Using Petri Nets in the analysis of sequential automata models with direct applications on the transport systems with accumulation areas

Dan Ungureanu-Anghel
Dep. of Automation and Applied Informatics,
"Politehnica" University of Timisoara
Faculty of Automation and Computers

1. Introduction

The development of flexible manufacturing systems (FMS) led implicitly to the revaluation of place and significance of the transport systems. A FMS cannot be considered efficient if the afferent transport system is not itself an efficient one. In these circumstances, the importance of the transport system had risen, from the classical phase, in which this had a role preponderant to transport the pieces from one machine to another, to a phase in which the tasks of the transport system were extended (automatic sorting, the automatic selection of the destination for a piece, computing and choosing the optimal route, etc.), talking now about intelligent transport systems.

Transport systems with accumulation area (TSAA) belongs to the class of intelligent transport systems, being capable, based on some algorithms or on some functioning specifications (static – determined during design, or dynamic – which can be modified during the operation depending on the operator demands) to execute transfers of pieces or other transport elements (trolleys), according to the requirements. TSAA founds applicability in the most diverse places: automated warehouses, flexible fabrication lines, sorting systems, etc.

In specialized literature the field of industrial transport systems was not neglected. However, the majority of the authors approach these fields from the prospect of the analysis of the materials flux and of the allocated transport time (Chiang et.al 2008)(Ge et. al. 2008)(Polic&Jezernik 2008).

Other approaches related to transport systems can be found in the papers that deals with either the road or rail system (Li et al 2008) (Giua & Seatzu 2008).

The approach of industrial transport systems, within this chapter, it is done different from the way of approach existing in literature, once by direct concentration on a certain type of transport system (TSAA), and the second is done not by the analysis of the materials flux, but by the analysis the mode of administration at the lowest level of materials transport. Starting from the base structure of the TSAA it was identified a basic mechanical structure, called node, for which it was elaborated the adequate model of automatic type. After the

validation of this model, following the structural and behaviour analysis, it passed to the obtaining of generic models.
The higher complexity of these systems, make almost impossible the modelling based on differential equations, which led to the TSAA approach as a discrete event systems (DES).
For the modelling of the constitutive elements of TSAA, and of TSAA on the whole, considered as DES, there will be used models of untimed deterministic automatic type. Because of the fact that the structural and behaviour analysis of the larger models of automatic type is difficult, as well as the lack of dedicated tools, it led to the use of one methodology less common, but which has a very high efficiency. The used method of approach is based on the conversion of the models of automatic type in models of untimed labelled Petri Net type, the structural and behaviour analysis being made using the PetriNets Toolbox from Matlab.
In order that the obtained results from the analysis of the model of Petri Net type to be applicable (in fact to validate) to the model of automatic type it is necessary to fulfil the following condition: a model of Petri Net type obtained from a model of automatic type need to have the topology of state machine type. Only if this condition is complete satisfied the results can validate entirely the model of automatic type. This compulsoriness is imposed especially by the construction manner of the models of automatic type, respectively Petri Nets. Models of automatic type are based on the propagation of the model topology, while the models of Petri Net type are based on the propagation of the markings.
Within this chapter, is pursued the creation of a unified procedural and methodological background, for the conjunction between the models of untimed deterministic automatic type and the untimed label Petri Nets type with direct applications on the transport systems with the accumulation areas.
For the beginning, within the chapter is done a presentation of the theoretical aspects of the connections between the models of automatic type and the models of Petri Nets type connected to: theoretical notions concerning the relation between the models of automatic type and the models of Petri Nets type; the algorithm used for the conversion of the models of automatic type in models of Petri Net type; the validation conditions of the obtained results through the analysis of the models of Petri Net type equivalent to the basic models of automatic type.
Having as a base the theoretical considerations, the approach of the TSAA field is done in the first phase through a general presentation of TSAA and of their basic structures, continuing with the modelling of the constitutive elements of TSAA with the help of the models of automatic type, starting from the basic structure and ending with generic models. The structural and behaviour analysis of the established models of automatic type is realized by their conversion in models of untimed Petri Net type, on these models is being made the structural and behaviour analysis.
Some conclusions are presented at the end of the chapter.

2. Theoretical considerations connected to the models of automatic type, respectively Petri Nets

Both modelling methods, automata respectively Petri Nets, have at the base the using of states and transitions for the description of a system, it results that between them are similarities and differences (Cassandras & Lafortune 2001).

