INT. J. PROD. RES., 1998, VOL. 36, NO. 12, 3323–3346

A Petri-net-based structure for AS/RS operation modelling

S. HSIEH†*, J.-S. HWANG† and H.-C. CHOU†

This paper presents a Petri-net (PN)-based hierarchical structure for AS/RS operation modelling. The complete AS/RS operation is modelled by four different layer modules, including the command typing layer (CTL), the command profile layer (CPL), the command compilation layer (CCL) and the command execution layer (CEL). To simplify the structure, macro-places are created to represent complicated activities associated with some specific groups of functions. The complete modelling system is implemented in a compact 'Visual C'-based object-oriented computer program. The 'click' function is introduced to allow macro-places to be zoomed in for detailed internal activity visualization. In a typical AS/RS, operation commands are generated and issued by the management system. Each command is first typed and queued in the CTL module and then delivered one by one to the CPL module for work-domain (i.e. the starting and ending locations of the crane) evaluation. After the work-domain is decided, the information is delivered to the CCL module for command sequence compilation. The compiled command sequence is then executed in the CEL module by the crane device. By employing the proposed four-layer modelling procedure, basic AS/RS operation modules of any size can be developed and then assembled to construct a complete AS/RS-PN model. By carefully analysing the properties of the complete AS/RS-PN model, the robustness of the AS/RS structure is verified.

1. Introduction

Automated storage/retrieval systems (AS/RSs) are of great current interest due to many attractive benefits, e.g. lower cost of building and land, saving of labour, reduction of inventory levels, improvement of material tracking and higher system throughput. Once the system is installed and operated, the measurable benefit of an AS/RS is predominantly dependent on the control policies used. The control policy of an AS/RS is usually application-dependent and dynamic. To effectively operate an AS/RS, system simulations are often required to determine the best control policies. Thus, we seek ways to efficiently build simulation models for AS/RSs.

The control policies that determine the performance of an AS/RS have been studied by many researchers. Prior research has focused on the crane travel time. Bozer and White (1984) developed travel-time models that can be used to establish throughput standards for existing systems and evaluate the design configuration for new systems. Hwang and Lee (1990) presented a travel-time model that considered the operating characteristics of the stacker crane, including acceleration, deceleration and the maximum velocity restriction. Chang *et al.* (1995) proposed a travel-time model that considered various travel speeds with known acceleration/deceleration rates. Egbelu (1991) improved the system performance by dynamically determining

Revision received January 1998.

† Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan 10764, Republic of China.

* To whom correspondence should be addressed.

the dwell point of the stacker crane. Randhawa and Shroff (1995) pointed out that the system performance measures are affected by the design specification of an AS/RS. A simulation model was developed to evaluate several design specifications.

Despite the success of these studies, a comprehensive AS/RS simulation model has not yet been established since optimal control policies are dynamic and basic AS/RS operational modules common to all applications do not exist. In the study of the crane operation, it is found that the crane operation is quite basic and common to all applications. If the crane operation can be modelled in advance, the simulation model can be constructed with ease by combining the basic operational modules and the main control model.

Petri-nets (PNs) have been adopted as a general tool for modelling the operation of manufacturing systems in recent years (Moore and Gupta 1996). Archetti *et al.* (1991) adopted PN models and a stochastic optimization method to study optimal control policies of an AS/RS. Knapp and Wang (1992) and Lin and Wang (1995) used stochastic PNs (SPNs) to model AS/RS. A PN graph of an AS/RS leads to the generation of a reachability table that can be converted to a state representation of a Markov process model of the AS/RS. The efficiency, control rules, bay assignment and other performance issues can then be studied by using the SPN model. Chincholkar *et al.* (1994) and Chincholkar and Chetty (1996) adopted stochastic coloured PNs (SCPNs) as a modelling and analysis tool to construct an AS/RS model. The influence of crane operation modes and scheduling policies on system performance was studied. The invariants were used to study the structural properties of the SCPNs. Zhou (1996) presented a PN model for buffers and a methodology to include buffers in a system without introducing deadlocks or overflows. D'Souza and Khator (1994), Jafari (1992), Sun *et al.* (1994) and Zhou *et al.* (1993) used PNs to model flexible manufacturing cells/systems. Kazuo *et al.* (1993), Ramaswamy and Valavanis (1994) and Moore and Gupta (1995) proposed an extended PN model to analyse and simulate the material handling system. Although some excellent works in material flow modelling have been published, little effort has been focused on the development of a generic AS/RS model structure.

In a typical AS/RS of unit load type, the system can comprise rack structures, input/output buffer stations, crane machines, sub-system controllers and the main controller of the system. A crane is installed and performs loading/unloading tasks in the aisle following the command issued by the sub-system controller. The sub-system controller receives commands from the main controller. Let a crane, its left and right side racks, input/output buffer stations and the sub-system controller be treated as a set, called the 'unit operation module'. Then, any complete unit load type AS/RS can be composed of n unit operation modules (n is the number of unit operation modules) and a main controller. The control strategies may vary from one system to another. The number and size of unit operation modules may be different. The basic elements of a unit operation module are the same. The PN model of the unit operation module for a general unit-load AS/RS can therefore be established in advance. By incorporating the pre-modelled unit operation modules and the main control model, an AS/RS model can be constructed.

In this paper, we propose a PN based four-layer structure for the AS/RS. We focus on modelling the unit operation module using a modular concept. Our purpose is to establish a unit operation module for use in a general AS/RS model. With this unit operation module structure, the AS/RS model of any unit-load type can easily be obtained by assembling these modules. PNs are adopted as the modelling tool and

the object-oriented 'Visual C^{++}' is used to generate PN graphs for detailed process visual verification. A built-in 'click' function is adopted for zoom-refinement that alllows a lower-level in-process model to be displayed in detail. To ensure the proposed four-layer structure is robust, structural properties are analysed.

2. Petri-net modelling

Petri-nets theory is a widely used tool for modelling complex systems of interacting concurrent components. The net theory allows a system to be represented by a mathematical model. Important information of the system concerning basic structures and dynamic behaviours is well characterized by the net model. Thus, the model can be used for structural analysis and/or performance evaluation. Some definitions of PN structure are given (Peterson 1981, Murata 1989) as follows:

Definition 1: A PN structure, N, is a 4-tuple, $N = (P, T, F, G)$ where :

(1) $P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places, $m = 0$;
(2) $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions, $n = 0$. $P \cap T = \varnothing$ and $= \cup T \neq \varnothing$;
(3) $F \subseteq (P \times T)$ is a function from P to T, commonly known as an input function;
(4) $G \subseteq (T \times P)$ is a function from T to P, commonly known as an output function.

Definition 2: A marking μ of a PN is a function $\mu : P \rightarrow Q$, $Q = \{0, 1, 2, \ldots\}$, $\mu_i = \mu(p_i) \in Q$. In a system with m places, the mapping function μ consists of m elements that define the state of the places and tokens at a particular time, $\mu = (\mu_1, \mu_2, \ldots, \mu_m)$, $\mu_i = \mu(p_i)$, $p_i \in P$. A PN structure N that contains a marking μ is marked PN. A marked PN is denoted by $M = (N, \mu)$.

