, using the client/server paradigm. The cell controller is composed of three main functions: a scheduller, a driver and an information collector. A cell controller is responsible for integrating the functions of the equipment controllers, each of which has its own logic. The modeling of the global behavior of the controller is based on Distributed Artificial Intelligence (DAI) and a multi-agent system. The overall behavior is the result of the agents' interactions and the system's environment. Reliable behavior components and a multi-level checking system assure system reliability. An action is taken only if its execution is possible, and it is executed only if all conditions for its success are satisfied. A distribution of decision centers to the agents results in a system that reacts to various situations. The cell itself does not plan any actions. Rather its environment and behavior manage the cell's actions according to the services that must be satisfied.

1. THE PROBLEM

We propose a reactive cell controller for a Flexible Manufacturing System (FMS), using the client/server paradigm. This paradigm is chosen as a result of procedures communicated by requirements statement and by the decentralized hierarchical organization of our workshop, which allows reactions to know information.This production activity is classified in real time process control [1].

This type of system is well-adapted to the simultaneous production of various small product series. In addition, there is no connection between the physical structure of the system and the manufacturing process.

Due to the complexity of the physical structure of these cells, the first step in the solution is to describe how the system functions. It contains a group of tasks [2] or independent behaviors with their own conditions for activation and validation. These conditions vary according to the specific equipment (Programmable Controller, Computer Numerical Controller, Robot Controller...). Each behavior, therefore, is reliable because it applies to only one sub-system and a small number of events.

Both conception and implementation are improved because each phase is treated independently. The global behavior of the system is prescribed by a judicious ordering of tasks, which takes into account the requested production and occurrences in the system. These occurrences can be influenced by the system's normal functioning or by internal or external disturbances to the cell.

The internal disturbances are essentially machine faults, such as equipment breakdown, tool breakage, etc… External disturbances are, for example, changes in production objectives, supply problems, or a breakdown of service cells. Very often, these disturbances require a stut-down and a complete reevaluation of the manufacturing process.

One way to obtain a reactive system is to delay the decision-making process until the last moment, in order to obtain information as close as possible to a real state. This, however, necessitates real-time construction of the system's sequencing. To solve this problem, we propose a solution based on Distributed Artificial Intelligence (DAI) and a multi-agent system.

2. DISTRIBUTED ARTIFICIAL INTELLIGENCE AND MULTI-AGENT SYSTEMS

Distributed Artificial Intelligence and multi-agent systems distribute intelligence and decision-making among independent centers called «agents» [3]. These individual agents form groups of planning agents that work together in a common environment. This interaction allows them to solve complex problems, which could not be solved by individual agents [4]. This distribution of expertise can be directed by a material and/or a functional approach.

We propose an agent model based on social behavior (negotiations, service demands), that possesses biological reactions (aggression, fleeing, satisfaction).

An agent does not have a global view of the system; it has only a partial picture of its environment, obtained through its sensors and its interactions with neighboring agents. It does have, however, its own local memory and can send messages to other agents.

The agent has three distinct behaviors: the will to be satisfied; the obligation to flee, and the will to be free [5].

The will to be satisfied is the will to reach a goal. If an given agent does not have the necessary competence to resolve a problem, its desire for satisfaction leads it to seek the services of its neighboring agents. An agent can also create other agents to help with problem-solving.

Fleeing is a reaction to aggression from another agent. An agent flees its position in its environment to suppress or diminish conflicts with other agents. This behavior, however, destroys the agent's satisfaction, which leads it to negotiate with other agents and even to create new agents.

Freedom state for an agent is when there are no conflicts between this agent and others. In particular, an agent must be free before taking any action. An agent does not act for freedom, but tells agents that prevent its freedom state to act-i.e to flee.

Finally, an agent acts only when he gets either the will to be satisfied or the obligation to flee. Agents continue to act until a stable state, corresponding to their actions, is attained.