Modelling with automata respectively Petri Nets, as well as the model type choosing it is left to the modeller latitude (Murata 1989).
In this way, if it is desired the modelling of a DES only based on its external events, without having an interest in an explicit way on the hidden activities, and then a model of automatic type is satisfactory. If instead is desired a refinement of the internal operations then a model of Petri Net type is more favourable (Cassandras & Lafortune 2001) (Pastravanu 1997).
For the model of automatic type it was considered that these are of deterministic type. A deterministic automata is defined as follows:

Definition 1. Untimed deterministic automata (Cassandras & Lafortune 2001)

A deterministic automata, marked by G, is a sextuple

$$G = (X, E, \delta, \Gamma, x_0) \tag{1}$$

where:

- X is the set of states;
- $\sum$ is a finite set of events associated with the transitions from G;
- $\delta : X \times \sum \rightarrow X$ is the transition function: $\delta(x,e) = y$, means the appearance of a transition labelled through the event e in the state x, which has as a result the transition in the state y; in general, δ is a partial function on his domain of definition.
- $\Gamma : X \rightarrow 2^E$ is the active event function; $\Gamma(x)$ is the set of all events e for which $\delta(x,e)$ is defined and is called by the active event from G accordingly to the active state x.
- x_0 is the initial state of the system.

For the model of Petri Net type it was considered that these are of labelled type and untimed. An untimed labelled Petri Net is defined as follows.

Definition 2. Labelled Petri Net (Cassandras & Lafortune 2001)

A labelled Petri Net is a weighted graph

$$PN = (P, T, F, W, \Sigma, l, M_0) \tag{2}$$

(net topology), where:

- P represents the finite multitude of positions, *with* $P = \{p_1, p_2, p_3, \dots, p_n\}$;
- T represents the finite multitude of transitions, with $T = \{ t_1, t_2, t_3, \dots, t_m\}$;
- $F \subseteq (P \times T) \cup (T \times P)$ represents the multitude of arcs from the positions to transitions and from transitions to positions, each arc being represented by (p_i, t_j), respectively (t_j, p_i), where $i,j \in N$;
- $W : A \rightarrow \{1,2,3...\}$ represents the balanced function of arcs;
- $\sum$ represents the set of events for the transitions labels;
- $l : T \rightarrow \sum$ represents the labelled transition function;
- $M_0 \in N^n$ represents the initial marking of the net .

As it was mentioned earlier, the automatons and the Petri Nets operate with states and transitions. In this way are possible transformations of the models of automatic type in models of Petri Net type and vice versa. The subject of synthesis of the models from one way of representation in the other one is presented in detail in literature.(Pastravanu 1997)(Cortadella et al 1995)(Cortadella et al 1998)(Hellgren et al 2001)(Cassez & Roux 2004).. The algorithm used for the conversion of untimed deterministic automata $G = (X, \Sigma, \delta, \Gamma, x_0, X_m)$, into an untimed Petri Net $PN = (P, T, F, W, M_0)$ is:

Step 1. The determination of the multitude of positions *P*

Each state from X is transformed into an equivalent position in *P*.

$$\forall x_i \in X : x_i \rightarrow p_i, p_i \in P \tag{3}$$

which means that the multitude *P* has exactly so much elements that are in the multitude *X*.

Step 2. The determination of the transitions multitude T

For each pair of states from *X*, designated by *(x,x′)*, with

$$x' = \delta(x, e), \quad e \in \Gamma(x) \tag{4}$$

it has associated a transition labelled $t_i \in T$, in the Petri Net.

In the end result the multitude:

$$T = \bigcup \{t : (x, x'), x, x' \in X, x' = \delta(x, e), e \in \Gamma(x)\} \tag{5}$$

Step 3. The determination of the arc multitude F

It is attached the arcs $(p,t),(t,p') \in F$ all having the weight 1,*w(p,t)=1, w(t,p′)=1* if exists the correspondence $(x,x') \rightarrow (p,p')$

With other words:

$$if\,(x,x') \rightarrow (p,p') : x' = \delta(x,e), e \in \Gamma(x)\ then\ \exists (p,t)\ and\ (t,p')\ p, p' \in P, t \in T \tag{6}$$

In the end, result the multitude:

$$F = \{(p,t) : p \in P, t \in T\} \cup \{(t,p') : p' \in P, t \in T\} \tag{7}$$

and

$$W : F \rightarrow \{1\} \tag{8}$$

Step 4. The determination of the event multitude Σ

The set of events is obtained as being:

$$\Sigma = E \tag{9}$$

Step 5. The determination of transition functions *l*. The label attached to the transition is the event *e* which led to change of the state *x* in *x′*.

For the transition $t \in T : (p,t),(t,p') \in F$ the event and function l and associated label are defined as:

$$l(t) = e,\ e \in \sum and\, x' = \delta(x,e) \tag{10}$$

Step 6. The determination of initial marking M_0
The initial marking M_0 is assigned accordingly to the initial state x_0.

