Process specification framework in a service oriented holonic manufacturing system

Francisco Gamboa Quintanilla, Olivier Cardin, A. l'Anton, Pierre Castagna

Process Specification Framework in a Service Oriented Holonic Manufacturing Systems

F. Gamboa Quintanilla, O. Cardin, A. L'Anton, P. Castagna

LUNAM Université, IUT de Nantes – Université de Nantes, IRCCyN UMR CNRS 6597 (Institut de Recherche en Communications et Cybernétique de Nantes), 2 avenue du Prof. Jean Rouxel – 44475 Carquefou
(e-mail: francisco.gamboa/olivier.cardin/anne.lanton/pierre.castagna@univ-nantes.fr).

Abstract.– Holonic and Service-oriented Architectures have been proposed as solutions for the conception of flexible and reactive systems. Flexibility, being one of their main objectives, depends greatly on the way information is presented in the system which can limit the flexibility of higher levels strategies as in process planning and reconfiguration. Although many works propose the use of services, none have been found describing what services stands for in a manufacturing context or how these can form manufacturing processes. This paper proposes a methodology for designing manufacturing-process specifications based on workshop-services that welcomes product customization and suitable for product driven applications. Conceptual models for processes and services are proposed in this work designed to preserve the fractal characteristics of products and processes and facilitate service reutilization. Such models form part of specification framework that will serve as a reference for the design and conception of manufacturing processes and services in Service-oriented Holonic Manufacturing Systems.

Keywords: Holonic Manufacturing, SoA, Product Specification, Manufacturing-Services, Process Families.

1 Introduction

An evolution in the goods market (namely: highly customized products and shorter product lifecycles) has forced companies to adopt an exhaustive search for achieving responsiveness, flexibility, a reduction of costs and an increased productivity in order to stay competitive in such new changing environment. The conception of these so called Next Generation Manufacturing Execution Systems has been challenging the community of Intelligent Manufacturing systems for two decades now to incorporate such attributes. Holonic Architectures and Service-Oriented Architectures (SoA) have been two of the most studied and referenced solutions to this problem, in manufacturing and informatics respectively. Both of these solutions provide the necessary guidelines to create open, flexible and agile control environments for the next generation manufacturing systems. The combination of both of these principles appears to be a very attractive option as seen in works relating these two paradigms,[1–3]. However, there are no works been found describing how a process is composed, based on services, nor has been the composition of a service representing integrally a manufacturing process with an eye on its application on HMS. Moreover, flexibility, being the main attribute soled by these paradigms depends on the intrinsic flexibility found at all levels of the system. The way information is presented in the system will greatly define the flexibility limits at higher levels for finding new solutions; as in process planning and reconfiguration. For instance, if the information describing the system's components and activities fails to express the underlying capacities and possibilities, the intrinsic flexibility present in the production floor will not be identified nor control strategies will be exploited to its best. The objective of this paper is to propose a methodology for designing manufacturing process specifications based on Workshop services that welcomes product customization and is suitable for their application on distributed and product-driven systems. Conceptual models for Manufacturing Process and Workshop-services are proposed in this work designed to preserve the fractal characteristics of products and promote the reutilization of operations. Fractality of services (i.e. a same model of services from highest to lowest level services, compound or atomic) is especially efficient as it naturally fits to the nature of the requester. The products, resulting from the execution of services, therefore needs to be fractally modelled in order to take advantages of the structure of the corresponding services.

Section 2 describes briefly some approaches of process specification of works related to HMS and SoA and make as small review on existing works on process specification which inspired the propositions of this paper. Section 3 proposes a manufacturing service model, a process model for each of the process types identified and finally a specification framework based on such models.

