The International Journal of Flexible Manufacturing Systems, 8 (1996): 159–182

Object-Oriented Graphical Modeling of FMSs

BYOUNG K. CHOI
Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, 373-1, Kusung-dong Yusong-gu, Taejon, Korea

KWAN H. HAN
Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, 373-1, Kusung-dong Yusong-gu, Taejon, Korea

TEA Y. PARK
Department of Automation and Design Engineering, Korea Advanced Institute of Science and Technology, 373-1, Kusung-dong Yusong-gu, Taejon, Korea

Abstract. Presented in the article is a method for constructing a graphical model of an FMS by using a new modeling tool called JR-net (Job Resource relation-net). JR-net is an object-oriented graphical tool for modeling automated manufacturing systems (AMSs), such as FMSs, FASs, and AS/RSs. As with the object-oriented modeling paradigm of Rumbaugh et al. (1991), the JR-net modeling framework supports the three stages of models: static layout model (object model); job flow model (functional model); and supervisory control model (dynamic model). In this article, the existing JR-net structure (Park 1992, Han et al., 1995) is extended further to make it a graphical tool for FMS modeling. Using the extended JR-net, a step-by-step procedure for constructing a graphical model of FMSs is presented. Also addressed are issues of classifying FMSs in terms of their generic functions and of utilizing the JR-net model of FMSs.

Key Words: FMS, graphical modeling, object-oriented, supervisory control.

1. Introduction

Traditionally, the main purpose of FMS modeling has been to analyze system performance. However, in order to support the emerging concepts of *virtual manufacturing*, it has become more important to have a "synthesis tool" for FMS design. In order to support this new requirement, a modeling tool is required to be a *prototyping tool* for FMS design engineers as well as a *communication tool* for managers and other personnel involved. The purpose of this article is to present a new method for constructing a graphical model of an FMS supporting these requirements.

The FMS modeling method presented in the article is based on a new modeling tool called *JR-net* (job resource relation-net). The basic structure of JR-net was first proposed by Park (1992) as an aid to constructing *modular* Petri net models, and has been enhanced somewhat to support an *object-oriented modeling* framework (Han et al., 1995). In this article, the existing JR-net structure is extended further to make it a graphical tool for FMS modeling.

The majority of existing research on graphical modeling of FMS is based on Petri nets (Wadhwa and Browne, 1989, Teng and Black, 1990, Cossins and Ferreira, 1992, Liu and

Wu, 1993, Zhou and DiCesare, 1993, Yim and Barta, 1994, Jeng 1995). Event graph and ACD (activity cycle diagram) have also been used in FMS modeling (Sargent, 1988, Carrie, 1988). The advantage of these *formal* modeling tools is that they are well-structured and may be easily converted to a simulation program. However, they all suffer from modeling difficulty and near-intractability: it is very difficult for an FA (factory automation) engineer to describe a real-world FMS with these modeling tools, and the resulting model would become too abstract, making it difficult to understand and communicate with. It is not surprising that the three graphical tools share the same characteristics, because they are more or less interchangeable with each other (Hutchinson, 1981, Schruben and Yucesan, 1994). Other modeling tools used in FMS modeling inlcude IDEF (Banerjee and Al-Maliki, 1988) and concurrent logic programming (Dotan and Ben-Arieh, 1991).

The JR-net modeling framework is reviewed in the next section, followed by a section on JR-net structure for FMS modeling. In section 4, details of the FMS modeling method are explained, and a characterization of FMS configuration together with additional FMS modeling examples are presented in the following section. The issue of implementing a JR-net model is addressed in section 6.

2. Review of JR-net modeling framework

The JR-net modeling framework proposed in Han et al. (1995) is based on the object-oriented modeling paradigm of Rumbaugh et al. (1991). It has been developed based on the observations that modern AMSs (automated manufacturing systems) have a modular and hierarchical structure constructed from standard resources, that they can be decomposed into disjointed subsets in terms of their generic functions, and that AMS design is a sequential and iterative process. This observation is particularly true for FMSs, as the general trend in FMS design is to use the modular design concept employing standard resources (Yamazaki, 1991; Makino, 1992, Mason, 1994).

In the proposed JR-net modeling scheme, a static resource in an AMS is a standard item (or catalog item) belonging to one of the standard resource types. A set of resources may form a *resource-set* which would become a station of the AMS. A *station* is defined as a generic and disjointed subset of AMS which performs a specific function such as job processing, material handling, or storage. Each individual resource in a specified AMS is called a *resource instance.* The relationships among the static resource-type objects (i.e., station, resource-set, resource class, resource instance) are indicated in the bottom row of figure 1.

A *job* flows through a series of stations in an AMS, according to its own *process plan*, in order to be processed or handled by the static resources. It may represent a part, a component, an assembly, or a product. This job flow is controlled by *controllers*. The "processing demand" of the jobs and "processing supply" by the resources are mapped into a *job-resource relation* which becomes the basis of the job flow control. Further, the flows of *auxiliary resources*, such as pallets and tools, which support the processing or handling of jobs are determined based on the job-resource relation. The relationships among these four objects (i.e., job, process plan, controller, and job-resource relation) are shown in the upper part of figure 1.

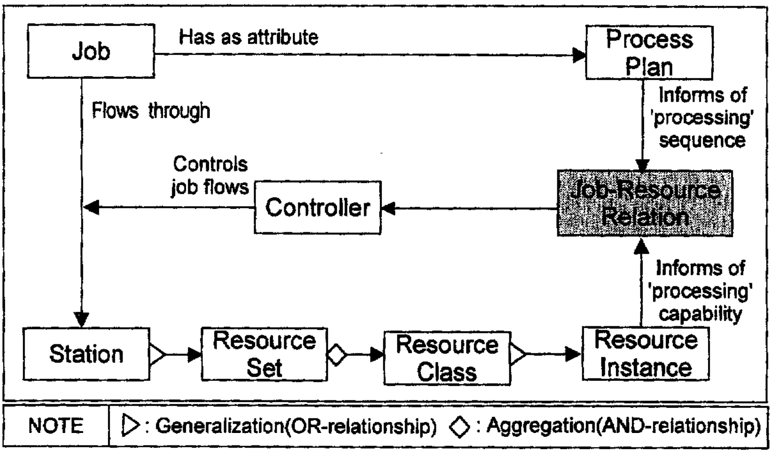


Figure 1. Relationships among AMS objects.

The JR-net modeling framework mimics the following real-life AMS design steps: based on the job processing requirements, a rough layout of the AMS is configured in terms of stations, and then a set of standard resources is selected for each station. Over this "static" layout model, job flows together with auxiliary resource flows are defined first, and then detailed design parameters such as processing cycle times are specified. With this initial functional model, the overall capacity of the system is estimated to see if the processing requirements of the jobs can be properly handled by the AMS. If the designers are satisfied with this initial design, more elaborate supervisory control structures are defined to obtain the control model of the AMS.

As with the object-oriented modeling of Rumbaugh et al. (1991), which is mainly concerned with data flow modeling, the JR-net modeling framework supports the three stages of models: static layout model (object model); job flow model (functional model); and supervisory control model (dynamic model). As depicted in figure 2, the "three-phase approach," which is basically the same as the steps being taken by FA engineers, is as follows:

(Level 1–0): decompose an AMS into its functional subsets (i.e., stations) and determine the overall layout of the AMS in terms of its stations.
(Level 1–1): determine the layout of each station by arranging the standard resources selected for the station.
(Level 2–0): specify job flow paths (transfers) among the stations.
(Level 2–1): specify job flow paths among the resources within each station.
(Level 3–0): specify supervisory control requirements between the stations.
(Level 3–1): specify control requirements between resources in each station.

3. JR-net structure for FMS modeling

As discussed earlier, the JR-net modeling framework is based on the *three-phase modeling* paradigm: static layout modeling, functional modeling, and supervisory control modeling. In this section, the structure of the JR-net at each stage of FMS modeling is presented.