The way of using the conversion algorithm from the model of untimed deterministic automatic type in the model of untimed labelled Petri Net type is exemplified through the next example.
Example . It is considered the untimed deterministic automata from figure 1.

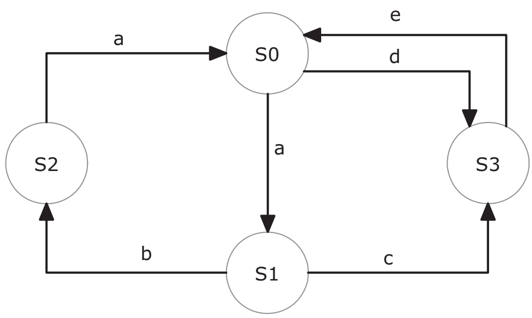

Fig. 1. The structure of the considered untimed deterministic automata

The considered automata formalized mathematical by: $AS = (X, E, \delta, \Gamma, x_0)$ where:

- The multitude of states: $X = \{S_0, S_1, S_2, S_3\}$
- The multitude of events: $E = \{a, b, c, d, e\}$
- The multitude of possible events and the transition functions of the states:

$\Gamma(S_0) = \{a, d\}$	$\delta(S_0, a) = S_1$ $\delta(S_0, d) = S_3$
$\Gamma(S_1) = \{b, c\}$	$\delta(S_1, b) = S_2$ $\delta(S_1, c) = S_3$
$\Gamma(S_2) = \{a\}$	$\delta(S_2, a) = S_0$
$\Gamma(S_3) = \{e\}$	$\delta(S_3, e) = S_0$

- The initial state: $x_0 = S_0$.

For obtaining the equivalent untimed labelled Petri Net is applied the presented algorithm:

Step 1. The determination of multitude of positions P:

$$S_i \rightarrow p_i;\ i = \overline{0,3}$$

from where results that $P = \{p_0, p_1, p_2, p_3\}$

Step 2. The determination of multitude of transitions T:

State	States connection	The transition function	The associated transition
S_0	(S_0, S_1)	$\delta(S_0, a) = S_1$	t_1
	(S_0, S_3)	$\delta(S_0, d) = S_3$	t_2
S_1	(S_1, S_2)	$\delta(S_1, b) = S_2$	t_3
	(S_1, S_3)	$\delta(S_1, c) = S_3$	t_4
S_2	(S_2, S_0)	$\delta(S_2, a) = S_0$	t_5
S_3	(S_3, S_0)	$\delta(S_3, e) = S_0$	t_6

From where results that $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$

Step 3. The determination of multitude of arcs F and of their shares

Transition	States connection	The connection of equivalent positions	Arcs
t_1	(S_0, S_1)	(p_0, p_1)	$(p_0, t_1), (t_1, p_1)$
t_2	(S_0, S_3)	(p_0, p_3)	$(p_0, t_2), (t_2, p_3)$
t_3	(S_1, S_2)	(p_2, p_3)	$(p_1, t_3), (t_3, p_2)$
t_4	(S_1, S_3)	(p_2, p_4)	$(p_1, t_4), (t_4, p_3)$
t_5	(S_2, S_0)	(p_3, p_1)	$(p_2, t_5), (t_5, p_0)$
t_6	(S_3, S_0)	(p_4, p_0)	$(p_3, t_6), (t_6, p_0)$

From where results that
$$F = \{(p_1, t_1), (p_1, t_2), (p_2, t_3), (p_2, t_4), (p_3, t_5), (p_4, t_6)\} \bigcup \{(t_1, p_2), (t_2, p_4), (t_3, p_3), (t_4, p_4), (t_5, p_1), (t_6, p_1)$$

The multitude of arcs shares:
$$W = \{w(p_1, t_1) = 1, w(p_1, t_2) = 1, w(p_2, t_3) = 1, w(p_2, t_4) = 1, w(p_3, t_5) = 1, w(p_4, t_6) = 1,$$
$$w(t_1, p_2) = 1, w(t_2, p_4) = 1, w(t_3, p_3) = 1, w(t_4, p_4) = 1, w(t_5, p_1) = 1, w(t_6, p_1) = 1\}$$

Step 4. The determination of multitude of events Σ
$$\Sigma = \{a, b, c, d, e\}$$
Step 5. The determination of transition functions l

Transition	Connected arcs	Associated events
t_1	$(p_1,t_1),(t_1,p_2)$	a
t_2	$(p_1,t_2),(t_2,p_4)$	d
t_3	$(p_2,t_3),(t_3,p_3)$	b
t_4	$(p_2,t_4),(t_4,p_4)$	c
t_5	$(p_3,t_5),(t_5,p_1)$	a
t_6	$(p_4,t_6),(t_6,p_0)$	e

Step 6. The determination of initial marking M_0

The initial state is $x_0 = \{S_0\}$ results the initial marking $M_0^T = [1,0,0,0]$

The graphical representation of the equivalent Petri Net, obtained after the application of the conversion algorithm, is presented in figure 2.

