Dynamic Modelling of Reconfigurable Manufacturing Systems with Petri nets towards System Development Automation Support

Linda L. Zhang
Department of Management
IESEG School of Management (LEM-CNRS)
Lille-Paris, France
l.zhang@ieseg.fr

Abstract
Recognizing the significance of reconfigurable manufacturing systems (RMSs) in dealing with the production of high product variety, many authors have been approaching RMSs from different aspects and by addressing various issues. However, limited attempts have been made in modeling the dynamics of RMSs, i.e., how system elements are configured based on given customer orders. In view of its importance and the lack of investigation, this study focuses on the dynamic modeling of RMSs, in attempting to support RMS development automation. To deal with the technical difficulties in the dynamic modeling, this study develops a formalism of colored, object-oriented Petri nets with changeable structures. The proposed modeling formalism and the corresponding dynamic modeling of RMSs are discussed using an illustrative case example.

Keywords
Reconfigurable manufacturing systems, reconfiguration, dynamic modeling, Petri nets, product variety.

1. Introduction
Reconfigurable manufacturing systems (RMSs) have been well recognized as an effective means for quickly producing a high variety of customized products at small quantities. An RMS allows for flexibility not only in producing an increased variety of products in low, yet unpredictable production volumes but also in changing the system itself through reconfiguring process modules in terms of both software and hardware. In this regard, an RMS is characterized by diverse reusable system components, high product variety, and differences among system components and their relationships in specific RMSs for producing different products. Since the concept was first discussed in (Koren and Ulsoy, 1997), a myriad of studies have been reported to approach RMSs from different aspects and by addressing varies issues. These issues include key enabling technologies (Mehrabi et al., 2000), system reconfiguration optimization (Youssef and ElMaraghy, 2006), workflow management (Zhang et al., 2010), reconfigurability index (Gumasta et al., 2011), RMS impacts (Koren and Ulsoy, 2002), and many others. For a comprehensive review of the literature on RMSs, the readers may refer to (Rehman and Babu, 2013). On the other hand, very limited attempts have been made to explore the dynamic modelling of RMSs, that is, the modelling of dynamic processes of reconfiguring system components based on given customer orders. As with structural representation, dynamic modelling presents itself as one important stage in developing complex systems (Arora and Kumar, 2000). In view of the lack of investigation and its importance, this study thus focuses on the dynamic modelling of RMSs, in attempting to support system development automation.

As a graphical and mathematical modelling tool, Petri nets (PNs) excel in modelling, simulating, and analysing complex systems. This study thus adopts PN technique to model the dynamic reconfiguration processes of RMSs. Resulting from RMS characteristics introduced above, there are several technical difficulties in modelling, including the organization of diverse reusable system components, the handling of product variety, and the representation of differences among specific systems reconfigured. To deal with these modelling difficulties, this study integrates the principles of several well-defined PN extensions, including coloured PNs (CPNs; Jesen, 1992), object-oriented PNs...
(OPNs; Wang, 1996), and PNs with changeable structures (Jiang et al., 1999), and develops a new formalism of coloured, object-oriented PNs with changeable structures (COOPNs-CSs). While object-oriented concept in OPNs is used to organize the large number of reusable system components, coloured tokens in CPNs are to capture product variety, and the structure change handling mechanism in PNs with changeable structures is used to model the differences among specific systems reconfigured for producing different customer orders.

The formalism of COOPNs-CSs and the corresponding dynamic modelling of RMSs are explained in detail using an illustrative case example, which is first introduced in the section below.

2. An Illustrative Example

The structure of a customized product, FP, from a product family is shown in Fig. 1. Also shown are the operations which are necessary for producing FP. Another customized product, FPV, can be obtained by replacing two units of part g with one unit part h, as indicated in the figure. For producing FPV, a new assembly machine, A_i, needs to replace the assembly machine, A_j, which is used to produce FP. The RMS for producing all customized products in the family consists of (1) two buffers B_a and B_b for raw materials and parts, (2) three vehicles V_a, V_b, and V_c for transporting tools, raw materials, and parts, (3) a set of 30 tools/fixtures T, \( \forall i = 1,2,..,30 \) for machining parts, (4) a set of 16 machining machines M_i, \( \forall i = 1,2,..,16 \) for manufacturing parts, and (5) a set of 13 assembly machines A_i, \( \forall i = 1,2,..,13 \) for forming component assemblies and final products.

