A Service-Oriented and Holonic Control Architecture to the Reconfiguration of Dispersed Manufacturing Systems

Robson Marinho da Silva[1,2], Mauricio F. Blos[2], Fabrício Junqueira[2], Diolino J. Santos Filho[2], and Paulo E. Miyagi[2]

[1] Universidade Estadual de Santa Cruz, Ilhéus, BA, Brazil,
rmsilva@uesc.br
[2] Universidade de São Paulo, SP, Brazil,
{blosmauf,fabri,diolinos,pemiyagi}@usp.br

Abstract. Manufacturing control systems must quickly react to variations of product, process specifications, fault occurrence, changes in the resources functional capabilities, and other operational demands. Besides, to gain competitive advantages, dispersed manufacturing systems must cooperate with each other. The combination of holonic control system and service-oriented architecture techniques can be effective to integrate these heterogeneous environments since the agents must also cooperate to achieve their services. Therefore, this paper introduces a service-oriented and holonic control architecture using Petri net to represent the workflow of design method itself, structure and dynamic of the entities. The modeling of an example is presented to demonstrate application at self-organization and adaptation, fault treatment, degeneration, and relatively shorter implementing time.

Keywords: reconfiguration, holonic control system, service-oriented architecture, dispersed manufacturing system.

1 Introduction

Manufacturing systems have evolved from mass production, lean manufacturing and flexible manufacturing to the reconfiguration of dispersed manufacturing systems (RDMS) which is conceived to be “adjustable" into pursuance of the businesses processes. Reconfiguration is associated with changes (i) in addition, modification and removal of some part of a workflow; (ii) in processing time, number and availability of existing resources; and (iii) in development of mechanisms for fault treatment [1].

A workflow management framework must be able to deal with the changes mentioned. The holonic control system (HCS) technique [2], [3], [4] allows the integration of heterogeneous environments. Furthermore, the service-oriented architecture (SOA) [5], [6], [7] has been used to undertake inter-enterprise collaboration.

Hence, this paper introduces a service-oriented and holonic control architecture (SOHCA) and its method to design RDMS. By exploring the potential of techniques which ensure the integration of design, the method uses the Petri net markup language (PNML) [8] and production flow schema (PFS) [9] techniques.

L.M. Camarinha-Matos et al. (Eds.): DoCEIS 2014, IFIP AICT 423, pp. 111–118, 2014.

2 Relationship to Collective Awareness Systems

Collective awareness systems (CAS) allow its members to work together to improve self-organization and adaptation capacity. This is possible by integration of value-added activities, information, resources and knowledge between heterogeneous businesses workflow. RDMS is also related to the effective degree of collaboration, because its subsystems need to be aware of each other to attain the global objective [10].

We exploit HCS and SOA techniques to automatically allow the RDMS and provide a middleware for integration between the factory control and shop floor control layers in dispersed systems. SOHCA and its design method enable designers to develop collaboration and sequence diagrams, which assimilate the implementation of CAS characteristics, regarding workarounds to: (i) develop supervisory control system with interoperability and portability; (ii) propose strategies to recover the functionality of the system or maintain operations so that parts affected by the fault are disabled without affecting other parts (also called degeneration), and (iii) allow designers to create workflow for adding or modifying the productive processes, regarding the reuse of models to avoid repetition and overlapping tasks.

3 State of the Art and Related Works

There are already many suggestions applying the concept of holons or agents with different scope, focus and methodology [1], [2], [3], [4], [10]. Mendes *et al.* [5] 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.* [6] proposed the use of hardware and software accessed remotely, and where operational activities are described as services. Morariu *et al.* [7] present the design and implementation of the customer order management module which integrates HCS with SOA enabled shop floor devices using industry standards. Furthermore, control architecture should be based on standards ontologies, as proposed in ANSI/ISA 88/95[11] and ADACOR [10]. In previous studies [12], [13], [14], we applied the HCS with fault tolerance mechanisms for intelligent building, while in [15] our methodology was applied for manufacturing system (MS) and for the description of the control mode switching between two operational modes.