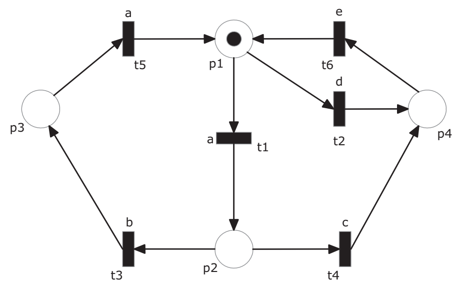

Fig. 2. The equivalent Petri Net

3. The principles of realization and functioning of the transport systems with accumulation areas

The industrial transport systems are of different types, depending on the effective mode in which is realized the transport. In this way exists horizontal transport systems or suspended transport systems. Each type of system has its own constructive and functional features, both of them belonging to the TSAA field. In figure 3 is presented a general view of TSAA.

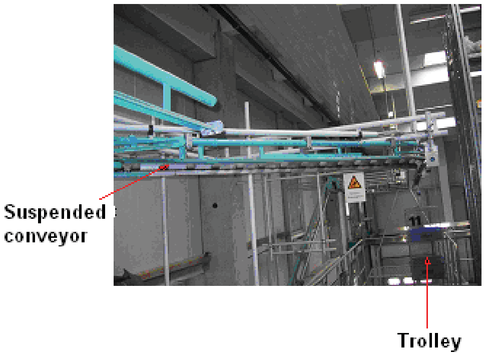

Fig. 4. Suspended transport system

Within the chapter it will be taken into consideration only the suspended transport systems, but the obtained results can be applied also to the horizontal transport systems. Constructive, these are based of the using of brush conveyors as a transport support and have as a main property the assurance of accumulation areas (jams) of the transported elements without being necessary the stopping of conveyor moving motors. The conveyor is moved by an electric motor connected through a mechanical reducer, situated at one of the conveyor extremities. This motor assures the moving of the conveyor with constant speed. The transport element used in the case of these transport systems is the trolley. Each trolley is identified in the system by a unique code bar. This code is used for the determination of the point in which is situated the trolley and for the computing of the route on which this has to follow.
In figure 4 is presented a general picture of the way in which the transport is performed inside a complete automated warehouse of exchange pieces.

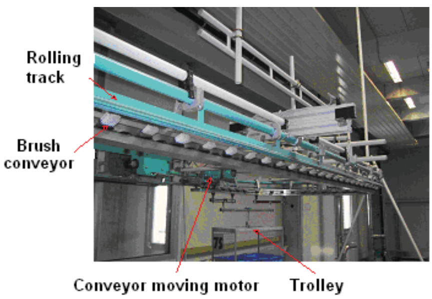

Fig. 5. General picture of a transport system

The main element which assures the stopping of a trolley depending on the used controlling algorithm is the stopper. In figure 5 is presented a stopper used in the case of these systems.

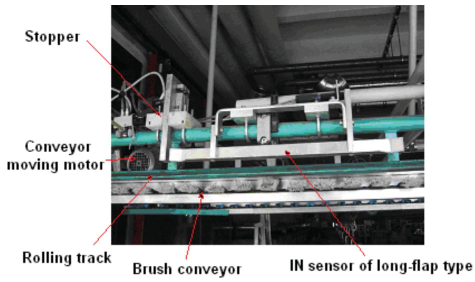

Fig. 6. Stopper and long-flap type sensor

In figure 5 is observed also the sensor which signalize the fact that in the front of the stopper there is a trolley.

4. Modelling the systems with accumulation areas

The basic idea in the modelling of such systems is that of assuring the modularization of the elements which makes the transfer of trolleys from one accumulation area into another (Ungureanu-Anghel 2006a)(Ungureanu-Anghel 2006b).

For the modelling of such systems it was established a basic structure (node of type 1 – an input an output), with the help of which can be obtained a general model regardless of its complexity. The observation which has to be made is connected to the fact that further will be no more references connected to the administration of the trolley flux and of their routes in the system, approaching only the issues connected to the modelling of nodes (Ungureanu-Anghel 2006a).

4.1 Node of type 1 – an input an output

The basic structure of such a node is presented in figure 6.

