MFG/PFS Methodology in Manufacturing Industries

Wanderley Liu, Paulo Eigi Miyagi
Lab. de Automação e Sistemas (Mecatrônica)
Universidade de São Paulo,
BRAZIL

Gerhard Schreck
Fraunhofer Institute for Productions
Systems and Design Technologies (IPK)
Pascalstrasse 8 - 9
D - 10587, Berlin
GERMANY

Abstract: MFG (Mark Flow Graph) and PFS (Production Flow Schema) have been used successfully in the development of control architecture for manufacturing systems. This paper is concerned with the application of such modelling tools in the specification of FMS and FAS controllers for an integrated information systems approach. This work is being developed in the context of the ECLA FlexSys project, establishing a cooperative program between USP and IPK.

Keywords: manufacturing systems integration, methodology for design, Petri nets

1. Introduction

The design of FMS/FAS implies the task of integrating the functions of devices such as robots and machines into a manufacturing environment. This task is not trivial, as it deals with such aspects as parallelism and concurrence of complex activities, which cannot be properly treated with the ordinary tools of system theory. MFG [1] and PFS [2] are modelling techniques, suitable for meeting these needs, and defining a design methodology which will be presented in this paper.

The methodology is concerned with the design of FMS/FAS from a control perspective. Actually, it was intended for modelling and specifying control systems for discrete event production systems. In this approach, a system is viewed as a graph composed of a set of activities connected by flows and the control is exerted by properly specifying the dynamic behaviour of this graph.

In a manufacturing environment, two major kinds of flows can be identified, a material flow and an information flow [3]. When specifying the control of the material flow, designers are primarily concerned with the process control, i.e. they have to define the sequence of activities to be performed so that parts should be properly processed. On the other hand, in the specification of the information flow, designers are concerned with the development of software and hardware as well as with the communication network that implements the process control. These are complementary approaches and can both be simultaneously treated by using the MFG/PFS methodology.

In this paper, first the MFG/PFS technique is presented, then the application of this methodology in either material flow or information flow is discussed. At the end of the discussion, it will be shown a preliminary activity planning for the cooperation program between USP and IPK.

2. Mark Flow Graph (MFG)

2.1 MFG/PFS Technique

Mark Flow Graph (MFG) is a modelling technique derived from Petri net theory, which was developed for discrete event systems control [1]. It is composed of the following elements (Figure 1).

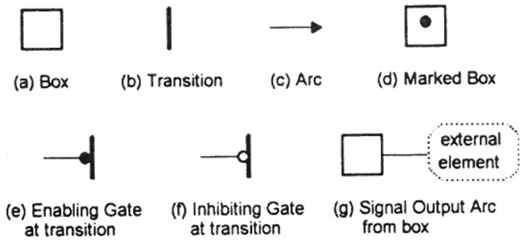


Figure 1. MFG Elements

These elements are connected in such a way that they form a bipartite oriented graph. One major feature of a MFG graph is the capability of modelling the dynamic behaviour of the system. This can be achieved through what is called transition firing. Figure 2 shows the firing of a transition.

Figure 2. Firing of a Transition in MFG

When a transition fires, all the tokens from the input boxes disappear and immediately all the output boxes get marked. A box can contain at most only one token. For a transition to fire, all the input boxes of the transition must be marked and all the output boxes must be empty. Besides , all enabling gates must be associated with marked boxes and all inhibiting boxes have to be connected to empty boxes.

2.2. Production Flow Schema (PFS)

A PFS (2) graph consists of a set of active elements, called activities, and passive elements (distributing elements). The activities are interconnected through oriented arcs, which represent the flows of the system. In PFS, primary and secondary flows are defined. The Figure below shows these elements (Figure 3).

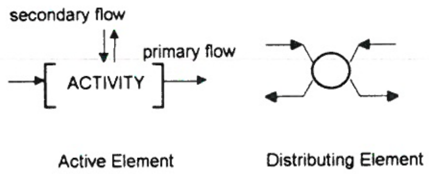


Figure 3. PFS Elements

The proposed methodology is concerned with the task of design by a top-down approach. In the design process, we can identify three main steps, namely problem definition, solution specification and implementation. According to structured analysis, a failure at the earliest steps can cause an improper implementation.