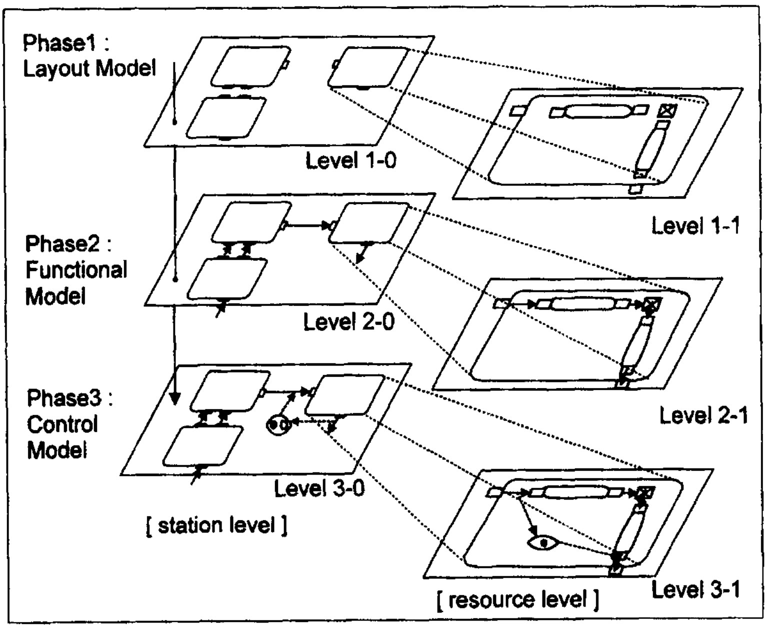


Figure 2. JR-net modeling procedure.

As the term "part" is widely used to represent a job in FMSs, "part" will be used in place of "job" hereafter. Parts in FMSs are normally held in pallets of some sort for transport and for locating on machine tables (Carrie, 1988). Two types of pallet usage are common. One type is a *common pallet* having removable fixtures that are detached from the pallet when not required. The other is a *dedicated pallet* having fixtures permanently assigned to it.

3.1. FMS layout model structure

In the layout modeling stage, individual resources of an FMS, such as machining center, AGV, and storage, are represented on the layout by using predefined *resource symbols*, and a number of stations are defined by grouping the resources according to their generic functions. As introduced in Han et al. (1995), the resources found in modern automated manufacturing systems can be grouped into eight *resource types* as follows:

- Machine: for processing parts on its own table.
- Robot: for handling or processing parts (without its own table).
- APC (automatic pallet changer): for changing parts at the machine table.
- Table: for putting a part on during processing or handling.
- Vehicle: for transporting parts among (multiple) "ports."

- Conveyor: for conveying parts from one port to another.
- Diverter: for diverting part flows in a conveyor net.
- Storage: for storing parts.

This grouping turned out to be valid also for FMSs. Summarized in figure 3 are resource symbols for the eight resource types together with individual standard FMS resources belonging to each resource type. The standard resources are treated as predefined *primitives*. The rectangles appearing in the resource symbols represent *ports* on which a part (or an auxiliary resource) may be put. A detailed description of each individual resource may be stored elsewhere in the form of *resource specification* (see Han et al., (1995)). In this modeling stage, a process plan for each part entering the FMS is also defined.

3.2. *Functional JR-net structure for FMS modeling*

As shown in figure 1, the JR-net modeling scheme places a special emphasis on the relationships between the parts and the resources. Among these relationships, the flows of parts and of auxiliary resources are described in a functional JR-net. A functional JR-net model is a directed graph where each resource becomes a node and each transfer becomes an arc of the graph. When a part (or an auxiliary resource) is to be transferred between a

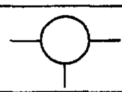
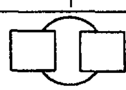
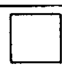
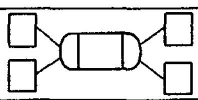
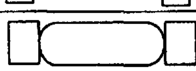
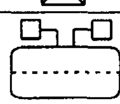
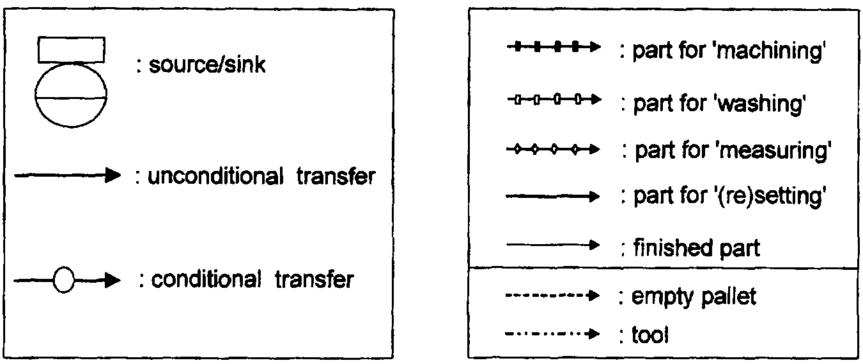
AMS Resource Type	JR-Net Resource Symbol	Standard FMS Resources
Machine		* Machining Center * Washing Machine * C M M
Robot		* Worker(Operator) * Robot(Manipulator)
APC (Automatic Pallet Changer)		* Pallet Shuttle * Carousel-type Pallet Magazine
Table		* Loading Table * Input/Output Buffer * Unload Area
Vehicle		* A G V * R G V * Stacker Crane
Conveyor		* Roller Conveyor
Diverter		* Diverter
Storage		* Storage Rack * Common Buffer

Figure 3. FMS resource symbols.

pair of adjacent resource ports, a part (or an auxiliary resource) flow is defined by using the *physical flow symbol* of figure 4(a): *source/sink*, *(unconditional) transfer*, and *conditional transfer*. The start and end of a part flow are indicated by the source/sink symbols. Part flows are indicated by solid arrows of different types, and auxiliary resource flows are represented by dashed arrows as shown in figure 4(b). If a transfer needs certain conditions to be fulfilled, it is represented by a conditional transfer symbol. In a full scale FMS, there could be seven or more transfer types. Among the seven transfer types shown in figure 4(b), five are for parts, one is for empty pallets, and one is for tools.

3.3. Control JR-net structure for FMS modeling

The third level model, control model, is defined on top of the functional JR-net model by specifying its supervisory control requirements. For this purpose, *control flow symbols* are introduced, as shown in figure 5. The meaning of the five types of control flow symbol are as follows:



(a) (b)

Figure 4. Physical flow symbols for FMS modeling.

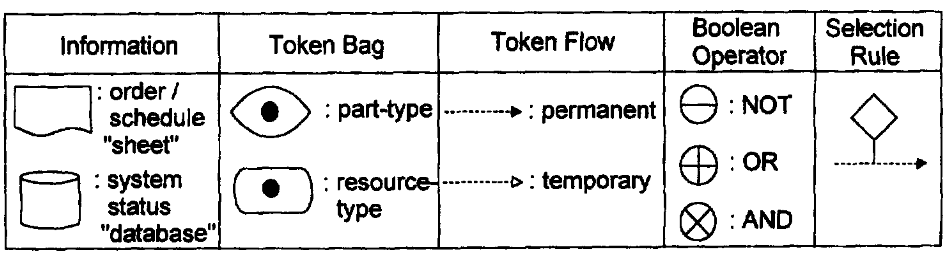


Figure 5. Control flow symbols for FMS modeling.

1. Information storage: *order/schedule "sheet"* is used to represent orders or production schedules, while the *system status "database"* is used to represent the status of the system or subsystems.
2. *Token "bag"* is used to store colored tokens representing the available capacity of a static resource (resource-type token bag) or the availabilities of parts or auxiliary resources (part-type token bag).
3. *Token flow* is represented by a dashed arrow: a *permanent flow* is denoted by a solid arrow head, and a *temporary* flow by a hollow arrow head.
4. *Boolean operator symbols* for *NOT, OR, AND* are used to represent a compound condition for a conditional transfer. There may be priority rules associated with the OR operator.
5. *Selection operator* (diamond shape) is used to denote *selection rules* such as random, FIFO, cyclic, max/min available capacity, balanced, etc.

The existence of a token in a token bag represents a predicate condition, while a token flow arrow is regarded as a Boolean variable. The control flow symbols defined in figure 5 turned out to be enough for describing most of the supervisory control logic or real-world FMSs.

It is also necessary to express the supervisory control logic in an algebraic form. For this purpose, the following notations are introduced:

1. *Boolean operators:* $\vee$ *(or),* $\wedge$ *(and),* $\neg$ *(not);*
2. *Token flows:* $\rightarrow$ (permanent), $\leftrightarrow$ *(temporary);*
3. *Priority associated with "*$\vee$*" is denoted by superscripts;*
4. *Selection rule is enclosed by a pair of angle brackets "*$\langle\ \rangle$*".*

Shown in figure 6(a) is a precondition for the conditional transfer T1 (from the port PO1 to PO2). The transfer T1 is *fired* if *"the resource R1 is available and there is an order for the part of type P1 (a part is selected on a FIFO basis)."* Thus, the precondition shown in figure 6(a) is expressed as

$$Pre\text{-}Cond(T1) \equiv (P1\langle \text{FIFO} \rangle\rightarrow) \wedge (R1\rightarrow).$$

In the above expression, the arrow "$\rightarrow$" denotes a permanent token flow.

