Modeling of Mechanisms for Reconfigurable and Distributed Manufacturing Control System

Robson Da Silva, Edson H. Watanabe, Maurício F. Blos, Fabrício Junqueira, Diolino J. Santos Filho, Paulo E. Miyagi

Modeling of Mechanisms for Reconfigurable and Distributed Manufacturing Control System

Robson Marinho da Silva[1,2], Edson H. Watanabe[2], Maurício F. Blos[2]
Fabrício Junqueira[2], Diolino J. Santos Filho[2], Paulo E. Miyagi[2]

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

Abstract. This paper presents the modeling of fault diagnosis mechanisms extending a method to design reconfigurable and distributed manufacturing control system. The method combines different techniques: service-oriented architecture, holonic and multi-agent system, production flow schema and input-output place-transition Petri net. An application example demonstrates advantages of the proposal, such as at cloud-based engineering, reuse, implementation and reconfiguration flexibility.

Keywords: reconfigurable control system; service-oriented architecture; holonic and multi-agent system; Petri net; cloud manufacturing system.

1 Introduction

Reconfigurable control systems have been proposed to implement manufacturing systems with different strategies because its business processes can modify due new demands and in case of fault occurrence [1]. Furthermore, geographically distributed manufacturing subsystems need communicating and collaborating with each other in a dynamic environment. To propose a feasible solution to design Reconfigurable and Distributed Manufacturing Control System (RDMCS), characteristics of different techniques should be combined. In [1,2], a method which associates top-down and bottom-up approaches through Petri Net (PN) [3, 4] combining Service-Oriented Architecture (SOA) [5] and Holonic and Multi-Agent System (HMAS) [6] concepts to model RDMCS was presented. The HMAS concept facilitates the development and integration of distributed and heterogeneous systems and SOA paradigm allows considering interoperability with other manufacturing systems. This paper extends that method to model RDMCS focusing on modeling of fault diagnosis mechanisms and describes an application example.

2 Relationship to Cloud-based Solutions

In preview studies, the HMAS, SOA and PN concepts are combined to propose a method to design RDMCS [1, 2]. The basic assumption of this research is meeting the needs of goal-directed behavior in a distributed, dynamic and unpredictable environment, using patterns to facilitate the development of new models and a suitable semantic. The SOA paradigm is applied to define interfaces for communication in models based on HMAS concepts. The semantic definition of business processes is combined in a top-down and bottom-up approach based on PN to organize specifications, reuse and integrate social structure. The dynamic models are edited using a web tool for Input Output Place Transition (IOPT) PN [7] which has modules for edition and simulation.

Therefore, according to cloud-based manufacturing statements [8], this method is related to this vision due characteristics as: (i) avoiding repetition and overlapping tasks through reuse; (ii) facilitating the analysis and communication between different de sign staffs through PN models and their available web tools; (iii) allowing reconfiguration flexibility for both rational use of resources and larger safety to fault detection through proposed control mechanisms; (iv) considering human interaction in dynamic environment of productive processes; and (v) allowing design a large complex system through modular models and collaboration between distributed subsystem.

3 State of the Art and Related Works

An overview of methodologies, architectures and applications as well as trends and challenges in HMAS are presented in [9]. Some architectures have a product-oriented vision and pay more attention to scalability and generic components of the system [10]. Others have a machine-oriented vision, paying more attention to an optimal utilization of the machine level than to the specific product performance [6]. However, in real applications much more complex and/or distributed subsystems may be combined into larger systems. On the other hand, SOA has revolutionized the distributed computing paradigm and made various kinds of inter-enterprise collaboration a reality [5]. Although SOA provides the framework for integrating cross-company services or technical interoperability, it does not address formal methods to ensure a systematic implementation of the design specifications [11]. According to [1, 3], the PFS associated to ordinary PN is a solution for transformation of abstraction into application level, a gap into design of discrete event systems and specifically in RDMCS. Furthermore, tools for edition of IOPT PN [4, 7] can be used to model a control system with the additional advantage that the models can quasi-automatically be converted to programs for industrial controllers.

4 Modeling of Diagnosis Mechanism for RDMCS

Modeling of the holons starts using a derivation of channel/ agent PN, called

Production Flow Schema (PFS) [3]. The PFS models represent modifications on the flow[1] of items, activities and inter-activities. Then successive refinements are applied and the PFS model is more detailed at each level to generate the dynamic models in non-autonomous PN named IOPT [4]. This combined approach allows modeling system structures as well as system behaviors in a more simplified and transparent manner than ordinary Petri net representations.

