Analytical design methodology of agent oriented manufacturing systems

Patriti V.F., Schäfer K., Charpentier P., Martin P.
Université Henri Poincaré, CRAN (Centre de Recherche en Automatique de Nancy)
BP 239 54506 Vandoeuvre Cedex, France.
Phone : (33) 3 83 91 20 00. Fax : (33) 3 83 91 23 90
{patriti, schafer, charpentier, martin,}@cran.u-nancy.fr

Abstract

Multi-agent manufacturing lacks of design methodology. The aim of this paper is to present a formalisation method that enables the designer to characterise multi-agent manufacturing systems in reliability and robustness. This characterisation allows the designer to estimate some characteristics of the system to be designed before its realisation. Finally we present an example of design on the multi-agent machine-tool Shiva.

Keywords

multi-agent, holonic, machine-tool, design, reliability

1 INTRODUCTION

The inherent limitations of the original centralised control architecture caused the apparition of agent oriented control paradigms. Multi-Agent, Holonic Manufacturing, Heterarchical and Bionic Systems contribute to **agent-oriented manufacturing**. Among the numerous questions that arise from these new control paradigms are the problems of agent determination and characterisation of the system.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35390-6_58

L. M. Camarinha-Matos et al. (eds.), *Intelligent Systems for Manufacturing*
© IFIP International Federation for Information Processing 1998

At the workshop level, agent determination seems rather straightforward. Indeed, most of the work at the workshop level follow a natural decomposition of the workshop. But as soon as the studied level changes, the typology changes. Thus there are agents for tools (Veeramani & al., 1993), for the basic components of robots (Regnier & Duhaut, 1995), flexible cell (Sohier & al., 1997) and machine-tool (Patriti, 1998), scheduling agents (McFarlane & al., 1995) and watchdog agents (Parunak, 1997). But to help the designer in his choices, few design methodologies adapted to the manufacturing problem are presented, with the exception of Parunak (1997) .

This problem could easily be solved if we could characterised the performances of the overall system. But most of the agent-oriented manufacturing systems properties cannot be evaluated without simulation based on realistic numbers of agents and interchanges (Parunak, 1997). That usually means that the multi-agent control system **is** already realised. Thus to characterise the system at an early stage of the design process seems to be of great interest and utility. The aim of this paper is to present an analytical modelling technique enabling a designer to characterise his multi-agent control system, thus providing him with information at the specification level. The properties chosen are : reliability and robustness.

2 THE MULTI-AGENT SYSTEM

Although our approach could be adapted to other types of multi-agent systems, the analytical model proposed fits a specific type. This chapter presents this type through two different aspects : the relationship between the agents and the physical system on the one hand and their organisation on the other hand.

The agents and the physical system

A manufacturing system is composed of both a control and an physical system. The physical system transforms, transports and stocks the parts, receives orders from the control system and sends back information. We will name the elements of the physical system **components**. With an agent oriented approach, the basic element of the control system is called **agent**. We suppose that :

- Each agent controls one or more components, but the control of one component cannot be shared among several agents.
- Each component or agent is defined by a state variable which equals 1 if the element is functional and 0 if the element is not functional.

Although we do not consider a degraded behaviour within an agent, the degraded behaviour of the system is possible with the re-organisation of the agents.

Organisation

There are a lot of different methods to make the organisation emerge. Like our model, most of the currently applied methods in manufacturing are inspired by social science. The only assumption this model makes is that the organisation is obtained through a **service**-based protocol. A service is a specification of tasks that the agents are able to provide to other agents.

3 THE ANALYTICAL MODEL

The system is composed of :

- A, set of the **basic agents**.
- C, set of the **physical components**.
- F, set of the **services** that the components provide. $F = \{F_1, ..., F_f\}$. f is the number of different services in the system or the cardinal of F.
- **Ω**, that represents the relationships between agents and components
- **Φ**, that represents the **physical architecture** of the system. That is the set of the relationships between the components.
- **Ψ**, that represents the **needs** of each components. That is the set of the relationships between the services that the components provide and the services they need to realise it.

