Product Specification for Flexible Workflow Orchestrations in Service Oriented Holonic Manufacturing Systems

Francisco Gamboa Quintanilla, Olivier Cardin, and Pierre 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
{francisco.gamboa,olivier.cardin,
pierre.castagna}@univ-nantes.fr

Abstract. Holonic Manufacturing Systems are a response solution for the emergent need of flexible, reactive and productive manufacturing systems. This paper relies on PROSA, a classical holonic reference architecture, which makes use of a product specification, a process specification and a means to determine a resource's production abilities and capacity, but does not define a specific method for representing such. This paper proposes an approach to define a product's process specification model that integrates the principles and advantages of Service-oriented Architectures, Petri-Nets and Product Families. Then, a re-definition of the basic holons is given to have a glimpse on a possible exploitation of this new approach, together with a short-term forecasting strategy, for the flexible orchestration of workflows. Finally, it is shown how the proposed product's process specification model enhances the HMS's flexibility, reactivity and productivity giving rise to a Service-oriented Holonic Manufacturing System.

Keywords: Holonic Manufacturing, PROSA, Petri-Nets, Product specification, Manufacturing-Services, Process families, Workflow exploration.

1 Introduction

For the last few decades it has been seen an evolution in the goods market trend which manifests with an increasing demand of customized products. This evolution has been boosted by the rise of the e-commerce market which makes available customization platforms to customers via internet. Companies, in their search to compete in the marketplace, have been looking for ways to expand their production lines and differentiate their offer with the belief of improving their sales [1]. However, as [2] pointed out, as variety increases the law of diminishing returns does not keep pace. Thus, the problem of Customization i.e. reaching Production Efficiency (PE), relies on process design, whose main concern is manufacturability and cost. For this reason, to attain PE and respond to product variety, the next generation

T. Borangiu et al. (eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics*, Studies in Computational Intelligence 544,
DOI: 10.1007/978-3-319-04735-5_12,

Manufacturing Execution Systems (MES) should provide increased levels of flexibility, re-configurability and intelligence [3].

Holonic Manufacturing Systems (HMS) has been recognized as a paradigm providing to MES the above mentioned attributes by means of a decentralized architecture. Such attributes are obtained thanks to the identification and recognition of autonomous intelligent entities, each one attributed with subset of the various responsibilities in the system. These entities, called holons, are capable of cooperating with other entities and organize themselves for the achievement of a specific goal. PROSA [5], a reference holonic architecture, identifies and classifies three main holon roles, i.e. product, resource and order holons, each one in charge of managing a part of the production control system such as: product and process specifications, resources' capacity utilization and logistics, respectively.

PROSA, in its definition, structures the design principles of a HMS. Such definition recognizes the existence of a product specification, process specification and the capacity of determining a resource's production abilities and capacity. However it does not establish specific tools to implement these. The objective of this paper is to propose a modelling strategy for a product's process specification that welcomes product customization and enhances the HMS' flexible, reactive and productive potential to attain production efficiency. A second objective is to propose a methodology to determine the Resource Holons' (RHs) production capabilities that can interface with the proposed product's process specification.

The second section of this article gives a brief description of the type of system of application. The third section of this article deals with the specification of the product, leading to a model including product and process families' specification. This section ends up with the proposal of using Petri Nets in order to represent product recipes. Finally the fourth section is intended to show how these new concepts can be integrated into a HMS with the use of SOA's principles for flexible workflow orchestration.

2 Description of System of Application

Before going further in introducing the work presented in this paper, it is important to have a look on what kind of systems this work is addressing. This will introduce the reader into the context, in order to have a better understanding of the ideas and concepts discussed in this paper.

This work is mainly directed to companies needing to implement new production systems with enough flexibility to produce a great product variety that the new trend of product customization implies. Such need of flexibility comes from the idea that such flexibility will translate into a greater competitiveness in terms of product quality, speed of product delivery, greater product offer, and the ability to introduce faster new products into production and market. Hence, this work is intended for those companies looking to push to the limit the efficiency of their production systems while keeping a high degree of flexibility in opposition to companies seeking for high volume productions where the system's physical configuration inclines more towards

continuous flow production lines which favour high production flows by scarifying flexibility.

