A Coloured Petri Net Model for Automated Storage and Retrieval Systems Serviced by Rail-Guided Vehicles: a Control Perspective

Mariagrazia Dotoli, Maria Pia Fanti

Dipartimento di Elettrotecnica ed Elettronica

Politecnico di Bari

Via Re David, 200 – 70125 Bari, Italy

E-mail: {dotoli, fanti}@poliba.it,

Abstract—An Automated Storage and Retrieval System (AS/RS) automatically stores incoming material and retrieves stored parts with no direct human handling. This paper proposes a modular and unified modelling framework for heterogeneous automated storage and retrieval systems, comprising rail guided vehicles and narrow aisle cranes. We employ coloured timed Petri nets, representing a concise and computationally efficient tool for modelling the system dynamic behaviour, particularly suitable for real time control implementation. Indeed, the model can be utilized in a discrete event simulation to apply control policies in order to solve scheduling problems, as well as to avoid deadlock and collision occurrences.

Keywords: Automated storage and retrieval systems, coloured timed Petri nets, real time control.

1. Introduction

Automated storage and retrieval systems (AS/RSs) are widely used in warehousing and manufacturing systems for storing and retrieving finished products and parts. An AS/RS is a combination of automatic handling, storing/retrieval equipment and control systems, characterized by high accuracy and speed. Typically, an AS/RS consists of several aisles with storage racks on either side, each serviced by a crane (or, equivalently, S/R machine), operating storage and retrieval of the parts. The cranes move in three directions: along the aisle to perform transfers, sideways between the aisle and the racks at the sides, and vertically to reach the storage/retrieval location. Each aisle is equipped with a storage and a retrieval conveyor. Moreover, we suppose that the AS/RS aisles are serviced by Rail Guided Vehicles (RGVs), unloading the parts to be stored or loading them after retrieval (Lee et al. 1996, Amato and Basile 2001). Finally, a main input and an output buffer station, where the RGVs load or unload pallets, are present.

The benefits of AS/RSs include low labour cost, low inventory cost, enhanced space exploitation, improved material tracking and high system throughput. Nevertheless, advantageous operation of an AS/RS clearly depends on the control policies implemented. In the control architecture of an AS/RS three hierarchical levels may be identified (Linn and Wysk 1990): the strategic level, containing control policies for long term expected system performance, the tactical level, collecting control rules for short term performance, and the operational level, dealing with real time behaviour. This paper focuses on the operational control problems. In particular, the proposed management strategy is organized in two subsequent layers. The first and higher one is the scheduler level, taking decisions on selecting a suitable batching policy, proper storage/retrieval policies, a suitable position of the cranes when idle and RGV management policy. The second level is the resource controller, that is in charge of taking decisions on resource allocations in order to avoid conflicts and deadlocks. Numerous studies in related literature deal with typical operational problems, such as defining proper storage and retrieval sequencing policies, in order to maximize the system throughput (Lin and Wang, 1995, Van den Berg and Gademann 2000). However, authors do not examine the real time controller issues, including the RGVS and cranes management. In this context investigating on the AS/RS modelling appears an important and crucial issue (Lin and Wang 1995). The model should feature the following properties: it has to be modular to be suitable for easily modifying the system layout, it has to describe in detail the different material handling subsystems to evaluate the performance of the control algorithms, it has to guarantee the implementation of different management policies.

This paper suggests a unified modelling framework for the heterogeneous AS/RS transport system, comprising both the RGVs and cranes subsystems. To this aim, we employ Coloured Timed Petri Nets (CTPNs), representing a concise and computationally efficient tool for modelling the dynamic behaviour of the system, particularly suitable for industrial applications and control strategies implementation (Feldmann and Colombo 1998, Nandula and Dutta 2000, Wu and Zhou 2001). Moreover, the proposed CTPNs model is modular and resource oriented, i.e., places are resources, tokens are jobs, vehicles and cranes, while colours represent the assigned picking and storing tasks. In addition, transitions model controllable events involving resource acquisition and release. This modelling choice simplifies the realization of the controller, i.e., the management of the real time acquiring and release of conveyor and transport resources. Indeed, advantages of the proposed modelling framework are modularity and compactness in describing an AS/RS serviced by RGV. Additionally, adopting the CTPN model is particularly suitable for simulation verification and validation (Dotoli and Fanti 2004, Wang and Wu 1998). Indeed a discrete event simulation of a simple AS/RS is implemented in the Matlab-Stateflow environment, reflecting the modularity of the introduced model. Simulation results show that the CTPN model is suitable for testing different control and management strategies and allows us to compare different performance indices.

The paper is organized as follows. Section 2 describes the AS/RS under study and section 3 outlines the operational level control structure. Moreover, in section 4, following a brief overview of CTPNs, a modular model of the AS/RS is outlined and in section 5 a case study description and the simulation results are reported. Finally, section 6 draws the conclusion.

2. System Description

We consider a large scale AS/RS that is composed of different sub-systems (Lee et al. 1996): storage conveyors, retrieval conveyors, bi-directional conveyors, narrow-aisle stacker cranes, a Rail-Guided Vehicle System (RGVS), input storage stations and output retrieval stations (see Figure 1).

We call $J=\{j_k: k=1,\ldots,N_J\}$ the set of all the possible jobs to be stored or retrieved in the system and $V=\{v_h: h=1,\ldots,N_V\}$ and $G=\{g_i: i=1,\ldots,N_G\}$ the sets of RGVs and stacker cranes available in the system, respectively. The RGVS rail is divided into disjoint zones, each representing a location adjacent to a storage or a retrieval conveyor or to the input or output station, where vehicles can go through or stop (see Figure 1). Each zone of the RGVS is a resource that vehicles can acquire and is denoted by r_i for $i=1,\ldots,N_Z$, where N_Z is the number of zones in the RGVS. In addition, we generically call resources the storage and retrieval stations as well as the unidirectional and the bi-directional conveyors. In the following we indicate with r_i for $i=N_Z+1,\ldots,N_Z+N_R$ such material handling resources, that are N_R in number. Moreover, all the aisle positions, where each crane can move, are resources that the corresponding S/R machine can acquire. These are denoted by r_i for $i=N_Z+N_R+1,\ldots, N_Z+N_R+N_A$, where N_A is the overall number of crane positions in the aisles, including the vertical, horizontal and sideways shifts, as well as the home positions. Finally, let N_L be the total number of storage locations available in the racks at the sides of the aisles. We indicate with r_i for $i=N_Z+N_R+N_A+1,\ldots,N_Z+N_R+N_A+N_L$ these resources that parts can acquire and with r_0 a fictitious resource modelling the output of the system. Summing up, we call resources both the physical zones of the system and the actual AS/RS material handling resources. Hence, the set $R=\{r_i\ i=0,\ldots,\ N_Z+N_R+N_A+N_L\}$ denotes the resource set of the system. Since each RGVS zone, transport resource or rack location can respectively accommodate only one vehicle, crane or part at a time, each resource $r_i \in R$ with $i \neq 0$ has unit capacity, while r_0 is always available and exhibits infinite capacity. In the sequel when a job $j_k \in J$ (a vehicle $v_h \in V$ or a crane $g_i \in G$) is unambiguously identifiable, subscripts are omitted and the part (the RGV or crane) is referred to as j (v or g, respectively). Moreover, **r**(j), **r**(v) and **r**(g) denote the retrieval operation respectively assigned to the job, the vehicle and the S/R machine by the scheduler. Similarly, **s**(j), **s**(v) and **s**(g) denote the storage operation respectively assigned to job j, vehicle v and crane g. Finally, **rr**(j), **rr**(v) and **rr**(g) (**rs**(j), **rs**(v) and **rs**(g)) denote the corresponding residual sequences that have to be performed to complete the preset retrieval (storage) operation starting from a system configuration.

Example 1. We consider the system in Figure 1, that shows the layout of a multi-product AS/RS serviced by several RVGs. The RVGS consists of 6 zones (denoted by $r_1,\ldots,r_6$), each with unit capacity, and comprises $N_V=2$ vehicles. The AS/RS includes a storage (r_7) and a retrieval (r_8) station, a unidirectional storage conveyor (r_{10}), a unidirectional retrieval conveyor (r_{11}), two bi-directional conveyors (r_9 and r_{12}), and three narrow aisles ($N_G=3$). Each aisle includes a rail with 5 positions, along which the corresponding stacker crane moves, in addition to a home position (r_{13}, r_{39}, r_{65} respectively) where the idle S/R machine waits for the next task. Moreover, each crane can move sideways and vertically to serve the aisle racks, for a total of 26 positions associated to each corridor. Every aisle comprises two racks with 10 storage locations partitioned in a lower and an upper level, for a total of 6 racks and 60 locations (see Figure 1). Hence, in the AS/RS $N_Z=6$, $N_R=6$, $N_A=78$, $N_L=60$.