We observe that: (i) a suitable ontology can provide a semantic model for interaction in RDMS; (ii) a trading mechanism is more favorable than a request-response communication format between holons; (iii) patterns emerge without an environment that facilitates the development of new models; (iv) there are few proposals regarding strategies for the system reconfiguration; (v) another challenge is how to compose holarchies that accomplish a productive task with constraints to consider the lead time; and (vi) there are still a small number of applications combining HCS and SOA concepts, or more specifically with RDMS.

4 SOHCA and Proposed Design Method

The proposed method to design RDMS is structured in stages (Fig. 1): *analysis of requirements; modeling; analysis of models; implementation; operation*; and *integration*. The *integration* stage ensures a closed-loop for re-design and re-engineering.

In the *analysis of requirements* stage specifications are defined, such as: aim, control devices, orders, tasks, operations, interactions between parts of the system, and strategies for the cases of fault treatment and degeneration. Based on this survey, the holons are identified. Figure 2 illustrates the SOHCA holons. The *product holon* (*PrH*) manages the requests of products, searches for the corresponding holons, creates a work order (*wo*) comprising the product type and the amount. The set of *PrHs* composes the *production plan*, which represents the union, transformation and operation of the intermediated products to get the final products, i.e., the required input types and output types between *PrHs*. The *task holon* (*TH*) manages the recipe for business processes, such as, manufacturing, work order, operation and resource; and the reconfiguration strategies. The *OpH* represents the resources, such as humans or equipment (control objects). The supervisor holon (*SuH*) contains all the knowledge to coordinate the services of all holons.

At the *modeling* stage, the PFS and PN models represent the processes. To model the workflow of a holon, a ***place*** (terms of PN are ***highlighted***) represents a state in the workflow while a ***transition*** represents an event or operation that conducts the flow from one state to another. The synchronization of PN models is made by ***enabler*** and ***inhibitor arcs***, as well as by ***auxiliary places***. To process an order, each holon is modeled as an atomic service that can be combined with atomic services by other holons. *OpHs* models include the control object which must regard fault states of these objects. The *PrH*, *TH* and *SuH* specify a set of operations, precedence constraints of the operations and control supervisory. PN models are used to represent the holons workflow and activities. The calculations of timing constraints are made using the ***temporized transitions*** in PN models.

In the *analysis of models* stage structure and dynamic behavior of the models are simulated and verified. The qualitative analysis is based on structural analysis using editors of PN models. Quantitative analysis is performed through the simulation of ***temporized transitions***. At this stage, scenarios are identified with models built for each one. The models must meet the restrictions and achieve the objectives outlined in the hypothesis. It is also possible to review the control system models identifying the ***places*** and ***transitions*** that must be considered in each service.

The *implementation and operation* stages move toward the implementation and real-time monitoring system for supervision and control aspects. For low-level control applications, the code generation is made following the IEC60848 GRAFCET language. The code generation of a high-level language is made in Java using JADE and its extensions [15]. The real-time monitoring system is accomplished by synchronizing the operation of the PN models with sensor signals that represent the devices state. Communication channels are specified in the SOHCA ontology, a Java method is called and it creates the channels with the specified IP addresses and ports, while other holons associate a behavior to the specified port for each communication channel.