Due to product customization and the great product variety that it engenders, there is a high uncertainty in product demands. For this reason this work is mainly directed to production systems implementing a push strategy or Make-to-Order (MTO) strategy where orders can arrive at any moment during production time, requesting estimations on delivery dates. These so called emergent orders impose a dynamic behaviour to the systems, changing its state with each new arrival. Such dynamism makes the implementation of traditional scheduling systems not a viable solution as it makes its calculations based on a static state therefore having to recalculate with each new arrival. The degree of the MTO strategy can be either an Assemble-to-Order (ATO) or a Build-to-Order (BTO) strategy where product parts are already available or they can be ordered as production orders arrive. Orders can come in small batches or individual products as it is the customer who submits them.

The intended system of application owns a physical topology resembling that of a Flexible Job-Shop (FJS). This is natural, as a job- shop is typically the initial production floor configuration for manufacturers willing to offer a variety of products thus; needing of flexibility. The constraints remain the same: all jobs are formed of a certain number of manufacturing operations that can be executed by one or more of the resources available in the production floor.

The production floor is composed of three major components: a set of workstations, a transportation system and a set of work-in-process (WIP) products. It is a multi-station manufacturing system where there exists more than one workstation or machine capable of doing one same operation. Such workstations can count with stocks of materials or sub-products that will be needed to provide a certain manufacturing transformation to products in order to allow the implementation of an ATO and/or BTO strategy.

The transportation system gives physical interconnectivity between the different workstation. Due to the FJS characteristic of having more than one production sequence, the transport system is considered to be a multi-routing system where products can follow jumbled routings among the different workstations. Such routing system might not be designed to have full reachability, thus it is considered the possibility of non-reachable physical states.

Due to the great number of product variants that can exists in the work-in-process, products are considered to possess an appropriate identification for its proper treatment and in order to keep track of its production evolution. A product in the WIP can use of auto-identification technologies as proposed in [14], in order to communicate its identification to the system so that its environment can interact with it accordingly.

Taking into account system integration, the possibility of a system integrating all types of different technologies will be considered. For instance, a work station could well be an automated machine, an automated work cell or an operator in a manual or semi-automated workstation. Although re-configurability is out of the scope of this paper, this aspect will later be demonstrated for adding flexibility in the system's reconfiguration process.

Here are some characteristics or assumptions on the manufacturing process:

- Operations are non-preemptive. Once a manufacturing operation has started it cannot be interrupted unless the product in question is going to be fully discarded.
- There is no parallelism in the execution of manufacturing operations for one product. Parallelism is only present in an indirect way with the simultaneous production of composing sub-products.

Productivity in a Job-Shop production system is strongly linked to its physical layout. One of the main challenges is to design a layout that minimizes material handling costs, process inventories, idle times and that attaints full reachability. The proposed product's process specification, presented in this paper, intends to exploit to a maximum the intrinsic flexibility of the manufacturing process itself according to its precedence rule with no regard on the physical resources. The strategies on how this flexibility will be exploited for the formation of workflows in terms of sequence and providers is out of the scope of this paper, however section 4 will give a slight insight on a possible solution.

3 Product Specification

3.1 Product Families

In their attempt to achieve mass customization, companies face the problem of an increased internal complexity due to a gain in product variety which raises production cost [2]. In order to solve this complexity problem and achieve economy of scale while satisfying Customer Needs (CN), companies have been adopting the development of product families, which seems to be a well-recognized solution to keep competiveness in the marketplace [6].

The principle of product families' development is based on the exploitation of the inherent commonality between different product variants. A product family refers to a set of individual products that share a set of common structural characteristics and yet are differentiated one from another by certain specific features [5]. Such commonality among product structures inside a product family inherently enables commonality as well in the corresponding production process [6]. This gives origin to process families, which in turn takes advantage on the existing commonality in operations and sequences among the different family members.

A process family is therefore a collection of manufacturing tasks that respond to the realization of the corresponding structural modules within a modular product architecture. Process families, in the same way as product families, carry the attributes of commonality, modularity, reutilization and scalability [1] [5] [7]. This process specification is actually the production recipe of a product family member and thanks to its modular nature, it can be reconfigured into different sequences/workflows, also called process orchestrations.

All in all, a product data model, in a customization production system implementing product families, comprises a product family specification (physical domain) and a process family specification (manufacturing domain).