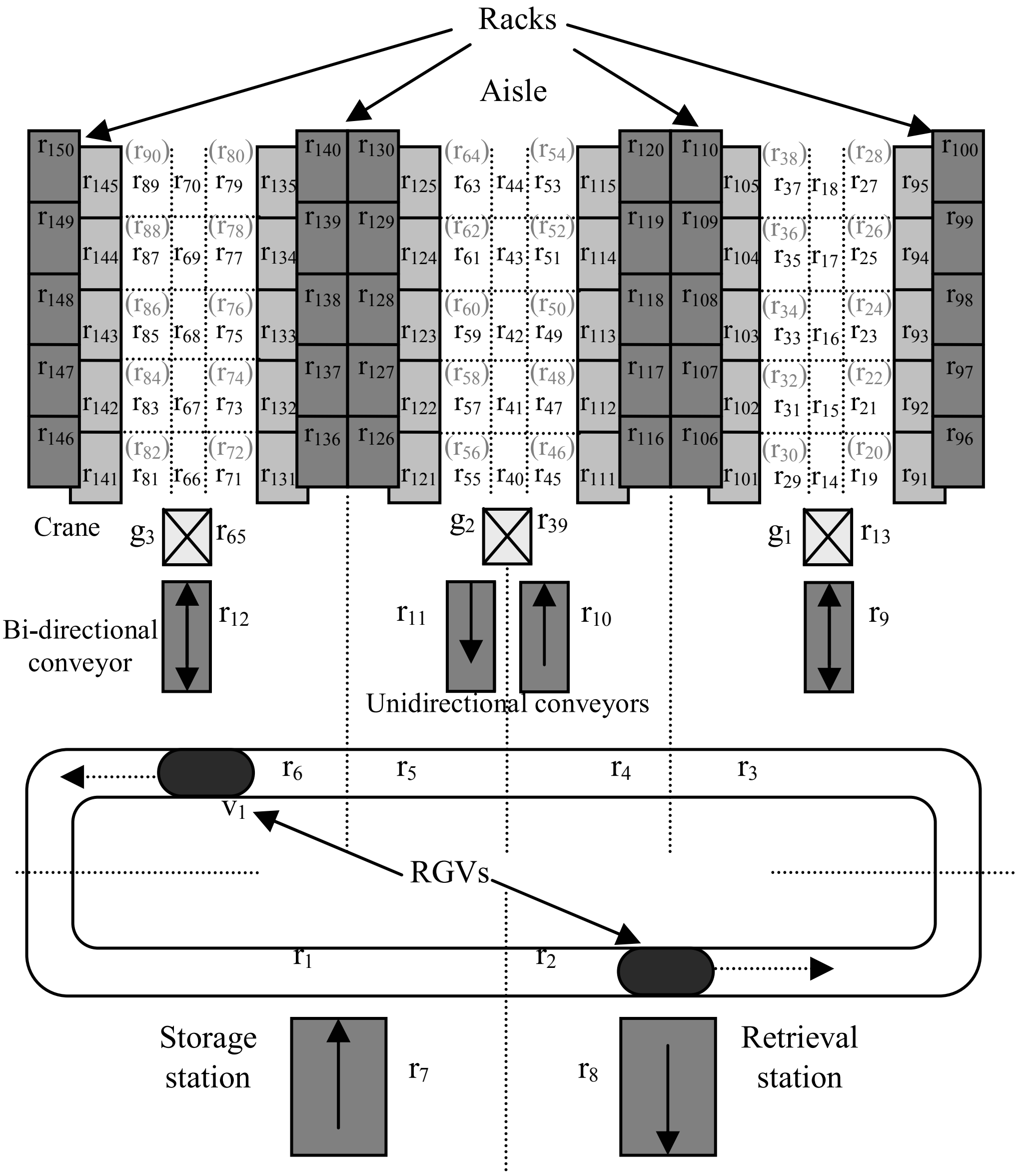


Figure. 1 System layout for the system in Example 1.

3. The Controller Structure

This section describes the hierarchical control structure to take decisions at the system operational level shown in Figure 2. Specifically, the control scheme is constituted of two layers. The first and higher layer (*scheduler*) selects a suitable batching policy, defines proper storage and retrieval sequencing policies and selects the time instant when to assign a new operation.

The second control layer is represented by the *resource controller* that validates the proposed operation and enables or inhibits the acquisition of resources.

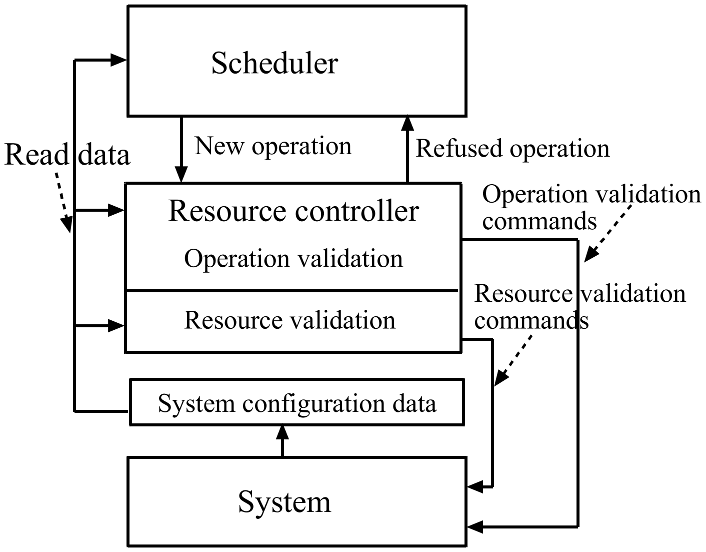


Figure 2. The control architecture.

The main scheduler control activities are described in the following (Linn and Wysk 1990, Eben-Chaime and Pliskun 1996, Van den Berg and Gademann 2000).

i) *Storage Location Assignment*: a control policy imposes constraints on the selection of open locations for incoming parts. Well-known policies in the related literature are the following: the *random assignment policy*, that allows a product to be stored anywhere in the rack, the *dedicated storage policy*, that assigns specific locations in the racks to each product, *the class-based storage policy*, that partitions products among a number of classes and reserves a region within the racks to each class.

ii) *Queue Selection:* a control policy determines which queue (storage or retrieval) to serve next. Widespread policies are as follows: the *first come first serve (FCFS) policy,* that assigns priority to the queue with the oldest request, the *shortest process time (SPT) rule,* selecting the queue with the request requiring the shortest completion time, the *interleave rule,* alternating the queues to be served, the *storage priority rule,* serving storage operations first, the *retrieval priority policy,* completing retrieval requests first.

iii) *Storage/Retrieval Sequence Selection*: different rules can be used to select the next storage or retrieval request upon completion of the current task. The most common rules are: the *first in first out (FIFO) rule,* that assigns priority to the oldest request, the *last in first out (LIFO) rule,* allocating precedence to the latest job, the *SPT rule,* picking first the item with shortest completion time, the *FIFO batch SPT rule*, selecting the job with shortest process time in the first *n* items which have been in the system for the longest time.

iv) *Stacker Crane Mode (Dwell Point Selection)*: a control strategy determines the generic crane command cycle. AS/RSs are typically unit-load retrieval systems, i.e., cranes have unit capacity. Accordingly, cranes either perform one stop (storage or retrieval) or two stops (storage followed by a retrieval) in a single trip along the corresponding aisle. Hence, crane trips are usually referred to as a *single command cycle* and a *dual command cycle*, respectively (Van den Berg and Gademann 2000). In the following sections we assume that cranes are operated in single command mode.

The resource controller is in charge of performing the following two activities.

Resource validation: when an RGV has to move to the next position, or a job has to acquire a resource, the controller validates movements to prevent collisions.

Storage/retrieval validation: when the scheduler proposes a new storage/retrieval operation, the controller has to validate the demand so that no deadlock situation occurs.

3.1 The Storage Operation

When a part j arrives at one of the storage stations, it is placed on a pallet. The scheduler assigns a storage task to the job j, i.e., a route starting from the storage station, going through a certain number of zones up to the final one where the corresponding storage conveyor is located. The task terminates with crane selection and indication of the destination position in the corresponding rack. Moreover, as soon as the job arrives to the storage station, the controller books an idle RGV where the part is loaded. When the RGV carrying the job arrives to the storage conveyor, the part is mechanically loaded from the RGV onto the storage conveyor, and it is picked up by the narrow aisle crane that has been requested by the supervisor to transport the part to the assigned rack location. To clearly describe the formal definition of a storage task, we refer to Example 1 and Figure 1. Therefore, a storage task assigned to a part j_1 can be of the following form: $\mathbf{s}(j_1)=(r_7,r_1,r_2,r_3,r_9,r_{13},r_{14},r_{19},r_{91})$, where r_7 represents the storage station, (r_1,r_2,r_3) is the sequence of zones that the RGV is to go through, starting from zone r_1 before the storage station r_7, up to zone r_3 located in front of the storage conveyor r_9. Moreover, r_{13} is the home position of the assigned aisle crane, (r_{14}, r_{19}, r_{91}) is the sequence of positions in the aisle to reach the assigned rack, where the part will be stored. In addition, the controller assigns to the booked RGV, say v_1 that is idle in zone r_6, the following path: $\mathbf{s}(v_1)=(r_6,r_1,r_7,r_1)$. More precisely, the path of the RGV associated with part j_1 is composed by the sequence of zones from r_6 to r_1, the latter being the zone before the storage station r_7 where the part is loaded. After loading the part, the full vehicle is assigned an additional travel according to the part task: $\mathbf{s}(v_1)=\mathbf{s}(j_1)=(r_1,r_2,r_3,r_9,r_{13},r_{14},r_{19},r_{91})$. In other words, the new path starts from zone r_1 itself and terminates in zone r_{91}. Analogously, the scheduler assigns the following path to the booked crane g_1: $\mathbf{s}(g_1)=(r_{13},r_9,r_{13})$, i.e., the path of the crane is the succession of zones to load the job. After loading the job, the full crane acquires the path of the job: $\mathbf{s}(g_1)=(r_{13},r_9,r_{13},r_{14},r_{19},r_{91},r_{19},r_{14},r_{13})$, i.e., its path is given by the succession of zones to reach the rack destination location (r_{14},r_{19},r_{91}) and, finally, by the succession of zones to reach the home position (r_{19},r_{14},r_{13}).

