Modeling Framework for Automated Manufacturing Systems Based on Petri Nets and ISA Standards

E. G. HERNANDEZ-MARTINEZ[1], E. S. PUGA-VELAZQUEZ[2], S. A. FOYO-VALDES[2], J. A. MEDA-CAMPAÑA[2]

[1] Departamento de Ingenierías, Universidad Iberoamericana Ciudad de México,
Prolongación Paseo de la Reforma 880,
Lomas de Santa Fe, Del. Álvaro Obregón, C.P. 01219, México, D.F.
eduardo.gamaliel@ibero.mx

[2] Departamento de Ingeniería Mecánica, Sección de Estudios de Posgrado e Investigación,
ESIME Zacatenco, Instituto Politécnico Nacional,
Av. IPN s/n, Col. Lindavista, Del. Gustavo A. Madero, C.P. 07738, México, D.F.
erika_selenep@hotmail.com, sergiofoyo@live.com.mx, jmedac@ipn.mx

Abstract: This work presents a systematic methodology to model Automated Manufacturing Systems using Petri Nets. The modelling strategy consists in the definition and the interconnection of some generic Petri Net models applied to the discrete-event dynamic behaviour of the equipments and its procedures. It is based on the industrial standards ISA-88 and ISA-95, where the classification of equipment and the definition of their generic process tasks are suggested, separately of the product manufacturing recipes. The approach provides a formal and ordered methodology to study industrial automated systems where the equipment availability, storage limitations, sharing resources and logic precedence between process tasks appear in the Petri model. A complete case of study related to an automated cell is presented which includes a network of PLC's and industrial robots.

Keywords: Petri nets, Discrete event dynamic systems, Manufacturing systems, ISA Standard.

1. Introduction

The coordination of equipment of Automated Manufacturing Systems (AMS's) has been widely studied by the Discrete-event systems (DES) community, due to the asynchronous and concurrent dynamical behaviours that can be captured by some logical and temporal mathematical structures as Finite State Automata (FSA) or Petri Nets (PN) [1].The PN formalism has the advantage of a clear graphical description and mathematical support to represent logical dependences, process synchronization, resource allocation, etc. avoiding in this way, the exponential state growth of the FSA in the modelling of real AMS's [2, 3].

The AMS DES-based modelling is commonly addressed in two control levels. The low level deals with the design of routines of pneumatic, hydraulic and electrical devices and its translation to the programming languages of local controllers like PLC's, for instance[4, 5].On the other hand, the high level studies the case of equipment and modules coordination of AMS, where the concurrency, blocking, fault detections, time optimization of routines and the computer-based supervision appear in the DES model to facilitate the manufacturing of different and concurrent products [1, 6].

Despite of the mathematical advances in the dynamics of the original and modified PN formalisms applied to AMS's, the most of the approaches do not provide the clarity to the automation engineers to visualize the advantages of a DES model in a real AMS and the use of its mathematical analysis for the productivity improvement. In this sense, a recent interest in the DES community focuses on the modelling and translation of the PN models to the local controllers of the AMS attending the restrictions and suggestions of industrial standards. Thus, the control implementation is extrapolated from the pure mathematical analysis to the industrial automation context. Two of these standards are the ISA-88 [7] and ISA-95[8] which proposes the AMS coordination through the hierarchical classification of the equipment and the definition of generic process tasks. Thus, the manufacturing of products is reduced to a recipe composed by the serial or concurrent ordering of the process tasks obeying logic precedences, storage limitations and the availability of the human and material resources. This task-based modelling is suitable for the case of flexible AMS, where the market

conditions, customer requirements and delivery times demand the manufacturing of different products within the same day. So the coordination of the AMS equipment must enable the maximal concurrence and less reprogramming time of the local controllers for product changes.

Recent works about the mixing of industrial standards and DES are [9, 10], where some service-oriented frameworks allow the integration of PN models in 2D/3D digital software tools. The IEC61499 standard is applied to the design of supervisory control of AMS in [11]. Manufacturing Execution Systems(MES) provides specifications to PN in [12] and decision-making systems are applied to the design of PN only for a disassembling process of electronic products in [13]. Specifically, for the case of the ISA-95 and ISA-88 standards, the approach in [14], defines partially some models of ISA-88 for batch production activities and scheduling. Note that all the previous works are restricted to specific cases of AMS's. Some examples of general modeling frameworks of AMS and product feasibility using ISA standards, were preliminary studied in previous work in [15, 16], but for the case of FSA only.

Inspired in [14, 16], this work proposes a modelling framework applied to any AMS based in the suggestions of the ISA standards. Therefore, the main contribution of this paper is the definition and interconnection of the generic models of equipment, tasks, storages and logic precedence, mentioned in the ISA-95, to construct complex models of AMS that describes equipment availability, storage capability, resources sharing, process restrictions, and others. The framework is more general than the one given in [14], avoiding the state explosion and the plant supervision of [15, 16]. To illustrate the PN modeling procedure, a complete example of a real AMS with the presence of industrial robots and process workstations is studied.

The paper is organized as follows. Section 2 summarizes the main concepts of PN. Section 3 explains the ISA-95 and ISA-88 standards. Section 4 presents the modelling framework. Section 5 shows the PN models and Section 6 addresses the case of study. Finally Section 7 presents some concluding remarks.

2. Petri Nets Definitions

Definition 1: Based in [1, 2], a PN with finite capacity is a weighted and bipartite graph given by a 5-tuple

$$PN= (P, T, F, W, M_0) \quad (1)$$

where:

- $P= \{p_1,p_2,..,p_m\}$and $T= \{t_1,t_2,\ldots t_n\}$ are the disjoint sets of nodes called *places* and *transitions*, respectively.
- $F \subseteq (P \times T) \cup (T \times P)$isthe set of arcs, connecting places to transitions and vice versa, with elements $w(p,t), w(t,p), \forall t \in T, \forall p \in P$.
- W:F$\rightarrow Z^+$is the function that assigns the weights to each arc.
- $M(p): P \rightarrow Z^+$, expressed as am-entry vector represents the number of tokens residing inside each place. LetM_0as the initial marking. Consider that each $p_i \in P$has a finite capacity c_i of tokens, i.e. $M(p_i) \leq c_i$.

