ISSN 2176-5480

22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil

DESIGN OF COLLABORATIVE AND RECONFIGURABLE CONTROL ARCHITECTURE FOR DISPERSED PRODUCTIVE SYSTEMS

Robson Marinho da Silva
Universidade Estadual de Santa Cruz, Ilhéus, BA, Brasil
rmsilva@uesc.br
Fabrício Junqueira
Diolino José dos Santos Filho
Paulo Eigi Miyagi
Escola Politécnica da Universidade de São Paulo, São Paulo, SP, Brasil
fabri@usp.br, diolino.santos@poli.usp.br,pemiyagi@usp.br

Abstract. *In general, the control systems for conventional manufacturing systems are organized into hierarchical layers such as, enterprise control, factory control, and shop floor control. In an approach based on division in scope and breadth of the problems these levels are designed to act with relative autonomy. However, currently manufacturing systems must necessarily be reconfigurable to assure features of adaptability to unpredictable changes, agility, flexibility and efficiency, i.e., the parts of the reconfigurable manufacturing system (RMS) must collaborate each other. A RMS must combine a global vision with a relatively agile reaction to unexpected variations of the plant. On the other hand, collaborative activities in RMS can be treated as services and therefore, service oriented architecture (SOA) can be effective for integration of the control layers. Besides, in a global market, the RMS must also consider a reconfiguration of subsystems that can be geographically dispersed and that are processing information, physical transformations and material handling. In this context, the integration of multi-agent system and holonic system techniques named as holonic control system (HCS) has been considered a promissory tendency for the control system of the RMS. Therefore, this paper introduces a procedure to design a service oriented active holonic control system (SOAHCS) and also presents an application example for a benchmark dispersed productive system. SOAHCS combines bottom–up and top–down approaches using Petri net technique, and its extensions to represent the structure and control operations under faults occurrence, and explores the combination of HCS and SOA techniques, such as autonomy, reactivity, proactivity, cooperation, learning, reuse, interoperability and portability.*

Keywords: *manufacturing system; reconfigurable system; service oriented architecture; holonic control system; dispersed productive system*

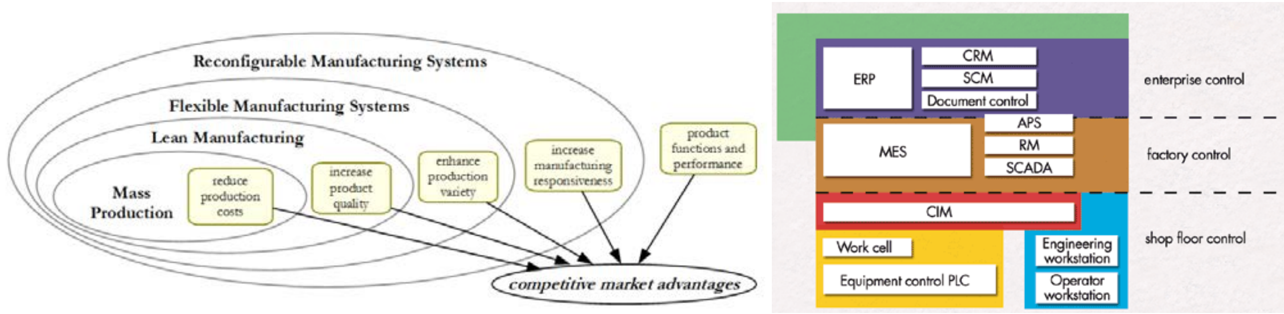
1. INTRODUCTION

Manufacturing systems (MSs) have been designed to meet the demands of production of goods, i.e., to transform materials and process information in the execution of the work orders. The growing competitiveness and the need for efficiency also imposed changes in MSs requiring greater flexibility under different aspects, such as the volume of production, the type of product and resources. The challenges in the evolution of MSs are characterized by a gradual migration of production paradigms: (i) mass production and lean manufacturing are concerned with the production of cheaper products, and the elevation of production quality; (ii) flexible manufacturing system (FMS) is concerned with the production diversity; and (iii) reconfigurable manufacturing system (RMS) is concerned with the need to be "adjustable" to the business and market interests. Figure 1a shows as the economic goals are associated with this evolution (Mehrabi *et al.*, 2000). Besides, there are various studies for dispersed productive systems (DPSs) which are composed by several MSs installed and running in different geographic locations (Garcia Melo *et al.*, 2008; Fattori *et al.*, 2011). Thus, the DPS is a type of system that is formed by several subsystems which perform different processes (material transformation and/or information processing) and interact each other in different ways (Ali *et al.*, 2005; Silva *et al.*, 2011, 2012a,b).

Despite the advantages of a RMS, its implementation is not trivial, because there is a need for practical solutions for issues such as: (i) how is the effective way to increase the redundancy and technology in machines and their controllers, (ii) how to provide optimized alternative routes of transport subsystems, and (iii) how systematize the use of qualified staff to update (or replace) legacy system. This, associated with the lack of quantitative performance data to evaluate these aspects, restrict the expansion of this paradigm (Vrba *et al.*, 2011).

In general, the control functions in MSs are organized into hierarchical layers, as in Fig. 1b: (i) *enterprise control* — where decisions and control are of management level and it is based on a long-term planning; (ii) *factory control* — where decisions and control are related with the ordination of requests (based on some produtive strategy) and supervision of the productive processes; and (iii) *shop floor control* — where decisions and control are of machines operations (which are automatically executed through programmable controllers connected to sensors and actuators) (Groba *et al.*, 2008). However, based on the above mentioned aspects, a RMS must review this hierarchical structure and combine it with

ISSN 2176-5480



(a) Evolution of paradigms (b) Control layers

Figure 1. Manufacturing system

heterarchical control architecture, i.e., considering a global vision of the system goal and it performance with a framework for a relatively agile reaction to unexpected variations of the plant (Silva *et al.*, 2012a).

Therefore, to ensure that a RMS meets its purpose, it is also necessary to update their control system. The reconfigurable manufacturing control system (RMCS) must ensures the system functionalities and the requirements of actual production processes, considering the appropriate use of mechatronic and computer-aided technologies. The design of RMCS should consider the reuse of models developed for specification of productive processes and control functions, their interoperability and portability (Tommila *et al.*, 2005) to avoid repetition of tasks and overlapping of project scope.

Based on intrinsic feature of faults occurrence, in man-made systems, the design of RMCS should also consider mechanisms and solutions based on active fault-tolerant control system (AFTCS) (Zhang and Jiang, 2008). AFTCS involves fault detection, study of its effects, identification of its causes and finally, reconfiguration of the system through relocating and choosing alternative productive processes. According to Squillante *et al.* (2012), faults can be classified as critical or noncritical. In case of occurrence of noncritical fault, the strategy is to recover the functionality of the system (also called *regeneration*) or maintain operations so that parts affected by the fault critical are disabled without affecting other parts of the system (also called *degeneration*) (Silva *et al.*, 2012d).

In this context, the multi-agent system (MAS) allows the conception of an intelligent control component (Ferber, 1999) for RMCS. Moreover, the agent-based control software, which is part of the resource, can be endowed with communications and information processing capabilities, transforming it into a self-reconfiguring, intelligent element, i.e., a holon (Koestler, 1968). The result is a distributed intelligent automation system associated with the lowest layer of a holonic system (HS). This multi-agent based control system is called holonic control system (HCS) (Colombo *et al.*, 2001).