3.2 The Retrieval Operation

If a job is to be retrieved, the appropriate narrow aisle crane is activated by the supervisor to travel to the rack location, load the part and transport it to the retrieval conveyor. An RGV is then requested to transport the part to a retrieval station. Again, with reference to Example 1 and Figure 1, a retrieval task associated with j_2 in r_{122} can be expressed by the following sequence of resources: $\mathbf{r}(j_2)=(r_{122},r_{57},r_{41},r_{40},r_{39},r_{11},r_5,r_6,r_1,r_2,r_8,r_0)$, where r_{39} is the home position of the crane loading the part, r_{11} is the corresponding retrieval conveyor, (r_5,r_6,r_1,r_2) is the sequence of zones that the RGV carrying the part will visit, starting from zone r_5 before the retrieval conveyor r_{11}, up to zone r_2 in front of the retrieval station r_8, with which the part sequence terminates. As soon as the retrieval task is assigned, the crane in r_{39} (say g_2) begins its path $\mathbf{r}(g_2)=(r_{39},r_{40},r_{41},r_{57},r_{122},r_{59})$. When the crane is in r_{122}, the job is loaded and starts its travel $\mathbf{r}(j_2)$. Simultaneously, an RGV is booked and assigned to the part with the preset retrieval task. More precisely, the controller assigns to the booked RGV, e.g. v_2 that is idle and waiting in zone r_2, the path $\mathbf{r}(v_2)=(r_2,r_3,r_4,r_5,r_{11},r_5)$, i.e., the path of the RGV associated with part j_2 is composed by the succession of zones starting from zone r_2 where the RGV is initially located, up to zone r_5 before the retrieval conveyor, where the part is loaded. Next, the full vehicle starts a new travel to the retrieval station r_8, with the following assigned path: $\mathbf{r}(v_2)=\mathbf{r}(j_2)=(r_5,r_6,r_1,r_2,r_8,r_0)$. After loading the job on r_8, the vehicle remains idle in r_2, i.e., $\mathbf{r}(v_2)=r_2$.

4. The Modular Model of the System

This section describes the CTPN modelling an AS/RS comprising both an RGVS and several crane subsystems. More precisely, we propose a modular and resource oriented CTPNs model. In particular, places are resources, tokens are jobs, vehicles and cranes, while colours represent the assigned picking and storing tasks. In addition, transitions model controllable events involving resource acquiring and release. In the following the AS/RS under study is modularly modelled, i.e., we model separately the RGVS, the storage and retrieval stations and conveyors, the cranes.

4.1 Overview of Coloured Timed Petri Nets

A coloured timed Petri net is an 8-tuple CTPN=(P, T, Co, Inh, **Post**, **Pre,** Ω, M_0) where P is a set of places, T is a set of transitions, Co is a colour function defined from $P\cup T$ to a set of finite and not empty sets of colours (Jensen 1992). Co maps each place $p\in P$ to a set of possible token colours $Co(p)$ and each transition $t\in T$ to a set of possible occurrence colours $Co(t)$. Inh is a weight function for an inhibitor arc which connects a transition to a place. The inhibitor arc between a place $p\in P$ and a transition $t\in T$ (i.e. $Inh(p,t)=1$) implies that transition t can be enabled if p does not contain any token. **Post** and **Pre** are the post-incidence and the pre-incidence $|P|\times|T|$ matrices respectively, so that **Post**(p,t) associates to each set of colours of $Co(t)$ a set of colours of $Co(p)$. **Post**(p,t) (**Pre**(p,t)) is represented by means of an arc from t to p (from p to t) labelled with the function **Post**(p,t) (**Pre**(p,t)). Note that we use symbol $|A|$ to denote the cardinality of a generic set A. The set Ω is defined as follows: $\Omega=\cup_{x\in P\cup T}\{Co(x)\}$. A marking M is a mapping defined over P so that M(p) is a set of elements of Co(p), also with repeated elements (i.e., a multi-set) corresponding to token colours in the place p. M_0 is the initial marking of the net. Just like in ordinary Petri nets, we can define the flow incidence matrix **C**=**Post**-**Pre**. In particular, a transition $t\in T$ is enabled at marking M with respect to a colour $c\in Co(t)$ if and only if for each

$p \in \bullet t$, $M(p) \geq \mathbf{Pre}(p,t)(c)$. Upon firing, the transition leads to a new marking M' that is obtained as follows: $M'(p)=M(p)+\mathbf{Post}(p,t)(c)-\mathbf{Pre}(p,t)(c)$. As regards the definition of multi-sets and operations on multi-sets, the reader can refer to (Jensen 1992).

Now, to investigate the performance of the system, it is convenient to extend the coloured Petri net with the time concept (Jensen 1992). To this aim, we introduce a global clock. Note that the clock values $\tau \in \Re^+$ represent the model continuous time. Moreover, we define on the place set P the function δ: $P \rightarrow \Re^+$ where $\delta(p)$ describes the earliest model time at which the token can be removed by the enabled transition. In addition to token colours, we attach to each token a time stamp *ts*. The time stamp is reset as soon as the token arrives in the place. When the stamp equals or is larger than $\delta(p)$, the transition enabled by the considered token is ready for execution.

4.2 The CTPN Modelling the System Dynamics

In our model the CTPN=(P, T, Co, Inh, **Post**, **Pre**, Ω, M_0) describes the complete AS/RS system, comprising both the RGV and crane subsystems. A place $r_i \in P$ denotes a resource $r_i \in R$ and there is a one to one relationship between resources and places. A transition $t \in T$ models the flow of jobs, vehicles and cranes into and out of the system or between consecutive resources. In particular, the transition set T can be partitioned in two subsets: T_L collects transitions t_{0i}, modelling a job entering the system through resource $r_i \in P$ and transitions t_{i0} modelling a job leaving the system from resource r_i; T_F is the set of transitions t_{im} modelling the flow of parts vehicles and cranes from r_i to r_m. More precisely, each $t_{0i} \in T_L$ is a source transition ($\bullet t_{0i}=\varnothing$) and $t_{0i}\bullet=\{r_i\}$, where r_i is a storage station. In addition, the sink transitions $t_{i0} \in T_L$ are such that $t_{i0}\bullet=\varnothing$ and $\bullet t_{i0}=\{r_i\}$, where r_i is a retrieval station. Moreover, transitions $t_{im} \in T_F$ if r_i and r_m are two consecutive resources in the system and they are such that $t_{im} \in \bullet r_m$ and $t_{im} \in r_i\bullet$. To admit just one vehicle or crane in each zone and one job in each resource, there is an inhibitor arc between each place $r_m \in P$ and transition $t_{im} \in T_F$, i.e. $Inh(r_m,t_{im})=1$.

A coloured token in a place can represent a piece, an RGV or a crane, respectively idle or carrying. The colour of each token can be one of the following items:

i) <**rr**(j)> (<**rs**(j)>), where **rr**(j) (**rs**(j)) is the residual sequence of resources in a retrieval (storage) operation that the job j has to accomplish;

ii) <**rr**(v)> (<**rs**(v)>), where **rr**(v) (**rs**(v)) is the residual sequence of zones that the booked vehicle $v \in V$ has to visit for a retrieval (storage) operation,

iii) <**rr**(g)> (<**rs**(g)>), where **rr**(g) (**rs**(g)) is the residual sequence of zones that the booked crane $g \in G$ has to visit for a retrieval (storage) operation.

iv) <r_m>, where $r_m \in P$ is the resource occupied by an idle crane or truck.

The colour domain of a place $r_i \in P$ is:

Co(r_i)={<**rr**> or <**rs**>, where **rr** (or **rs**) is a sequence of resources and r_i is the first resource of **rr** (or **rs**).

Moreover, *Co* associates with each transition $t_{im} \in T_F$ a set of possible occurrence colours:

Co(t_{im})={<**rr**> or <**rs**> such that **rr** or **rs** is a sequence of type (r_i,r_m…) or (r_m,r_i,r_m…)}.

Here, the CTPN dynamics is described by the incidence matrix **C** that contains a row for each place $r_i \in P$ and a column for each transition $t \in T$. Each element **C**(r_i,t) is a function that assigns an element of *Co*(t) with $t \in T$ to an element of Co(r_i) with $r_i \in P$. The incidence matrix is computed as **C**(r_i,t)=**Post**(r_i,t)-**Pre**(r_i,t), where the pre- and the post-incidence matrices **Pre** and **Post** are respectively defined as follows:

D1) for each $(r_i,t_{im}) \in F$, **Pre**(r_i,t_{im})=*Id*, where *Id* stands for "the function makes no transformation in the elements", otherwise **Pre**(r_i,t_{im})=0. This definition means that each token leaving a resource $r_i \in P$ is not modified;

D2) for each $(t_{im},r_m) \in F$, **Post**(r_m,t_{im})=*UP*, where *UP* is "the function that updates the colour <**rr**> with the colour <**rr**'>", otherwise **Post**(r_m,t_{im})=0. More precisely, **rr**' is the residual sequence of resources obtained from **rr** by cutting the first element r_i. When a token leaves r_i and reaches r_m, its colour, i.e., its residual path, is updated;

D3) for each $(t_{im},r_i) \in F$, **Post**(r_i,t_{im})=<r_i>. When a token leaves r_i and reaches r_i again, its colour becomes <r_i>.

