Science and Technology, 7 (2002) 177-185

Multi-Agents Architecture for Scheduling and Control of Robotic Manufacturing Systems

Brahim Bouzouia

Centre de Development des Technologies Avancees. 128 Mohamed Gacem, El-Madania, Algiers (Algeria). Email: bbouzouia@cdta.dz.

هيكلة متعددة الأوجه للتخطيط والتحكم الآلي في أنظمة الانتاج اللينة

ابراهيم بوزوية

خلاصة: تتضمن هذه الورقة منهجية متعددة الأوجه للتحكم الآلي في أنظمة الانتاج اللينة. تعتمد هيكلة المنهجية المقترحة على تقنية الذكاء الاصطناعي الموزع والأنظمة المتعددة الأعضاء. يحتوي النظام المتوخى عن هذه المنهجية على الصلاحيات الآتية: التخطيط،التوزيع، المتابعة والصيانة. نقترح كذلك خطة تفاوض جديدة لأنظمة الإنتاج. الهدف من هذه الخطة هو التوزيع الحيوي لموارد نظام الإنتاج لغرض تأدية العمل المبرمج.

ABSTRACT: This paper presents an innovative Multi-Agent approach related to an advanced real time control of flexible manufacturing systems. The proposed architecture is based on the paradigm of Distributed Intelligence and Multi-Agent Systems. The developed Multi-Agent prototype system integrates the following functions: Scheduling, dispatching, monitoring and error handling. A new negotiation protocol for manufacturing systems is presented in this paper. The purpose of this protocol is to assign dynamic operations to the resources of the Manufacturing System in order to accomplish the proposed tasks. This protocol is able to deal with exceptions.

KEYWORDS: Multi-Agent Systems, Flexible Manufacturing Systems, Distributed Artificial Intelligence, Resource Allocation, Real Time Scheduling, and Renegotiation Phase.

1. Introduction

Global competitiveness has been imposing important changes to all components of manufacturing systems, including the real time scheduling. From an emphasis on the scheduling optimality in the past, the focus has now moved to scheduling flexibility (Trenteseaux, 1996).

The function of a real time manufacturing cell control system can vary depending on the size of a cell, its type and the degree of decision-making capabilities. The major functions of a cell control system include the need to schedule and monitor cell resources, and the ability to react to abnormal conditions or exceptions (Ouelhadj *et al.* 1999 ; Ouelhadj *et al.* 1998).

The major approaches of scheduling are represented by the traditional operational research techniques as combinatorial procedures, heuristic approaches and constraint analysis (Balasubramania and Norrie, 1997; Ferber and Ghallab, 1999). Nevertheless, these approaches are NP-complete and has shown many limitations. Such systems soon became too heavy to be implemented as a centralized control program in only one computer. Fortunately, distributed computing has already given some answers to the problem of how to implement efficiently communities of interactive systems. This emerging area is Distributed Artificial Intelligence (DAI) (Ferber and Ghallab, 1999; Ramos, 1994; Ferber, 1995). The framework of DAI, particularly Multi-Agent System (MAS), seams more suitable for the dynamic control of manufacturing systems. The Multi-Agent System paradigm represents one of the most promising approaches to build complex and flexible advanced intelligent systems. The application of MAS approach in dynamic scheduling and control is based on the idea that the scheduling and control agility can be extremely improved once these are based on the following key points: (i) distributed and autonomous systems instead of centralized and non-autonomous solutions; (ii) negotiation-based decision making instead of the totally pre-planned processes; (iii) uses of different problem-solvers in the same environment instead of only one fixed problem solver; and (iv) concurrent execution instead of sequential processing.

