Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-3 (2015) 1111–1116

Implementation of a Process Orchestration Model in a Service Oriented Holonic Manufacturing System

Francisco Gamboa Q.*, Olivier Cardin*, Anne L'Anton*, Pierre Castagna*

** L'UNAM Université, IUT de Nantes-Université de Nantes, IRCCyN UMR CNRS 6597,2 avenue du Prof. Jean Rouxel 44475 Carquefou, France (e-mail: firstname.lastname@irccyn.ec-nantes.fr).*

Abstract: Process models are usually represented by linear sequences, where the order of the conforming operations is fixed a priori. In a FMS, flexibility can be considered as proportional to the number of alternatives the system has when a decision needs to be made. Considering only flexibility related to the choice and the order in which the products are treated on machines is therefore limitative when the design of the products could enable flexible recipes, where groups of operations could be permuted according to the state of the system. The objective of this paper is to describe the implementation of service and process models as well as the orchestration of such processes at a product level.

Keywords: Intelligent manufacturing system, holonic manufacturing system, service-oriented architecture, production control, process orchestration.

1. INTRODUCTION

Holonic and Service Oriented Architectures have been very attractive solutions for developing the so called "Next Generation Manufacturing Systems". Works such as (Cândido et al., 2009; Jammes and Smit, 2005; Jammes et al., 2005) among others, have recognized the advantages of mixing both paradigms in terms of flexibility. Flexibility on the global behavior of the system depends on the strategies to exploit the inherit flexibility at all levels of the system. The way information is presented will greatly define the limits of the system for finding new solutions in the production control, namely scheduling, resource allocation, and system reconfiguration.

In scheduling, flexibility can be found at different levels. Most of the approaches found in literature (Cardin and Castagna, 2009; Holvoet and Valckenaers, 2006; Leitão and Restivo, 2008; Mendes et al., 2010) consider flexibility just at a resource level i.e. the proper allocation of resources. However, there is also flexibility found at a product level i.e. the process structure. Indeed, the way a process is decomposed, based on its nature, provides possibilities to find new solutions in terms of operations sequencing. Product-level flexibility therefore consists in exploring the order of manufacturing operations according to the process' structure and constraints.

Process models are usually represented by linear sequences, where the order of the conforming operations is fixed a priori. In a FMS (Flexible Manufacturing System), flexibility can be considered as proportional to the number of alternatives the system has when a decision needs to be made. Considering only flexibility related to the choice and the order with which products are treated on machines is therefore limitative when the design of products could enable flexible recipes, i.e. groups of operations could be permuted according to the state of the system.

The objective of this paper is to describe the application of a process orchestration model, based on manufacturing services, able to handle flexible recipes on a multipath intelligent flexible workshop with three workstations commanded by a Service-oriented Holonic Manufacturing System. Section 2 introduces the conceptual models of manufacturing processes and services that will be used in section 3. Section 4 presents a study case describing how these models are implemented in a SoHMS. Finally Section 5 describes the successive steps necessary to orchestrate services at a product level in the SoHMS.

2. SERVICE SPECIFICATION MODELS

2.1. Manufacturing Processes in a SoHMS

In (Gamboa Q., 2013) manufacturing processes are classified according to their fractal characteristics and to the concurrency of its composing operations as shown in Fig.1.

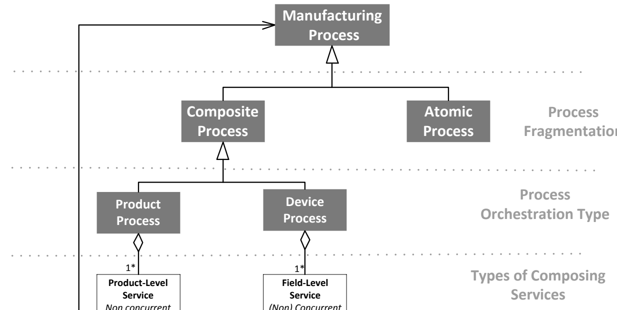

Fig. 1. Process Classification

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.06.232

According to the principles of SoA, manufacturing processes can be either composite or atomic. A *composite process* is a process that can be divided into more granular operations. The relations between such operations determine the structure of the process. Within the composite processes, the nature of the relations between the composing operations distinguishes two classes of composite processes, namely *Product Processes* and *Device Processes*. The distinction is based on the possible concurrency among the operations. Product processes are those composed of only non-concurrent services i.e. no more than one service is executed at the time. These types of processes can be found at job-shop configurations and product driven systems where the product travels from one station/machine to the other to suffer transformations. Device processes are those offered at a lower level, at workstations/machines, where the relations between the composing operations have a tighter coupling needing of parallel synchronization.. On the other hand there are the atomic processes which represent indivisible operations. As illustrated in Fig.1, such processes are offered by manufacturing services: *Product-Level Services* and *Field-Level Services* respectively. Process recursivity is kept as processes can be composed of services which in turn can represent more granular processes and so on down to atomic field-level services and up to composite services offered to the client as production orders.