From these elements, we can compute :

- **S**, the **availability** of the services for the components or the agents. That means that S indicates if a service can be used by others or not.

Let MS be the manufacturing system. MS is the six-tuple :

$$\mathbf{MS} = \left\langle \mathbf{C}, \mathbf{F}, \mathbf{A}, \Psi, \Omega, \Phi \right\rangle . \qquad (1)$$

The set of the physical components is C

$C = \{C_1, ... ,C_c\}$. c is the number of the components in the system or the cardinal of C. C is composed of the active components like axis, robots and of passive components like tools and parts. The state of component C_i is defined by c_i.
If $c_i = 0$, the component is out of order, it cannot provide any service.
If $c_i = 1$, the component is working, it can provide its services if its needs are satisfied.

The set of the basic agents is A

$A = \{A_1, ... ,A_a\}$. a is the number of basic agent in the system or the cardinal of A. The state of agent A_i is defined by the variable a_i.

*The relationships between agents and components is the matrix **Ω***

Ω is a matrix of dimension (a, c) such as :
$\Omega_{i,j} = 1$, if the component C_j belongs to agent A_i.
$\Omega_{i,j} = 0$, if the component C_j does not belong to agent A_i.
A component can belong to only one agent.

The state variable of the agents a_i

Now we can define the state variables of the agents.

If $a_i = 0$, then the agent A_i is down and cannot provide any service ;

If $a_i = 1$, then the agent is functional and it can provide services if its need are satisfied.

The agent being an aggregation of components, a_i is a function of c_i :

$$\mathbf{a_i} = \prod \mathbf{c_k} \text{ such as } C_k \text{ is a component of } A_i \Rightarrow \mathbf{a_i} = \prod_{\mathbf{k=1}}^{\mathbf{k=c}} \left(\mathbf{c_k} + \overline{\Omega_{\mathbf{i,k}}} \right). \tag{2}$$

Figure 1 presents an example of the relationship between C and A. There are four components : C_1, C_2, C_3, C_4 and two agents A_1, A_2. C_1 and C_2 belong to A_1, while C_3 and C_4 belong to A_2. That means that :

$$\Omega = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}. \tag{3}$$

*The architecture is represented by the matrix **Φ***

While all the components of the system can provide services, there are physical limitations to the co-operation between agents. One of them is the physical architecture of the system. For example a non-mobile robot cannot work with every machine-tool of the workshop but with only those in range. These limitations are modelled with $\mathbf{\Phi}$. $\mathbf{\Phi}$ is a matrix of dimension (c, c), c being the number of components in the system.

If $\mathbf{\Phi}_{i,j} = 1$, the component C_j can provide its services to the component C_i.

If $\mathbf{\Phi}_{i,j} = 0$, the component C_j cannot provide its services to the C_i.

An example of architecture is given in Figure 2. The resulting matrix is :

$$\Phi = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{4}$$

The availability of a service for a component is expressed by the matrix S

If $S_{i,j} = 1$, then the component C_i may use the service F_j ;

If $S_{i,j} = 0$, then the component C_i cannot use the service F_j.

We will see how to compute the availability later.

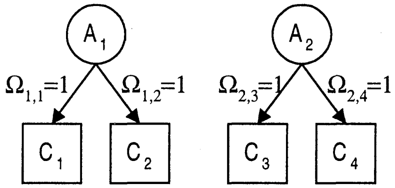

Figure 1 : Example of relationships between C and A.

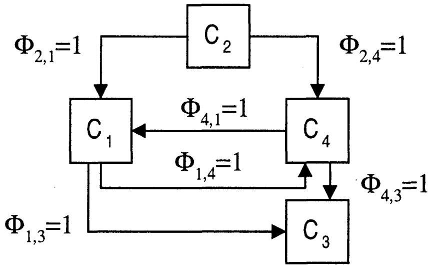

Figure 2 : Example of an architecture.