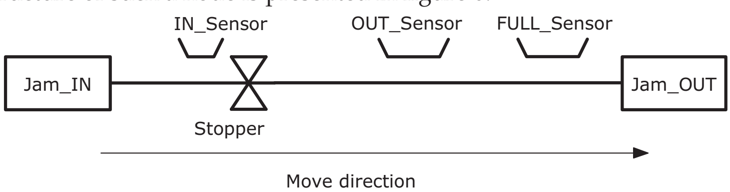

Fig. 7. Node of type 1. An input an output

As can be observed in the mentioned figure, such a node contains a single stopper, having an input and a single output. The sensors used have the following functions:

IN_Sensor	The sensor from stopper. With the help of this sensor it is signalized the fact that a trolley is in the front of the stopper.
OUT_Sensor	The output sensor. With the help of this sensor is followed the moment in which a trolley has left the stopper area (of the node).
FULL_Sensor	Full sensor. With the help of this sensor it is checked the available space in the next jam.

Jam_IN represents the input jam and Jam_OUT the output jam of the node. The moving motors were not presented because, as it was mentioned, they function continuous.
Before to present the functioning mode of the node of type 1, it is necessary to clearing up the mode of testing of FULL_Sensor.
Since the trolleys are in motion, the signal taken from FULL_Sensor is processed as it is presented in figure 7.

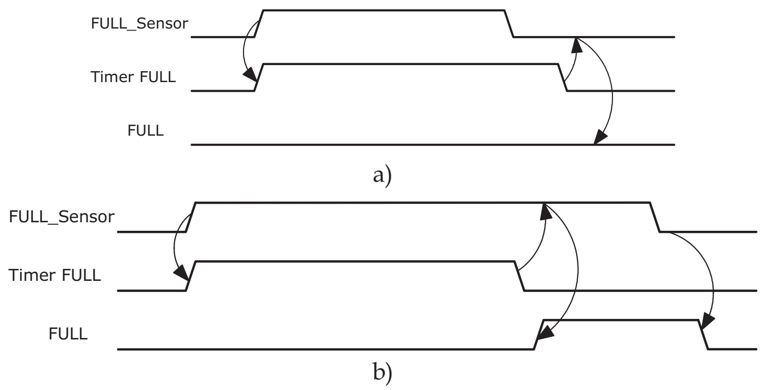

Fig. 8. The mode of processing the signal from FULL_Sensor. a) situation without detection FULL; b) situation of detection FULL

The FULL timer is used for verification with a certain delay of the signal from FULL_Sensor. If at the expiration of the timer the FULL_Sensor is inactive then the event FULL is inactive (figure 7.a), situation in which in Jam_OUT there is still space for at least a trolley. If at the expiration of the FULL timer the FULL_Sensor is active then the event FULL is set as being active, which corresponds to the situation Jam_OUT full. Deactivation of the FULL event is done in the moment in which FULL_Sensor becomes inactive (figure 7.b).
The functioning mode of the node of type 1 starts from the premise that the stopper is closed. A trolley situated in Jam_IN is transported by TEF and at one moment activates the IN_Sensor. The condition for moving the trolley in Jam_OUT area is given by: OUT_Sensor not to be active and FULL not to be active. In other words: in the area after the stopper doesn't have to be a trolley and in the Jam_OUT area has to be at least one free place. The mechanical structure of the node is thus realized so at the activation of OUT_Sensor the trolley has passed entirely by the stopper. The activation of OUT_Sensor has as an effect the closing of the stopper, realizing in this way the separation of two trolleys. A new trolley can be transferred from Jam_IN area in Jam_OUT area only after the OUT_Sensor is inactive, the previous trolley has left the area covered by the OUT_Sensor. For avoiding the uncontrollable situations, for example the stopper is opened and the trolley is jammed and it not touched the OUT_Sensor, are introduced two additional control timers named T1, respectively T2. T1 is used for verification of touching the sensor OUT_Sensor, while T2 for leaving of OUT_Sensor. If one of these timers expires before the touching, respectively the leaving of OUT_Sensor, it means that had appeared an abnormal state of functioning.
The functioning of the node of type 1 without error, described on the base of chronograms is presented in figure 8.

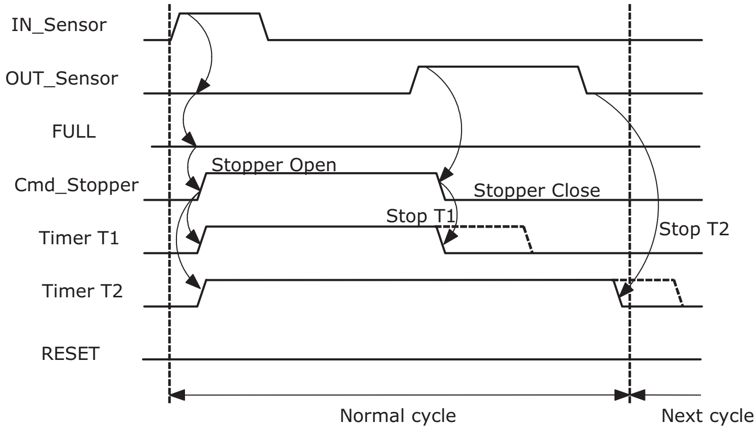

Fig. 9. The functioning chronograms of the node of type 1

The RESET event was introduced for giving the possibility of return in the WAIT state from the error states caused by the expiration of the timers T1 and T2.
The segments with doted line for the timers T1, respectively T2, represents the normal duration of functioning of the timers.