Shown in figure 6(b) is a precondition for the conditional transfer T2. The transfer T2 is fired *"if a part of type P2 or P3 is available (P2 has priority over P3): when P2 is selected, its token is permanently removed from the token bag; when P3 is selected, the token is held during the firing of T2 and then it is returned to its token bag (temporary token flow)."* The precondition shown in figure 6(b) is expressed as

$$Pre\text{-}Cond(T2) \equiv (P2\rightarrow)^1 \vee (P3\leftrightarrow)^2.$$

4. JR-net modeling of an FMS

In this section, the proposed method for constructing a graphical model of an FMS is explained using a Mazatrol FMS (Morito et al., 1991, 1992) as an example. Mazatrol is

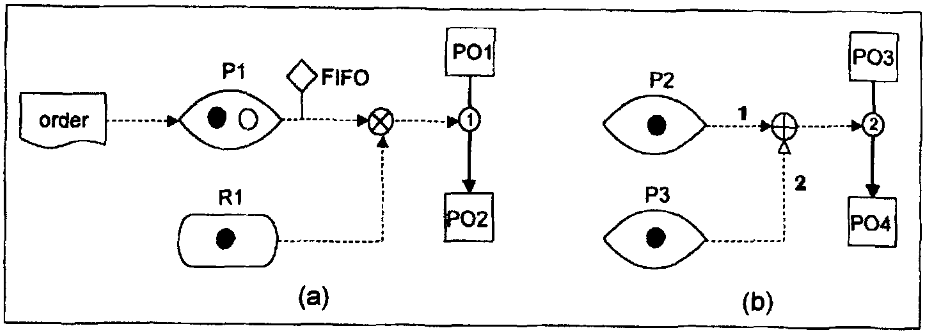


Figure 6. Compound precondition examples.

known to be a popular FMS (more than 300 systems installed worlwide). The layout of the Mazatrol FMS is shown in figure 7. In actual installations, a tool-handling system may be added as an option, but this feature is not treated here (a JR-net model of an FMS with tool-handling system will be presented in section 5.2).

JR-net models of the FMS are constructed using the primitive symbols defined in figures 3, 4, and 5, and by following the "JR-net modeling procedure" of figure 2.

4.1. Layout modeling

As shown in figure 7, the FMS has: eight identical machining centers, with each machining center having two pallet stands, one as an input buffer and the other as an output buffer; four loading tables, with an unload area attached to each loading table; a part storage having 120 storage racks; and a stacker crane. Since there are four distinctive functions associated with the FMS, it can be decomposed into four stations: machining center station, load/unload station, part storage station, and part transport station. In JR-net modeling, a single resource is also treated as a station as long as it is responsible for a generic function of an FMS. By using the FMS resource symbols of figure 3, the layout model of the FMS is easily obtained, as shown in figure 8.

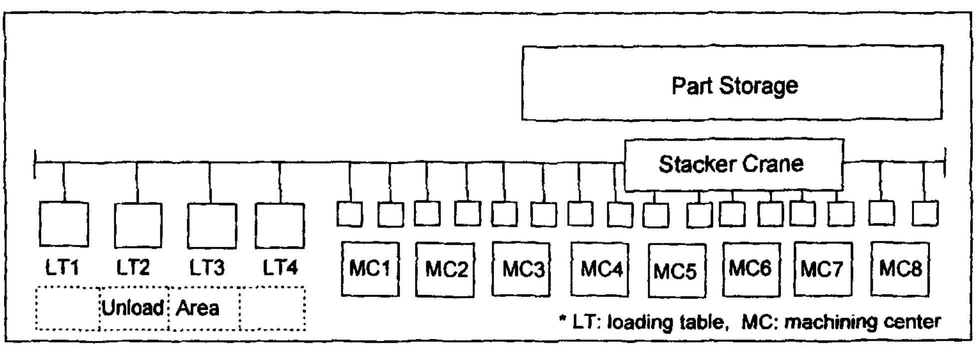


Figure 7. Layout of Mazatrol FMS.

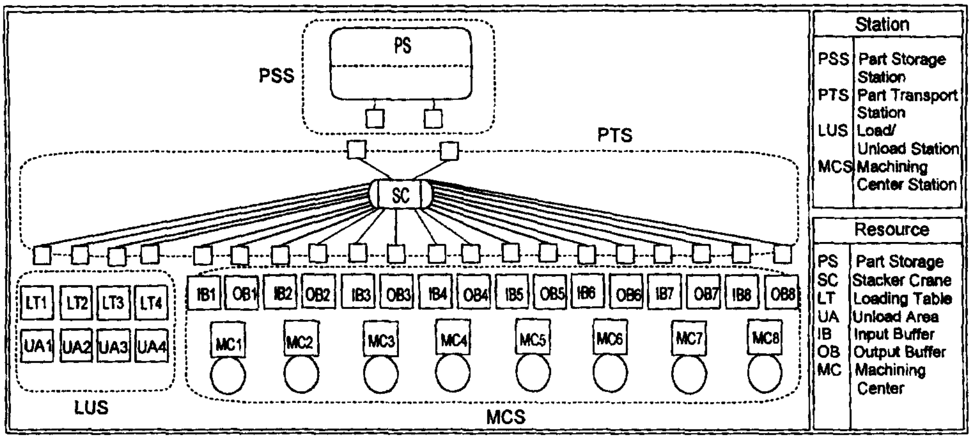


Figure 8. Layout model of Mazatrol FMS.

4.2. Functional JR-net modeling

In the Mazatrol FMS, dedicated pallets with permanently assigned fixtures of different types are used for different part types and for different processing stages of the same part. Therefore, every time a new part is introduced, an empty pallet with fixtures of matching type may have to be brought to the loading table from the part storage. A "semifinished" part is sent back to the loading table for resetting, which needs a new empty pallet to be brought from the part storage and leaves an empty pallet behind to be returned to the part storage (unless a part of matching type is waiting for the empty pallet at the unload area). As a result, frequent movements of empty pallets are made between the loading table and the part storage.

In the Mazatrol FMS, every incoming part is sent to the part storage on its way to a machining center. Similarly, every finished part is temporarily stored in the part storage before leaving the system. This strategy is widely used in other FMSs as well, because of its simplicity in control. The other strategy is to store the part in the part storage only when the target location is blocked. (This case will be dealt with in the next section.) A typical part in the FMS goes through the following sequence of "*primary transfers*":

T1 ≡ *INTRODUCE (for setting):* put the part in an unloading area (UA).
T2 ≡ *LOAD-UP (for setting):* transfer from UA to a loading table (LT).
T3 ≡ *LT-to-PS (for machining):* move from LT to the part storage (PS).
T4 ≡ *PS-to-MC (for machining):* move from PS to a machining center (MC).
T5 ≡ *MC-to-PS (for resetting):* move from MC to PS (semifinished part).
T6 ≡ *PS-to-LT (for resetting):* move from PS to LT (semifinished part).
T7 ≡ *MC-to-PS (for release):* move from MC to PS (finished part).
T8 ≡ *PS-to-LT (for release):* move from PS to LT (finished part).

In order to support these part flows, the required movements of the empty pallets are:

T9 ≡ *P-RETRIEVE (pallet retrieval):* move from PS to LT.
T10 ≡ *P-STORE (pallet storage):* move from LT to PS.

Since a part changes its status as it passes through the system, it is necessary to group the part flows into three types:

- parts waiting for machining
- parts waiting for setting or resetting, and
- parts waiting for release (finished parts).

Thus, there are four types of transfer in the system, three for parts and one for empty pallets.

The functional JR-net model of the FMS representing the flows of parts and of empty pallets is shown in figure 9. There are 24 transfers in the functional JR-net model, denoted by the numbers on the transfer arcs in the figure. The first ten transfers (T1 to T10) correspond to the primary transfers. In fact, these primary transfers, usually become the conditional transfers in a control JR-net model.

