5

Intelligent Production Systems Reconfiguration by Means of Petri Nets and the Supervisory Control Theory

Zapata M. Germán[1], Chacón R. Edgar[2] and Palacio B. Juan[3]
[1]Universidad Nacional de Colombia, sede Medellín
[2]Universidad de Los Andes, Mérida
[3]Universidad Nacional de Colombia, sede Medellín
[1,3]Colombia
[2]Venezuela

1. Introduction

The current trend in industrial production processes is to have agile and flexible systems that respond quickly to the permanent changes and disturbances in the production environment. This trend has created an important volume of research and papers aimed at having production control, supervision and programming systems that respond to these demands. Most of the proposals are grouped inside what has been labeled as Intelligent Manufacturing Systems (IMS). Among them are virtual, fractal, bionic, holonic manufacturing. These proposals initially appeared for discrete manufacturing processes. However, continuous production processes such as oil and gas, chemical plants, and power generation, also face demands for flexibility and rapid response. Therefore the IMS proposals can be applied to these types of processes.

Holonic Production Systems (HPS) is one of the proposals that has advanced the most. It already shows evidence of its application in industrial systems.

In general terms, a HPS is formed by autonomous entities that cooperate proactively to reach a common goal. These entities are labeled holons and, through aggregation relationships, they can form groups to form the so called holarchies. The grouping of various holons or holarchies with the objective of carrying out a productive process is called a Holonic Production Unit (HPU).

The principal attributes of holons are: autonomy, cooperation, proactiveness, and reactivity. Other key characteristics are the distribution of intelligence and self-similarity to build complex structures from simpler systems.

In order to achieve agility and flexibility, HPSs need distributed coordination and supervision functions. These functions enable them to dynamically reconfigure the production structure and the control laws to accommodate to the new operative conditions.

Reconfiguration can only be faced if the production system has flexibility with regards to the allocation of manufacturing operations and of control architectures, which enable using different control policies for different types of services, or adapting control strategies to achieve new requirements.

The selection of a new configuration is basically a problem of state reachability in discrete dynamics systems. Therefore, Petri nets and the supervisory control theory have been deemed appropriate to find solutions that perform well in real time. Due to the demands for temporary performance, reconfiguration has been considered a function of real time in control architecture.

The work that is presented is inspired in the holonic paradigm, and supported by Petri nets and the supervisory control theory, to define, in real time, a new configuration for the production structure that adjusts to new requirements or disturbances.

For the holonic paradigm perspective, each production resource is seen as a HPU, with skills, availability and capacities. The HPU makes offers and negotiates its objectives or missions. Each holon, for a determined operation condition, offers its services based on its current capacities and state and, in this manner, a highly reconfigurable system is formed.

Petri Nets (PN) are a mathematical and graphic tool with the ability to capture precedence relationships and structural relationships, to model blocking, sequences, concurrent processes, conflicts and restrictions.

Products manufactured by an HPU can also be modeled through PN, because of the ease to represent precedence relationships. A global HPU model is established through PN composition operations, composing the resources with the products. The initial marking of this global PN is generated from the state of the resources in real time which, in order to conduct a production mission, presents offers based on current operative conditions.

Once the initial marking for the resources state is established, the PN is executed and the complete state space of the possible HPU behavior is generated. The states space, or reachability tree, is an finite automata and the trajectories between states define the possible configurations to reach the objective. The configuration selected must be feasible and controllable and must also guarantee that there neither blocking nor forbidden states in the system. Besides, the selected trajectory must lead to a satisfactory termination of the product. The supervisory control theory (SCT) is suitable to solve this type of problems which are present in DES – Discrete Event Systems.

By guaranteeing properties as boundedness, a finite states space and the obtaining of a solution in finite time are assured.

Once a configuration to reach an objective, based on the capacities and the state of resources is selected, the holarchies are formed.

Generated for each holarchy, in a recursive manner, is a PN model of its behavior, following the same construction structure explained for the HPU. When a disturbance occurs, the holarchy tries to solve it internally by adjusting its production structure and following the proposed analysis technique. If the holarchy is incapable of achieving its mission for the new operative condition, it requests cooperation from the rest of the HPU. The new mission redistribution is carried out by following the same analysis technique, generating a global PN model formed by holarchies, and a marking for the current condition.

In this manner, a proposal for determining a new configuration, which shows the advantages of using the recursivity of the holonic paradigm and the description and analysis power of Petri nets, has been developed. Real time performance is noticeably improved as the reachability tree is considerably reduced.

This chapter is structured as follows. The first part presents related works highlighting research made from the holonic paradigm or making use of PN and SCT. Theoretical concepts are presented in the second part. The third part presents the construction of the HPU's global model and the obtaining of the reachability tree from the marking determined by resources states, and continuing with the definition of the holarchies. Once the holarchies are established, it is shown how the HPU responds to disturbances and how it reconfigures itself to cover the fault conditions. Finally, an application and implementation example is presented through CPNTools.

2. Previous works

It was only in 1989 that the concept of Holon proposed by Koestler in 1968 (Koestler, 1968), was applied to manufacturing concepts. Suda, in his work, proposes a "plug and play" proposal to design and operate manufacturing systems that combined optimum global performance with robustness to face disturbances. This gave birth to what is labeled Holonic Manufacturing Systems (HMS) (Suda, 1989). As a concept, HMS represents a methodology, tools and norms to design manufacturing control systems that are flexible and reconfigurable. A work that will always be a fixed reference, as it is a forerunning proposal, is the work presented by Jo Wyns in (Wyns, 1999), labeled PROSA: Product - Resource - Order – Staff Architecture. Concerning the control system's flexibility, PROSA, through the formation of temporary holarchies, establishes that mixed scenarios that combine hierarchies with heterarchies are the most optimal. Recently, PROSA began to add auto-organization capacities taken from biological systems and labeled its architecture PROSA + ANTS (Leitao et al., 2009).

ADACOR architecture (Adaptive Holonic Control Architecture) (Leitao, 2004), presents the holonic approach to introduce dynamic adaptation and agility to face disturbances. This architecture is based in a group of autonomous, intelligent and cooperative entities to represent Factory components. In this approach it is important to highlight the modeling of holon dynamics through Petri Nets.

Other architectures that can be highlighted include: Holobloc Fischer John (n.d.), Metamorph (Maturana et al., n.d.), Holonic Control Device (Brennan et al., 2003), Interrap (*Holonic Manufacturing Systems*, 2008), Holonic Component Based Architecture (Chirn & McFarlane, n.d.), HoMuCS (Langer, 1999), MaSHReC (El Kebbe, 2002) and HOMASCOW (Adam et al., n.d.).

The largest volume of research is centered around discrete manufacturing systems. In reality, there have only been a few works in which the holonic approach was applied in the industry of continuous processes. Nevertheless, these industries are also subject to the demands of today´s markets. Mass production personalization trends, rapid response times, shorter product life cycles, and the efficient use of energy and resources, force the process industry to consider aspects as flexibility, agility reconfigurability, decentralization and integration.

The works of (Chokshi & McFarlane, 2008b), can be cited as references of the holonic approach in continuous processes (Chokshi & McFarlane, n.d.) (McFarlane, 1995) (Agre et al., 1994) (Chacón & Colmenares, 2005). They mention that the principal works are conducted by groups aimed at automating the chemical industries, specially concerning for planning and production scheduling.

A group from Universidad de los Andes de Venezuela has proposed the concept of Holonic Production Unit- HPU for its acronym in Spanish - in which holonic principles are brought into continuous processes. Some of the works related are found in (Chacón et al., 2008) (Chacón & Colmenares, 2005) (Lobo, 2003) (Peréz, n.d.) (Durán, 2006) (Chacón & Colmenares, 2005) (Chacón et al., 2003).

The techniques of Discrete Even Systems (DES) have been used in some works related to holonic or similar systems - as multi-agent systems or distributed systems - to model dynamics, to model production control functions and to test/prove performance properties.
In (Caramihai, n.d.) a unified theory base is presented, based on the theory of DES, to model interactions between agents by composing Petri Nets (PNs). This method uses the Supervisory Control Theory (SCT) of DES through PNs, in the sense that all interactions among agents are obtained by analyzing the states space generated by the execution of the PN, and through the specification of undesired states such as blockings. A supervisory model eliminates the interactions that lead to those states.
For Celaya, the development of theoretical bases to guarantee the properties of a Multi-agent system (MAS) is critical. These systems can be considered as DES and use PNs to model interactions and to guarantee that the structural properties of these interactions are met. Blockings of the MAS are avoided, evaluating the properties of liveness and boundedness (Celaya et al., 2009).
According to Balasubramanian, control of holonic systems in real time requires responses that are radically different. These must adapt automatically and reconfigure according to the constantly changing requirements of the production system. It presents the architecture for dynamically reconfigurable systems based on IEC 61499 and in a structure of control levels driven by events. (Balasubramanian et al., n.d.)
A work by Hsieh (Hsieh, 2006) establishes a fusion of PNs with Contract Net, because of the difficulty of this protocol to avoid undesired states, such as blockings in the HMS. The modeling and analysis capacities of PNs with Contract Net are combined for the distribution of tasks among holons and the so-called collaborative Petri Nets are proposed. The complete process of production orders is proposed as a problem of supervisory control as different orders must compete for limited resources. These generate conflict situations that need to be solved through the coordination of PNs. The condition of liveness (non blocking) must be guaranteed in order to facilitate the feasibility of agreements.
The problem of configurations for scheduling and rescheduling has been deeply explored in (Ramos, n.d.), (Sousa & Ramos, 1998), (Bongaerts, 1998) and (Cheng et al., 2004). In (McFarlane, 1995), a Holon Configuration, which acts to reconfigure the system when required, is proposed.
A review of the works concerning distributed planning and scheduling of reconfigurable holons can be consulted at (Mcfarlane & Bussman, 2000), (Chokshi & McFarlane, 2008a) and (Chokshi & McFarlane, 2008b). These propose a distributed architecture, based on the holonic paradigm, for the reconfigurable control of operations in continuous processes. This approach distributes the functionalities of planning, scheduling, coordination and control.
PNs have gained the spotlight as a solution for the problem of scheduling and rescheduling in discrete and in continuous and batch processes (Tuncel & Bayhan, 2007), (http://research.curtin.edu.au/, 2010), (Fu-Shiung, n.d.), (Ghaeli et al., 2005), (Tittus & Akesson, 1999), (Tittus & Lennartson, 1997), (Zhou, 1995) and (Tittus et al., n.d.).
According to (Tuncel & Bayhan, 2007), PN based methods directly describe the current dynamic behavior and the control system's logic. The most significant advantage of using PNs, is their ability to capture precedence and structural relationships, and to model blockings, conflicts, buffer sizes and multi-resource restrictions. Concurrent and asynchronous activities, shared resources, route flexibility, limited buffers and complex restrictions in the process sequences can be, specifically and concisely, modeled through PNs.

Furthermore, according to (Music & D., 1998), synthesis methods based on PNs exploit the nets' structures avoiding the need to explore all the states spaces.
The characteristics of PN models have lead many authors to propose reconfiguration as a problem of supervisory control (Ramadge & Wonham, 1989), (Akesson, 2002), (Pétin et al., 2007) and (Tittus & Akesson, 1999). From this theory, a feasible combination of the resources in the time to reach an objective must be found. Therefore, a minimal restrictive supervisor that makes the system synchronize the optimal use of the resources by the products is synthesized. This guarantees behavior specifications and the reachability of the objective, preventing the system from going into blocked states or into forbidden states. Solutions based on this approach, that elaborate a model composed by products and resources through PNs and develop a synthesis from the supervisory control theory through an analysis of the reachability tree, can be found in (Tittus & Akesson, 1999), (Lennartson, Tittus & Fabian, n.d.), (Akesson, 2002), (Pétin et al., 2007), (Falkman et al., 2009), (Music & D., 1998), (Reveliotis, 1999) and (Lennartson, Fabian & Falkman, n.d.) . This last paper established that the supervisor must guarantee that the system remains alive (enabled to reach states where all products are produced) and safe (it never reaches undesired states).
Production scheduling based on Petri Nets and in the theory of supervisory control, along with the holonic concept of the formation of holarchies, can be appropriate for the rescheduling of continuous production systems with high demands for re-configurability.

3. Theoretical bases

3.1 Holonic Production Systems (HPS)

An HPS is understood as a system composed by individual autonomous units that cooperate proactively through temporary reconfigurable hierarchies to obtain a global goal. Each autonomous component is called a *Holon* and a group of holons is called a *Holarchy*. The aggregation of holons and holarchies, through properties of auto similarity, enables the construction of very complex systems as shown in 1. The added holons are defined as a set of grouped holons, forming a major holon with its own identity and a structure that is similar to the holons that form it.