In previous studies (Silva *et al.*, 2012a,b, 2011), the HCS was combined with AFTCS and the resulting system was named active holonic control system (AHCS) that was applied in intelligent buildings, another important class of DPS. Based on this initiative, in Silva *et al.* (2012c) the focus was the description of a mechanism for the holarchy composition problem (HCP) that allowed AHCS to execute productive processes meeting the timing constraints. In Silva *et al.* (2012d), AHCS was applied for RMS where the focus was on reconfiguration and description of the control mode switching (between two operational modes).

The RMCS must certainly consider also the collaboration activities among entities that compose the system and according Han *et al.* (2008), the service oriented architecture (SOA) is an effective solution. In the control functions structure, an integration layer is introduced between the *factory control* and *shop floor control* layers. The integration layer is a middleware which have available application tools and interfaces to communicate with other applications in other layers. For example purely computational systems CORBA (Mowbray and Zahavi, 1995) is a practical example of this middleware.

Some other studies presented new perspectives about how to combine the concepts of SOA and MAS, but there are few publications that deal with the systematization of a design procedure. Mendes *et al.* (2009) explain how to combine the service-oriented agents in industrial automation, sharing resources in the form of services by sending requests between agents. Nagorny *et al.* (2012) proposed SOA as *automation service cloud,* i.e., the use of hardware and software accessed remotely via communication network, and where operational activities are described as services.

In the light of the foregoing, this paper introduces a procedure for design service-oriented active holonic control system (SOAHCS). The text has the following structure: in Section 2 the SOAHCS is presented; in Section 3 the design procedure for SOAHCS is presented using an application example, i.e., a manufacturing system that emulates a DPS; and in Section 4 the main conclusions are presented.

2. SOAHCS ARCHITECTURE

In this paper, RMS is approached as a class of discrete event system, i.e., Petri net (PN) and its extensions can be used for description of the system behavior (Murata, 1989). Specifically, this work adopts a channel/agent Petri net

ISSN 2176-5480

type called production flow schema (PFS) (Miyagi *et al.*, 1988) for conceptual description of the system, and a extended place/transition Petri net (E-PN) to which `temporized transitions, inhibitor arcs` and `enabling arcs` (terms related to PN are in `Typewriter`) were added (David and Alla, 1994) for functional description of the system. The idea of SOAHCS is to combine bottom-up approach from the productive processes models in E-PN, and the top-down approach of stepwise refinement associated of the PFS model of the system's activities associated with the flows of discrete items (information, material) .

The proposed architecture, in Fig. 2, considers the following control levels of a RMS:

(i) the planning level holon (*product holon – PrH*) contains the necessary knowledge for the general operation of MS and for choosing the general strategy that attains the planned objectives. Each *PrH* has an internal process flow, the required input types and output types. The *PrHs* represent the products of the RMS which can be intermediates, i.e., which have some manufacturing operation to get the final products;

(ii) the ordination level holon (*strategies holon – StH*) contains the knowledge to manage the execution of the strategies to attend the requests (market demands);

(iii) the supervision level holon (*supervisor holon – SuH*) contains all the knowledge to coordinate the holons of lower hierarchical levels, i.e., coordinating their operations, registering the abilities of each component and providing services combined with other entities of the control system; and

(iv) the local control level holon (*operational holon – OpH*) represents the RMS physical resources (equipment for e.g.) that have specific control devices for its automatic operation, and determines the behavior of these resources in accordance with its objectives and abilities.

Following the specification of FIPA for MASs (Fipa, 2002), the structure and relations between these holons are showed in Fig. 3 according to the UML class diagram (Booch *et al.*, 2005).

To process an production order, SOAHCS forms a holarchy which is a productive process dynamically created based on the collaboration of holons. A holon can belong simultaneously to different holarchies. The challenge is to determine the best holarchy formed to fulfill the production order based on the available resources. To achieve this objective, E-PN models are used to represent workflows and activities of holons must satisfy certain timing constraints so that the overall collaborative workflow can meet the RMS requirements. SOAHCS does the calculations using the `temporized transitions` in E-PN models. For example, let $[H_n]$ be the a production sequence *n* in the RMS formed by *PrHs* to obtain the final product. Let, consider the following sequences: *$[H_1]$: [A0]→ [A1]→ [A2]→ [B0]→ [B1]→ [B2]→ [A2+B2]* and *$[H_2]$: [A0]→ [A3]→ [B0]→ [B3]→ [A3+B3]*. Suppose the due date set by *PrH-[B2]* is $t_e[B2]$. This order due date imposes a timing constraint on *PrH-[B1]* in sequence of $[H_1]$. Let t_e[B1] be the latest time that *PrH-[B1]* must complete all its operations. Let $t_e[t_n]$ be the time that `transition` *n* must complete its operations. To meet the due date, the constraint of Eq. (1) must be satisfied by *PrH-[B1]* and so on for another holons.

$$t_e[B1] \leq t_e[B2] - t_e(t_{18} + t_{19} + t_{20} + t_{21} + t_{22}) \tag{1}$$

SOAHCS associates a variable named cost to H_n or *PrH* to formation of holarchies. Let C_N be the cost of a production sequence *N* of a H_n or a *PrH*. Let c_n be the cost of a `transition` *n* of E-PN model. For composite holarchies SOAHCS calculates the sequence that offers the shortest cost. In the example, it compares the cost of C_{H1} and C_{H2} to decide what is the better sequence for production, according Eq. (2) and Eq. (3). The c_n for each *PrH* is calculated according the case for *PrH-[B2]* in Eq.(4).

$$C_{[H1]} = C_{[A0]} + C_{[A1]} + C_{[A2]} + C_{[B0]} + C_{[B1]} + C_{[B2]} + C_{[A2+B2]} \tag{2}$$

$$C_{[H2]} = C_{[A0]} + C_{[A3]} + C_{[B0]} + C_{[B3]} + C_{[A3+B3]} \tag{3}$$

$$C_{[B2]} = c_{t18} + c_{t19} + c_{t20} + c_{t21} + c_{t22} \tag{4}$$

The application of the fault-tolerance concept in SOAHCS is divided into four phases for each holon independently of the hierarchical level. The "estimation phase" involves the detection of symptoms and the isolation of faults that allow their identification. The reconfiguration is decided in the "planning phase", which is based on predefined priorities such as reduction of performance, shorter recovery time, etc., and on historical data, from which it is possible to measure the statistical significance of each type of fault in terms of frequency rate, recovery time, and operational cost. The "execution phase" involves sending commands for the execution of the selected action plan. The last phase is the "learning phase", which involves the storage of relevant data for use in further cases. Therefore, SOAHCS acts in accordance with the following rules:

(i) *if <symptoms> then <selects fault>*;

(ii) *if <selected fault> then <selects action>*;

(iii) *if <selected action> then <activates reconfiguration>*; and

(iv) *if<executed reconfiguration> then <store relevant data>*.