The behaviour of dynamic systems can be described in terms of system states and their changes. In a dynamic system, a state or marking in a PN is changed according to the transition firing rules:

(1) a transition t is said to be enabled if each input place p of t is marked with at least $w(p, t)$ tokens, where $w(p, t)$ is the weight of the arc from p to t;
(2) an enabled transition may or may not fire (depending on whether or not the event actually takes place);
(3) a firing of an enabled transition t removes $w(p, t)$ tokens from each input place p of t and adds $w(t, p)$ tokens to each output place p of t, where $w(t, p)$ is the weight of the arc from t to p.

An inhibitor arc is a directed arc from a place p_i to a transition t_j with a small circle rather than an arrowhead at the transition. The small circle means 'not'. The firing rule is given as follows: a transition is enabled when tokens are in all of its normal inputs and zero tokens are in all of its inhibitor inputs; the transition fires by removing tokens from all of its normal inputs.

Definition 3: For a PN structure, N, with n transitions and m places, the incidence matrix $A = [a_{ij}]$ is an $n \times m$ matrix of integers and its typical entry is given by

$$a_{ij} = a_{ij}^{+} - a_{ij}^{-}$$

where $a_{ij}^{+} = w(i,j)$ is the weight of the arc from transition i to its output place j and $a_{ij}^{-} = w(i,j)$ is the weight of the arc to transition i from its input place j.

Definition 4: A marked graph (MG) is an ordinary PN, such that each place has exactly one input transition and exactly one output transition.

2.1. *Structural properties*

The structural properties of a system can be characterized by the topological relation of PNs. They are independent of the initial marking M_0 in the sense that these properties hold for any initial marking or are concerned with the existence of certain firing sequences from some initial marking. Thus, these properties can often be characterized in terms of the incidence matrix and its associated homogeneous equations or inequalities (Murata 1989). Some important structural properties are given by Murata as follows.

(1) *Structural liveness*: a PN N is said to be structurally live if there exists a live initial marking for N. A MG is structurally live.
(2) *Controllability*: a PN N is said to be completely controllable if any marking is reachable from any other marking. A MG is completely controllable iff Rank $A = m$.
(3) *Conservativeness*: A PN N is said to be (partially) conservative if there exists a positive integer $y(p)$ for every (some) place p such that the weighted sum of tokens, $M^{\mathrm{T}}y = M_0^{\mathrm{T}}y =$ a constant, for every $M \in R(M_0)$. A PN N is (partially) conservative iff there exists an m-vector y if positive (non-negative) integers such that $Ay = 0$, $y \neq 0$.
(4) *Repetitiveness*: a PN N is said to be (partially) repetitive if there exists a marking M_0 and a firing sequence σ from M_0 such that every (some) transition occurs infinitely often in σ. A PN N is repetitive iff there exists an n-vector x of positive (non-negative) integers such that $A_X^{\mathrm{T}} \geq 0$.
(5) *Consistency*: a PN N is said to be (partially) consistent if there exists a marking M_0 and a firing sequence σ from M_0, such that every (some) transition occurs at least once in σ. A PN N is consistent iff there exists an n-vector x of positive (non-negative) integers such that $A^{\mathrm{T}}x = 0$.

3. AS/RS-PN operation model

As mentioned before, an AS/RS consists of several unit operation modules. Every module is independent from each other. Whenever a job is requested and issued by the AS/RS management system, one of the operation modules will receive the command and execute the job. Therefore, instead of dealing with the AS/RS operations as a whole, one can begin by constructing each unit operation module first and then combining relevant unit operation modules with a management system to establish a complete AS/RS model.

An AS/RS crane usually executes task assignments in a fixed area within the scope of a single unit operation module. All activities of a stacker crane can be described by a unit operation module. Therefore, in this paper we should focus on the detailed model structure of a basic unit operation module and the related structural properties. The presentation focuses entirely on the structure of unit operation modules. The issues of scheduling policy, and the policies for operating and storage will not be addressed in this paper.

3.1. *A hierarchical four-layer structure*

Since a unit operation module comprises both the information flow component and the crane operation component, both components are required for a functional unit module. The tasks involved in the information flow component comprise the job request, destination location, current crane location, responses of sensors, and the crane status. The tasks involved in the crane operation components consist of the operations of storage, retrieval and homing. The storage task can be one of three different storage functions, including the function from rack to buffer and from buffer to rack, the function from buffer to rack, and the function from home to buffer, and then from buffer to rack. The retrieval operation also has three functions, including the function of rack–rack–buffer, the function of buffer–rack–buffer and the function of home–rack–buffer. The homing operation can be either a task from rack to home, or a task from buffer to home. The crane moves along the guidance path and loads/unloads parts from/to the rack and buffer. The crane returns to its home position once a homing command is executed. All of these activities are included in a basic unit operation model.

To simplify the model structure, macro functions are introduced and implemented into the model. After a detailed study on the activities of a basic unit operation module, these activities are further classified into four different hierarchical layers. The four-layer model is illustrated in figure 1 and is elaborated as follows.

(1) *Command Typing Layer* (*CTL*): in this layer, the operation command embedded in the information flow issued by the top management system is received, decoded, typed and queued in sequence. Once the command is decoded and typed, an end command token is issued to the top management system. The typed command is passed to the next layer for the working domain search, sequence compilation and action execution. The type of commands that can be detected in CTL include storage, retrieval, homing and end.
(2) *Command Profile Layer* (*CPL*): the work domain (i.e. the starting and ending locations) of a crane for each specific command is searched, and uniquely defined by the data of the current location and the end destination of the device.
(3) *Command Compilation Layer* (*CCL*): a sequence of crane actions that satisfy the requirement of a specific operation command is compiled and issued to the command execution layer.
(4) *Command Execution Layer* (*CEL*): in this layer, the physical actions of the crane device are carried out in sequence. These actions include the movement of a crane from its starting position to the end position and the performance of a part loading or unloading operation at a buffer station or rack.

A macro place in a layer is used to represent some complicated activities associated with a group of functions. A clicking on a macro place in the upper layer allows the user to bring up a refined structure and observe the activities in detail in the next lower layer. In order to do that, several conditions need to be satisfied:

(1) the input (output) transitions of the next lower layer model are the input (output) transitions of the upper-layer macro place;
(2) The initial marking of the first place in the next lower layer model is equivalent to the initial marking of the macro place in the upper layer model;

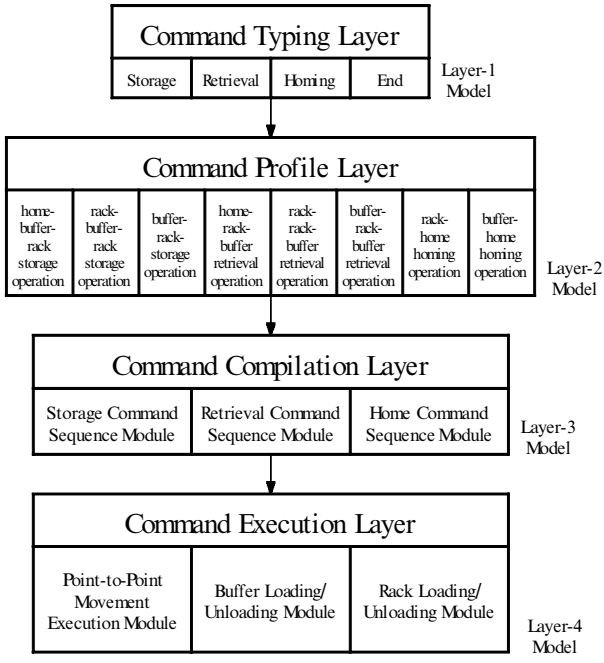

Figure 1. The AS/RS operational PN model structure.

