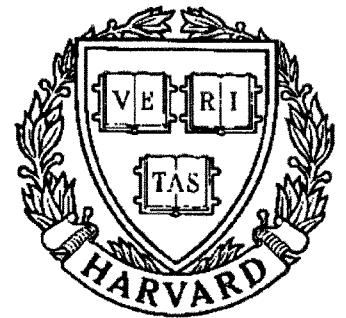


TECHNICAL RESEARCH REPORT



S Y S T E M S
R E S E A R C H
C E N T E R



*Supported by the
National Science Foundation
Engineering Research Center
Program (NSFD CD 8803012),
Industry and the University*

A Step-Wise Specification of a Manufacturing System Using Petri Nets

*by J.F. Claver, G. Harhalakis, J.M. Proth,
V.M. Savi, and X.L. Xie*

A STEP-WISE SPECIFICATION OF A MANUFACTURING SYSTEM USING PETRI NETS

CLAVER J.F., HARHALAKIS G., PROTH J.M., SAVI V.M. and XIE X.L.

INRIA-Lorraine

CESCOM, Technopôle Metz 2000

4 rue Marconi

F-57070 METZ, FRANCE.

Abstract - Our experience, mainly based on two recently conducted real-life studies for some European companies, reveals that a natural way to proceed with the modeling and specification of large manufacturing systems is to decompose the model of the entire system into sub-systems (referred to as modules). Models for modules are developed, and then integrated to model the entire system. Several problems arise in this process. If we express these problems from the users' point of view, they can be summarized by the following two issues: (i) how to decompose the whole system into sub-systems modules in order to have tractable (thus small) models exposing "good" properties? (ii) how then to integrate the module models in order to reach a global model also exposing "good" properties?

Key words - Manufacturing system design, Petri nets, Marked graphs, Event graphs, Integration, Modeling.

1. INTRODUCTION

A natural way to specify and model large size discrete manufacturing systems is to decompose them into small, thus tractable, sub-systems (also called modules), to model each of these modules, and finally to integrate the module models such that the resulting model exposes that the classical "good" properties, namely boundedness, liveness and reversibility. Boundedness is needed to ensure that work-in-process (WIP) remains upper bounded in the

manufacturing system. Liveness guarantees that any operation that can be performed by the system is reachable, regardless of the sequence of previously performed operations. Reversibility is of less importance in the case of manufacturing systems, but it may sometimes be important to be able to come back to the initial state (also referred to in the literature as the home state).

The approach presented in this paper reflects a bottom-up modeling technique. A good review of this kind of techniques is presented in Mu Der Jeng and Franck DiDesare [6]. They report an approach proposed by Agerwala and Choed-Amphai [1]. It starts with a simple structure and, at each step, a set of places of the module models is merged into a place. Other approaches are proposed by Narahari and Viswanadham [8], Krogh and Beck [5], Koh and DiCesare [4]. However, as outlined by the authors, these approaches are based on the invariant method, which makes it difficult to investigate some properties like liveness and reversibility.

The approach presented in this paper is more constrained. However, according to the experience gained in some industrial problems, we claim that the constraints introduced herein are manageable in most real-life cases.

In section 2, we present the properties the model of the module should have, in order to be able to be integrated to a global model with the same properties. We propose to decompose the whole manufacturing system into decision-free modules, which are

modeled using extended event graphs (also called marked graphs).

Section 3 shows how to integrate the above mentioned models in order to reach a system model with the same "good" properties. This integration is performed by an upper level system, called **control system**, which synchronizes the module models in an adequate manner.

Section 4 proposes a small illustrative example. We assume that the reader is familiar with the basic paper of Murata [7], and with the papers of Hillion and Proth [2] and Holloway and Krogh [3].

2. PROPERTIES OF THE MODULE MODELS

In this section, we present the properties of a module model in accordance to the approach proposed in this paper. Basically, a module model is a strongly connected event graph (EG), enriched in the following ways:

- The model can be influenced by exogeneous inputs (controls). These inputs are provided by means of **control places**, as proposed in Holloway and Krogh [3] and in some papers referenced herein. Note that the places of the initial EG are referred to as **state places**.