Several works on flexible manufacturing system control followed the paradigm of DAI. Here we will refer only to two of these systems: ICOSS and MASSYVE. ICOSS (Intelligent Cell Objects/Intelligent Supporting Shell, Lee and Sen, 1994) architecture consists of two layers: the inner layer, called ICO, contains cell control knowledge that perform generic cell control functions such as scheduling, dispatching, monitoring; the outer layer called ISS contains cell databases that represent the specific cell environment. The two layer approach allows more efficiency in developing and implementing intelligent cell control by considering generic cell control knowledge and cell-specified databases separately. MASSYVE (Multi-Agent Agile manufacturing Scheduling Systems for Virtual Enterprises, Rabelo, 1997) aims to investigate the use of multi-agent systems in agile scheduling, towards the operation in a virtual enterprise environment. These activities use the HOLOS framework as base line for advanced scheduling. The information integration approach supporting a multi-agent system in MASSYVE is based on the PEER information management framework.

This paper presents a new Architecture and Negotiation Protocol (**JT**ransactions **M**achines) for an intelligent cell control system. We consider that requests involve due dates (deadlines) for the tasks to be carried out and some applications of this framework to Distributed Robotic Systems. Section 2 presents a new architecture for real time scheduling of Manufacturing Systems. Section 3 and 4 illustrate how the negotiation/renegotiation protocol is established. Finally, conclusions are presented in section 5.

2. A New Architecture for Dynamic Scheduling and Control of FMS

An automated manufacturing cell typically consists of numerical control machines, industrial robots, storage devices, automatic inspection devices, tools and fixtures and control computers. The machine tools in the cell are physically interconnected by automated material handling devices such as conveyors, AGVs and robots. Communication networks provide the information links within the cell.

Figure1 illustrates the Architecture proposed in this paper. This architecture includes, essentially, two parts : one concerning resources and another concerning agents. The number of resources does not vary, except when resources are introduced or removed from the Manufacturing System.

The system architecture contains the following agents:

Initiator Agent : interfaces with the user receiving orders of new tasks for the Manufacturing System. This agent is responsible for launching Task Announcement Message whenever a new task is ordered (starts the execution of scheduling process). It also maintains a global view on the activity of the resources (Mirrors Agents).

Mirror Agent (MA): an autonomous agent called Mirror Agent represents every resource in the cell. MA_i represents the current situation of resource i (its status and activity). The activity is a sequence of operation to be carried out which is represented as an agenda. This agent is responsible to perform the four control functions, which are : scheduling, dispatching, monitoring and error handling.

For a better output, two production targets direct the control system:

- Reduction of the number of late tasks : The initiator agent will start by seeking the tasks in the following order: tasks for rescheduling, urgent tasks and finally shortest tasks.

- Manufacturing products at the earliest: The mirror agent contract an operation that offers the completion date nearest to the current date.

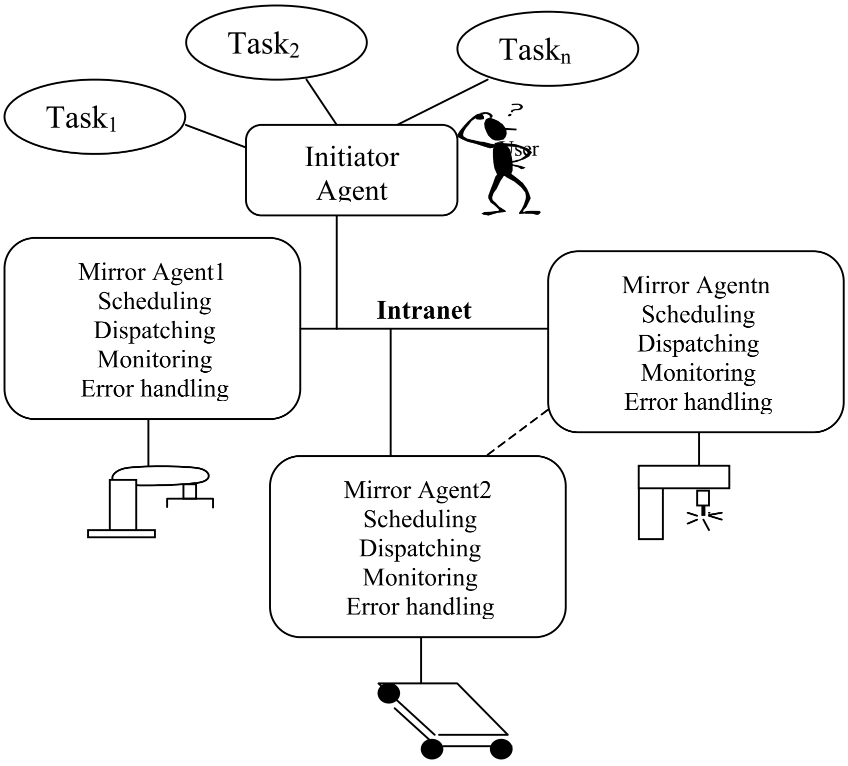
Task$_2$
Task$_n$
Task$_1$
Initiator Agent
User
Mirror Agent1 Scheduling Dispatching Monitoring Error handling
Intranet
Mirror Agentn Scheduling Dispatching Monitoring Error handling
Mirror Agent2 Scheduling Dispatching Monitoring Error handling