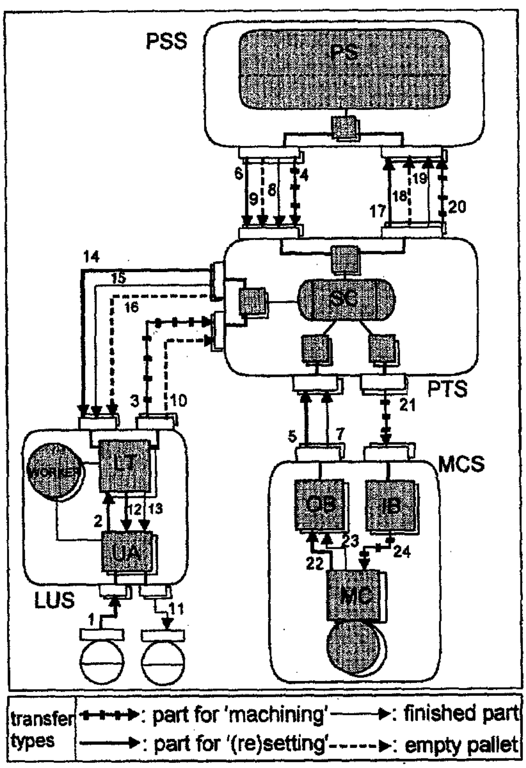


Figure 9. Functional JR-net model of Mazatrol FMS.

A functional JR-net model is basically a part flow model similar to the data flow diagram of Rumbaugh et al. (1991). A part flows through the FMS either by being "transported" (by a vehicle) or by being "transferred" between adjacent resources. It should be noted that only transfers are explicitly modeled in the functional JR-net, because transport is an attribute (i.e., a *method* in object-oriented programming) of the resource. The attributes of a resource (i.e., speed, pickup/deposit time, dispatching rules, etc.) are described in the form of *resource specifications* during the layout modeling stage.

The functional JR-net model is constructed from the layout model by connecting the resource ports along the path of each transfer. The procedure for constructing a functional JR-net is as follows:

1. Encapsulate each station with ports attached, while aggregating "multiple identical resources" by using *duplicate symbols.*
2. Define inter-station part flows by joining the stations' ports.
3. Define intra-station part flows and connect the resources' ports to the station's ports.
4. Define auxiliary resource flows to support the part flows defined in 2) and 3) above.

4.3. Control JR-net modeling

In the JR-net modeling of an FMS, conditions for transfers are explicitly expressed in the control model. Thus, the first step in control modeling is to identify all the conditional transfers that require certain conditions to be fulfilled. In the functional JR-net model of figure 9, the first ten transfers (T1 to T10), which have a one-to-one relationship with the primary transfers, turned out to be conditional transfers.

The transfer T1 is an *INTRODUCE* operation. The condition for a new part to be introduced into the unload area is: "*The part is on the loading schedule list*" and "*a matching pallet is on the loading table* or *is available in the part storage.*" When empty matching pallets are available both on the loading table and in the part storage, the one on the loading table is used. Thus, the precondition for T1 may be expressed as follows:

$$\textit{Pre-cond}(T1) \equiv (LS\rightarrow) \wedge ((POLT\leftrightarrow)^1 \vee (PIPS\rightarrow)^2) \quad (1)$$

where LS = loading schedule ready,
POLT = available pallet on loading table,
PIPS = available pallet in part storage.

In (1), a temporary token flow is applied to the POLT token (namely, the token is returned right after T1 is executed), because this token is checked again by the next transfer T2 (*LOAD-UP* operation).

When *T1* is *fired* with the PIPS condition (namely, a matching pallet is in the part storage but not on the loading table), a request for the empty pallet is issued by putting an *empty pallet request token* into the token bag. This situation is modeled as a post-condition of T1. Thus, the post-condition for T1 may be expressed as

$$Post\text{-}cond(T1) \equiv (\neg POLT) \Rightarrow (PR\rightarrow) \quad (2)$$

where PR = empty pallet request token,
¬POLT = no empty pallet on loading table,
⇒: "implies."

Shown in Figure 10 is a control JR-net model of the FMS. Conditions for the conditional transfers (denoted by a small circle with an id number in the middle of the transfer arrow) are modeled by using the control flow symbols. Compare the pre-conditions and post-conditions of T1 in figure 10 (bottom left) with (1) and (2).

Among the seven token bags shown in figure 10, the four bags in "American football" shape are part-type token bags (**a**, **b**, **c**, **d**), and the three in "football field" shape are resource-type token bags (**e**, **f**, **g**).

a : holds tokens for "loading schedule ready."
b : holds tokens for "empty pallet in part storage."
c : holds tokens for "empty pallet retrieval request."
d : holds tokens for "empty pallet on loading table."

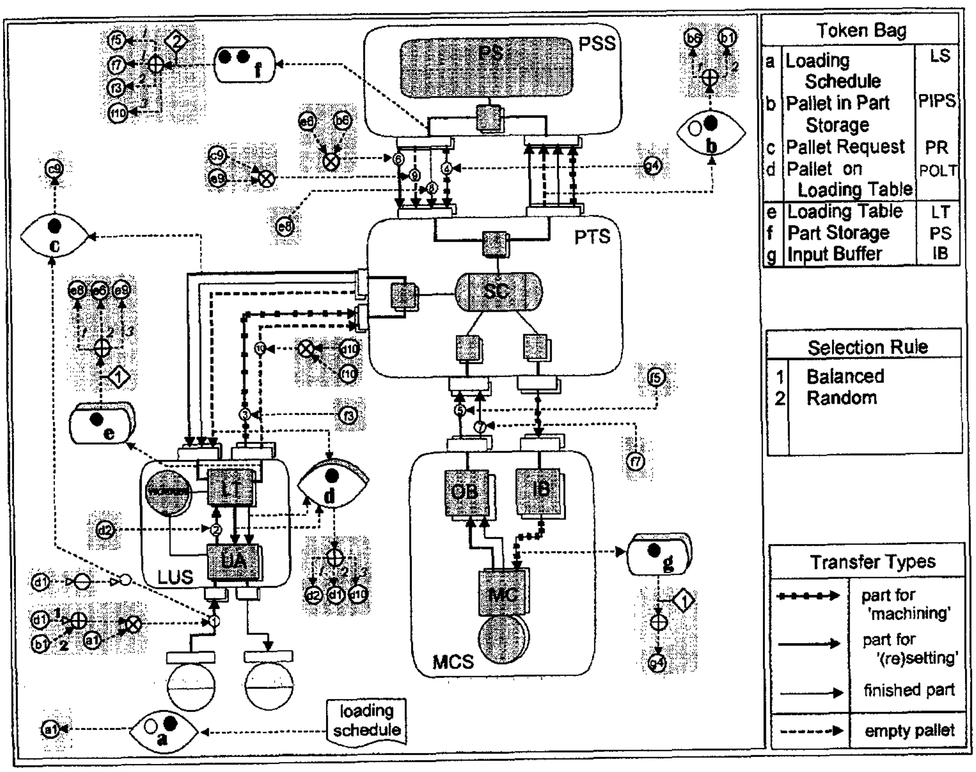


Figure 10. Control JR-net model of Mazatrol FMS.

e : holds tokens for available capacity of "loading table."
f : holds tokens for available capacity of "part storage."
g : holds tokens for available capacity of "input buffer."

As a convention, each *token flow symbol* is given an id by concatenating its token bag id and transfer id. For example, the token flow from the token bag **a** to the transfer Tl has an id of "*al*" written inside a small circle. Precondition expressions for a few more transfers are given below:

Pre-cond(T2) ≡ *(POLT →):* load up if there is an empty *p*allet *o*n the *l*oading *t*able.
Pre-cond(T3) ≡ *(PS⟨*RAN*⟩→):* transfer to the stacker crane if the *p*allet *s*torage has room (a place is selected at *ran*dom).
Pre-cond(T6) ≡ *(LT⟨*BAL*⟩→)* ∧ *(PIPS→):* retrieve a part for resetting if a *l*oading *t*able is available *and* there is an usable empty *p*allet *i*n the *p*art *s*torage (one of the loading tables is chosen so that their utilizations be *bal*anced).

Two *selection rules*, random selection and balanced selection, are used in the above preconditions for the selection of "multiple identical" resources.