The reachability$R(PN)$ is the set of all possible markings reachable from M_0. A k-thstate or marking in a PN, denoted by M_k is achieved according to the following transition rule:

Definition 2: A transition $t_i \in T$is said to be enabled in a PN with finite capacity if:

- $M_k(p_j) \geq w(p_j, t_i), \forall p_j | w(p_j, t_i) \in F$, with j=1,…,m and i=1,…,n.
- $w(t_i, p_j) + M_k(p_j) \leq c_j, \forall p_j \in P$.

If $t_i$is enabled in the k firing in some firing sequence, then defines the next marking

$$M_{k+1}(p_j) = M_k(p_j) - w(p_j, t_i) + w(t_i, p_j) \quad (2)$$

*Definition 3:*The incidence Matrix for PN with n transitions and m places is$A = [a_{ij}]$, an $n \times m$ matrix of integers and its typical entry is given by $a_{ij} = a_{ij}^+ - a_{ij}^-$ where $a_{ij}^+ = w(t_i, p_j)$ and $a_{ij}^- = w(p_j, t_i)$. The equation (2) can be also expressed as $M_{k+1} = M_k + A^T u_{k+1}$after de $k+1$th firing, where the control vector u_k is a $|u_k| \times 1$ vector of zeros except in entry i, that is equal to 1, where each element of u_k corresponding to a transition of the PN.

3. ISA-95 and ISA-88 Standards

The ISA-88 and ISA-95 standards provide the modelling framework to classify equipment and control procedures related to the manufacturing of products. Also, they establish the interface between management systems, coordination control and low-level control technologies [8]. The ISA-88 is related for batch control systems [7], however, the main concepts have been extended for other continuous and discrete processes [15]. As shown in Figure1, these standards propose to establish an Assets Model (AM) as the hierarchical-modular set of physical process components and a Procedural Control Model (PCM) including the control activities executed in the equipment to carry out process tasks and finally, the Process Model (PM) as the result of mixing the AM and PCM. Note that the PM is not related to a specific product and it only contains the generic tasks according to the equipment capabilities.

The recipes or Control Recipe Procedures (CRP's) describe with a certain degree of generality what equipment and procedures must operate for the manufacturing of one product or list of products. Next section describes the application of ISA-88 and ISA-95 for the modelling methodology of AMS modelling.

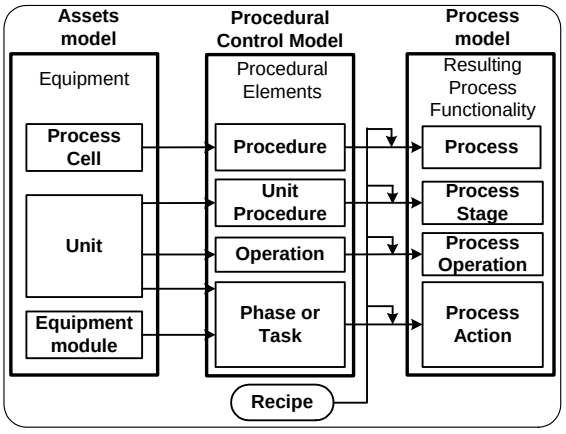

Figure 1. ISA-88 and ISA-95 Reference Model

4. Modelling Framework

According to the ISA-88 and ISA-95 standards, the general model is obtained from the definition and interconnection of basic models of sets of equipment (E), process tasks (PT), storage (A)and restriction of logical precedences (DL).

The equipments can be all kind of entities that are responsible to operate some process tasks, such as robots, conveyor, machines, people, or their combinations when they work together. The tasks consider all actions or jobs performed by equipments. So, the execution of every task requires the availability of its responsible equipment and probably, the use of material allocated in storages and the precedence of another tasks to be executed. The storages in AMS are commonly classified in three types: a) Manual load-Automatic unload (ML-AU), appeared as dispensers of raw material like entries of the AMS, b) Automatic load-Manual unload (AL-MU), that involves the storages of final product that exit of the AMS and c) Automatic load-Automatic unload (AL-AU), for intermediate storages of subparts of products placed temporally for the material-handling devices when the workstations are busy. Note that the storages can be assumed as initially full or empty.

The general model is constructed from the interconnection of the sets described below. Thus, the evolution of the tasks of AMS considers the resources and restrictions of the logical precedences between tasks. The designer could easily modify or assign more resources and logical precedences according to the equipment changes in AMS, preserving the PN structure.

In order to establish a formal definition of the PN models, consider:

- $E = \{E_1, \dots, E_n\}$, as the set of equipment where each $\mathrm{E_i}$, $i = 1, \dots, n$realizek_i tasks belonging the set $H(E_i) = \{T_{ij}\}, 0 \leq j \leq (k_i - 1)$.Consider each E_i let available in a$\mathrm{C(E_i)}$ quantity.
- PTas the set of all the process tasks, i.e. $PT = H(E_i) \cup \dots \cup H(E_n), i = 1, \dots, n$.Define sT_{ij} and fT_{ij} as the start and end of the task T_{ij}, respectively. All the start and end of tasks are included in the general sets$sT = \{sT_{ij}\}, \forall T_{ij} \in PT$ and $fT = \{fT_{ij}\}, \forall T_{ij} \in PT$.
- Let $A = \{A_1, \dots, A_h\}$the set of storages, where $C(A_\ell)$, with $\ell = 1, \dots, h$ is the capacity of the storage $\mathrm{A_\ell}$.Let $U_{in}(A_\ell)$, $U_{out}(A_\ell)$ be the sets of transitions for the load and unload of the storage A_ℓ.If $mT_{in} = \{mT_{in_1}, \dots, mT_{in_q}\}$

and $mT_{out} = \{mT_{out_1}, \ldots, mT_{out_n}\}$ are the sets of manual loads and manual unloads transitions, then

- $U_{in}(A_\ell) \subseteq mT_{in}, U_{out}(A_\ell) \subseteq sT$ for the case of ML-AU.
- $U_{in}(A_\ell) \subseteq fT, U_{out}(A_\ell) \subseteq mT_{out}$ for the case of AL-MU.
- $U_{in}(A_\ell) \subseteq fT, U_{out}(A_\ell) \subseteq sT$ for the case of AL-AU.