- The module model is also permitted to receive tokens, representing parts or raw materials, from outside. This is made through the use of **input places**.

- Finally, the module model is allowed to send tokens, representing parts or raw materials, outside the module. This is made through the use of **output places**.

Output transitions of a control place are called **controlled transitions**.

Figure 1 illustrates the case of two machines m_1 and m_2 separated by a three-position buffer B .

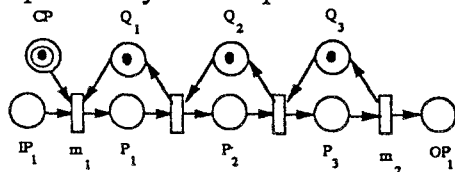


Fig. 1: A module

In the module model presented in figure 1:

- State places are $P_1, P_2, P_3, Q_1, Q_2, Q_3$.
- There is only one control place denoted by CP.
- There is only one input place (resp. output place) denoted by IP_1 (resp. OP_1).

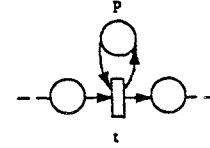


Fig. 2: Firing control model

In the following, we assume that a firing time is associated with each transition (i.e. the module model is a timed graph), and that a transition can begin firing only after the previous firing is complete. This can be represented as shown in figure 2; from here on the loop of figure 2 is assumed to be incorporated in every transition.

The fact that the module model without any input, output and control places is a strongly connected event graph, leads to the fact that any transition has at least one state place in the set of its input places. As a consequence, the following result holds:

Result 1:

If all the control places contain one token each, and if the time between two consecutive arrivals of a token in each input place is upper bounded, then the module model is alive and reversible; the concept of reversibility is restricted to the state places.

Proof:

This is obvious, if we consider the fact that the module model, in which input, output and control places have been removed, is a strongly connected event graph.

Q.E.D.

The following result also holds:

Result 2:

A module model, which is restricted to the state places, is bounded (assuming that the initial marking is finite).

Proof:

The total number of tokens in each elementary circuit of the related strongly connected event graph is constant, and the number of elementary circuits is upper bounded. Thus, result 2 holds.

Q.E.D.

Note that, if each elementary model is based on a strongly connected event graph, as shown above, it exposes the "good" properties, i.e. work-in-process is upper bounded, and any operation as well as the initial state are reachable. Nevertheless, it is impossible to make a choice of a transition to fire inside a module. Thus, choices such as this are left to the control system. We study this problem in the next section.

3. SYNCHRONIZING THE MODULE MODELS: THE CONTROL SYSTEM

The control system is in charge of synchronizing the module models which are derived from strongly connected timed event graphs.

The control system acts in two ways:

(i) it controls the cycle time of each module, by removing a token from one or several non-empty control places, or by placing a token in one or several empty control places;

(ii) it conveys the tokens which appear at the output of a module model to the input places of other module models; a choice of the module to follow is possible if several identical or similar module models are available.

The goal of this section is to characterize the synchronization activities of the control system in order to obtain a complete model that remains the "good" properties presented above. When a module model MM_2 follows a module model MM_1 , it means that there exists a relationship between the output places $O(MM_1)$ of MM_1 and the input places $I(MM_2)$ of MM_2 . If f is this relationship, the following property holds:

$$\bigcup_{p \in O(MM_1)} f(p) = I(MM_2)$$

This means that the number of places in $O(MM_1)$ is greater than or equal to the number of places of $I(MM_2)$.

The complete model is obtained by merging $f(p)$ and p for any $p \in O(MM_1)$ and for any pair (MM_1, MM_2) where MM_2 is a module model following MM_1 .