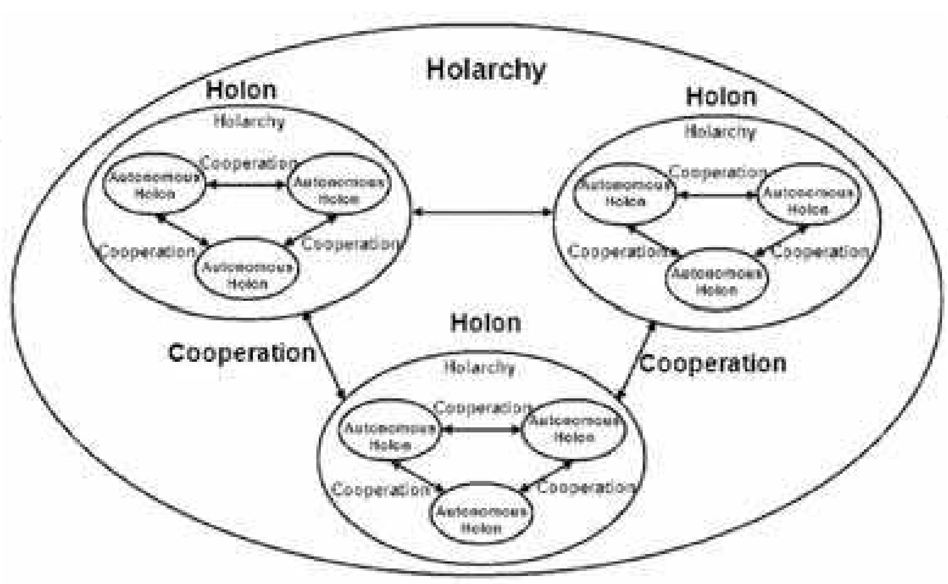


Fig. 1. Holarchy

The Consortium HMS defines a holon as: "a constitutive, autonomous and cooperative component of a manufacturing system whose purpose is to transport, transform, store, and/or validate information or physical objects"(HMS, 2004).
The principal attributes of a holon are: decentralization for taking decisions, autonomy, cooperation, and capacity to auto-organize. These enable a dynamic evolution and the reconfiguration of the organizational control structure, which in turn combine global optimization of production and agile reaction, to face unpredictable disturbances. Holons are also systems with objective oriented behaviors. HMS Consortium defined the following basic attributes:

Autonomy: The capability of an entity to create, control, and supervise the execution of its own behavior plans and/or strategies.
Cooperation: a set of entities that develop mutually acceptable plans and strategies and execute them in order to achieve common goals or objectives.
Proactiviness: the capability to anticipate changes in its plans and objectives.
Reactivity: the capability to react to stimuli in the environment.

In order to bring holonic attributes and characteristics into continuous production systems had been presented the Holonic Production Unit proposal (HPU).
Presented in this proposal, from an integrated vision of the productive process, is the conception of the production unit as a holon. Representation techniques must enable the understanding of relationships among them, for each one of the different components: graphic representations, capability attributes, availability, reliability and evolution of dynamics.
The HPU is conceived as the composition of a set of elemental units or resources that are organized and configured in a manner that enables performing the transformation processes in the value chain. The objective is to obtain the demanded products. The HPU takes its own decisions with regards to achieving its objective, but is obliged to inform its state in the compliance of a goal, or if such goal cannot be accomplished due to a fault or to errors in its behavior.
The HPU is formed by the following components:

- Mission, which describes the objective of production.
- Resources, which describe the necessary components for obtaining the products, which in turn can also be another HPU being consequent with the paradigm´s recursiveness.
- Engineering, which describes the necessary knowledge to obtain the products.

The relationship among these components is shown in Figure 2. Figure 3 shows the class diagram of the HPU proposal (Chacón et al., 2008).
Highlighted in this proposal are a component of Control and Supervision that measures, modifies, controls and supervises the production process; and the relationships that exist between the Configuration and the Production Method, and between the Process and the Resources.
The behavior of the HPU is presented in (Chacón & Colmenares, 2005) from the Supervisory Control Theory (SCT), describing its dynamics as a Discrete Event System (DES). The global behavior of the HPU is the result of coupling the behavior of the control system (supervisor) and the behavior of the process and the resources. Each component of a HPU is model through a Petri Net (PN).
In this manner, the behavior of a HPU focused in reaching a production goal, can be formulated as a supervisory control problem. In such, an entity (supervisor) restricts the behavior of a system so it achieves the desired states and does not go into forbidden states.

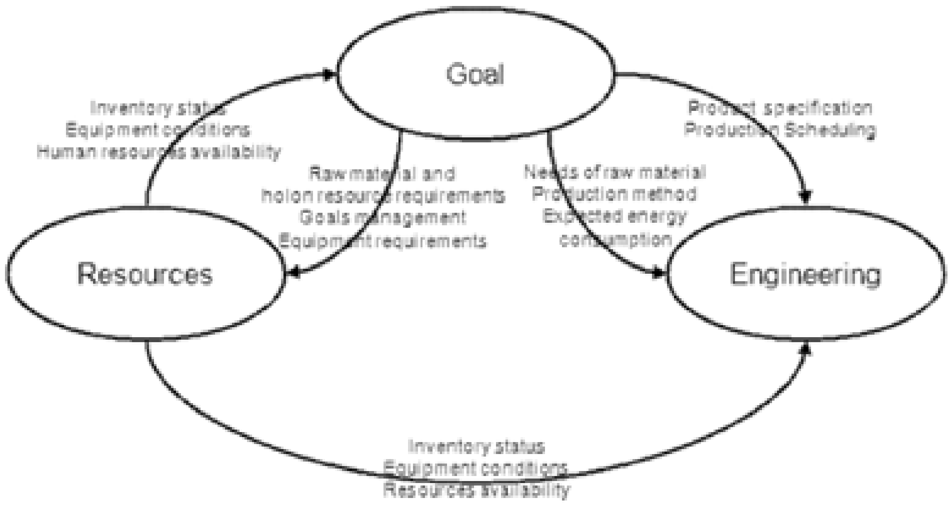


Fig. 2. HPU Components

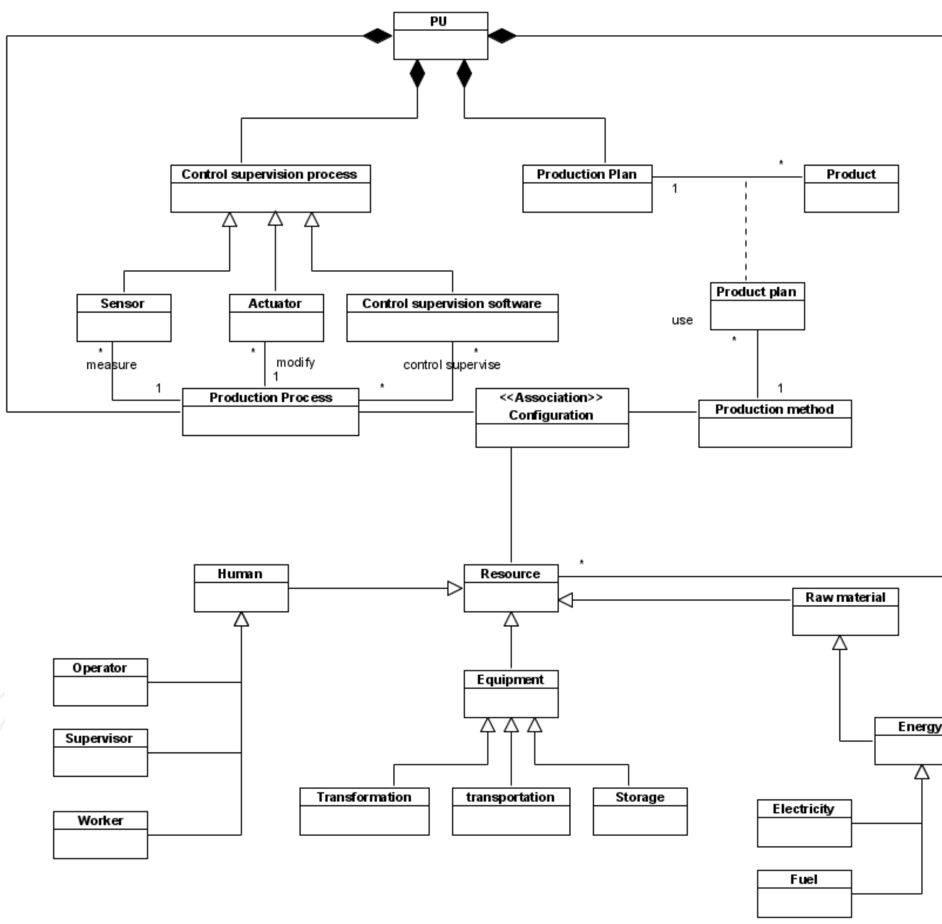


Fig. 3. Holonic Production Unit (From (Chacón et al., 2008))

3.2 Automata and languages

A Discrete Event System (DES) is defined as a dynamic system with a states spaces that is finite and numerable and that evolves because of the occurrence of spontaneous events. The decisions to "select" a configuration of the production resources to obtain a product in a continuous process, is classified in the DES category.

To specify a DES model, it is necessary to specify the set of states, the set of events and the structure of the system's transition. Formally, a DES is represented through an Automata $G = (Q,\Sigma, \delta, q_0,Q_m)$, where Q is the finite set of states; Σ is the finite set of events formed by two sub-sets: Σ_{nc} of non-controllable events and Σ_c of controllable events; a transition function $\delta: \Sigma \times Q \rightarrow Q$ that establishes the dynamic of the system; an initial state $q_0 \in Q$; and a set $Q_m \in Q$ of final or marked states. In addition, a set of co-reachable states Q_{co}, is described as the set formed by q states for which there is at least one path to take them from q to a marked state.
The behavior of a DES is characterized by a sequence of events produced during its operation. An event is called a string and is formed through the concatenation of events. Kleene closure is a set defined as the set of all strings formed through the concatenation of the elements of the set in any combination, including the identity element for the concatenation operator denoted ϵ and called the silent event. Kleene closure of the set of events is denoted by Σ^*.
A prefix of a string s is a sequence of events, which is also an initial sequence of s. This is $s \in\Sigma^*$, u is a prefix of s if $uw = s$ for some $w \in \Sigma^*$.
The set that includes all the prefixes of all its elements is said to be a prefix-closed. It is clear that Σ^* is a prefix-closed. The prefix closure of a set A is defined, and $\overline{A}$ is denoted as the set that contains all the prefixes of the elements of A.
A language is a set of strings (or words) formed by the concatenation of events. The language generated by automata G, denoted by $L(G)$, is the set of strings $L(G) = \{s \mid s \in \Sigma^*, \delta(s, q_0) \textit{ is defined}\}$
The marked language, denoted by $L_m(G)$,is a subset of $L(G)$ formed by all strings that lead to marked states. Formally: $L_m(G) = \{s \mid s \in \Sigma^*, \delta(s, q_0) \textit{ is defined}, \delta(s, q_0) \in Q_m\}$

3.3 Supervisory control theory

The results of the theory of automata and languages were used by Ramadge and Wonham (Ramadge & Wonham, 1989) to propose a theory for DES control called Supervisory Control Theory (SCT). In this theory, a supervisor controls the behavior of an automata representing the plant - enabling and disabling events - affecting the actual sequence and a trajectory to reach the desired states and avoid the forbidden ones.
To synthesize a supervisor, one departs from a language k called Specification which expresses the plant's desired behavior. The supervisor must guarantee that all marked states are reached from any reachable state. This property is called nonblocking and is defined by expressing that an automata is reachable and nonblocking if it is capable of reaching a marked state from any reachable state. It is formally described as: $\overline{L_m(G)} = L(G)$.
Furthermore, controllable specifications with regards to the plant must be guaranteed. Therefore, to verify this property, the following must apply: $\overline{k}\,\Sigma_{nc} \cap L(G) \subseteq \overline{k}$.
The nonblocking property is fundamental for the reconfiguration of a production system as, when a disturbance occurs and the failed system abandons a marked state, the control system must be able to bring it from the new state to another state $q \in Q_m$.
Along these lines, algorithms for synthesis of the DES supervisor are used in this work to establish a trajectory that leads the system to satisfactory termination states of the product when a disturbance occurs.
In synthesis, from a global behavior of the system, the forbidden states are removed through a purge function, the states that led to blockings are suppressed, and controllability is

verified. It is expressed in the following algorithm where A is an automata, M_A is the set of marked states of A, X_A is the set of forbidden states of A, Q_x is the set of blockable states plus the previous forbidden states.