2 Existing Process Specification in Distributed Systems

Traditionally, in manufacturing systems, process models are usually represented by linear sequences where the order of the conforming tasks is fixed a priori by process engineers based on their insight about the process nature and the production system. Such is the case in Flexible job-shops [4] where the main problem is to find a solution for scheduling resource allocation for a given collection of sequences called jobs. Recent works on HMS, propose to enrich the process model by considering the existence of different alternative operations for a given production state such as in [5] and [6], where the authors propose Petri-net Controllers for modeling alternative operations. In the same matter, [7] and [8] suggest the creation of a *Logical Operating Sequence* independent from the production floor. However, none of these works elaborate on how to create such models or on a way to describe the process structure. Regarding SOA, many works suggest the integration of Web Service technology in Industrial Systems such as [9] pointing out the issues for its application, [5] and [6] proposing Petri-net controllers for processes formed by services, [10] proposing a

Service-oriented Manufacturing Architecture with Multi-Agent technology, among others. Nevertheless, there has not been found works, apart those on Semantic Web-Services [1, 11, 12], with a detailed description of what a service represents in the manufacturing context and what are its composing elements in order to build fractal processes and ensure their integral description.

In terms of process specification, there has been a great deal of effort devoted to the formal and philosophical specification of processes as well for the development of process models. Among these efforts there is the Process Specification Language (PSL) [13] which is a proposition of a formal ontology providing a formal description of the components and relations that form a process; the IDEF3 [14], a process specification capturing method with a graphical language conceived to describe and represent the structural nature of a process; the Web services Description Language (WSDL) [15] dedicated to the abstract description of a service interface for its proper invocation and the Business Process Execution Language (WS-BPEL) [16], used to describe the different workflows that can be composed by the collection of Web services found in a process. The issue is that most of these languages and methods are not intended nor suitable for describing processes in industrial applications (they are intended for enterprise-level systems) specially those involving intelligent products, or active as proposed in the analysis framework of [17] where decision making is embedded.

3 Customizable Service-Oriented Process Specification

In product customization, companies tend to adopt the development of *Product Families* [18] which recognizes the existence of scalable and modular product customization platforms. According to [19, 20], commonality found in a product family structure usually translates into a commonality in the process domain. Such idea brings rise to the concept of *Process Families* which, in the same way as product families, possess the attributes of commonality, modularity, reutilization and scalability but in the process domain [18] . A Process Family can then be seen as a collection of manufacturing operations that respond to the realization of the corresponding structural modules of a Product Family. These, now called manufacturing services, possess a proper identification and description independent from the service provider i.e. they are identified according to the added transformations, with no regard on the methods that are used for their implementation. Thanks to this, manufacturing services can be standardized and be readily available to integrate different production processes thus bringing the benefit of reusability that will reduce reprogramming efforts. Moreover, customization can be incorporated into the process at a scalable level through the parametrization of services and at a modular level through the choice of services to be involved in the process.

3.1 Work-shop Services and Model

At a workshop level two main classes of services are Manufacturing Services and Supply Services. A *Manufacturing Service* (M-services) performs manufacturing transformations to the main product adding value to it. A *Supply Service* (S-service) refers to the service of providing a product or sub-product to a client namely; a costumer or a resource. The conceptual model for both, M-services and S-services is illustrated in Fig. 1.