4.3 The CTPN Modelling the RGVS Dynamics

In this section we define the CTPN describing the RGVS behaviour. A place $r_i \in P$ for $i=1,\ldots,N_Z$ denotes the zone $r_i \in R$ and a token in r_i represents a vehicle $v \in V$ *idle*, *booked* or *carrying* in r_i. As an example, Figure 3 shows the CTPN describing the RGVS of Figure 1. We remark that the inhibitor arcs model the unit capacity of the N_Z zones in the RGVS. Each vehicle $v \in V$ is modelled by a coloured token and its token colour can be one of the following three types:

i) if vehicle $v \in V$ is booked, then $\mathbf{rr}(v)=(r_{i1},\dots r_{iL})$ is the sequence of zones starting from the zone occupied by the RGV (r_{i1}) up to the resource r_{iL} from which the vehicle will load the piece;

ii) if vehicle $v \in V$ is idle, then $\mathbf{rr}(v)=(r_i)$, where r_i is the zone where the RGV is waiting for the next task;

iii) if vehicle $v \in V$ is carrying job j, then $\mathbf{rr}(v)=\mathbf{rr}(j)$, i.e., $\mathbf{rr}(v)$ is equal to the sequence of zones starting from the zone occupied by v, up to the destination resource of the job.

Hence, the state of the RVGS is represented by the CTPN marking and the following mutually exclusive situations hold for each $r_i \in P$ with $i=1,\dots,N_Z$:

$M(r_i)=<\mathbf{rr}(v)>$ ($=<\mathbf{rs}(v)>$) i.e., r_i is occupied by $v \in V$ that is booked by a job j for a retrieval (storage) operation;

$M(r_i)=<\mathbf{rr}(j)>$ ($=<\mathbf{rs}(j)>$) i.e., r_i is occupied by $v \in V$ that is carrying job j for a retrieval (storage) operation;

$M(r_i)=<r_i>$ i.e., r_i is occupied by an idle $v \in V$;

$M(r_i)=<0>$ i.e., no vehicle is in r_i.

Moreover, considering that N_v vehicles are available in the system, the initial marking M_0, previous to any path assignment from the scheduler to the RGVs, is defined as follows: if $\mathbf{rr}(v)=r_i$ for some $v \in V$, then $M_0(r_i)=<r_i>$, else $M_0(r_i)=<0>$.

A transition t_{im} is enabled with respect to colour **rr** if the following two conditions are simultaneously verified for $r_i, r_m \in P$ with $i,m=1,\dots,N_Z$ and $i \neq m$:

C1) $M(r_m)=<0>$.

C2) $M(r_i) \geq \mathbf{Pre}(r_i,t_{im})(<\mathbf{rr}>)$, with $M(r_i)=<\mathbf{rr}>$, $\mathbf{rr}=(r_i,r_m,\dots)$.

Condition C1) follows from the inhibitor arc related to the transition and condition C2) represents the enabling condition of the CTPN at marking M. Let us suppose that a token of colour $<\mathbf{rr}>$ arrived in place r_i at time τ; after $\delta(r_i)$ time units the time stamp of $<\mathbf{rr}>$ is $ts(<\mathbf{rr}>)=\delta(r_i)$, so that transition t_{im} is ready and enabled at marking M and at time $\tau+\delta(r_i)$. Now, if $t_{im} \in T$ fires, then the new marking M', such that $M[t_{im}(\mathbf{rr})>M'$, is the following:

$$M'(r_i)=M(r_i)-\mathbf{Pre}(r_i,t_{im})(<\mathbf{rr}>)=<0> \qquad (1)$$

$$M'(r_m)=M(r_m)+\mathbf{Post}(r_m,t_{im})(<\mathbf{rr}>)=<\mathbf{rr}'> \qquad (2)$$

where $\mathbf{rr}'=(r_m, \ldots)$ is obtained by applying the function *UP* to $\mathbf{rr}=(r_i\ r_m, \ldots)$. Finally, the stamp $ts(<\mathbf{rr}'>)$ is set to zero.

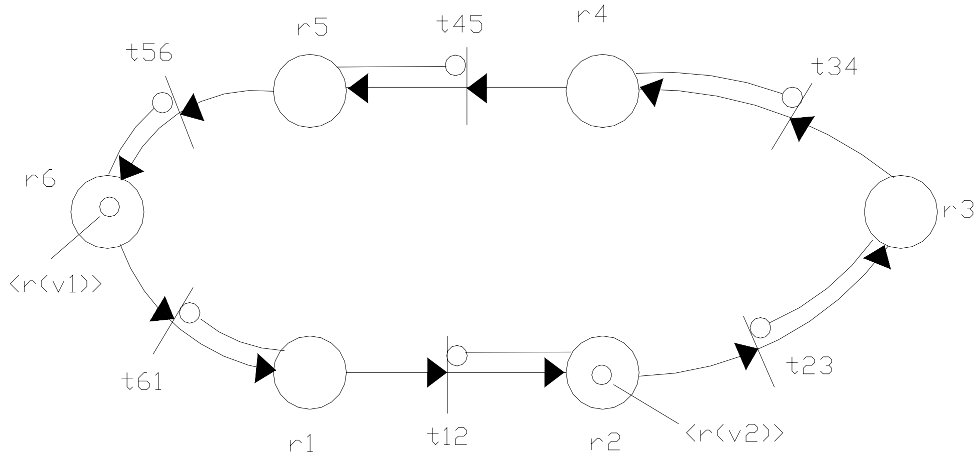


Figure 3. The CTPN modelling the RGV subsystem.

4.4 The CTPN Modelling the Dynamics of Storage Stations

Figure 4 shows the CTPN modelling a storage station, where resource r_i is the storage station and $t_{0i} \in T_L$ models a piece entering the system. Now, we name r_m the rail zone before r_i and describe the loading of the job onto a vehicle by means of transition $t_{im} \in T_F$. In particular, transition $t_{im} \in T_F$ is such that $t_{im}\bullet=\{r_m\}$ and $\bullet t_{im}=\{r_i, r_m\}$. If $M(r_i)=<0>$, transition t_{0i} can fire and a job $j \in J$ enters the system. Once the scheduler has assigned to j the task $<\mathbf{s}(j)>$ and transition t_{0i} has fired, the new marking is $M(r_i)=<\mathbf{s}(j)>=<r_i, r_m,\ldots>$. In such a case, t_{im} is enabled if there is a token representing $j \in J$ in r_i with colour $<\mathbf{s}(j)>=<r_i, r_m,\ldots>$ and if there is a token representing the booked vehicle $v \in V$ in r_m with colour $<r_m,r_i,r_m>$.

Formally, if we define $\mathbf{Pre}(r_m,t_{im})=Id$, then the t_{im} enabling conditions are:

C3) $M(r_i) \geq \mathbf{Pre}(r_i,t_{im})(<\mathbf{s}>)$, i.e. $M(r_i)=<\mathbf{s}(j)>=<r_i, r_m, \ldots>$.

C4) $M(r_m) \geq \mathbf{Pre}(r_m,t_{im})(<r_m,r_i,r_m>)$, i.e., $M(r_m)=<r_m, r_i, r_m>$.

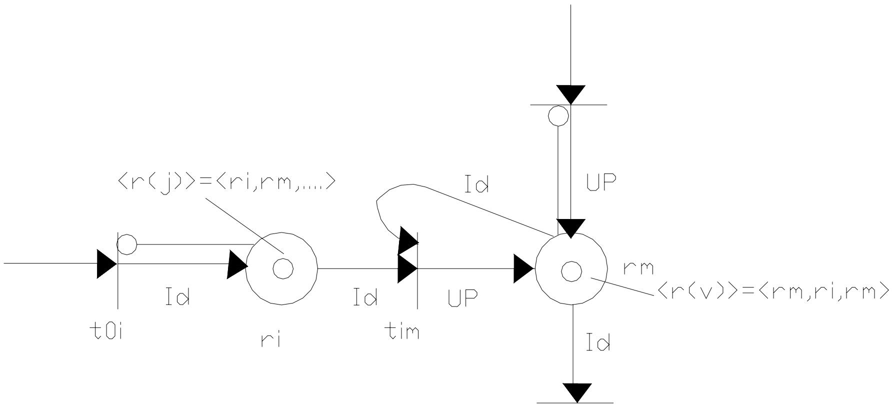


Figure 4. The CTPN modelling the storage stations.

Now, if $t_{im} \in T$ is ready and fires, then the new marking M' is the following:

$$M'(r_i)= M(r_i)-\mathbf{Pre}(r_i,t_{im})(<\mathbf{s}(j)>)=<0> \tag{3}$$

$$M'(r_m)=M(r_m)+\mathbf{Post}(r_m,t_{im})(<\mathbf{s}(j)>)-\mathbf{Pre}(r_m,t_{im})(<r_m,r_i,r_m>)=<\mathbf{rs}(j)>=<r_m,\ldots> \tag{4}$$

where $\mathbf{rs}(j)=(r_m, \ldots)$ is obtained by applying the function *UP* to $\mathbf{s}(j)=(r_i\ r_m, \ldots)$. Finally, the time stamp $ts(<\mathbf{rs}(j)>)$ is reset.