3.2 Product Model

In the context of product customization and product families design, there are two main challenges in the organization of a product's production data model. First, instead of being a collection of individual product variants, the model should explain the relationships between these variants. Second, an individual product variant should be defined out of the selection of the parameters related to the product family. Such parameters are the result of a customer specification process, i.e. product customization. Thus, a specific description of a product's variant production process is a function of both parameters specification and a process family description. For such, the following Product Manufacturing model, Fig.1, is proposed for representing the process family of a specific product family.

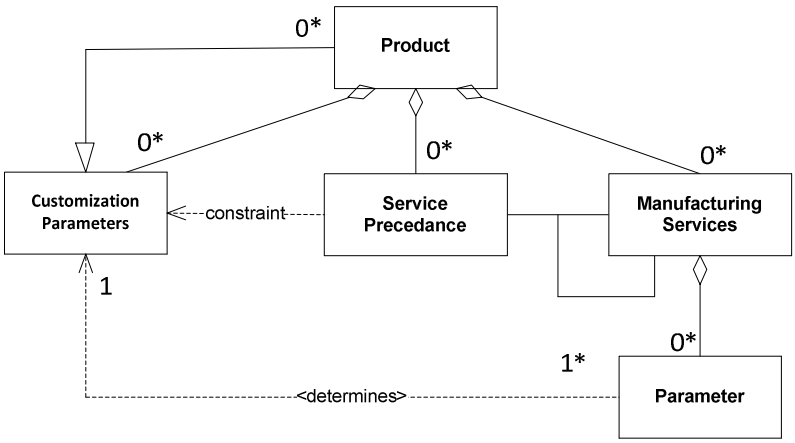

Fig. 1. UML Product Manufacturing-Model

This model is based on the product model described in ISA SP-95 standard, which contains all the necessary information for the manufacture of a product. To adapt this model to product families and product customization, the information is clustered in three main classes:

- *Customization Parameters:* Collection of variables to be defined by the customization process for a given product family. They possess an identifier and a range of allowed values corresponding to a structural module. The level of customization for a given product family is reflected by the cardinality of this class: more customizable parameters exist, more product variants a family represents. The instantiation of all customizable parameters of a product family results in a list of defined parameters for a product variant. Examples of these parameters are: laptop colour,

hard disk capacity, type of screen, type of Wi-Fi antenna, type of keyboard, optional Bluetooth, etc. These will then be mapped into process parameters which are explained later with the Manufacturing-Service class.

- *Manufacturing-Service:* Represents a manufacturing-task or group of tasks forming part of the products production process; it results from the mapping of a given structural module from the physical domain into the process domain. It is a manufacturing process module describing manufacturing transformation ability with no regard of the methods and technology for its implementation. This class also contains the class *Parameters* which correspond to certain variables needed to be determined for the correct execution of the manufacturing task. These service parameters are determined according to the choices made for the customization parameters in order to reproduce such specifications.
- *Manufacturing-Service Precedence / Production Conditions:* Information explaining the relation and interdependencies between the different Manufacturing Services forming the given process family. It represents the precedence rules between services for the orchestration of production workflows. Its cardinality can be zero, considering the possibility of the existence of a non-decoupled production process characterized by a single manufacturing service (no need of precedence).

It is therefore the instantiation of these three elements that completely determine the information required for the realization of a product variant. Such specification is independent of the physical platform as manufacturing services are mere operation descriptions with no consideration of the resources or methods implementing them. This quality makes the product manufacturing-specification compatible with all types of resource models as long as they can provide the required services.

Product differentiation is then achieved by both; parameter specification and configuration of the different manufacturing service modules through the addition, subtraction and/or substitution of these. In this manner, the model explains the process family description through services and their interdependencies, customer specification through customization parameters specification, and the bill of manufacturing services for configurable product families as some manufacturing services can be held out for some family members.

3.3 Manufacturing-Services

As mentioned before, manufacturing services are the result of the direct mapping of structural features of a product family into the process domain. Such services, as stated in [9] for Service-oriented Architectures, represent a single operation or a series of operations of more or less intangible nature, that normally take place in the interactions between a customer and a provider, given as a solution for a customer problem. Services can then be standardized and a bank of these reusable services (operations) can be created for further reutilization in case of existing commonality with other product families.