For problem definition, a conceptual model in PFS has been built. In a conceptual model, we can visualise the relationship between activities in a system represented by flows. This conceptual model is then refined into a functional model through MFG, which specifies the control solution adopted for the problem. A specification of MFG is concerned with controlling the flows defined in the PFS.

3. Modelling the Flows in MFG/PFS and Their Control

3.1 Modelling the Material Flow

The following example consists of a station which contains a machine of which task is parts' processing. The part comes to the machine through an input conveyor (Cv1). A robot is placed so that the part can be loaded from the conveyor to the machine. After having processed it, another robot picks the part off the machine and put it on an output conveyor (Cv2). This is shown in Figure 4.

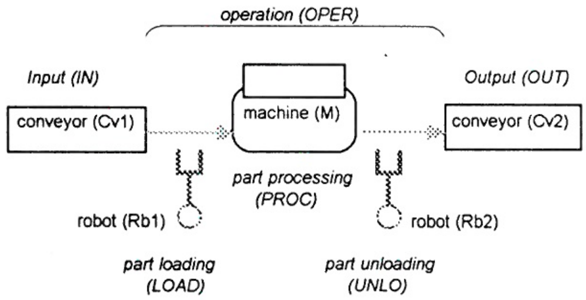


Figure 4. Example of a Material Flow

A material flow can be identified from the above example. From the input conveyor, the part is loaded, then processed, then unloaded to an output conveyor. This situation yields the following conceptual model in PFS (Figure 5), which models the involved activities as well as the material flow of the process.

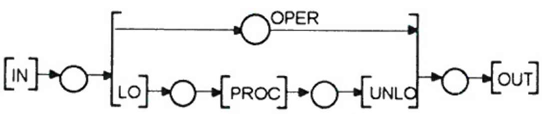


Figure 5. PFS Modelling of the Example

In the proposed methodology, the conceptual model in PFS is refined through a functional model in MFG (Figure 6). Further information on how to obtain a MFG model from a PFS representation can be found in [2]. The marked boxes in Figure 6 define conditions of resources availability for carrying out a certain activity. For instance, the LOAD activity requires that the robot be available so that the task should be executed. When the LOAD activity is finished, both the robot and the conveyor are released, becoming available for further tasks.

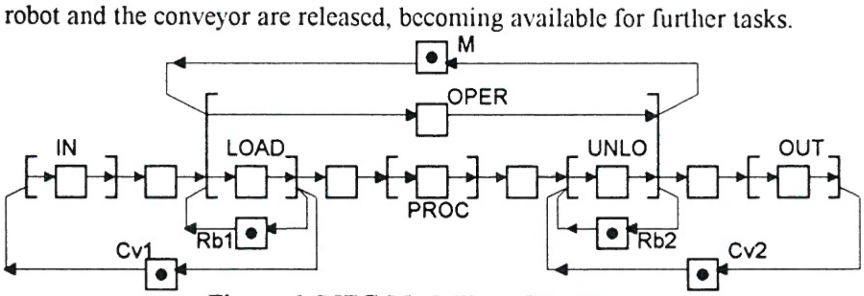
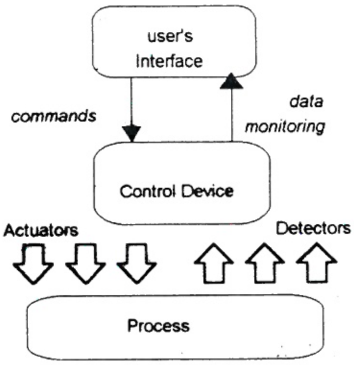


Figure 6. MFG Modelling of the Process

The MFG represents the structure, components and the dynamic behaviour of the system. Thus, Figure 6 can be considered as a representation of the process to be controlled. If the real conditions in the process are properly mapped in the graph, the dynamic evolution of the real process will correspond to the evolution of the tokens in the graph, so the process control can be achieved by controlling the graph evolution.