The holons of this control architecture are: (i) *product holon* (*PrH*) which represents the *raw material*[2], *intermediate* or *final product*. The set of *PrHs* form the *production plan* holarchy which represents a recipe of how to join *raw material* and *intermediate products* to obtain the *final product*; (ii) *task holon (TH)* which represents the *business processes* strategies to attend the *product order*. The *TH* also is conceived to request the *reconfiguration* strategies; (iii) *supervisor holon* (*SuH*) which contains all the knowledge to coordinate the *operations*, registering components and the messages exchange and providing services combined with other entities; and *(iv) operational holon* (*OpH*) which represents the *resource* such as, the equipments, the staff manager and operators. Therefore, *PrH, TH and SuH* form the holarchies to represent the set of operations of the high level control (enterprise and factory control level); and each *OpH* represents an atomic service which can be combined with atomic services by other holons forming holarchies to represent the supervisory and shop floor control level.

Modeling of *PrHs* and *THs* is made by *activities* of PFS and the internal workflow is modeled using IOPT. To model the workflow, a *place* represents a state while a *transition* represents an event or operation that conducts the *flow* from one state to another. Modeling of *SuH* involves the workflow of resources abilities, messages exchange between holons, supervision and command to implement actions related to fault treatment. The register of messages are represented though PFS *distributors*. *OpHs* represent resources (control object) and are directly modeled in IOPT. To model them with the requirement of fault tolerance, each control object in IOPT includes normal and fault states (represented by *places*) and conditions (represented by *transitions*) for the change of states; these conditions are represented by logical represented by *guards.* Thus, the *OpHs* models consider real component behavior including influence of the environment and their faults. The synchronization of the holons can be made with *auxiliary places* or through *guard expression* which contains command signals represent by *input* or *output signal* which in turn can be fired by *input* or *output event*.

Concerning fault diagnostic, the method presents a mechanism called diagnoser. To model the diagnoser, it is necessary to define observable and non-observable events [12] that are related to fault specification. The steps for this modeling are: (i) diagnoser construction, initially considering normal operations only; (ii) linking the strategies implemented by means of

---

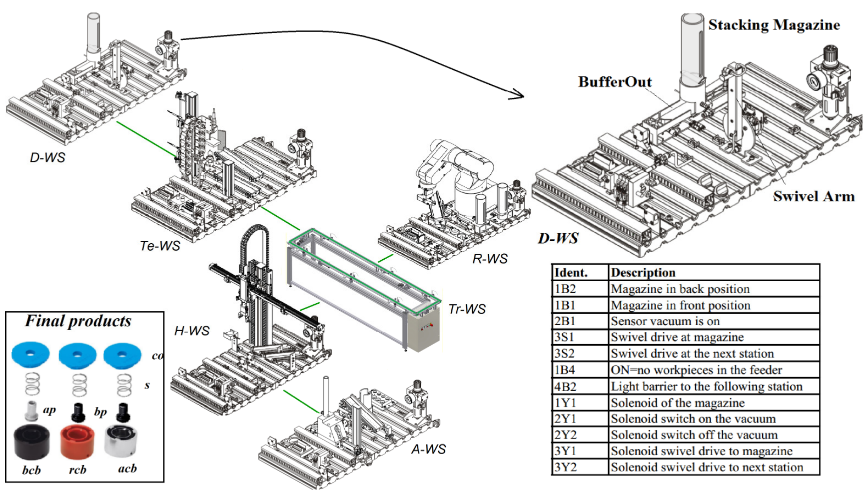
1 Terms associated to PN are in Arial *font*.

2 Terms associated to ontology are in Courier *font*.

transitions and *guard* to the possible `observable` and `non-observable events` that may occur from the `initial state`, and linking the states obtained according to *input* or *output signal* of actuators and sensors states; and (iii) linking the strategies implemented by means of *transitions* and *guard* to the events.

5 Modeling of Diagnosis Mechanism for RDMCS

The method is applied in a modular manufacturing system composed by WorkStations (*WSs*) described in Fig. 1. It is noteworthy that all the models are linked through of the PN elements representing real situations. Each *WS* controller is connected to internet allowing operators and clients interact with the system, that fits to the *engineering cloud* vision. The aim of this system is joining workpieces (*wp*s). The devices, their control functions, commands and signals of actuation and detection of each *WS* are also identified. The identification is made according to specifications DIN/ISO 1219-2:1996-11 and IEC 61346-2:2000-12. For instance, in the nomenclature, 1*S*2: 1 = circuit number, *S* = device code, and 2 = device number.