The global state of an agent can be defined by the triplet S, F and L where:

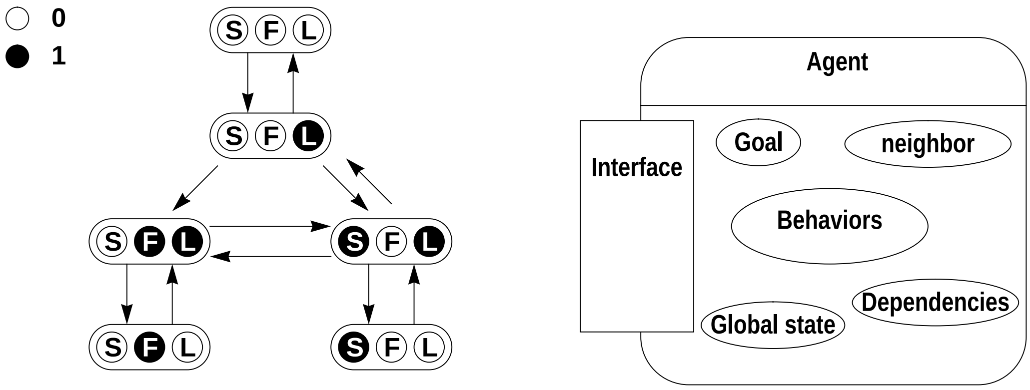

Figure 1. Global State and Graphic Representation of an Agent.

- S is the state of satisfaction corresponding to the desire for satisfaction. S=0, the agent is not satisfied; S=1, the agent is satisfied.
- F is the fleeing state corresponding to the necessity to flee. F=0, the agent is not fleeing; F=1, the agent is fleeing.
- L is the liberty state corresponding to the desire to be free. L=0, the agent is not free; L=1, the agent is free.

An agent cannot be satisfied and fleeing at the same time. Therefore, of the eight possible global states, only six can occur.

A master/slave relationship exists among the agents because of their dependency on one another. An agent is a slave because its satisfaction depends on the group's satisfaction.

The agent's goal is set at the time of its creation, but it can be modified depending on the services needed. In this way, the agent controls its behavior and its interactions with its environment.

3. MODELING THE CELL CONTROLLER

We describe in this section a host structure which accepts our multi-agent system. Such a structure has been designed and constructed in our laboratory, and has been tested by a logistic cell controller [6] on a global approach. It consists of a generic control structure [7] composed of three functional blocks: a scheduler, a driver and an information collector.

The Scheduler

The scheduler receives a predicted scenario of production requests, which describes the future needs of production, but not its organization. The necessary production resources (cutting tools, fixture, gripper...) and the components used