(3) The final marking of the last place in the next lower layer model is equivalent to the final marking of the macro place in the upper layer model.

To make the proposed PN model simple in structure, various shape places and tokens, as given in figure 2, are used to represent different physical conditions. Although these places and tokens may be different in shape, their PN definitions are the same. To further distinguish the crane operation procedures from the information command flow, solid and dashed lines are used to represent the directed arcs of crane operations and the information commands, respectively.

To ensure the robustness of the proposed architecture, the structural properties are analysed for each layer of the model. Structural properties, e.g. liveness, controllability, conservativeness, repetitiveness and consistency are characterized by incidence matrices.

3.2. *Command typing layer*

The CTL model, as shown in figure 3, is a layer model that possesses two ordinary places and one macro place. Ordinary places W and F represent the command waiting state and the command complete state, respectively. Since a command operation macro place B1 is a kernel of the layer, it can be zoomed into the next layer for a refined profile picture. Transitions t_{00} and t_{03} could be transitions of a macro place in the top management system model. They are involved with the management policies and therefore they will not be discussed in this paper.

3.2.1. *Token movement rules in CTL*

Once a job is requested, a corresponding command is issued to a specific unit operation module in the system. This is done by firing the transition t_{00} in the

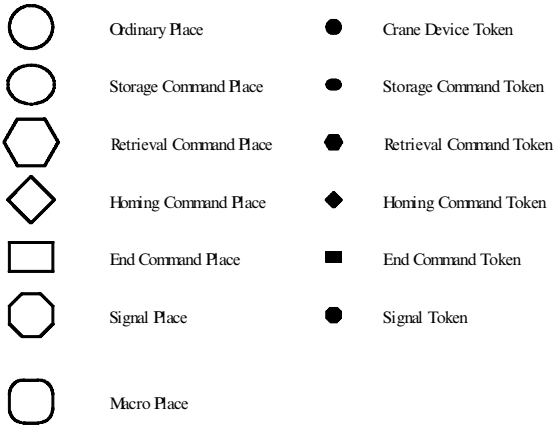

Figure 2. Physical meanings for different shapes of places and tokens in the AS/RS operational PN model.

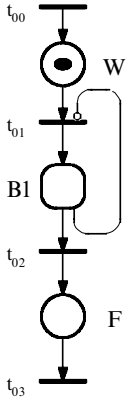

Figure 3. The command typing layer model. (Note: t_{00} and t_{03} are the input and output transitions of the CTL model.)

management system model. A corresponding operation command token for storage, retrieval or homing operation is deposited immediately into the place W. Since a unit operation module may receive several different commands from the top management system, the commands may stay in the waiting state and be queued in place W. Only one command can be executed at a time in a command operation macro place. The inhibitor arc from B1 to t_{01} is used to make sure at most one command token can stay in the command operation macro place B1. Once the transition t_{01} is fired, a command token is deposited to the command operation macro place B1. The received command is executed. An end command token is finally deposited to the ordinary place F to signal the completion of one operation command event. A job completion message is sent to the management system by firing transition t_{03}.

3.2.2. *Structural properties of the CTL*

The technique to analyse a net having an inhibitor arc is not currently available in literature. By eliminating the inhibitor arc, the CTL model becomes a MG. Different command types (i.e. storage, retrieval and homing) can best be represented

by tokens of different shapes. Using the coloured PNs presented by Jensen (1996), the CTL model can be expressed in terms of a 7×12 incidence matrix, which is given as

$$A_{\mathrm{CTL}} = \begin{array}{c} \\ t_{00} \\ \\ \\ t_{01} \\ \\ \\ t_{02} \\ \\ \\ t_{03} \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc|ccc|c} W & & & & B1 & & F \end{array} \\ \left[\begin{array}{ccc|ccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{array}\right] \end{array}$$

From this incident matrix, one can easily deduce several key structural properties of the CTL.

(1) *Liveness*: since the source transition t_{00} is unconditionally enabled, the CTL model is live.
(2) *Controllability*: since the CTL model is a MG and $\mathrm{rank}(A_{\mathrm{CTL}}) = 7$, the CTL mode is completely controllable.
(3) *Conservativeness*: it is not required for the CTL to have this property.
(4) *Repetitiveness*: there exists a 12×1 vector, $x = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T$ Since x is of positive integers and $A_{\mathrm{CTL}}^T x \geq 0$, the CTL model is structurally repetitive.
(5) *Consistency*: there exists a 12×1 vector, $x = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T$ Since x is of positive integers and $A_{\mathrm{CTL}}^T x = 0$, the CTL model is structural consistent.

3.3. *Command profile layer*

The CPL model possesses three types of function macros (i.e. storage, retrieval and homing). Judging from the current location of the crane device, one can further classify a storage function macro into three operations of different work domains. They are the operation of home–buffer–rack, the operation of rack–buffer–rack and the operation of buffer–rack. Similarly, a retrieval function macro can also be classified into three operations of different work domains, i.e. the operation of home–rack–buffer, the operation of rack–rack–buffer and the operation of buffer–rack–buffer. However, a homing function macro can only be classified into two operations of work domains including the operation of rack–home and the operation of buffer–home.

Figure 4 shows a complete CPL model. This model is composed of a crane model (i.e. the portion where places are connected to each other by solid lines) and three different control loops (i.e. the portion where places are connected by dashed lines). A crane model indicates the current position of a crane and determines the work

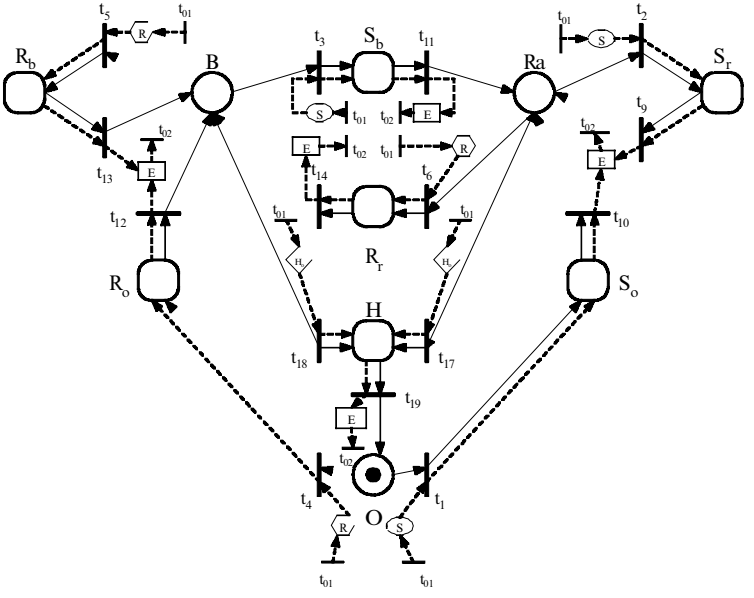

Figure 4. The command profile layer model. (Note: t_{01} and t_{02} are the input and output transitions of the CPL model.)

domain of a possible operation. The starting address of an operation in the CPL is the current position of the crane. The path and end position of an operation command are derived from one of eight possible transition operations. The ordinary places, O, Ra and B, stand for the physical positions of home, rack and buffer, respectively. The ordinary token in an ordinary place indicates the location of a crane device. The macro places, S, R and H, represent the storage operation macro, retrieval operation macro and homing operation macro, respectively. The subscripts, o, r and b attached to the macro places S and R, stand for the starting position at home, rack or buffer for each associated operation. The macro place H has two possible starting positions, at the buffer or rack.