The need of the physical components are represented by the functions $\Psi_{i,j}$

$\Psi_{i,j}$ are boolean functions of S_i

If $\psi_{i,j}(S_i) = \psi_{i,j}(S_{i,1}, \ldots, S_{i,f}) = 1$, then the needs of the component C_i for the service F_j are fulfilled. Therefore if the component and the associated agent are functional, the service F_j can be provided.

If $\psi_{i,j}(S_i) = \psi_{i,j}(S_{i,1}, \ldots, S_{i,f}) = 0$, then the needs of the component C_i for the service F_j are not fulfilled. Therefore even if the component and the associated agent are functional, the service F_j **cannot** be provided.

For example a tool provides the service **machining feature X**. Its needs for that service are **rotation in z** and **translation in z**. The boolean function Ψ for the component tool will be :

$$\psi_{\text{tool, machining feature X}} = S_{\text{rotation in z}} \cdot S_{\text{translation in z}} . \qquad (5)$$

Availability

Now we have enough information to compute the availability of the services, which can be done with or without the influence of the control system. In the latter case we will refer to it by the name of **potential availability Sp** :

$$Sp_{i,j} = \sum_{k=1}^{k=c} \Phi_{i,k} \cdot c_k \cdot \psi_{k,j}\left(Sp_k\right) . \qquad (6)$$

$Sp_{i,j} = 1$ means that a service F_j is available if the asking component C_i is in relation with a provider component C_k that is functional, has the service and has its needs satisfied. For the overall system, we have to integrate the effect of the aggregation of components in agents. The availability is then called the **overall availability Sr** :

$$Sr_{i,j} = \sum_{k=1}^{k=c} \sum_{l=1}^{l=a} \Phi_{i,k} \cdot \Omega_{l,k} \cdot a_l \cdot \psi_{k,j}\left(Sr_k\right) . \qquad (7)$$

$Sr_{i,j} = 1$ means that the service F_j is available to the component C_i if the provider agent A_l is functional, has a component C_k that can realise the service, has its needs satisfied and is in relation with C_i.

4 CHARACTERISING THE SYSTEM

The former model enables us to build the overall properties of the system from the distributed models of each component and agent. This is based on the theory of reliability of non-renewable components (Bon, 1995). The properties can be computed for both the overall system and the potential system. The overall system integrates the influences of the control system (the agents), while the potential system does not.

Reliability

The designer may compute the reliability of any service in the system. In a manufacturing system the most important service is associated with the product : **manufacture the product**. The **reliability of the potential system** is defined as the probability that the service would be realised without the influence of the control system. The formal definition is then :

$$\mathbf{R_p} = \mathbf{P}\left[\Psi_{\mathbf{manufacture\_product}}\left(\mathbf{Sp_i}\right).\mathbf{c_i} = 1\right]. \qquad (8)$$

If there are more than one product in the system, or if there is an inexhaustible amount of raw material at disposal, we can assume that $c_i=1$. The **reliability of the overall system** integrates the influence of the agents in the probability. Its formal definition is then :

$$\mathbf{R_r} = \mathbf{P}\left[\Psi_{\mathbf{manufacture\_product}}\left(\mathbf{Sr_i}\right).\left(\sum_{j=1}^{\mathbf{a}} \Omega_{\mathbf{j,i}} \cdot \mathbf{a_j}\right) = 1\right]. \qquad (9)$$

Robustness

The designer may also compute the robustness of any service in the system. We give our own definition of this property based on the notion of the critical path :

The path Pa_{S_i} for $\phi(S_i) = 1$ is a set, such as: $\mathbf{Pa_{Si}} = \{j : S_{i,j} = 1\}$.

Pa_{S_i} is a critical path if : $\forall \mathbf{S_k} < \mathbf{S_i} \Rightarrow \phi(\mathbf{S_k}) = 0$.