Figure 1. The proposed Architecture.

3. Negotiation Protocol (JTransactions Machines, JT.M)

Let us consider the example of Figure 2, which consists to produce **Q** parts **XY**. The part must be polish on front and drill on sides in a specific deadline **DL**.

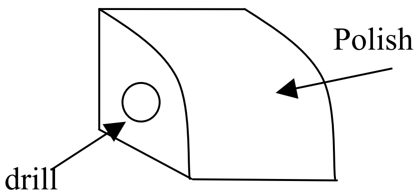

Figure 2. List of operations for the XY part.

Feed_drilling_machine(xy) AND
opr 1 - drill(xy) → Xy part.
Feed_turning_machine(Xy) AND
opr 2 - polish(Xy) → XY part AND
opr 3 - transport(XY)

The JT.M protocol is based on the Contract-Net protocol proposed in (Reid, 1992). However, this protocol improves the basic behaviour of Contract-Net by the distribution of control functions over all the agents of the system. The Figures 3 and 4 illustrate the Negotiation Protocol contracting resources for execution of task. When a new request for task execution appears, the user specifies the name of the task to be carried out as well as its deadline. In order to guarantee the deadline, the negotiation between the Mirror Agents is performed in backward chaining. The negotiation can be summarised in the following steps :

Note: All messages supported by the control system contains the fields: {SEN, ADR} where SEN is a sender of the message and ADR is the addressee.

Step one

The Initiator Agent extracts the last operation of the task and using knowledge about the material configuration, constructs the list of possible resources for the operation (Figure 3). The Mirror Agents corresponding to this operation receive a message with the following format :

{CD, TD, Q, DL, LOC}

where : CD is customer identifier, TD is a task descriptor, Q is a number of operation sequences (e.g. 5 objects XY for the example of Figure 2), DL is deadline for a task delivery(date and time) and LOC is the list of operations including their constraints.
This message will be referred as the ' Task Announcement Message ' (***T.A.M***).

Step two

The Mirror Agent, that had received Announcement proceeds as follows:
It extracts all the operations which are concerned with the task (operations are considered from last, last but one,...). After that, the Mirror Agent tests the possibility to carry out at least the last operation before a deadline DL. If so, it sends 'Request Messages' (***R.M***) on the (N–k), where k is a number of operations extracted by Mirror Agent, operations corresponding to the sub-hierarchy of operations (Figure 3). The propagation of requests is reiterated until there is no operation to negotiate. The Request Message has the following format :