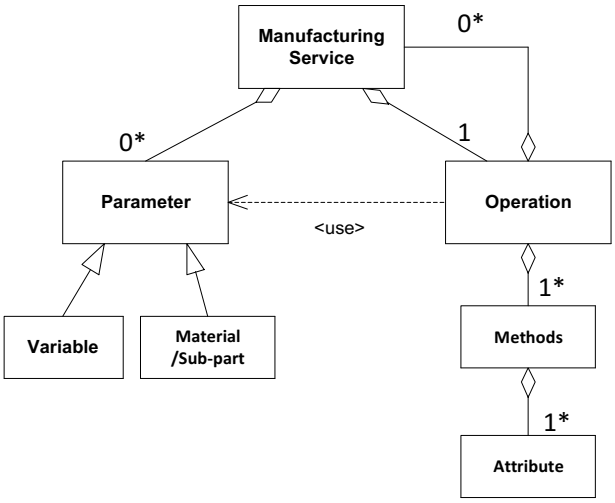

Fig. 2. UML Manufacturing-Service Model; (provider's perspective)

As the Manufacturing-Service model shows, Fig. 2, a **manufacturing service** is composed of two elements:

- *Operation*: Represent the activities related to the service. From the consumer perspective, these are descriptions of the transformations made on the product, more generally, a service identifier. From the provider's perspective, this is the function with the algorithms executing the service, as shown in Fig 2. Contrary to service description, service algorithms are proprietary to the provider and therefore dependent on the resource's technology. The service has a list of attributes on which its performance can be evaluated. Such attributes values are different for each of the methods that implement the service as it is seen in the aggregation relation with the Method class. For instance, an evaluation attribute could be the operation energy consumption. A machine using newer technology could consume less energy than other machines when providing the same manufacturing service.
- *Parameter*: They can be of two forms: variables and materials. In the first case, they are variables with a range of allowed values corresponding to a design parameter from the physical domain (e.g., element X positioned by coordinates). In the latter case, they indicate the category of the component to be added to the main product by the operation. The selection of the material or sub product is done inside a range of component variants belonging to the same category (e.g., *Category*: hard disk, *Range*: {200 GB, 300 GB, 400 GB, 1Tb}). It is determined after the customization process and used to fully determine the operation description.

It is the decoupling of parameters from the operation that allows bringing product customization to the process domain. Therefore, as product customization is based on the reutilization of structural features, the manufacturing services used to produce such features can be adapted to the different product variants and/or families through

service parameterization. The group of services that can be derived from this model can be seen as a family sharing a same operation description but differentiated by the values in their parameterization.

3.4 Sequence Modelling through Petri-Nets

As mentioned in the Product Manufacturing-model, a product specification should express all the information needed for the manufacture of a specific product: parameters, manufacturing services and service precedence conditions. Its modelling tool should be capable of expressing all the possible service choreographies (workflows) that can produce the specified product. In addition, it must also facilitate the online edition, be easy to understand and program and should have low memory footprint for potential embedded applications.

Traditionally in process design, static models are implemented by specifying only one single predefined production plan. Mendes et al. in [13] enrich the process model by considering the existence of different alternative services for a given production state. They used the Petri-Nets formalism as modelling tool with the objective of involving decision motors in the system. The approach proposed in this article has the intention of going farther in enriching the process model by increasing the decision area by questioning the order of execution of tasks. Petri Nets, a well-known modelling formalism in the academic and industrial domain, turns out to be a very good candidate for this purpose. This is mainly due to its characteristic ability to capture the synchronous and asynchronous aspects between the manufacturing services involved in the production process. Thanks to this and to their evolution mechanisms, Petri-Nets have the capacity of representing a great number of sequence combinations with a single net. This is of great importance as the main goal is to design a product manufacturing information model that allows the exploitation of the HMS's inherent flexibility, which in turn will project through the exploration of all the possible alternative production workflows that produces a specific product.

The proposed product production model is represented using a Petri-Net extension which includes the inhibitor arcs and the test arcs. For more information on the Petri-Net formalism, refer to [12]. In this model, manufacturing services are represented by the net's transitions. The different production states are indicated by the net markings. These markings, implicitly determine the manufacturing services that have been executed at a given point. The service interdependencies/precedence rules are inherently defined by the collection of arcs relating places and transitions and the evolution rules of the Petri-Nets formalism. Test arcs together with the inclusion of permission places are used to indicate the selection of the optional modules that differentiate versions of the product i.e. product sub-families. A token is added to the permission places of those optional services that have been selected to be included in the process. Inhibitor arcs on the other hand are used to include more complex precedence conditions like those of mutual exclusion and negation which can be especially present in chemical processes. Finally, the set of parameters is added as attributes associated to the manufacturing services (transitions).