purge(S_0);
$i = 1$;
repeat
    $Q_{CO} = M_A$; $Q_A = X$; $Q = \varnothing$;
    repeat
        $Q = 0 \ \forall \ Q \in [Q_A \setminus (Q_{co} \cup Q_x)]$
        IF $[Q_A \setminus (Q_{co} \cup Q_x)] = \varnothing$ **THEN**
        END
        ELSE
            $\exists \sigma \in \Sigma_A / \{[\delta(q,\sigma) \in Q_x]$ or $[\forall \delta(q,\sigma)$ not defined$]\}$
            $Q_x = Q_x \cup \{q\}$;
            $Q_{co} = Q_{co} \cup Q$;
        END IF
    UNTIL $Q = \varnothing$;
    $Q_{si} = Q_{co} \setminus Q_x$;
    REPEAT
        $X_A = Q_A \setminus Q_{co}$;
        $Q_x = X_A$; $Q = \varnothing$;
        $\forall q \in Q_A \setminus Q_x$ **IF** $\exists \sigma \in \Sigma_{nc} / \delta(q,\sigma) \in Q_x$ **THEN**
            $Q = Q \cup \{q\}$;
            $Q_x = Q_x \cup Q$;
    UNTIL $Q = \varnothing$;
    $X_A = Q_x$;
    $Q_{si+1} = Q_{co} \setminus X_A$;
    $i = i + 1$;
UNTIL $Q_{si} = Q_{si+1}$;
$Q_s = Q_{si}$ //those are states of Supervisor

3.4 Petri nets as language generators

Petri Nets relate to the theory of supervisory control theory through automata and languages. The most appropriate PNs for language generation are the ones called labeled PN. Through attaining all the states spaces of the PN labeled, called reachability tree or graph, the net's automata and the languages generated and marked are obtained.

A Petri net can be presented through a tuple of the form $R = \langle P,T,A,B \rangle$, where P is the set of states of the system, T is the transitions set, A is the input incidence matrix, that represent arcs from a p_i to a transition t_j with weight $a(p_i, t_j) \in \mathbb{N}$, B output incidence matrix, that represents arcs from t_j to p_i with weight $b(p_i, t_j)$.

A labeled PN is a tuple $N = \langle P,T, F, l \rangle$, where P and T have the same meaning as in the previous case, $F \subseteq (P \times T) \cup (T \times P)$ is the set or arcs and l is a label that assigns to each transition an event $l : T \rightarrow 2^{\Sigma} \cup \epsilon$.

A marking vector is $M : P \to \mathbb{N} \cup \{0\}$ which assigns to each place a non-negative number of tokens. The net (N, M_0) is a net marked con initial marking M_0. Notation $M_0[\sigma\rangle M$ is used to express that the trigger of σ in the marking M_0 leads to M.
To convert a PN to an automata, it is used the procedure that is shown bellow
Formally, the equivalent automata of a net (N,M_0) can be expressed as $G = (Q,\Sigma, \delta, q_0,Q_m)$, with $Q= R(N,M_0)$, meaning all reachable states in the Petri net N from M_0; $\Sigma=U_{t\in T}l(t)\setminus\{\epsilon\}$, δ so that $M' \in \delta(M,\sigma)$ iff $M[t\rangle M'$ and $\sigma \in l(t)$, $q_0 = M_0$, obtaining thereby the reachability graph of (N,M_0) expressed as automata.
Now, the link between Petri nets and languages it is given by marked or closed language ($L_m(G)$)and generated lenguage ($L(G)$).

$$L_m(G) = \{l(\sigma) \in \Sigma^* \mid \sigma \in T^*, M_0[\sigma\rangle M \in Q_m\} \text{ and } L(G) = \{l(\sigma) \in \Sigma^* \mid \sigma \in T^*, \exists M' \ni M_0[\sigma\rangle M'\}$$

Both lenguages can be found since the state space and that is way how PN, automata and lenguages have relation. It is had the PN behavior such an automata so it is posible to use Supervisory control theory (SCT) to restrict the system. In order to have finite and non blockeable graph, PN have to satisfy boundedness and liveness properties. However, by the re-scheduling characteristic of this proposal, PN have to be safe. The Net (N,M_0) is:

- Bounded, if there exist a non negative integer k such that $M(p) \leq k$ for every place p and for every marking $M \times R(N,M_0)$.
- Safe, if $M(p) \leq 1$ for every place p and for every marking $M \times R(N,M_0)$.
- Live, if for every marking $M \times R(N,M_0)$ there exist a firing sequence containing all transitions.

3.5 Composition through fusion places

The idea with composition methods is that from modular models, represented by automatas or PNs, a global model is determined without change the system´s dynamic evolution and, in this manner, the reachability tree of the whole system is obtained.
The fusion places has been proposed by Silva (n.d.) as a simplification method and by Jensen (2003) as a method for the hierarchical PNs.
In Jensen's proposal the idea with fusion is that places that undergo fusion represent the same place, even if they are represented as individual places. For instance, in figure 4(a) there are two modular models in which places labeled as **A** belong to the same fusion set, meaning that they are the same place and the model of figure 4(b) can be composed from them.

4. Description of the reconfiguration proposal

The proposal for the reconfiguration of continuous production systems uses models obtained through Petri Nets (PN), that combine holonic production resources with the products that they are in capacity to produce. The products are divided into operations and if a holon resource (HR) is able to perform an operation, then such holon is said to have the *skill* for this operation. A global PN of the Holon Production Unit (HPU) is built through models of the products and the HRs. When the HPU receives a mission, expressed as a production order, a negotiation process with the holon resources that form it is launched. If an HR has the skills to perform the operation for which the proposal is sent, it sends an offer that includes its availability, capacity and the cost of the operation. This enables determining the initial marking of the PN.

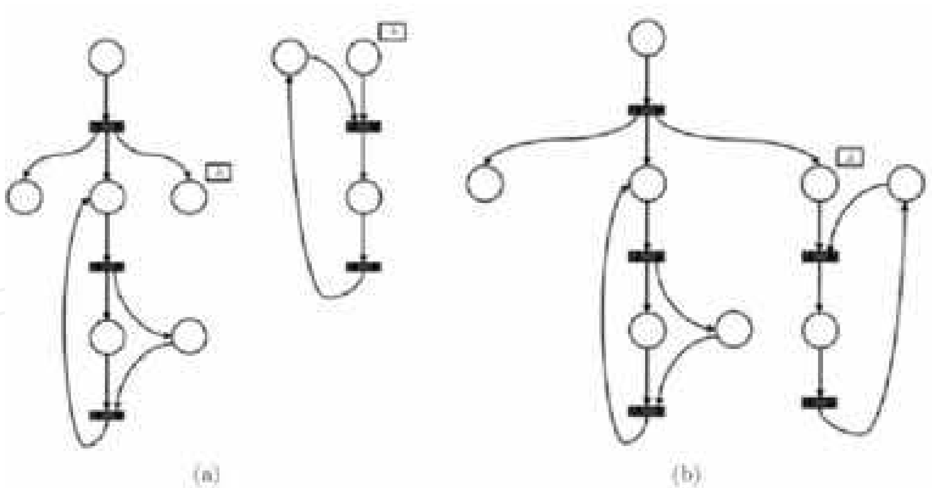


Fig. 4. Fusion Places

Established in the first phase of the negotiation is the existence of configurations in the HPU to develop the product, and the feasibility of the mission from the perspective of configuration. Configurations are obtained from a reachability tree of the PN, applying the supervisor synthesis algorithm that leads to satisfactory terminations of the product. If there are several possible configurations, the criteria of optimization are used to select the definite configuration of the HPU for the mission under negotiation. This phase of the negotiation is conducted out of line and it enables forming the holarchies that will be responsible for the production objective. A PN is generated for each holarchy following the same construction principles applied for the HPU. This evidences the recursivity of the holonic paradigm.

Once production is launched and disturbances appear response mechanisms are launched based on the attributes of autonomy and cooperation between holarchies. PN models are executed in each one of the cooperation levels in order to determine the new configuration that enables continuing with mission compliance despite the failure situation.

4.1 PN model construction

4.1.1 Product model

Construction of the global PN is based on the product´s model. There will be a model for each product made by the HPU, and each product will be specified independently of any production system. The required operational sequences required to obtain the product, are obtained in the model. P-Graphs proposed by Fiedler (Chokshi & McFarlane, 2008b), which are bigraphs used to model net structures, are used for the representation of products in continuous production processes. The graph´s vertices represent operations and raw material, and the connections represent material flow (see Figure 5).

The product’s discrete dynamic is considered to determine a configuration. Its continuous dynamic is not used for this. This means that to obtain a product, the availability of raw material, and of the process node that performs the operation, are necessary. The model does not consider flows of mass and energy, but only the necessary conditions to produce them. Process nodes represent transformation operations as "heat", "cool", "mix", "separate", "pressurize" and these operations indicate the skills demanded from the resources that must perform them.

Product representation through a PN is made following the structure of figure 6, which shows the raw material, the operation required (resource skill) and the product obtained.

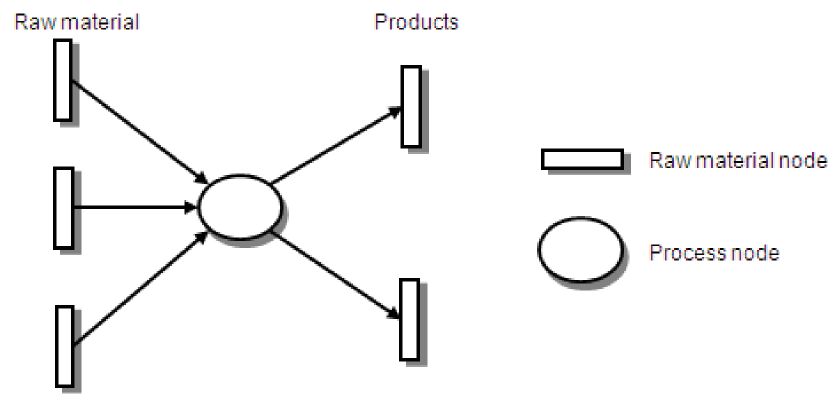


Fig. 5. Process graphs (P-graphs)

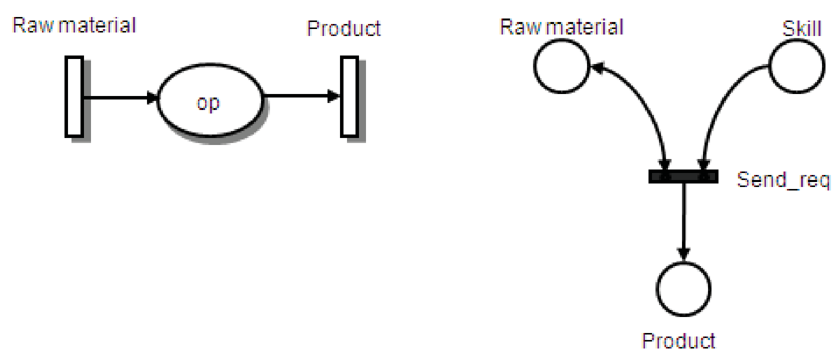


Fig. 6. PN product model

Example: the production line shown in Figure 7 enables obtaining a product with a model represented in Figure 8. The PN model of the discrete dynamic of the product is shown in Figure 9.

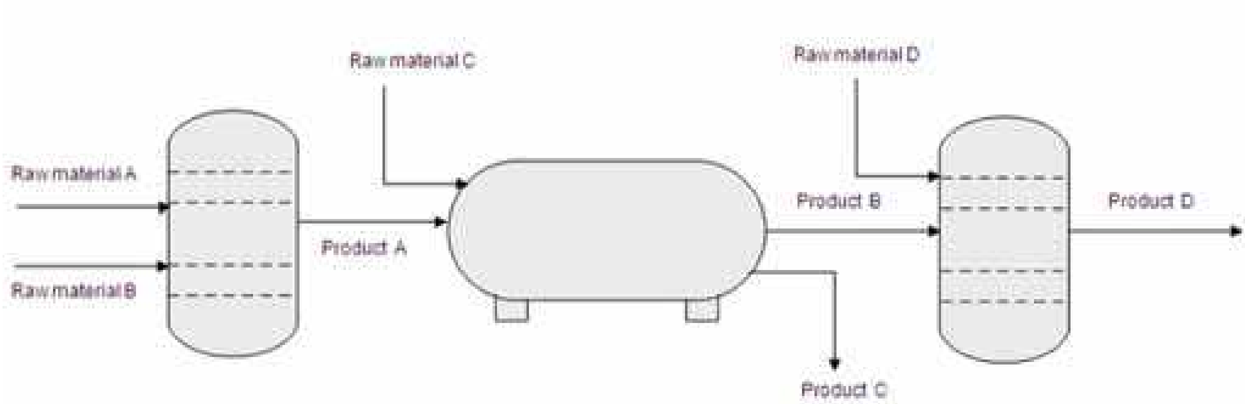


Fig. 7. Production line

4.1.2 Model of a Holon Resource

It is necessary to know the availability and skills of the HR to determine its configuration. Availability enables establishing the initial marking of the PN, and skills enable the link with the product's model.

