Petri net control of an automated manufacturing cell

Thomas Boucher

Advanced Manufacturing Engineering

Computers ind. Engng Vol. 17, Nos 1-4, pp. 459-463, 1989
Printed in Great Britain.
0360-8352/89 $3.00 + 0.00

PETRI NET CONTROL OF AN AUTOMATED MANUFACTURING CELL

Thomas O. Boucher, Mohsen A. Jafari, and Glenn A. Meredith

Department of Industrial Engineering
Rutgers University
P.O. Box 909, Piscataway, New Jersey 08855

ABSTRACT

This paper discusses the Petri Net approach to real time production control. Petri nets were developed to model concurrent and asynchronous systems. Having modeled a machining cell or a production system as a Petri net, one can analyze the effects of controller behavior on the system before implementing the controller.

Petri net principles are illustrated for a machining cell being implemented in the Manufacturing Automation Laboratory at Rutgers University. Observations are made concerning the difference between a Petri Net Controller and a Programmable Logic Controller for the same application.

INTRODUCTION

Flexible automation is playing an increasingly important role in improving the productivity of manufacturing industry. Recent advances in computer technology have accelerated the realization of flexible automation. The use of computers in manufacturing, such as CNC machine tools or programmable robots adds programmability and thus versatility into manufacturing systems.

A common device used for coordination (sequencing of the events) on the shop floor is the programmable logic controller (PLC). In recent years microcomputers have also been used for this purpose.

In principle, a batch manufacturing system is a discrete event system where parallelism or concurrency, aynchronous behavior, and conflicts are quite common. It has been widely accepted that Petri nets (PN) are appropriate formalism to represent the flow in these systems. Moreover, Petri nets also provide us with appropriate means of analyzing such systems.

Petri nets were originally introduced by Petri in his Ph.D. dissertation [1962], where he formulated a basis for a theory of communication between asynchronous components of a computer system. Over the last two decades, there have been enormous extensions to the original work of Petri, both in application and theory. On the theoretical side, the emphasis was on extending the modeling power of Petri nets and on the formulation of a mathematical framework in which different properties of a petri net can be analyzed. On the application side, however, Petri nets have been used for specification, verification, control, and performance evaluation of computer systems, communication protocols, and (most recently) manufacturing systems.

In this paper, we intend to illustrate a Petri net-based real time controller for a machining cell located in the Manufacturing Automation Laboratory at Rutgers University. We also discuss some issues concerning the relationship between a Petri net controller and a PLC.

APPLICATIONS BACKGROUND

For the last several years the National Bureau of Standards has been developing a manufacturing system (AMRF) together with its controller (Jones and McLean [1986]). This real time controller has a hierachical structure with each level modeled as a finite state machine (Albus et al. [1981]). Finite state machines are a special case of Petri nets. A major problem with state machines is that, in the implementation stage, it requires the state of the whole system as opposed to Petri nets where only a local state is needed. In France, a PN net like representation method, GRAFCET, was proposed as a standard specification of sequence controllers (L'AFCET [1977]). In Japan, Hitachi developed a commercial workstation controller (SCR), which is based on an enhanced PN and allows for input from sensory devices (Komoda et al. [1985] and Murata et al. [1986]). National Research Council of Canada has also been working on a manufacturing technology center with a PN based real time controller (Merabet [1986]).

PETRI NETS AND PROGRAMMABLE LOGIC CONTROLLERS: AN EXAMPLE

It is of interest to illustrate certain differences that exist in the implementation of a cell controller using the formalisms of a Petri net as opposed to a programmable logic controller (PLC) The example used is the simple machining cell of Figure 1, which contains a CNC lathe attended by a robot, an input parts feeder, an output conveyor, and a cell host controller. The CNC lathe and robot have their own controllers which are dedicated to programmed operations and communication with the cell host. The host computer is responsible for overall cell control and initiating action by the robot and/or the lathe. Table 1 lists input and output channels of the host computer and their relationship to units within the cell.

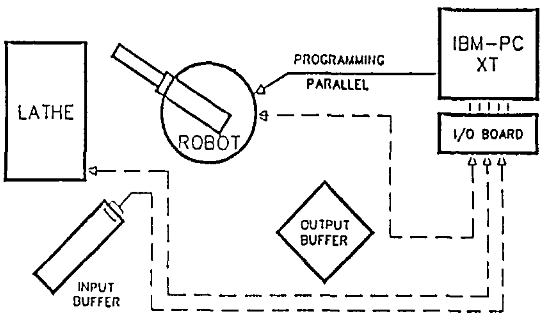

Figure 1. A ROBOT ATTENDING A LATHE

TABLE 1

Inputs and Outputs of Cell Host Corresponding to Figure 1

Input Channel	Channel Activated by	Message
0	Input Feeder Sensor	Workpiece available at feeder
1	Lathe	Machining cycle complete
2	Robot	Loading cycle complete
3	Robot	Unloading cycle complete
Output Channel	**Connected to**	**Message**
0	Robot	Load lathe; disable I3
1	Robot (lathe)	Unload lathe; (disable I1)
2	Lathe (robot)	Start lathe; (disable I2)

It is possible to control the cycle of events in the cell using a PLC as host. Figure 2 is a ladder representing the control logic, where I denotes external inputs and C denotes internally set contacts. Rung 1 initializes the cycle by setting O1 (unload lathe) of Rung 2. The response I3 (unloading cycle complete) enables C2 of Rung 3 when the condition I0 (workpiece available at feeder) is true. This enables O0 (load lathe) at Rung 4. Finally, the response I2 (loading cycle complete) energizes C3, which enables O2 (start lathe) at Rungs 5 and 6. The cycle repeats with the response I1 from the lathe.

