# A DISTRIBUTED ARCHITECTURE MODEL FOR THE DRIVING AND CONTROL OF A ROBOTIZED MANUFACTURING SYSTEM

Z. TOUKAL, Y. AMIRAT, M. AISSA
LIIA Lab.University Paris 12, France

**98ADM067**

***Abstract:*** *Current and/or future industrial needs orient to a greatest flexibility of production systems and to the optimization of performances such that quality, products costs and manufacture delays. The implementation of these systems implies problems of control, synchronization, communication and management of conflicts between entities that constitute the process to control. The origin of these problems is the distribution of processing and data that are generally submitted to spatial and temporal constraints imposed by the application. In this paper, we propose on one hand a dynamic and static generic modeling of a manufacturing and robotics system and on the other hand a new approach to process issues of disruption management that rests on negotiation heuristics between entities in order to satisfy precedence constraints. Control and driving architecture model is based on the concept of autonomous, generic and cooperative entity class. Each entity possesses the same event and functional model based on a real time multitasking model. This model offers to each entity capacities of processing asynchronous events that characterize applications of robotised manufacturing. Furthermore, the coordination and the collaboration between entities of the global system is based on the client-server model that insures both the communication between these entities and the spatial and temporal consistency of informations according to a full-duplex protocol. The management of a disruption within an entity rests on algorithms allowing to determine priorities of each operation constituting the operation plan assigned to the disturbed entity. To validate this approach, we have developed a multi-robots platform simulator. Simulation results obtained from typical applications are presented and analysed.*

*Key words: Robotics, Manufacturing, Driving, Control, Architecture, Real time, Multitasking Communication network, DAI, Scheduling, Precedence constraints.*

## *1. Introduction*

Robotics and manufacturing systems which constitute a particular class of complex processes are generally composed of several equipments, and integrate several functions (control, manufacturing management , scheduling, etc.). Thus, decisions associated to the management of the whole process have to be taken according to several levels. The implementation of these systems involves consequently problems of management, control, synchronization, communication and conflicts management between entities which constitute the. process. Origin of these problems are the distributed processing and data that are generally submitted to spatial and temporal constraints imposed by the application.

Distributed architectures constitute robust and reactive solutions, to solve the control problem of distributed manufacturing and robotics systems problems. They facilitate treatment of varied problems such as interruption ability, reactivity, time processing, tasks organizing, etc. Furthermore, the heterogeneous entities and their spacial distribution implies to treat the problem of cooperation by taking into account constraints on data exchange between entities. More, these architectures allow the treatment of the spatial and temporal consistency of data. This is done from the building of a global referential imposed by the global supervisor of the system. The driving and control system has to insure both the communication with all components of the system and the compatibility of exchanged data. It has also to transmit informations about the quality of products and to propose modifications

when a disruption occurs. The validation of the driving and control system involves the following steps:
- The modelization of design data (parts, supports, motions procedures, etc.).
- The modelization of qualities of transformation (building, manufacturing, etc.).
- The implementation of correction actions when a disruption occurs.

The implementation of these models provides to the control system the ability to inform local control modules and the supervisor on required physical states and performed states [BAJ 91].

The design of such systems requires on one hand a functional and evenemential specifications of the control architecture and on the other hand the specification of communication and cooperation models between entities. These specifications allow to describe functions associated to each entity and relationships that link them.
In this paper, we propose a generic model for control and driving architecture of a manufacturing and robotics system as well as communication and cooperation models between entities. The model of control architecture is based on the concept of autonomous, generic and cooperative entity. Indeed, each entity of the system has the same functional and evenemential model based on a multitasking real time model. This model provides to each entity processing ability of asynchronous events (interruptions and signals of the work environment) that characterize robotics and manufacturing applications. Furthermore, client - server model is used to insure both the communications between entities and the spatial consistency of informations. This model allows an easy integration of necessary information for the negotiation between entities when a disruption occurs. To implement the proposed models, we have developed a multi - robots simulator platform. Simulations results and some situations are presented and analyzed.