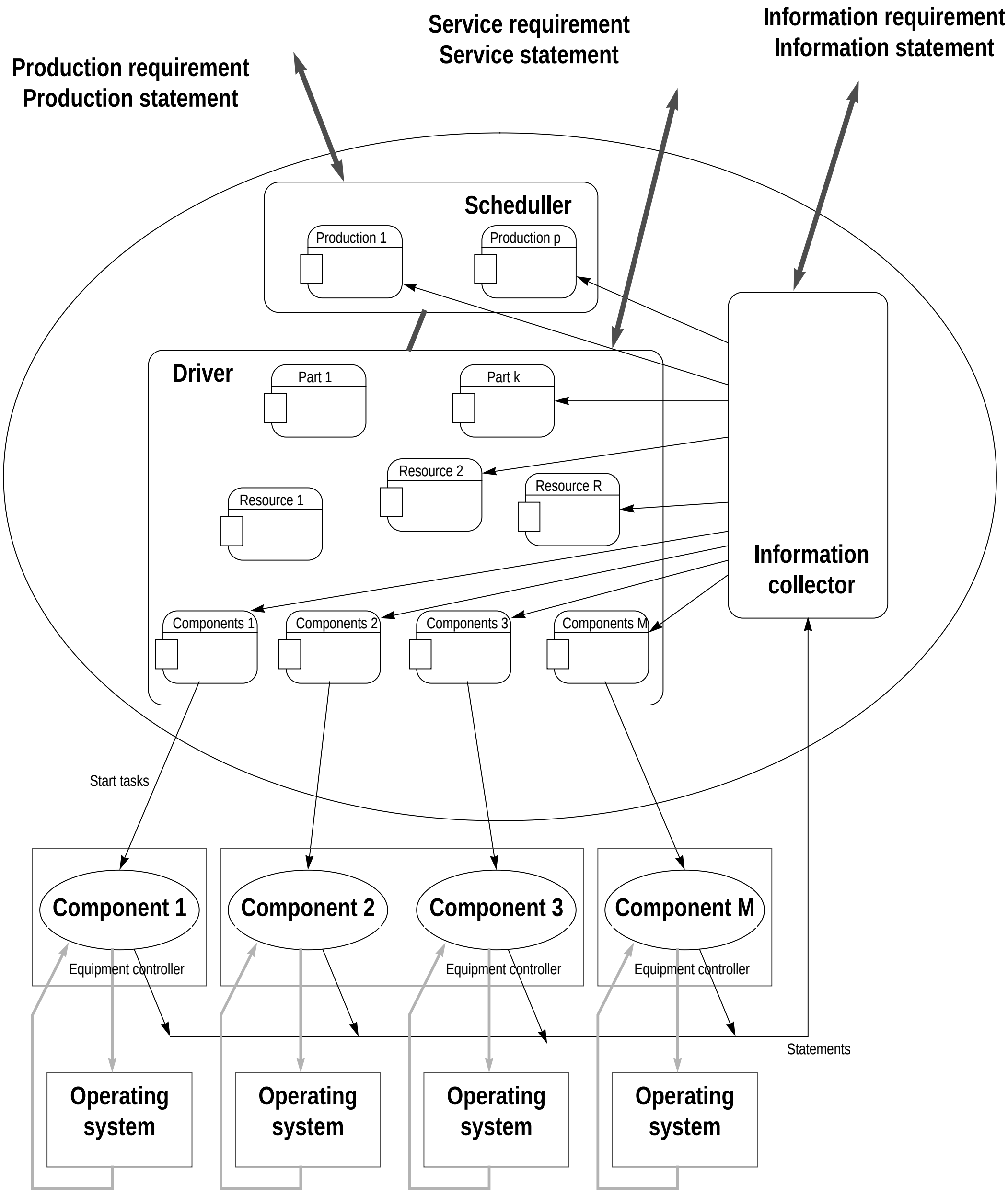

Figure 2. Functional Structure of the Cell Controller.

(machine, robot, pick up and deposit station, storage carousel...) are also given. For example, one production might use a machine, a part batch on a tray, a manipulator robot and frequent of tool changes. To create these scenarios [8], it is necessary to know external controller criteria, such as the optimal cutting conditions according to the size of the batchs; the availability of external resources; and the optimization of the external product flow. The necessary conditions to begin a production include the availability of products, resources and components. The acceptance of a production is determined by the availability of the means needed for its proper progression.

The Driver

The driver's function is to organize how the components and resources are used, according to the needs of production and of states of operating systems. It also drives and coordinates the different elements of the operating system. At this level, the components are chosen according to their availability.

The Information Collector

The information collector's function is to collect and organize the information needed to operate the controller. It also sends information to the outside from the cell, (supervision, maintenance, quality control, planning...) and represents the cell's sensor group.

In this paper, we study the modeling of the scheduler and the driver; the information collector is not presented.

Our method is the following:

- identification of possible production types and their association to agents modeled by the scheduler.
- identification of the components, resources and products manipulated by the cell and the association of each one to the agent modeled by the driver.
- construction of interfaces and behaviors according to the services rendered by each agent.
- identification of functional agents capable of sub-treating specific work.

The behavior and goals of each agent are dependent on the system and on the modeled entities. We can use libraries of agents to adapt behaviors to the studied systems. The linkage and the dependencies - which can be static or dynamic - are determined by the internal organization of the cell.