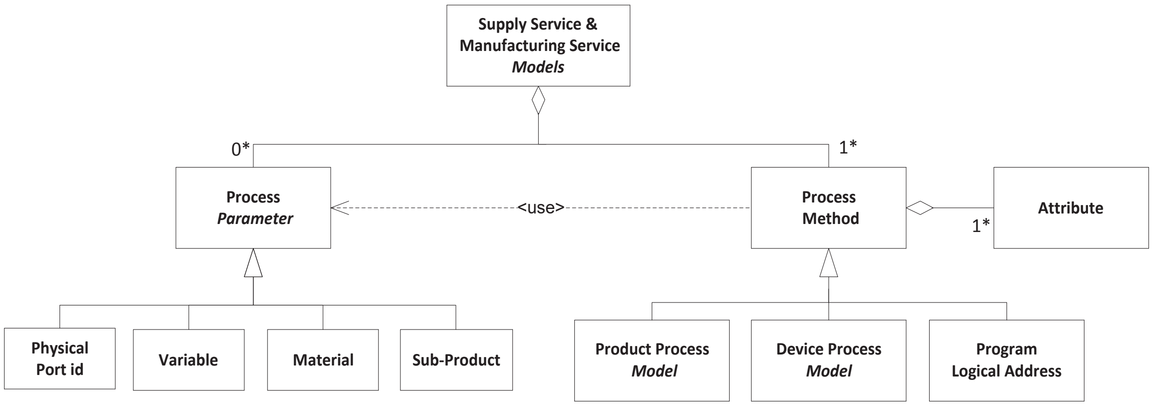

Fig. 1. Manufacturing (and Supply) Service Model

A M-service is composed of one or more *Process Methods* and by a collection of *Process Parameters*. A *Process Method* represents an action or a structure of actions that transform the product and/or the world as described by the service class. Three types of process methods where identified: *Product-Process Model*, *Device-Process Model* and a simple *logic address*. The first two, belong to *Composite Services* which provide processes composed of other more granular services while the last represents a program in the provider's controller executing an *Atomic service*. Each method, implementing the service, has its own set of *Attributes* used to evaluate its eligibility over other methods. A *Process Parameter* embodies a piece of information needed by the process method in order to determine the limits of the process, namely a *variable*, *port id* (for service delivery), a *material* or *sub-part*. Its cardinality reflects the flexibility of the service to reproduce a different result that adapt to different needs.

The two main virtues of this model are: it keeps the fractal character of products and processes and welcomes the integration of scalable customization. Fractality at the product level is kept by the process parameters class which can specify the integration of a material or sub-product. Fractality is also kept at a process level through composite services whose models will be described in the next section. As seen in Fig.1, a service can have more than one process method associated; this is due to the fact that methods are proprietary to provider and not to the service itself.

This allows the integration of different technologies that will increase the system's flexibility for re-configuration and expansion makes process specification platform independent. Reusability of services is leveraged by the decoupling of parameters from methods which through parametrization allow them to fit in different processes, thus integrating customization at a scalable level into the process domain.

3.2 Product's Process Model

Manufacturing processes are characterized by different aspects such as its granularity, its taxonomy's category and in the concurrency of its composing services. As shown in Fig.2, the granularity of a manufacturing process can be either composite or atomic. Composite processes can be of two types: Product Processes and Devices processes characterized by the composing services concurrency. *Product Processes* are those composed of only non-concurrent services i.e. there can only be one service executed at the time. *Device Processes*, on the other hand, are those having concurrent services i.e. more than one service can happen simultaneously and require of synchronization for their execution. Fractality is also highlighted in this diagram, Fig.2, by indicating that the composing Product-level and Device-level services can in turn represent other composite processes and so on until having just atomic services.