![Diagram of a customized product FP and its operations](image)

Figure 1. A customized product FP and its operations

3. Definitions of COOPNs-CSs

Based on the principles of CPNs and OPNs, COOPNs are defined first. The formalism of COOPNs-CSs is then defined based on COOPNs and the structural change handling mechanism in (Jiang et al., 1999). In accordance with the principle of OPNs, each object (i.e., system component, such as machines, buffers, and vehicles) is modelled as a COOPN. A COOPN of any object includes the static structure of the object and dynamic changes after transition firing. In the following, machines are used to explain the definition of COOPNs.

In line with the semantics of OPNs, the static attributes of COOPN of a machining machine are generalized shown in Fig. 2. Because the specific members of a machine class have the same static attributes, this static COOPN model is the same with the model for the corresponding machine class.
Based on the principle of CPNs, different colours are assigned to tokens representing product variety; arc expressions are formulated to capture the number of tokens to be removed (or added) after transition firing. After transition firing, the dynamic COOPN of a specific machining machine $M_1$ is obtained shown in Fig. 3.

In accordance with the above definitions, the COOPN of an object $O$ in an RMS after the $i$th change is described as an 8-tuple: $COOPN_i = (SP_i, AT_i, IM_i, OM_i, IOR_i, C_{(i,j)}, E_{(i,j)}, M_{(i,j)})$. The first five elements, including (1) the state place set $SP_i$, (2) the activity transition set $AT_i$, (3) the input message place set $IM_i$, (4) the output message place set $OM_i$, and (5) the input and output relationships $IOR_i$ between places and transitions, determine the static structure of a COOPN. The other three elements, including the colour set $C_{(i,j)}$ after the $i$th change, the arc expression function $E_{(i,j)}$ of input and output relations between transitions and places after the $i$th change, and the initial marking set $M_{(i,j,0)}$ after the $i$th change, specify the dynamic characteristics of a COOPN, that is, a COOPN at a particular state.

To capture design changes in different customer orders and their influences in system reconfiguration, in the proposed formalism proposed, representing design parameter and value pairs describing customized products, $PV_i$ is used to connect places and transitions. Further, different colours are assigned to different $PV_i$. Thus, for each design change, the corresponding changes in manufacturing resources, routing, operation types, and sequences can be determined.
efficiently. In another words, by assigning different colours to different design parameter and values pairs, the formalism can shed light on the interconnections between product design and and the corresponding manufacturing systems to be reconfigured.

Structure changes: The changes in an RMS may be caused by either adding new manufacturing resources, such as, machines, tools, and fixtures, or changing operation types, sequences, and routings. The mechanism to handle system structural changes in (Jiang et al., 1999) is adopted in this study. In their mechanism, a system is described by constituent objects; the relationships among objects are represented by message passing relations between sending and receiving objects. The mechanism deals with system model’s structural changes by (1) modifying message passing relations or (2) adding or removing objects and the relevant connections with other unchanged objects or (3) both. This mechanism of handling structural changes is incorporated in COOPNs, forming COOPNs-CSs.

4. Modeling RMSs with COOPNS-CSs

The dynamic modelling of RMSs includes two parts: the modelling of the initial system before reconfiguration and the modelling of the changed system after reconfiguration.

4.1. Modelling the Initial RMS before Reconfiguration

To construct the initial model of the RMS from the given illustrative example, COOPNs of objects are first built based on the definitions given in Section 3 to model manufacturing resources, including machining machines, assembly machines, three types of vehicles, two types of buffers, and tools/figures. Because the states and activity transitions of the three types of vehicles are similar in nature, one general COOPN is used to illustrate the COOPNs of the three types of vehicles. This is the same for the other resources. Four general COOPNs are built to represent the static COOPNs of assembly machines, buffers, vehicles, and tools, as shown in Fig. 4. (The static COOPNs model of machining machines is shown in Fig. 2.)

![Diagram of COOPNs for manufacturing resources](image_url)

Figure 4. Static COOPNs of manufacturing resources

To capture the relationships between two objects, a special type of transitions, gate set \( G_{gdr} \), is introduced in the formalism. A gate \( g \) passes messages from sending objects to receiving objects. While the out arc connection links \( g \) with the output message place of the sending objects, the in arc connection joins \( g \) to the input message places of receiving objects. Fig. 5 shows the initial COOPNs-CSs of the RMS for producing \( FP \). Thanks to the advantage of object-oriented modelling, that is, the encapsulation of private or internal behaviours of an object, a rounded rectangle

© IEOM Society International
with two interfaces (i.e., input and output message places) is used to represent COOPN of an object shown in the figure. With “0” indicating the initial status, the initial COOPNs-CSs model of the RMS is defined below.

