Modular Approach to Petri Net Modeling of Flexible Assembly System

T. Y. Park and B. K. Choi

IE Dept, KAIST
P.O. Box 150 Cheongryang, Seoul, Korea

Abstract

Presented in the paper is a systematic approach to constructing a Petri net model of FAS (flexible assembly system). Petri net is widely used in modeling automated manufacturing systems. But, it found to be very difficult for an FA engineer to build a correct model of an FAS with Petri net symbols (ie, place, transition, and token) from the beginning. An automated manufacturing system in general is built from a set of "standard" hardware components. An FAS in particular is usually composed of assembly robots, work tables, conveyor lines, buffer storages, part feeders, etc. In the proposed modeling scheme, each type of standard resources is represented as a standard "module" which is a sub Petri net. Then, the model of a FAS can be conveniently constructed using the predefined modules the same way the FAS itself is built from the standard components. The network representation of a FAS is termed a JR-net (job resource relation net) which is easy to construct. This JR net is then mechanically converted to a formal Petri net (to simulate the behavior of the FAS). The proposed modeling scheme may easily be extended to the modeling of other types of automated manufacturing systems such as FMS and AS/RS.

1. Introduction

Proposed in the paper is a modular approach to systematically constructing a Petri net model of an FAS (flexible assembly system). In the proposed modular approach, a set of "primitives" (standardized components of the FAS) is predefined and then the entire FAS is described with relevant primitives by specifying "relations" among them. This modular concept is not new and in fact has widely been employed in diverse disciplines such as software engineering, solid modeling, and system modeling.

In the field of system modeling, Righini[1] and Pflug and Prohaska[2] have employed modular network methods in "discrete event system" modeling. Righini[1] uses a *subnet concept* where a subnet is a Petri net which shares one or more transitions with other subnets. A subnet is associated with every unit in the manufacturing system and the description of all links between units is used to connect all corresponding subnets. Pflug and

Prohaska[2] proposes a *entity-connection approach* where ENTITY-icons and CONNECTION-icons are introduced. The system model is then constructed by describing a set of many-to-many relationships between ENTITY-icons and CONNECTION-icons.

The modular modeling scheme proposed in the paper is similar to these approaches. But, it is unique in that a set of predefined "standard primitives" are employed. This scheme is based on the observation that an FAS is usually built from a set of "standard components" such as assembly cells, robots, and conveyors. The network description using the standard primitives is termed a JR-net (job-resource relation net). Under the proposed scheme, an FAS under consideration is modeled by using JR-net symbols and then the resulting JR-net is mechanically converted to a comprehensive Petri net model. As will be seen later it is very simple to construct a JR-net of an FAS because it resembles the physical configuration of the FAS.

2. JR-Net (Job Resource Relation Net) for FAS Modeling

Typical hardware components in an FAS are assembly robot, handling robot, APC, work table, conveyor, stocker, etc. These hardware components of an FAS may be grouped into the following:

1) machine without work table (assembly robot, APC, etc.),
2) work table on which assembly operations are performed,
3) machine with work table (assembly cell),
4) conveyor, and
5) storage (stocker).

These hardware components are called "resources" through which "jobs" flow. These resources are related with each other *via* job processing requirements. A JR-net is a graphical representation of these relationships.

A set of basic JR-net symbols for FAS modeling is depicted in Figure 1. A "machine" without work table is denoted by a circle, anda "work table" on which assembly operations are performed (by machines) is denoted by a rectangle. A machine with a work table is denoted by a circle containing a square in it. A rounded rectangle is used for both conveyor and storage. A job