Note that the automatic load or unload of the storage is assumed when a process task of material-handling is executed.

– Let $DL(ij, km)$, be the logic precedences between the tasks T_{ij} and T_{km}, indicating that the task T_{ij} should be completed to enable the start of the task T_{km}. The precedences between any pair of tasks is suggested by the ISA-95. However, it can be extended for the case of the conjunction of many tasks, as treated below. Define DL as the set of all the possible logic precedences.

$(sT \times PT) \cup (PT \times fT) \cup (fT \times DL) \cup (DL \times sT)$

– $M_0 = [M_0(E), M_0(PT), M_0(A), M_0(DL)]$

The initial marking M_0 and finite capacity of places are described in the next section. Also, a major explanation about the classification of precedences is studied in Section 5d.

Note that the quantity of tokens contained in the places of equipments, storage conditioning the firing of the start of task. When a task is being executed, a token is collocated in its respective place. When a task has finished, the tokens of availability of its equipment and storages are returned and the conditions of precedences are activated for the beginning of subsequent tasks. The three types of storages are illustrated in Figure 2. Observe that only the possible arcs *F* defined in the equation 3, appears in the scheme of Figure 2.

In a main perspective, the network does not contain self-loop, is non-ordinary, it has finite capacity and bounded [1].The structure of the

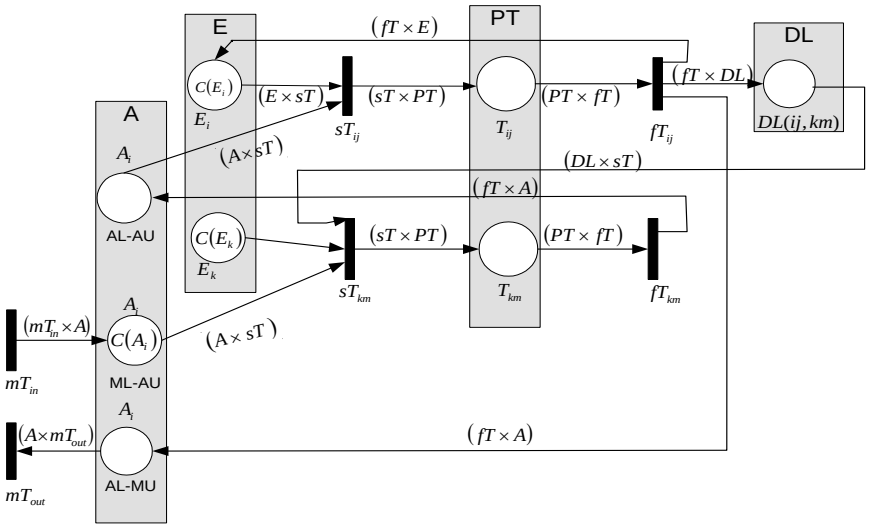

Figure 2. General PN model

Using the previous definitions, Figure 2 shows a scheme of the PN general structure proposed. Based in equation 1, the PN model proposed is given by

$$PN = (P, T, F, W, M_0) \quad (3)$$

where

– $P = E \cup PT \cup A \cup DL$

– $T = sT \cup fT \cup mT_{in} \cup mT_{out}$

– $F=(E \times sT) \cup (fT \times E) \cup (A \times sT) \cup (fT \times A) \cup (mT_{in} \times A) \cup (A \times mT_{out}) \cup$

PN has vivacity and reversibility guaranteeing the dynamical evolution of the AMS. In the next section, the basic models and a complete explanation about their construction is given.

5. Definition of PN Basic Models

5.1 Equipment model (E)

The Figure 3 shows the PN "equipment model" where a place is associated to each equipment. This equipment place contains initially the quantity $C(E_i)$ of tokens. Then, an arc is drawn

from the equipment place to each start transition of the tasks related toE_i.It establishes that an equipment device makes a task at a time. Similarly, when an end task transition is fired, a token is returned to the equipment place, allowing the resource availability again.

If the AMS increases the number of the same type of equipmentE_i, this modification involves only the addition of tokens in the place ofE_i. Therefore, the topology of the framework does not change and permits concurrency, i.e. different tasks of the same equipment E_i can be used at the same time using various the equipment resources. The arcs and their weights, capacity and initial marking of places are given by

Arcs and weight

- $(E \times sT) = \{(E_i, sT_{ij})\}, (fT \times E) = \{(fT_{ij}, E_i)\}$, $\forall T_{ij} \in H(E_i)$, $i=1,\ldots,n$, and $j=0,\ldots,(k_i-1)$.
- $w(E_i, sT_{ij}) = 1 \forall sT_{ij} \mid T_{ij} \in H(E_i)$.
- $w(fT_{ij}, E_i) = 1 \ \forall fT_{ij} \mid T_{ij} \in H(E_i)$.

Capacity and initial marking

- $K(E_i) = C(E_i), i = 1,..,n.$
- $M_0(E_i) = C(E_i), i = 1,..,n.$

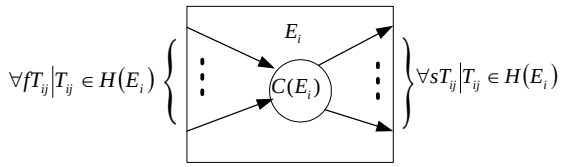

Figure 3. PN Model of equipment

5.2 Storage model (A)

Figure 4 shows the PN "storage model". This model is similar to the equipment model. The place associated to each storage, initially contains$M_0(A_\ell)$ tokens that represent the amount of raw material, parts, or final product in the storage. The storage capacity is given by $K(A_\ell)$, invalidating any new entry when the storage is full. Each transition belonging to $U_{out}(A_\ell)$ removes a token from the storage place, through an arc with the weight according to the number of material extracted from the task start or manual transition. The token will be returned with an arc drawing from the transitions that belong to $U_{in}(A_\ell)$to the storage place. Remember that the sets $U_{in}(A_\ell)$ and $U_{out}(A_\ell)$ could include the start or end of tasks or manual operation, depending if the storage is ML-AU, AL-MU or AL-AU.