ISSN 2176-5480

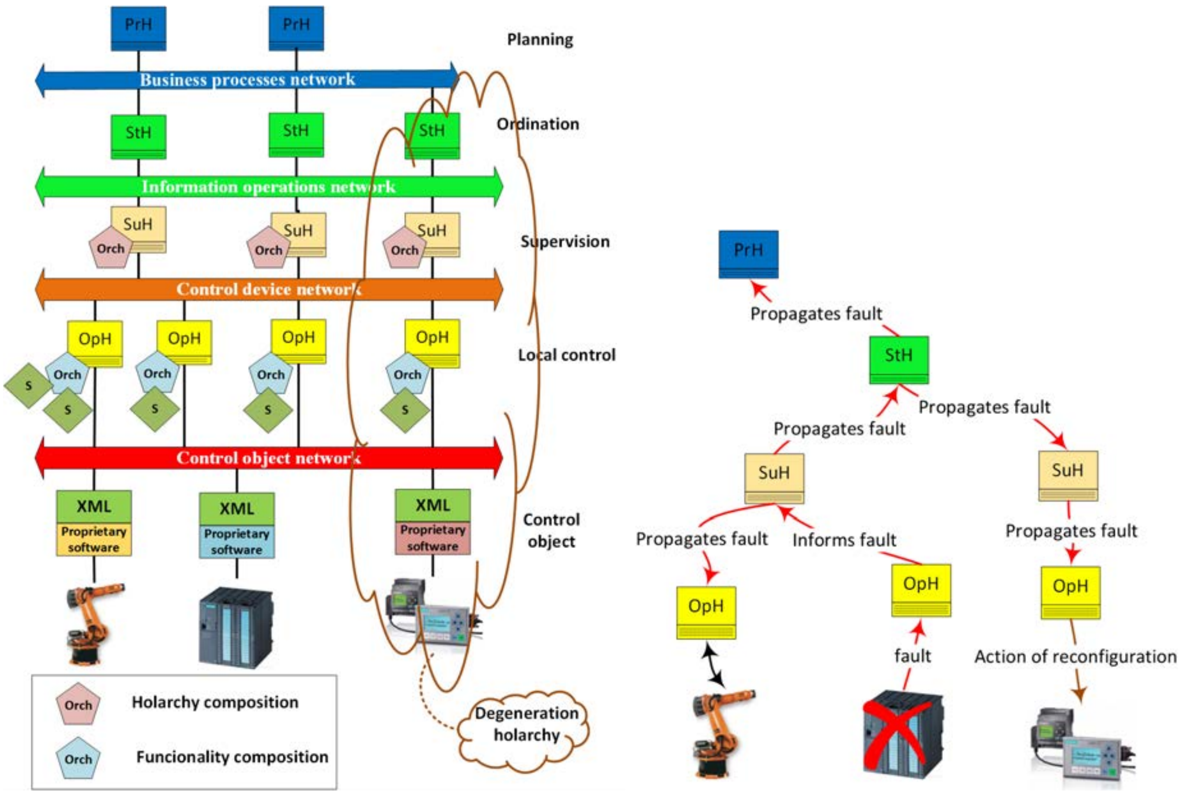


Figure 2. SOAHCS architecture control and the interaction in fault occurrence

In case of fault occurrence, the holons propagate messages to ensure agile response to faults as illustrated in the right of the Fig. 2. The holon that detected a fault sends this information to *SuH* starting the propagation of messages and indicating the need for reorganization. Thus, *StHs* receive the message propagating this need to neighboring holons in control structure. To prevent accidents, a degeneration holarchy can be formed allowing quicker reaction to critical faults.

Figure 4 illustrates how to estimate the reconfiguration time. If the resource affected by the fault become unavailable for a long period of time, the *OpH* estimates the recovery time (t_r), checks the planned orders during the recovery period and then cancels the current allocation of these orders notifying the *StH*. For this, *the OpH* estimates two different parameters:

(i) t_e, the time that the order needs to return to the *StH*, and

(ii) t_c, the time to check if the fault was recovered, and to re–estimate and to re–apply the parameters of time if the fault was not recovered as expected.

To calculate these two values it is determined the t_r spent in the previous treatment faults and it is considered on this time 50% for t_c and 90% for t_e. If the fault is not recovered, it is necessary to re-estimate the parameters of time and to cancel the strategies that are planned for this new time interval. It is obtained by $t_c + t_e$. During the time that the resource is unavailable, the *OpH* only receives new commands if they can be run outside the range of estimated time for recovery.

The negotiation mechanism is based on contract net protocol (CNP) (Smith, 1980) and activity-based costing (ABC) (Cooper, 1988). Thus, SOAHCS adopts rules that allow negotiation between holons based on credits (rewards) and fee (penalties) depending if the order is completed in due time or not. When a *StH* is responsible for implementing a particular strategy, it receives the following information of the *PrH*: the strategy chosen, a quantitative measure called “order production fund” (π), the scheduled time, an amount of penalty for delay (φ), and a reward value (ε) to be finalized successfully. The *StH* should manage the negotiation with the *OpHs* to achieve the goal without exceeding the service fund and, in resource allocation process, regarding the performance of them given by μ. Table 1 summarizes the evolution of this mechanism.

3. PROCEDURE FOR SOAHCS DESIGN

This section presents a description of the procedure since the initial conception phase of a SOAHCS until RMS operation phase. In the following explanation, an application example of a RMS uses a benchmark dispersed productive system (BDPS), illustrated in Fig. 5a.

Phase 1 – analysis of requirements – on this phase, SOAHCS’ specifications are defined: aim of the system, control object, control devices, definition of tasks, strategies and control functions, description of the interaction among parts of