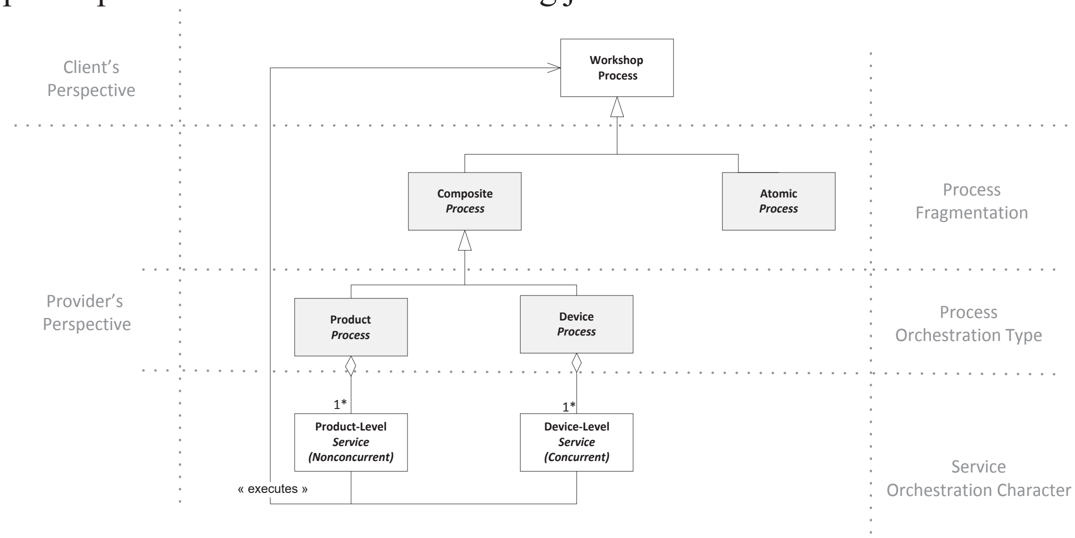

Fig. 2. Processes Types

On Fig.3, it can be seen the conceptual model for both types of composite processes namely; *Product-Level Processes* and *Device-Level Processes*. Such models where designed taking as reference the required production information of a product specified by the standard ISA SP-95 [21]. However, its information was readapted and clustered for the convenience of product driven systems with customized products.

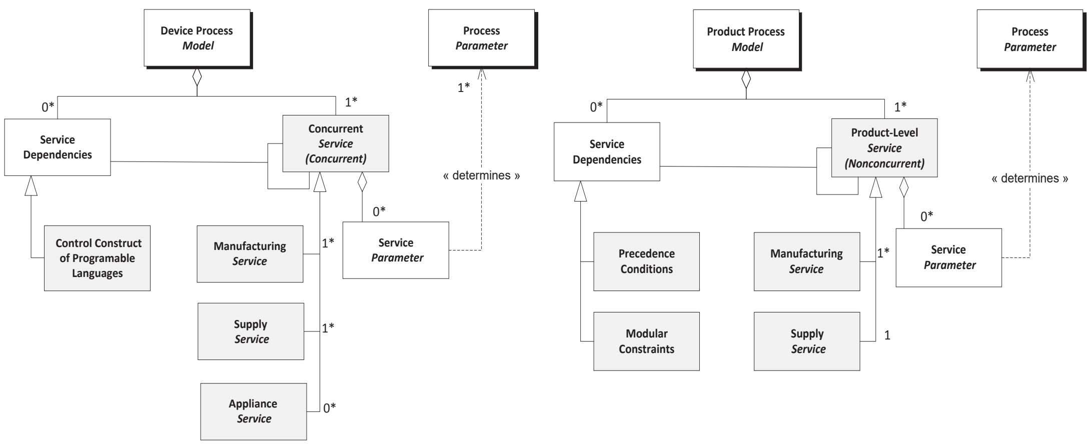

Fig. 3. Product- and Device-Process Models

These models correspond to the production recipe of a given product family whose production order is offered as a supply service. Both models are formed by a collection of workshop services, describing the required manufacturing operations, and a collection of service dependencies which describe the relations between the services which actually describe the structure of the process. Each service in the process possess a collection of service parameters needed for its execution which is determined out of the higher order Process Parameters. As seen in the last section, the main difference between these two models relies in the concurrency characteristic of the composing services. Such difference becomes tangible in the specification of the process structure i.e. dependencies among services. In a product-level process dependencies are stated with a *predecessor perspective* with a table of precedence conditions i.e. what services need to be executed before a given service. In a Device-level process, dependencies are more complex due to their to their more tight relationship thus these are suggested to be represented through control constructs found in programmable languages with an event-driven approach.

3.3 Process Specification Framework/ Walkthrough

Now that the conceptual models of both: processes and services, have been presented, Fig.4, describes the product specification process. It describes the steps from the definition of customer needs and functional requirements all the way up to the definition of a Process Family Model to be ready to go through the specification of customizable parameters. The application of these steps for the creation of the process model for each of the products declared in the system will lead to the construction of a library of manufacturing services based either on an application-specific ontology or on a unified domain ontology, as that presented in the German standard DIN 8593[22] for assembly processes.

4 Conclusion

This work describes a framework for the specification of manufacturing processes implemented in distributed systems based on the web-services whose application in HMS gives rise to a new paradigm: Service-oriented Holonic Manufacturing Systems (SoHMS).The methodology identifies different types of manufacturing namely: Product-level, Device-Level and Atomic processes. For each, a process model is proposed describing its composition and structure. A manufacturing-service model is also proposed describing its composition. Both models allow the description of fractality at a product level and at a process level. Moreover, the capability of resources can then be described by its offer of workshop-services, more than on its internal model. This facilitates the introduction of different resource technologies as manufacturing-service classes are independent of the technologies or methods used for their implementation.
Future work will concentrate on adding flexibility to the SoHMS through the creation of an orchestration model and engine that will allow to the scheduler explore the other valid sequences that will bring the system closer to an optimal point.