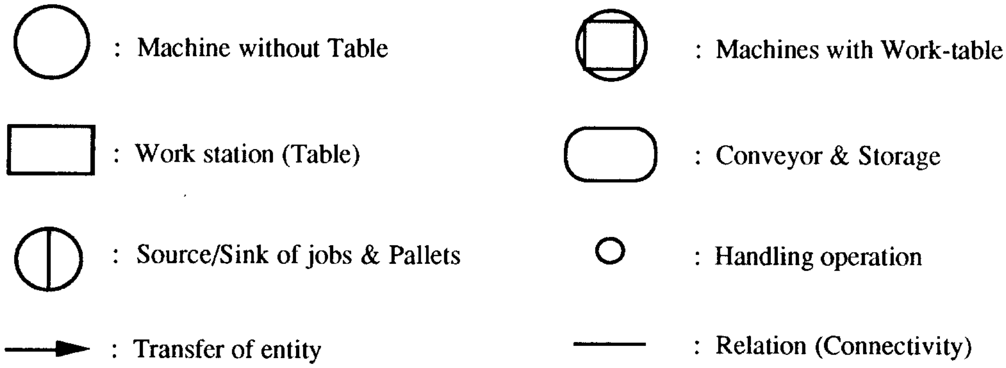

Figure 1. Basic JR-net Symbols

transfer is denoted by directed arcs (arrows), and a small circle on a directed arc denotes a handling operation. A solid line from a big circle (machine) to a small circle on an arrow (handling operation) indicates that the operation is performed by the machine. A divided-circle shape represents a source/sink of a job or a pallet. The basic JR-net symbols introduced in Figure 1 are far from comprehensive. One may introduce additional symbols as required.

Shown in Figure 2 are basic JR-net "primitives" useful for describing FAS. Figure 2-(a) represents a "machine with work table" in an assembly line. The machine **M** has a processing time of "t". The arrow to **M** indicates a loading operation and the arrow from **M** an unloading operation. The primitive denoting an "assembly operation on a table **T**" by a robot **R** is shown in Figure 2-(b). Handling (transfer) of a job between two tables by a robot **R** is depicted in Figure 2-(c), and the transportation of a job via a conveyor **C** with a capacity of "q" (and conveying time of "t") is shown in Figure 2-(d). The conveyor primitive in (d) would become a "storage primitive" if conveying time is neglected.

Now we show how to build a JR-net using the primitives in Figure 2. Shown in Figure 3 is a portion of FAS (flexible assembly system) reported in the literature[3]. The FAS consists of

- 6 work tables (T1 to T6),
- 4 assembly robots (R1 to R4),
- 4 queue buffers (B1 to B4) and a queue conveyor (C1), and
- a few feeders (not to be included in the model).

Eight assembly operations (A1 through A8) are performed on the FAS line as indicated in the figure:

Assembly operations A1,A2 : on table T1 by robot R1.
Assembly operation A3 : on table T2 by robot R1 or R2.
Assembly operation A4 : on table T3 by robot R2.
Assembly operation A5 : on table T4 by robot R3.

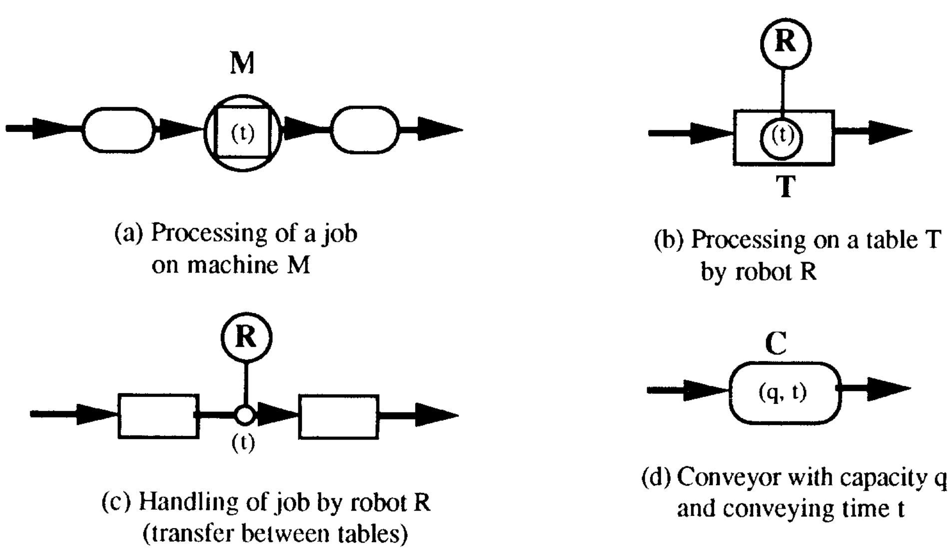

Figure 2. Basic JR-net Primitives

Assembly operations A6,A7 : on table T5 by robot R3 or R4.
Assembly operation A8 : on table T6 by robot R4.