A type-part agent negotiates its placement near a fixture with a robot-type agent. If the agent is rejected or if the robot is not available, the agent is capable of negotiating with another agent (an operator, for example) that can offer the same service.

The system's ability to react is due to the distribution of the decision centers. Each agent is capable of handling its own local events and of modifying its behavior. Disturbances can, therefore, be treated as natural system events rather than abnormal occurrences.

4. APPLICATION OF THE EXPERIMENTAL CELL CU3L

The experimental milling CU3L cell (fig. 3) is a component of LURPA' s Flexible Manufacturing System. At present, the system is constructed around a generic controller, but the production sequence is generated by fixed GRAFCET [9] scenarios. We are in the process of applying a multi-agent approach to the CU3L cell. Here are some first results.

The cell is composed of the following elements:

- components: a milling machine, an anthropomorphic robot, three pick up and deposit stations (Pl0,Pll, and Pl2) and a storage carousel at two sites.
- resources: fixtures, cutting tools, tool containers, grippers, trays, and parts containers.

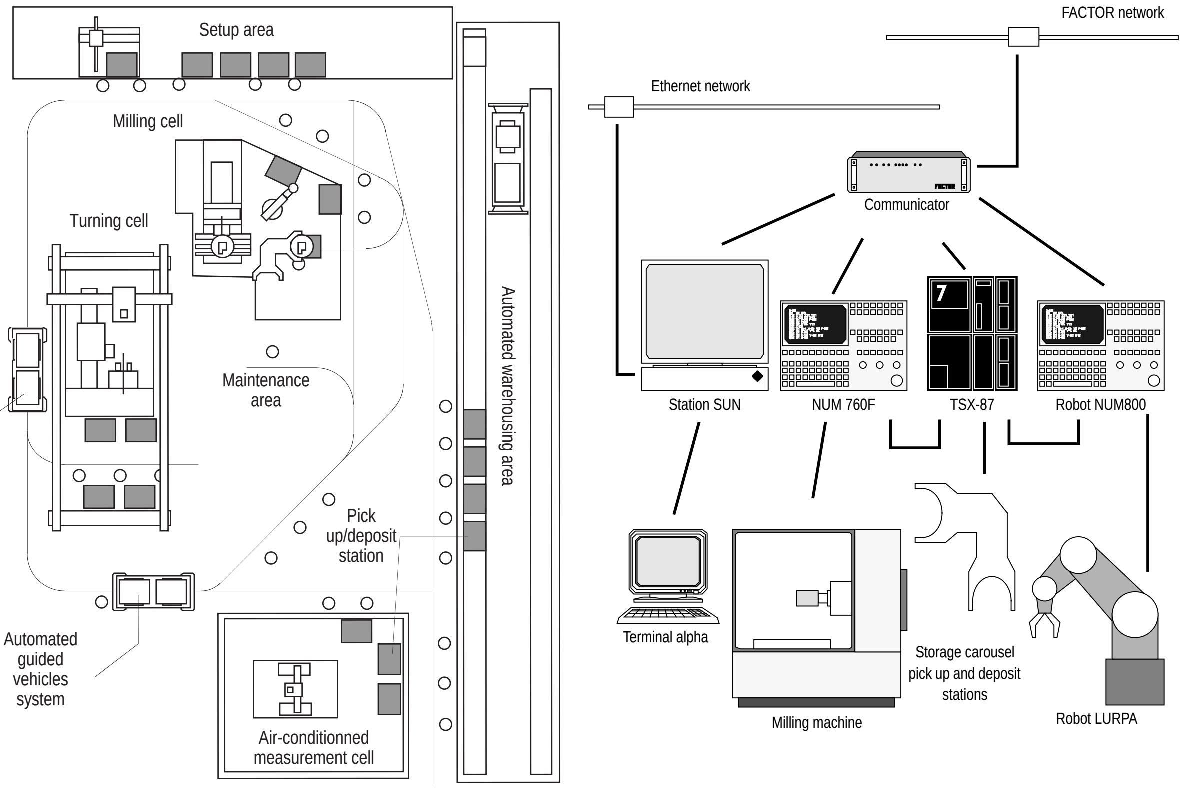

Figure 3. Experimental Site (LURPA' s Flexible Manufacturing System) and the Physical Architecture of CU3L.