The arcs and their weights, capacity and initial marking of places are given by

Arcs

- $(A \times sT) = \{(A_\ell, sT_{ij})\} \forall sT_{ij} \in U_{out}(A_\ell)$for ML-AU or AL-AU storages.
- $(fT \times A) = \{(fT_{rs}, A_\ell)\} \forall fT_{rs} \in U_{in}(A_\ell)$ for AL-MU or AL-AU storages.
- $(mT_{in} \times A) = \{(mT_{in_b}, A_\ell)\}$, $\forall mT_{in_b} \in U_{in}(A_\ell)$ for ML-AU.
- $(A \times mT_{out}) = \{(A_\ell, mT_{out_b})\} \forall mT_{out_b} \in U_{out}(A_\ell)$ for AL-MU storages.

Capacity and initial marking

- $K(A_\ell) = C(A_\ell)$.
- $M_0(A_\ell) = 0$, when the storage is initially empty and $M_0(A_\ell) = C(A_\ell)$ when is initially full.

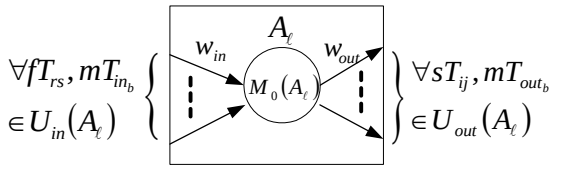

Figure 4. PN Model of storage

Note that the weights of input or output arcs contained in the sets w_{in} and $w_{out}$of the Figure 4, are chosen according to the number of material that arrives or leaves the storage.

5.3 Logic precedences (DL)

The logic precedences represent a correct functional flow of the tasks during the AMS execution. There exist two kinds of precedences: Direct logical precedences (DL-direct) and Inverse logical precedences (DL-inverse).

A DL-direct is established when, in the normal functional flow of the AMS operation, a previous task enables the beginning of a subsequent task. Figure 5ashows a simple graphical representation of a DL-direct between tasksT_{ij} and T_{rs}, where the boxes are tasks(left-side = start and right-side=end). The diagram represents that the *nf*amount of end-tasks of T_{ij} enables the *ns* starts of T_{rs}. The PN translation of the DL-direct is given in the Figure 5b. Note that the initial marking is equal to zero.

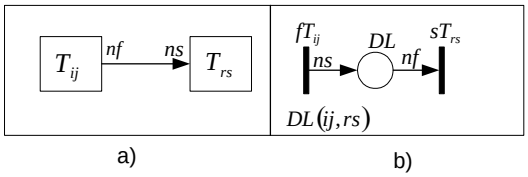

Figure 5. PN model of DL-direct

ADL-inverse occurs when the finish of a posterior task enables again the start of an initial task, in a normal functional flow of the AMS. For example, when the end of a lathe machining enables the feeding of a new part again. Figure 6a shows a DL-inverse, where the continuous line has been changed to a dotted line. Now, the *nf* occurrences of the end of the subsequent T_{rs} enable the start of the *ns* occurrences of $T_{ij.}$ Note that the tokens in the place of the DL-inverse is different to zero, because it is necessary to enable the begin of the task T_{ij} in a first-time of the functional flow. This quantity of tokens is given by α calculated according to the minimum tokens to enable the initial task including the quantity of the related equipment.

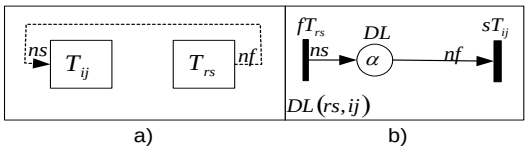

Figure 6. PN model of DL-inverse

If the DL-direct or DL-inverse presents the conjunction of different tasks, the models of Figures 5 and 6 can be extended to the models of the Figure 7.

The generalized models for DL-direct are presented in the rows 1a, 2a and 3a of the Figure 7. The row 1a shows how a previous task enables the beginning of one or ("∨") more subsequent tasks. The row 2a shows the start of one subsequent task after the finish of various tasks and finally, the row 3a presents the general case where multiple previous tasks enable multiple subsequent tasks. In the same way, the models of the rows 1b, 2b, 3b of Figure 7, address the same models but for DL-inverse precedences.

Table 1 summarizes the suggested initial marking, limited capacity and weights of the arcs of the DL-direct or DL-inverse precedences of the Figures 5, 6 and 7.Similar to