{CD, TD, Q_i, DL, LOC, NLO, SD, TDRR, CO}

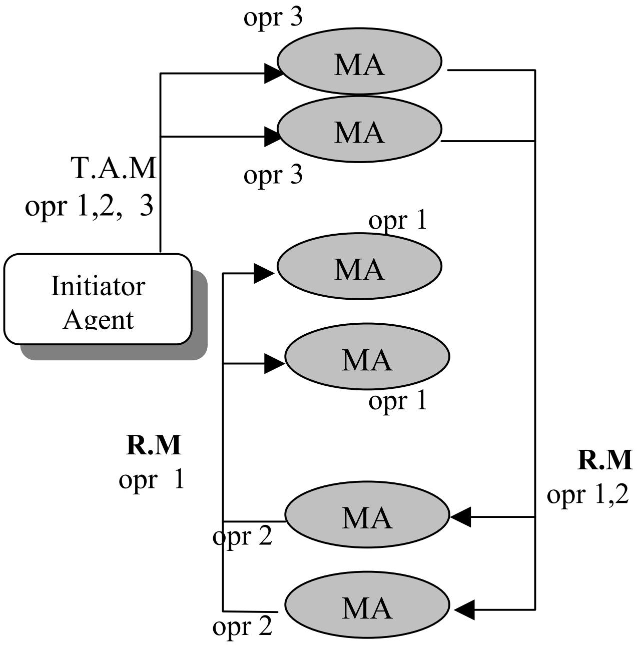

Figure 3. Task Announcement and Request Propagation.

where : Q_i is a maximum amount of operations to guarantee the deadline by sender (MA_i) of request, NLO is the list of remaining operations, SD is the time where the Q_i operations could be started, TDRR is a temporal delay required by the SEN to ADR on propagation request phase and OC are the constraints related to the operative duration preceding current request.

Step three

The Mirror Agents which receive a request for elementary operations are the first that broach intra-level negotiation (these agents control the same resources). They proceed as follows :
Each MA_i sends to its contact the information's about its capacity to make Q_i operations among the amount Q ($Q_i <= Q$) and the due date DL_i for these Q_i operations ($DL_i <= DL$). Candidates start the same algorithm in order to select the best agents. Therefore, each one obtains an identical result and knows if it participates or not at the production of XY parts(Figure 4).
Note : - The criteria for selection favours Agents that offer the earliest date of manufacturing and consider the possibility of splitting the operations by several resources ($\sum Q_i = Q$) .
- An agent can negotiate with null values if it did not receive invitations to tender concerning the task in progress.
The 'Negotiation Message'(*N.M*) has the following format :

{CD, $Q_{max(i)}$, $DL_{max(i)}$}

where : $Q_{max(i)}$ and $DL_{max(i)}$ are respectively the maximum number of operations and their deadline which **Mirror$_i$** of the last level can assure.

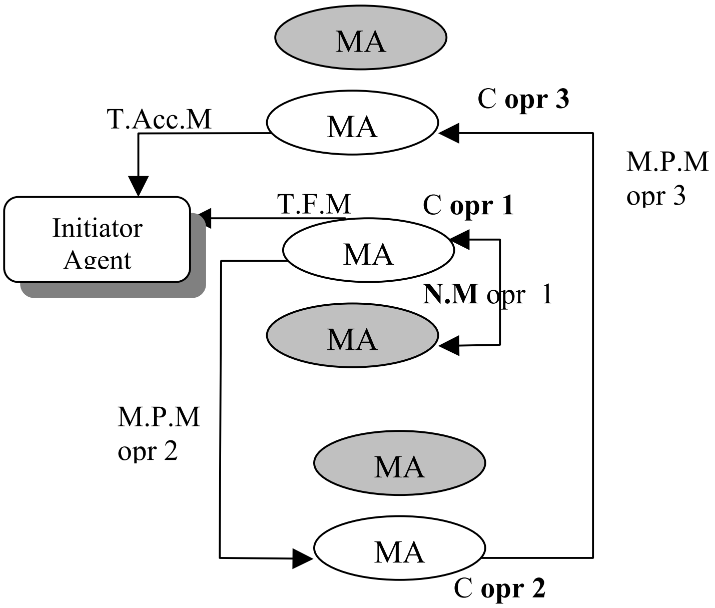

Figure 4. Negotiation, Contract and Proposition Feedback.

