Control Engineering Practice 49 (2016) 87–100

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Control architecture and design method of reconfigurable manufacturing systems

Robson Marinho da Silva [a,b,*], Fabrício Junqueira [b], Diolino J. Santos Filho [b], Paulo E. Miyagi [b]

[a] Universidade Estadual de Santa Cruz, Ilhéus, BA, Brazil
[b] Escola Politécnica da Universidade de São Paulo, São Paulo, SP, Brazil

ARTICLE INFO

Article history:
Received 3 July 2015
Received in revised form
22 January 2016
Accepted 26 January 2016
Available online 5 February 2016

Keywords:
Reconfiguration
Holonic and Multi-Agent System
Service-Oriented Architecture
Petri net
Manufacturing system

ABSTRACT

This paper proposes a control architecture for reconfigurable manufacturing systems and its design method based on Petri nets (Input Output Place Transition and Production Flow Schema), Service-Oriented Architecture and Holonic and Multi-Agent System techniques, among other good practices. The control architecture integrates value-added activities, information and resources. The method considers the exchange of knowledge among heterogeneous business workflow of manufacturing subsystems in different geographical locations, and allows modeling process, product, machine and device control. An example demonstrates advantages of the resulting control system.

1. Introduction

The evolution of manufacturing systems is characterized by a migration of paradigms related to its concerns: mass manufacturing for cheaper products; lean manufacturing for continuous quality improvement; flexible manufacturing for products diversity; and reconfigurable manufacturing that must be self-adjustable to market interests (Koren & Shpitalni, 2010; Mehrabi, Ulsoy, & Koren, 2000). Other paradigms combine the aforementioned one, such as mass customization for flexibility to match a wide product family modified according to specific customer needs. A recent paradigm is called "the fourth industrial revolution" (Industry 4.0) in which technologies are combined to integrate machines and humans to compose value chains of entities (such as industrial plants) in different geographical locations (distributed manufacturing systems), which must provide services and products in an autonomous manner (Leitão, Rodrigues, Barbosa, Turrin, & Pagani, 2015). Industry 4.0 considers the former paradigms including reconfiguration and technological advances related to cyber-physical systems and cloud computing environment.

Despite the potential advantages of the Industry 4.0 paradigm, its implementation depends on a Reconfigurable Manufacturing Control System (RMCS) that ensures:

(i) reconfiguration flexibility to increase safety in fault occurrence, and to allow creating, updating or replacing products and/or legacy systems in case of new demands (da Silva, Junqueira, dos Santos Filho, & Miyagi, 2014);
(ii) autonomous and intelligent components for the adequate use of resources (Mařík & Lažansky`, 2007);
(iii) supervision of the system goals combined with this autonomy (Giret & Trentesaux, 2015);
(iv) responsiveness, considering reconfiguration in functionality and in production according to the market, product and resources changes. RMCSs can allow this responsiveness considering six core reconfiguration characteristics: customization, convertibility, scalability, modularity, integrability and diagnosability (Koren & Shpitalni, 2010).

Engineering RMCS is not a trivial task and methods are necessary to overcome the challenges involved (Mařík, Schirrmann, Trentesaux, & Vrba, 2015). For instance, when an unexpected event occurs and a reconfiguration in the production profile is needed, within a very short time, entities such as supervisors and controllers need to interact, considering domains and technologies with different operational systems. From the topology standpoint, the RMCS requires a review of the decision-making structure, because the businesses and productive processes are not static and

* Corresponding author at: Universidade Estadual de Santa Cruz, Ilhéus, BA, Brazil.
E-mail addresses: rmsilva@uesc.br (R.M. da Silva), fabri@usp.br (F. Junqueira), diolinos@usp.br (D.J.S. Filho), pemiyagi@usp.br (P.E. Miyagi).

http://dx.doi.org/10.1016/j.conengprac.2016.01.009
0967-0661/

the system must promptly react to any abnormal situation including faults in the communication line of the central control. Nevertheless, absence of any hierarchical coordination is not feasible because it requires each equipment to have its own capacity of decision-making to improve its performance (Giret & Trentesaux, 2015).

In fact, there is not a unique technique to overcome all the challenges; an alternative solution is combining complementary characteristics of different techniques to specify a RMCS. The adequate use of Holonic and Multi-Agent System (HMAS) concepts can facilitate the development and integration of distributed heterogeneous systems combining aspects of hierarchical and heterarchical structures (Pujo, Broissin, & Ounnar, 2009). HMAS explores the superposition of multi-agent and holonic systems concepts, such as autonomy, reactivity, cooperation, social capacity and learning resources, taking advantage of complementary features in the implementation of holons by means of agents standards and tools already available (Giret & Trentesaux, 2015). In fact, the number of scientific topics and the achievements in the HMAS field is growing and it has been explored to propose solutions for Industry 4.0 (Mařík et al., 2015). However, usual HMAS solutions do not consider interoperability with other systems at least not explicitly.