4.2 The modelling of the node of type 1 with the help of automata

The structure of the sequential automata taken into consideration for the modelling of the node of type 1 is presented in figure 9.
The meanings of the sequential automata states *AS_N1* are presented in the table 1.

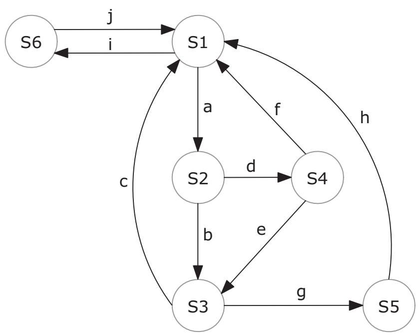

Fig. 10. The structure of the sequential automata corresponding to the node of type 1

State	Comment
S1	WAIT – The initial state of the system. In this state is waited that a trolley to reach the stopper (IN_Sensor = 1). Also in this state it is assured the transport of a trolley from the input sensor from the nod until the IN_Sensor.
S2	OPEN – the state in which the stopper is open, assuring in this way the transfer of the trolley in the next jam.
S3	CLOSE – state corresponding for closed stopper, but the trolley is still in node area, not being completely passed to the next jam.
S4	ERROR 1 – error state which is installed if during the moving time of the trolley it has appeared a critical situation. T1 has expired
S5	ERROR 2 – error state which is installed if during the moving time of the trolley it has appeared a critical situation. T2 has expired
S6	ERROR 3 – error state which is installed if there is no trolley in IN_Sensor but OUT_ Sensor = 1

Table 1. The meanings of automata states AS_N1

The events under which take place the transitions between states are presented in table 2.
The net topology (result obtained with the help of PNT) is shown in figure 10.

Fig. 12. The Petri Net topology validated by PNT

As can be seen, following the transformation of the model of automatic type in the Petri Net model, the net topology obtained is also of type automatic (“state machine”), which allows the analysis of the model of automatic type using the methods from the Petri Nets.

Event	Comment
a	Event which assures the passing from state S1 in state S2. Validated (active) based on the condition: IN_Sensor * !OUT_Sensor * !FULL.
b	Event which assures the passing from state S2 in state S3. Validated (active) based on the condition: OUT_Sensor = 1.
c	Event which assures the passing from state S3 in state S1. Validated (active) based on the condition: OUT_Sensor = 0.
d	Event which assures the passing from state S2 in state S4. Validated (active) based on the condition: Timer T1 expirat .
e	Event which assures the passing from state S4 in state S3. Validated (active) based on the condition: OUT_Sensor = 1.
f	Event which assures the passing from state S4 in state S1. Validated (active) based on the condition: RESET = 1.
g	Event which assures the passing from state S3 in state S5. Validated (active) based on the condition: Timer T2 expirat.
h	Event which assures the passing from state S5 in state S1. Validated (active) based on the condition: OUT_Sensor + RESET = 1.
i	Event which assures the passing from state S1 in state S6. Validated (active) based on the condition: !IN_Sensor * OUT_Sensor.

Table 2. The meanings of the events corresponding to the automata AS_N1

The corresponding incidence matrices are:

The input incidence matrix:

$$A_I = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The output incidence matrix:

$$A_O = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Incidence matrix:

$$A = A_O - A_I = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

The corresponding cover tree is presented in figure 11.

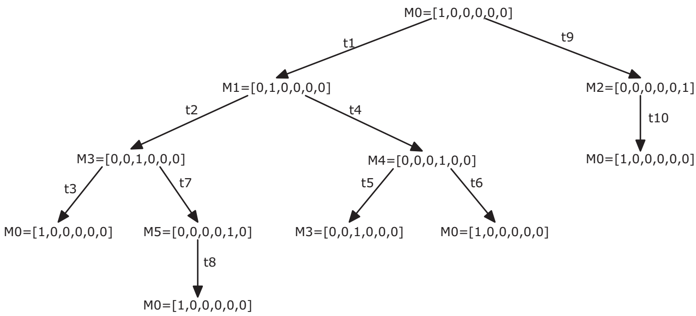

Fig. 13. The cover tree of Petri Net *PN_AS_N1*

Studying the behaviour properties of the Petri Net *PN_AS_N1*, based on the cover tree results:

- the net is *limited* (the symbol ω does not appear in the cover tree);
- the net is *safe* (the markings from all the cover tree nodes contains only „0” and „1”);
- the net is *non-blocking* (all transitions have associated arcs in the cover tree);
- the net is *accessible* (any marking can be reached starting from M_0).