3.2 Modelling the Information Flow and the Control System Structure

Beside modelling process structure and dynamics, the MFG/PFS methodology can also be used to specify the control system. The Figure below presents an architecture of the general control functions (Figure 7(a)). A control device receives signals from detectors and issues control actions to the process through actuators. It provides monitoring data to the users when a command is received. Figure 7(b) shows a conceptual model for a generic control device in PFS.



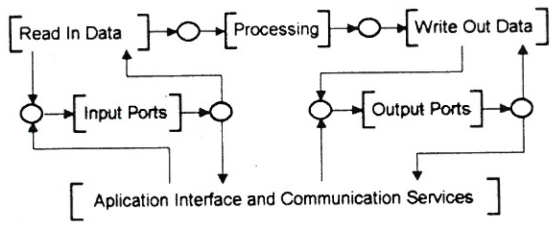


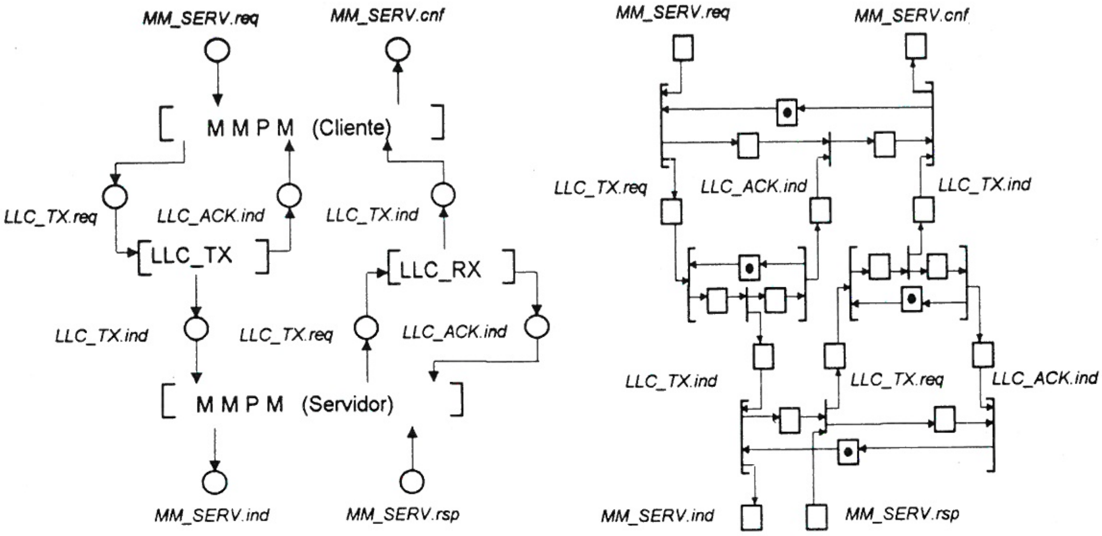
Figure 7 (a). Control System Architecture Figure 7 (b). PFS Model of the Control Device

In this structure, the control device contains in its data structure a MFG/PFS graph, which represents the process to be controlled. The graph is connected to the real process by means of input and output ports. The input ports contain command and sensors data and the output ports contain logical signals to actuators. This structure is suitable for small scale systems, but in case of real FMS/FAS the system is rather complex and different conceptual control levels can be identified. In this environment, a communication network must be available to transport data, so the control device has a module ensuring the interface between the device and the network. This module offers communication facilities to the controller, such as status reading and remote operation on resources.

4. Application Example

The MFG/PFS methodology can also help in the development of communication protocols for industrial networks. In small FMS and FAS, sometimes, the implementation of a proprietary network should be considered in order to meet the communication needs for data transfer. The MFG inherits the capability of Petri nets of modelling machine protocols. Machine protocols are state machines which describe the internal behaviour of a protocol. They must be specified so that they could generate the proper primitives performing the communication task. In this way, open system architectures, as the ISO- recommended ones, can also be considered [4].