To better understand the Petri-Net approach, Figs. 3-5 show an illustrative example of a theoretical process family using Legos. It consists of a Lego platform representing the transporter of the product, and a set of blocks each one standing as an instance of a theoretical service type. Three types of Legos are used to represent such types and are differentiated according to their sizes as can be seen in Fig.3. Manufacturing service type 1 is represented by a 2x2 Lego block, a service type 2 by a 2x4 and finally a service type 3 by a 2x6 Lego block. Each service has a set of parameters for its instance from which a subset is not customizable and is defined by the process designers. The other subset of parameters is used to capture the customers' choices in the process domain; they include the scalable aspects of customization into the production process while test arcs and permissions places capture the configuration aspects of product customization. The idea behind this example is to illustrate the structural interdependencies in the Lego structure and to illustrate the process family precedence conditions and how this can be represented with a Petri-Net.

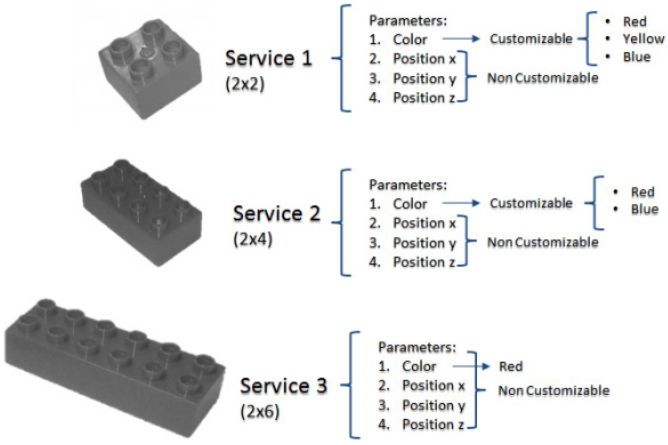

Fig. 3. Types of Services

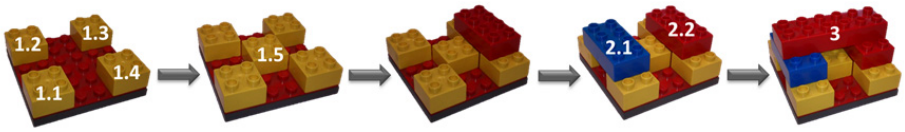

Fig. 4. Product Base Configuration

In this example, the colour of the Lego block is the customizable parameter while its coordinates are set a priori during process design creating an instance of the service type. The base configuration of the product family is exposed in Fig.4. Such base sub-product can then be transformed into a member of one of the different product sub-families, as shown in Fig.5, belonging to the process family to be modelled. As can be seen, it can result in two different versions. Version 1 includes two service-type 1 instances which are differentiated by their coordinates from those instances in

version 2. This shows a possibility to configure (available for the customer), which in this case would be of mutual exclusion. Therefore, this product family is customizable in colour; each of the blocks of the structure can take one of the colours available, as indicated in Fig.3, plus a certain customization level in the product structure defined by the optional modules issuing a version 1 or a version 2 product.