The flow of a job in the FAS line is as follows:

T1(a1,a2)➡B1➡T2(a3)➡B2➡T3(a4)➡C1➡T4(a5)➡B3➡T5(a6,a7)➡B4➡T6(a8).

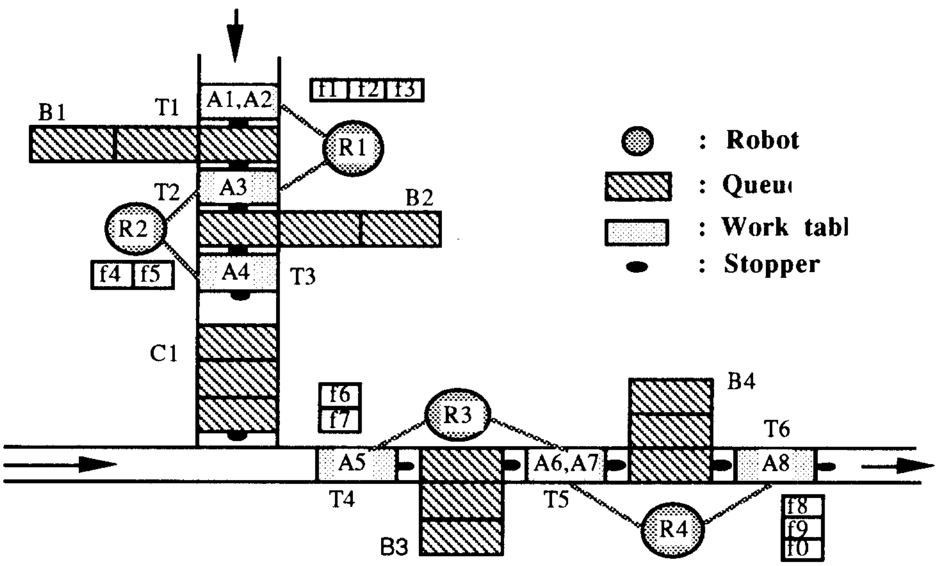

Figure 3. A Simple FAS (Flexible Assembly System)

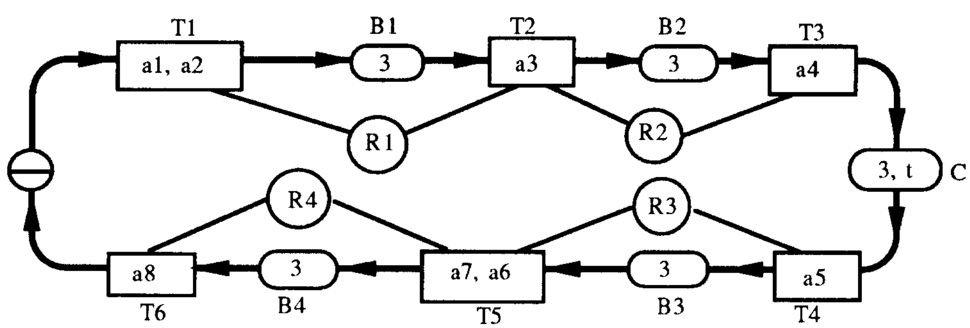

Figure 4. JR-net of the FAS in Fig. 3

A JR-net model of the FAS of Figure 3 is given in Figure 4 which is nothing but a logical duplicate of the physical configuration of the FAS. The "job" starts from a source/sink node and moves through a sequence of "resources" (T1➡B1➡T2➡B2➡ • • • ➡T6) to return to the source/sink node. To each work table served by a robot a line (undirected arc) is drawn from the robot resource.

Associated with each assembly operation is a time a_i. The number "3" in each buffer storage indicates its capacity. Recall that each "transfer" of a job (indicated by an arrow) has its own handling time which is not shown on the

JR-net. The conveyor C1 is fully described by its capacity "3" and conveying time "t" which is the time for a job to move from the "load position" to the "unload position" on an empty conveyor.