Step four

After that, the selected Mirror Agents establish their contracts **(*C*)** and send 'Mirror Proposition Messages' **(*M.P.M*)** to Mirrors of the up-hierarchy (Figure 4). After the reception of the proposition message, the Mirror Agent concerned by the last operation of all the XY parts, detects the end of the scheduling process and sends a 'Task Acceptance Message' **(*T.Acc.M*)** to the Initiator Agent (Figure 4). The Mirror Proposition Message has the following format : **{CD, TD, Q_i, DD, DL_i, TDRB, CO}**
where: TDRB is a temporal delay required by the SEN to ADR on feedback phase proposition. On the other hand, if Mirror Agents are not able to guarantee the deadline ($\sum Q_i < Q$) the negotiation is

finished without contract and a ‘Task Failure Message’(***T.F.M***) is sent to the Initiator Agent by Mirror which offers the best criteria in intra-level negotiation phase (Figure 4).

3.1 Dispatching, Monitoring and Error Handling

First, the module extracts a contract (operation) from its agenda and sends its preconditions to the inference engine in order to check them. The precondition operation calls the sensory equipment of the resource in order to obtain data concerning the presence of the part and its identity, and compares them with the initial status of reference. In the favourable case (no exceptions are detected after the inference), the operation is launched and a starting order is sent to the resource controller.

When the operation is finished, the resource controller sends a signal specifying the operation characteristic to the inference engine. After that, the postcondition operation is launched. This operation, contrary to the first, checks the status of the operation carried out and compares them with the final status of reference. If no exceptions appear, the dispatching module executes the next operation (Figure 5).

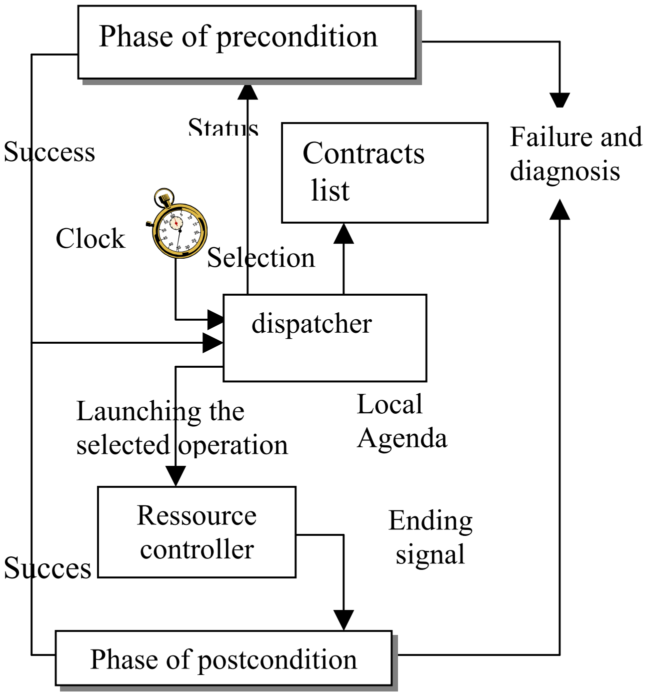

Figure 5. Operation execution mechanism.

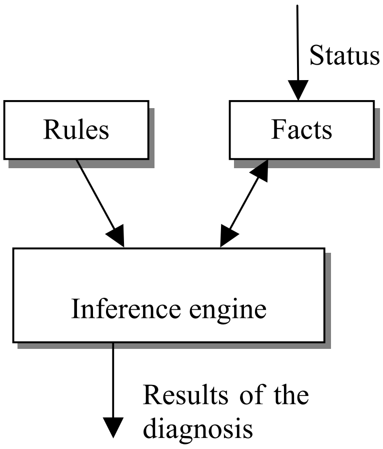

Figure 6. Diagnosis module.

3.2 The inference engine

After an abnormal event, the diagnosis module (Figure 6) proceeds by:

- Formatting facts and loading them in the facts base.
- Launching the inference.
- It sends, the results of the inference to the concerned process(communication, rescheduling, etc).

4. Exceptions and renegotiation

Once a task has been accepted it will be accomplished in the specified deadline if no exceptions appear. The Mirror Agent sends starting-orders to their appropriate resources and wait for ending signal or abnormal events. However, exceptions (e.g. machine in failure or operation has a fault) may happen and the manufacturing system control has to deal with this kind of problem.