Determined from the Petri Net of the HR the availability of resource (if it is on operation, or if it presents a failure or if it is on maintenance). Figure 10 shows the model and Table 1 presents the states and the events.

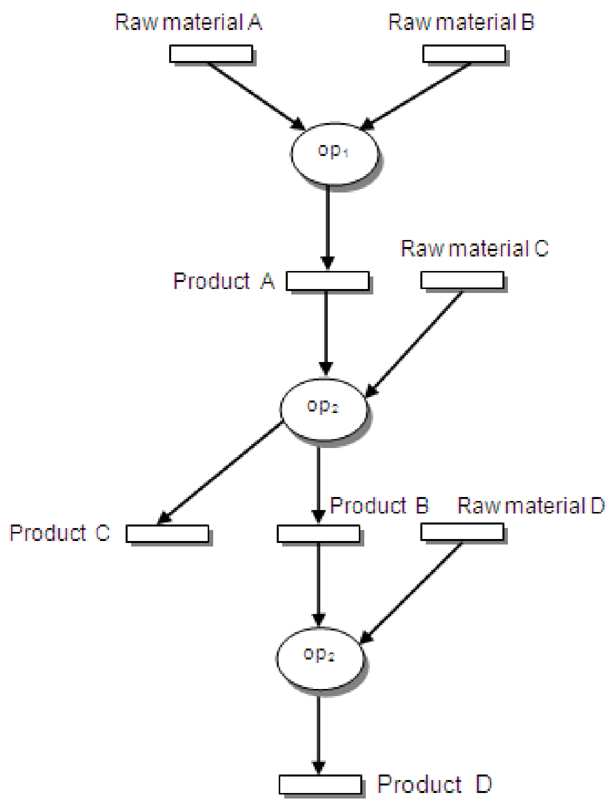


Fig. 8. P-graph of the production line

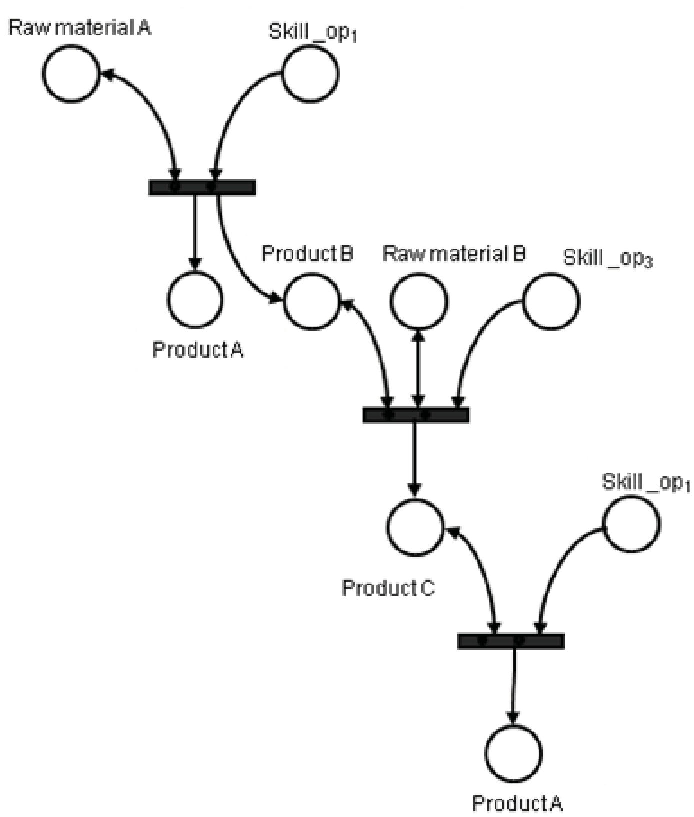


Fig. 9. PN of the production line

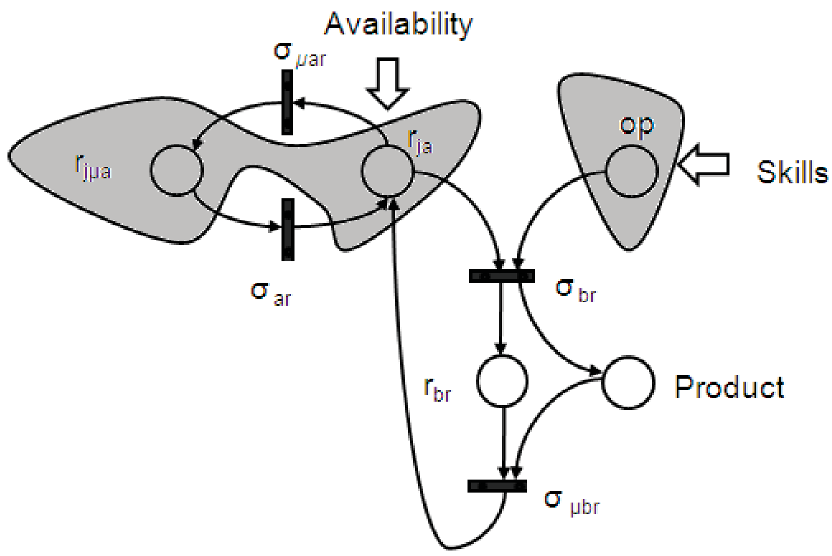


Fig. 10. PN Resource Model

States	**Description**
r_{ja}	resource *j* available
r_{jua}	resource *j* unavailable
op	skills
r_{jb}	resource *j* booked
Events	**Description**
σ_{nar}	non-availability
σ_{ar}	availability
σ_{br}	booked
σ_{ubr}	unbooked

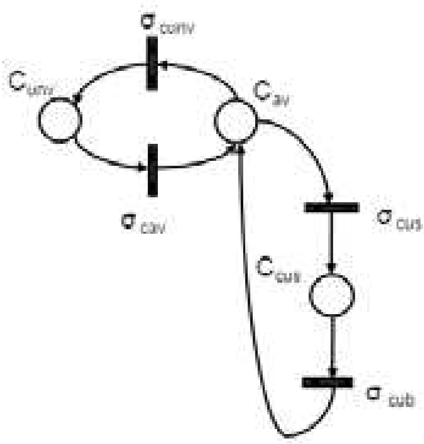
Table 1. Resource model States and events description

4.1.3 Connections Model

Connections guarantee the flow of a product between resources and their availability, or lack of, affects the definition of the configuration. Connection models are obtained based on the operational and physical restrictions imposed by the process. Each resource has input and output ports. The PN for the connections is shown in Figure 11.

- **Asynchronous Junction (Multi-product):** enables the distribution, simultaneous or individual, of a product to two destinations. See figure 12
- **Synchronous Junction (Separation):** enables material flow only to one destination at a time. See figure 13
- **Asynchronous Union (Multi-feeding):** receives material flow from two sources, simultaneously or individually. See figure 14
- **Synchronous Union (Mix):** receives material flow from just one source at a time. See figure 15

Taken into account to connect resources is that an upstream resource enables the output port, and that connection is an additional condition to enable a downstream resource. The PN showing two resources connected is given in Figure 16.



States	Description
C_{av}	Connection available.
C_{un}	Connection unvailable.
C_{cus}	Connection used.
Events	**Description**
σ_{cav}	Available.
σ_{cus}	Used.
σ_{cun}	Unbooked.