The robustness of a system H is defined as the number or critical paths of realisation of the service. Its formal definition is then :

$$\mathbf{H} = \mathbf{card}\left\{\mathbf{Pa_{S_i}} : \phi(\mathbf{S_i}) = 1, \forall \mathbf{S_k} < \mathbf{S_i} \Rightarrow \phi(\mathbf{S_k}) = 0\right\}.. \qquad (10)$$

Because this is a manufacturing system, the service used is again **manufacture the product**. That means that the robustness of the potential system Hp is computed for :

$$\phi(S_i) = \Psi_{manufacture\_product}(S_i).c_i = 1. \tag{11}$$

And that the robustness of the overall system is computed for :

$$\text{For } \phi(S_i) = \Psi_{manufacture\_product}(S_i).\left(\sum_{j=1}^{a}\Omega_{j,i}.a_j\right) = 1. \tag{12}$$

5 DESIGNING A SYSTEM

Two policies are then available for a designer. First use the analytical formalism to search the solution space and generate the optimal solutions with an optimisation algorithm. Second generate manually several solutions and compare the emergent properties to choose the best one. These steps do not exclude simulation, in fact one of the most important properties of a manufacturing system , namely productivity, is not covered by this formalism and still needs a validation through simulation.

The following design is based on Shiva (Figure 3) a multi-agent machine-tool built at the CRAN (Centre of Research for Automatic Control in Nancy). Shiva is basically a double milling machine with two autonomous three axis workstations.

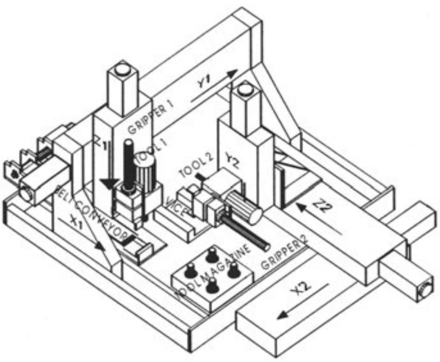

Figure 3 : Shiva.

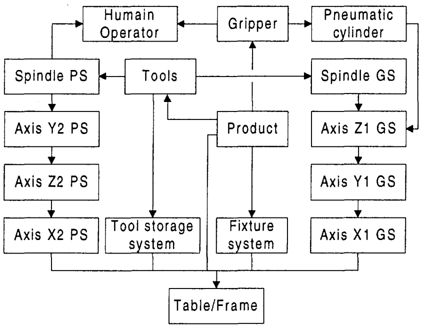

Figure 4 : Architecture of Shiva.

5.1 The physical system

The architecture of Shiva is given in Figure 4. As we can see there are six linear axis grouped in two stations (PS and GS). The workstation GS has in addition to the spindle a gripper with a roto-linear pneumatic cylinder that enables it to transport workpieces. If the gripper is occupied otherwise, the human operator can move the product himself. Because of its vertical position the GS spindle is able to change its tool without manual aid, while the PS spindle must use the human operator.

The set C is composed of 21 elements :

C = {Table, Tool Storage System, Fixture System, Spindle GS, Spindle PS, Gripper, Cylinder, Axis 1 GS, Axis 2 GS, Axis 3 GS, Axis 1 PS, Axis 2 PS, Axis 3 PS, Product, Tool1, Tool2, Tool3, Tool4, Tool5, Tool6, Operator}. (13)

The set F is composed of 26 services :

F = {roughing(-x), roughing(z), semi-finishing(-x), semi-finishing(z), finishing(-x), finishing(z), stock_tool, machining(-x), machining(z), move_tool, move(x), move(y), move(z), stock_component, Fixture A, Fixture B, Fixture C, Fixture D, Fixture E, stock_product, rotation(x), rotation(y), rotation(z), transport_product, translation(z), manufacture_product}. (14)

Roughing, **semi-finishing** and **finishing** are machining operations. They are given for a specific direction. All these services are provided by the tools. **Machining** is the service provided by the spindles for the tools. This service combines rotation and movement. **Rotation**, **translation** and **transport_product** are transport services provided by the cylinder and the gripper for the product. **Stock_tool** and **move_tool** are services provided for the tools to move them in and out of the tool storage system. **Fixtures** are the services given by the fixturing system. Each different fixture represents a different position of the part. **Moves** are the services provided by the axis. They represent controlled linear movements. **Stock_components** is the service provided by every component that physically supports another component. For example the axis Y1 GS is supported by the axis X1 GS. **Stock_product** is the service provided by the table. To give all the Ψ functions would exceeds the authorised length of this paper, so we limit the description to the most important functions.