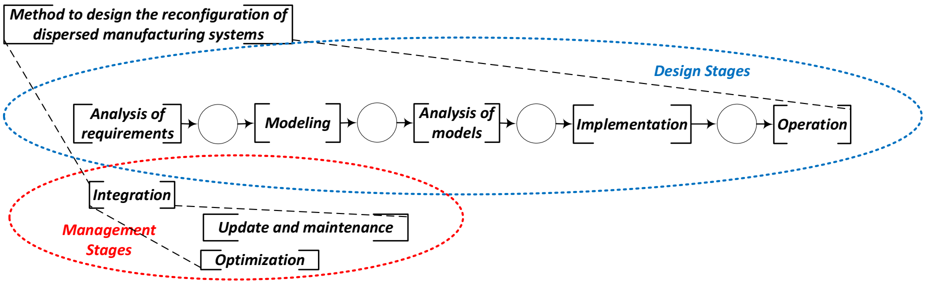


Fig. 1. Schema to design the reconfiguration of dispersed manufacturing systems

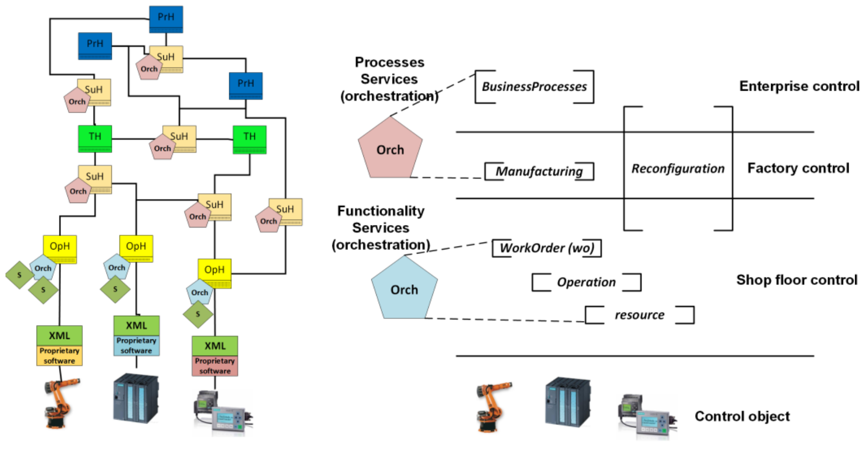


Fig. 2. SOHCA. Holarchies are formed (orchestration) to represent the control levels. The system is represented by the composition of these holarchies (choreography). Orchestration and choreography are terms related to SOA technique.

5 Application Example

Here, a benchmarking dispersed manufacturing system (BDMS), illustrated in Fig. 3, is used to describe the application of the proposed method. For each web service (WS) the devices, their control functions, commands and signals of actuation and detection are identified. The identification is made according to the specification DIN/ISO 1219-2:1996-11 and the codes recommended in specification IEC 61346-2:2000-12. For example, in the nomenclature 1S2: 1 =circuit number, S =device code, and 2 =device number.

The ***temporized transitions*** are used for calculating the holarchies formation. For example, let $[H_n]$ be the production sequence n formed by *PrHs* to obtain a final product. Let us consider the following sequences (the caption of Fig. 3 lists the abbreviations used): $[H_1]$: $[bcb] \rightarrow [bcb + ap] \rightarrow [bcb + ap + s] \rightarrow [bcb + ap + s + co]$ and $[H_2]$: $[rcb] \rightarrow [rcb + ap] \rightarrow [rcb + ap + s] \rightarrow [rcb + ap + s + co]$.

Suppose the lead time set by $PrH-[bcb+ap+s+co]$ is $t_e[bcb+ap+s+co]$. This imposes a timing constraint on $PrH{-}[bcp+ap+s]$ in $[H_1]$ sequence. Let $t_e[bcb+ap+s\ ]$ be the latest time that $PrH{-}[\ bcb+ap+s]$ must complete all its operations. Let $t_e[t_n]$ be the time in which ***transition*** n must complete its operations. To meet the due date, the constraint of $t_e[bcb+ap+s] \leq t_e[bcb+ap+s+co] + t_e(t_1+t_2+\cdots+t_n)$ must be satisfied by $PrH{-}[bcb+ap+s]$ and so on for other holons, i.e., $t_e[bcb+ap+s]$ must be less than or equal to $t_e[bcb+ap+s+co]$ minus the sum of the times of ***transitions*** of the $PrH{-}[bcb+ap+s+co]$. A variable named cost Hn is associated to 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 PN model. For composing holarchies, SOHCA compares the cost of $C_{[H1]} = C_{[bcb]} + C_{[bcb+ap]} + C_{[bcb+ap+s]} + C_{[bcb+ap+s+co]}$ and $C_{[H2]} = C_{[rcb]} + C_{[rcb+ap]} + C_{[rcb+ap+s]} + C_{[rcb+ap+s+co]}$ to decide what is the better sequence for production is at a certain time.