Figure 8(a) shows the PFS model of a network architecture for FMS/FAS, considering the primitives exchange defined by ISO. This example is based on a protocol called MMS (Manufacturing Message Specification). The functionality of the MMS is defined by a protocol machine called MMPM (Manufacturing Message Protocol Machine). Figure 8(b) is an MFG model which specifies the internal behaviour of the protocol so that the graph should generate the primitive sequence standardized by ISO. The analysis methods of the MFG theory can be used to validate the model, by verifying its conformity to ISO standards.



(a) PFS model (b) MFG model

Fig. 8 - Models for protocol machines

5. Conclusion

The topics reported in this paper refer to a first step of the co-operative program between USP and IPK within the context of the ECLA FlexSys project. In this first step, an overview of the MFG/PFS methodology was offered. Given the necessity for the material flow control, the methodology has provided an efficient way to specify the process control. On the other hand, modelling the information flow control leads to the specification of hardware and software (and of the communication network) in the control system. Both approaches are complementary and demonstrate the performance of the tool applied in designing FMS and FAS.

The second step of this work is related to the generation of control system software. At this step researches must include the definition of specification languages for programmable controllers. In this case, MFG/PFS provides its own language for programmable controllers, which can be found in [5]. The language must be converted to the controller internal data representation by means of a proper compiler. As each control device has its particular way of representing data, the development of a compiler generator is to be desired. At last, a real application example must be defined. A possible application of this work could be the design of a underwater CIM for petroleum extraction. This application would take advantage of the IPK's experience in the development of prototypes of robots and manipulators in hazardous environments. It would also consider the experience of USP in the discrete event production systems control.

Some remarks of recent researches on MFG will be a proper conclusion. They are dedicated to extending models using individual tokens, as it is a common trend related to Petri net theory. This brings about extensions like Tagged MFG [6] and Extended MFG [7]. These new extensions aim at enhancing the expressiveness power of the tool. On the other hand, application of MFG in information systems is also the subject of several investigations. Particularly these researches are concerned with specifying a control architecture for CIM [8].

REFERENCES

1. HASEGAWA, K. et al, **Proposal of Mark Flow Graph for Discrete System Control.** TRANS. OF SICE, Vol.20, No.2, Tokyo, 1984, pp.122-129 (in Japanese).

2. MIYAGI, P. E., **Control System Design, Programming and Implementation of Discrete Event Production Systems by Using Mark Flow Graph.** Doctoral Thesis. Graduate School of Science and Engineering, Tokyo Institute of Technology, Tokyo, 1988, 125 p.

3. HASEGAWA, K. et al, **Simulation of Discrete Production Systems Based on Mark Flow Graph. SYSTEM SCIENCE,** Vol.13, Nos.1-2, Warsaw, 1987, pp. 1-22.

4. LIU, W. , **Application of MFG/PFS Methodology in the Development of Information Systems for Manufacturing Industries.** Master Dissertation. Escola Politécnica da Universidade de São Paulo (in Portuguese), São Paulo, 1993, 118 p.

5. MIYAGI, P. E. et al, **A Programming Language for Discrete Event Production Systems Based on Production Flow Schema and Mark Flow Graph,** TRANS. OF SICE, Vol.24, No.2, Tokyo, 1988, pp. 183-190.

6. HASEGAWA, K. and TAKAHASHI, K., **Proposal of Tagged Mark Flow Graph for Flexible Maufacturing Systems**, Proceedings of the 30th Joint Conf. on JAACE, Society of Automatic Control Engineers, Osaka, 1980, pp. 453-456.

7. SANTOS, F. D. J., **Proposal of Extended MFG for the Modelling and Control of Integrated Manufacturing Systems,** Master Dissertation. Escola Politécnica da Universidade de São Paulo (in Portuguese), São Paulo, 1993.

8. CAMARINHA-MATOS, L.M. and OSÓRIO, L.O., **Information Based Control Architecture for CIM.** Proceedings of IFIP Towards World Class Manufacturing Conf., Phoenix, USA, 1993, 14p.