1) The object set \( O_0 = (B_a, b, P_a, V_a, V_m, T_1, T_2, T_3, T_4, A_1, A_2, A_3, A_4, M_1, M_2, M_3) \)

2) The colour set \( C_0 = (PS_0, RS) \), where \( PS_0 = (RM_a, RM_b, RM_m, RM_v, P_aV_a, P_bV_b, P_vV_v, P_2V_2, P_3V_3, P_4V_4, P_1V_1, P_3V_3, P_5V_5, P_9V_9, SA_1, SA_2, SA_3, FP) \)

represents the set of product states in the initial system that is ready to produce \( FP \) and is determined by the product structure in Fig. 1: \( RS = (a) \) indicates the set of resource state with \( a \) denoting their availability. Thus, \( C_0 = (PS_0, RS) = ((RM_a, RM_b, RM_m, RM_v, P_aV_a, P_bV_b, P_vV_v, P_2V_2, P_3V_3, P_4V_4, P_1V_1, P_3V_3, P_5V_5, P_9V_9, SA_1, SA_2, SA_3, FP)) \).

3) The message passing relation \( R_0 \):

\[
R_0 = (R_{bB_0}, R_{bP_0}, R_{P_0V_0}, R_{P_0V_1}, R_{P_0V_2}, R_{P_0V_3}, R_{P_0V_4}, R_{P_0V_5}, R_{P_0V_6}, R_{P_0V_7}, R_{P_0V_8}, R_{P_0V_9}, R_{P_0V_{10}}, R_{P_0V_{11}}, R_{P_0V_{12}}, R_{P_0V_{13}}, R_{P_0V_{14}}, R_{P_0V_{15}}).
\]

To explain the formulation of message passing relations, \( R_{v,M,i} \) between object \( V_a \) and \( M_i \), is used as an example. \( R_{v,M,i} \) is represented by 1) \( FA_{v,M,i} \) to arc set from \( V_a \)’s output message place \( om^v \) to gate \( G_{v,M,i} \), 2) gate set \( G_{v,M,i} \), 3) \( IA_{v,M,i} \) in arc set from \( G_{v,M,i} \) to \( M_i \)’s input message place \( im^{M_i} \), and 4) \( EV_{v,M,i} \), expression function of in and out arcs. As shown in Fig. 5, out arc expression \( EV_{v,M,i}(OA_{v,M,i}) \) is defined as \( EV_{v,M,i}(OA_{v,M,i}) = \Gamma_{RM_a} \lor \Gamma_{RM_b} \lor \Gamma_{RM_m} \lor \Gamma_{RM_v} \lor \Gamma_{RM_{M_1}} \), and in arc expression is defined as \( EV_{v,M,i}(IA_{v,M,i}) = \left((1T_i + 1RM_a) \lor (1T_i + 1RM_b) \lor (1T_i + 2RM_m) \lor (1T_i + 2RM_v) \lor (1T_i + 2RM_{M_1}) \right) \cdot EV_{v,M,i}(OA_{v,M,i}). \)

\( OA_{v,M,i} = (om^v)^{-} - gm^v \), \( IA_{v,M,i} = (gm^v - im^{M_i}) \).
Thus, \( R_{v,M,O} = (O_{A,M,O}, G_{v,M,O}, I_{A,M,O}, F_{v,M,O}) = ((\text{om}^+_{v} - g_{v}), (g_{v}, (g_{v} - \text{im}^{v}_{M})), ((1' \text{RM}, \lor 2' \text{RM}, \lor 1' \text{RM}, \lor 1' \text{RM})�), \) where “\( \lor \)" denotes 

\( (1' T_{i} + 1' \text{RM}_{i}) \Rightarrow (1' T_{i} + 1' \text{RM}_{i} \lor 1' \text{RM}_{i} \lor 1' \text{RM}_{i}) \).

OR, “\( \land \)" means AND, “\( \Rightarrow \)" means if (the precondition text before \( \Rightarrow \) ) then (the consequence text after \( \Rightarrow \) ).