As given in the CTL model of figure 3, when the transition t_{01} is fired, a command token is deposited to the macro place B1. After the refinement of macro place B1, the corresponding command token is moved directly into the CPL. When a job is completed, an end command token is generated in the CPL and then returned to the CTL. Figure 4 contains eight (three for storage, three for retrieval and two for homing) control loops that are used to convey the operation and the end command token from the CTL/CPL to the CPL/CTL. Besides transitions t_{01} and t_{02}, a storage control loop comprises a storage command place, end command place and storage operation macro place. Similarly, a retrieval control loop comprises transitions t_{01} and t_{02}, a retrieval command place, end command place and retrieval operation macro place. A homing control loop comprises transitions t_{01} and t_{02}, a homing command place, end command place and homing operation macro place. The physical meaning of each place in figure 4 is detailed in table 1.

3.3.1. *Token Movement Rules in CPL*

When a crane is idle, it can be at home, at a buffer or at a rack. An ordinary token residing in an ordinary place O indicates that the crane is at home. Once a job

Types	Symbols	Description
Ordinary places	O	home location
	B	I/O buffer station
	Ra	rack
Macro places	S_0	home–buffer–rack storage
	S_r	rack–buffer–rack storage
	S_b	buffer-rack storage
	R_0	home–rack–buffer retrieval
	R_r	rack–rack–buffer retrieval
	R_b	buffer–rack–buffer retrieval
	H	homing
Command place	S	storage command
	R	retrieval command
	H_0	homing command
	E	end command

Table 1. Elements of the command domain layer.

request command is detected and zoomed into the CPL, an operation command token is deposited to the command place through the corresponding command control loop. However, the crane device token remains at one of the ordinary places. When the job is completed, the crane stays where it was and the operation command token is removed from the operation macro place. An end command token is deposited to the end command place and then returned to the CTL through the control loop. A crane token is deposited to an ordinary place (O, B, or Ra), where it waits for the next command. When a transition is fired, the command token moves along the directed dashed arc and the crane token moves along the directed solid arc.

3.3.2. *Structural Properties of the CPL*

The structural analysis of the CPL model can be performed by checking the crane model, three control loops and the complete model, respectively.

3.3.2.1. The crane model and its structural properties. The incident matrix of a crane model is given as

$$A_{\text{crane}} = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_9 \\ t_{10} \\ t_{11} \\ t_{12} \\ t_{13} \\ t_{14} \\ t_{17} \\ t_{18} \\ t_{19} \end{array} \begin{array}{c} \begin{array}{cccccccccc} O & Ra & B & S_0 & S_r & S_b & R_o & R_r & R_b & H \end{array} \\ \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

The structural properties of the crane model are addressed as follows.

(1) *Liveness*: the crane model is a closed system. A crane token is always in the system. Since the crane token is in an ordinary place, O, Ra or B and the initial marking M_0 is equal to [1 0 0 0 0 0 0 0 0 0], [0 1 0 0 0 0 0 0 0 0], or [0 0 1 0 0 0 0 0 0 0], the net is live. Thus, the model is structurally live.
(2) *Controllability*: since the crane model is not a MG, the completely controllable property canot be identified.
(3) *Conservativeness*: there exists a 10×1 vector $y = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$ Since $y \in N^+$ and $A_{\text{crane}} y = 0$, the crane model is strictly conservative. This means that the number of cranes in CPL is a constant value of 1. This ensures the existence in the system.
(4) *Repetitiveness*: there exists a 15×1 vector $x = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2]$ Since x is of positive integers and $A^T_{\text{crane}} \geq 0$, the crane model is structurally repetitive.
(5) *Consistency*: there exists a 15×1 vector $x = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2]$ Since x is of positive integers and $A^T_{\text{crane}} x = 0$, the crane model is structurally consistent.

3.3.2.2. Three control loops. Since the structures of the three control loops are the same, we use the storage control loop as an example. The incident matrix of the storage control loop is given as:

$$A_{\text{CL}} = \begin{array}{c} \\ t_{01} \\ t_1 \\ t_2 \\ t_3 \\ t_9 \\ t_{10} \\ t_{11} \\ t_{02} \end{array} \begin{array}{c} \begin{array}{ccccc} S & E & S_0 & S_r & S_b \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

The structural properties of the control loop are addressed as follows.

(1) *Liveness*: each control loop contains a source and sink transition. Hence the control loop is structurally live.
(2) *Controllability*: since the control loop is not a MG, the completely controllable property cannot be identified.
(3) *Conservativeness*: there does not exist an m-vector, $m = 5$, $y \neq 0$, $y \in N^+$, that $A_{\text{CL}} y = 0$. Therefore, the control loop is not conservative.
(4) *Repetitiveness*: there exists an 8×1 vector $x = [3\ 1\ 1\ 1\ 1\ 1\ 1\ 3]$ Since x is of positive integers and $A^T_{\text{CL}} x \geq 0$, the control loop is structurally repetitive.
(5) *Consistency*: There exists an 8×1 vector $x = [3\ 1\ 1\ 1\ 1\ 1\ 1\ 3]$ Since x is of positive integers and $A^T_{\text{CL}} x = 0$, the control loop is structurally consistent.

3.3.2.3. The complete model. The incident matrix of the complete model is given as:

$$A_{\mathrm{CPL}} = \begin{array}{c} \\ t_{01} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_9 \\ t_{10} \\ t_{11} \\ t_{12} \\ t_{13} \\ t_{14} \\ t_{17} \\ t_{18} \\ t_{19} \\ t_{02} \end{array} \begin{array}{c} \begin{array}{ccccccccccccccc} O & Ra & B & S_o & S_r & S_p & R_o & R_r & R_b & H & S & R & H_o & E \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ -1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

The structural properties of the CPL model are addressed as follows.

(1) *Liveness*: each control loop contains a source transition. The crane token is always in the crane model. If an initial marking M_0 is equal to [1 0 0 0 0 0 0 0 0 1 0 0 0], the net is live. Thus, the CPL model is structurally live.

(2) *Controllability*: since the CPL is not a MG, the completely controllable property cannot be identified.

(3) *Conservativeness*: there exists a 14×1 vector $y = [0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0]$ Since y is of non-negative integers and $A_{\mathrm{CPL}}y = 0$, the CPL model is partially conservative. This is because the crane model is conservative and the three control loops are not.

(4) *Repetitiveness*: since both the crane model and three control loops are repetitiveness, the complete model must be repetitiveness. By observing the incident matrix A_{CPL}, there exists a 17×1 vector $x = [3\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 8]$ Since x is of positive integers and $A^T_{\mathrm{CPL}}x \geq 0$, the CPL model is structurally repetitive.

(5) *Consistency*: since both the crane model and three control loops are consistent, the complete model must be consistent. By observing the incident matrix A_{CPL}, there exists a 17×1 vector $x = [3\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 8]$ Since x is of positive integers and $A^T_{\mathrm{CPL}}x \geq 0$, the CPL model is structurally consistent.