The corresponding cover graph is presented in figure 12.

The resulted structure for the cover graph confirms the fact that the model of Petri Net derived from the model of automatic type corresponding to the node of type 1 is *accessible*.

Concluding, it can be stated that the chosen model is *viable* from the behaviour point of view.

The analysis of the Petri Net *PN_AS_N1* from structural point of view it is done based on the analysis of the existence of the invariants of type *P* respectively *T*. To find out the number of invariants *P*, respectively *T*, it was applied the theorem for the determination of the invariants (Pastravanu 1997), which states that if the incidence matrix A (of dimension nxm) of a Petri Net, has the rank r, then: the net posses m-r base invariants P, and each invariant P of the net PN can be written as a linear combination of these; the net posses n-r base invariants T, and each invariant T of the net PN can be written as a combination of these.

In the considered case, the rank of the matrix A is: $rank\ A = 5$.

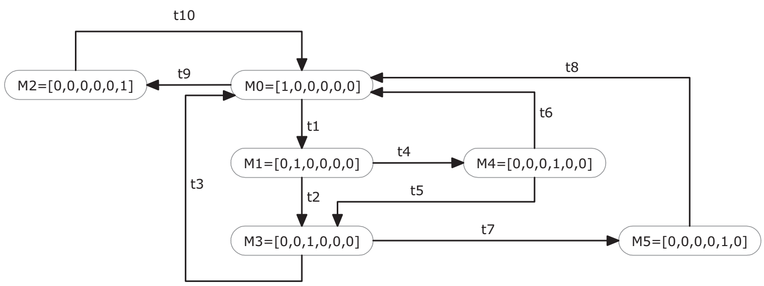

Fig. 14. The cover graph of Petri Net *PN_AS_N1*

Results that the Petri Net *PN_AS_N1* is covered by 1 invariant of type *P*. In a similar way, in the case of the invariants of type *T* results a number of 5 such invariants. The existence of

the invariants of type P, respectively T leads to the conclusion that *the Petri Net PN_AS_N1 is conservative and structural limited, respectively is consistent and repetitive* (Pastravanu 1997).
In the end can be *stated* that the chosen structure for a classic sequential automata corresponding to the node of type 1 is viable, the structure can be implemented, being sure that don't exists conditions of appearance of the blocking or appearance of uncontrollable situations.

4.3 The modelling of the node „one" input and „m" outputs with the help of automata

The base structure for such a node is presented in figure 13 (Ungureanu&Prostean 2007) (Ungureanu et al 2008).

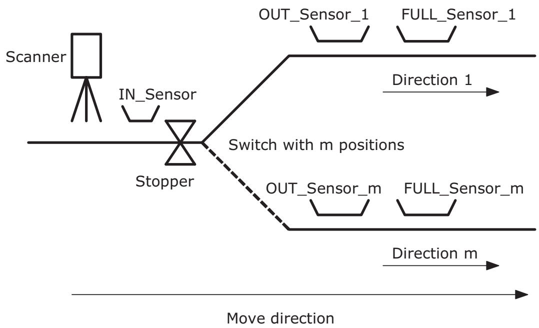

Fig. 15. The base structure of a node with one input and m outputs

Additional to the aspect of the node of type 1, is observed the appearance of a scanner and of a switch with *m* positions. With the help of the scanner, by reading the barcode from a trolley and following the list of destinations attached to the trolley, can be determined the direction on which has to be transferred the trolley and is commanded the corresponding switch positioning.
After the switch positioning everything is reduced to a node of type 1.
The position of the switch is not verified through some sensors, because if the switch is not accurate positioned it will appear an abnormal situation of functioning of no touching the corresponding output sensor.
The structure of the sequential automata considered for the modelling of such a node is presented in figure 14.