\( E_{v,M,O}(O_{A,M,O}) \) indicates four types of tokens representing four choices at \( \text{om}^+_{v} \). Among the four choices, only the second one, \( \text{RM}_{i} \), has two token elements. When a token, e.g., \( \text{RM}_{i} \), presents at \( \text{om}^+_{v} \), the selected arc expression will be \( 1' \text{RM}_{i} \). The \( E_{v,M,O}(I_{A,M,O}) \) indicates that if a token, \( \text{RM}_{i} \), at \( \text{om}^+_{v} \) together with another one \( T_{i} \) at output message place, \( \text{om}^+_{v} \) of \( V \) appears at the same time, \( g_{v} \) will be fired. The firing results in two tokens, \( \text{RM}_{i} \) and \( T_{i} \), to be removed from \( \text{om}^+_{v} \) and \( \text{om}^+_{v} \) and subsequently to be added to \( \text{im}^{v}_{M} \). It indicates that \( M_{i} \) and \( T_{i} \) are available and ready to machine raw material \( \text{RM}_{i} \).

4) The input and output logic relationship function set \( L_{o} \) of the gate set \( G_{o} \):

\( G_{o} = (g_{\cdot}, g_{\cdot}, g_{\cdot}, g_{\cdot}, g_{\cdot}, g_{\cdot}, g_{\cdot}, g_{\cdot}) \),

\( L_{o} = \bigcup_{i=1}^{g_{\cdot}} L_{o}(g_{\cdot})\), where \( g_{\cdot} \) is the number of gates.

\( L_{o} \) is used to specify the token flows that go from message sending objects through gates to receiving objects. \( g_{\cdot} \) is used as an example to explain this concept.

\( g_{\cdot}^{+} = (\text{om}^+_{v} \land \text{om}^+_{v}) \) : the output message place set connected with \( g_{\cdot} \),

\( g_{\cdot}^{+} = (\text{im}^{v}_{M} \lor \text{im}^{v}_{M} \lor \text{im}^{v}_{M}) \) : the input message place set connected with \( g_{\cdot} \), where “\( \land \)" stands for AND. Thus, \( L_{o}^{+} = (L_{o}(g_{\cdot}^{+})) \).

Finally, \( L_{o} \) can be obtained by applying this formulation to other gates involved.

5) The initial marking set \( M_{s_{o}} : M_{s_{o}} = (MM_{s_{o}}, SM_{s_{o}}) \), where \( MM_{s_{o}} = \phi \) represents the initial markings of place sets; \( SM_{s_{o}} \) represents initial markings of state places. While the first “0" in their subscripts indicates the number of system change, the second means the initial status after each system change.

\( SM_{s_{o}} = \{(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a), a + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a) + 1'(1' \text{P}^{w}_{1}, a)

To generalize, the initial COOPNs-CSs model \( S_{i} \) of an RMS after the ith change can be represented as a 5-tuple as follows:

\( S_{i} = (O, C, R, L, M_{s_{i}}) \), where \( i \) indicates the ith change of the system, after which all of the system model components, such as \( O, C, R, L \), and \( M_{s_{i}} \) are subject to change,

\( C = (PS, RS) \).

\( R = \bigcup_{i=1}^{O} (O \subset O, O_{m} \subset O, m \neq 0) = \bigcup_{i=1}^{\text{om}^{+}_{v}} (O \subset O, O_{m} \subset O, m \neq 0)

\( = \bigcup_{i=1}^{\text{om}^{+}_{v}} (\text{om}^{+}_{v} - g^{\cdot}) \lor (g^{\cdot} - \text{im}^{+}_{M}), (E_{o_{v}}, E_{o_{v}}) \lor (E_{o_{v}}, E_{o_{v}})) \lor (O_{m} \subset O, O_{m} \subset O, m \neq 0).

\( L = \bigcup_{i=1}^{O} (L_{o}(g_{\cdot}^{+}), L_{o}(g_{\cdot}^{+})) = \bigcup_{i=1}^{O} (\text{om}^{+}_{v} \lor \text{om}^{+}_{v}, \text{om}^{+}_{v} \lor \text{im}^{+}_{M}, \text{im}^{+}_{M} \lor \text{im}^{+}_{M}, \text{im}^{+}_{M} \lor \text{im}^{+}_{M}). \)

\( g_{\cdot}^{+} \) : the output message place set connected to \( g_{\cdot} \) after the ith change,

\( g_{\cdot}^{+} \) : the input message place set connected to \( g_{\cdot} \) after the ith change, and

\( M_{s_{i}} = (MM_{s_{i}}, SM_{s_{i}}) \).