The Mirror Agents support Monitoring functions that recognises and analyses exceptional conditions or errors of their resource, and provides a possible corrective action to these problems (renegotiation, operator call,...). When a Mirror Agent executes an operation, its sensory equipment informs the Monitoring about the status of operation, resource or parts. This information is sent to the Error handling to diagnose the errors by the use of production rules based on predicate logic formalism.

For completeness reasons, we suppose that the product XY is assigned to a productive way between Mirror Agents MA1, MA3 and MA6. The Mirror Agent related with the exception (Figure 7, MA3) establishes a list for all contracts affected by the failure. After that, it will send 'Contract Cancellations' messages to Agents appearing in the list, as well as, an 'Operation Failure' message to the Initiator Agent enclosing the amount of penalty for unaccomplished operations. Finally, it destroys the list. The Operation Failure Messages has the following format: **{CD, TD, ANO}.** Where : ANO is the amount of penalty for unaccomplished operations. The Contract Cancellations Messages has the following format : **{CD, TD, LCAF},** where : LCAF is a list of contracts affected by the failure.

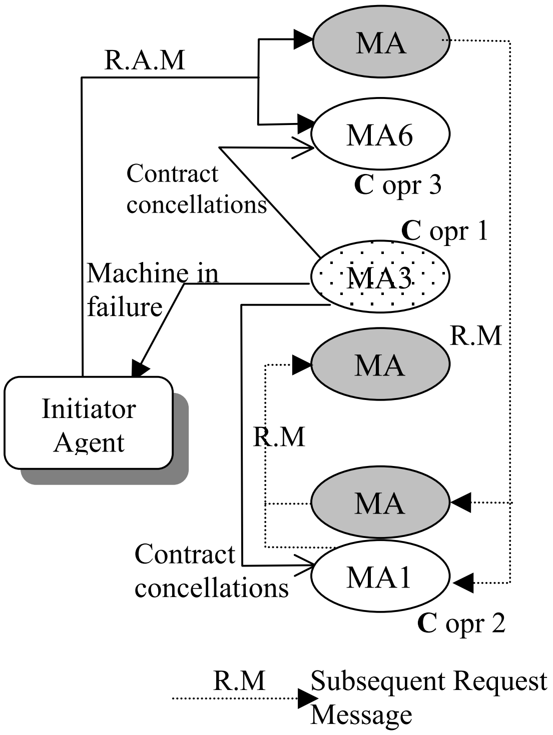

Figure 7. Renegotiation protocol.

5. Implementation Issues

5.1 Software Architecture

The system is implemented in Object Oriented Programming with JAVA2 (Sun Micro System) that allows multi-threading and supports agents executing on different platforms (Microsoft Windows 9X, UNIX, OS2). The Agents of the system use TCP/IP protocol for communication and apply a recursive algorithm for sending/receiving *Message Object.* Finally, the error handling is implemented with JESS5.0a6 (Java Expert Shell System of Sandia National Laboratories Livermore, CA).

5.2 Material Architecture

The flexible cell on which we have applied our control system is formed by a conveyor, a CNC turn machine and a handling GT6 robot. The resources are controlled by four Pentium III 600Mhz microcomputers, among them one for the operator, interconnected by a local network having a star topology (Figure 8).