The milling cell has a logistic service cell consisting of two transport robots (AGVS), which supply the fixture, gripper, cutting tools and parts batch.

The available combinations of cells give six different production strategies (with robot, non-robot, one or two pick up and deposit stations…). Two available sites on the storage carousel (two fixtures) allow the cell to handle a maximum of two different productions simultaneously.

In applying this method, we obtain the following minimal structure (fig. 4) (the population of all agents is not represented).

We explain the cell's function by describing the system's behavior during the execution of a production requirement.

The first step is to verify the feasibility of the requirement. This is done by a «production 1» agent, that negotiates with the components involved in the requirement. Those components that cannot be divided are set aside and a statement of the acceptance of the requirement is given. A three-stage elementary plan is applied: installation, execution and de-installation of the production.

Three agents designed to handle each of these steps are created, and dependency relationships are established in order to respect the sequencing (fig. 5).

The «install production 1» agent determines the resources needed to start production, and creates corresponding agents with their own goals and dependencies.

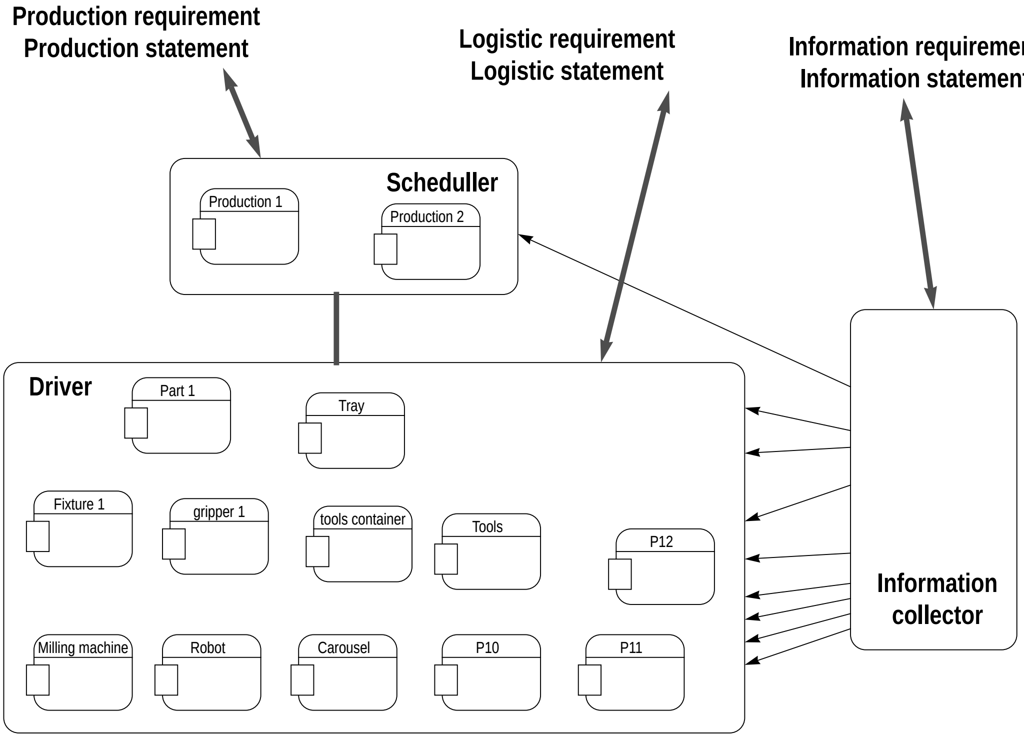

Figure 4. Functional Structure of the Cell CU3L Controller.