The above example is a straightforward one, but it does show the usefulness of the JR-net modeling. An FAS having a very complex structure may also be easily modeled as long as one has a complete set of primitives corresponding to the hardware components of the FAS.

3. Petri Net Representation of JR-net Models

This section presents a method of systematically converting a JR-net to a Petri net. The reader who is not familiar with Petri net is referred to the classical book on the subject by Peterson[4]. In the following, we will first introduce *sub Petri nets* corresponding to the JR-net primitives given earlier in Figure 2. The Petri net to be used is an extended net with *colored tokens* and *timed transitions*. As a general rule, a color (of tokens) is associated with each job type and each resource in the JR-net, and a timed transition is associated with an operation. Sometimes a *timed place* is used for brevity (a timed place is readily replaced by a place-transition-place sequence).

Shown in Figure 5 are *sub Petri nets* (sub-PN) for the JR-net primitives given in Figure 2. The JR-net primitive "processing of a job on a machine" (Fig.2-a) becomes a sub-PN as shown in Figure 5-(a). A separate colored token is

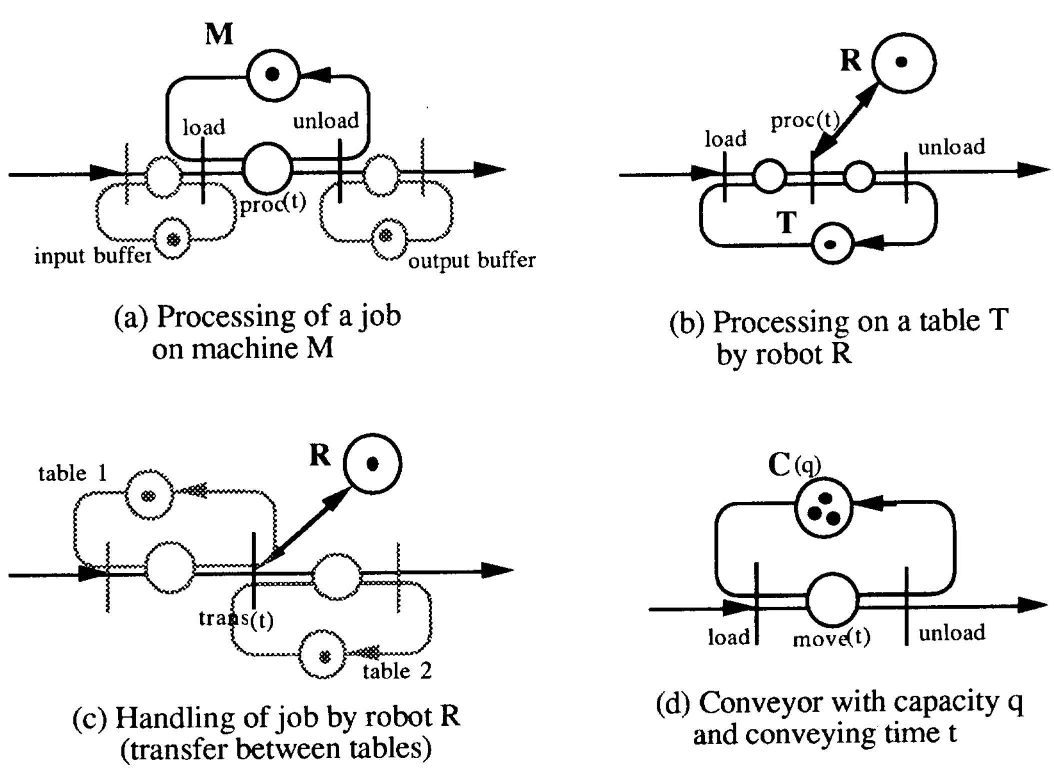

Figure 5. Sub Petri Nets for the Basic JR-net Primitives in Fig. 2

assigned to each of the resources (machine, input buffer, and output buffer).

A job transfer becomes a timed transition, and the processing operation becomes a timed place. The JR-net primitives "processing of job on table by robot" (Fig.2-b) and "transfer of job between tables by robot" (Fig.2-c) are similarly converted to sub-PNs as shown in Figure 5-(b) and (c). "Transportation of job by conveyor" (Fig.2-d) is converted to the sub-PN shown in Figure 5-(d) where the number of tokens denotes the capacity "q" of the conveyor. The conveying time "t" is captured in the timed place "move(t)".