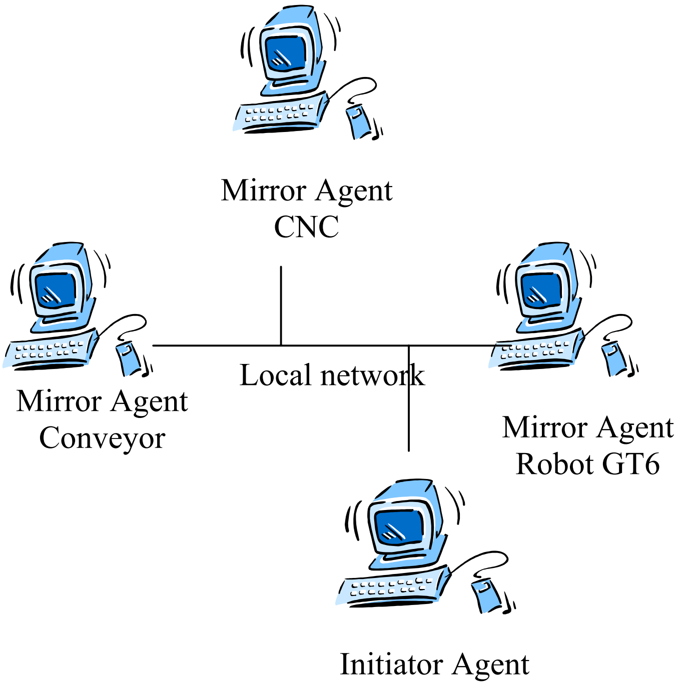

Figure 8. Material Architecture.

6. Conclusion

This paper addresses a cell dynamic scheduling and control architecture that applies a Multi-Agents approach to the problem of Robotic manufacturing systems. It has also presented a new Negotiation Protocol (**JT.M**) suitable for the dynamic scheduling of manufacturing tasks. This Negotiation Protocol is able to deal with exceptions, since a renegotiation phase of the protocol can be activated. The main advantages of this architecture are : agent autonomy, decentralised decision-making capability, flexibility in more changing environment, sophisticated communication system, increased fault-tolerance, real time behaviour, the agents are totally co-operative and an inherent reconfiguration. Control functions (scheduling, dispatching, monitoring, error handling) are totally distributed over the entire agents in the system. There are also other functions, such as: cell initialisation, communication, user interface and tools that support visualisation of agent performance.

References

BALASUBRAMANIA, S. and NORRIE, D.H. 1997. A Multi-Agent Intelligent Design System Integrating Manufacturing And Shop Floor Control, *University of Calagray, Canada, AB, T2N IN4*.

FERBER, J. 1995. Les Systèmes Multi-Agents: Vers une Intelligence Collective. n° ref. 5548.4 1174, *iia*.

FERBER, J. and GHALLAB, M.1999. Problématique des univers multi-agents intelligents. *Journées nationales PRC-GRECO. Intelligence Artificielle, Toulouse, France,* page 295-320.

LEE, K.H. and SEN, S. 1994. ICOSS : A Two-Layer. Object Based Intelligent Cell Control Architecture Computer Integrated Manufacturing Systems, 7(2) 100_112 *Department of Industrial Engineering, Northern Illinois University, DeKalb, IL 60115-2854, USA.*

OUELHADJ, D., HANACHI, C. and BOUZOUIA, B.1998. Multi-Agent system for dynamic scheduling and control in manufacturing cells. ICRA'98, *IEEE International Conference on Robotics and Automation, Louvain (Belgium).*

OUELHADI, D., HANACHI, C., BOUZOUIA, B., MOUALEK, A. and FARHI, A.1999. A multi-contract net protocol for dynamic scheduling in flexible manufacturing systems. *IEEE, International Conference on Robotics & Automation Detroit, Michigan, USA*.

RABELO, R.J. 1997. A framework for the development of manufacturing agile scheduling systems – a multi-agents approach*, Ph.D. Thesis, New University of Lisbon, Portugal*.

RAMOS, C. 1994. Architecture and Negotiation protocol for the dynamic scheduling of manufacturing systems. *Proceedings of IEEE International Conference on Robotics and Automation*, Page 8-13.

REID, G.S. 1992. The Contract-Net protocol: High level communication and control on a distributed problem solver. *IEEE Transactions on Systems. Man & Cybernetics,* Page 20-35.

TRENTESEAUX, D. 1996. Conception d'un Système de Pilotage Distribué, Supervisé et Multicritère pour les Systèmes Automatisés de Production, *Thèse de doctorat à l'Institut National Polytechnique de Grenoble, France.*

Received 4 June 2001
Accepted 26 February 2002