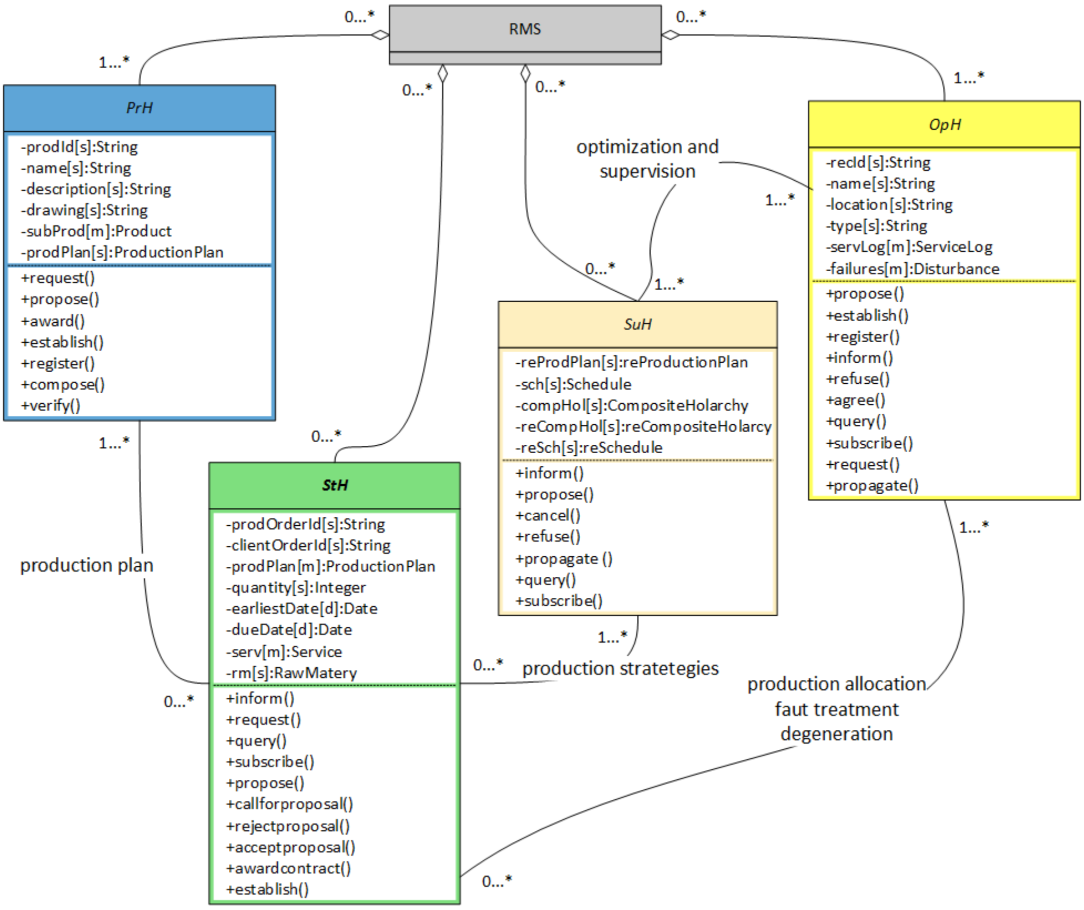


Figure 3. UML class diagram of SOAHCS

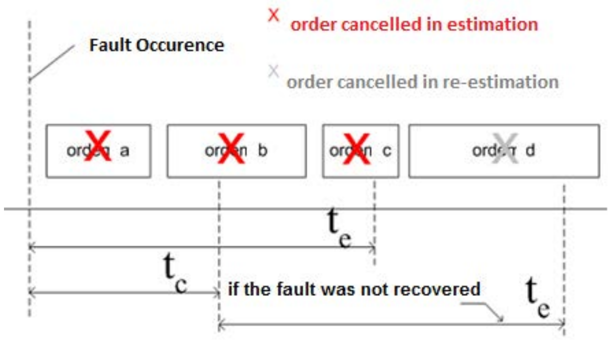
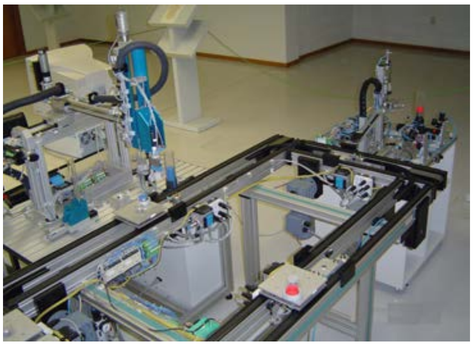


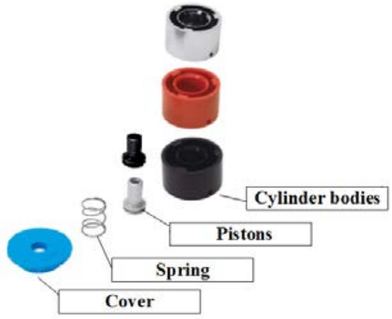
Figure 4. Estimation of the reconfiguration time

Table 1. CNP of SOAHCS adapted from Smith (1980)

HCP	StHs	OpHs
Operation allocation process.	Contracts the operation execution by ξ and the penalty by φ.	Contracts the operation execution by ξ and the penalty by φ.
Finish of an operation with success.	Pays the value ξ to the OpH ($\pi \leftarrow \pi - \xi$).	Increases the total credits by ξ ($\mu \leftarrow \mu + \xi$).
End of an operation with delay	Pays the value ξ and receives the value φ from the OpH ($\pi \leftarrow \pi - \xi + \varphi$).	Decreases the total credits by φ and increase by ξ ($\mu \leftarrow \varphi$).
Operation cancelled (delay, failure, etc.)	Receives the value φ from the OpH ($\pi \leftarrow \pi + \varphi$).	Decreases the total credits by φ ($\mu \leftarrow \mu - \varphi$).

ISSN 2176-5480





(a) (b)

Figure 5. Benchmark dispersed productive system

the system, and cases of reconfiguration.

The purpose of the BDPS is automatically assembling products composed of workpieces (wps), Fig. 5b: a cylinder body (black $[bcb]$, red $[rcb]$ or aluminum $[acb]$), a piston (black $[bp]$ or aluminum $[ap]$), a spring $[s]$ and a cover $[co]$. The aluminum piston is assembled only on black cylinder body while black piston is assembled on red or aluminum cylinder bodies. Springs and covers are the same in all assemblies.

Thus, this RMS joins wps to obtain the final products:

(i) $[bcb + ap + s + co]$;

(ii) $[rcb + bp + s + co]$; and

(iii) $[acb + bp + s + co]$.

The BDPS is composed by six workstations WSs that represent each one an autonomous subsystem: distributing workstation $(D - WS)$, testing workstation $(Te - WS)$, transporting workstation $(Tr - WS)$, handling workstation $(H - WS)$, assembly workstation $(A - WS)$ and robot workstation $(R - WS)$, see diagram in Fig. 6. Work orders (wos) can be executed on each WSs which have its respective controller and, this way, the WS operates independently as a stand-alone industrial plant. Each controller is connected to microcomputer by means of a dedicated hardware. Each microcomputer in turn, is connected to internet allowing operators and clients interact with the system, as in a *automation service cloud.* The objective of each WS is to perform work orders (wo) on workpieces to compose the final products. Each workstation processes the workpieces at each time, and has buffers with limited storage capacity, besides another resources. For each WS should be identified the control devices, their control functions, commands and signals of actuation and detection. The identification of these devices is realized according the specification DIN/ISO 1219-2:1996-11 (DIN – Deutsches Institut für Normung)/(ISO – International Organization for Standardization) and using the codes recommended in the specification IEC 61346-2:2000-12. For example, in the nomenclature "$1S2$": 1 =circuit number, S =device code, and 2 =device number. Table 2 shows the proper primitives to perform the control task to the $D - WS$.

Sub-phase 1.1 – identification of holons – on this sub-phase the holons are identified. The recursive structure (holons made up of holons) allows designing each holon in order to figure out the advantages of decomposing it into a new holarchy. This process is repeated until every holon is completely defined and there is no need for further decompositions. In Tab. 3 are listed some *OpHs* of the BDPS, representing the WSs, which still can be decomposed in another *OpHs* representing their devices.

Holonic processes are production workflow created dynamically and based on the collaboration of the holons. Each *PrH* represents a product and has an internal process flow, the required input types and output types. There are the intermediate *PrHs:* $[bcb]$, $[rcb]$, $[acb]$, $[ap]$, $[bp]$, $[s]$, $[co]$, $[bcb + ap]$, $[rcb + bp]$, $[acb + bp]$, $[bcb + ap + s]$, $[acb + bp + s]$, $[rcb + bp + s]$ and there are three final *PrHs*: $[bcb + ap + s + co]$, $[rcb + bp + s + co]$ and $[acb + bp + s + co]$.

Sub-phase 1.2 – definition of interaction patterns between holons – the synchronization of E-PN models is made by `enabling arcs` and `inhibitor arcs`. The following interactive processes are considered in the modeling SOAHCS: request products, implementation services, fault treatment and reconfiguration. These processes are described using UML diagrams (Booch *et al.*, 2005) and the proposed CNP protocol, following the specifications of FIPA for MAS. Figure 7 presents an example to the interactions of fault treatment in the propagation of messages.

Phase 2 – modeling considering reconfiguration – the AFTCS mechanisms for SOAHCS are modeled on this phase, with the "diagnoser" and the "decider" to fulfill the requirements of the diagnosis and decision phases. Silva *et al.* (2011) present an example and the steps to design the E-PN models of these mechanisms.

Figure 8 illustrates the modeling procedure with follow examples: Fig. 8a has models of the product order for a cylinder body workpiece and following submodels: PFS of the production plan, E-PN of [*executesOp_WS*] of the control objects robot and its controller. It is observed that E-PN of the control object must consider the influence of the transmission of control signals and the states of faults of these objects. Figure 8b presents an example for fault treatment and its degeneration is in Fig. 8c.