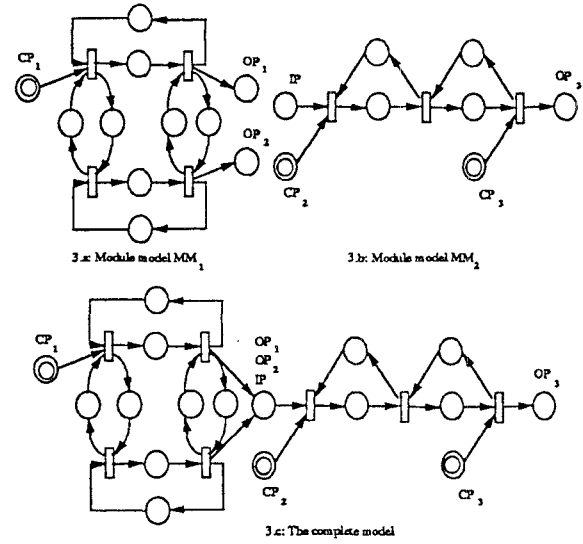


Fig. 3: A linear merging

If (MM_1, MM_2) is a pair where MM_2 is the unique module model following MM_1 , then the only possible control on this pair of modules is the synchronization of the cycle times, assuming that control places exist for at least one of the models. In figure 3, we present the connection of two module models, denoted by MM_1 and MM_2 . MM_1 models a set of two machines (m_1 and m_2) producing two types of parts (p_1 and p_2). The production is cyclic, and the ratios are the same for both parts. MM_2 models a FIFO managed inventory system with two places. The control place CP_1 allows to stop the production, while the control place CP_2 allows to stop the entrance to the stock. The control place CP_3 controls the output of the stock.

Figures 3.a and 3.b represent MM_1 and MM_2 , respectively. Figure 3.c presents the complete model.

In the example presented in figure 3, $f(OP_1) = f(OP_2) = IP$. The control system will control the flow in MM_1 and MM_2 by placing a token in CP_1 and/or

CP₂ and/or CP₃, or by removing a token from CP₁ and/or CP₂ and/or CP₃.

Since the basic models are strongly connected event graphs, we can compute the minimal cycle time among those of the elementary circuits MM₁, MM₂. Assuming that there is always only one token in CP₃ (i.e. parts are removed from stock as soon as possible), and that $C(MM_2) < C(MM_1)$ (i.e. parts are going through stock faster than they are produced), then a possible control is to keep one token in CP₁ and CP₂. If $C(MM_2) > C(MM_1)$, then we will have to stop production (i.e. to remove the token from CP₁) when the stock is full. As a consequence, the proportion of time CP₁ contains one token is $C(MM_1) / C(MM_2)$.

By **linear merging**, we denote the case when only one module model follows the module model under consideration and is merged with it.

The decision of putting a token in an empty control place or removing a token from a control place is made by the control system, based on the state of the physical system.

Let us now assume that a given module model MM₁ is followed by several identical or similar module models. In this case, the control system acts in both ways presented at the beginning of the current section: it controls the cycle time of the module by acting on the control places and by deciding where to convey the tokens which appear in the output places of MM₁ (i.e. in the places of O (MM₁)). If

MM₂¹, MM₂², ..., MM₂^r are the module models

following MM₁ and I (MM₂¹), I (MM₂²), ..., I (MM₂^r) the related input places, then there exist r relationships

f₁, f₂, ..., f_r, f_i between O (MM) and I (MM₂ⁱ) for i = 1, 2, ..., r. The following properties hold:

$$\bigcup_{p \in O(MM_1)} f_i(p) = I(MM_2^i) \text{ for } i = 1, 2, \dots, r$$

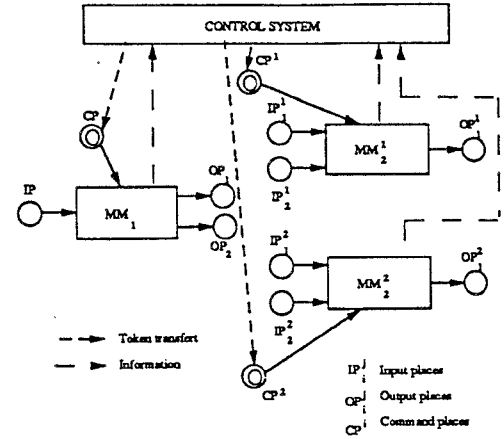


Fig. 4: A dynamic merging

The relationships f_i are activated at the same time, according to the decisions made by the control system. Those decisions are based on the state of the physical system. Decisions are also made to modify, or not, the state of the control places. We call it a **dynamic merging**, because the activation of the f_i relationships connects dynamically the output places

O (MM₁) of MM₁ to the input places I (MM₂ⁱ) of one

of the module models (MM₂ⁱ). Figure 4 represents a dynamic merging in the case when r = 2. Note that a linear merging is a particular case of a dynamic merging. So far, the merging techniques described above have been capable of modeling a variety of physical systems.