4.5 The CTPN Modelling the Dynamics of Retrieval Stations

The dynamics of retrieval stations is modelled by the CTPN in Figure 5. More precisely, let us assume that r_m is a retrieval station and that $t_{m0} \in T_L$ models a job leaving the system. Moreover, r_i is the zone in front of r_m where a vehicle stops and transition $t_{i,m} \in T_F$ models the unloading of the job from the RGV to the retrieval station. When $t_{im} \in T_F$ fires, the job token occupies r_m and a vehicle token returns to r_i. To model this situation, transition $t_{im} \in T_F$ is such that $\bullet t_{im}=\{r_i\}$ and $t_{i,m}\bullet=\{r_m, r_i\}$. So, if in zone r_i there is a vehicle carrying job j, it holds $M(r_i)=<\mathbf{rr}(j)>=<r_i,r_m,r_0>$. Since there is an inhibitor arc between r_m and t_{im}, transition t_{im} is enabled if $M(r_m)=<0>$ and

$M(r_i)=<\mathbf{rr}(j)>=<r_i,r_m,r_0>$. Since we defined $\mathbf{Post}(r_i,t_{im})(<\mathbf{rr}>)=<r_i>$ (see definition D3 in the previous section), if $t_{im}\in T_F$ fires, the new marking M' is the following:

$$M'(r_i)=M(r_i)+\mathbf{Post}(r_i,t_{i,m})(<\mathbf{rr}(j)>)-\mathbf{Pre}(r_i,t_{i,m})(<\mathbf{rr}(j)>)=<r_i>. \tag{5}$$

$$M'(r_m)=M(r_m)+\mathbf{Post}(r_m,t_{im})(<\mathbf{rr}(j)>)=<\mathbf{rr}'(j)>=<r_m,r_0>. \tag{6}$$

Note that at marking M' the token in r_m denotes the job j that enables t_{m0} to leave the system (its colour is $<r_m, r_0>$) and the token in r_i represents now the idle RGV (with colour $<r_i>$).

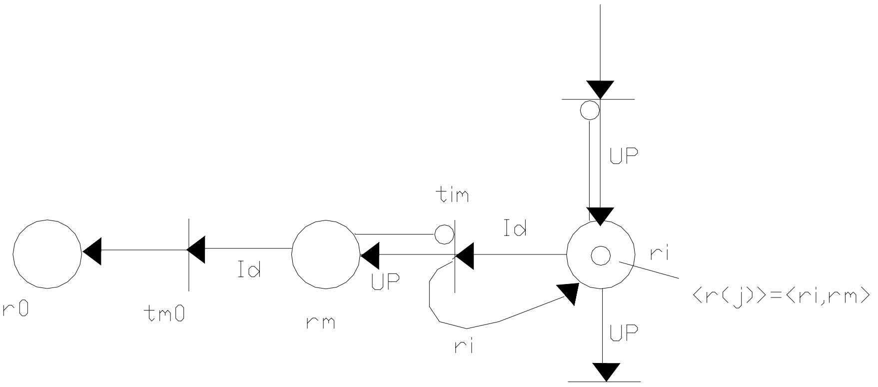


Figure 5. The CTPN modelling retrieval stations.

4.6 The CTPN Modelling the Dynamics of Retrieval and Storage Conveyors

Following the previous procedure, it is possible to build the CTPNs modelling the storage and retrieval conveyor dynamics respectively, that are correspondingly very similar to the dynamics of the retrieval and storage stations. In such a case, the only difference is that there is a place representing the crane home position as an output to the storage conveyor and an input to the retrieval conveyor.

4.7 The CTPN Modelling the Crane Dynamics

In the CTPN modelling the crane subsystem places are associated with the crane positions and cranes are modelled as coloured tokens. When a crane is booked its token colour is the path necessary to reach the rack location where the piece is to be loaded. If a crane is busy and it carries a job, the colour of the crane token is the residual path assigned to the job.

Hence, the state of each crane is represented by the CTPN marking and the following mutually exclusive situations hold for each $r_i \in P$ with $i=N_Z+N_R+1,\ldots,N_Z+N_R+N_A$.

$M(r_i)=<\mathbf{rr}(g)>$ ($<\mathbf{rs}(g)>$): r_i is occupied by $g \in G$ that is booked by a job j for a retrieval (storage) operation;

$M(r_i)=<\mathbf{rr}(j)>$ ($<\mathbf{rs}(j)>$): r_i is occupied by $g \in G$ that is carrying a job j for a retrieval (storage) operation;

$M(r_i)=<r_i>$: r_i is occupied by an idle $g \in G$;

$M(r_i)=<0>$: no crane is in r_i.

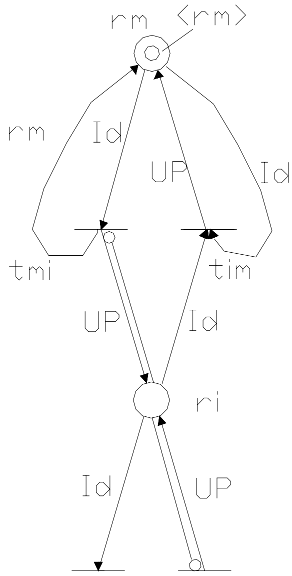


Figure 6. The CTPN modelling the connection between a bidirectional conveyor and a crane.

Moreover, each crane can load or unload a piece from unidirectional or bidirectional conveyors. Hence the CTPN modelling the interactions between cranes and conveyors is modelled as storage and retrieval stations. More precisely, we consider the CTPN shown in Figure 6 representing a bidirectional conveyor where resource r_i is the conveyor, r_m is the crane position, transition $t_{im} \in T_F$ describes the loading of the job onto r_m and transition $t_{mi} \in T_F$ describes the unloading of the job from r_m to the conveyor r_i. In particular, transition $t_{im} \in T_F$ is such that $t_{im}\bullet=\{r_m\}$ and $\bullet t_{im}=\{r_i,r_m\}$. Let us suppose that there is a token representing $j \in J$ in r_i with colour

<**rs**(j)>=<r_i,r_m,…,r_L> where r_L denotes the rack position where the job must be stored. If there is a token representing the booked crane $g \in G$ in r_m with colour <r_m,r_i,r_m>, then t_{im} is enabled. Formally, the t_{im} enabling conditions are:

C5) $M(r_i) \geq \mathbf{Pre}(r_i,t_{im})(<\mathbf{rs}>)$, i.e. $M(r_i)=<\mathbf{rs}(j)>=<r_i,r_m, \ldots,r_L>$.

C6) $M(r_m) \geq \mathbf{Pre}(r_m,t_{im})(<\mathbf{rs}>)$, i.e. $M(r_m)=<\mathbf{rs}(g)>$, with $\mathbf{rs}(g)=(r_m,r_i,r_m)$.

Now, if $t_{i,m} \in T$ is ready and fires, then the new marking M' is the following:

$$M'(r_i)= M(r_i)-\mathbf{Pre}(r_i,t_{im})(<\mathbf{rs}(j)>)=<0> \tag{7}$$

$$M'(r_m)=M(r_m)+\mathbf{Post}(r_m,t_{im})(<\mathbf{rs}(j)>)-\mathbf{Pre}(r_m,t_{im})(<\mathbf{rs}(g)>)=<\mathbf{rs}'(j)>=<r_m,\ldots,r_L> \tag{8}$$

On the other hand, transition t_{mi} is enabled if $M(r_i)=<0>$ and $M(r_m)=<\mathbf{rr}(j)>=<r_m,r_i,\ldots, r_0>$. Since we defined $\mathbf{Post}(r_m,t_{mi})=r_m$, if $t_{im} \in T_F$ fires, the new marking M' is the following:

$$M'(r_m)=M(r_m)+\mathbf{Post}(r_m,t_{mi})(<\mathbf{rr}(j)>)-\mathbf{Pre}(r_i,t_{im})(<\mathbf{rr}(j)>)=<r_m> \tag{9}$$

$$M'(r_i)=M(r_i)+\mathbf{Post}(r_i,t_{mi})(<\mathbf{rr}(j)>)=<\mathbf{rr}'(j)>=<r_i,\ldots,r_0> \tag{10}$$

We model the interaction between cranes and unidirectional conveyors analogously.

Example 2. With reference to Figure 1 and Example 1, in Figure 7 the previously modelled subsystems are shown in the merged CTPN where transition labels are neglected. In particular, with reference to Figure 7, places r_{13}, r_{39} and r_{65} are associated with the home positions of the cranes and these are modelled as coloured tokens. Transitions $t_{12,65}$, $t_{10,39}$ and $t_{9,13}$ represent the storage operations and $t_{65,12}$, $t_{39,11}$, $t_{13,9}$ are the retrieval operations. In the system there is a job j_1 to be stored in r_{91} and a job j_2 to be retrieved from r_{122}. So, the vehicle in r_2 is booked by j_2 and the vehicle in r_1 is booked by j_1. The marking of the AS/RS subsystem in Figure 7 is defined as follows: $M(r_7)=<\mathbf{rs}(j_1)>=(r_7,r_1,r_2,r_3,r_9,r_{13},r_{14},r_{19},r_{91})$, $M(r_1)=<\mathbf{rs}(v_1)>=(r_1,r_7,r_1)$, $M(r_{39})=<\mathbf{rr}(g_2)>= =(r_{39},r_{40},r_{41},r_{57},r_{122})$, $M(r_2)=<\mathbf{rr}(v_2)>=(r_2,r_3,r_4,r_5,r_{11},r_5)>$, $M(r_{65})=<r_{65}>$, $M(r_{13})=<r_{13}>$ and $M(r_i)=<0>$ for $r_i \in P$ with $i \neq 1, 2, 7, 13, 39, 65$. Moreover, we focus on one of the narrow aisles, since the other corridors may be modelled accordingly. For instance, consider crane g_1 in the rightmost narrow aisle in Figure 1. The crane services two opposite racks with 10 locations each. The racks are partitioned in two 5-locations rows that are situated in a lower level (resources r_{91} to r_{95} in the right-hand rack and r_{101} to r_{105} in the left-hand one) and an upper level (resources r_{96} to r_{100} and r_{106} to r_{110}). Place r_{13} is the crane home position, places r_{14} to r_{18} model the aisle positions, r_{19} to r_{28} and r_{29} to r_{38} model the horizontal positions at the sides along the racks, partitioned in the two levels. Figure 8 shows the resulting CTPN model of the crane: for sake of simplicity the lower level only is depicted and the 10 lower locations in the racks are represented. Again, the transitions arc labels are omitted.