ISSN 2176-5480

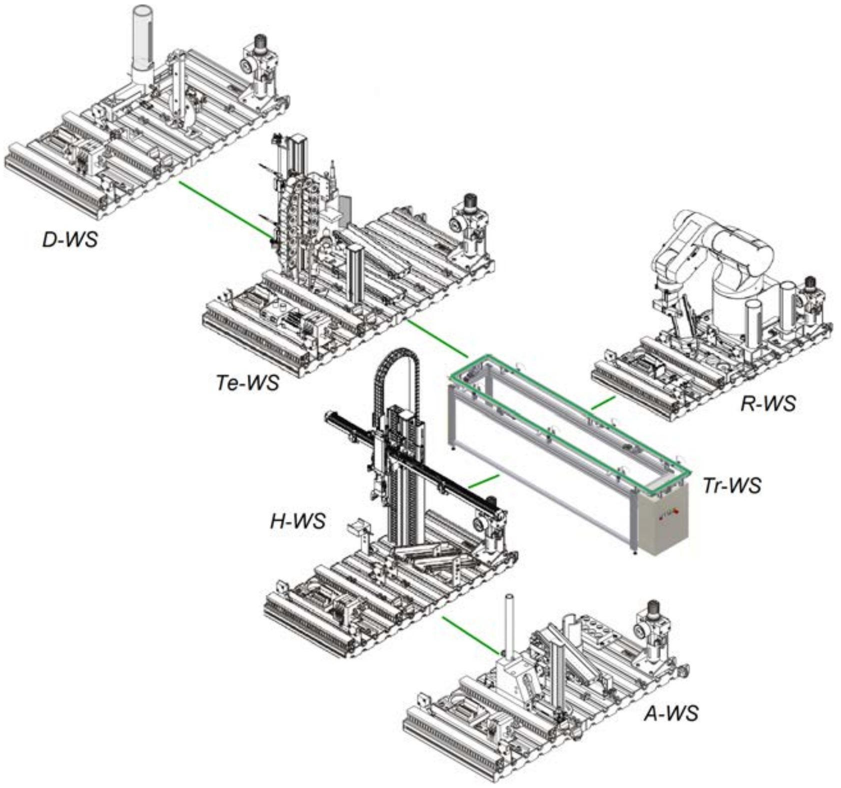


Figure 6. Diagram of benchmark dispersed productive system

Figure 9a illustrates a PFS of the global production process in BDPS, i.e., representing the combination of the intermediates *PrHs* to get the final *PrHs*, and the required input types and output types between the *PrHs*. Each PrH has an internal process flow. To model the workflow of a holon in E-PN, a `place` represents a state in the workflow while a `transition` represents an event or operation that brings the flow from one state to another. Figure 9b exemplify this flow in E-PN for *a PrH*.

The Fig. 10 represents the E-PN and the list of conditions (`places`) and actions (`transitions`) of the control process of D-WS.

Phase 3 – analysis/ simulation – the analysis of the structure and the dynamic behavior are based on E-PN properties. This type of analysis allows re-design and re-engineering of the control system during the design phase. In this paper the software PIPE Bonet *et al.* (2007) is used. This software is a tool for Petri net edition and simulation that has a relatively intuitive interface, facilitating the use and supports the analysis of E-PNs. The qualitative analysis is based on structural analysis of the E-PN models. Quantitative analysis is performed through the simulation of E-PN with `timed transitions`. To this phase, scenarios also are identified with models built for each case. The models must meet the restrictions and achieve the objectives outlined in the hypothesis. Thus, it is also possible to review the control system models identifying the `places` and `transitions` that must be attended in each service.

Table 2. List of devices and functions of the distributing workstation

Nomenclature	Function	Device
1A	removes the workpiece cylinder body of the buffer	pneumatic cylinder double acting
2A	takes the wokpiece cylinder body	vacuum generator and sucker
3A	transports the workpiece cylinder body to the next workstation	pneumatic rotary actuator
1Y1	It modifies the state of the valve-1 for piston retreating position	solenoid
2Y1	It modifies the state of the valve-2 for vacuum generator that activate position	solenoid
2Y2	It modifies the state of the valve-2 for vacuum generator that deactivate position	solenoid
3Y1	It modifies the state of the valve-3 for swivel arm in supply workstation	solenoid
3Y2	It modifies the state of the valve-4 for swivel arm in testing workstation	solenoid
1B1	It detects the piston retreated position	magnetic proximity sensor
1B2	It detects the piston extended position	magnetic proximity sensor
2B1	It detects if the suction gripper caught a workpiece	vacuum sensor
3S1	It detects if the swivel arm is in the testing workstation position	electrical switches limit
3S2	It detects if the swivel arm is in the distributing workstation position	electrical switches limit
1B4	It detects the lack of the workpiece in the magazine	optical sensor

ISSN 2176-5480

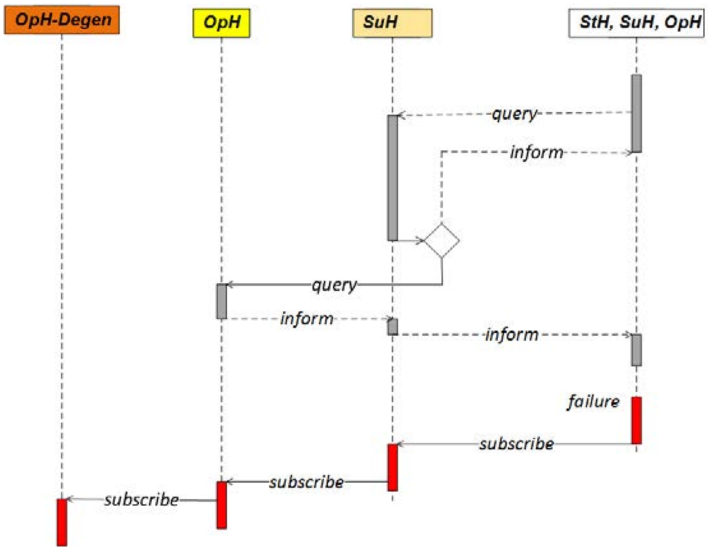


Figure 7. Interactive process of fault treatment

Phase 4 – implementation – for the practical use, the resulting models are interpreted as control program specifications to be performed by computers (supervisory control) and programmable controllers (local control level). Here should be listed the allocation of control signals associated with the programmable controllers. Table 4 lists the addresses of the detectors and sensors following the PLC I/O ports names of the D-WS. For low-level control applications, due its similarities to PN models, the code generation is made following IEC60848 GRAFCET language, a wide used programmable controller language. And the code generation of a high-level language is made in Java using JADE (Java Agent Development Framework) (Bellifemine *et al.*, 1999), a software framework fully implemented in Java language. JADE simplifies the implementation of MAS through a middleware that complies with FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases. Figure 11 shows implementation and JADE fragment code examples. Figure 12 shows the GRAFCET to control the D-WS.