## *2. Requirements and constraints of robotics and manufacturing distributed systems*

In a robotics and manufacturing system, equipments to control can be several and heterogeneous. It is therefore necessary to define classes of applications of these systems to take into account specific problems and common problems of each class. Without giving an exhaustive list of applications performed with such systems, we can enumerate two classes of applications:
- The mobile robotics class;
- The manufacturing robotics class;

Since the last decade, the flexibility of robotics and manufacturing systems has become an important field of research. Indeed, new models control have been implemented in order to increase modularity and simplicity of design. This actual tendancy is due to research progress in the distributed computation area [TSU 95] [TSU 96]. However, the distribution of data processing, algorithms and decision necessitate a sophisticated cooperation and coordination. This distribution allows also a better faults tolerance, an easy modification, and an optimal exploitation of parallel computation.
In mobile robotics, an entity evolving in an unknown environment is constrained by the complexity of this environment. The perception of this entity depends on its own sensory means. The cooperation problem is generally approached in order to solve some particular problems which can be divided according to two categories: The cooperation problem at knowledge level (cognition) and the cooperation at action level. In the first one, robots exchange data allowing them to perform specific operation. This exchange of data must optimize this operation. A significant example is the exploration in an unknown environment. In this case, entities are self organized in society. Each entity informs this society of data exchanged during exploration. The collaboration can also be built at action level. Constraints which are associated with this collaboration type are severe. The evolution of each entity depends on the evolution of the other. The cooperation is performed for the realization of a specific application where each entity participates by its actions and its perceptions.

The problem treated in this paper is the resolution of control and driving problem by the use of a multi-agent system based on DAI (Distributed Artificial Intelligence) technics. This multi-agent system appears as the result of coordinated actions carried out by an autonomous set of entities endowed by certain a degree of intelligence [MON 93] [ATT 97]:

## *3. Control/driving architecture and communication models*

In order to satisfy real time constraints on communication, cooperation, portability, and conflicts management of a robotics and manufacturing distributed system, we propose hereafter a model of control and driving architecture as well as communication model between entities.

### *3.1. Driving/ control distributed architecture*

Generally, a manufacturing system includes a set of entities where each one performs a specific function. An entity in terms of manufacturing can includes several resources which cooperate between them in order to perform a particular function such as assembly, manufacturing, convoying, etc [TOU 95][TOU 97]. In the proposed approach, entities are cooperative. For consistency communication reasons, each entity has the same level of decision and the same view of the whole system.
The control and driving architecture includes two distribution driving levels: horizontal and vertical levels. The horizontal distribution constitutes in this case the functional model of the system. This architecture offers two driving levels; a supervisor level and an entity level. The generic control and driving architecture of a manufacturing system is illustrated on figure 1, it includes two distribution types:

-**An horizontal distribution**: in this level, entities execute elementary operations of different natures such as assembly, convoying, etc. Each entity is drived by a local controller which receives orders from the supervisor.
-**A vertical distribution**: this distribution is organized according to the two following levels:

- The supervisor level: It manages both flows of informations received from the different entities and risks of production.
- The cell level: It manages the local flow of informations in order to perform a task assigned by the supervisor.

**The global supervisor**: It includes the following modules:

- The control module : It insures management of flows informations with respect to all entities.
- The global planning module: This module receives orders from the control module and transforms each one into a plan of actions.
- The resumption module : It is invoked when a disruption or alea occurs. This module proposes a new plan of actions.
- The communication module: It insures data exchange between the supervisor and entities.
- The report module: It generates a report on state of each entity such as demand of the control module or the resumption module.

**The supervisor of entity**: The supervisor structure includes the following modules:

- The module of control : It insures the local supervision of the entity and watch over to the good progress of operations at the resource level.
- The planning module: It constitutes a generator of plans of operations after taking into account orders from the control module and accessing database. The provided plan is a sequence of operations that undertakes a replanification when a disruption occurs.
- The report module: It allows to give an execution report on the mission assigned to the entity.
- The resources management module : It insures the update of database of the entity.

- The communication module at supervisor level : It insures the exchange of informations between the global supervisor and the control module of entity.
- The communication module at entity level: It insures the transmitting of informations between modules of the entity and its resources.
- The negotiation module: It is invoked by the planning module in case of disruption. It determines priorities of operations within the disturbed entity. The allocation of priorities is performed according to informations received from other entities. In the case of a multi - mobile robots system, this module is invoked in case of disruption such as the simultaneous access of two mobile robots to a common path section, or again in case of delay due to the presence of obstacles on the reference path , etc. In such situation, it starts negotiations with robots in order to establish for each new priorities which satisfy precedence constraints.

**The communication system**: It builds data frames. It insures also the sending and the verification of integrity of data frames. The communication model is of client-server type.