4.2. Modelling the RMS after Reconfiguration

Suppose at a state that the machining of part \( a \) and the assembling of subassembly \( SA_{k} \) have been finished and they are being transferred by vehicle \( V_{r} \) to assembly machine \( A_{r} \), an urgent order for FPV comes to the shop floor. Therefore, instead of using 2 units \( g \) to assemble FP by \( A_{r} \), one unit \( h \) is used to assemble FPV by \( A_{r} \). Assume this is the first change to the system (i.e., \( i=1 \)). In accordance with this change, the initial COOPNs-CSs model needs to be changed as well. The dynamic COOPNs-CSs model of the RMS after the reconfiguration change, \( S_{1} = (O, C, R, L, M_{s_{1}}) \), is constructed, as shown in Fig. 6. The change of replacing \( A_{r} \) with \( A_{s} \) and \( g \) with \( h \) enables
to change the model by both adding, removing object and modifying the message passing relations. Along with the methods used, this new model, $S_t = (O_t, C_t, R_t, L_t, M_{t,0})$, is described below.

Figure 6. The dynamic COOPNs-CSs model of the RMS after reconfiguration change

1) The new object set $O_t$ is obtained by removing $A_t$ and adding $A_{t'}$. $O_t^\omega = (A_t)$: the set of removed objects,
$O_t^{\omega'} = (A_{t'})$: the set of added objects,
$O_t = O_t - O_t^\omega \cup O_t^{\omega'} = O_t - (A_t) \cup (A_{t'}) = (B_x, B_y, V_x, V_y, T_x, T_y, A_x, A_y, A_{t'}, M_{t'}, M_{t_1})$.

2) By (1) removing the color set $C_t^\omega$ associated with the removed object set and (2) adding new color set $C_{t'}^{\omega'}$ associated with the added object set, the new color set $C_t$ is obtained.
$C_t^\omega = (P_x V_x, FP)$, $C_t^{\omega'} = (P_y V_y, FPV)$
$P_{S_0} = (R_{M_x, x}, R_{M_y, y}, P_{V_x, V_x}, P_{V_y, V_y}, P_{V_y, V_y}, P_{V_y, V_y}, P_{V_y, V_y})$
$C_t = C_t^\omega \cup C_t^{\omega'} = (P_{S_0}, RS) - (P_x V_x, FP) \cup (P_y V_y, FPV) = (P_{S_0}, RS)$,

3) The message passing relation set $R_t$ is obtained by removing message passing relations $R_t^\omega$ associated with the removal of object set and adding $R_{t'}^{\omega'}$ associated with the adding of object set.
$R_t = R_t - R_t^\omega \cup R_{t'}^{\omega'} = (R_{B_x, B_x}, R_{B_x, B_x}, R_{B_x, B_x}, R_{B_x, B_x}, R_{B_x, B_x}, R_{B_x, B_x}, R_{B_x, B_x}, R_{B_x, B_x}, R_{B_x, B_x}, R_{B_x, B_x})$
$= (R_{B_y, B_y}, R_{B_y, B_y}, R_{B_y, B_y}, R_{B_y, B_y}, R_{B_y, B_y}, R_{B_y, B_y}, R_{B_y, B_y}, R_{B_y, B_y}, R_{B_y, B_y}, R_{B_y, B_y})$.

It is also necessary to modify the passing relations caused by the changing from g to h. These changes are reflected on the arc expressions. For example, changing the message passing relation $R_{B_x, B_x}$ between $B_x$ and $V_x$ needs to modify...
the corresponding arc expression to get the new relation $R_{h,v_0}$. $E'_{h,v_0}$ and $E_{h,v_0}$ are used to represent the removed and added arc expressions caused by the change from $g$ to $h$.

$$E_{h,v_0} = (E_{h,v_0}(O_{h,v_0})), \quad E_{h,v_0} = (E_{h,v_0}(I_{h,v_0}))).$$

No change is made to the out arc $O_{h,v_0}$, the in arc $I_{h,v_0}$, and the gate $g^s$ between $B_g$ and $V_P$. Thus, $R_{h,v_0} = (O_{h,v_0}, G_{h,v_0}, I_{h,v_0}, E_{h,v_0})$, where $O_{h,v_0} = (om^g_i - g^s_i)$, $G_{h,v_0} = g^s_i$, $I_{h,v_0} = (g^s - im^g_i)$.