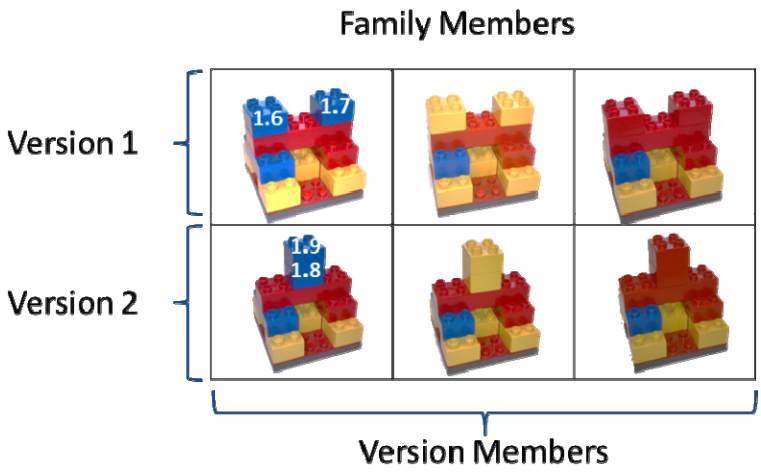

Fig. 5. Product Variants of a Family

As it is illustrated in Fig.4, the base configuration of the Lego product is realized by the application of 5 instances of service type 1 {1.1, 1.2, 1.3, 1.4, 1.5}, two instances of service type 2 {2.1, 2.2} and one instance of service type 3 {3}. Then parting from such base configuration, versions 1 and two can be realized by the application of two service type 1: {1.6, 1.7} and {1.8, 1.9}. Table 1 contains the bill of manufacturing services involved in the realization of the given product family and the precedence conditions for each one of them. Using the Lego structure to demonstrate the interdependencies between services, it can be seen that services 1.1 through 1.5 are independent of any other service - other than the existence of a platform. This same independence is expressed in Table 1 with a zero precedence condition. Service 2.1 and 2.2 are directly dependent on the previous application of services 1.1&1.2 and 1.3&1.4 respectively. However, in order to make a richer example, a more complex precedence condition is used as indicated in Table1. In this case the main interest is to avoid the situation of applying services 2.1 and 2.2 before service 1.5 has been applied. This can represent a scenario with a physical limitation like that of a robotic arm, in an assembly process, being unable to place block 1.5 once blocks 2.1 and 2.2 have been placed. Hence, the possible sequences (with respect to this three services) are {1.5→2.1→2.2}, {1.5→2.2→2.1}, {2.1→1.5→2.2} and {2.2→1.5→2.1}. It is important to note that complex conditions like this should be expressed in canonical form as a sum of minterms. Service 3, on the other hand, has a compound condition. It depends on the previous application of services 2.1 and 2.2, followed by the rest of the services in the list with its single precedence conditions.

Table 1. Example: Manufacturing Services Precedence Table

Manufacturing Service Type	Precedence Condition
Service 1.1	-
Service 1.2	-
Service 1.3	-
Service 1.4	-
Service 1.5	-
Service 2.1	(1.1*1.2* $\overline{2.2}$) | (1.1*1.2*1.5* 2.2)
Service 2.2	(1.3*1.4* $\overline{2.1}$) | (1.3*1.4*1.5* 2.1)
Service 3	(2.1*2.2)
Service 1.6	3
Service 1.7	3
Service 1.8	3
Service 1.9	1.8

Serving from the precedence table and a series of modelling rules, the following Petri-net structure can be derived (Fig.6). As mentioned before, the production state of the product in question is given by the net's marking which enables the triggering of certain transitions. Thanks to this, at a certain production state, the allowed manufacturing services to be next executed, that will respect the service precedence rules, can be known. This allows the exploration of the alternatives given by the asynchrony between certain services. The selection of production modules, resulting from the personalization process, is done by adding tokens to those services forming part of the different product versions. From the example, if version 1 is to be done, a token will be added to the permission place linked to both Services 1.6 and 1.7 by the test arcs which will avoid the rehabilitation of those transitions.

In short, a single Petri-Net can generate a state-automaton representing the arborescence of all possible production workflows while consuming a small amount of memory and a more straight forward programming and design.

4 Integration into SoHMS

The proposed approach for modelling product specification through services and Petri-Nets gives origin to a Service-oriented Holonic Manufacturing System (SoHMS). This takes the core of a Service-oriented Architecture (SOA) with the provider and customer entities having a holonic behaviour with roles defined by PROSA. Next, there will be explained some of the activities of the PROSA's basic Holons, indicating how to integrate the proposed approach and how this gives answer to the paper' objectives: enhancing the HMS flexibility, reactivity and productivity and a way to determine the RHs production capabilities.