Ident.	Description
1B2	Magazine in back position
1B1	Magazine in front position
2B1	Sensor vacuum is on
3S1	Swivel drive at magazine
3S2	Swivel drive at the next station
1B4	ON=no workpieces in the feeder
4B2	Light barrier to the following station
1Y1	Solenoid of the magazine
2Y1	Solenoid switch on the vacuum
2Y2	Solenoid switch off the vacuum
3Y1	Solenoid swivel drive to magazine
3Y2	Solenoid swivel drive to next station

Fig. 1. Modular manufacturing system composed by workstations (*WSs*): distributing (*D-WS*), testing (*Te-WS*), transporting (*Tr-WS*), handling (*H-WS*), assembling (*A-WS*) and robot (*R-WS*). The aim is joining workpieces (*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*], to be obtained the final products: (i) [*bcb* + *ap* + *s* + *co*]; (ii) [*rcb* + *bp* + *s* + *co*]; and (iii) [*acb* + *bp* + *s* + *co*].

This paper describes some $D-WS$ models, illustrated on the right of Fig. 1 with its related sensors and actuators. The $D-WS$ function is providing [*bcb*], [*rcb*] or [*acb*] *wps* (on left at the bottom in Fig. 1) which are stored in the "stacking magazine" buffer. A double-acting piston pushes one *wp* out this buffer one at a time to the magazine. A swivel arm gets a *wp* in magazine via a suction gripper vacuum to move it to the transfer point of the *Te−WS*.

Figure 2 illustrates the PFS of `production plan` of the modular manufacturing system, refinement of `intermediate product` *PrH*-[*rcb*] and sub-refinement of *TH*-[*request provides wp of D-WS to Te-WS*] wich is detailed using IOPT model in Fig. 3. To clarify how communication/translation between different notations is assured, in this figure the related *activities* of the PFS in Fig. 2 are also illustrated inside the *transitions* and *places* of their IOPT model.

Figure 4 presents some control objects models of *D−WS* (represented according to codes listed in Fig. 1) which put the "magazine" in front or back position. As above stated, the *OpHs* are directly modeled in IOPT considering faults states.

Each *TH* has `diagnoser` and its `sub-diagnosers` model which are modeled at a similar manner of the models in Fig. 5 where an example for *D−WS* (represented by *TH-[Diagnostic for D-WS]* and *TH-[Diagnostic for 1ARetreating]*) are illustrated. The models evolve together and according to the executed control strategy, the possible states are diagnosed (see highlighted blue dashed line). The transformation of PFS into IOPT model is not presented but is similar to example in Fig. 3. Commands signals (represented in Fig. 3) and the *input* and *output signals* of all control objects of *D−WS* (such as represented in Fig. 4) are linked in these models.

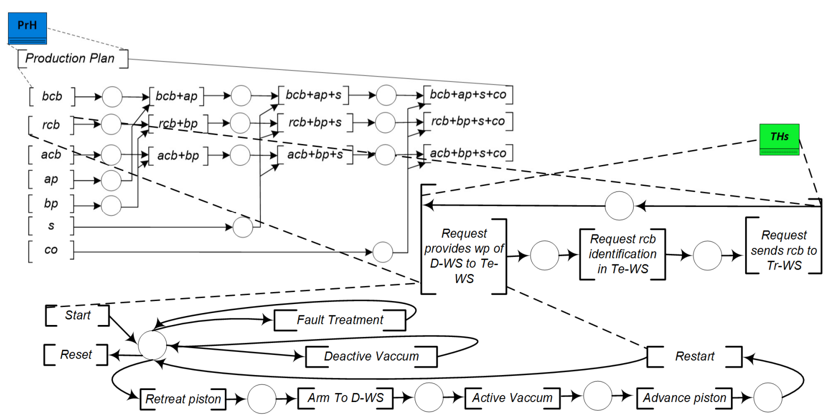


Fig. 2. PFS of workflow to obtain PrH-[*rcb*]. The services are represented by *THs* which are reuse in other products, such as *[request to provide wp of D−W S to Te−WS]* wich is used by all *wp*s. The *THs* are: *[request to provide wp of D−WS to Te−WS]* in which *D−WS* through "swivel arm" takes a *wp* in its buffer out and put it in "buffer in" of *Te−WS*; *[request rcb identification in Te−WS]* in which *Te−WS*s checks color and height to identify *wp*; and *[request sends rcb to Tr−WS]* in which *Te−WS*s sends the identified *wp* to *Tr−WS*.