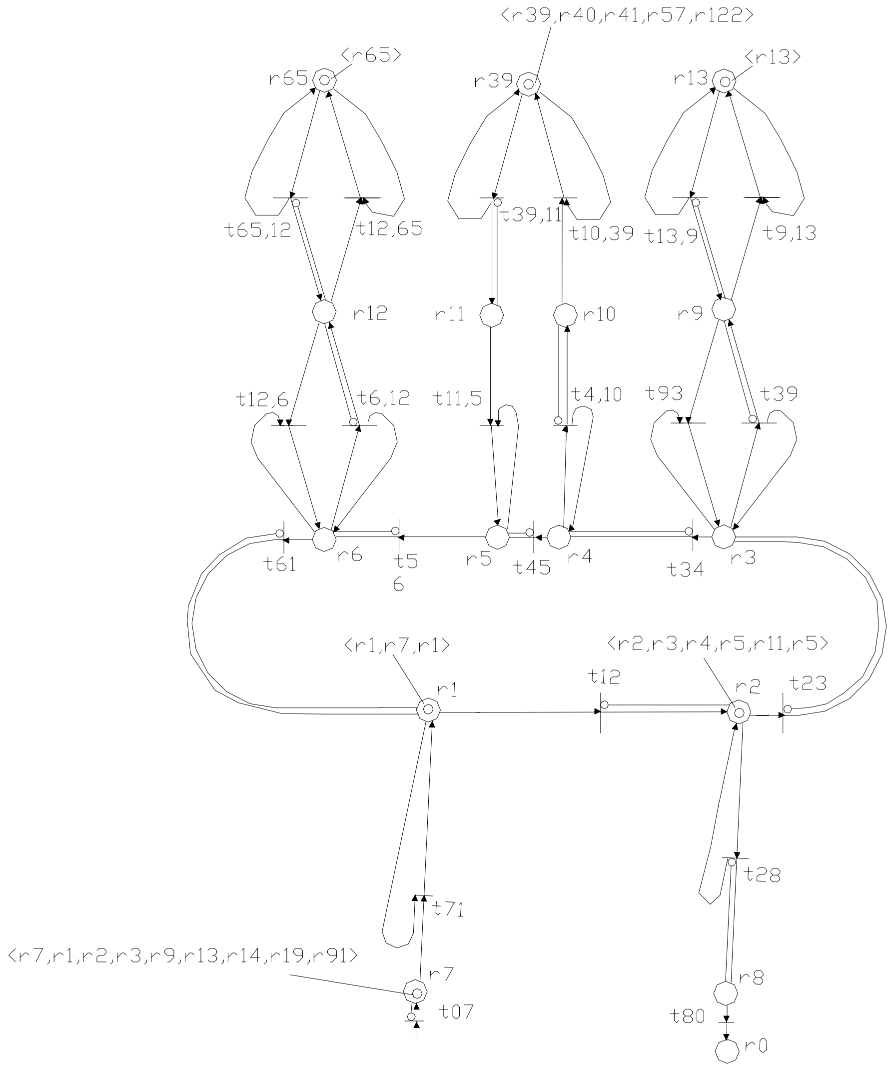


Figure 7. The CTPN at marking M for Examples 1 and 2.

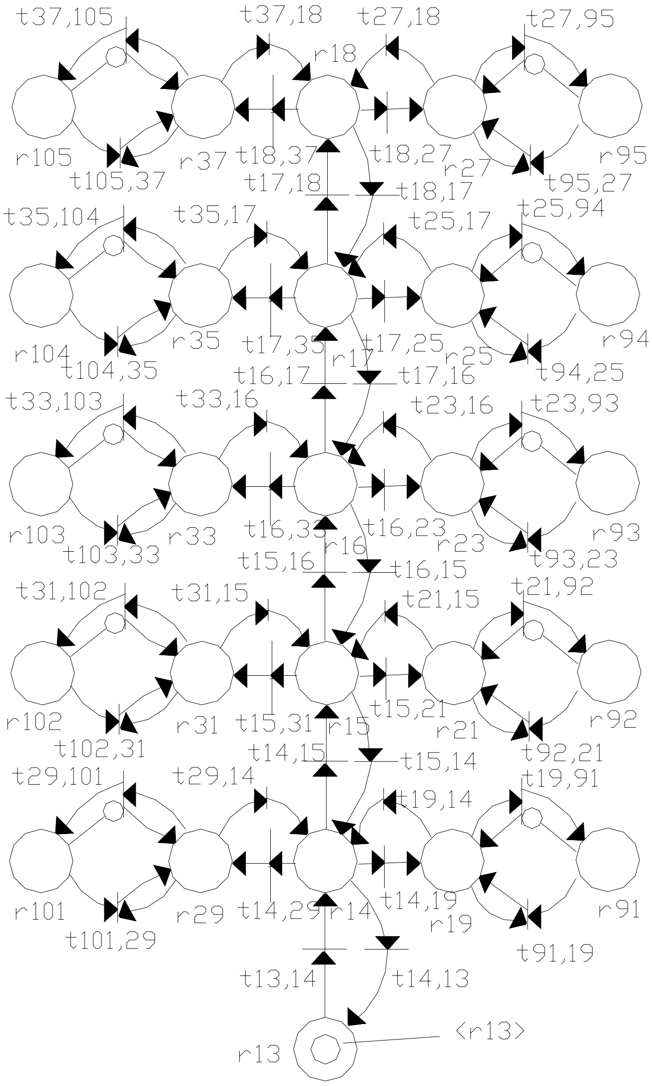


Figure 8. The CTPN modelling the narrow aisle subsystem.

4.8 The System Events

We close this section defining the three types of events that can change the system state, i.e., the marking of the CTPN:

type 1 event: a job j enters the system for a storage or retrieval operation. The event is identified by the pair σ_1=(j,**r**) (σ_1=(j,**s**)) where **r** (**s**) is the retrieval (storage) operation assigned to j;

type 2 event: a job j books a crane g or a vehicle v for a storage (retrieval) operation. The event is identified by the triple σ_2=(g, j, **s**) (σ_2=(g, j, **r**)) or σ_2=(v, j, **s**) (σ_2=(g, j, **r**)), where **r** (**s**) is the operation assigned to g or v for a retrieval (storage) operation;

type 3 event: an idle/booked vehicle v or crane g or a job j moves from a resource to another. The event is identified by the symbol σ_3=v, σ_3=g or σ_3=j, respectively;

type 4 event: a job leaves the system or is stored in a rack. This event is identified by the pair $\sigma_4=(j,r_m)$, where $r_m \in \{r_0, r_i$ with $i=N_Z+N_R+N_L+1,\ldots, N_Z+N_R+N_A+N_L\}$ represents the resource that a job j acquires when a retrieval or a storage operation goes to an end.

The occurrence of a type 1 event σ_1 determines the income of a new token with colour **r**(j) or **s**(j) into the system. Consequently, a type 2 event σ_2 occurs because a new route is assigned of type **r**(g) and **r**(v) (for a retrieval operation) or **s**(g) and **s**(v) (for a storage operation) to the corresponding transportation resource. On the other hand, when a type 3 event σ_3=j, σ_3=v or σ_3=g happens, transition $t_{im} \in T_R$ fires in the CTPN and the marking M is updated accordingly.

5. The Case Study and Simulation Results

This section presents an implementation and validation of the introduced model. To this aim we consider the system shown in Figure 9 describing a simple multi-product AS/RS. In particular, the RVGS consists of six zones ($r_1,\ldots,r_6$) and comprises N_V vehicles. In the sequel, N_V=1 and N_V=2 are considered. The AS/RS includes an input (r_7) and an output (r_8) station, two unidirectional storage conveyors (r_9 and r_{11}), two unidirectional retrieval conveyors (r_{10} and r_{12}) and two narrow aisles (N_G=2). Each aisle includes a rail with four positions, along which the corresponding crane moves, in addition to a home position (r_{13} and r_{26} respectively), where the idle S/R machine waits for the next task. Moreover, each crane can move sideways to serve the aisle racks, for a total of thirteen positions associated to each corridor. Every aisle comprises two racks with four locations. Hence, with reference to the AS/RS in Figure 9, it holds: N_Z=6, N_R=6, N_A=26, N_L=16.

Several simulation experiments are conducted for the considered AS/RS under the following assumptions. A set containing all the 32 possible retrieval and storage operations is created and five sets containing respectively 50, 100, 200, 300 and 500 tasks are generated by randomly picking an operation in the first set. The resulting series of tasks are executed with N_V=1 and N_V=2 by the controlled AS/RS. In particular, the scheduler control activity is designed as follows. As regards the storage location assignment, the simplest and most common rule is considered, i.e., the random assignment policy. Moreover, the FCFS queue selection policy with

FIFO storage/retrieval sequence selection and single command mode is considered. Results are compared to the adoption of the random assignment policy and the interleave rule with FIFO storage/retrieval sequence selection and dual command cycle. In the latter case, the interleave rule is investigated both for one dual cycle and four dual cycles executed in succession on the same aisle and stacker crane. Hence, six experiments are carried out under the following assumptions: FCFS policy, interleave rule with one dual cycle and interleave rule with four dual cycles. We remark that under the interleave rule a storage always precedes a retrieval task.