Figure 4 has PN and PFS models for some holons and their workflow. Figure 5 presents the production plan, and reconfiguration examples. Figure 6a shows the PFS and PN models, and the programmable logic controller (PLC) I/O addresses list of some detectors and sensors of the *D-WS*. The mapping between the PFS and the JAVA implementation is depicted in Fig. 6b. Scenarios were simulated to validate the advantages of the reconfiguration and degeneration mechanisms. For example, by controlling the production speed through the pneumatic pressure and by disabling the swivel arm (represented by *OpH*, [$sa_{D\ WS}$]) then its function was assumed by another *OpH, [R-WS]*.

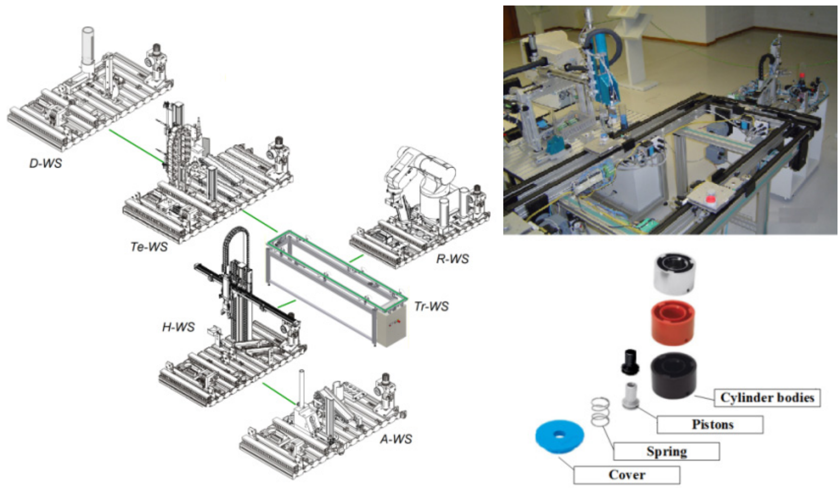


Fig. 3. BDMS composed of workstations (*WSs*): distributing (*D-WS*), testing (*Te-WS*), transporting (*Tr-WS*), handling (*H-WS*), assembling (*A-WS*) and robot (*R-WS*). The production plan is joining work pieces (*wps*): a cylinder body (black [*bcb*], red [*rcb*] or aluminum [*acb*]), a piston (black [*bp*] or aluminum [*ap*]), a spring [*s*] and a cover [*co*].

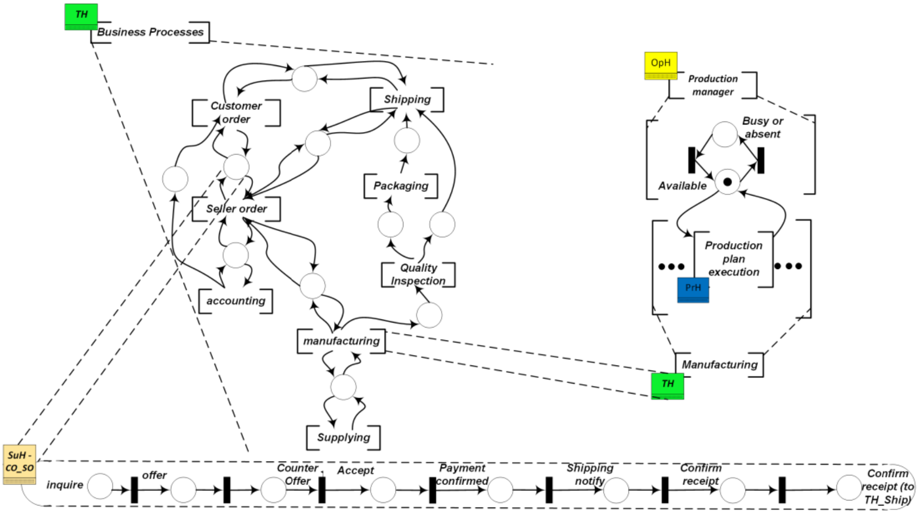


Fig. 4. Choreography of *THs-[Business Processes]*, an orchestration of *OpH* and *TH*, and the detailing of *SuH-[CO_SO]* for the message exchange between the holons