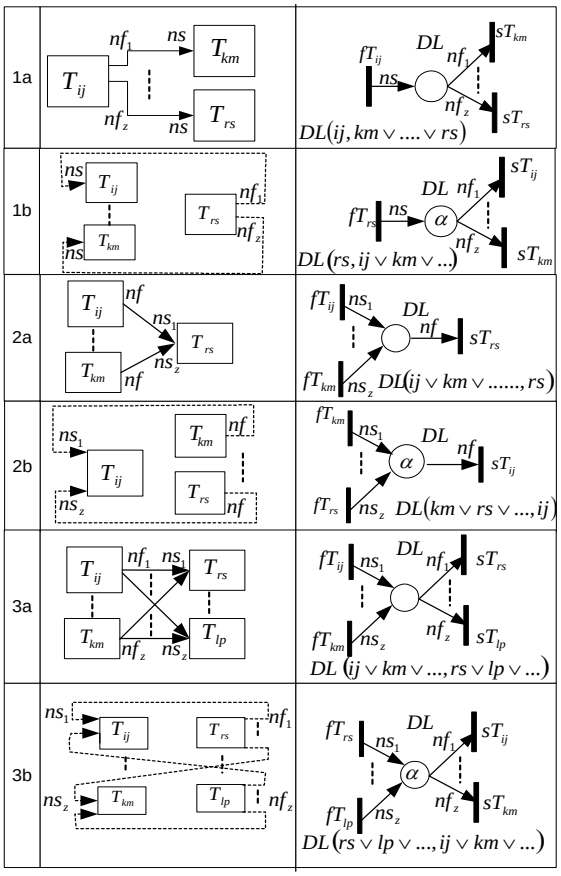

Figure 7. Logic Precedences for the case of multiple tasks

Table 1. M_0 (DL), K(DL) and weights for precedence

Logic Precedence	$M_0(DL)$		$K(DL)$	$w(fT_{ij},DL)$	$w(DL,sT_{ij})$
Directs		0	$C(E_i)$	ns	nf
	1a	0	$C(E_i)$	ns	nf_y
	2a	0	$\max(C(E_{i...k}))$	ns_x	nf
	3a	0	$\max(C(E_{i...k}))$	ns_x	nf_y
Inverses		$\alpha = ns*nf$		ns	nf
	1b	$\alpha = ns*\max(nf_y)$		ns	nf_y
	2b	$\alpha = \max(ns_x)*nf$		ns_x	nf
	3b	$\alpha = \max(ns_x)*\max(nf_y)$		ns_x	nf_y

the previous cases of Figures 5 and 6, the initial marking for a DL-direct is$M_0(DL) = 0$, but for a DL-inverse is $M_0(DL) \geq \alpha$,where $\alpha \geq 1$ is defined as the calculation of the minimum amount of tokens needed to enable a first-time the initial task. As shown in the Table 1, the calculation of α contemplates the multiplication of the maximum value of the weights of the arcs of subsequent tasks, denoted by the quantifier *max(●)*. Note that the capacity K(DL) is the same value of the initial marking for the case of DL-inverse precedences.

5.4 Process tasks (PT)

The process task is the central model of the framework. Figure 8 shows the PN model of a process task T_{ij}. A place is associated to each task (named task-place) attached to two transitions related to the start and end of this task. The start-task transition, denoted by sT_{ij}, depends of the available tokens from the equipment E_i, storages A_ℓ where $sT_{ij} \in U_{out}(A_\ell)$ and the logic precedences DL required to enable the start of the task T_{ij}.

- $w(sT_{ij}, T_{ij})=1, \forall sT_{ij} \mid T_{ij} \in PT.$
- $w(T_{ij}, fT_{ij}) = 1, \forall fT_{ij} \mid T_{ij} \in PT.$

Capacity and initial marking

- $K(T_{ij}) = C(E_i), i = 1,..,n.$
- $M_0(T_{ij}) = 0.$

Next section presents a complete case of study of an assembly AMS using the proposed methodology.

6. Case of Study

6.1 AMS Description

Figure 9 shows the AMS of the case of study. It can be divided in a machining station of subparts, matrix storage and an assembly station. The machining station includes two CNC machines, *Lathe* (*LT*) and *Mill* (*MI*) and a six-DOF *Gantry robot (GR)*that moves the raw material from an automatic *Hexagonal-shaped storage (HS)* to these two machines. The raw materials manually placed in the columns of the

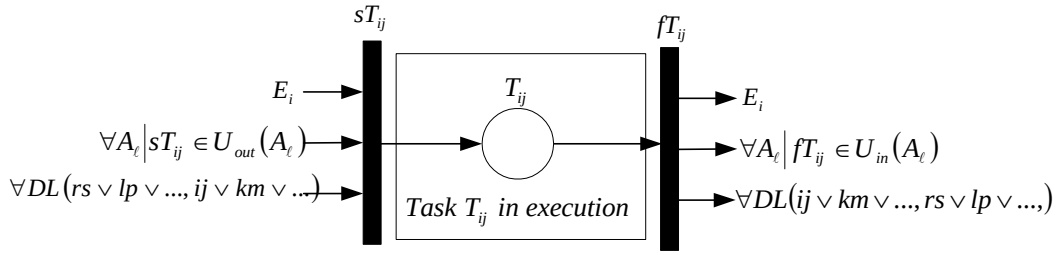

Figure 8. PN Model of a process task

When sT_{ij} is fired, a token is assigned in the task-place, indicating its execution. Once the task T_{ij} finishes, the end-task transition fT_{ij} is fired,it removes the token of the task-place and, as mentioned above, returns the tokens to the equipment E_i, the storages A_ℓ where $fT_{ij} \in U_{in}(A_\ell)$, and some precedences DL for the beginning of subsequent tasks. Note that the capacity of the task-place depends of the number of available equipments E_i for the execution of the task. So, the arcs and their weights, capacity and initial marking of the places are given by

Arcs and weight

- $(sT \times PT) = \{(sT_{ij}, T_{ij})\}, i=1,\ldots,n$ $j=0,\ldots,k_i\text{-}1, \forall sT_{ij} \mid T_{ij} \in PT.$
- $(PT \times fT) = \{(T_{ij}, fT_{ij})\} i = 1,..,nj=0,\ldots k_i\text{-}1, \forall fT_{ij} \mid T_{ij} \in PT.$

HS, each assigned previously for the LT or the MI. Note in the scheme of Figure 9 that the HS, MI, LT and the local controller of the GR are connected to a PLC1.

When the LT or MI have finished, the GR moves the machined parts to the *matrix storages (MS)*. The MS is a *6×7* array that contains a XY *Cartesian Robot (CR)*equipped with a pneumatic gripper that receives the parts directly from the GR and put the pieces in one of the slots. Note that every column is assigned to storage the pieces processed for each machine or the final assembly product. The presence sensors of the slots of the MS and the local controller of the CR are connected to a different PLC2.

Finally, when an assembly product is required, the subparts are moved from the MS using the CR and GR devices and placed in the point A of

the *Conveyor Belt* (*CB*), to be transported to point B. Points A and B are loading/unloading devices with pallet detection and the CB is one-way traffic. When the parts arrive to point B, they are introduced to the temporal buffers (BUF-LT and BUF-MI) in the assembly station. Once the necessary subparts are gathered, a six-DOF *Assembly Robot* (*AR*) with interchangeable grippers and pneumatic clamping mechanisms, together a group of pneumatic devices, assemble the final product. After that, this product is transported from point B to point A to be placed in the column of final product of the MS, trough GR and CR actions. It is assumed that the final product is taken out manually by a posterior process. The CB is controlled by a PLC3 (master PLC) and the elements of the assembly station and the local controller of the AR are connected to a PLC4.