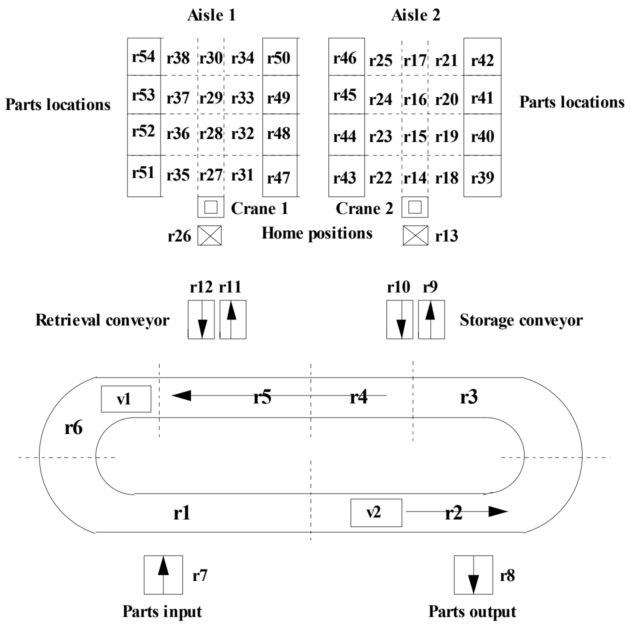


Figure 9: The layout of the simulated system.

Since the structure of the AS/RS is very simple and does not include bidirectional conveyors, deadlocks can not occur if the number of RGV is N_V=1, 2 (Lee *et al.* 1996). Hence, the specification of the resource controller is very simple in this case. More precisely, the operational level controller avoids collisions only and the specification of the deadlock avoidance controller will be the subject of future research.

5.1 Performance Measures

In order to compare the effectiveness of the selected scheduling strategies in assigning the operations to the AS/RS, several system performance measures are considered.

The first index of effectiveness of the management policies taken into account is the system *throughput*, i.e., the number of completed operations in a fixed run time simulation. A supplementary measure of performance for each simulation experiment is the *completion time*, i.e., the run time required to complete all the missions in each set of operations (50, 100, 200, 300, 500 tasks). Both the above indices assess the selected management policy efficiency in having the system accomplish paths.

Additional performance measures in the simulations are the RGVs and cranes utilization, evaluated by determining the average percentage time that such means of transport spend in one of their states.

In particular, trucks activity may be classified as follows.

(i) *Booked travel* – the booked RGV is either empty and travelling toward the loading area (i.e., the input station for a storage or a retrieval conveyor for a retrieval) or waiting in order to avoid collisions.

(ii) *Loaded travel* – the vehicle is carrying a part to accomplish a mission. This state comprises two sub-states: the truck is either transporting a piece toward the assigned unloading area (i.e., the selected storage conveyor for a storage or the output station for a retrieval), or blocked in a zone by the control system in order to avoid collisions.

(iii) *Idle* – the vehicle, after unloading a part at the destination zone, is idle waiting for an assignment.

Similarly, cranes activity can be classified as follows.

(i) *Loaded and booked travel* – the crane is either carrying a piece to the assigned unloading area (i.e., the selected rack location for a storage or the retrieval conveyor for a retrieval) or empty and travelling (i.e., toward the home position after performing a storage under the single command cycle, or toward a rack location to load a piece for a retrieval under the single or dual command cycles).

(ii) *Idle* – the crane is idle in its home position waiting for a task to accomplish.

Accordingly, the *cranes utilization* is evaluated, i.e., the mean percentage of the completion time that the stacker machines are busy in the corresponding aisle.

5.2 Validation of the CTPN Model in Matlab-Stateflow

The modularity of the CTPN model and the simple marking updating make easy the implementation and validation of the model in different programming and simulation languages. Here we use the Matlab-Stateflow environment (The MathWorks 1997), where it is possible to integrate modelling and simulation of Stateflow event-driven systems (e.g., the RGVS and cranes dynamics) with the execution of Matlab computation routines (e.g., developing the resource and operation validation algorithms), while keeping track of time by way of a software clock. The rationale for choosing Matlab-Stateflow as simulation environment, rather than more evolved and specific software, lies in its simplicity and immediacy. In fact, rather than mimicking the operation of an actual AS/RS in its complexity (number of rack locations, RGVs and cranes in the system, etc.), the purpose of the present simulation study is validating the proposed model and testing the efficiency of well-known scheduling policies with reference to several performance measures.

Here the CTPN is represented by a finite states automaton. Figure 10 shows the Matlab-Stateflow code implementing the CTPN modelling the AS/RS of Figure 9. The chart comprises eight subsystems: “In Block”, “RGVS Block”, “Update Marking Block”, “Save Marking Block”, “Storage1 Block”, “Storage2 Block”, “Retrieval1 Block” and “Retrieval2 Block”. More precisely, the “In Block” picks the operation to execute, according to the chosen queue selection and storage/retrieval sequencing rules, and defines the stop criterion of the simulation (run time expiration, pre-set tasks completion, etc.). The “RGVS Block” handles the vehicle bookings and travels in the system and prevents collisions when $N_V \geq 2$. Moreover, the “Update Marking Block” updates the current marking of the CTPN, while the “Save Marking Block” stores the actual state of the Petri net taking into account its timing. When a new route is assigned to part, the appropriate “Storage1 Block”, “Storage2 Block”, “Retrieval1 Block” or “Retrieval2 Block” is invoked, depending on which operation is assigned and which crane is involved. Hence, implementing the AS/RS in the Matlab-Stateflow environment results in a compact and modular model in which it is easy to modify the system implementation. Indeed, the Matlab-Stateflow machine reflects the modularity of the CTPN model: an additional “Storage Block” and a “Retrieval Block” are to be included in the chart if a supplementary aisle is to be added to the system, and some additional places, i.e., finite states machine sub-states, are to be inserted in the “RGVS Block”. On the other hand, if the number of trucks is increased and deadlock may occur

($N_v>2$), then an operation validation algorithm, i.e., a Matlab deadlock prevention routine, must be included in the “RGVS block”.

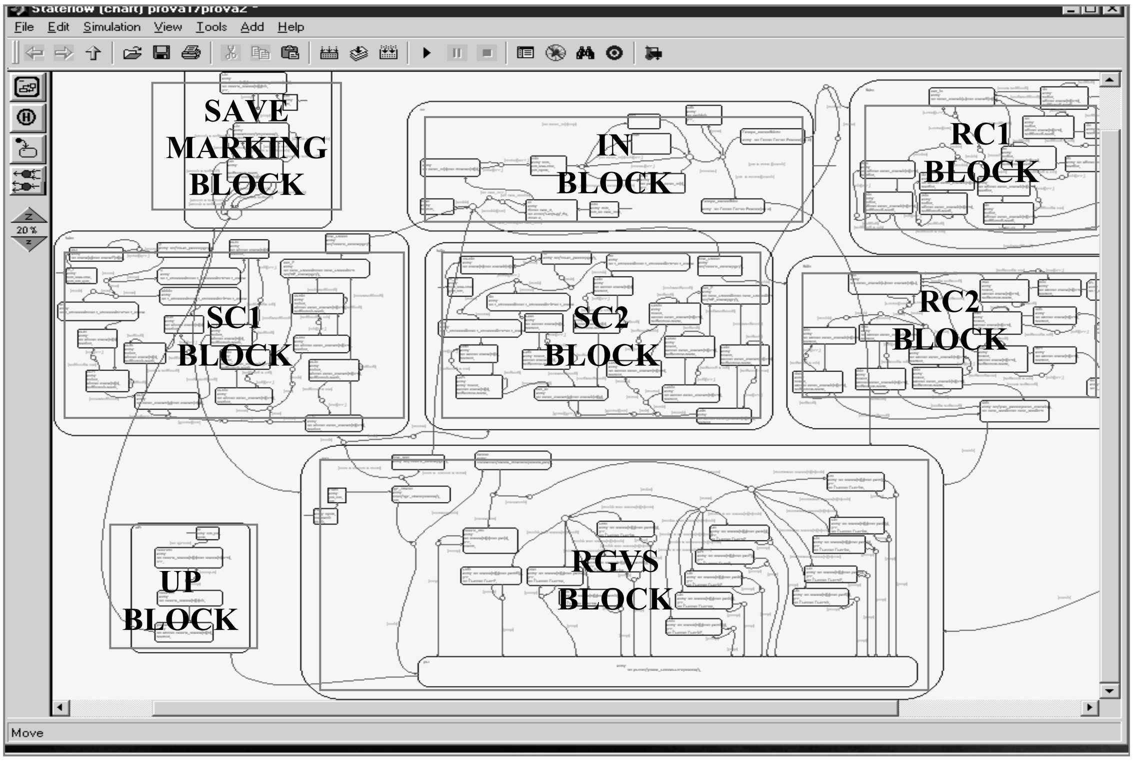


Figure 10. The Matlab-Stateflow machine implementing the CTPN model.

5.3 Simulation Results

Results of the simulation experiments are summarized in Figures 11 to 13 and in Table I. In Figure 11 results of a preliminary test on the case study are reported. In particular, three simulations with a fixed run time $T=10^6$ time units and one RGV ($N_V=1$) are carried out and the scheduling policies are compared with respect to throughput: the effectiveness of the interleave policy is apparent.