3.4. *Command compilation layer*

Since there are three types of operation macros in CPL, three different command operation sequence modules (i.e. storage, retrieval and homing) are needed to be developed in CCL. It is noted that in CCL the command flow and corresponding operation are required to be synchronized. In other words, the command and operation tokens should occur in the same place at the same time.

3.4.1. *Storage command sequence module*

Once a storage command token and crane device token enter into a macro place (S_o, S_r, or S_b) in CPL, a storage operation command sequence immediately takes place in CCL. A storage command sequence module may possess four major functions including:

(1) requesting a crane to move from its current location to the I/O buffer;
(2) requesting a crane to load parts from the buffer;
(3) requesting a crane to move from the buffer to a designated storage bay location in the rack; or
(4) requesting a crane to unload parts to a storage bay.

A storage command sequence consists of a sequence of four action macros. The first action macro is a movement macro that moves a crane device from its current location to a buffer station. The second action macro is a loading macro that performs part loading at the buffer. The third action macro is a movement macro that moves a crane device from the buffer to a rack. The last action macro is an unloading macro that performs part unloading at the rack.

The storage command sequence module is illustrated in figure 5. Note that an input transition of a storage operation macro place, t_i (i = 1, 2 or 3), in CPL is an input transition of a storage command sequence module in CCL. Similarly, an output transition of a storage operation macro place, t_j (j = 9, 10 or 11), in CPL is an output transition of a storage command sequence module in CCL. Here, macro places, M_b and M_r, are used to stand for point-to-point (PTP) movement execution macros. The subscripts, b and r, are used to indicate the movement from the current location to a buffer station and from a buffer station to a rack, respectively. Macro place B_ℓ represents a buffer loading/unloading execution macro. The subscript ℓ is used to signal that the loading action is requested. Macro place R_u stands for a rack loading/unloading execution macro. The subscript u is to signal that an unloading

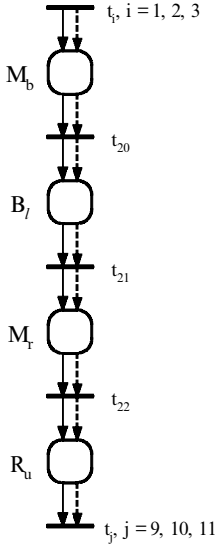

Figure 5. The storage command sequence module. (Note: t_i and t_j are the input and output transitions of the module, respectively.)

Point-to-point movement execution macro	M_b	to move a crane from its current location to a buffer station
	M_r	to move a crane from its current location to a rack
	M_o	to move a crane from its current location to home
Buffer loading/unloading execution macro	B_ℓ	to load parts to a ctrane at a buffer station
	B_u	to unload parts from a crane to a buffer station
Rack loading/unloading execution macro	R_ℓ	to load parts to a crane at a rack
	R_u	to unload parts to a crane at a rack

Table 2. Elements of the command compilation layer.

action is requested. The physical meaning of each storage command element in CCL is presented in table 2.

3.4.2. *Retrieval command sequence module*

The retrieval command sequence module is similar to the storage command sequence module. Whenever a retrieval command token and crane device token enter a macro place (R_o, R_r, or R_b) in CPL, the retrieval command sequence immediately takes place in CCL. A retrieval command sequence module may possess the following four major functions:

(1) requesting a crane to move from current location to a designated storage bay location in rack;
(2) requesting a crane to load parts from the storage bay;
(3) requesting a crane to move from the rack to the I/O buffer; or
(4) requesting a crane to unload parts to the buffer.

A retrieval command sequence consists of a sequence of four action macros. The first action macro is a movement macro that moves a crane device from its current location to a designated storage bay location in the rack. The second action macro is a loading macro that performs part loading at the rack. The third action macro is a movement macro that moves the crane device from the rack to the buffer. The last action macro is an unloading macro that performs part unloading at the buffer.

The retrieval command sequence module is illustrated in figure 6. Note that an input transition of a retrieval operation macro place, t_i (4, 5 or 6), in CPL is an input transition of a retrieval command sequence module in CCL. Similarly, an output transition of a retrieval operation macro place, t_j (j = 12, 13 or 14), in CPL is an output transition of a retrieval command sequence module in CCL. Here macro places, M_b, M_r, R_ℓ and B_u, are the same macro planes as those used in the storage command sequence module. The physical meaning of each retrieval command element in CCL is described in table 2.

3.4.3. *Homing command sequence module*

When a homing command token and crane device token enter a macro place, H, in CPL, the homing command sequence immediately takes place in CCL. A homing command sequence module may possess one of the following two functions:

(1) requesting a crane to move from its current buffer station back to its home location; or
(2) requesting a crane to move from its current rack location back to its home position.

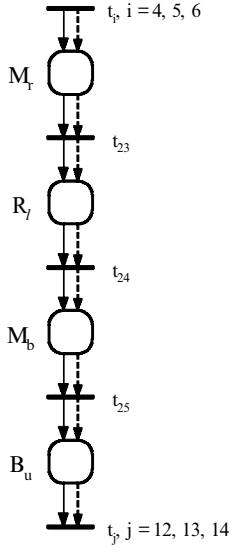

Figure 6. The retrieval command sequence module. (Note: t_i and t_j are the input and output transitions of the module, respectively.)

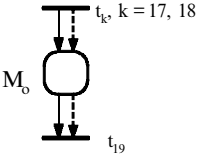

Figure 7. The homing command sequence module. (Note: t_i and t_j are the input and output transitions of the module.)

A homing command sequence consists of a single action macro. The action macro is a movement macro that moves a crane device from its current buffer/rack location to its home location. A homing command sequence module is given in figure 7. Note, a simple movement action from the current position to home position can satisfy the requirements of a homing command sequence module. Therefore, a homing command sequence module includes only one movement macro place, M_o. Transitions t_k (k = 17 or 18) and t_{19} are the input and output transitions of the module. The physical meaning of the homing command element is also presented in table 2.

3.4.4. *Token movement rules of the CCL*

The main function of the CCL is to decompose a requested job into a logic operation sequence. To allow a job to be executed by a crane device in CCL, a crane device token and command operation token must appear in the same place at the same time in CCL to enable a macro operation.

3.4.5. *Structural properties of the CCL*

All three operation modules in CCL have the same functional structure. For simplicity, the homing command sequence module is selected for illustration. The incidence matrix of the homing command module is given as

$$A_{\text{home}} = \begin{matrix} \\ t_k \\ t_{19} \end{matrix} \begin{matrix} \mathrm{M_o} \\ \begin{bmatrix} 2 \\ -2 \end{bmatrix} \end{matrix}.$$

Since all three modules have the same structure, their structural properties should also be the same. Based upon the above incident matrix of the homing command, one can easily reach the following conclusions for structural properties.

(1) *Liveness*: each module contains a source and sink transition. The CCL modules are structurally live.
(2) *Controllability*: since the CCL modules are not MGs, the completely controllable property cannot be identified.
(3) *Conservativeness*: there does not exist a 1×1 non-negative y vector, $y \neq 0$ such that $A_{\text{CCL}} y = 0$. Hence, the CCL modules are not conservative.
(4) *Repetitiveness*: there exists a 2×1 vector $x = \begin{bmatrix} 1 & 1 \end{bmatrix}$ Since x is of positive integers and $A_{\text{CCL}}^T x \geq 0$, the CCL modules are structurally repetitive.
(5) *Consistency*: there exists a 2×1 vector $x = \begin{bmatrix} 1 & 1 \end{bmatrix}$ Since x is of positive integers and $A_{\text{CCL}}^T x \geq 0$, the CCL modules are structurally consistent.