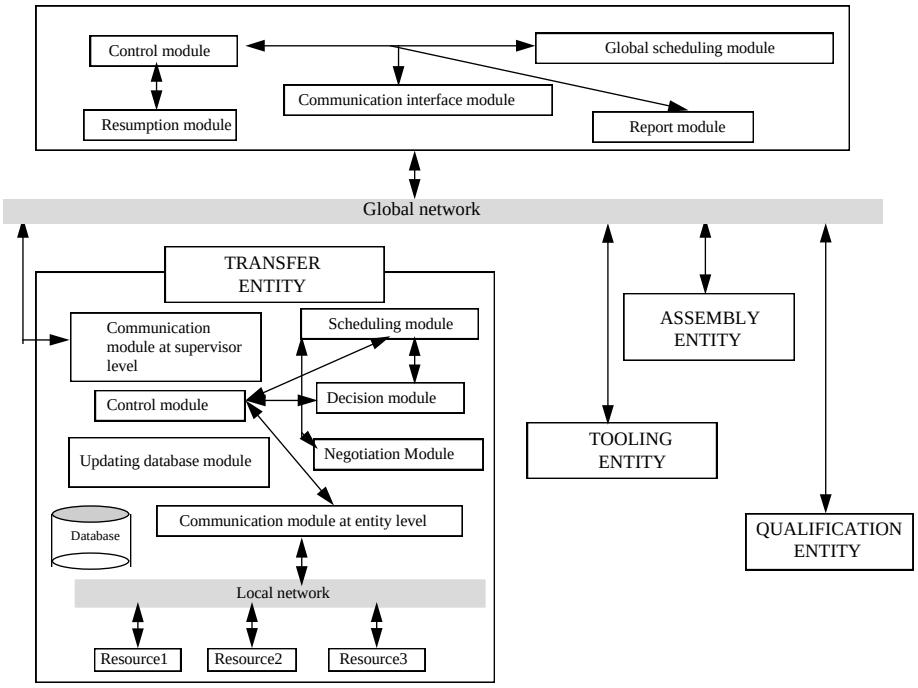


Figure 1: Real time scheduling operations in case of disruption

The scheduling method that we propose consists in making real time modification of the plan of operations assigned to a disturbed entity. The rescheduling is based on the respect of precedence constraints on operations assigned to the different entities. The method consists therefore in minimizing the propagation of a disruption occured within an entity of the system [TOU 97]. The search for an optimal solution which allows to minimize the number of affected entities is performed according to the two following steps:

- In the first step, the controller of an entity identifies the exception state produced when a disruption occurs then tries to carry out a local resumption without impact on the other entities. This resumption is carried out without invoking negotiation with entities of the global system.
- In the second step, in case of failure in the first step, the controller carry out search of an optimal scheduling of operations by the use of a negotiation mechanism with some entities. This mechanism is based on a set of heuristics which are presented hereafter.

***3.2. Scheduling mechanism***

When a disruption occurs within an entity, two situations can appear. In the first one, the solution consists in building a scheduling based on local informations of the entity. In the second one, the solution consists of the implementation of a negotiation mechanism between the different entities in such a way that to minimize the disruption propagation. We describe hereafter the two situations.

- *Scheduling without negociation*

We propose in this case a heuristic which determines an on line scheduling based on local informations without negotiation. The local scheduler executes the following operations:
1 - It identifies disruption occured at the entity i.
2 - It determines the action $a_i \in Gai$ which has the shortest execution time. Let Gai the set of actions which realize the transition from an exception state to a normal state by the application of a function $\Im_{ii}$ ( $\Im_{ii}$ is the effect of an action taken in entity i on the state of an entity j)
3 - It builds the operations plan by taking into account the delay produced by the disruption, while keeping the initial order of operations.
4 - It verifies if no operation in the disrupted entity i possessing at least a successor operation in an entity j, exceeds its precedence expiration. In this case, it implements the operations plan , otherwise it carry out a change in the order of operations related to the entity i.
5 - If no operations plan respects at least a precedence constraint, it invokes then the negotiator and waits for operations priorities of the entity i.

- *Scheduling with negociation*

The negotiator carry out the following steps:
- It sends the new scheduling to all disrupted entities and waits for a report. Each entity sends a boolean reply uj which is equal to 1 if it can manages locally the disruption produced by the entity i. In the other case (uj=0), the entity j transmits two parameters to the negotiator of the entity i:
  - the first is Tj the set of predecessor operations which exceed their precedence expirations in case of local resumption.
  - the second is CAR the set of disrupted entities in case of optimal local resumption. CAR =CARG $\cup$ CASG (CASG: the set of disturbed entities concerned by an optimal resumption in the entity i and capable to manage locally the disruption (ui=1). CARG: the set of entities that can not manage locally a disruption (ui=0).