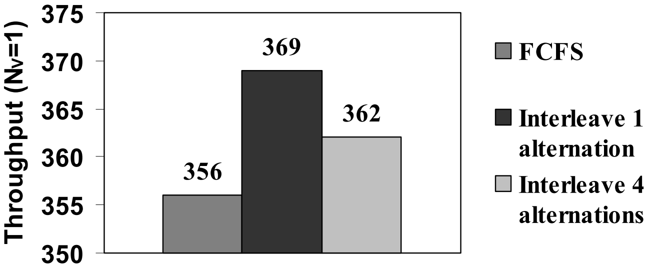


Figure 11. Throughput under different scheduling policies for a fixed run time $T=10^6$ for the simulated example ($N_V=1$).

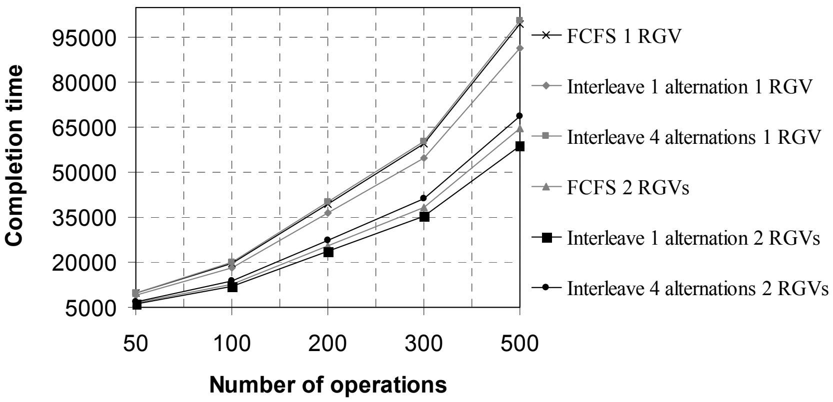
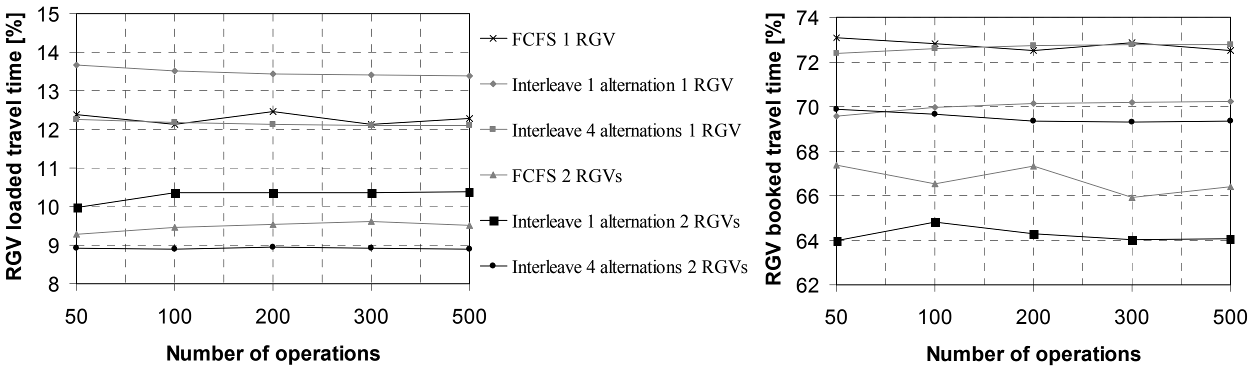


Figure 12. Completion times for several random sets of operations under different scheduling policies for the simulated example.



(a) (b)

Figure 13. RGVs mean percentage loaded (a) and booked (b) travel time for several random sets of operations under different scheduling policies for the simulated example.

Figure 12 reports the completion times of the simulations when the previously defined random sets of operations are executed under the scheduling policies. Figure 12 shows, as expected, that when the number of trucks increases from $N_V=1$ to $N_V=2$ the run times are considerably reduced, since operations may be carried out in parallel, although in general execution times are not halved, because trucks are occasionally blocked by the real time controller in order to avoid collisions.

The RGVs average percentage time of loaded and booked travel under the scheduling rules are respectively reported in Figures 13 a and 13 b. We remark that both the indices are substantially unaltered despite the variations in the scheduling rule and the increase in the number of tasks to perform. Moreover, both indices tend to diminish when the RGVs fleet sizes increases from $N_V=1$ to $N_V=2$.

Table I reports the cranes utilization under the different scheduling policies, showing that the index is nearly unaffected by a variation in the number of tasks or even in the scheduler control policies: about 50% of the simulation time cranes are busy either carrying parts or travelling empty to a loading zone, whereas in the rest of the execution time they are idle in the corresponding home position.

As a summary of the performance for the different scheduling policies considered, we remark that the AS/RS is more efficient in terms of throughput and execution time under the interleave rule with one storage and one retrieval task alternated on the same aisle. The same scheduling policy with four alternations follows, as regards effectiveness, whereas the less efficient management strategy is the FCFS rule. Moreover, the RGVs and cranes utilization indices are practically independent of the scheduling policy considered.

Tab. I. Cranes utilization for several random sets of operations under different scheduling policies for the simulated example.

Cranes utilization						
	Number of tasks	**50**	**100**	**200**	**300**	**500**
Scheduling policy	**FCFS 1 RGV**	50.00	50.00	50.00	50.73	50.00
	Interleave 1 alternation 1 RGV	51.73	50.00	50.00	50.00	50.00
	Interleave 4 alternations 1 RGV	49.99	50.00	50.00	50.00	50.00
	FCFS 2 RGVs	50.00	49.99	49.64	49.99	50.00
	Interleave 1 alternation 2 RGVs	50.00	50.00	50.00	50.00	50.00
	Interleave 4 alternations 2 RGVs	50.00	50.00	50.00	49.99	50.00

6. Conclusion

This paper proposes a Coloured Timed Petri Net (CTPN) model to describe in a concise and efficient way the dynamics of an Automated Storage and Retrieval System (AS/RS) serviced by Rail Guided Vehicles (RGVs). The paper defines in detail the CTPN modelling each subsystem: the storage/retrieval stations, the RGV system and the crane subsystems. The whole CTPN is modularly built so that if the system layout changes, the model can be easily modified. Moreover, the system state is the marking of the CTPN and is updated through the incidence matrix at each event occurrence. The unified framework allows us to obtain a resource-oriented model suitable for real-time control applications. Indeed, AS/RSs serviced by RGVs can suffer from collision and deadlock problems. The selected model simplifies both the application of several scheduling and sequencing policies for storage and retrieval tasks, as well as the implementation of resource allocation policies for deadlock avoidance. To validate the introduced model, an implementation has been performed by means of the software platform Matlab with Stateflow. Several control and management policies are tested to show how the simulated model can help to improve the overall system performance.

Research issues presently under investigation are the CTPN model implementation for more complex and large AS/RS, including the specification and the application of deadlock avoidance strategies.

References

AMATO F., and BASILE F., 2001, Optimal Control of Warehousing Systems with Simultaneous Crane and Shuttle Optimization, *Proc. of the 8th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Antibes, France.

DOTOLI M., and FANTI M.P., 2004, A Coloured Timed Petri net model for real time control of AGV systems, to appear on *Int. Journal of Production Research*.

EZPELETA J., and COLOM J. M., 1997, Automatic synthesis of Colored Petri Nets for Control of FMS, *IEEE Trans. on Robotics and Automation*, **13**, 327-337.

EBEN-CHAIME M., and PLISKIN N., 1996, An integrative model for automatic warehousing systems, *Int. J. Computer Integrated Manufacturing*, **9**, 286-292.

FELDMANN K., and COLOMBO A.W., 1998, Material Flow and Control Sequence Specification of Flexible Systems Using Coloured Petri Nets, *Int. J. Adv. Manufacturing Technology*, **14**, 760-774.

JENSEN K., 1992, *Colored Petri Nets, Basic Concepts, Analysis methods and Practical Use*, Vol.I, EATS Monography and Theoretical Computer Science, Springer Verlag, New York.

LEE S.G., DE SOUZA R., and ONG E.K., 1996, Simulation Modeling of a Narrow Aisle Automated Storage and Retrieval System (AS/RS) Serviced by Rail-Guided Vehicles, *Comp. in Ind.*, **30**, 241-252.

LIN S.C., and WANG H.P.B., 1995, Modeling an Automated Storage and Retrieval System Using Petri Nets, *Int. J. Prod. Res.*, **33**, 237-260.

LINN R.J., and WISK R.A., 1990, An Expert System Framework for Automated Storage and Retrieval System Control, *Comp. Ind. Engineering*, **18**, 37-48.

NANDULA, M., and DUTTA, S. P., 2000, Performance evaluation of an auction-based manufacturing system using coloured Petri nets. *International Journal of Production Research*, **38**, 2155-2171.

THE MATHWORKS INC., 1997, *Stateflow User's Guide Version 4*.

VAN DEN BERG J.P., and GADEMANN, A.J.R.M. 2000, Simulation study of an automated storage/retrieval system, *Int. J. Prod. Res.*, **38**, 1339-1356.

WANG L.-C, and WU S.-Y, 1998, Modeling with colored timed object-oriented Petri Nets for automated manufacturing systems, *Computers ind. Engng*, **34**, 463-480.

WU, N., and ZHOU, M.C., 2001, Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems. *IEEE Transactions on Robotics and Automation*, **17**, 657-668.