The following general results hold:

Result 3:

If the control provided by the control system is such that the number of tokens in each input place is bounded, then the complete model is bounded (i.e. the amount of work in process is upper bounded).

Result 4:

If at any time t it is certain that any module of the system will be used at some time after t, then the complete model is alive.

The properties of the control related to the reversibility of the complete model has still to be established.

In the next section, we illustrate the previous approach using an example derived from a real life study.

4. AN EXAMPLE

The system model in this section is represented in figure 5. Parts arrive at random on a conveyor C capable of holding n parts. They are then processed on one of the two identical machines M_1 or M_2 , and finally they are put in a stocking area S where consumers can pick them up. If the conveyor C is full, then $r < n$ parts are carried to a buffer Q by a crane. They are carried back to the conveyor as soon as there is some space available on it.

To reduce the size of the model, we assume that $n = 3$ and $r = 2$ (i.e. C is able to carry a maximum of three parts, and that two parts are moved to the buffer Q if C is full).

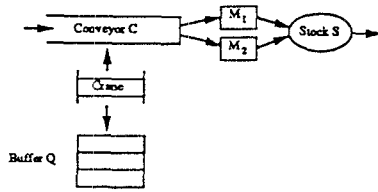


Fig. 5: The system

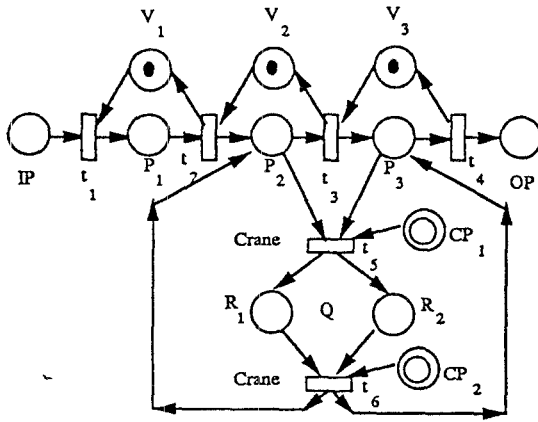


Fig. 6: The module model (C, Q)

The first module we consider is the set composed of the conveyor C and the buffer Q. Assuming that the time needed to carry the parts from C to Q does not depend on the final position of the parts in Q, the module model is given in figure 6. In this model, the

positions of the three parts on the conveyor are P_1 , P_2 and P_3 . Parts arrive in the system through the input position IP. Transitions t_1 , t_2 , t_3 and t_4 are timed to represent the time a part needs to reach the successive positions P_1 , P_2 , P_3 and OP.

OP is the only output place. Transition t_5 is timed to reflect the time the crane needs to carry simultaneously two parts from C to Q. The pair (R_1, R_2) represents the buffer Q. Transition t_6 is timed to reflect the time the crane needs to carry simultaneously two parts from Q to C. CP_1 is used to make the decision to carry two parts from C to Q, and CP_2 is used to make the decision to carry two parts from Q to C. Places CP_1 and CP_2 are control places.

The second and the third modules are M_1 and M_2 respectively. These models are illustrated in figure 7.a and 7.b respectively.

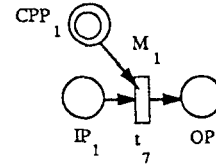


Fig. 7.a: Module model M_1

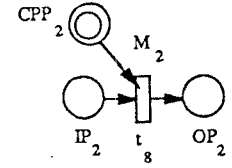


Fig. 7.b: Module model M_2

Fig. 7: Module models corresponding to machines M_1 and M_2

IP_1 (resp. IP_2) is the input place of module model M_1 (resp. M_2). Similarly, OP_1 (resp. OP_2) is the output place of module model M_1 (resp. M_2).

The control place CPP_1 (resp. CPP_2) is used to decide if M_1 (resp. M_2) is in working order or not. The module model corresponding to the stocking area is represented in figure 8. We assume that w places are available in this area. Thus, w tokens are put in place H. The entrance to the stocking area S can be blocked by means of the control place CPQ , by removing the token contained in CPQ .