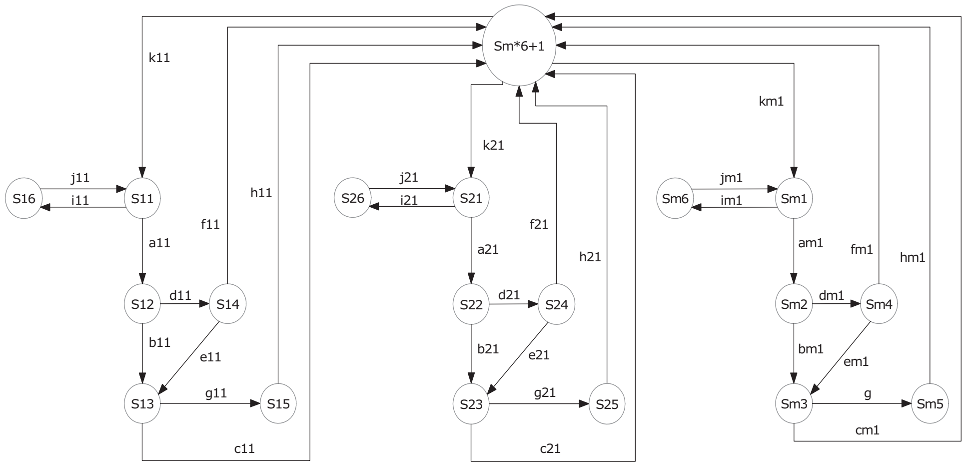

Fig. 16. The structure of the sequential automata corresponding to the node with „*one*” input and „*m*” outputs

The sequential automata of the node is realized by the synchronization of *m* sequential automata *AS_N1*, one for each input line (stopper).
Mathematically, the automata *AS_Nm* corresponding to the node with one input and m outputs is described as follows:

- the multitude of states: $X = \bigcup_{i=1}^{m} \{Si1, Si2, Si3, Si4, Si5, Si6\} \cup \{S(m \cdot 6 + 1)\}$
- the multitude of events: $\Sigma = \bigcup_{i=1}^{m} \{ai1, bi1, ci1, di1, ei1, fi1, gi1, hi1, ii1, ji1, ki1\}$
- the multitude of possible events and the transition functions of the states:

$\Gamma(S) = \bigcup_{i=1}^{m} \Gamma(Si)$ where:

$\Gamma(Si1) = \{ai\}$ $\quad \delta(Si1, ai) = Si2$

$\Gamma(Si2) = \{bi, di\}$ $\quad \delta(Si2, bi) = Si3,$ $\quad \delta(Si2, di) = Si4$

$\Gamma(Si3) = \{ci, fi\}$ $\quad \delta(Si3, ci) = S(4 \cdot m + 1),$ $\quad \delta(Si3, fi) = Si4$

$\Gamma(Si4) = \{ei, gi\}$ $\quad \delta(Si4, ei) = Si3$ $\quad \delta(Si4, gi) = S(4 \cdot m + 1)$

$\Gamma(S(4 \cdot n + 1) = \bigcup_{i=1}^{m} \{hi\}$ $\quad \delta(S(4 \cdot m + 1), hi) = Si1$

- initial state: $x_0 = S(4 \cdot m + 1)$.

The reasons linked to the transformation of the model of automatic type in the model of Petri Net type, presented in section 2, are valid also in the case of the node with one input and m outputs.
In figure 15 is presented the structure of the Petri Net associated to the sequential automata considered in the case of the node with one input and *m* outputs.

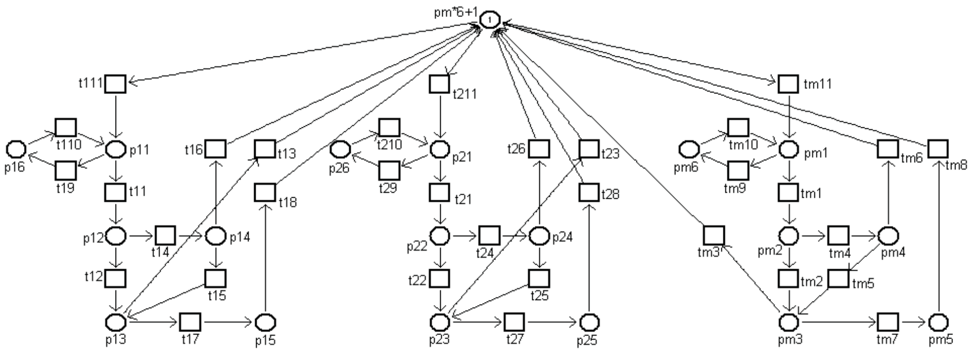

Fig. 17. The structure of the Petri Net associated to the sequential automata corresponding to the node with one input and m outputs

The net topology validated by PNT is presented in figure 16.

Fig. 18. The equivalent Petri Net topology validated by PNT

The obtained results confirms that after the effectuation of the conversion from the model of automatic type in the model of Petri Net type, the obtained Petri Net topology is also of automatic type („State machine”).
The considered Petri Net is formalized by the quintuple:

$$PN\_AS\_Nm = (P, T, F, W, M_0)$$

where:

- $P = \bigcup_{i=1}^{m} \{p_{i1}, p_{i2}, p_{i3}, p_{i4}\} \cup \{p_{4m+1}\}$