Note that there is a distributed control of the AMS decomposed in four PLC's, as shown in Figure 10. A control computer has communication with the master PLC3 only, and the slaves PLC's are linked by a PROFIBUS-DP network. Each local controller of the robots (GR, CR and AR) is communicated to its respective PLC through the General Port of Input-Output (GPIO) as shown in Figure 10.

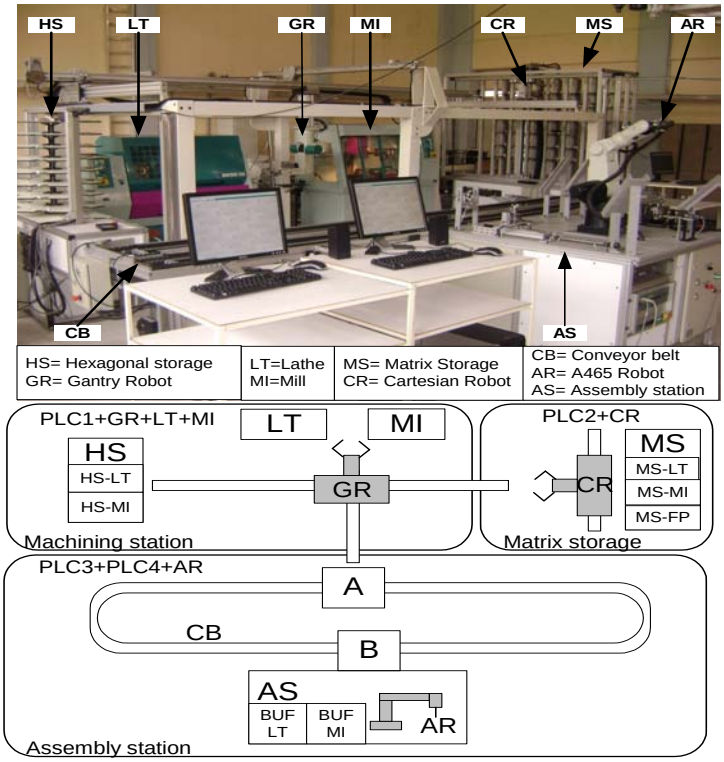

Figure 9. Photo and scheme of the AMS

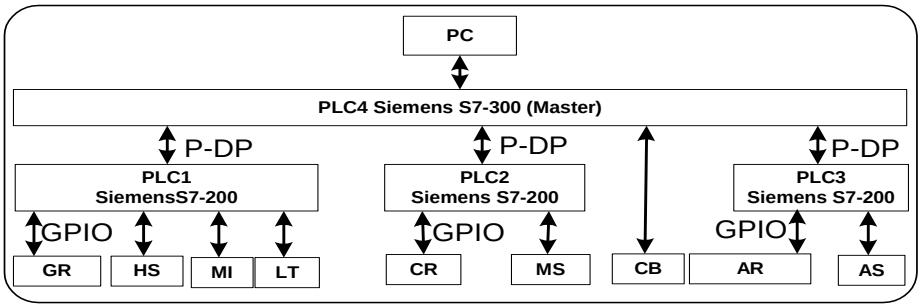

Figure 10. Controllers network

6.2 Tasks definition and logic precedences using ISA standards

Based on the industrial standard ISA-95 and ISA-88, an *Assets Model* and a *Procedural*

Control Model for the AMS is proposed according to the Figure 11. For the construction of the Assets Model, the main strategy is to separate the material-handling system from process workstations of the AMS as shown Figure 11. Note that due to the physical restrictions of functionality, some equipments work together always, therefore can be considered as a unique functional unit, for example GR and CR. Also, the AR appears twice because it performs as material-handling system, when moves the subparts in the temporal buffers in the assembly station, and also participates in the assembly of the products.

The Procedural Control Model is defined in the below part of Figure 11, where the process tasks are identified and assigned to each

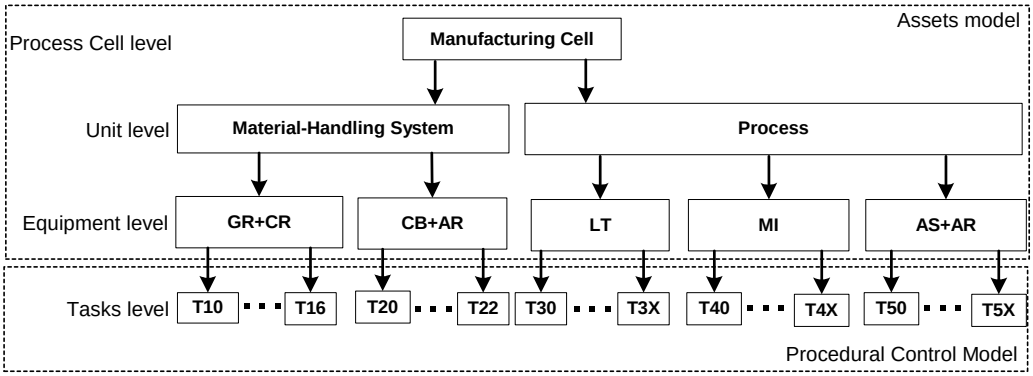

Figure 11. Task decomposition using the ISA standards

Table 2. Task decomposition according ISA standards

Manuf. Cell	Material-Handling System	E1= GR+CR	H(E1)	T10. GR transports from HS to LT T11. GR tansports from HS to MI
				T12. GR+CR transport from LT to MS-LT T13. GR+CR transport from MI to MS-MI
				T14. GR+CR transport a LT part from MS-LT to CB (point A) T15. GR+CR transport a MI part from MS-MI to CB (point A)
				T16. GR transports from CB (point A) to MS (FP column)
		E2= CB+AR	H(E2)	T20. CB+AR transport a LT part from CB (point A) to CB (point B) and local buffers of AS.
				T21. CB+AR transport a MI part from CB (point A) to CB (point B) and local buffers of AS.
				T22. CB transport FP from CB (point B) to CB (point A)
	Process	E3=LT	H(E3)	T30. Machining process of LT
		E4=MI	H(E4)	T40. Machining process of MI
		E5= AS+AR	H(E5)	T50. Assembly of the product MLLM