3.5. *Command execution layer*

Three types of tasks are required in CEL. These include: (1) the movement of a crane device from its current location to a 'specific location'; (2) the loading of parts onto the crane; and (3) the unloading of parts from the crane. Therefore, three corresponding operation modules are developed for the CEL. These operation modules include the point-to-point (PTP) movement execution module, he rack loading/unloading execution module, and the buffer loading/unloading execution module. These modules are responsible for the tasks of crane movement, part loading/unloading at a rack and part loading/unloading at a buffer station, respectively.

3.5.1. *PTP movement execution module*

Although there are eight types of movements in CPL, a PTP movement execution module can be used to describe all eight types of movements. A PTP movement execution module contains both the x-coordinate and y-coordinate ordinary places. These places are used to address the physical locations of bays in racks, the locations of buffer stations and the location of home. For instance, a unit operation module with a size of six × three is assumed having its home position located at a buffer station of $(x, y) = (0, 0)$. The addresses of three bays on three levels of the first column are $(x, y) = (1, 1)$, $(1, 2)$ and $(1, 3)$, respectively. Any location that a crane is allowed to move to is given an address in the model.

Besides the two (i.e. x-coordinate and y-coordinate) serial address places, some command operation signals are usually required to instruct a crane to move. A forward and backward series of signal places is constructed along each x- and y-coordinate address places. Figure 8 shows the basic unit of a PTP movement execution module. The module consists of two ordinary places and two command places. The ordinary places, X_0 and X_1, represent the two aligned positions in the same direction. The signal places, f_{x1} and b_{x0}, represent forward and backward moving signal places. If a signal token is deposited to the signal place, f_{x1} and a crane device token is in place X_0, the transition t_{x1} is fired and the crane is moved forward. If a signal token is deposited to the signal place, b_{x0} and the crane device token is in place

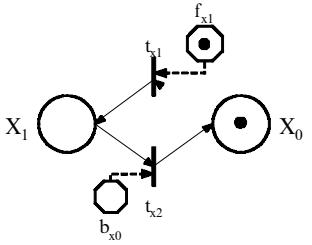

Figure 8. The basic unit of a PTP movement execution module.

X_1, the transition t_{x2} is fired, and the crane is moved backward. After the transition, t_{x1} or t_{x2}, is fired, the crane token is deposited to the output place of the transition and then the signal token is removed.

When a crane device token and command token enter a movement macro place in CCL, the macro place is zoomed in and these tokens are moved immediately into the PTP movement execution module. In the module, a pair of crane device tokens is deposited into its current address. To generate the 'from-to' guide path, the command token is split into two (x-signal and y-signal) series of forward or backward signal tokens and are deposited into the associated signal places.

A PTP movement execution module for a unit operation module of m by n is composed of m basic units in the x-direction and n basic units in the y-direction. Figure 9 shows a PTP movement execution module of 4 columns ($m = 4$) by 3 rows ($n = 3$). The home of the module is located at a buffer station of $(X, Y) = (0, 0)$. The X serial nets represent the x-direction movement execution nets and the Y serial nets represent the y-direction movement execution nets. Note that the transitions t_{in} and t_{tout} are connected to the first and last places of the module. In the module, there is a pair of the first places, X_i and Y_j, i is an integer, which represents the current location of the crane and a pair of the last places, X_k and Y_ℓ, ℓ is an integer, representing the destination of the movement task.

3.5.2. *Token movement rules in PTP movement module.*

The token movement rules in this module can be illustrated by using the following example. If the current crane location is at $(X, Y) = (0, 0)$ and the crane is instructed to move to the second column in the first row of the rack, the from–to path is (0, 0) to (2, 1). A pair of crane device tokens will be deposited into places X_0 and Y_0. Both x and y signal serial tokens will be deposited into forward signal places f_{x1} and f_{x2}, for x serial and place f_{y1} for y serial and the input transition, t_{in}, of this module is directed to these ordinary and signal places, respectively. The result is given in figure 9(a). When the crane arrives at $(X, Y) = (2, 1)$, the pair of crane device tokens is at places X_2 and Y_1 and all the signal tokens are removed. At this moment, the crane arrives at the 'to' position. The output transition, t_{out}, of this module receives these directed arcs from the pair of the ordinary places representing the 'to' position and signal places f_{x1}, f_{x2} and f_{y1}. Thus, the command and crane device tokens are retrieved through the transition t_{out}. The result is given in figure 9(b).

3.5.3. *Structural properties of the PTP movement module*

The PN of this module is varying with the size (i.e. the numbers of rows and columns) of the AS/RS. In this module, the pair of output places of transition t_{in} and the pair of input places of transition t_{out} are determined by the from–to path.

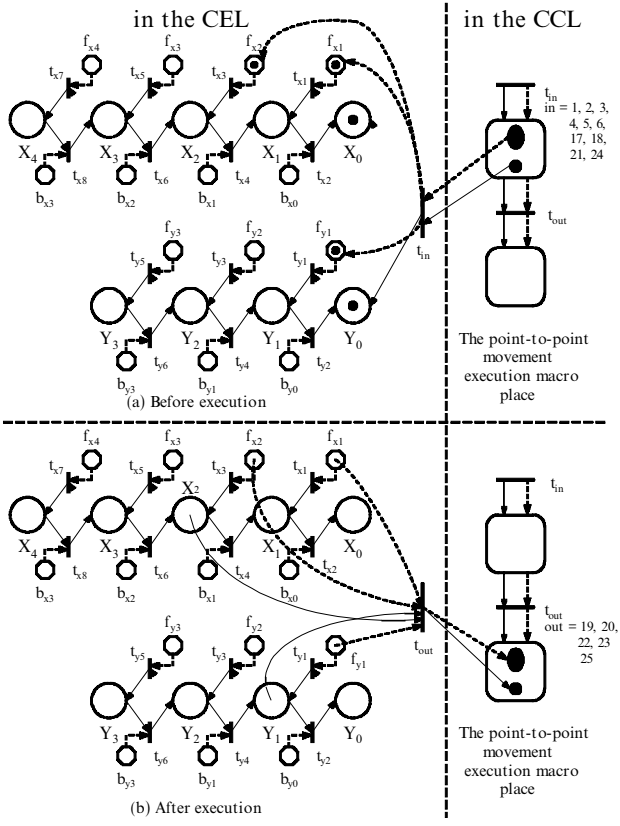

Figure 9. A 4×3 PTP movement execution module.