Tokens in a token bag may be consumed by a number of conditional transfers and are produced by some other transfers. For example, the *POLT* (pallet on loading table) tokens in the token bag "**d**" are generated by the three transfers T12, T13, T16 (the id numbers are shown in figure 9), and they are consumed by the three conditional transfers Tl, T2, T10. If more than one transfer is competing for the tokens, one is selected based on a *release priority.* In this case, T2 (LOAD-UP) has the highest priority and then Tl (*INTRODUCE*). T10 (P-STORE) has the lowest priority. This release priority for the tokens in the token bag **d** may be expressed as follows:

$$Release\text{-}priority(POLT) \equiv \{ (T2)^1 \vee (T1)^2 \vee (T10)^3 \}$$

Refer to figure 10 (bottom middle portion) to see how this situation is modeled in the control JR-net model.

5. FMS configurations and more examples of JR-net

In this section, a scheme for characterizing an FMS is proposed, and then a few examples of JR-net models of FMSs are presented.

5.1. FMS configurations

Characterization of FMS configurations can be useful in synthesizing a new FMS or in evaluating alternative designs. We propose an FMS characterization scheme based on the generic functions associated with an FMS. We restrict our discussions to the *machining-type* FMSs. As listed in figure 11, there are five functions in an FMS: part preparation, transport, storage, machining, and tool handling. Each function may be handled by one or more stations.

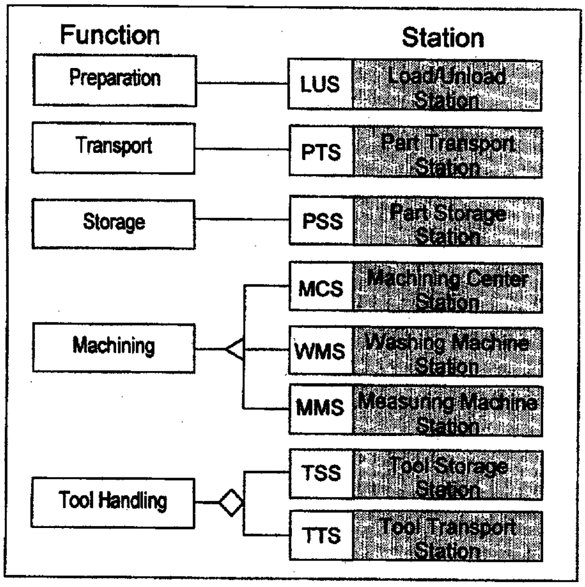


Figure 11. Functions and stations of FMSs.

Based on possible design alternatives for each of the functions, one may define different configurations of an FMS as follows (PTS: part transport station; MCS: machining center station; PSS: part storage station):

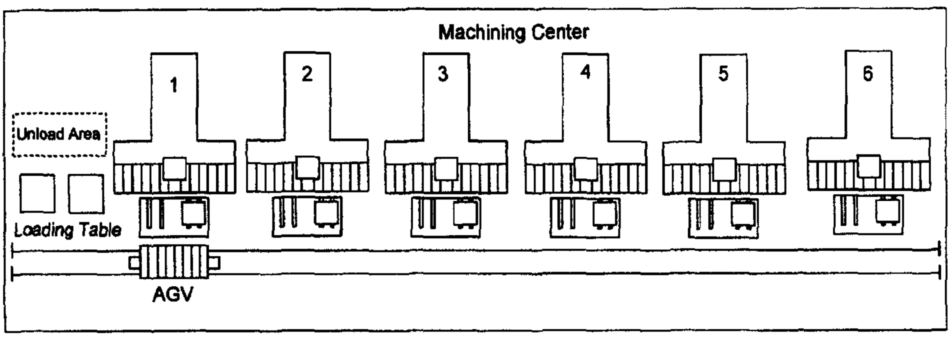
1. Part preparation (PP):
   PP = 1 : usage of common pallet with removable fixtures.
   PP = 2 : usage of dedicated pallet with permanently assigned fixtures
2. Part transport (PT):
   PT = 1 : single-stage transport (one PTS handles all transportations).
   PT = 2 : two-stage transport (MCS → PTS_1 → storage → PTS_2 → PSS).
3. Part storage (PS):
   PS = 0 : no separate part storage station.
   PS = 1 : mandatory storage (every in/out part is sent to the PSS)
   PS = 2 : selective storage (sent to PSS when the target location is blocked).
4. Part machining (PM)
   PM = 1 : machining only.
   PM = 2 : additional processing (washing, measuring, etc.)
5. Tool handling (TH)
   TH = 0 : off-line handling (no automated tool handling system)
   TH = 1 : in-line handling (integrated tool handling system)

There could be 48 (2×2×3×2×2) configurations, but only 40 combinations are feasible, because PS = 0 and PT = 2 are not compatible.

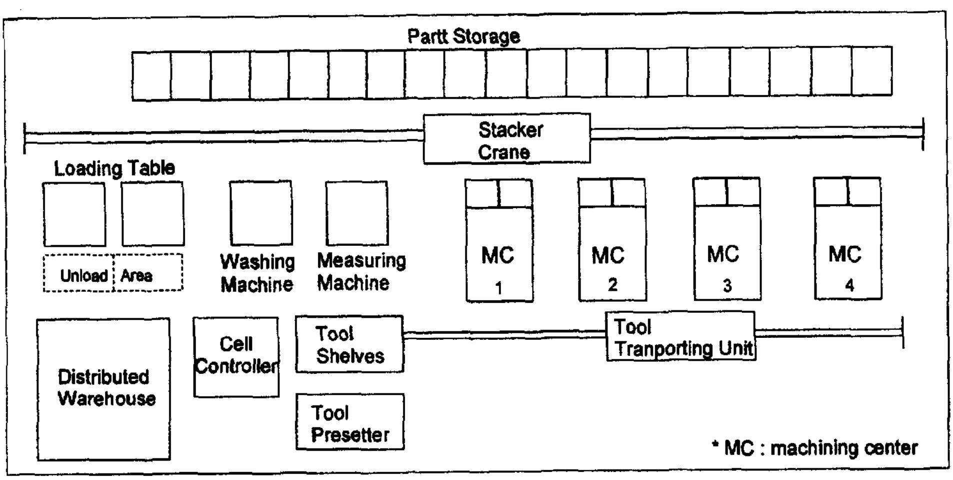
5.2. More examples of JR-net models of FMSs

Presented in this section are control JR-net models of two more FMSs documented in the literature. The first one, shown in figure 12(a), is the Anderson Strathclyde FMS introduced in Carrie (1988). The FMS has six identical machining centers served by a linear type AGV system. Each machining center has two pallet stands, one as an input buffer and the other as an output buffer. The load/unload station consists of two loading tables and an unload area where parts are put into fixtures.

Another example, shown in figure 12(b), is called the Fanuc Cell 60 System (Lee et al., 1993). The FMS consists of four machining centers, one measuring machine, one washing machine, one part storage, two loading tables together with an unload area, and a stacker crane serving all the part transport requirements. In addition, the FMS has an in-line tool handling system. Parts are loaded on dedicated pallets with permanently assigned fixtures (as in the Mazatrol FMS), but they are sent to the part storage only when the target location is blocked.



(a) Anderson Strathclyde FMS



(b) Fanuc FMS

Figure 12. Layout of example FMSs.

Under the FMS characterization scheme of section 5.1, the three FMSs treated in this article have the following configurations:

1. The Mazatrol FMS: { *PP = 2, PT = 1, PS = 1, PM = 1, TH = 0* };
2. The Anderson Strathclyde FMS: { *PP = 1, PT = 1, PS = 0, PM = 1, TH = 0* };
3. The Fanuc FMS: { *PP = 2, PT = 1, PS = 2, PM = 2, TH = 1* },

where PP is part preparation, PT is part transport, PS is part storage, PM is part machining, and TH is tool handling.