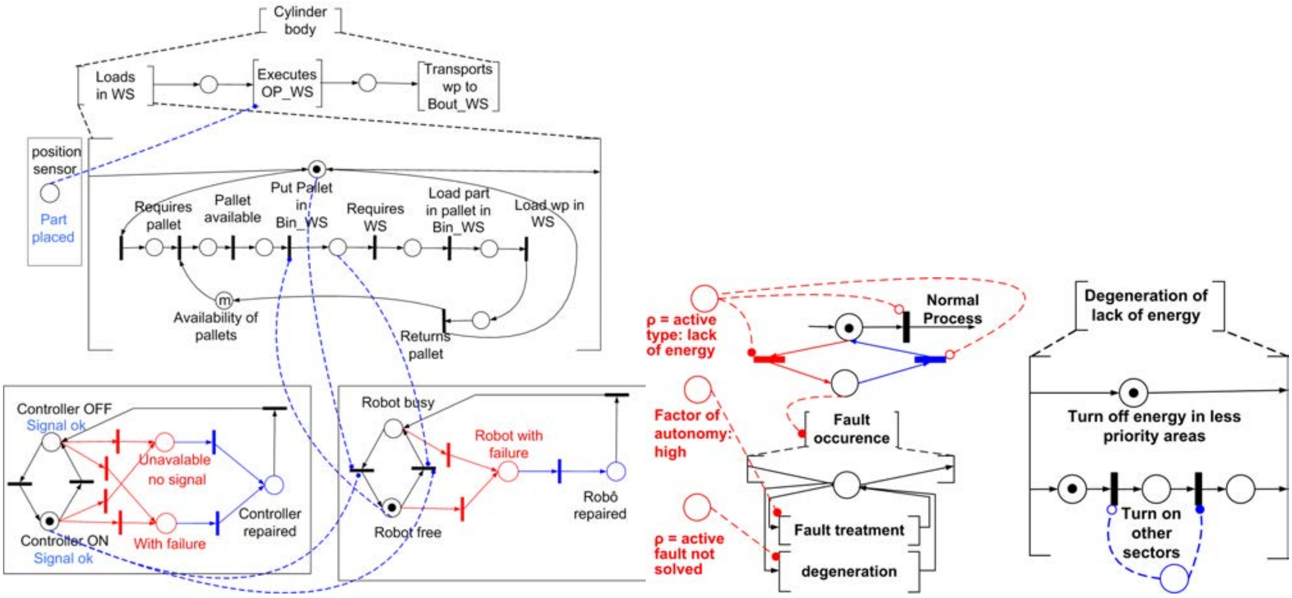
Phase 5 – operation – the real-time monitoring system for supervision and control is accomplished by synchronizing the operation of the E-PN models with sensor signals that represent the devices' state. Depending on the role of the user involved on the operation of the system, the accessible functions and user interfaces must vary. For example, an administrator is responsible for the user's account. An order manager is responsible for creation of a *StH* and re-initialization of the system. A product manager is responsible for definition and creation of *PrHs*. A resource manager is responsible for update of *OpHs*.

The SOAHCS mechanisms allowed the negotiation between holons and the reconfiguration of different productive scenarios with safety and correctness. The comparison of the conditions and actions that need to be met in each service, with `transitions` and `places` of E-PN models facilitated a systematic location of fault in a automatic manner. The reconfiguration is not only applied to resolve occurrence of faults but also to improve the system's performance by increasing the production gain or the number of final products. For example, by controlling the speed of resources through the pneumatic pressure control and by disabling the swivel arm (represented for *OpH*-[sa_{D-WS}]) then its function was assumed through of the *OpH*-[*R-WS*].

Table 3. *OpHs* identification.

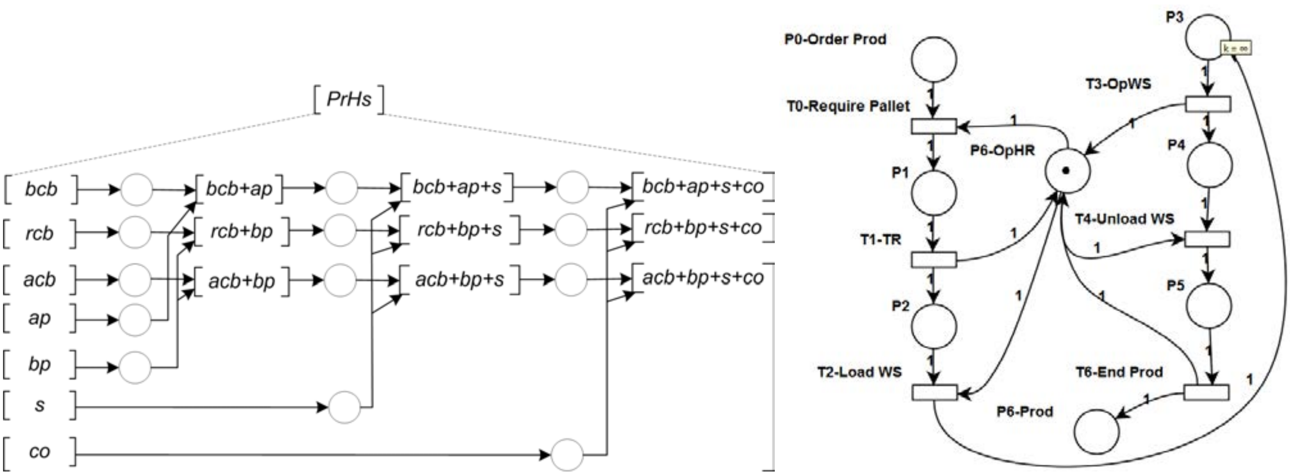
OpH	**description**	**functionality**	**un.**
D-WS	distributing workstation	provides the *wp* cylinder body, which are stored in a buffer with a capacity of eight pieces	1
Te-WS	testing workstation	controls the quality and identifies the physical characteristics of the *wp* cylinder body	1
Tr-WS	transporting workstation	transports/ loads/ unloads workpiece	1
R-WS	robot workstation	transports/ loads/ unloads workpiece of all the RMS	1
H-WS	handling workstation	transports/ loads/ unloads *wp* of the Tr-WS to A-WS and vice-versa, besides it handles *wp* in the A-WS to assembly the final product and dispatches it	1
A-WS	assembling workstation	stores and assembles *wp* to obtain the final product	1
H	human operator	supervision, inspection, maintenance, initialization and finalization operations	1

ISSN 2176-5480



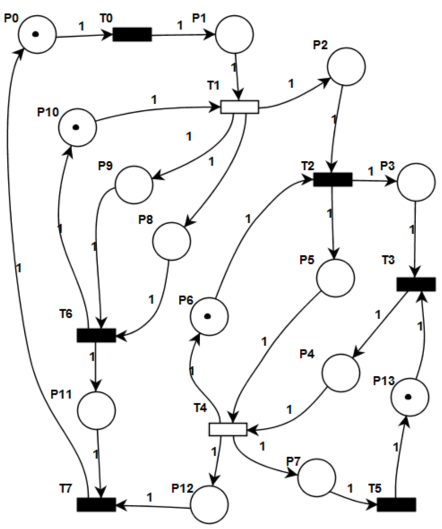
(a) Plan production $wp[cb]$ and control objects models (b) Fault treatment model (c) Degeneration model

Figure 8. Examples of modeling



(a) Global production (b) Workflow process of a PrH

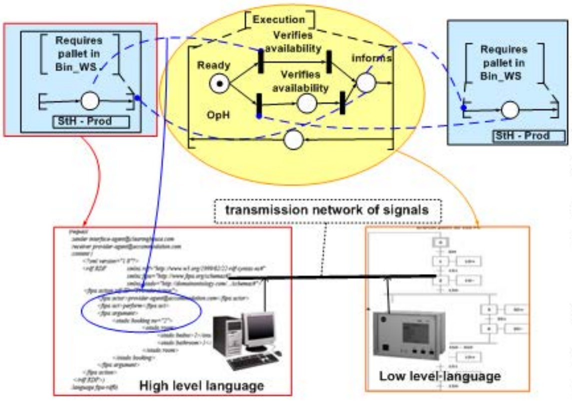
Figure 9. Production process



place/ transition	description
P0	station ready
P1	workpiece sensor detects the presence of the buffer
P10	valve energized to indent the pneumatic piston
P11	pneumatic piston retreat
P12	workpiece in magazine
P2	workpiece positioned to be removed from the next station
P3	swivel arm positioned on the D-WS
P4	suction gripper caught workpiece
P5	valve energized to position swivel arm over D-WS
P6	valve energized to position swivel arm over Te-WS
P7	swivel arm positioned on the Te-WS
P8	Extended pneumatic piston
P9	energized valve to extend pneumatic piston
T0	receives order production
T1	advancing pneumatic piston to remove the workpiece of the buffer
T2	energizing valve to move the swivel arm to magazine of D-WS
T3	energizes the valve that connects the suction
T4	move between stations the swivel arm
T5	deenergizes valve that connects the suction
T6	retreat the pneumatic piston
T7	free D-WS for new cycle