However, the basic structure of the module is the same. When a PTP movement execution macro place in CCL is zoomed in, all the necessary signal tokens occur in the signal places at once. Once a series of transitions is fired one by one, the crane token proceeds to the next ordinary place step by step and the signal token is deleted one by one. The signal place is an auxiliary place in CEL. Therefore, we remove signal places from the model when the structural properties are analysed. Since the from–to path is from $(0, 0)$ to $(2, 1)$, the output places of t_{in} include place X_0 and place Y_0 and the input places of t_{out} include place X_2 and place Y_1 The PN structure is structured and presented in figure 10. The incident matrix of the execution module is given as

$$A_{\text{POP}} = \begin{array}{c} \\ t_{\text{in}} \\ t_{x1} \\ t_{x2} \\ t_{x3} \\ t_{x4} \\ t_{y1} \\ t_{y2} \\ t_{\text{out}} \end{array} \begin{array}{c} \begin{array}{ccccc} X_0 & X_1 & X_2 & Y_0 & Y_1 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 0 & -1 \end{bmatrix} \end{array}$$

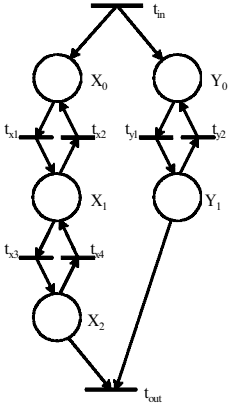

Figure 10. The effective PN structure of the PTP movement execution module.

Based upon the incident matrix given above, one may summarize the structural properties of the PTP movement module as follows.

(1) *Liveness*: each module contains a source and sink transition. The CCL modules are structurally live.
(2) *Controllability*: since the module is not a MG, the completely controllable property cannot be identified.
(3) *Conservativeness*: there does not exist a 5×1 non-negative y vector and $y \neq 0$ such that $A_{\mathrm{POP}}y = 0$. Therefore, the CCL modules are not conservative.
(4) *Repetitiveness*: there exists an 8×1 vector $x = [1\,2\,1\,2\,1\,2\,1\,1]^T$ Since x is of positive integers and $A_{\mathrm{POP}}^T x \geq 0$, these modules are structurally repetitive.
(5) *Consistency*: there exists an 8×1 vector $x = [1\,2\,1\,2\,1\,2\,1\,1]^T$ Since x is of positive integers and $A_{\mathrm{POP}}^T x = 0$, the CCL modules are structurally consistent.

3.5.4. *Rack loading/unloading execution module*

A rack loading/unloading module is also composed of the device operation and command operation. The device operation is used to instruct the crane device to perform loading/unloading tasks and the command operation is to provide the crane with two kinds of information (i.e. position confirmation and loading/unloading signal). A crane device token appearing in an ordinary place of the module represents the status of the physical crane. Two types of signal places, i.e. the position confirmation place and loading/unloading signal place, are used in the module. A signal token in a position confirmation signal place confirms the position that a part is to be deposited to or to be withdrawn from. For instance, when the crane arrives at a position, a token in the signal place, called 'Left', will direct the crane to serve the rack on the left-hand side. A token in the loading or unloading signal place will direct the crane for loading or unloading operations. The detail picture of a rack loading/unloading execution module is given in figure 11. All elements of a rack module are detailed in table 3.

3.5.5. *Buffer loading/unloading execution module*

The buffer loading/unloading execution module is similar to the rack loading/unloading execution module. The operations in the module can be classified into two categories, i.e. the device operation and command operation. They have the same

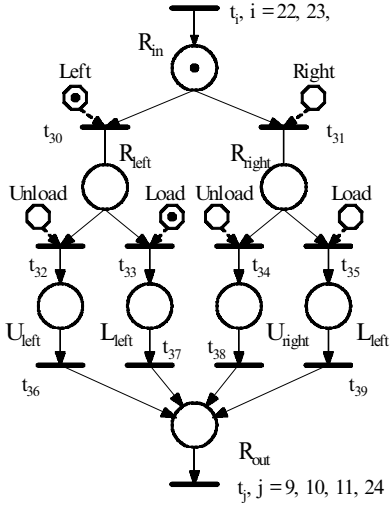

Figure 11. The rack loading/unloading execution module.

Ordinary places	R_{in}	a crane is located at the rack
	R_{left}	a crane is ready to serve the track on the left-hand side
	R_{right}	a crane is ready to serve the rack on the right-hand side
	U_{left}	a crane is to unload the part to the left-hand side of the rack
	L_{left}	a crane is to load the part from the left-hand side of the rack
	U_{right}	a crane is to unload the part to the right-hand side of the rack
	L_{right}	a crane is to load the part from the right-hand side of the rack
	R_{out}	a crane completes a loading/unloading task at the rack
Command places	Left	to instruct a crane to serve the rack on the left-hand side
	Right	to instruct a crane to serve the rack on the right-hand side
	Load	to instruct a crane to perform a loading task
	Unload	to instruct the crane to perform an unloading task

Table 3. Elements of the rack loading/unloading execution module.

physical meanings as those in the rack loading/unloading module. A signal token in the signal place indicates the crane is at the right position ready for buffer service. A token in the loading/unloading signal place can instruct the crane to perform loading/unloading operations. Usually there exists only one I/O buffer station in each unit operation module. For some specific cases, it is possible to have more than one buffer station in each unit operation module. Figure 12 shows the buffer loading/unloading execution macro for one I/O station. Table 4 lists all elements of the module.

3.5.6. *Token movement rules in rack (buffer) loading/unloading execution module*

The rules for token development in both the rack and buffer loading/unloading execution modules have the following distinguished features.

(1) When a command token enters a rack (or buffer) loading/unloading execution macro place from the CCL, a command token is split into the position confirmation token and loading/unloading token.

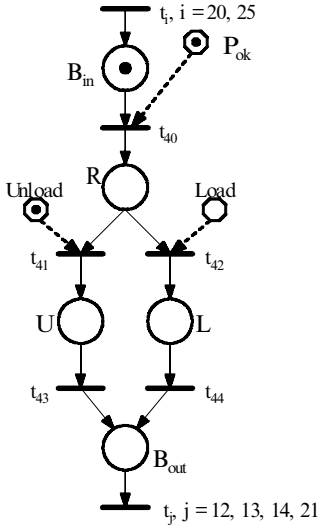

Figure 12. The buffer loading/unloading execution module.

Ordinary places	B_{in}	a crane is located at a buffer
	R	a crane is in the right position and ready to execute a task
	U	a crane is to unload the part to a buffer station
	L	a crane is to load the part from a buffer station
Command places	B_{out}	a crane completes a loading/unloading task at a buffer
	P_{ok}	to confirm that a crane is in the right position
	Load	to instruct the crane to perform a loading task
	Unload	to instruct a crane to perform an unloading task

Table 4. Elements of the buffer loading/unloading execution module.

(2) Each signal or ordinary place is restricted to have a single token. There is at most one device token appearing in the module since physically there is only one crane in existence.
(3) Whenever a transition is fired, the signal token is removed from the net and the crane device token is moved to the next ordinary place.
(4) Once a movement is executed, all of the signal tokens are removed. The original command token appears again and returns to the CCL along with the crane token.

3.5.7. *Structural properties of the rack (buffer) loading/unloading execution module*

The structures of the rack and buffer loading/unloading execution modules are similar. The rack module is the combination of two buffer modules. According to Murata's reduction rules (1989), the structures of the rack and buffer loading/unloading execution modules are the same. For simplicity, the buffer execution module is selected for illustration. Here, we remove signal places from the model when the structural properties are analysed. The incident matrix of the buffer execution module is given as

Property \ Layer/module	CTL	CPL			CCL	PTP	CEL
		Crane	CL	Crane + CL			Buffer/Rack
Conservative	-	+	-	+	-	-	-
Repetitive	+	+	+	+	+	+	+
Consistent	+	+	+	+	+	+	+
Controllable	+	≈	≈	≈	≈	≈	≈
Structurally live	+	+	+	+	+	+	+

Note: 'CL' represents control loop; 'PTP' represents the point-to-point movement execution module; 'Buffer/Rack' represents the buffer and rack loacing/unloading execution modules.