If f_1 is the relationship between the output place of (C, Q) and the input place of (M_1), then $f_1(OP) = IP_1$. Similarly, if f_2 is the relationship between the output place of (C, Q) and the input place of (M_2), then $f_2(OP) = IP_2$. The global model is shown in figure 9.

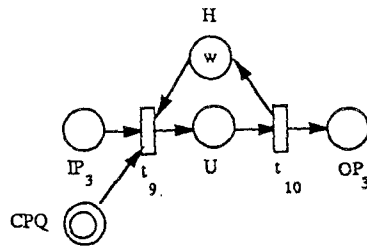


Fig. 8: S Module model

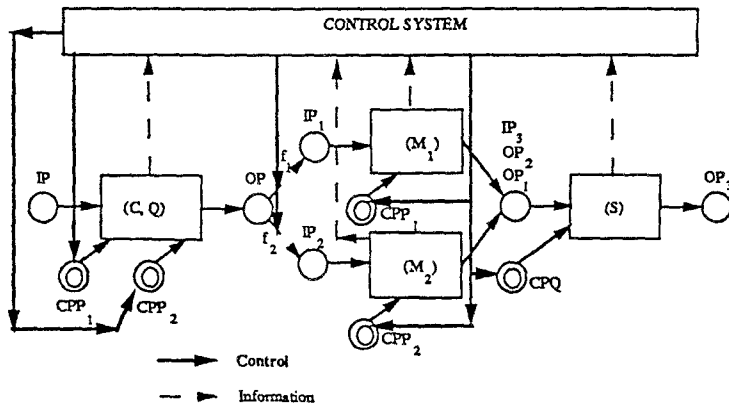


Fig. 9: The global model

5. CONCLUSION

Several real-life applications conducted by the authors in the past showed that the most natural way to model and analyse the respective systems is to split up the whole system into small -and thus tractable- subsystems. We then model these subsystems independently from each other, and integrate these models by means of a so called control system. The control system can be a computer program if the system is fully automated.

The main reason to adopt this bottom-up philosophy is based on the fact that in real-life situations a complex system is composed of subsystems which need different technical backgrounds, and thus are put in the hands of different technical teams. We suggest to consider subsystems whose models are derived from strongly connected event graphs, and which are integrated by means of a control system. Control is achieved through some control places and by activating some relationships in real time. In the various situations we had to deal with so far, the design of the control system was quite easy. The only objective was to

control the flows in the various modules, in order to avoid high work-in-process levels. Further research will be needed on the detailed properties of the control system.

REFERENCES

- [1] AGERWALA T. and CHOED-AMPHAI Y., "A synthesis rule for concurrent systems," *Proc. 15th Design Automation Conf.*, Las Vegas, pp. 305-311, June 1978,.
- [2] HILLION H.P. and PROTH J.M., "Performance evaluation of job-shop systems using timed event-graphs," *IEEE Transactions on Automatic Control*, vol. 34, No 1, pp. 3-9, 1989.
- [3] HOLLOWAY L.E. and KROGH B.H., "Synthesis of feedback control logic for a class of controlled Petri nets," *IEEE Transactions on Automatic Control*, vol. 35, No 5, May 1990.
- [4] KOH I. and DICESARE F., "Transformation methods for generalized Petri nets and their application in flexible manufacturing systems," *WP, ECSE Dep.*, RPI, Troy.
- [5] KROGH B.H. and BECK C.L., "Synthesis of place/transition nets for simulation and control of manufacturing systems," *Proc. IFIP Symp. Large Scale Syst.*, Zurich, August 1986.
- [6] MU DER JENG and DICESARE F., "A review of synthesis techniques for Petri nets," *Int. Conf. on Computer Integrated Manufacturing*, Troy, NY, pp. 348-355, 1990.
- [7] MURATA T., "Petri nets: properties, analysis and applications," *Proc. of the IEEE*, vol. 77, No 4, April 1989.
- [8] NARAHARI Y. and VISVANADHAM N., "A Petri net approach to the modeling and analysis of flexible manufacturing systems," *Annals of O.R.*, vol. 3, pp. 449-472, 1985.