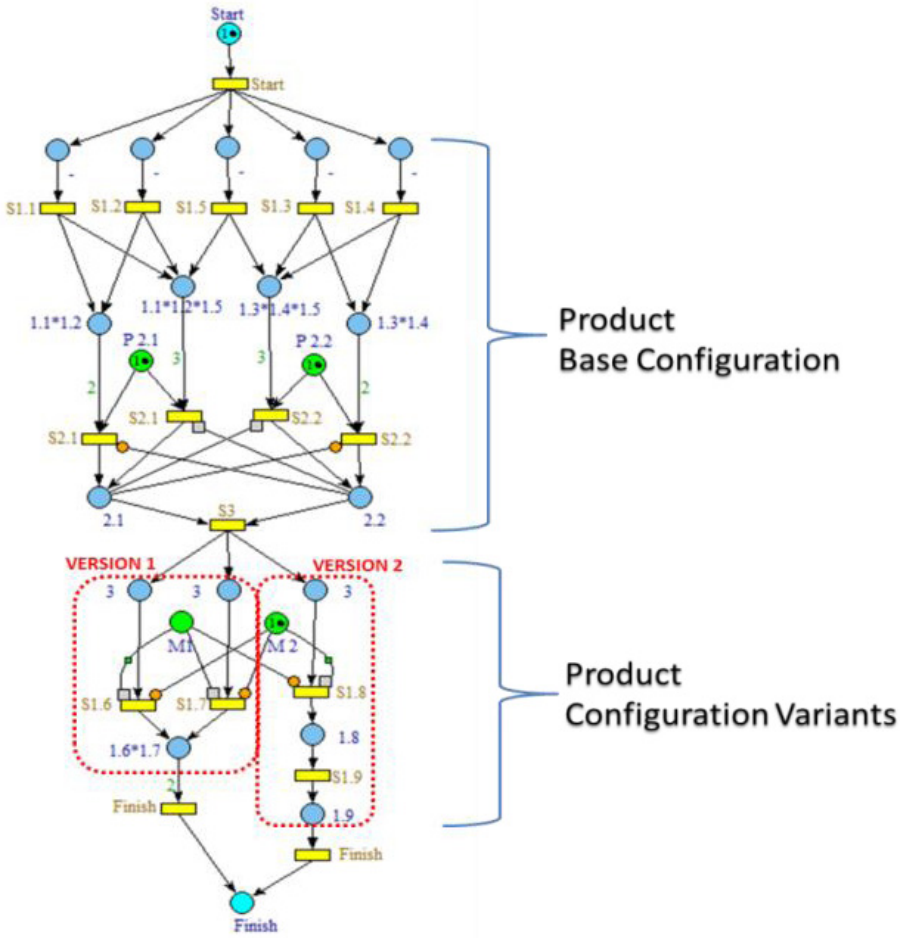

Fig. 6. Petri-Net Product Manufacturing Model

4.1 Holons' Roles

Product Holon (PH)
The PH, as in PROSA, contains the product specification using the present approach through Petri-Nets, Services and parameters specification. However, instead of standing as just an informational server, the PH leaves its passive character and adopts a more active one by involving itself in the decision process. Its main responsibility is then the exploration of the best possible production solutions according to the rules inherently expressed in the Petri-Net production recipe. It is also responsible for the evaluation and the selection of the best explored solution according to a certain criterion, e.g. the due date of the service. Exploration is done in two stages: prior to the order launching and during order production, with the intention of re-evaluating the system's state and react to changes by proposing new solutions based on the present state. Such decision is then communicated to the OH for their execution.

Order Holon (OH)
The OH contains the solution selected by the PH in the form of an execution table with all the information related to their proper execution, i.e. time constraints and service contractors. Its main responsibility is to ensure the proper execution of a task or series of tasks in the manufacturing system.

It is in charge of the product routing through the production plant (factory) from one resource to another according to the physical ports indicated in an execution table issued by the PH after workflow exploration. As production evolves, it notifies the production state to the associated PH for a continuous evaluation of new alternatives. Similarly, as production goes on, it sends intention confirmations to all the contractors to maintain contracts valid (reservations) and waits for their acknowledgments. Such reservations have a limited lifetime and become invalid in case of not being updated with a certain frequency. This is done in order to detect changes in the system's state as first stated in [11].

Resource Holon (RH)
A RH is a virtual representation of the physical resources that provide production capabilities in the factory floor. Such virtualization can be of one or of a group of physical resources for which manufacturing functions have been pre-programmed according to their internal models. In the same way as the agents in the operator level defined in [10], the RH can offer services that involve the interaction of various machines with a shared physical environment. The main idea behind this is that, by the unification of the individual physical resources' abilities, more complex manufacturing services could be offered, thus augmenting the manufacturing abilities of the SoHMS.