The same way of modifying the message passing relations is applied to others that are required to change. Finally, $R_i$ is obtained.

4) The new input and output logic functions $L_i$ can be obtained by removing and adding input and output message places of the removed and added objects. Because the change to the system is the replacement of $A_i$ with $A_i$, the only gate to be considered is $g^s$. The new $L_i^s$ of $g^s$ is obtained as follows:

- $g^s_i = (om^g_i)$: the output message place set connected to $g^s$;
- $g^s_i = \phi$: the output message place set to be removed;
- $g^s_i = \phi$: the output message place set to be added;

thus, $L_i^s(\cdot g^s_i) = g^s_i - g^s_i \cup g^s_i = (om^g_i - \phi \cup \phi) = (om^g_i).

\[g^s_i = (im^g_i, im^g_i, im^g_i, im^g_i): \text{the input message place set connect to } g^s,\]

$$g^s_i = (\phi): \text{the input message place set to be removed},$$

$$g^s_i = (\phi): \text{the input message place set to be added},$$

thus, $L_i^s(\cdot g^s_i) = g^s_i - g^s_i \cup g^s_i = (im^g_i, im^g_i, im^g_i, im^g_i - (im^g_i) \cup (im^g_i) = (im^g_i, im^g_i, im^g_i, im^g_i),$

then, $L_i^s = (L_i^s(\cdot g^s_i), L_i^s(\cdot g^s_i)) = (om^g_i, im^g_i, im^g_i, im^g_i)).$

By replacing $L_i^s$ in $L_s$ with $L_i^s$, $L_i^s$ is obtained. This is because only the logic function of $g^s$ needs to be modified by the machine change.

5) The initial markings $M_{i,0}$ of the system after the change is formulated as follows:

$$MM_{i,0} = I(om^g_i, PV_a) + I(om^g_i, PaV_{h}) + I(im^g_i, PV_a) : \text{the markings of message places at the state is to be changed,}$$

$$MM_{i,0} = \phi : \text{the marking set of message places to be added,}$$

thus, $MM_{i,0} = MM_{i,0} + MM_{i,0} + MM_{i,0} = (I(om^g_i, PV_a) + I(om^g_i, PaV_{h}) + I(im^g_i, PV_a)) \phi + \phi = I(om^g_i, PV_a) + I(om^g_i, PaV_{h}) + I(im^g_i, PV_a).$

$$SM_{i,0} = \text{the marking set of places at the state that the system is to be changed,}$$

$$SM_{i,0} = I(P^g_i, a) + I(P^b_i, a) + I(P^b_i, a) + I(P^g_i, a) + I(P^b_i, a) + I(P^b_i, a) + I(P^b_i, a) + I(P^b_i, a) + I(P^b_i, a) + I(P^b_i, a) + I(P^b_i, a).$$

$$SM_{i,0} = I(P^g_i, a); \text{the removed marking set associated with the removed object,}$$

$$SM_{i,0} = I(P^g_i, a); \text{the added marking set associated with the added object,}$$

thus,

© IEOM Society International
5. Conclusions

This study developed a new formalism of coloured, object-oriented Petri nets with changeable structures to model the dynamic reconfiguration processes of reconfigurable manufacturing systems, in attempting to support system development automation. Besides the modelling of the dynamic reconfiguration processes, this study showed how specific products from customer orders can be linked with the suitable manufacturing systems.

As initial efforts, this study used a simple case example to validate the proposed PN formalism and the corresponding modelling. In the future, more efforts might be made in validate the formalism using more realistic settings from practice. Moreover, a computerized system based on the proposed formalism might be developed to provide decision making support in reconfiguring system components.

References


Biography

Dr. Linda Zhang is a Professor of Operations Management in the Department of Management at IESEG School of Management, Lille-Paris, France. She obtained her BEng and Ph.D. degrees in Industrial Engineering from China in 1998 and Singapore in 2007, respectively and Habilitation in Management Sciences from France in 2012. Her research interests include manufacturing system design and modeling, operations management, and supply chain management. On these areas, she has published many articles in international refereed journals, such as Decision Support Systems, IE Transactions, IEEE Transactions on Engineering Management, European Journal of Operational Research, International Journal of Production Economics, etc. Dr. Zhang has the extensive teaching experiences in a number of countries, including the Netherlands, France, Austria, Singapore, and China. She has taught courses at the undergraduate, graduate and postgraduate levels.