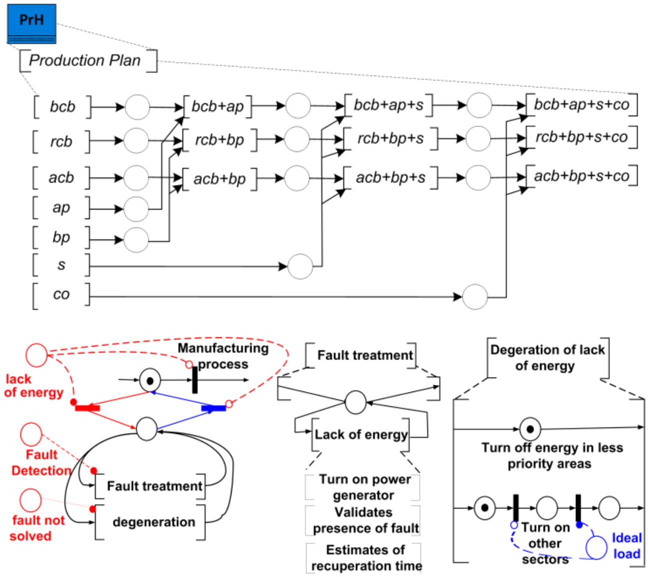
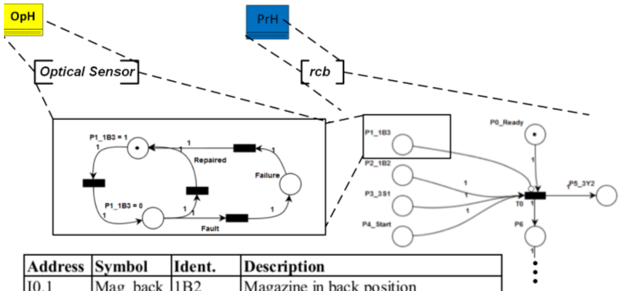
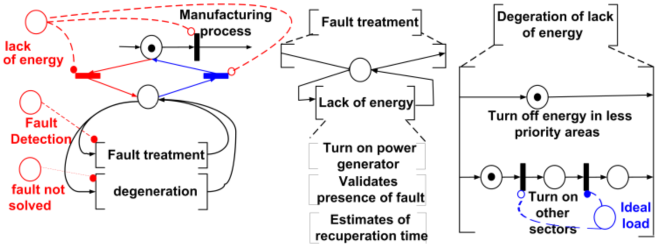


Fig. 5. PrH *[production plan]*, fault treatment and degeneration workflow



Address	Symbol	Ident.	Description
I0.1	Mag_back	1B2	Magazine in back position
I0.4	Arm_take	3S1	Swivel drive at magazine
I0.6	Mat_sen	1B3	ON=no workpieces in the feeder
I0.7	Start	Start	Start key
O0.4	Armright	3Y2	Solenoid swivel drive to next station

(a) *OpH-[optical sensor]*, OpH -[*rcb*] and the addresses list of PLC



(b) PN, PFS of the *TH-[Accounting]* and the transformation into JAVA language

Fig. 6. Examples of *implementation stage*

6 Conclusions and Further Work

This paper presented service-oriented holonic control architecture (SOHCA), and its method that describes the required data and techniques to design supervisory control system to the reconfiguration of dispersed manufacturing systems (RDMS). Different scenarios were elaborated for running an application example of a benchmarking dispersed manufacturing system. The scenarios met the restrictions and achieved the objectives outlined in the hypothesis. SOHCA responded in a faster and in a collaborative manner and showed to be useful to protect the system when hardware problems occur, implemented different thresholds of production and demonstrated operational advantages, such as better and more efficient use of manufacturing resources, production speed and ability to deliver products faster. The qualitative analysis was based on structural analysis using PN editors and quantitative analysis was performed through the simulation with ***temporized transitions***. It was also possible to review the control system models identifying the places and transitions that must be considered in each service. The reconfiguration was not only applied to solve fault occurrence. It also was applied to improve the system performance by increasing the production gain or the number of final products. SOHCA is generic and developed based on SOA and HCS techniques; it can be tailored for specific manufacturing applications. A larger project is being developed which involves in addition to modeling, simulation and validation of PN models, ontology description, tools for designer, and other case studies.