In turn, the Service-Oriented Architecture (SOA) was conceived to treat inter-enterprise collaboration through interoperability patterns for interfaces. Furthermore, there is a synergy between SOA and HMAS to be explored. For instance, agents can provide services to other agents in the same way as services are provided in SOA and both use messages to exchange data (Shen, Hao, Wang, Li, & Ghenniwa, 2007). However, there are still few proposal combining these concepts, or more specifically for RMCS; and both HMAS and SOA techniques have not associated formalisms for process/system modeling.

Through an adequate interpretation the RMCS structure and dynamic behavior can be based on formal properties of discrete event system (Murata, 1989). Consequently, the semantic, graphical descriptions or simulation models associated to the Petri Net (PN) technique can be used for RMCS. An additional advantage is that some classes of PN can automatically be converted to programs for industrial controllers, such as the PN named Input Output Place Transition (IOPT) (Gomes, Barros, Costa, & Nunes, 2007; Gomes, Costa, Barros, & Lima, 2007). Another PN class named Production Flow Schema (PFS) (Hasegawa et al., 1999) can also be explored as a workflow for system conception and communication among staffs that work with different technologies, information, resources and knowledge. According to Caire, Gotta, and Banzi (2008), workflow is understandable by domain experts as well as by developers, because a workflow reduces the need for programming skills.

Therefore, this paper proposes two different aspects of RMCS: architecture of the system and the work organization of the staff, combining strength aspects of the techniques stated above (SOA, HMAS, PN) and good practices for design method (such as the workflow-based approach and Unified Modeling Language (UML) diagrams). Section 2 summarizes the state of the art, related works and tools for engineering RMCS. Section 2.1 highlights the drawbacks and the advantages of existing methods for HMAS design and SOA concepts. Concepts of a hybrid top-down and bottom-up approach based on specific extension of PFS and IOPT are summarized in Section 2.2. A mapping of the control architecture is given in Section 3 and the proposed method to obtain it is detailed in Section 4. Section 5 has an application scenario example following the method statements and using the tools available (cited Section 2.3) for edition, debug, simulation and generation of controller programs derived from the models. Section 6 presents the main conclusions.

2. Related works, trends and challenges

For better understanding problem statement, trends and challenges for engineering RMCS, this section presents an overview of existing and main approaches related to concepts and frameworks of HMAS, SOA and PN techniques.

2.1. Holonic and Multi-Agent System and Service-Oriented Architecture

A comparative summary relating the analysis of various requirements to develop manufacturing system based on HMAS and SOA approaches is presented in Giret and Trentesaux (2015). Most HMAS are based on a reference control architecture named Product Resource Order Staff Architecture (PROSA) (Van Brussel, Wyns, Valckenaers, Bongaerts, & Peeters, 1998) that deals with centralized manufacturing functions in which only order holons are autonomous. However, manufacturing processes undergo several changes and disturbances, with certain level of uncertainty and unpredictability. For agile reaction, autonomous entities must be organized in structures where all holons can make decisions considering a certain supervisory control (Giret & Trentesaux, 2015). For this, the design of HMAS-based control system must identify constraints on the holons to they act as semi-autonomous entities. In this sense, Leitão and Restivo (2006) propose ADACOR, a productive system based on product, operational task and supervisor holons for agile reaction to changes in production scheduling. In Pach, Bekrar, Zbib, Sallez, and Trentesaux (2012), ADACOR is applied to dynamic allocation and routing in manufacturing system and the models developed are validated using software for multi-agent system simulation of different production scenarios. Still inspired in PROSA, Pujo et al. (2009) propose PROSIS (Product, Resource, Order and Simulation Isoarchic Structure) approach in which all the decisions are taken considering the autonomy of the holons (isoarchy).

In a HMAS architecture, holons may be considered services (Leitão et al., 2015; Shen et al., 2007) to be accessible from different holons, allowing interoperability of cross-company services. However, there is the need to develop mechanisms for service composition (holarchy) for this interoperability. To model this composition, in Hsieh (2010) the optimization of productive systems with timing constrains is considered using a two-layer contract net protocol with PN models. In Leitão et al. (2015), a system based on cyber-physical technologies is approached defining a product process specification aligned with Industry 4.0. This approach integrates SOA and HMAS concepts considering a strategy based on intelligent products, orchestrating services provided by agents.

In DeLoach and Garcia-Ojeda (2014), the Multi-agent System Engineering (MaSE) is proposed. MASE is divided into analysis and design phases. Another approach is the Methodology for Ontology-Based Multi-Agent System (MOBMAS) (Tran & Low, 2008), which is developed through iterative and incremental activities. Plant Automation BAsed on DIstributed Systems (PABADIS) (Cernuzzi, Cossentino, & Zambonelli, 2005) is a step-by-step requirement-to-code methodology for designing and developing multi-agent systems, integrating design models and concepts from object-oriented software engineering and artificial intelligence approaches using UML diagrams. ANEMONA (Botti & Boggino, 2008) is a methodology based on top-down and bottom-up development process of manufacturing systems and it is divided into specific aspects that form different views of the system. In Shen et al. (2007), an ontology agent provides semantic integration services, responds to service requests and performs ontology reasoning and match-making. The idea is that manufacturing control systems require functional units that can be connected to the different levels in the system by means of standardized interfaces and