Fig. 11. Connections model

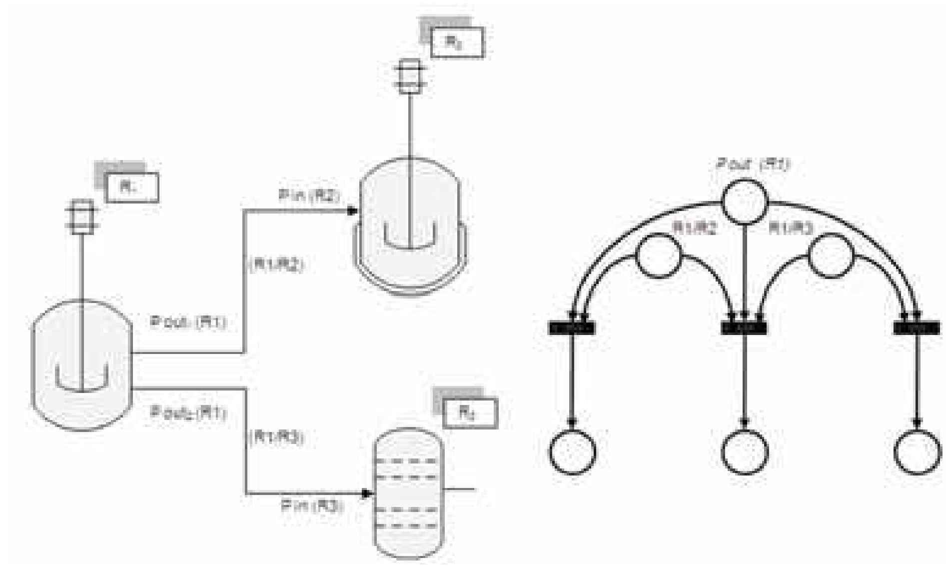


Fig. 12. Asynchronous Junction

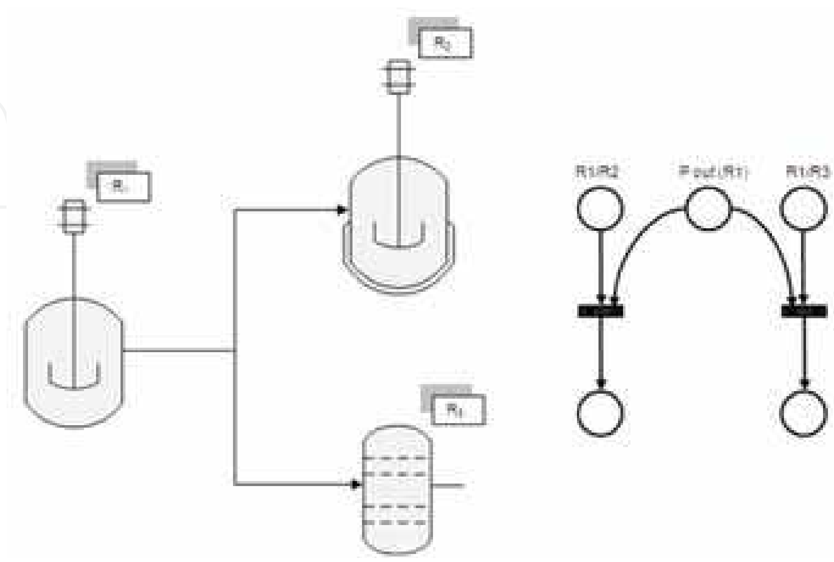


Fig. 13. Synchronous Junction

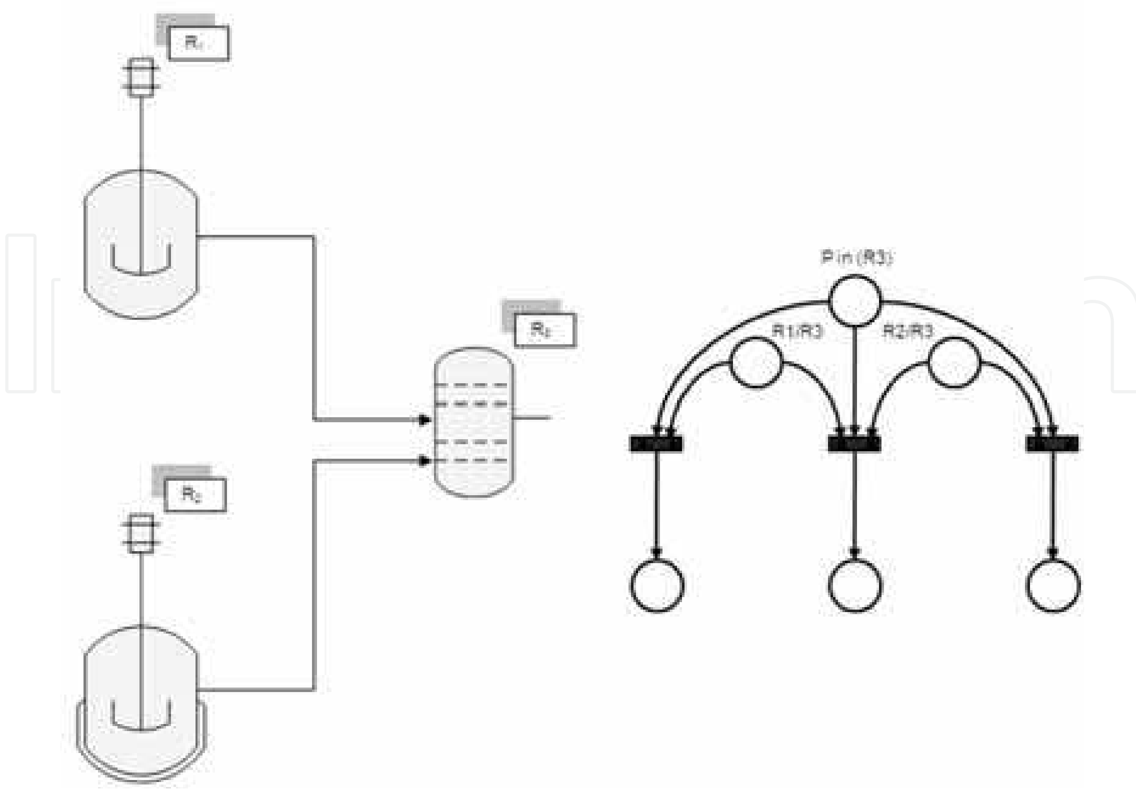


Fig. 14. Asynchronous Union

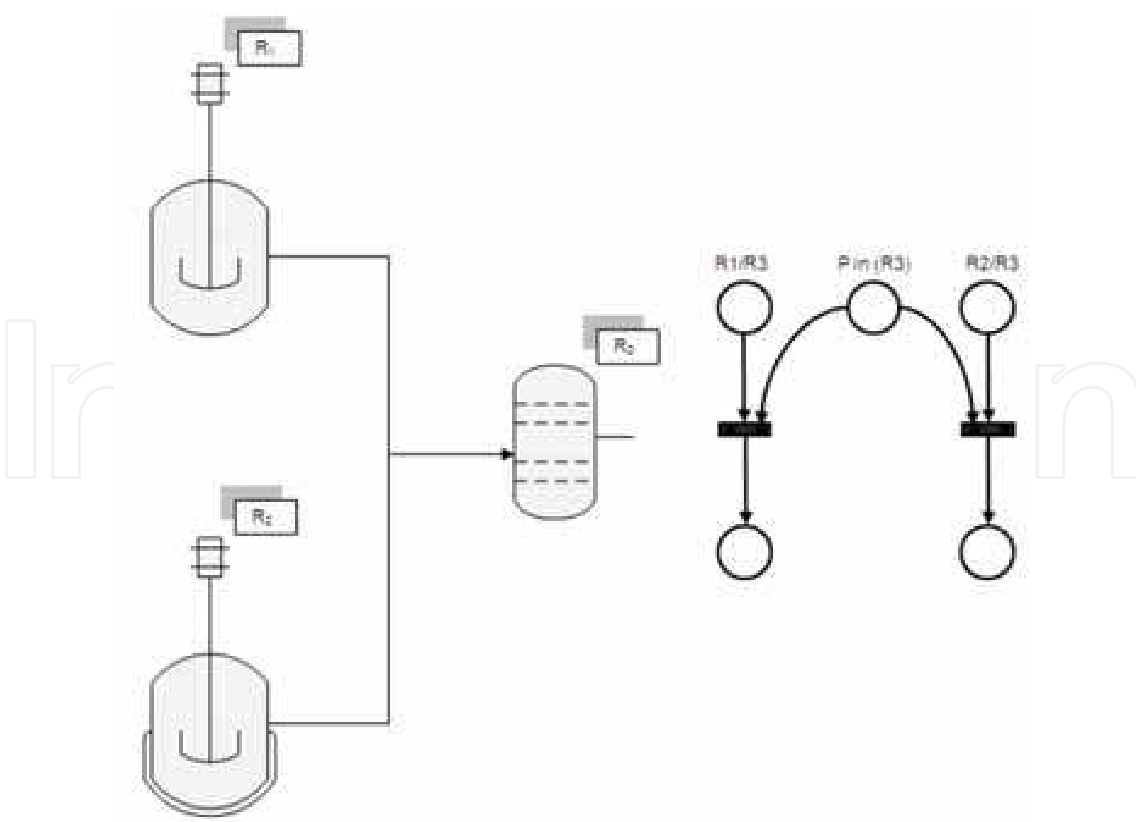


Fig. 15. Synchronous Union

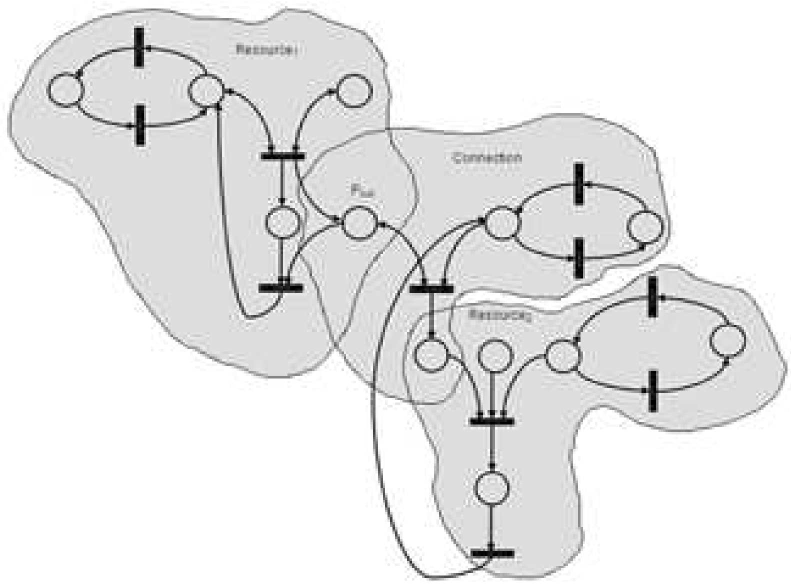


Fig. 16. Two connected resources

4.1.4 Service resources

Its function is to provide services to the HPU such as water, gas, steam and fuel. They have a unique skill and do not negotiate their capacity, thus the model used by them is equivalent to the one used be the connections.

4.1.5 Global model

The Petri Net model of the HPU is obtained through the composition of the models of the product, the resources and the connections. For the composition of the Product with the Resource the following construction is made: at the output of the transition "*send_req*" the "skill" places are created (see Figure 17). These are fusion places used by the product to call upon the HRs that have the skill to execute the operation. The product's "skill" places are fused with the "skill" places of the corresponding HRs, obtaining the model in Figure 18. The transition "*send_req*" is labeled as a silent event so it does not affect the generated and marked languages of the model, and the place "product" is activated when the resource goes into operation. This indicates the presence of a product.

Example: A continuous production plant as the one shown in Figure 19 is present. Table 2 shows resource skills. Figure 21 shows one of the products and its model in PN starting from figure 20. The structure of the connections is of asynchronous bifurcation/ asynchronous union. The plant's complete model is shown in Figure 22.