Sometimes, it is convenient to introduce "extended" JR-net primitives. Two examples of extended primitives and their sub-PNs are shown in Figure 6. Depicted in Figure 6-(a) is a work table where two assembly operations (a1,a2) are performed by a robot. In the JR-net primitive, there are four operations (loading, A1, A2, and unloading) which become timed transitions in Petri net. Associated with the resources (R,T) are separated colored tokens. Shown in Figure 6-(b) is another JR-net primitive where an assembly operation can be performed by either of the two robots. In this case, the assembly operation becomes a pair of timed transitions in parallel as can be seen in the figure.

With the mapping relationships between JR-net primitives and sub-PNs given in Figure 5 and 6, the FAS JR-net of Figure 4 may trivially be converted to a Petri net graph as shown in Figure 7. Starting from the source/sink node, a "generate" transition followed by a place is obtained. The arrow into the table T1 of the JR-net becomes the "T1.INPUT" transition, and the primitive T1 is converted to a sub-PN by using the relation in Figure 6-(a). The arrow from T1 to B1 becomes the "B1.LOAD" transition, and so on.

There are 24 timed transitions and 35 places (one of them is a timed place) in

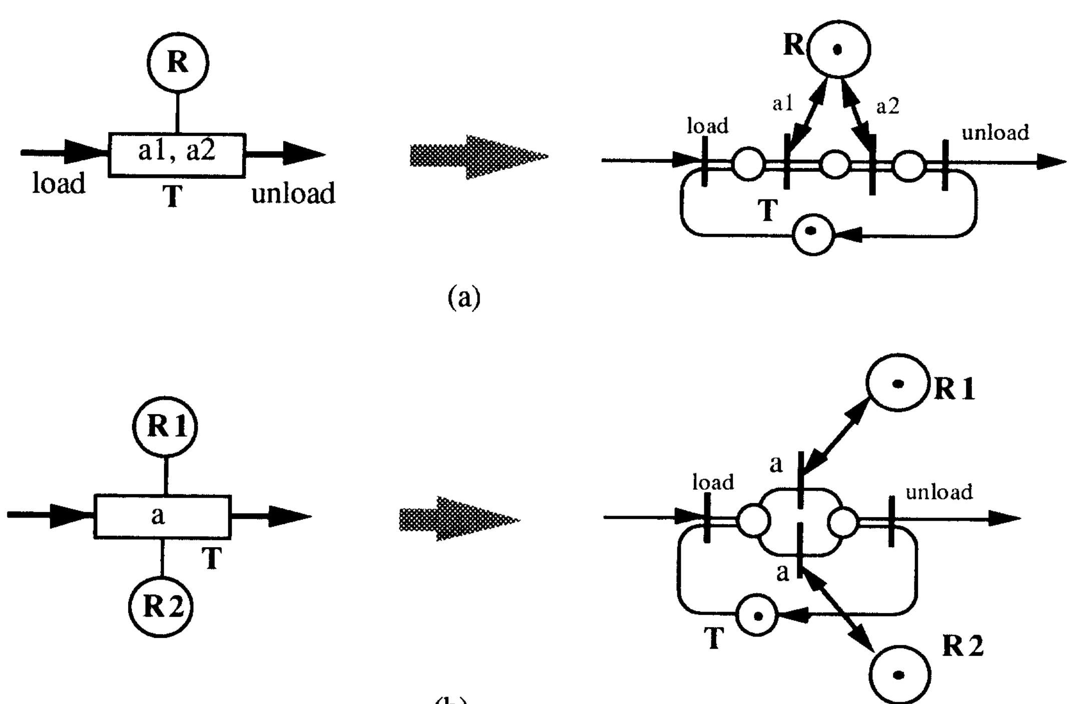

Figure 6. Extended JR-net Primitives and Sub Petri Net

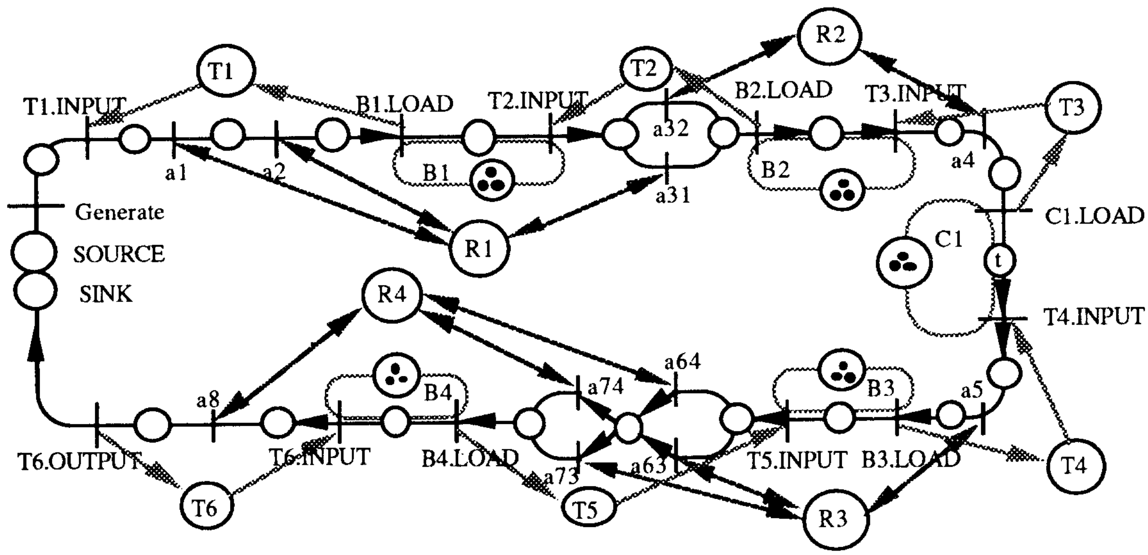

Figure 7. Petri Net Graph of the JR-net in Fig. 4

the Petri net. Recall that a timed place can be decomposed into a place-transition-place sequence making the numbers of transitions and places to be 25 and 36, respectively. The FAS consists of 15 resources (6 tables, 4 robots, 4 buffers, and a conveyor) assembling one type of jobs. Thus, there are 16 types of colored tokens in the Petri net. The resulting Petri net is a very simple one in terms of "real life" problems, but it is not a simple task to construct such a Petri net directly from the FAS description in Figure 3.