The operational characteristics of the Anderson Strathclyde FMS are described in figure 13 in the form of an ACD (activity cycle diagram). A newly arrived part waits in the unload

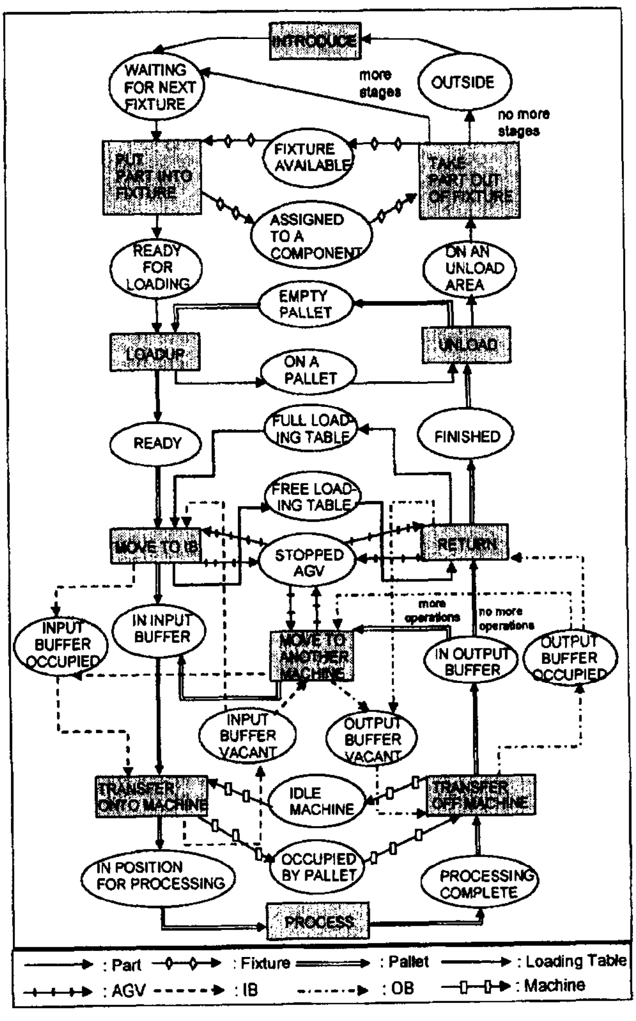


Figure 13. Activity Cycle Diagram for Anderson Strathclyde FMS.

area. If a fixture for the part is available, the part is put into a fixture at the unload area, and then the fixtured part is loaded up on the empty pallet of a loading table (there are enough pallets in the system). Then the loaded part is moved to the input buffer of a machining center by the AGV. One interesting feature of this FMS, which is different from the Mazatrol FMS, is that a semifinished part may move to a next machining center for further processing without resetting. Thus, a part goes through the following sequence of primary transfers:

T1 ≡ *INTRODUCE (for setting):* put the part in the unload area(UA).
T2 ≡ *LOAD-UP (for setting):* transfer from UA to a loading table (LT).
T3 ≡ *LT-to-MC (for machining):* move from LT to a machining center (MC).
T4 ≡ *MC-to-MC (for machining):* move to the next MC (semifinished part).
T5 ≡ *MC-to-LT (for resetting):* move from MC to LT (semfinished part).
T6 ≡ *MC-to-LT (for release):* move from MC to LT (finished part).

A control JR-net of the Anderson Strathclyde FMS corresponding to the ACD model is given in figure 14. By following the conventions in section 4.3, the precondition for the transfer T2 (LOAD-UP), for example, is expressed as

Pre-cond(T2) ≡ *(FOUA→)* ∧ *(POLT→)*
where FOUA = available Fixture On Unload Area,
POLT = available Pallet On Loading Table.

The meaning of the control JR-net model should be apparent.

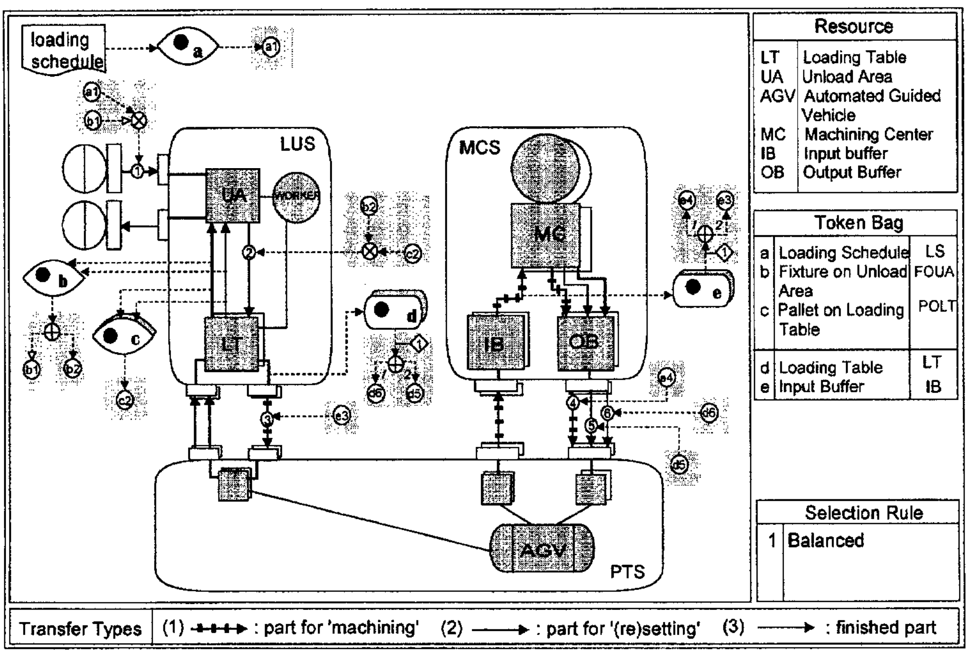


Figure 14. Control JR-net for the ACD model.

However, the ACD model of figure 13 is an oversimplification of the system operation (so is the control JR-net model of figure 14). Because the FMS does not have a separate storage for the empty pallets, they will certainly block the loading tables. An outgoing part (for release or resetting) may be blocked by an empty pallet on the loading table. Thus, the empty pallet should be moved from the loading table to a pallet stand in front of a machining center before sending a finished (or semifinished) part to the loading table, which again creates another blocking problem at the machining center. The authors failed to describe the situation with an ACD, but the control JR-net m odel shown in figure 15 gives a complete description of the behavior of the Anderson Strathclyde FMS, including the handling of the blocking problem.

For the modeling of the blocking avoidance feature, the transfer condition for T2 should be modified as

Pre-cond(T2) $\equiv$ *(FOUA→)* $\wedge$ *(POLT→)* $\wedge$ *(POUR→)* $\wedge$ *(IB⟨*BAL*⟩→)*
where FOUA = available Fixture On Unload Area,
POLT = available Pallet On Loading Table,
POUR = Part Ready On Unload Area for (Re)setting,
IB = Input Buffer available.

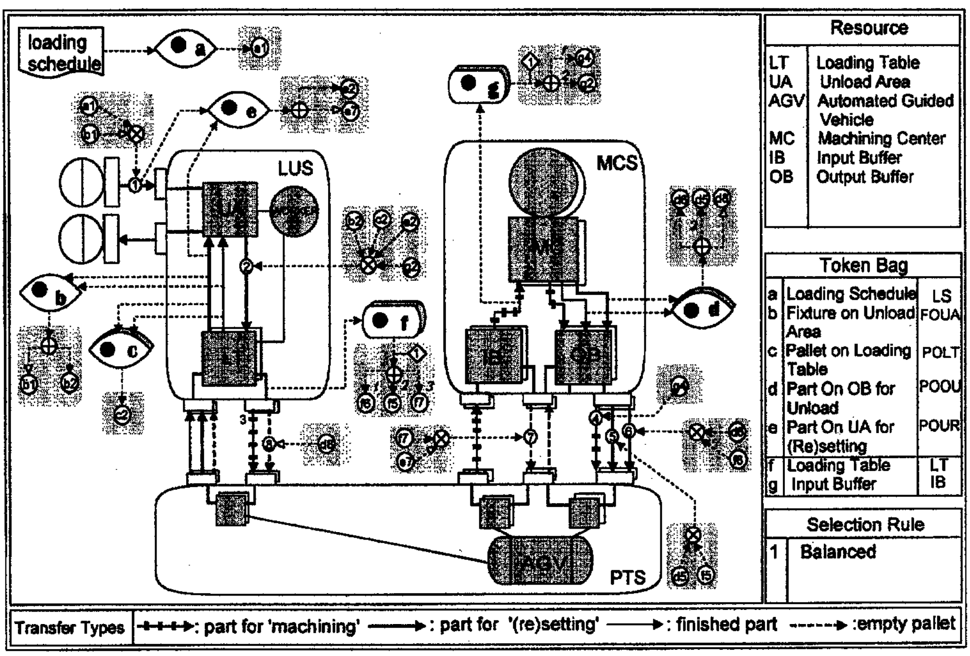


Figure 15. Control JR-net for Anderson Srathclyde FMS.