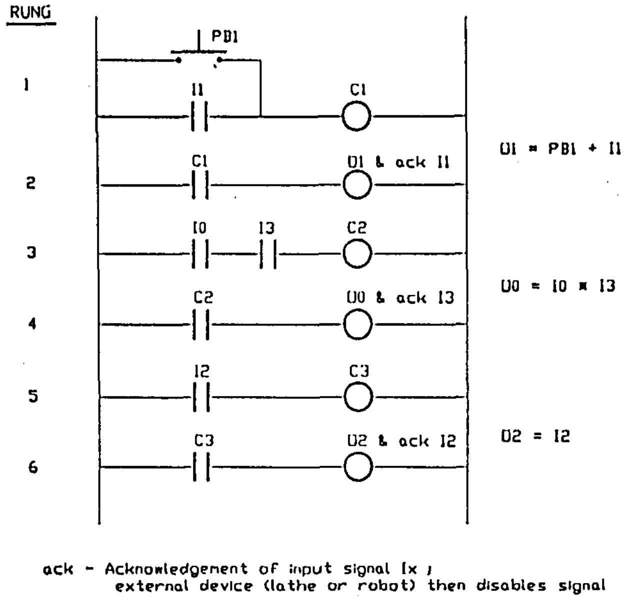

Figure 2. PLC IMPLEMENTATION OF FIGURE 1 AND TABLE 1

In this application, the PLC is simply handling communications and controlling the state of the system as opposed to operating the equipment in the system. Figure 2 is to be used later to illustrate certain aspects of a Petri Net.

The order of events, E, in Figure 2 can be partitioned into state conditions, P, and transition, t. A listing of the cycle of events might appear as follows:

Event No.	Activity	Type
E1	Lathe machining	P
E2	Request for unloading	t
E3	Robot unloading lathe	P
E4	Request for loading	t
E5	Robot loading lathe	P
E6	Start lathe	t
E1	Lathe machining	P

This events sequence can be used to construct a PN by representing states of the system as nodes (called places, P) and transitions between states as bars (called transitions, t). Directed arcs are used to indicate directional sequence. The PN is shown in Figure 3.

Figure 3 illustrates the interpretation of each transition. It is shown here that each transition may be associated with external inputs or outputs. The transition of a PN is fired when its input conditions are satisfied. This is shown by the boolean expressions in Figure 3. Transition t_1 fires when a token is in P1 (P1 is true) and I1 is enabled. When a transition fires, the appropriate output is enabled, tokens are removed from all input places and tokens are put into all output places. In software implementation, the boolean expressions are processed as their conditions become true. A place table is updated to keep track of the position of tokens; a transition table keeps track of which transitions have fired.

Comparing the boolean expressions of Figures 2 and 3 shows that the PN uses information concerning places, which is not normally encoded in the PLC. Table 2 shows the events list for implementations of Figure 2 and Figure 3. In Table 2, inputs and outputs are the same and transitions are analogous to contacts. The distinction is the additional information in PN provided by places. Under the scheme of I/O 's used in this example, the additional information seems redundant. In fact, this can be one of the advantages of PN implementation since keeping track of the system state in software can assist in avoiding ambiguities from external inputs.

TABLE 2
Comparison of States of the System
Figure 2 vs. Figure 3

PLC Events	Inputs I0	I1	I2	I3	Last Coil Energized C1	C2	C3	Outputs O0	O1	O2
1. Lathe Machining					0	0	1	0	0	1
2. Request for unloading		1			1	0	0	0	1	0
3. Robot unloading lathe					1	0	0	0	1	0
4. Request for loading	1			1	0	1	0	1	0	0
5. Robot loading lathe					0	1	0	1	0	0
6. Start lathe			1		0	0	1	0	0	1
1. Lathe machining					0	0	1	0	0	1

PN Event	I0	I1	I2	I3	Last Transition Fired t1	t2	t3	Places P1	P2	P3	O0	O1	O2
1. Lathe machining					0	0	1	1	0	0	0	0	1
2. Request for unloading		1			1	0	0	1	0	0	0	1	0
3. Robot unload lathe					1	0	0	0	1	0	0	1	0
4 Request for loading	1			1	0	1	0	0	1	0	1	0	0
5. Robot loading lathe					0	1	0	0	0	1	1	0	0
6. Start lathe			1		0	0	1	0	0	1	0	0	1
1. Lathe machining					0	0	1	1	0	0	0	0	1