-It determines the minimum of set $CARj_{(1\leq i\leq n)(j\neq j)}$ such as:$Min(CARj)_{(1\leq i\leq n)(i\neq j)} < n(CARG)+n(CASG)$. If this minimum is equal zero, the negociator applies the proposed scheduling. Otherwise, it carry out the following actions:
-In case of local resumption, it identifies in the entity j all displaced successors operations which produce an exceeding of the precedence expiration for some predecessor operations of the same entity j (uj=1). This set is called Wj .
- It builds the set of operations of the entity i predecessors of successor operations belonging to the sets Wj(j≠i). This set constitutes the class of higher priority predecessors operations.
- It builds a second set which includes all remaining predecessor operations. This set constitutes the class of lower priority predecessors opérations.
- It builds the third class which is the set of remaining operations of the entity i.

التعليق [G1]:

The scheduler applies the following algorithm to these three operations classes :

1 - Classify according to a crescent order of precedence expirations the higher priority predecessor operations. If n represents the number of operations, a label i (1≤ i≤n) is assigned to each operation.
2 - For each predecessor operation i scheduled in 1:

2.1. Compute the sum $\Delta j$ of unoccupied period of time between the end of processing of the operation i - 1 and the precedence expiration of the operation j. This treatment is done for each j ($i \leq j \leq n$ )

2.2. If it exists no scheduled predecessor operations of lower priorities whose execution time is lower than to $\min(\Delta j)$( $i \leq j \leq n$), the one which has the nearby precedence expiration is inserted before the operation i.

2.3. Go to 2.1.

3 - Schedule according to a creascent order of precedence expirations the remaining predecessor operations of lower priority. Let m the number of operations.

4 - Assign news labels to predecessor operations from the classification built in stages 2 and 3.

5 - For each predecessor operation i scheduled in 4:

5.1. Compute the sum $\Delta j$ of unoccupied period of time between the end of processing of the operation i - 1 and the precedence expiration of the operation j. This processing is carried out for each j ($i \leq j \leq n+m$).

5.2. If it exists no scheduled predecessor operations of lower priorities whose time of execution is lower than $\min(\Delta j)(i \leq j \leq n)$, the chosen one verifies the used distribution rule and is inserted before the operation i

5.3. Go to 5.1.

6 - Schedule the remaining operations according to the distribution rule.

## *4. Simulations results*

To illustrate the proposed method, we present hereafter some disruptions scenario. These simulations have been performed using of a simulator of multi - robots platform developed for this purpose.

We consider a distributed system which comprises a supervisor and two mobile robots entities. Operations assigned to robots are an exploration of work environment. These operations represent a set of trajectories the robots have to follow. As the three robots evolve in the same environnement, the management of the space resource is then needed. Thus, we define precedence constraints of operations assigned to these robots so as to solve problem of space resource sharing. The obtained results are given hereafter.

**entite A:**

| operation | time | order of execution |
|---|---|---|
| 1 | 5 | 1 |
| 2 | 8 | 2 |
| 3 | 10 | 3 |
| 4 | 3 | 4 |

| predecessor | successor | deadline | entity |
|---|---|---|---|
| 2 | 14 | 2 | B |

**entite B:**

| operation | time | order of execution |
|---|---|---|
| 1 | 15 | 1 |
| 2 | 7 | 2 |
| 3 | 8 | 3 |
| 4 | 9 | 4 |

| successor | predecessor | entity |
|---|---|---|

2 2 A

The robot entity A includes a predecessor operation 2 , its successor operation in the robot entity B is 2, the precedence deadline is equal to 14. We consider a disruption which produces a delay equal to 4 units of time in the execution of the operation 2 assigned to the robot entity A. This leads to a surpassing equal to 3 for the operation 2 with respect to its precedence deadline. By applying the first heuristic, we obtain a new scheduling illustrated on figure 2.

| operation | time | order of execution |
|---|---|---|
| 2 | 8 | 2 |
| 1 | 5 | 1 |
| 3 | 10 | 3 |
| 4 | 3 | 4 |