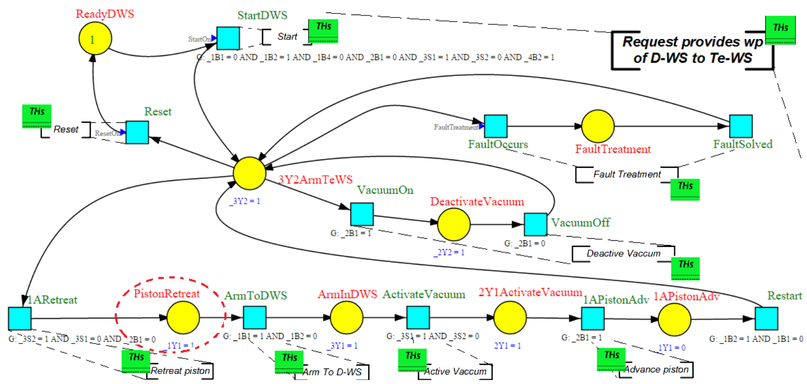


Fig. 3. IOPT for control of *D− W S*. Each transition has signals represented by "Guard (G)" which represent the equipments listed in Fig. 1. For instance, the "ArmToDWS" *transition* has associated the "G: 1B1=1 AND 1B2=0" that means "magazine is front position" and the "1B1 sensor" is sending signal "on". Thus, a *token* is fired for "ArmInDWS" *place* sending *an output signal* "3Y1="1 that in turn modifies the state of this solenoid. This IOPT has also *input events:* (i) "*StartOn*" of "*StartD-WS*" *transitions* to initiate the *D−W S*, (ii) "*ResetOn*" to reset the *D−W S*, and (iii) *FaultTreatment* to solve fault occurrence.

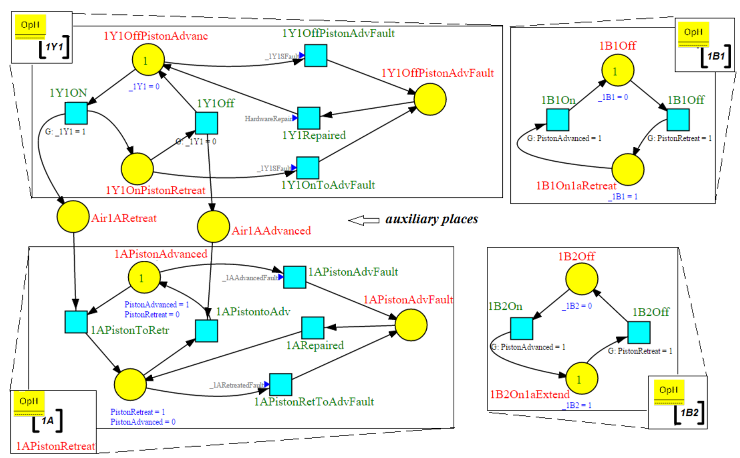


Fig. 4. *OpHs* of *D−W S* modeled with `normal` and `fault states` (see codes in Fig. 1). For instance, *OpH-[1Y1]* has `normal states` "1Y1Off" to advance piston and "1Y1On" to retreat piston; and `fault state` "1Y1OffPiston" which maintaining the piston only in advanced state. The models are linked, such as *OpH-[1Y1]* is linked to *OpH-[1A]* through "Air1ARetreat" and "Air1AAdvanced" *auxiliary places;* and *OpH-[1B1]* and *OpH-[1B2]* is linked to *OpH-[1Y1]* and *OpH-[1A]* through *input* and *output signals*. The *input events* represent external changes, which can be a fault, such as "1Y1SFault".