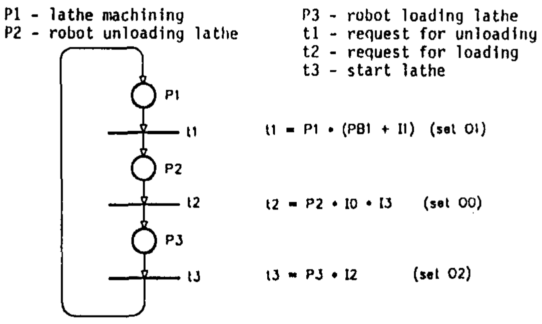

Figure 3. PN implementation of figure 1 and table 1

For example, if, due to a malfunction at the robot, I2 and I3 and I0 were simultaneously enabled, the lathe would be started at the same time the robot would be loading it. If this were to happen under PN control, the position of the token would govern which one of those two activities would occur.

EXTENDED IMPLEMENTATION OF PN CONTROL

For the simple example of Figure 1, the PN can be implemented with fewer inputs if a larger information set is carried within the PN software. This can be done by enlarging the event space and including the activities of the robot and the workpieces.

Figure 4 and Table 3 are a redefinition of the model. In Figure 4 we have added five places and three transitions and we have eliminated one input. Note that, in some cases, the firing of the transitions is based on data carried entirely within the net (t2 and t4).

TABLE 3
Inputs and Outputs of Cell Host
Corresponding to Figure 1

Input Channel	Channel Activated by	Message
0	Input feeder	Workpiece available
1	Lathe	Machining cycle complete
2	Robot	Robot done with lathe

Output Channel	Connected to	Message
0	Robot	Load lathe
1	Robot	Unload lathe
3	Lathe	Start lathe machining cycle

P1 - lathe machining
P2 - lathe awaits unloading
P3 - lathe being unloaded
P4 - lathe awaits loading
P5 - lathe being unloaded
IN - part in feeder

R - robot available
t1 - lathe completes operation
t2 - robot starts unloading lathe
t3 - robot completes unloading
t4 - robot starts loading lathe
t5 - robot completes loading
t6 - new part arrives at sensor

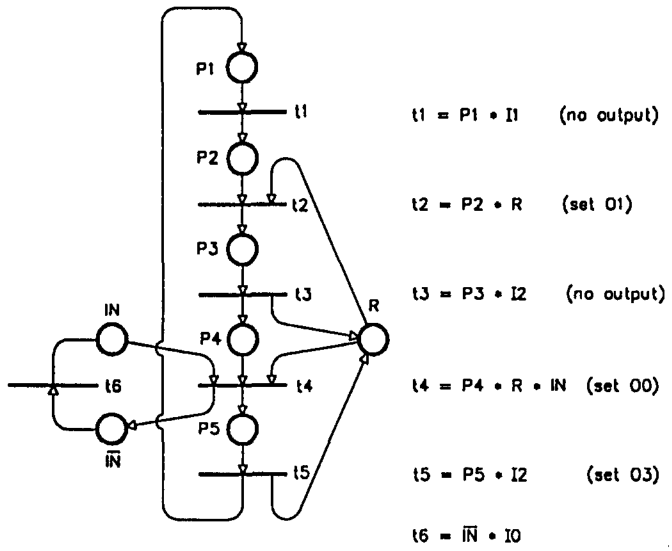

Figure 4. PN IMPLEMENTATION OF FIGURE 1 AND TABLE 3

SUMMARY

In this paper we have illustrated the application of a Petri net to the control of a machining cell. We have also pointed out some advantages of Petrinet over more traditional control methods. In particular, Petrinet has a natural representation of the state of the system, which provides information that is not available when programming depends exclusively on inputs and outputs.

REFERENCES

Albus, J. S., Barbera, A. J., and Nagel, R. N.: "Theory and Practice of Hierarchical Control," 23rd IEEE Computer Society Int'l Conference, September 1981.

Groupe de Travail Systemes Logiques de l'AFCET: "Pour Une Representation Normalisee du Cahier des Charges d'un Automatisme Logique (CRAFCET)," Automatique et Informatique Industrielles. No. 61, pp. 27-32, Nov. 1977.

Jones, A. T., and McLean, C. R.: "A Proposed Hierarchical Control Model for Automated Manufacturing Systems," J. of Manufacturing Systems, vol. 5, No. 1, 1986.

Komoda, N., Murata, T., and Matsumoto, K.: "Petri-Net Based Controller: SCR and its Applications in Factory Automation," Proceedings of ISCAS 85.

Merabet, A.: "Synchronization of Operations in a Flexible Manufacturing Cell: The Petri Net Approach," Journal of Manufacturing Systems, volume 5, number 3, pp. 161-169, 1986.

Murata, T., Komoda, N. and Matsumoto, K.: "A Petri Net Based Controller for Flexible and Maintainable Sequence Control and its Applications in Factory Automation," IEEE Transactions on Industrial Electronics, pp. 1-8, volume IE-33, number 1, February 1986.

Petri, C.: "Kommunikation mit Automaten." Ph.D. Dissertation, University of Bonn, Bonn, West Germany, 1962.