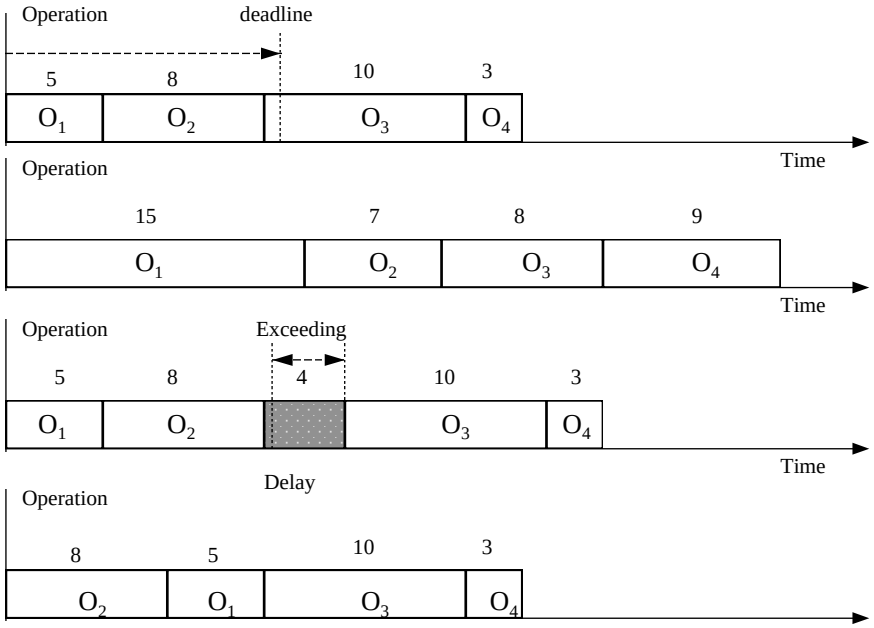


Figure 2. Rescheduling of operations

## *4. Conclusion*

In this paper, we have presented an approach which constitutes a generic solution to solve the problem of driving and control of a manufacturing and robotics system. This approach is based on a new model control for the managment of such systems which satisfy constraints of flexibilit, modularity and design simplicity . The distribution of the system according to several entities allows a better faults tolerance if we have a network which offers services of synchronization, verification of consistency. Furthermore, a mechanism of negotiation between entities has been proposed to treat the problem of the decentralized scheduling in case of disruption. This mechanism is based on an heuristic which allows to minimize the propagation of the disruption and satisfy constraints of precedence of operations. To validate these concepts, we have developed a simulator of platform multi - robots. The algorithm for the rescheduling of operations in case of disruption has been implemented on this simulator and is currently under way evaluation. Future works will be devoted on the one hand to the implementation of others heuristics to treat simultaneously several types of

constraints , on the other hand to the endowing of each entity with capacities of recognition of situations, reasoning and behavior generation using AI techniques [BEN 97].

*References*

[ATT 97] A. ATTOUI. Les systèmes multi-agents et le temps réel, Editions Eyrolles, 1997
[BAJ 91] E. BAJIC, T. DIVOUX, J. RICHARD. Définition d'une Cellule Flexible Auto-Contôlée de Fabrication Mécanique. RAPA, Hermes publishing. Vol, n°4/1991, pp. 403-434.
[BEN 96] R. BENAMARA, C. MORENO. Control agents. Implementing on industrials networks. Expersys' 97, Expert systeme application and artificial intelligence, IIIT conference, pp. 219-227, Champs/Marne, France, octobre 1996.
[MON 93] E. MONACELLI. Un système de programmation automatique de niveau tâche. of doctorate Thesis, University Pierre and Marie CURIE, Paris, France, 1993.
[TOU 95] Z.TOUKAL, Y.AMIRAT, S.BABACI, J.PONTNAU, Programming language for robotic assembly tasks, Proceedings of 1995 IEEE/IAS Conference On Industrial Automation and Control: Emerging Technologies, Taipei, TAIWAN, May 22-27, 1995, pp. 612-619.
[TOU 97] Z.TOUKAL, M. AISSA, Y. AMIRAT and PONTNAU J., Modélisation d'un système robotique distribué: Mécanisme de négociation pour le réordonnancement des tâches. CNRIUT'97 national conference of university research, Toulouse, France, May 14-15-16, 1997.
[TSU 95] T.K. TSUKADA and K. G. SHINN. A distributed approch to intelligent tool managment in flexible manufacturing systems. IEEE Transactions on Robotics and Automation. 1995.
[TSU 96] T.K. TSUKADA and K. G. SHINN. PRIAM: Polite Rescheduler for Intelligent Automated Manufacturing. IEEE Transactions on Robotics and Automation.vol. 12, n°2, April 1996.