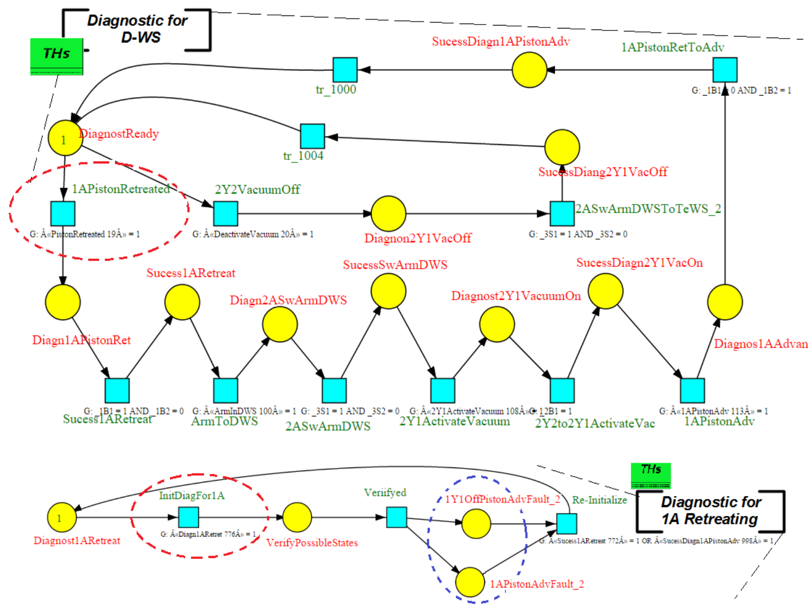


Fig. 5. `Diagnoser` for $D\!-\!W\,S$ linked to control strategy and signals of control objects. For instance (see highlighted dashed red line in both figures), "Piston Retreated" control strategy (*place* in Fig. 3) is linked to "1APistonRetreated" *transition* of `diagnoser` through the *guard* "G:PistonRetreated"; while the "InitDiagFor1A" *transition* in *TH-[Diagnostic for 1ARetreating]* has "G:Diagn1ARetreat" *guard* linking to *TH-[Diagnostic for D-WS]*.

6 Conclusions and Further Work

This paper proposes the modeling of fault diagnosis mechanisms for Reconfigurable and Distributed Manufacturing Control System (RDMCS). These mechanisms extend a method which uses holonic and multi-agent system and service-oriented architecture concepts through a combined top-down and bottom-up approach based on Petri nets extensions: production flow schema and input output place transition. This approach allows modeling the complex system structures of RDMCS as well as system behaviors in a more simplified and transparent manner than ordinary Petri net representations; avoiding repetition and overlapping tasks through reuse; utilization of web tools for automatic generation of controller languages; collaboration among distributed subsystems; and cloud engineering facilitating the analysis and communication between different design staffs. Furthermore, the fault-tolerant mechanisms allow reconfiguration flexibility for larger safety and new production demands. An application example presents the developing and composition of the models highlighting the advantages of the proposal. Plans for further work include modeling of decision-making mechanism as well as other case studies.

Acknowledgments. The authors thank the partial financial support of the agencies: CNPq, CAPES and FAPESP.

References

1. da Silva, R.M., Junqueira, F., dos Santos Filho, D.J., Miyagi, P.E.: A method to design a manufacturing control system considering flexible reconfiguration. In: Industrial Informatics (INDIN), 2014 12th IEEE International Conference on. pp. 82–87 (July 2014)
2. da Silva, R.M., Blos, M.F., Junqueira, F., Santos Filho, D.J., Miyagi, P.E.: A service-oriented and holonic control architecture to the reconfiguration of dispersed manufacturing systems. In: Technological Innovation for Collective Awareness Systems. Volume 423 of IFIP Advances in Information and Communication Technology. pp. 111–118 Springer Berlin Heidelberg (2014)
3. 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 Rob otic Systems 26(3-4), 423–437 (1999)
4. Gomes, L., Barros, J., Costa, A., Nunes, R.: The input-output place-transition Petri net class and associated tools. In: Industrial Informatics, 2007. INDIN'07. 2007 5th IEEE International Conference on. pp. 27–32, IEEE (2007)
5. Candido, G., Barata, J., Jammes, F., Colombo, A.W.: Applications of dynamic deployment of services in industrial automation. In: Emerging Trends in Technological Innovation. pp. 151–158 Springer (2010)
6. Barbosa, J., Leitão, P., Adam, E., Trentesaux, D.: Self-organized holonic manufacturing systems combining adaptation and performance optimization. In: Technological Innovation for Value Creation. pp. 163–170 Springer (2012)
7. Pereira, F., Moutinho, F., Ribeiro, J., Gomes, L.: Web based IOPT Petri net editor with an extensible plugin architecture to support generic net operations. In: IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society. pp.6151–6156, IEEE (2012)
8. Li, B.H., Zhang, L., Wang, S.L., Tao, F., Cao, J., Jiang, X., Song, X., Chai, X.: Cloud manufacturing: a new service-oriented networked manufacturing model. Computer Integrated Manufacturing Systems 16(1), 1–7 (2010)
9. Leitão, P., Marik, V., Vrba, P.: Past, present, and future of industrial agent applications. Industrial Informatics, IEEE Transactions on 9(4), 2360–2372 (2013)
10. Hsieh, F .S., Chiang, C.Y.: Collaborative composition of processes in holonic manufacturing systems. Computers in Industry 62(1), 51–64 (2011)
11. Obitko, M., Vrba, P., Marík, V., Radakovic, M., Kadera, P.: Applications of semantics in agent-based manufacturing system. Informatica (Slovenia) 34(3), 315–330 (2010)
12. Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D.C.: Failure diagnosis using discrete-event models. Control Systems Technology, IEEE Transactions on 4(2), 105–124 (1996)