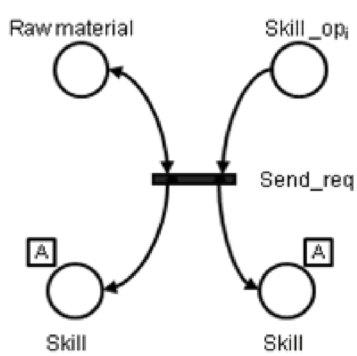


Fig. 17. Global model construction

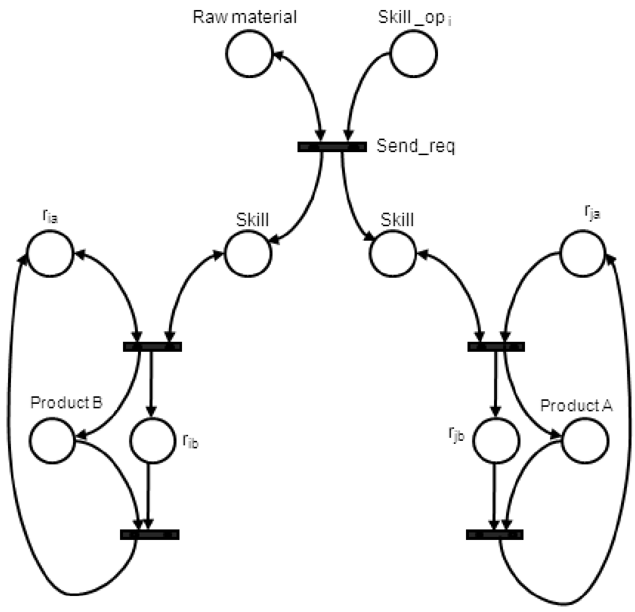


Fig. 18. Product - Resource composition

Holon	Skill
HR_1	op_1
HR_2	op_1
HR_3	op_2
HR_4	op_2
R_1	op_3
R_2	op_3

Table 2. Holon's skills

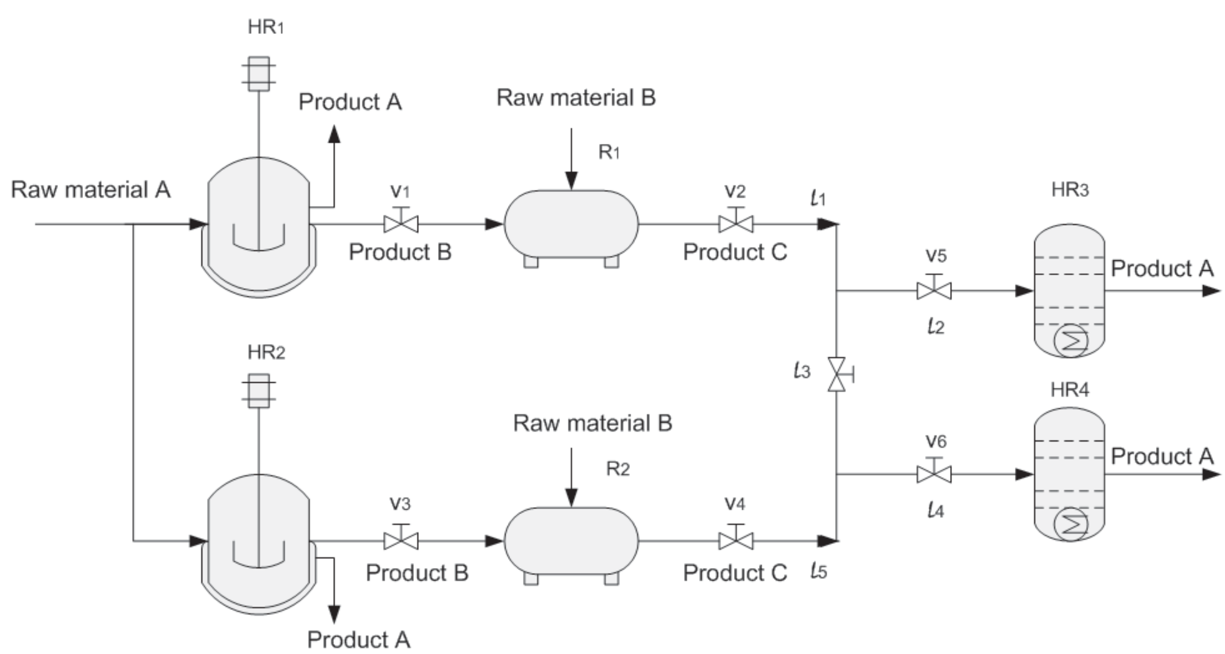


Fig. 19. Continuous Production plant

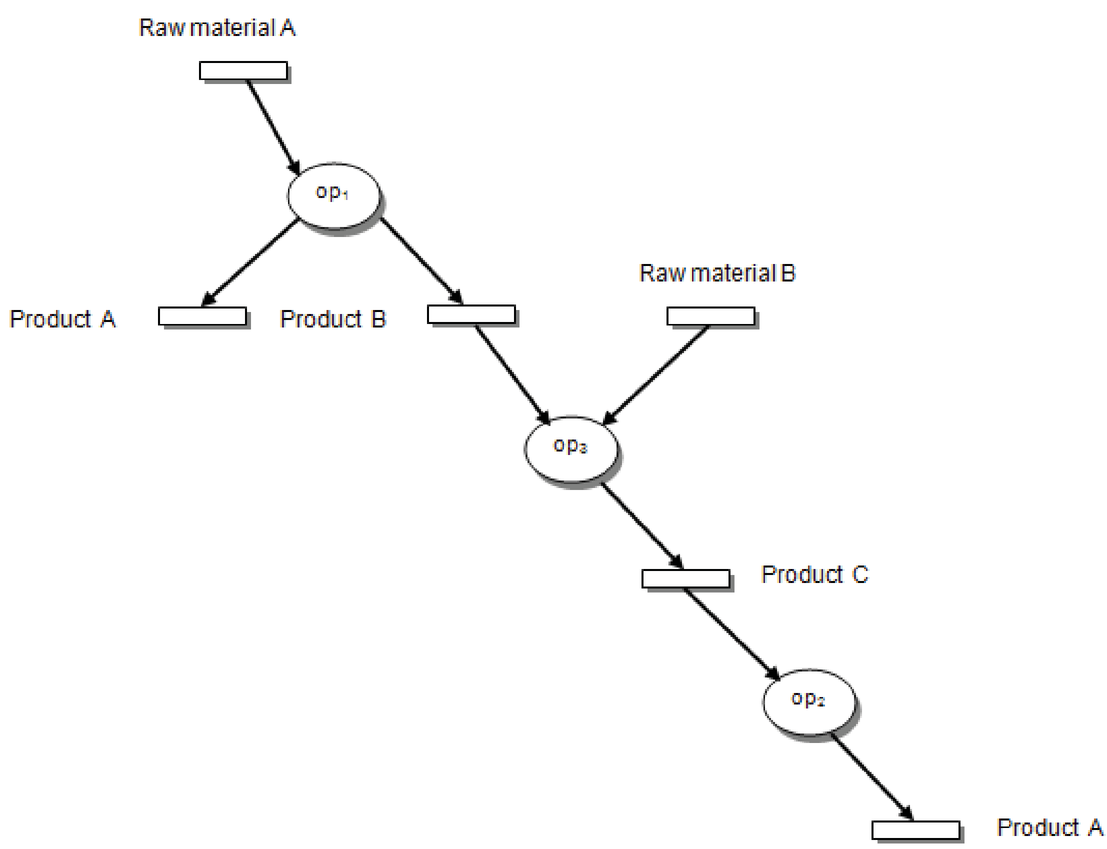


Fig. 20. P-Graph for a product of continuous plant

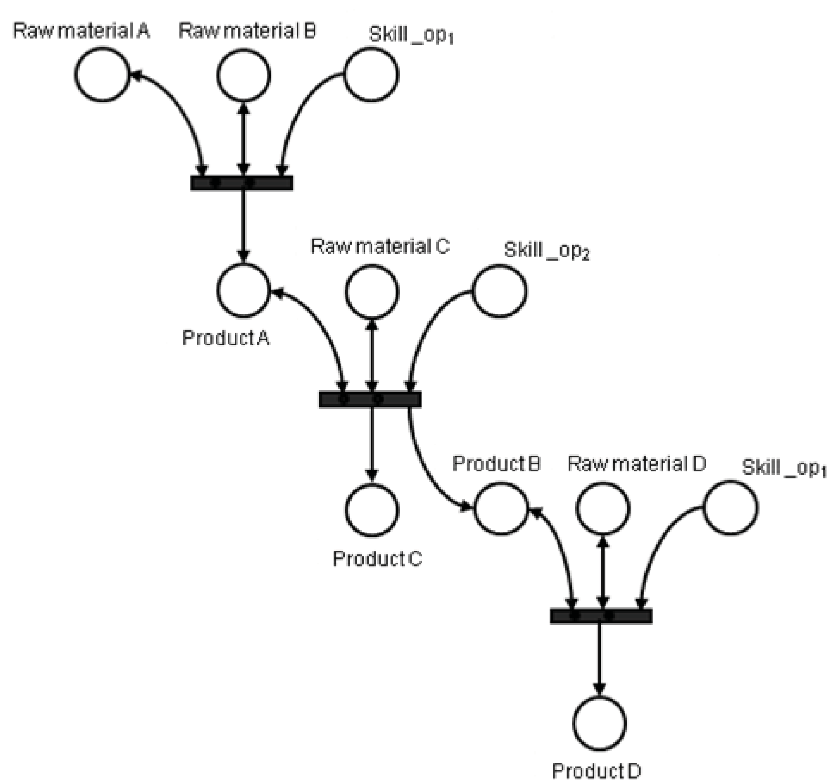


Fig. 21. PN product model for the example

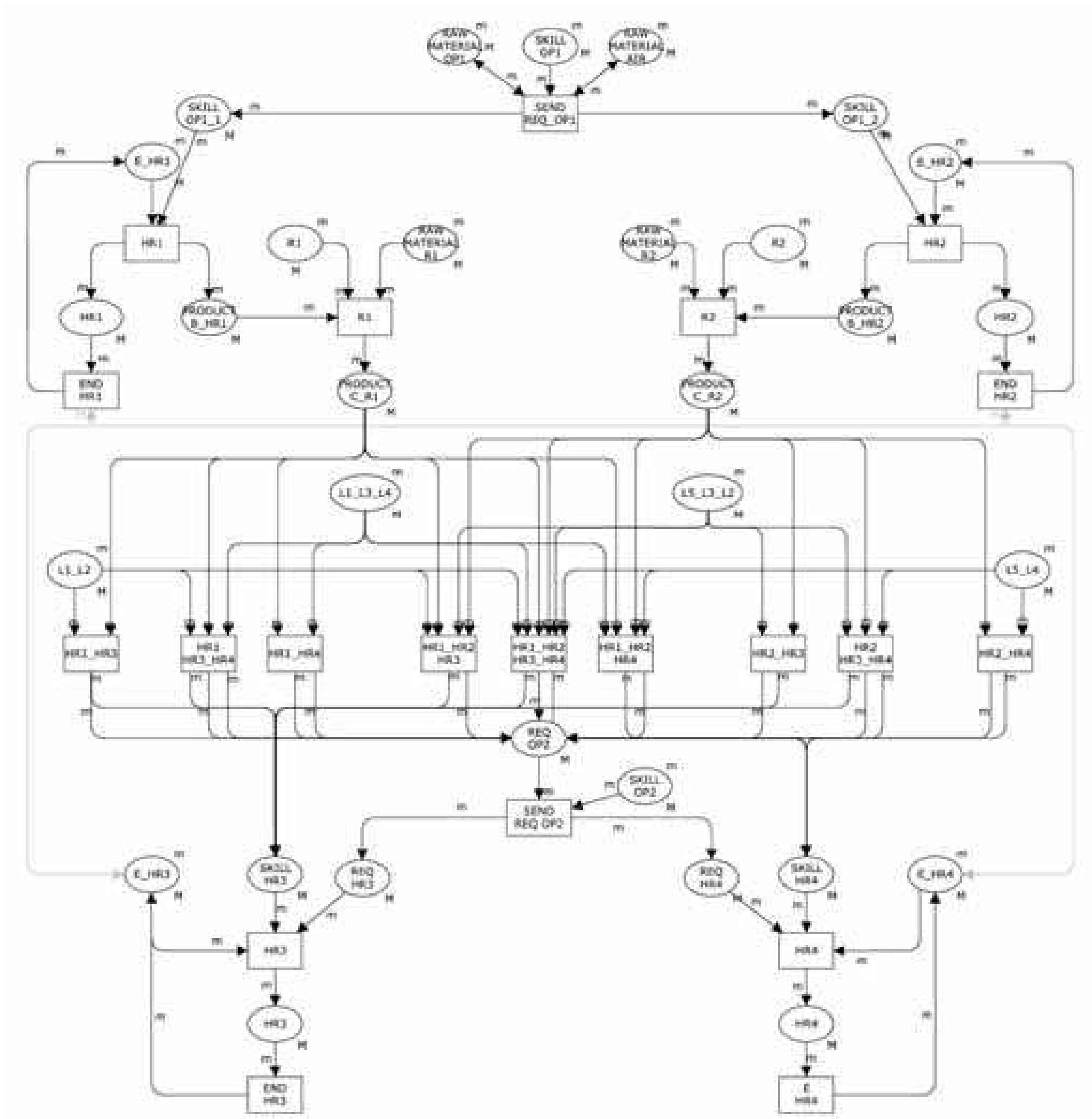


Fig. 22. Global PN model

The model obtained must comply with the properties of boundedness, safeness and liveness. This model was analyzed and simulated in CPNTools (Jensen, 2003) and the properties analysis gives a report where says that these properties are achieved.

4.2 Determination of configurations

When the negotiation process of a mission is launched, each holon sends a proposal that includes its availability, capacity and the cost of the operation. With the state (available / unavailable) the initial marking M_0 is determined. The PN is executed obtaining a reachability tree. The tree's arcs represent labeled events that enable determining the configurations through an operation of concatenation of the events.

For the example shown in Figure 19, if the following initial resource state is present ("1" for available).

$HR_1 = 1$ $HR_2 = 1$ $HR_3 = 1$
$HR_4 = 0$ $R_1 = 1$ $R_2 = 1$

Connection stretch $l_3 = 0$, which leads to the following state of the connections:

$HR_1 \rightarrow HR_4 = 0$
$HR_2 \rightarrow HR_3 = 0$

All other connections are available. The initial marking for this operative condition with a token in all states that represents an available resource and zero in the other one resources. Shown in the figure 23 are all the possible configurations of the HPU. The capacity of the configuration is shown in the units of the output variable or product of the HPU. Once the PN is obtained, the reachability tree of Figure 24 is found by means of CPNTools.

NUMBER	HPU	R	CONFIGURATION CAPABILITY (Product Units)
1	HR1		100
2	HR2		100
3	HR1 + HR2		200
4	HR1 + HR3	1	150
5	HR1 + HR4	1	150
6	HR1 + HR3 + HR4	1	150
7	HR2+ HR3 + HR4	2	150
8	HR2+ HR3	2	150
9	HR2 + HR3	2	150
10	HR1 + HR2 + HR3	1	250
11	HR1+ HR2+ HR3	2	250
12	HR1 + HR2 + HR3	1 + 2	300
13	HR1 + HR2 + HR4	1	250
14	HR1 + HR2 + HR4	2	250
15	HR1 + HR2 + HR3	1 + 2	300
16	HR1 + HR2 + HR3 + HR4	1	250
17	HR1 + HR2 + HR3 + HR4	2	250
18	HR1 + HR2 + HR3 + HR4	1 + 2	300

Fig. 23. Configurations

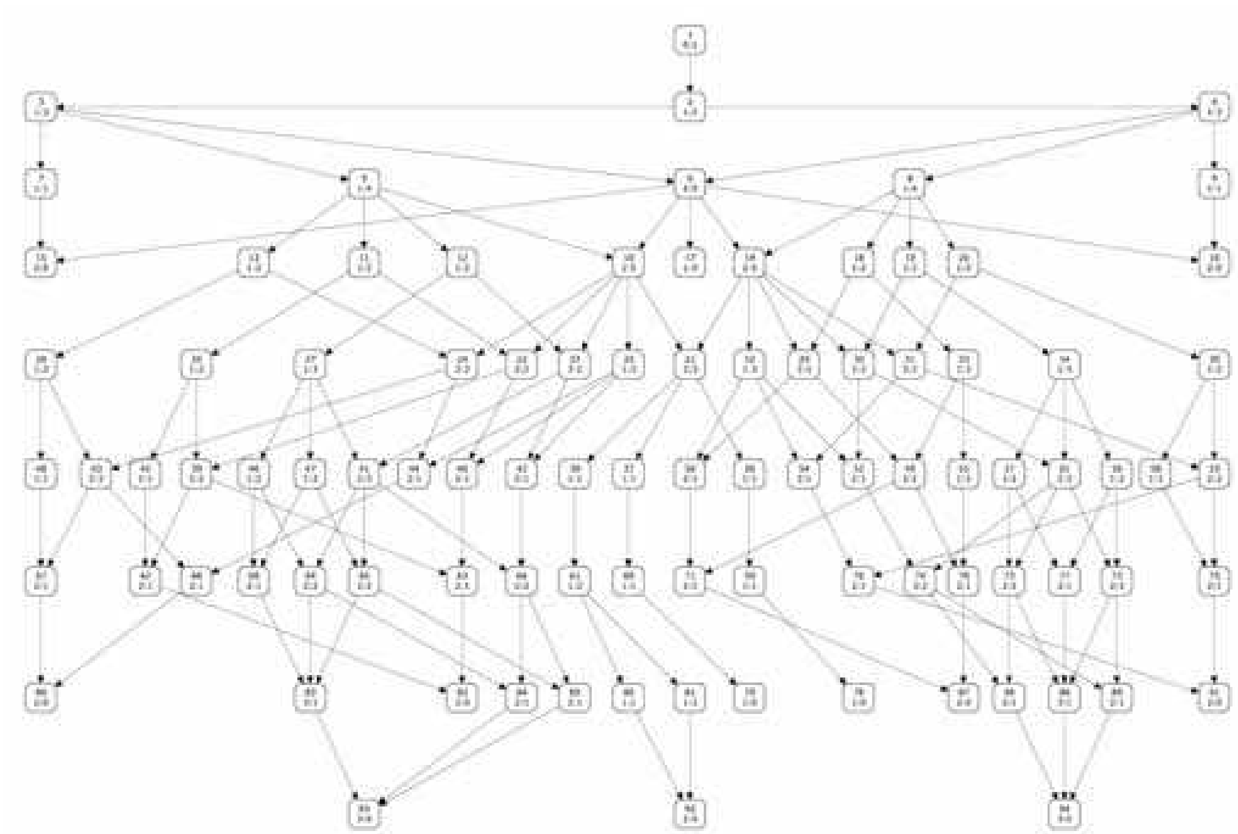


Fig. 24. Reachability tree

Analyzing the tree, the following is found:

- Nodes 13, 14, 15 y 32 lead to satisfactory terminations of the product, therefore, Q_m = {13, 14, 15,32}
- Nodes 17 and 31 do not lead to satisfactory terminations of the product, due to they enable R_2 with out HR_4 enable thus are forbidden states. Language $L = \{HR_2, R_2, HR_1, R_1\}$ is not permitted.
- The supervisor synthesis algorithm is applied, and the forbidden states and those leading to blockings are removed. The non-blocking property is verified with the base in the expression $\overline{L_m(G)} = L(G)$
- The possible configurations are established through the languages obtained from the initial state to the final states, following all trajectories. For instance, language $L = \{HR_1, R_1, HR_3, HR_2\}$ conduces to final state 32, thus 10 configuration is valid.
- Configurations incapable of achieving the mission are discarded, based in the capacity offers presented by the holons.
- The criteria of optimization are applied over the remaining configurations to select the definite configuration. This uses the operation cost information sent by the holon in the negotiation phase.
- The valid configurations for the example shown are: 1, 2, 3, 4, 10.

4.3 Holarchy formation

The holon resources that end up connected among themselves to enable cooperation, form a holarchy. The languages of the configurations enable establishing connections between holons and thus establish holarchies. For instance, from language $L = \{HR_1, R_1, HR_3\}$ the holarchy H_1 is obtained, formed by HR_1 and HR_3, in which these holons are connected by R_1. From language $L = \{HR_1, R_1, HR_3, HR_4, HR_2\}$ an HPU formed by holon HR_2 and holarchy $H = HR_1 + HR_3 + HR_4$ is obtained, connected through R_1.

5. Reconfiguracion

The holonic approach establishes the following disturbance response framework:

- If there is failure, the holon tries to adjust its control laws and its infrastructure to take care of the disturbance (autonomy attribute).
- If it is not capable, it turns to the holarchy to interiorly solve the situation through the cooperation of holons (cooperation attribute).
- And if the holarchy is not able to solve it, it turns to to other holarchies in the HPU.
- If the disturbance cannot be taken care of by the holarchies that form the HPU, a mission renegotiation is requested.

In the work presented, a PN of each holarchy is created, and the procedure presented for the determination of the HPU´s configurations is followed for reconfiguration.

Suppose, for the example presented, that the HPU operates with the configuration of Figure 25. The PN model of the holarchy is that of Figure 26. If it is presented a failure in HR_4, the PN marking is [1111001011010110000000. . .] (following the net from left to right and from top to bottom) and the tree it is in figure 27. And the HPU gets the configuration shown in figure 28.

The allowed states are 4 and 13. These states determine all possible configuration is $HR_1 + HR_3$ or HR_1 and the holarchy can resolve the disturbance inside. The criteria of optimization are applied to redistribute the mission among the holons. The holarchy has been reconfigured according to the method proposed which uses PNs and the supervisory control theory.