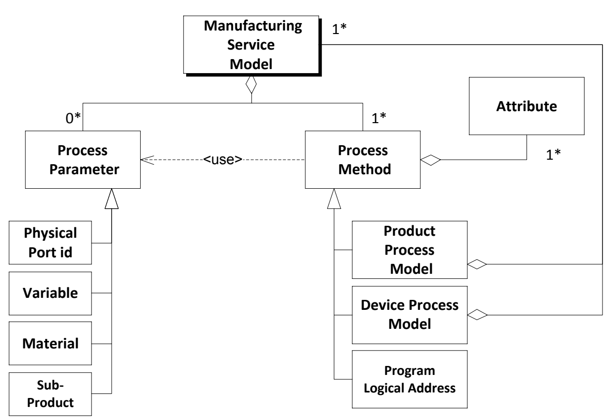

Fig. 2. Manufacturing Service Model

Fig.2 proposes a conceptual model that defines the composition of services, in the manufacturing context, in order to properly compose integral product processes. A manufacturing service can be offered by one or more process methods and by a collection of process parameters. Such process parameters represent fields of information that are relevant to the client requesting the service namely a variable, the specification of a material type, or sub-product. Methods on the other hand, contain the information on the way to provide such service, i.e. a process model with a list attributes describing the quality of the service. Such attributes will serve for its evaluation and comparison. Therefore, one same service type can be executed by product-process, a device-process or as an atomic-process. The encapsulation of the service implementation, orchestration included, allows the coexistence of different models in the system.

2.2. Product-Process Model

The main characteristic of product-level processes is the non-concurrency between its services. Its conceptual model, Fig. 3 is conceived based on the ISA SP-95 standard which specifies all the required production information of a product. It rearranges and clusters the information for the convenience of its applicability in product driven systems with customized products being a trend on the rise.

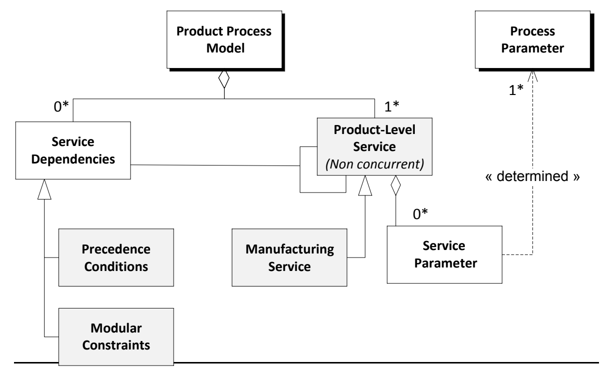

Fig. 3. Product-level Process Model

Product processes are composed of one or more non-concurrent manufacturing services, each having a collection of parameters issued from the process parameters and a binding function. . The most relevant part of the process model, which will determine the orchestration method, is the way service interdependencies are expressed. In this case, product processes interdependencies are declared in a *predecessor perspective* with a table of precedence conditions. Such precedence conditions indicate , with constructs of Boolean logic, what services need to be executed before the given service. Modular constraints are added in the precedence table to integrate modular customization on the composition of a process. The inclusion or choice of optional services can provide added features to the product.

2.3. Product-Level Service Orchestration

At a product-level the objective is to have an orchestration mechanism that is flexible and reactive at the same time; flexible in its capacity to give different solutions to the product's production lifecycle and reactive in giving valid solutions (not necessarily the best) on. Moreover, the orchestration mechanism should: (i) describe the general structure of a process all information on possible sequencing, (ii) welcome the scalable and configurational parameters, (iii) be computable on-line for reactivity and (iv) should be easy to understand and program with no need of specialized knowledge on computer science in order to be accessible for average process designers. In this matter, the Petri-Nets formalism results to be a very good means for constructing an orchestration model and provide the logic dynamics for the orchestration of production sequences due to its characteristic ability to capture the synchronous and asynchronous relations between tasks, to the ease of its graphical language and its minimal footprint, interesting for embedded applications as in mobile entities.