Product

The main function Ψ is the function of the product : manufacture_product. This function gives the structure of the chosen process plan.

Ψ= Stock_Product. Transport_Product. {Fixture A. Roughing(-x). SemiFinishing(-x). Finishing(-x). Roughing(z). SemiFinishing(z). Finishing(z) + Fixture B. Fixture C . Roughing(z). SemiFinishing(z). Finishing(z) + Fixture D. Fixture E. Roughing(-x). SemiFinishing(-x). Finishing(-x) }. (15)

Tool

To realise its function, the component part needs the services of the tools. The function Ψ of the service Roughing(-x) for the tool is the following :

$$\Psi = \text{Machining(-x). Stock\_tool.} \tag{16}$$

5.2 The potential system

Using each function of each component and comparing them with the architecture Φ, we can reconstruct the success conditions of **manufacture_the_product** according to the state variables of the components. The corresponding reliability block diagram (Bon, 1995) is represented Figure 5.

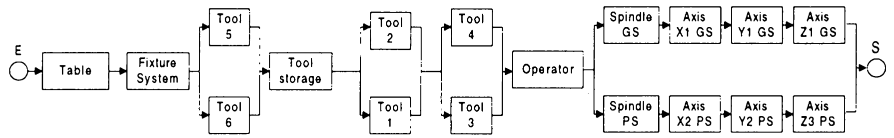

Figure 5 : Reliability block diagram of the potential system.

Then we can deduce the potential reliability Rp and the potential robustness Hp :

$$\mathbf{H_p} = 16$$

$$\mathbf{R_p} = \mathbf{R}_{\text{Fixture}} \cdot \mathbf{R}_{\text{Tool storage}} \cdot \mathbf{R}_{\text{Operator}} \cdot \mathbf{R}_{\text{Table}} \cdot \left(\mathbf{R}_{\mathbf{Tool5}} + \mathbf{R}_{\mathbf{Tool6}} - \mathbf{R}_{\mathbf{Tool5}} \cdot \mathbf{R}_{\mathbf{Tool6}}\right) \cdot$$

$$\left(\mathbf{R}_{\mathbf{Tool1}} + \mathbf{R}_{\mathbf{Tool2}} - \mathbf{R}_{\mathbf{Tool1}} \cdot \mathbf{R}_{\mathbf{Tool2}}\right) \cdot \left(\mathbf{R}_{\mathbf{Tool4}} + \mathbf{R}_{\mathbf{Tool5}} - \mathbf{R}_{\mathbf{Tool4}} \cdot \mathbf{R}_{\mathbf{Tool5}}\right) \cdot$$

$$\begin{pmatrix} \mathbf{R}_{\mathbf{SpindleGS}} \cdot \mathbf{R}_{\mathbf{AxisX1GS}} \cdot \mathbf{R}_{\mathbf{AxisY1GS}} \cdot \mathbf{R}_{\mathbf{AxisZ1GS}} \\ +\mathbf{R}_{\mathbf{SpindlePS}} \cdot \mathbf{R}_{\mathbf{AxisX2PS}} \cdot \mathbf{R}_{\mathbf{AxisY2PS}} \cdot \mathbf{R}_{\mathbf{AxisZ2PS}} \\ -\mathbf{R}_{\mathbf{SpindleGS}} \cdot \mathbf{R}_{\mathbf{AxisX1GS}} \cdot \mathbf{R}_{\mathbf{AxisY1GS}} \cdot \mathbf{R}_{\mathbf{AxisZ1GS}} \cdot \mathbf{R}_{\mathbf{SpindlePS}} \cdot \mathbf{R}_{\mathbf{AxisX2PS}} \cdot \mathbf{R}_{\mathbf{AxisY2PS}} \cdot \mathbf{R}_{\mathbf{AxisZ2PS}} \end{pmatrix}$$