Figure 10. E-PN of the control process of D-WS

ISSN 2176-5480




```
public void action() {
ACLMessagemsg = myAgent.receive();
if (msg != null) {
ACLMessage reply = msg.createReply();
        String content = msg.getContent();
if (content.equalsIgnoreCase("Is it possible to produce product A?")) {
reply.setPerformative(ACLMessage.INFORM);
reply.setContent("I will verify if there is any palet available");
myAgent.send(reply);
System.out.println("The seller "+msg.getSender().getName() + "ordered a product A");
System.out.println("OpH-P:Production Started");
}else{
if(content.equalsIgnoreCase("Paletavailable.Starting Production")){
        String content2 = msg.getContent();
System.out.println("-->"+msg.getSender().getName()+":"+content2);
}else{
if(content.equalsIgnoreCase("Palet is busy")){
        String content2 = msg.getContent();
System.out.println("-->"+msg.getSender().getName()+":"+content2);
```

Figure 11. Example of implementation and JADE fragment code

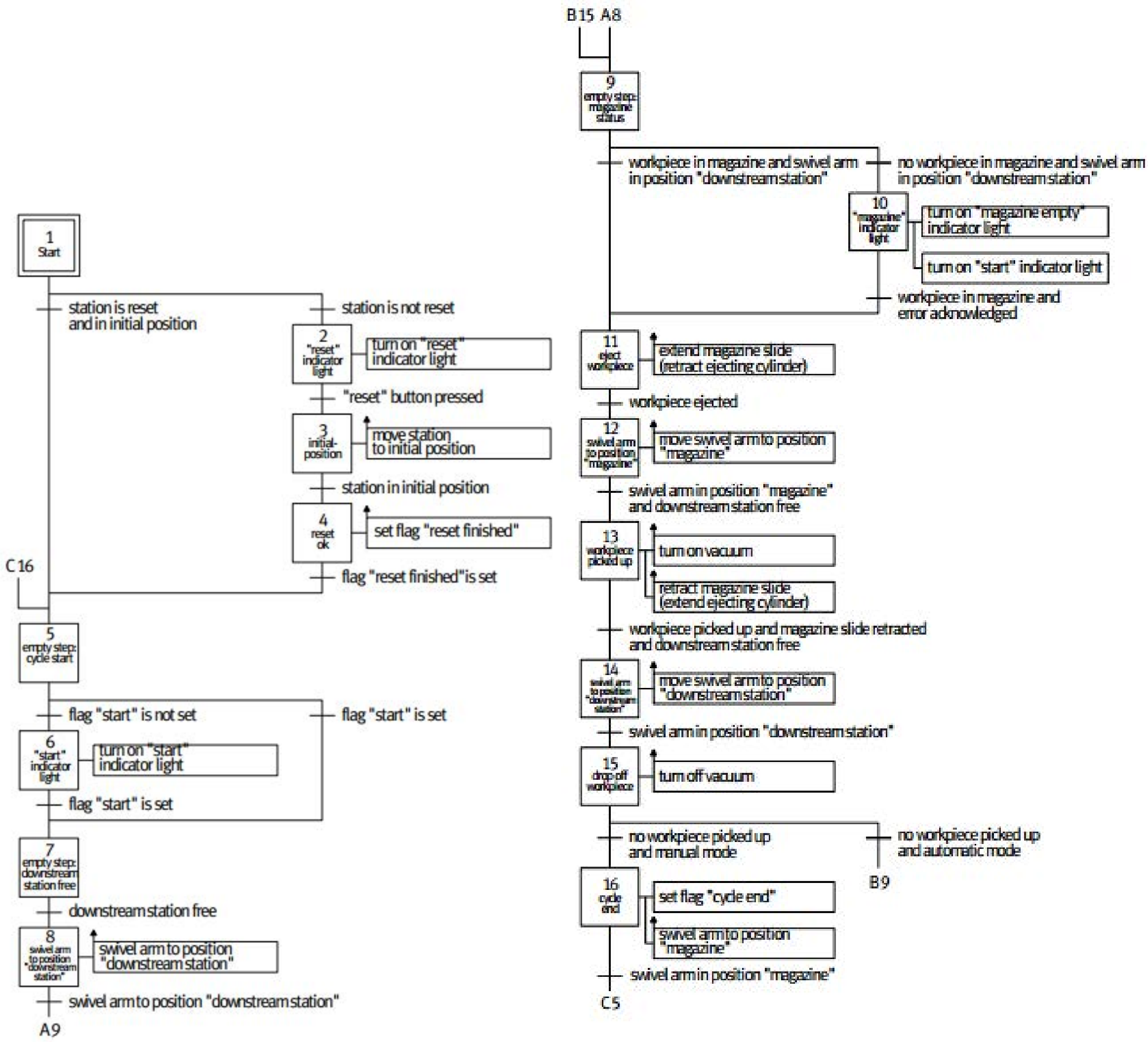


Figure 12. GRAFCET to control *D-WS*

Table 4. Allocation List of the *D-WS*

Address	Symbol	Ident.	Description
I0.1	Mag_back	1B2	Magazine in back position
I0.2	Mag_frnt	1B1	Magazine in front position
I0.3	Sen_vac	2B1	Sensor vacuum is on
I0.4	Arm_take	3S1	Swivel drive at magazine
I0.5	Arm_put	3S2	Swivel drive at the next station
I0.6	Mat_sen	B4	ON=no workpieces in the feeder
I0.7	Follow	IP_FL	Light barrier to the following station
O0.0	Feed	1Y1	Solenoid of the magazine cylinder
O0.1	Vacumon	2Y1	Solenoid switch on the vacuum
O0.2	Vacumoff	2Y2	Solenoid switch off the vacuum
O0.3	Armleft	3Y1	Solenoid swivel drive to magazine
O0.4	Armright	3Y2	Solenoid swivel drive to next station

4. CONCLUSIONS

This paper presents service-oriented active holonic control system (SOAHCS), a systematized procedure that describes the required data and technical to design control systems for reconfigurable manufacturing system (RMS).

SOAHCS extends previous studies (Silva *et al.*, 2012a,b,d, 2011) combining the service-oriented architecture (SOA) technique with active holonic control system (AHCS).

SOAHCS contributes for the technological innovation in methods design, because it regards requirements specifications to machine operation and maintenance, to assure flexibility, efficiency and robustness. The proposed procedure:

(i) integrates the design of the entire life cycle of control systems for RMS;

(ii) combines bottom–up and top–down approaches using extensions of Petri net (PN): the extended PN (E-PN) for functional description and production flow schema (PFS) for conceptual representation;

(iii) has mechanisms for quicker reaction to critical and noncritical faults, i.e., reconfiguration and degeneration mechanisms ensuring implementation of a hierarchical or heterarchical control structure;

(iv) proposes solutions to holonic control system, such as, a mechanisms to holarchy composition problem (HCP) and better collaboration of the holons based on contract net protocol (CNP).