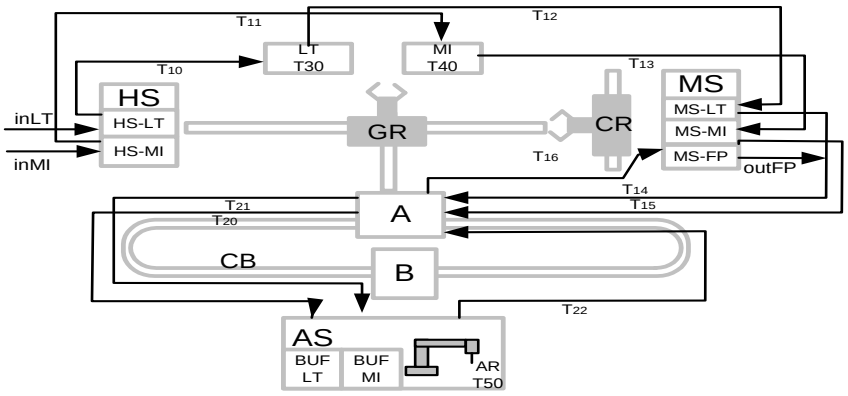

Figure 12. AMS scheme with trajectories of tasks

equipment of the AMS. Note that, for simplicity, it is considered only a task for LT and MI and a final assembly product composed by two parts manufactured by the LT and MI (denoted by MLLM). Also it is restricted that the CB transports a work piece at a time. The process tasks are briefly described in the Table 2. For a more clear explanation of all process tasks, Figure 12 presents graphically the trajectories of the tasks in the AMS.

but enables more concurrency of the tasks and the capacities of production of the AMS. In the top-down order of appearance of the logical precedences,

M_0(DL)={0,0,1,1,0,0,2,2,1,0,0,0,0}, **K(DL)=** {1,1,1,1,1,1,2,2,1,1,1,1,1},

These values were obtained from calculus of the Table 1.For a practical implementation of the control of the AMS, note that the task

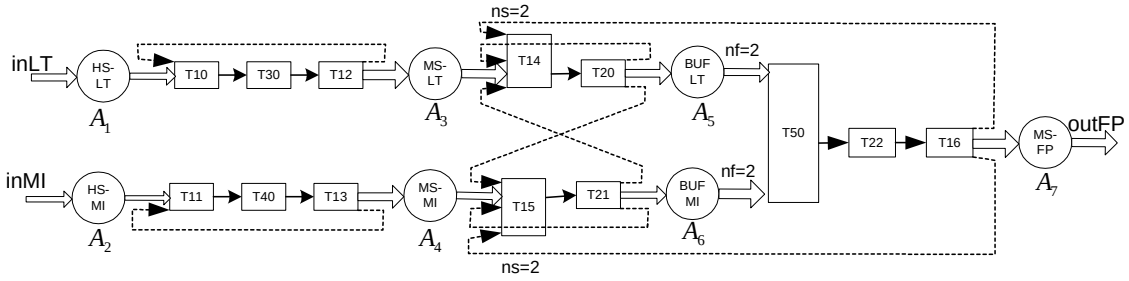

Figure 13. Diagram of precedences of the AMS

The logic precedences between tasks of the AMS are presented in Figure 13 using a Precedences Diagram suggested by the ISA-95. Note that Figure 13 does not refer to some product sequences; it only contains the relations of precedences between pairs of tasks and the storages due to process restrictions. For example, the Task T_{30} always begins only if the GR puts one work piece in it (finish of task T_{10}), and so on. Similar to the section 5C, the continuous lines represent DL-directs and the dotted lines represent DL-inverses, some of them include multi-tasks with the values of *nf* and *ns* explained before. The storages are included as circles and they appear between DL-direct precedences.

6.3 PN model of the AMS

Applying the PN models described in the section 5, we obtain 5 equipment models, 7 storages models, 13 process task models, 8 DL-direct and 5 DL-inverse precedences models Their interconnection is given in the general PN model of the Figure 14 using some connectors (hexagonal symbols) for a more clarity. Note that in the PN model, it is possible to visualize the places of storages and manuals events in the left side, the equipments and tasks in the middle side and the logical precedences in the right side of Figure 14. For the equipment models and storages: **M_0(E)**={1,1,1,1,1}, **K(E)**={1,1,1,1,1}, **M_0(A)** = {4,4,0,0,0,0,0}, **K(A)**={4,4,4,4,4,4,4}. Note that the change of these parameters does not involve any change in the structure of the PN, executions can be programmed in the controller network by communicating only the firing of the transitions to a high-level control, which could be programmed in the main computer based on the dynamical behaviour of the general PN model.

Comparing with the conventional PN modelling presented in [14], our approach considers more suggestions of the ISA standards including different types of storages, the interactions of storages and the transportation tasks, the analysis of single or multiple logical dependences and the addition of inverse logic precedences. Also, the modelling framework encompasses more real complex manufacturing systems than the simple example given in [14].

7. Conclusions

This work presents a methodology for the discrete-event modelling of AMS based on the task decomposition proposed by the industrial standards ISA-95 and ISA-88 and its translation to generic PN models. The interconnection of equipment, storages, process tasks and inter-tasks logical precedences composes a general PN model that describes the maximal concurrency obeying the process restrictions. The framework allows the possibility to modify the amount of equipment or storage limitations without changes in the network topology, preserving its static properties. Also, the task-based modelling gives a clear separation between the generic