Finding the agents

Finding the agents can be seen as the determination of Ω that maximises Hr and Rr. We have decided not to use an optimisation algorithm, but to make a heuristic search by comparing possible structures. Figure 6 shows the result : Three basic agents are composed of more than one component. The agent **GS** is composed of the axis **X1**, **Y1** and **Z1**. The agent **PS** is composed of the axis **X2**, **Y2** and **Z2**. The agent **Transport** is composed of the **Gripper** and the **Pneumatic cylinder**. The other basic agents consist of only one component. Thus Hr and Rr are the same than Hp and Rp. Shiva was then simulated to evaluate the productivity of the control architecture. This specification was programmed using the contract-net protocol for the control system of the machine-tool Shiva.

6 CONCLUSION

We have presented a design methodology based on a formalism, which enables the designer to evaluate the reliability and the robustness of a system at the specification level. Finally we have shown an application of this methodology through the design of the agents of the machine-tool Shiva.

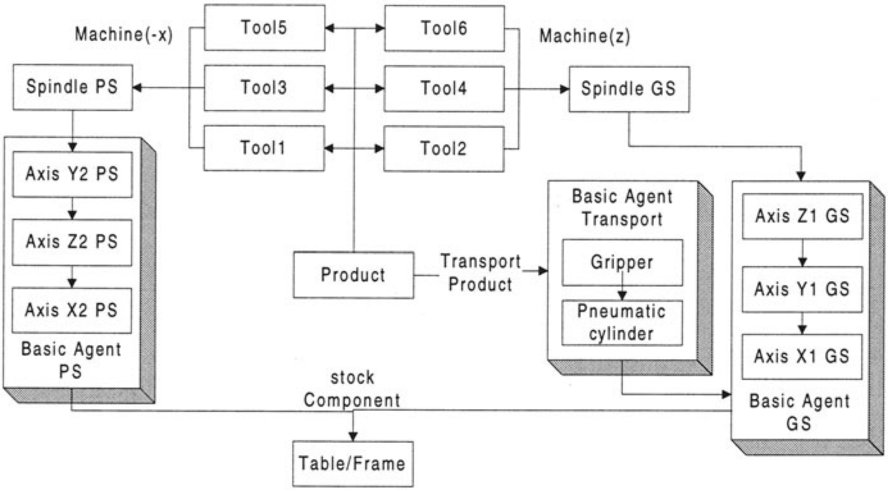

Figure 6 : The chosen architecture.

7 REFERENCES

Bon J.L., (1995) Fiabilité des systèmes, méthodes mathématiques, Masson, Paris.

MacFarlane D., Marett B., Elsley G., Jarvis D. (1995) Application of holonic methodologies to problem diagnosis in a steel rod mill. *IEEE Conference on systems man and cybernetics*, october 95, Vancouver, 940-945.

Patriti V. (1998) Systèmes de pilotage auto-organisés et gammes distribuées : méthode de conception et application à une machine-outil, University Henri Poincaré-Nancy I thesis, 16/01/98.

Parunak H.V.D., Sauter J., Clark S. (1997) Toward the specification and design of industrial synthetic ecosystems. *Workshop on Agent theories, architectures and languages: ATAL'97.*

Regnier S., Duhaut D. (1995), Une approche multi-agents pour la résolution de problèmes robotiques. *Troisième journées francophones sur l'IAD et les SMA*, Saint bandolf, savoie, France, 15-17 March, 91-97.

Sohier C. , Denis B., Vigny C. (1997) Validation du comportement dynamique d'un système multi-agents : application de l'éco-résolution à une cellule de production manufacturière. Actes du Congrès *MOSIM'97 : Systèmes de Production et de Logistique*, Rouen, France, 5-6 juin 1997.

Veeramani D., Bhargava B., Barash M.M. (1993) Information system architecture for heterarchical control of large FMSs. *Computer Integrated Manufacturing Systems*, Vol. 6, No 2, Mai, 1993.