The satisfaction of fixture 1 depends on the satisfaction of the tools. Therefore, the tools must be brought to the cell before fixture 1 (use of the same flow as in pick up and deposit station Pl2).

The tools 1, which depend on fixture 1, are positioned on a tool container. Their goal is to line up in the tool magazine. The tool support, which depends on the tools, is brought to tray 1 to be emptied. The goal of tray 1 is to place itself on Pl2.

Tray 1 sends a requirement to the logistic cell requesting supplies for station P12. In return, the logistic system informs tray l of the action taken. Next, the tray negotiates with the different agents involved in the production process. As a result, the different agents operating at the time of the production are locked (not free) in place. If the trapping operation proceeds correctly, tray l is placed on station Pl2, and the tool container rests on one site of the carousel.

The satisfaction of tray l results in the installation of the tools in the machine magazine. Once the tools are satisfied, the emptying of the tool support and of tray l is demanded. Fixture 1 creates tray 2. Its goal is to be brought to station Pl2. Two situations can then arise: l) there is a free site on the storage carousel, which presents no problem since tray 2 can be led to the cell. 2) there is not an empty site on the storage carousel (it is occupied by another production, for example), in which case, fixture 1 will attack the tool container.

The reaction of the tool container is to find a free site and to flee to it. There are two ways to flee: occupy the milling machine or be emptied from the cell. However, only one of these possibilities satisfies the goal of the tool container, and will,

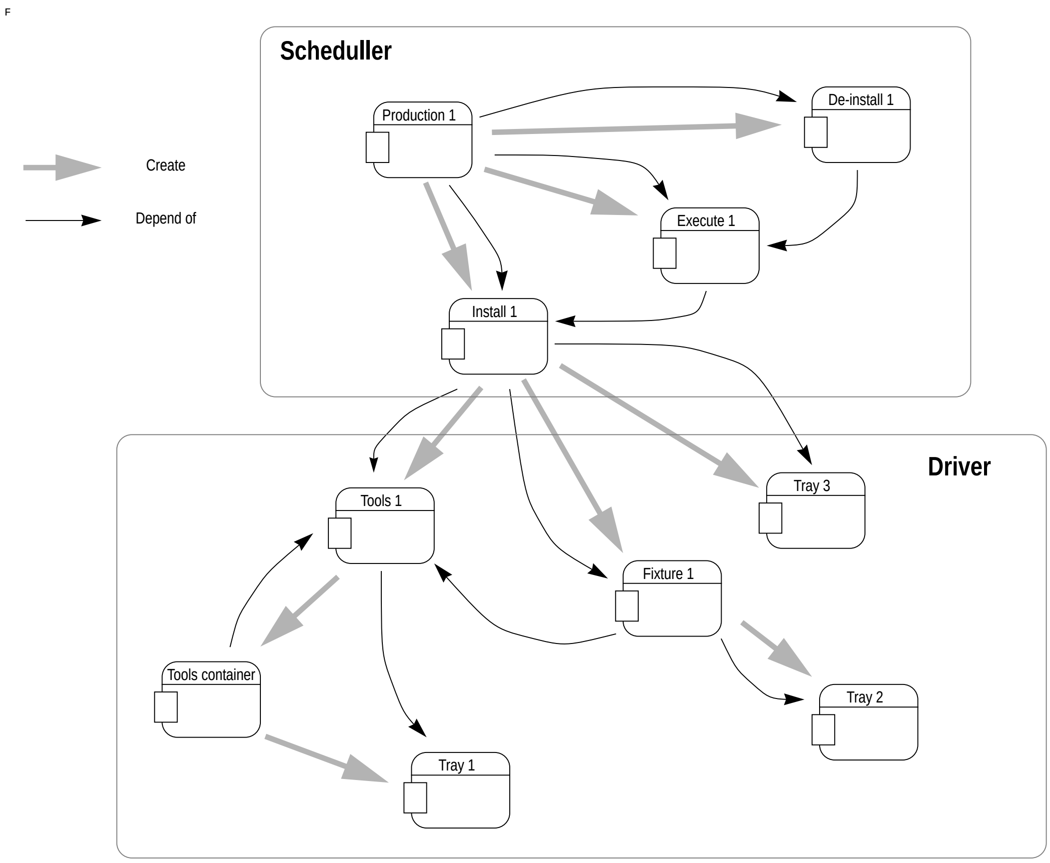

Figure 5. Creation and dependence among agents at the beginning of a production.