4. Conclusions and Discussions

Among the various modeling tools, Petri net is gaining more popularity mainly because of its theoretical soundness and modeling power[1,5]. However, it is very difficult for a practitioner to build a Petri net model of a real life FAS. The proposed modeling scheme found to be very convenient. The JR-net is a "natural" view of the system, and as a result, it is easy to construct and verify. The JR-net model can serve as a convenient tool for communication with FA engineers.

The proposed modeling scheme may be useful in developing a computer simulation package. The "user" describes the system in terms of JR-net and then the "package" converts the JR-net into a Petri net which is "executed" by a Petri net executor. The same modeling scheme may easily be extended to other types of manufacturing systems (by introducing additional JR-net primitives). Further research is needed in formalizing the JR-net structure. JR-net is an *ad hoc* graph consisting of a set of nodes and a set of arcs. In a way, it is related to the "process view" of a real world[6].

5. References

1 G. Righini, FIRST: A Petri net-based system for simulation of complex

distributed manufacturing system, *CIMS*, Vol 3, No 4 (1990).
2 G. Pflug and M. Prohaska, The entity-connection approach to modeling and simulation, *Simulation*, October 1990.
3 S.K. Son, B.K. Choi, T.U. Park, and H.Lee-Kwang, Resolution of Multiple Conflicts in the Petri-net Model of FAS, *IJCIM*, Vol 4, No 4 (1991).
4 J.L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall.
5 S.H. Teng and J.T. Black, Cellular manufacturing system modeling: The Petri net approach, *Journal of Manufacturing Systems*, Vol 9, No 1 (1990).
6 B.P. Zeigler, System Theoretic Representation of Simulation Models, *IIE Transactions*, Vol 16, No 1 (1984).