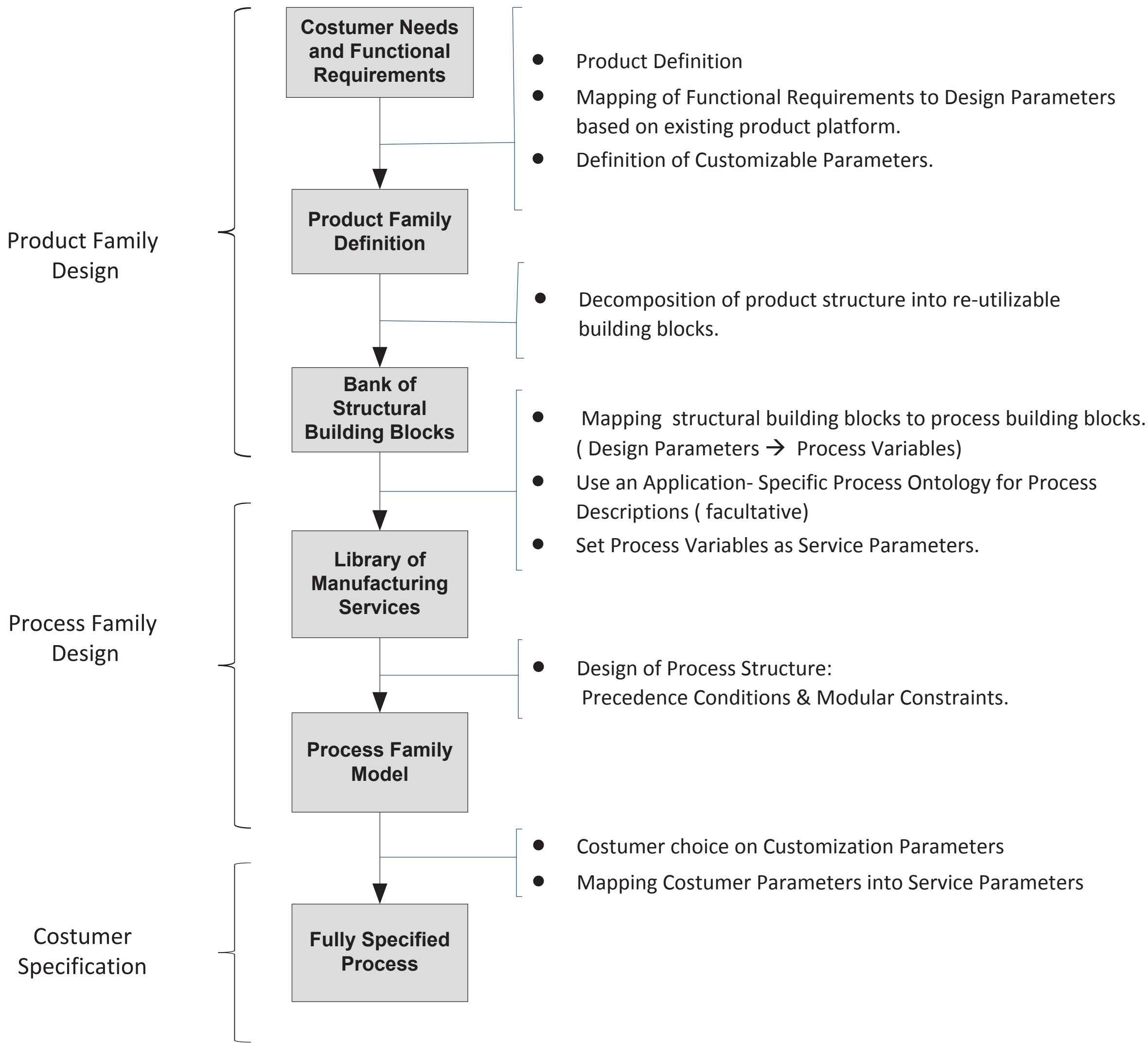

Fig.4. Process Specification Design

5 References

1. Jammes, F., Smit, H., Lastra, J.L.M., Delamer, I.M.: Orchestration of service-oriented manufacturing processes. 10th IEEE Conference on Emerging Technologies and Factory Automation, 2005. ETFA 2005. p. 8 pp.–624 (2005).
2. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. Wiley, Hoboken, NJ (2007).
3. Jammes, F., Smit, H.: Service-oriented paradigms in industrial automation. IEEE Trans. Ind. Inform. 1, 62–70 (2005).
4. World Records: Flexible Job Shop Scheduling Problem, http://www.quintiq.com/optimization/fjssp-world-records.html.
5. Mendes, J.M., Restivo, F., Leitao, P., Colombo, A.W.: Customizable service-oriented Petri net controllers. 35th Annual Conference of IEEE Industrial Electronics, 2009. IECON ’09. pp. 4341–4346 (2009).

6. Mendes, J.M., Leitão, P., Restivo, F., Colombo, A.W.: Process optimization of service-oriented automation devices based on Petri nets. 2010 8th IEEE International Conference on Industrial Informatics (INDIN). pp. 274–279 (2010).
7. Toguyeni, A.: Design of Modular and Hierarchical Controllers for Reconfigurable Manufacturing Systems. IMACS Multiconference on Computational Engineering in Systems Applications. pp. 1004–1011 (2006).
8. Bourdeaud'huy, T., Toguyeni, A.: Analysis of Reconfiguration Strategies Based On Petri Nets Models and Optimization Techniques. 2006 8th International Workshop on Discrete Event Systems. pp. 319–324 (2006).
9. Komoda, N.: Service Oriented Architecture (SOA) in Industrial Systems. 2006 IEEE International Conference on Industrial Informatics. pp. xxiii–xxiii (2006).
10. Nagorny, K., Colombo, A.W., Schmidtmann, U.: A service- and multi-agent-oriented manufacturing automation architecture: An IEC 62264 level 2 compliant implementation. Comput. Ind. 63, 813–823 (2012).