Different scenarios were elaborated for running an application example of manufacturing system that emulates a dispersed productive system. SOAHCS responded in faster and collaborative manner and useful for both to protect the system when hardware problems occur as to implement different thresholds of production. SOAHCS demonstrated operational advantages such as, better and more efficient use of manufacturing resources, speed of production and ability to deliver products faster.

SOAHCS is presented in generic form and it can be tailored for specific RMS applications. A larger project is being developed (Silva *et al.*, 2011), which involves in addition to modeling, simulation and validation of E-PN models, tools for designer, and others case studies of SOAHCS.

5. ACKNOWLEDGEMENTS

The authors would like to thank the partial financial support of the Brazilian governmental agencies: CNPq, CAPES, FAPESP and the universities: UESC, USP.

6. REFERENCES

Ali, S.A., Seifoddini, H. and Sun, H., 2005. "Intelligent modeling and simulation of flexible assembly systems". In *Proceedings of the 37th conference on Winter simulation*. Winter Simulation Conference, pp. 1350–1358.

Bellifemine, F., Poggi, A. and Rimassa, G., 1999. "JADE–A FIPA-compliant agent framework". In *Proc. of PAAM 4th International Conference on Practical Application of Intelligent Agents and Multi-Agent Technology*. Vol. 99, pp. 97–108.

Bonet, P., Lladó, C.M., Puijaner, R. and Knottenbelt, W.J., 2007. "Pipe v2. 5: A petri net tool for performance modelling". In *Proc. of CLEI 23rd Latin American Conference on Informatics*.

Booch, G., Rumbaugh, J. and Jacobson, I., 2005. *Unified Modeling Language User Guide*. Addison-Wesley.

Colombo, A.W., Neubert, R. and Schoop, R., 2001. "A solution to holonic control systems". In *In: Proc. of ETFA 8th IEEE InternationalConference on Emerging Technologies and Factory Automation*. Vol. 2, pp. 489–498.

Cooper, R., 1988. "The rise of activity-based costing. part one: what is an activity-based cost system?" *Journal of Cost Management*, Vol. 2, No. 2, pp. 45–54.

David, R. and Alla, H., 1994. "Petri nets for modeling of dynamic systems: a survey". *Automatica*, Vol. 30, No. 2, pp.

175–202. doi:10.1016/0005-1098(94)90024-8.

Fattori, C.C., Junqueira, F., Santos Filho, D.J. and Miyagi, P.E., 2011. "Service composition modeling using interpreted petri net for system integration". In *Mechatronics (ICM) IEEE International Conference on Mechatronics*. pp. 696–701.

Ferber, J., 1999. *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley Longman Publishing Co., Inc.

Fipa, A., 2002. "Foundation for intelligent physical agents". *http://www.fipa.org*.

Garcia Melo, J.I., Junqueira, F., Morales, R.A. and Miyagi, P.E., 2008. "A procedure for modeling and analysis of service-oriented and distributed productive systems". In *International Conference on Automation Science and Engineering*. IEEE, Washington, pp. 941–946. doi:10.1109/COASE.2008.4626561.

Groba, C., Braun, I., Springer, T. and Wollschlaeger, M., 2008. "A service-oriented approach for increasing flexibility in manufacturing". In *International Workshop on Factory Communication Systems*. IEEE, pp. 415 –422. doi: 10.1109/WFCS.2008.4638735.

Han, R., Liu, K., Ju, Y. and Zhao, J., 2008. "A Petri net theory-based method for modeling web service-based systems". In *International Conference on Wireless Communications, Networking and Mobile Computing*. pp. 1–7. doi: 10.1109/WiCom.2008.2839.

Koestler, A., 1968. "The ghost in the machine."

Mehrabi, M.G., Ulsoy, A.G. and Koren, Y., 2000. "Reconfigurable manufacturing systems and their enabling technologies". *International Journal of Manufacturing Technology and Management*, Vol. 1, No. 1, pp. 114–131.

Mendes, J.M., Leitão, P., Restivo, F. and Colombo, A.W., 2009. "Service-oriented agents for collaborative industrial automation and production systems". In *Holonic and Multi-agent Systems for Manufacturing*, Springer, pp. 13–24.

Miyagi, P.E., Hasegawa, K. and Takahashi, K., 1988. "A programming language for discrete event production systems based on production flow schema and mark flow graph". *Trans. of the Society of Intrument and Control Engineers (SICE Japan)*, Vol. 24, No. 2, February, pp. 77–84.

Mowbray, T.J. and Zahavi, R., 1995. *The essential CORBA: systems integration using distributed objects*. John Wiley & Sons, Inc.

Murata, T., 1989. "Petri nets: properties, analysis and applications". *Proceedings of the IEEE*, Vol. 77, No. 4, pp. 541–580.

Nagorny, K., Colombo, A.W. and Schmidtmann, U., 2012. "A service-and multi-agent-oriented manufacturing automation architecture: An iec 62264 level 2 compliant implementation". *Computers in Industry*.

Silva, R.M., Arakaki, J., Junqueira, F., Santos Filho, D.J. and Miyagi, P.E., 2012a. "Modeling of active holonic control systems for intelligent buildings". *Automation in Construction*, Vol. 25, pp. 20–33.

Silva, R.M., Arakaki, J., Junqueira, F., Santos Filho, D.J. and Miyagi, P.E., 2012b. "A procedure for modeling of holonic control systems for intelligent building (hcs-ib)". *Advanced Materials Research*, Vol. 383, pp. 2318–2326.

Silva, R.M., Junqueira, F., Santos Filho, D.J. and Miyagi, P.E., 2012c. "Modeling of holonic manufacturing control systems considering reconfiguration and timing constraints". In *Proceedings of CBA, XIX Congresso Brasileiro de Automática, Campina Grande, PB*.

Silva, R.M., Miyagi, P.E. and Santos Filho, D.J., 2011. "Design of active holonic fault-tolerant control systems". *Springer IFIP Advances in Information and Communication Technology*, Vol. 349, pp. 367–374.

Silva, R.M., Junqueira, F., dos Santos Filho, D.J. and Miyagi, P.E., 2012d. "Design of reconfigurable and collaborative control system for productive systems". *ABCM Symposium Series in Mechatronics. 1ed.Rio de Janeiro, RJ*, Vol. 5, pp. 813–822.

Smith, R., 1980. "The contract net protocol: High-level communication and control in a distributed problem solver". *IEEE Transactions on Computers*, Vol. 100, No. 12, pp. 1104–1113.

Squillante, R., Santos Filho, D.J., Riascos, L.A.M., Junqueira, F. and Miyagi, P.E., 2012. "Mathematical method for modeling and validating of safety instrumented system designed according to IEC61508 and IEC61511". *ABCM Symposium Series in Mechatronics*, Vol. 5, No. 2, pp. 484–493.

Tommila, T., Hirvonen, J., Jaakkola, L., Peltoniemi, J., Peltola, J., Sierla, S. and Koskinen, K., 2005. "Next generation of industrial automation concepts and architecture of a component-based control system". *VTT Tiedotteita. Research Notes*, Vol. 2303.

Vrba, P., Tichý, P., Mařík, V., Hall, K.H., Staron, R.J., Maturana, F.P. and Kadera, P., 2011. "Rockwell automation's holonic and multiagent control systems compendium". *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, Vol. 41, No. 1, pp. 14–30.

Zhang, Y. and Jiang, J., 2008. "Bibliographical review on reconfigurable fault-tolerant control systems". *Annual Reviews in Control*, Vol. 32, No. 2, pp. 229–252.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.