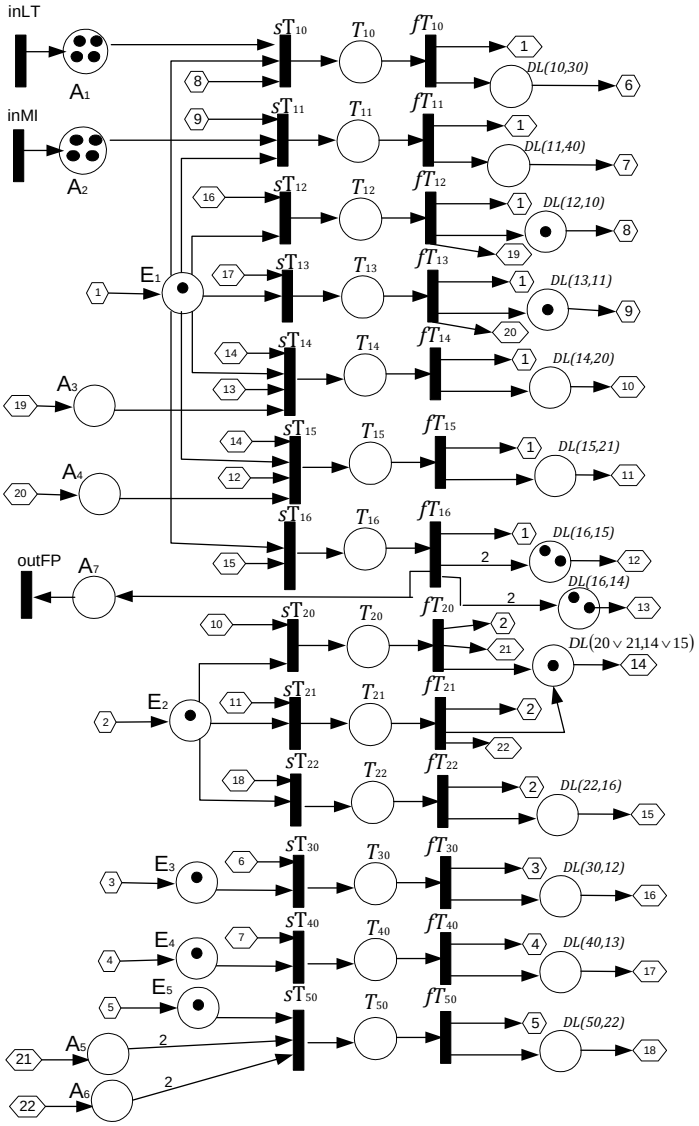

Figure 14. General PN model of the AMS

process tasks related to the equipment capacities and their interconnections for the manufacturing of different and concurrent products. The PN model provides the dynamical structure for a high-level coordination of the AMS in a control computer and the set up of the process tasks, which can be programmed in the local controllers. The modelling framework is a systematic method to close the concepts of Discrete-event systems in an engineering manufacturing context.

Acknowledgements

This work was partially supported by Universidad Iberoamericana, CONACyT (SNI scholarship) and IPN through research project 20130760, and scholarships EDI, COFAA, and PIFI.

REFERENCES

1. CASSANDRAS, C. G., S. LAFORTUNE, **Introduction to Discrete Event Systems**, Kluwer Academic, 2008.
2. MURATA, T., **Petri nets: Properties, Analysis and Applications**, Proceedings IEEE, vol. 77(4), 1989, pp. 541-580.
3. ZHUO, M.C. AND F. DICESARE, **Petri Net synthesis for Discrete Event Control of Manufacturing Systems**, Boston, MA. Kluwer,1993.
4. ESTRADA-VARGAS, A. P., J. LESAGE, E. LOPEZ-MELLADO, **Stepwise Identification of Automated Discrete Manufacturing Systems**, IEEE 16th Conf. on Emerging Technologies & Factory Automation (ETFA), 2011, pp. 1-8.
5. DE QUEIROZ, M. H., J. E. R. CURY, **Synthesis and Implementation of Local Modular Supervisory Control for a Manufacturing Cell**, Sixth International Workshop on Discrete Event Systems, 2002, pp. 377-382.
6. CASTILLO, I., J. S. SMITH, **Formal Modelling Methodologies for Control of Manufacturing Cells: Survey and Comparison**, Journal of Manufacturing Systems, vol. 21, no. 1, 2002, pp. 40-57.
7. INSTRUMENT SOCIETY OF AMERICA, **ISA-88.01 Batch Control Systems, Part 1. Models and Terminology**, ISA Standards, 1995.
8. INSTRUMENT SOCIETY OF AMERICA, **ISA-95.1. Enterprise-control System Integration, Part 1. Models and Terminology**, ISA Standards, 1999.
9. ZHAO, Z., G. ZHANG, Z. BING, **Scheduling Optimization for FMS based on Petri net Modeling and GA**, IEEE International Conference on Automation and Logistics (ICAL), 2011, pp. 422-427.
10. LEITÃO, P., J. M. MENDES, A. BEPPERLING, D. CACHAPA, A. W. COLOMBO, F. RESTIVO, **Integration of Virtual and Real Environments for Engineering Service-oriented Manufacturing Systems**, Journal of Intelligent Manufacturing vol. 23(6), 2012, pp. 2551-2563.
11. PETIN, J. F., D. GOUYON, G. MOREL, **Supervisory Synthesis for Product-driven Automation and its Application to a Flexible Assembly Cell**, Control Engineering Practice, vol. 15, no. 5, 2007, pp. 595-614.
12. RICKEN, M., **Modeling of Manufacturing Execution Systems: An Interdisciplinary Challenge**, IEEE Conference on Emerging Technologies and Factory Automation (ETFA), 2010.
13. DUTA, L., F. G. FILIP, **Control and Decision-making Process in Disassembling Used Electronic Products**, Studies in Informatics and Control, ISSN 1220-1766, vol. 17(1), 2008, pp. 17-26.
14. GRADIŠAR, D., G. MUŠIČ,**Petri Nets - Manufacturing and Computer Science,** In Tech, edited by Pawel Pawlewski, ISBN 978-953-51-0700-2, 2012, pp. 5-26.
15. NAVA, J. A., **Architecture for the Control of Continuous, Discrete and Batch Systems (in Spanish)**, Master Thesis, CINVESTAV, 2005.
16. SANCHEZ, A., E. ARANDA-BRICAIRE, F. JAIMES, E. HERNANDEZ ANDA. NAVA, **Synthesis of Product-driven Coordination Controllers for a Class of Discrete-event Manufacturing Systems**, Robotics and Computer-Integrated Manufacturing, vol. 26(4), 2010, pp 361-369.