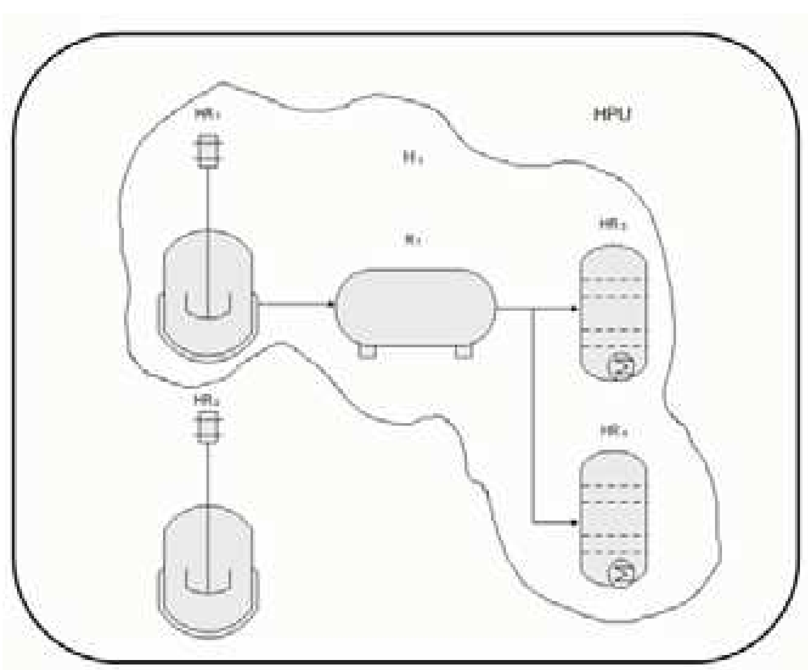


Fig. 25. Holarchy

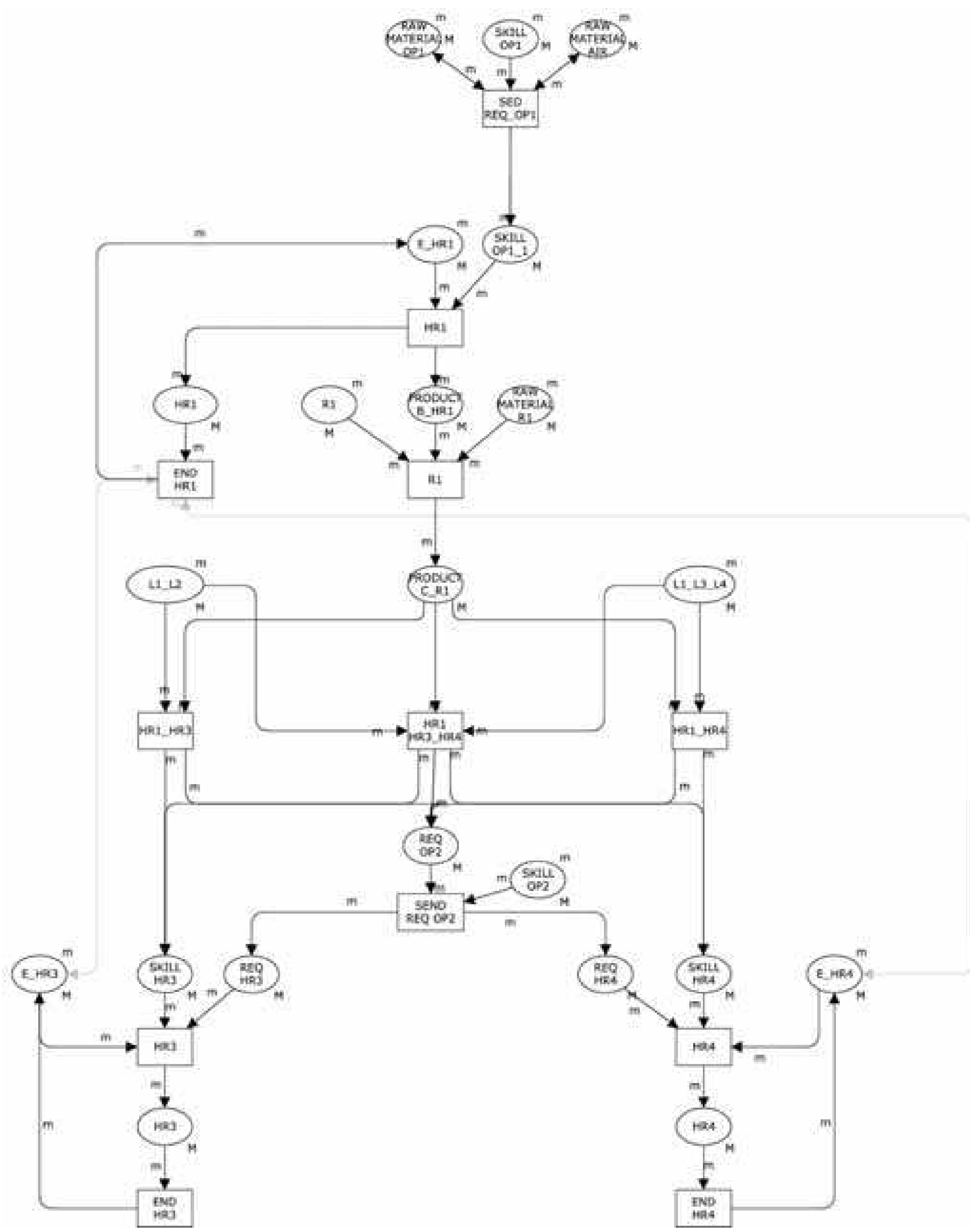


Fig. 26. Holarchy PN model

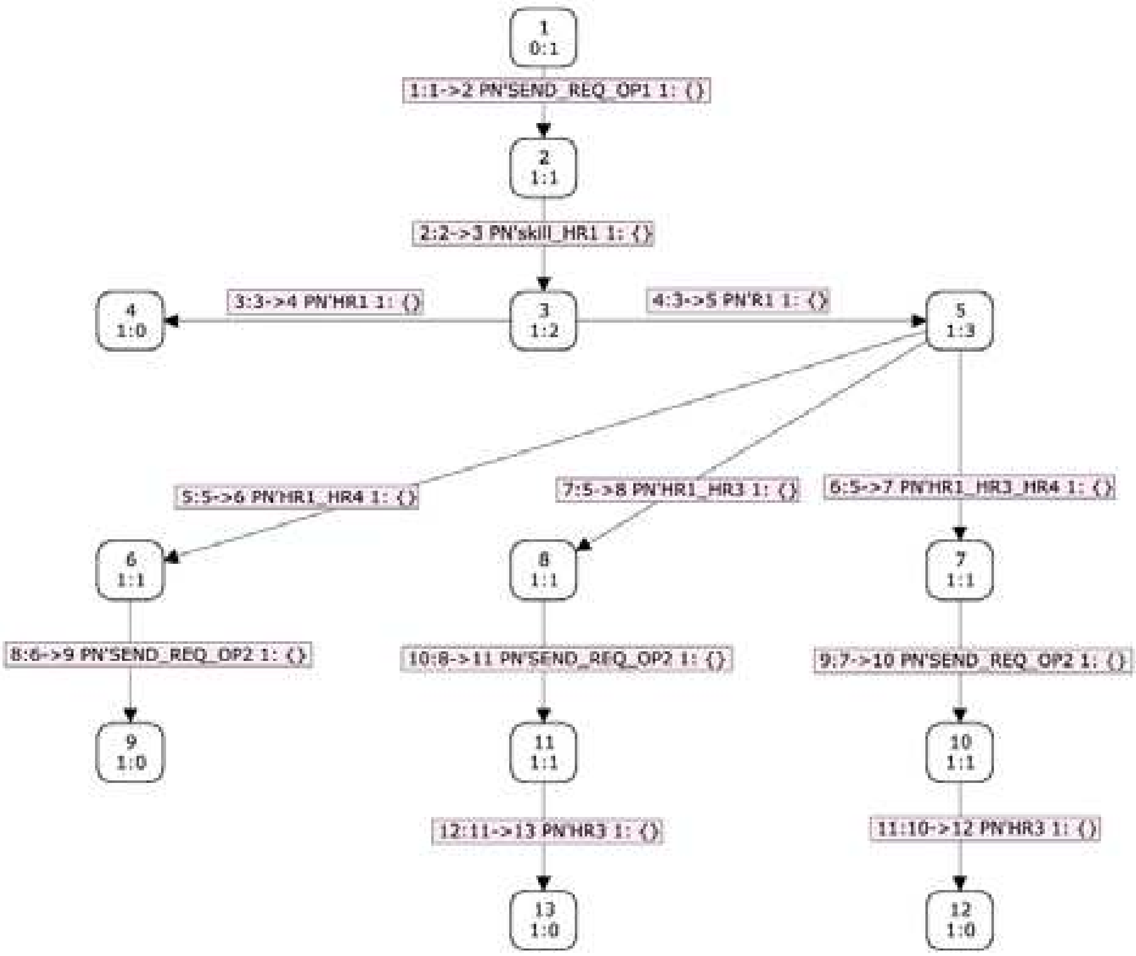


Fig. 27. New reachability tree

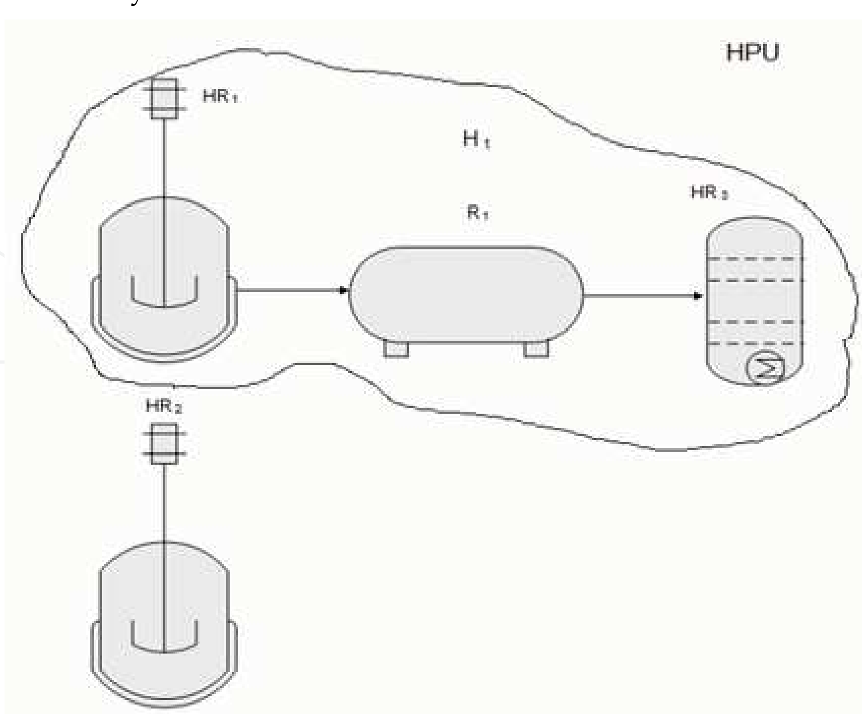


Fig. 28. New configuration

6. Conclusions and future papers

The work presented has shown the potential of the holonic paradigm by combining Petri Nets and the Supervisory Control Theory to solve configuration problems of continuous production processes in real time. The decrease in the problem´s complexity by applying the concept of holarchies is evident. This enables generating solutions with good temporary performances. Improving response times to face disturbances, an advantage of the holonic approach, is complied with in this manner.

In order to preserve the criteria of global optimization in the determination of the initial configuration, a composed model of the complete HPU is used. This model may not have a good performance in real time because it may be subject to explosion of the states. However, the determination of the initial configuration is part of the production scheduling function which can occur out of line. In this manner, another characteristic of the holonic approach is complied with: to conserve hierarchical structures that guarantee global optimums.

With regards to future papers, it is important to advance in the automatic generation of PNs from P-Graph models of the product, resource models and connections between them, and their automatic execution based on Petri Net engines.

The proposed methodology has been successfully proven in academic applications of production scheduling in thermal energy power plants.

7. References

Adam, E., Mandiau, R. & Kolski, C. (n.d.). Homascow: a holonic multi-agent system for cooperative work, *Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on*, pp. 247–253.

Agre, J., MCFarlane, D., Elsley, G., Cheng, J. & Gunn, B. (1994). Holonic control of a water cooling system for a steel rod mill.

Akesson, K. (2002). *Methods and Tools in Supervisory Control Theory*, PhD thesis.

Balasubramanian, S., Brennan, R. W. & Norrie, D. H. (n.d.). Requirements for holonic manufacturing systems control, *Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on*, pp. 214–218.

Bongaerts, L. (1998). Integration of scheduling and control in holonic manufacturing systems.

Brennan, R. W., Hall, K., Marik, V., Maturana, F. & Norrie, D. H. (2003). *A Real-Time Interface for Holonic Control Devices*, pp. 1088–1088.

Caramihai, S. (n.d.). An agent-based des supported modeling framework for enterprises.

Celaya, R., Desrochers, A. & Graves, R. (2009). Modeling and analysis of multi-agent systems using petri nets., *Journal of Computers*. 4: 981 – 996.

Chacón, E., Besembel, I., Narciso, F., Moltival & J y Colina, E. (2003). An integration architecture for the automation of a continuous production complex.

Chacón, E., Besembel, I., Rivero, D. & Cardillo, J. (2008). The holonic production unit: an approach for an architecture of embedded production process, *Advances in Robotics, Automation and Control* .

Chacón, E. & Colmenares,W. (2005). A way to implement supervisors for holonic production units.

Cheng, F.-T., Chang, C.-F. & Wu, S.-L. (2004). Development of holonic manufacturing execution systems, *Journal of Intelligent Manufacturing* 15(2): 253–267.

Chirn, J. & McFarlane, D. (n.d.). Evaluating holonic control systems: A case study.

Chokshi, N. & McFarlane, D. (2008a). A distributed architecture for reconfigurable control of continuous process operations, *Journal of Intelligent Manufacturing* 19(2): 215–232.

Chokshi, N. N. & McFarlane, D. C. (2008b). *A Distributed Coordination Approach to Reconfigurable Process Control*, Springer Series in Advanced Manufacturing, Springer, Hardcover.

Chokshi, N. N. & McFarlane, D. C. (n.d.). Rationales for holonic manufacturing systems in chemical process industries, *Database and Expert Systems Applications, 2001. Proceedings. 12th International Workshop on*, pp. 616–620.

Durán, J. (2006). Técnicas emergentes para la automatización integrada de procesos de producción integraciÓn en automatizaciÓn- reporte técnico 2.

El Kebbe, D. (2002). *Towards the MaSHReC Manufacturing System under real time constraints: a contribution to the application of real time system advances to production control*, PhD thesis.

Falkman, P., Lennartson, B. & Tittus, M. (2009). Specification of a batch plant using process algebra and petri nets, *Control Engineering Practice* 17(9): 1004–1015.

Fischer John, B. T. O. (n.d.). *Workbook for Designing Distributed Control Applications using Rockwell Automation's HOLOBLOC Prototyping Software*.

Fu-Shiung, H. (n.d.). Collaborative timed petri net for holonic process planning, *American Control Conference, 2003. Proceedings of the 2003*, Vol. 1, pp. 344–349 vol.1.

Ghaeli, M., Bahri, P. A., Lee, P. & Gu, T. (2005). Petri-net based formulation and algorithm for short-term scheduling of batch plants, *Computers & Chemical Engineering* 29(2): 249–259.

HMS (2004). Holonic manufacturing systems consortium. http://hms.ifw.uni-hannover.de.

Holonic Manufacturing Systems (2008). pp. 7–20.

Hsieh, F.-S. (2006). Analysis of contract net in multi-agent systems, *Automatica* 42(5): 733 740.

http://research.curtin.edu.au/ (2010). Chemical engineering example a. summary of proposed research program for doctor of philosophy, scheduling of batch and mixed batch/continuous process plants using petri-nets.

URL:http://research.curtin.edu.au/guides/hdrguidelines/docs/CandEx_ChemEng_A.pdf

Jensen, K. (2003). *Coloured Petri Nets : Basic Concepts, Analysis Methods and Practical Use. Volume 1 (Monographs in Theoretical Computer Science. An EATCS Series)*, Springer.

Koestler, A. (1968). The ghost in the machine.

Langer, G. (1999). Homucs - a methodology and architecture for holonic multi - cell control systems.

Leitao, P. (2004). *An Agile and Adaptive Holonic Architecture for Manufacturing Control Dissertation*, PhD thesis.

Leitao, P., Valckenaers, P. & Emmanuel, A. (2009). Self-adaptation for robustness and cooperation in holonic multi-agent systems.

Lennartson, B., Fabian, M. & Falkman, F. (n.d.). Control architecture for flexible production systems, *Automation Science and Engineering, 2005. IEEE International Conference on*, pp. 307–312.

Lennartson, B., Tittus, M. & Fabian, M. (n.d.). Modeling, specification and controller synthesis for discrete event systems, *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, Vol. 1, pp. 698–703 vol.1.

Lobo, C. (2003). Sistema multiagente para coordinar unidades de producción.

Maturana, F. P., Tichy, P., Slechta, P., Staron, R. J., Discenzo, F. M., Hall, K. & Marik, V. (n.d.). Cost-based dynamic reconfiguration system for intelligent agent negotiation, *Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on*, pp. 629– 632.

McFarlane, D. (1995). Holonic manufacturing systems in continuous processing: concepts and control requirements, *In Proceedings of ASI' 95* p. 11.

Mcfarlane, D. C. & Bussman, S. (2000). Developments in holonic production planning and control, *Production Planning & Control* 11(6).

Music, G. & D., M. (1998). Petri net based supervisory control of flexible batch plants.

Peréz, L. (n.d.).

Pétin, J.-F., Gouyon, D. & Morel, G. (2007). Supervisory synthesis for product-driven automation and its application to a flexible assembly cell, *Control Engineering Practice* 15(5): 595–614.

Ramadge, P. J. G. & Wonham, W. M. (1989). The control of discrete event systems, *Proceedings of the IEEE* 77(1): 81–98.

Ramos, C. (n.d.). A holonic approach for task scheduling in manufacturing systems, *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, Vol. 3, pp. 2511–2516 vol.3.

Reveliotis, S. (1999). Real-time control of flexibly automated production systems, *AutoSimulations Symposium '99* .

Silva, M. (n.d.). Las redes de petri en la automática y la informática, *Technical report*.

Sousa, P. & Ramos, C. (1998). A dynamic scheduling holon for manufacturing orders, *Journal of Intelligent Manufacturing* 9(2): 107–112.

Suda, H. (1989). Future factory system formulated in japan, *Japanese Journal of Advanced Automation Technology* 1.

Tittus, M. & Akesson, K. (1999). Petri net models in batch control.

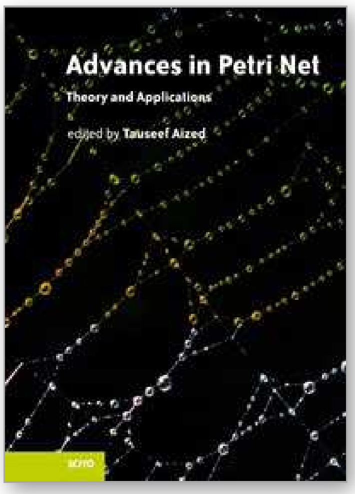
Tittus, M., Akesson & K (n.d.). Deadlock avoidance in batch processes.

Tittus, M. & Lennartson, B. (1997). Hierarchical supervisory control for batch processes, *Control Systems Technology, IEEE Transactions on* 7(5): 542–554.

Tuncel, G. & Bayhan, G. (2007). Applications of petri nets in production scheduling: a review, *The International Journal of Advanced Manufacturing Technology* 34(7): 762–773.

Wyns, J. (1999). Reference architecture for holonic manufacturing systems.

Zhou, M. (1995). Petri nets in flexible and agile automation.



Advances in Petri Net Theory and Applications
Edited by Tauseef Aized

ISBN 978-953-307-108-4
Hard cover, 220 pages
Publisher Sciyo
Published online 27, September, 2010
Published in print edition September, 2010

The world is full of events which cause, end or affect other events. The study of these events, from a system point of view, is very important. Such systems are called discrete event dynamic systems and are of a subject of immense interest in a variety of disciplines, which range from telecommunication systems and transport systems to manufacturing systems and beyond. There has always been an intense need to formulate methods for modelling and analysis of discrete event dynamic systems. Petri net is a method which is based on a well-founded mathematical theory and has a wide application. This book is a collection of recent advances in theoretical and practical applications of the Petri net method and can be useful for both academia and industry related practitioners.