And the following two transfers of empty pallets are added to the primary transfers:

T7 ≡ *PALLET-RETRIEVE:* move from a pallet stand to a loading table.
T8 ≡ *PALLET-STORE:* move from a loading table to a pallet stand.

The preconditions for T7 and T8 are expressed as

$$Pre\text{-}cond(T7) \equiv (LT\langle \text{BAL} \rangle \rightarrow) \wedge (POUR \leftrightarrow),$$
$$Pre\text{-}cond(T8) \equiv (POOU \rightarrow)$$

where LT = Loading Table available,
POUR = Part Ready On Unload are for (Re)setting,
POOU = Part ready On Output buffer for Unload.

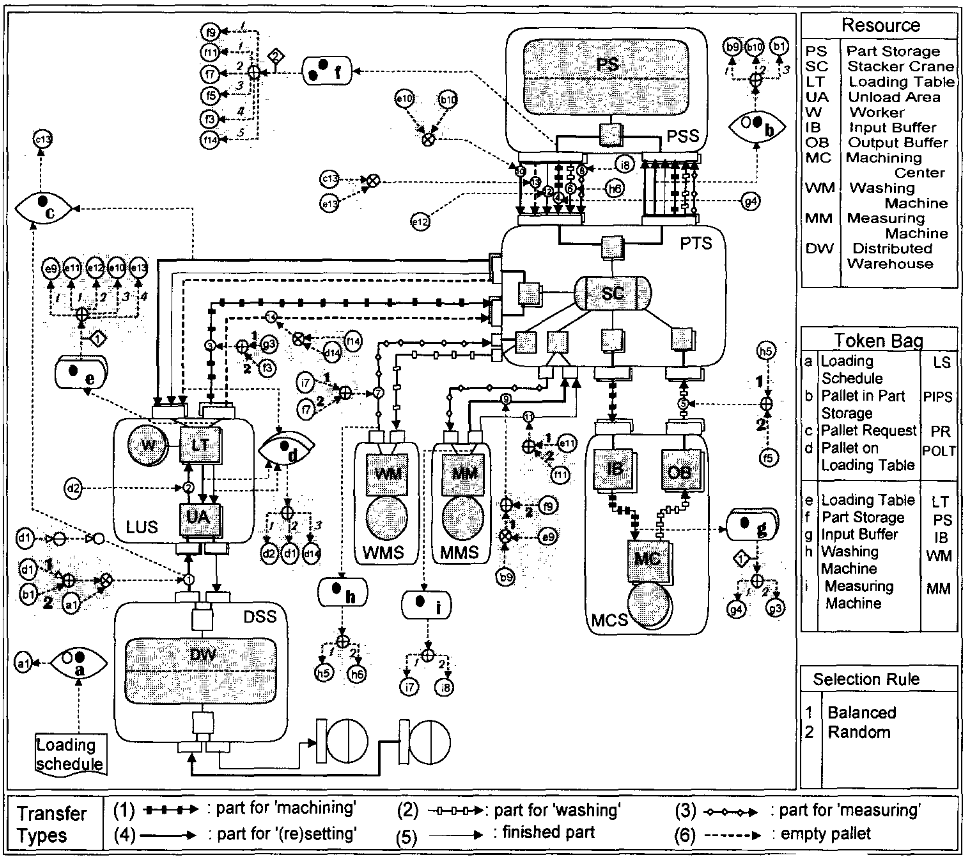
A control JR-net model (for part flow) of the Fanuc FMS is shown in figure 16(a). The part flow of the Fanuc FMS is identical to that of the Mazatrol system, except for the following points: after being machined a part has to go to the washing machine and then to the measuring machine; a part is moved to the part storage only when the destination location is blocked (the part is then moved to its destination when it becomes available). There are six types of transfers, five for parts and one for empty pallets. The five transfer types for parts are:

- parts waiting for machining,
- parts waiting for washing,
- parts waiting from measuring,
- parts waiting for setting or resetting, and
- parts waiting for release (finished parts).

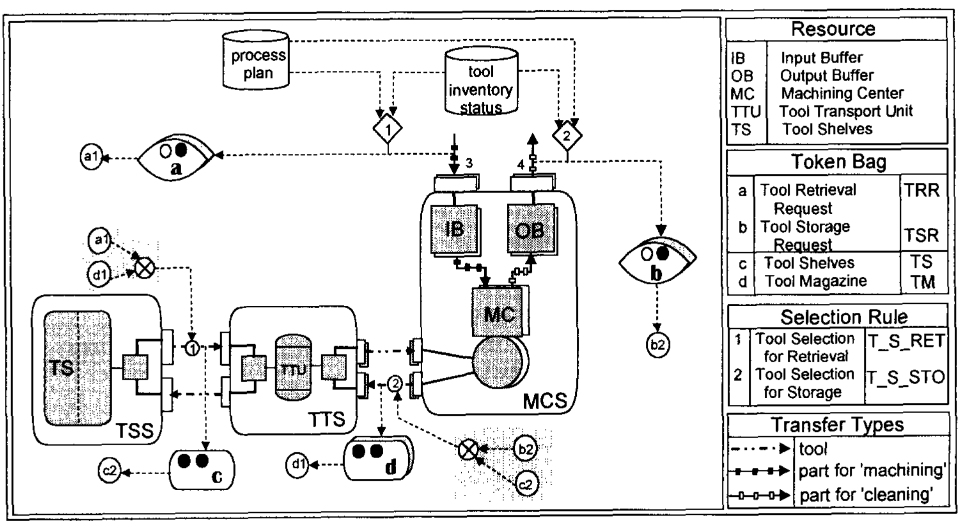
There are 14 primary transfers (T1 to T14), each of whcih needs a precondition for its execution, and there are nine token bags in the JR-net. The pre-condition and post-condition for the transfer T1 are the same as those of the Mazatrol FMS. Refer to the (1) and (2) in section 4.3. Precondition expressions for a few more transfers are given below:

$$Pre\text{-}cond(T3) \equiv (IB\langle \text{BAL} \rangle \rightarrow)^1 \vee (PS\langle \text{RAN} \rangle \rightarrow)^2,$$
$$Pre\text{-}cond(T5) \equiv (WM \rightarrow)^1 \vee (PS\langle \text{RAN} \rangle \rightarrow)^2,$$
$$Pre\text{-}cond(T10) \equiv (PIPS \rightarrow) \wedge (LT\langle \text{BAL} \rangle \rightarrow)$$

where T3 = move from a loading table to an input buffer,
T5 = move from an output buffer to the washing machine,
T10 = move from the part storage to a loading table for resetting,
IB = input buffer available,
PS = part storage available,
WM = washing machine available,
PIPS = empty pallet in part storage,
LT = loading table available.



(a) Part flow



(b) Tool flow

Figure 16. Control JR-net of Fanuc FMS.

The tool-handling system of the Fanuc FMS is in fact an AS/RS having a two-stage storage system: the main storage of the tool shelves and the secondary storage of the tool magazine at the machining center. As shown in figure 16(b), a part transfer from the stacker crane to the input buffer of a machinng center issues a tool retrieval request, and the transfer of a machined part from the output buffer of the machining center to the stacker crane generates a tool storage request. It is assumed that the required tools for a machining operation are stored in the *process plan database* and that necessary data concerning available tools are stored in the *tool inventory status database.* Thus, after the execution of the tool selection for retrieval (No. 1 selection rule in (figure 16(b)), only the tools that are not available at the tool magazine are requested by sending *tool retrieval request tokens* to the token bag **a**.

There are two primary transfers: T1 (*TOOL-RETRIEVE*) and T2 (*TOOL-STORE*). The precondition for T1 is "there is a tool retrieval request <u>and</u> the tool magazine has room," which is expressed as

$$\begin{aligned} \textit{Pre-cond(T1)} &\equiv (\textit{TRR}\rightarrow) \wedge (\textit{TM}\rightarrow) \\ \textit{where TRR} &= \textit{tool retrieval request,} \\ \textit{TM} &= \textit{tool magazine available.} \end{aligned}$$

The precondition for T2, similar to T1, is expressed as

$$\begin{aligned} \textit{Pre-cond(T2)} &\equiv (\textit{TSR}\rightarrow) \wedge (\textit{TS}\rightarrow) \\ \textit{where TSR} &= \textit{tool storage request,} \\ \textit{TM} &= \textit{tool shelves available.} \end{aligned}$$

6. JR-net implementation issues

In general, required properties of an FMS modeling tool are: ease of model building, ease of communication, high modeling power, ease of implementation, and analysis power. The FMS modeling scheme proposed in the article is designed with the first three requirements in mind. Compared to other graphical modeling tools, it is more convenient to construct and easier to understand. Obviously, it has a very high modeling power.