11. Delamer, I.M., Lastra, J.L.M.: Ontology Modeling of Assembly Processes and Systems using Semantic Web Services. 2006 IEEE International Conference on Industrial Informatics. pp. 611–617 (2006).
12. Lastra, J.L.M., Delamer, I.M.: Semantic web services in factory automation: fundamental insights and research roadmap. IEEE Trans. Ind. Inform. 2, 1–11 (2006).
13. National Institute of Standards and Technology (NIST): Process Specification Language (PSL), http://www.mel.nist.gov/psl/.
14. Knowledge Based Systems Inc.: IDEF3, http://www.idef.com/IDEF3.htm.
15. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web Service Description Language (WSDL), http://www.w3.org/TR/wsdl.
16. Jordan, D., Evdemon, J.: Web Services Business Process Execution Language, http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.
17. Sallez, Y.: Proposition of an Analysis Framework to Describe the "Activeness" of a Product during Its Life Cycle. In: Borangiu, T., Trentesaux, D., and Thomas, A. (eds.) Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics. pp. 257–270. Springer International Publishing (2014).
18. Jiao, J. (Roger), Simpson, T.W., Siddique, Z.: Product family design and platform-based product development: a state-of-the-art review. J. Intell. Manuf. 18, 5–29 (2007).
19. Martinez, M.T., Favrel, J., Fhodous, P.: Product Family Manufacturing Plan Generation and Classification, (2000).
20. Schierholt, K.: Process configuration — combining the principles of product configuration and process planning. In: Mertins, K., Krause, O., and Schallock, B. (eds.) Global Production Management. pp. 391–398. Springer US (1999).
21. ISA-95 | ISA-95 is the international standard for the integration of enterprise and control systems ISA-95.01 Models & Terminology, http://www.isa-95.com/subpages/technology/isa-95/isa-95-01.php.
22. Deutsches Institute für Normung e. V.: Manufacturing Processes Joining, (2003).