Table 5. Structural properties of the four-layer AS/RS model, where +, - and ≈in each row indicate holding, not holding, or undecided the structural properties in the row for each net indicated in the column.

$$A_B = \begin{array}{c} \\ t_i \\ t_{40} \\ t_{41} \\ t_{42} \\ t_{43} \\ t_{44} \\ t_j \end{array} \begin{array}{c} \begin{array}{ccccc} B_{in} & R & U & L & B_{out} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

Based upon the incident matrix given above, the structural properties of the buffer and rack loading/unloading execution modules can be easily observed and summarized as follows.

(1) *Liveness*: each module contains a source and sink transition. The two modules are structurally live.
(2) *Controllability*: Since the two modules are not MGs, the completely controllable property cannot be identified.
(3) *Conservativeness*: there does not exist a 5×1 non-negative vector y, and $y \neq 0$ such that $Ay = 0$. The two modules in CEL are not conservative.
(4) *Repetitiveness*: there exists a 7×1 vector $x = [2\,2\,1\,1\,1\,1\,2]$ Since x is of positive integers and $A_B^T x \geq 0$, the buffer module is structurally repetitive. The rack module can be identified to be structurally repetitive by the same reason.
(5) Consistency: there exists a 7×1 vector $x = [2\,2\,1\,1\,1\,1\,2]$ Since x is of positive integers and $A_B^T x = 0$, the module is structurally consistent. The rack module can be identified to be structurally consistent by the same reason.

The results of the above structural property analysis indicate that the proposed hierarchical four-layer AS/RS model structure is robust. The analysis results are summarized in table 5.

4. Conclusions

This paper presents a PN based hierarchical structure for AS/RS operation modelling. The proposed structure has several distinct features including: (1) the

approach is general, since the proposed unit operation module structure is common to every unit-load AS/RS; (2) the method is effective since the function of information flow is distinguished from the operation of the crane; (3) the implementation is systematic since a hierarchical four-level structure is adopted as the general AS/RS operation mode; (4) the system model is robust since the robustness of the system is insured layer by layer; (5) the system management policies can be incorporated and coordinated by employing the modular concept in the process of modelling; and (6) various basic operation modules of an AS/RS can be established first and then assembled to construct complicated AS/RS models.

Subsequent work on the analysis and evaluation of the performance of an AS/RS will be effectively done by extending the PN model to the coloured-timed PN model. For an AS/RS, the state of the system varies from time to time. Therefore, the PN structure should be equipped with temporal capabilities for modelling the dynamic aspects of AS/RSs. The use of asynchronous system models for modelling temporal behaviour requires that they be augmented by introducing a time parameter. This parameter provides a common frame for the expression of the speeds of their components. Also, since tokens are divided into two general categories (availability of shared resources and actions), different colours are used to distinguish different tokens.

References

Archetti, F., Sciomachen, A. and Gaivoronski, A. 1991, Optimal control policies for automated storage/retrieval system using PN models and stochastic optimization. In *Proceedings of the Fourth International Workshop on Petri Nets and Performance Models*, J. Billington, *et al.* (eds) (IEEE Computer Society, Melbourne, Australia) 2–5 December, pp. 258–267.

Bozer Y. A. and White, J. A. 1984, Travel-time models for automated storage/retrieval systems. *IIE Transactions*, **6**, 329–338.

Chang, D. T., Wen, U. P. and Lin, J. T. 1995, The impact of acceleration/deceleration on travel-time models for automated storage/retrieval systems. *IIE Transactions*, **27**, 108–111.

Chincholkar, A. K., Chetty, O. V. K. and Kuppuswamy, G. 1994, Analyses of an automated storage and retrieval system using stochastic coloured Petri nets. *Advances in Modelling & Analysis C:System Analyses, Control & Design*, **44**, 19–30.

Chincholkar, A. K. and Chetty, O. V. K. 1996, Simultaneous optimisation of control factors in automated storage and retrieval systems and FMS using stochastic coloured Petri nets and the Taguchi method. *International Journal of Advanced Manufacturing Technology,* **12**, 137–144.

D'Souza, K. A. and Khator, S. K. 1994, A Petri net approach for modelling controls of a computer-integrated assembly cell. *International Journal of Computer Integrated Manufacturing*, **6**, 302–310.

Egbelu, P. J. 1991, Framework for dynamic positioning of storage/retrieval machines in automated storage/retrieval system. *International Journal of Production Research*, **29**, 17–37.

Hwang, H. and Lee, S. B. 1990. Travel-time models considering the operating characteristics of the storage and retrieval machine. *International Journal of Production Research*, **28**, 1779–1789.

Jafari, M. A. 1992, An architecture for a shop-floor controller using coloured Petri nets. *International Journal of Flexible Manufacturing Systems*, **4**, 159–181.

Jensen, K. 1996, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use* (Berlin: Springer).

Kazuo W., Kenji, Y., Masako, N. and Yasuyoshi, K. 1993, Simulation system for material flow in FA line based on extended Petri nets. *Research and Development*, **43**, 54–57.

KNAPP, G. M. and WANG, H.-P. 1992, Modeling of automated storage/retrieval systems using Petri nets. *Journal of Manufacturing Systems*, **11**, 20–29.

LIN, S.-C. and WANG, H.-P. 1995, Modeling an automated storage and retrieval system using Petri nets. *International Journal of Production Research*, **33**, 237–260.

MOORE, K. E. and GUPTA, S. M. 1995, Stochastic coloured Petri net models of flexible manufacturing systems: material handling systems and machining. *Computers Ind. Engng*, **29**, 333–337.

MOORE, K. E. and GUPTA, S. M. 1996, Petri net models of flexible and automated manufacturing systems: a survey. *International Journal of Production Research*, **34**, 3001–3035.

MURATA, T. 1989, Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, **77**, 541–580.

PETERSON , J. L. 1981, *Petri Net Theory and the Modelling of Systems* (Englewood Cliffs: Prentice-Hall. NJ).

RAMASWAMY, S. and VALAVANIS, K. P. 1994, Modeling, analysis and simulation of failures in a materials handling system with extended Petri nets. *IEEE Transactions on Systems, Man and Cybernetics*, **24**, 1358–1373.

RANDHAWA, S. U. and SHROFF, R. 1995, Simulation-based design evaluation of unit load automated storage/retrieval systems. *Computer Ind. Engng*, **28**, 71–79.

SUN, T.-H., CHENG, C.-W. and FU, L.-C. 1994, A Petri net based approach to modeling and scheduling for an FMS and a case study. *IEEE Transactions on Industrial Electronics*, **41**, 593–601.

ZHOU, M. 1996, Petri net modelling of buffers in automated manufacturing systems. *IEEE Transactions on Systems, Man and Cybernetics, part B: Cybernetics*, **26**, 157–164.

ZHOU, M., MCDERMOTT, K. and PATEL, P. A. 1993, Petri net synthesis and analysis of a flexible manufacturing system cell. *IEEE Transaction on Systems, Man and Cybernetics*, **23**, 523–531.