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

References

1. Hsieh, F.S., Lin, J.B.: A self-adaptation scheme for workflow management in multi-agent systems. Journal of Intelligent Manufacturing, 1–18 (2013)
2. Brennan, R., Fletcher, M., Norrie, D.: An agent-based approach to reconfiguration of real-time distributed control systems. IEEE Transactions on Robotics and Automation 18(4), 444–451 (2002)
3. Chirn, J., McFarlane, D.: A holonic component-based approach to reconfigurable manufacturing control architecture. In: Proceedings of 11th International Workshop on Database and Expert Systems Applications, pp. 219–223. IEEE (2000)
4. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.: Reference architecture for holonic manufacturing systems: PROSA. Computers in Industry 37(3), 255–274 (1998)
5. Mendes, J.M., Leitão, P., Restivo, F., Colombo, A.W.: Service-oriented agents for collaborative industrial automation and production systems. In: Mařík, V., Strasser, T., Zoitl, A. (eds.) HoloMAS 2009. LNCS, vol. 5696, pp. 13–24. Springer, Heidelberg (2009)
6. Nagorny, K., Colombo, A.W., Schmidtmann, U.: A service-and multi-agent-oriented manufacturing automation architecture: An IEC 62264 level 2 compliant implementation. Computers in Industry (2012)
7. Morariu, C., Morariu, O., Borangiu, T.: Customer order management in service-oriented holonic manufacturing. Computers in Industry 64(8), 1061–1072 (2013)
8. Billington, J., Christensen, S., van Hee, K.M., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C., Weber, M.: The Petri Net Markup Language: Concepts, Technology, and Tools. In: van der Aalst, W.M.P., Best, E. (eds.) ICATPN 2003. LNCS, vol. 2679, pp. 483–505. Springer, Heidelberg (2003)
9. Hasegawa, K., Miyagi, P.E., Santos Filho, D.J., Takahashi, K., Ma, L., Sugisawa, M.: On resource arc for Petri net modelling of complex resource sharing system. Journal of Intelligent and Robotic Systems 26(3-4), 423–437 (1999)
10. Leitão, P., Restivo, F.: Adacor: A holonic architecture for agile and adaptive manufacturing control. Computers in Industry 57(2), 121–130 (2006)
11. Instrumentation, S., Society, A.: Enterprise-control system integration: Part 1:models and terminology. 95.00.01 (2000)
12. da Silva, R.M., Miyagi, P.E., Santos Filho, D.J.: Design of active holonic fault-tolerant control systems. In: Camarinha-Matos, L.M. (ed.) Technological Innovation for Sustainability. IFIP AICT, vol. 349, pp. 367–374. Springer, Heidelberg (2011)
13. Silva, R.M., Arakaki, J., Junqueira, F., Santos Filho, D.J., Miyagi, P.E.: A procedure for modeling of holonic control systems for intelligent building (HCS-IB). Advanced Materials Research 383, 2318–2326 (2012)
14. Silva, R.M., Arakaki, J., Junqueira, F., Santos Filho, D.J., Miyagi, P.E.: Modeling of active holonic control systems for intelligent buildings. Automation in Construction 25, 20–33 (2012)
15. Silva, R.M., Junqueira, F., dos Santos Filho, D.J., Miyagi, P.E.: Design of reconfigurable and collaborative control system for productive systems, 1st edn. ABCM Symposium Series in Mechatronics, pp. 813–822. Rio de Janeiro, RJ (2012)