therefore, be chosen: the tool support empties and brings fixture 1 with it. Similarly, the tray 3 parts container is led to station Pl0.

At each step of the process, agents that are no longer useful remove themselves from the environment.

The «install production 1» agent is satisfied. The «execute production 1» agent, in its attempts to be satisfied, creates a group of agent parts whose goal is to be worked on the tray 3. Dependency relationships are established among the parts so that working order is respected.

Production 1 is satisfactorily executed when the ensemble of parts are satisfied. The «de-install production 1» agent then seeks satisfaction, leading to the emptying of the resource group, which is necessary for production.

Once the production 1 agent is satisfied, a statement of the end of the production is produced. This ends «production 1».

5. IMPLEMENTATION

The implementation of the component behaviors and a minimal communication layer are programmed to the specific language of each equipment controller

(GRAFCET, ladder, CNC program…). The multi-agent system is implemented in a UNIX workstation, and we use ada language [10] for programming. Each agent corresponds to an ada task; the concurrent execution is performed by ada tasking; and the communications and synchronizations are performed by the mailbox abstraction and ada «rendezvous».

6. CONCLUSION

Our first experimental results show that the proposed multi-agent approach provides a solution to driving complex automated systems. It reacts in real time to automated system events and generates a coherent plan for tasks as they appear in the system. Because of the interaction of equipment and the sharing of services, the system also provides a solution to flexible systems with a large number of scenarios. The structure permits us to add agents capable of resolving specific problems. The synchronization of agents assures that the system functions correctly.

7. REFERENCES

1 BROWNE J., Production Activity Control - a key aspect of production control, International Journal of Production Research, Vol.26 - N°3, pp. 415-427, 1988.

2 GENDREAU D., Génération automatique des procédures de pilotage d'une cellule flexible de production, Thèse de doctorat de l'Ecole Centrale de Paris, France, Décembre 1991.

3 ERCEAU J., FERBER J., L'intelligence artificielle distribuée, La recherche, Vol.22, pp. 750-758, Juin 1991.

4 FERBER J., Eco problem solving: How to solve a problem by interactions, Proceeding of the Ninth Workshop on Distributed Artificial Intelligence, pp. 113-128, Rosario Resort, Eastsound, Washington, September 1989.

5 FERBER J., JACOPIN E., A Multi-Agent Satisfaction Planner for Building Plans as Side Effects, LAFORIA Report, July 1990.

6 LESAGE J.-J., TIMON G., An extension of the Production Management Concepts towards the Real Time Cell Production Control, the 4th IFIP CAPES, Bordeaux, France, September 1991.

7 BOURDET P., GENDREAU D., KIEFER F., LESAGE J.-J., TIMON G., A new approach to cell automatic supervision in F.M.S., 3rd CIRP Int. Conf. on Automatic Supervision, Monitoring and Adaptative Control in Manufacturing, Rydzyna, Poland, September 1990.

8 GENDREAU D., LESAGE J.J., TIMON G., An integration of production management rules and fabrication know-how for real time cell production control, 5th International Conference on Manufacturing Science and Technologie of the Future, Enschede, the Netherlands, June 1991.

9 Norme Française, Diagramme fonctionnel GRAFCET pour la description des systèmes logiques de commande, Norme Française NF C03-190, juin 1982.

10 Ada Programming Language, Department of Defense, Washington, D.C., ANSI/MIL-STD-1815A-1983.