In this section, the issue of implementing the JR-net is addressed. There may be three approaches to implementing the JR-net model: converting it into other formal models such as Petri nets, building a JR-net based simulator, and using it as a preprocessor for a commercial simulation package. Yet another possibility is to use the JR-net model as a *reference model* in developing a simulation program of the FMS.

There has been an attempt to develop an automatic conversion scheme for obtaining Petri nets from the JR-net model (Park, 1992), and the authors are still working on the subject with some promising results. The basic idea of this approach is as follows: The standard resource primitives are represented as *macro-place* (sub-Petri net) nodes, and the transfers in the functional JR-net model are converted into *transition* nodes. Thus, an initial Petri net can be constructed from the functional JR-net model by connecting the place nodes and the transition nodes. For dynamic control of the conditional transfers, the token bags in the control JR-net model are converted into *control places* which are then added to the initial Petri net model in order to control the firing conditions.

The authors (and their colleagues) are also currently working on developing an FMS simulator in which the JR-net model is internally converted into an object-oriented DEVS (discrete event system specification) model (Zeigler 1990). As the DEVS formalism provides a modeling framework in which a system can be decomposed into modules hierarchically (Kim, 1994), it is compatible with the JR-net modeling scheme. In this approach, the standard resource primitives are represented as *DEVS models*, while the transfers in the functional JR-net model are represented as *coupling* between DEVS models. Finally, supervisory control requirements of FMSs are embedded in a *controller DEVS model.*

It was demonstrated in Han et al. (1995) that the JR-net model can be utilized as a graphical modeling tool for commercial simulation packages. For this purpose, the conversion of the JR-net model to an AutoMod simulation package (Norman, 1992; AutoSimulations Inc., 1993) has been applied to the case of AS/RS. While the AutoMod does not support the object-oriented paradigm, its modeling view is quite compatible with the AMS design practices. The conversion procedure of an AutoMod program generator is as follows: an AutoMod layout is interactively constructed from the JR-net layout model and the *resource specification*; the functional JR-net model is converted into a directed graph structure (called a *resource graph*); for dynamic control of the conditional transfers, AutoMod "functions" are prepared from the JR-net control model; for each arc of the resource graph, an AutoMod "process" is generated by traversing the resource graph, while calling the prepared "functions."

7. Conclusions and discussion

An object-oriented modeling scheme for obtaining a graphical model of an FMS is presented. Distinctive features of the proposed JR-net modeling scheme may be characterized as follows:

1. It employs a *three-phase modeling* paradigm similar to the actual FMS design process (making it easier for the FA engineers to build the JR-net model).
2. There is a one-to-one mapping relationship between actual FMS components and the JR-net objects (making it easier to understand and verify, leading to enhanced communication).
3. It has an object-oriented structure (making it easier to implement with an object-oriented language).

The following are a list of possible research directions related to the proposed JR-net modeling scheme:

1. Validation of the proposed modeling method by applying it to highly sophisticated FMSs.
2. Formalization of the JR-net structure in terms of its semantics and syntax, and developing a mapping scheme for converting the JR-net model into other formal models such as Petri net and DEVS.
3. Development of a JR-net based simulation package that can be used as a prototyping tool for FMS design.

Acknowledgments

This research was supported by Daewoo Information Systems Co., Ltd. and the Korean Government.

References

AutoSimulations Inc., *AutoMod User's Manual*, Bountiful, UT (1993).

Banerjee, S.K. and Al-Maliki, I., "A Structured Approach to FMS Modeling," *International Journal of Computer Integrated Manufacturing*, Vol. 1, No. 2, pp. 77–88 (April 1988).

Carrie, A., *Simulation of Manufacturing Systems*, John Wiley & Sons, Chichester, UK (1988).

Cossins, R. and Ferreira, P., "Celerritas: A Colored Petri Net Approach to Simulation and Control of Flexible Manufacturing Systems," *International Journal of Production Research*, Vol. 30, No. 8, pp. 1925–1956 (1992).

Dotan, Y. and Ben-Arieh, D., "Modeling Flexible Manufacturing Systems: The Concurrent Logic Programming Approach," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 1, pp. 135–148 (February 1991).

Han, K.H., Park, T.Y., and Choi, B.K. "An Object-Oriented Modeling Framework for an Automated Manufacturing System," Technical Report CAM-95-01, Department of Industrial Engineering, KAIST, Korea (1995).

Hutchinson, G.K., "The Automation of Simulation," *Proceedings of the 1981 Winter Simulation Conference*, Atlanta, GA, pp. 489–495 (December 1981).

Jeng, M.D., "Modular Synthesis of Petri Nets for Modeling Flexible Manufacturing Systems," *The International Journal of Flexible Manufacturing Systems*, Vol. 7, No. 3, pp. 287-310 (July 1995).

Kim, T.G., *DEVSIM++ User's Manual*, Department of Electrical Engineering, KAIST, Korea (1994).

Lee, B.J., Ono, Y., and Lee, Y., "Introduction to FANUC Cell 60 and Its Application on MAP Based Large Manufacturing FA System for 72 Hour Continuous Unmanned Operation: A Case Study," *Computers and Industrial Engineering*, Vol. 24, No. 4, pp. 593-605 (1993).

Liu, C.-M., and Wu, F.-C., "Using Petri Nets to Solve FMS Problems," *International Journal of Computer Integrated Manufacturing*, Vol. 6, No. 3, pp. 175–185 (May 1993).

Makino Milling Machine Co., Ltd., *FMS Guide*, Tokyo, Japan (1992).

Mason, F., "Off-the Shelf Machining Cells," *Manufacturing Engineering*, pp. 37–42 (October 1994).

Morito, S., Takano, T., Mizukawa, H., and Mizoguchi, K., "Design and Analysis of Flexible Manufacturing System with Simulation," *Proceedings of the 1991 Winter Simulation Conference*, Phoenix, AZ, pp. 294–301 (1991).

Moritao, S., Mizukawa, H., Mizoguchi, K., and Awane, H., "Using Discrete Event Simulation for Design and Operation of Flexible Manufacturing Systems," in *Human Aspects in Computer Integrated Manufacturing*, in G.J. Olling, and F. Kimura, Elsevier Science Publishers, Amsterdam, pp. 805-816 (1992).

Norman, V.B., "AutoMod," *Proceedings of the 1992 Winter Simulation Conference*, Arlington, VA, pp. 328–334 (December 1992).

Park, T.Y., "Modular Approach to Petri Net Modeling of Automated Manufacturing System," Doctoral dissertation, Department of Industrial Engineering, KAIST, Korea (1992) (in Korean).

Rumbaugh, J., Blaha, M., Premerlani, W., and Losenson, W., *Object-Oriented Modeling and Design*, Prentice-Hall Inc., Englewood Cliffs, NJ (1991).

Sargent, R.G., "Event Graph Modeling for Simulation with an Application to Flexible Manufacturing Systems," *Management Science*, Vol. 34, No. 10, pp. 1231-1251 (October 1988).

Schruben, L. and Yucesan, E., "Transforming Petri Nets into Event Graph Models," *Proceedings of the 1994 Winter Simulation Conference*, Lake Buena Vista, FL, pp. 560–565 (December 1994).

Teng, S.-H. and Black, J.T., "Cellular Manufacturing Systems Modeling: the Petri Net Approach," *Journal of Manufacturing Systems*, Vol. 9, No. 1, pp. 45–54 (1990).

Wadhwa, S. and Browne, J., "Modeling FMS with Decision Petri Nets," *The International Journal of Flexible Manufacturing Systems*, Vol. 1, No. 3, pp. 255–280 (June 1989).

Yamazaki Mazak Co., *Intelligent Mazatrol FMS*, Aichi, Japan (1991).

Yim, D.S. and Barta, T.A., "A Petri Net Based Simulation Tool for the Design and Analysis of Flexible Manufacturing Systems," *Journal of Manufacturing systems*, Vol. 13, No. 4, pp. 251–261 (1994).

Zeigler, B.P., *Object-Oriented Simulation with Hierarchical, Modular Models*, Academic Press Inc., San Deigo, CA (1990).

Zhou, M.C. and DiCesare, F., *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer Academic Publishers, Boston, MA (1993).