In contrast to PROSA's initial definition, it does not contain the controller of the resource. Its main activity in the HMS platform is the exposition of services and the negotiation for the allocation of the resources' activities according to a specific criterion (e.g. maximize resource utilization). On a lower level, there is a corresponding RH that can be called the "Operator-RH" (inspired by [10]), which contains the list of pre-programmed functions for the cluster of physical resources forming an RH. This operator-RH contains the utilization time-table of each of the physical resources involved, that the higher level RH accesses to manage its allocation. This separation is important as both activities - resource allocation and service execution - require execution environments with different time constraints.

4.2 Holonic Interaction

Fig.7 shows a UML sequence diagram to express, in a general way, the production lifecycle of a single product. It all starts with the exploration of the possible production sequences according to the product specification Petri-Net.

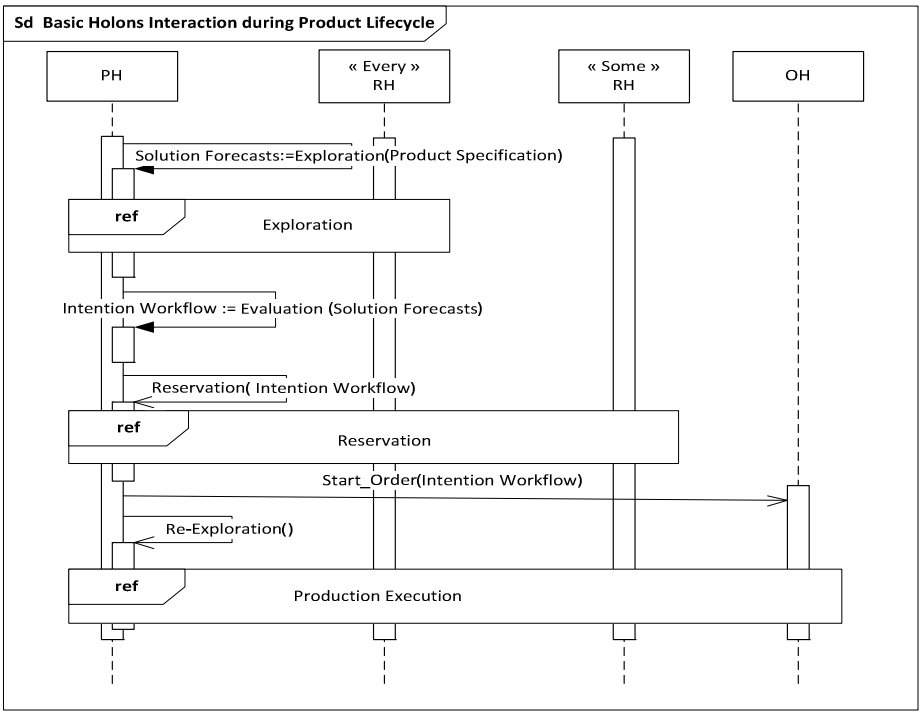

Fig. 7. Order Launching in a SoHMS

The issue of such exploration is a set of different trajectories that can achieve the production of a terminated product. Such trajectories are differentiated one from another by the order of execution of their services according to their precedence rules, the candidate service providers and the estimates of their execution. Once the exploration is concluded, the PH evaluates the alternative solutions according to a specific criterion (depending on interests) and selects one solution, making it the intention workflow. After defining the production workflow, the PH attributes contracts to establish reservations based on the RHs' service proposals and waits for contract establishments confirming the validity of the production plan. Once the intention has been established, the PH passes the confirmed intention to the associated OH for managing its execution through the production factory. At the same time, the PH enters in a mode of reactivity to re-explore and re-establish a new solution in the case of a disturbance on the system that invalidates the original production workflow.

Exploration of Alternative Workflows

Exploration starts by attributing an initial marking to the product specification Petri-Net. The marking, as stated before, represents production possibility states, i.e. states that do not depend on history but on the permissible future actions. At a given marking, the set of enabled transitions represent the permissible services that can be executed for that production state. The PH then sends tasks announcements to the cloud

of RH requesting for proposals coming from the RHs capable of providing such services. These are then added to the solution graph and continue exploration in time of each of these proposals in terms of sequence and of provider selection. This process continues until all possible trajectories are explored, generating an automaton with solution forecasts, as illustrated in Fig. 8. Such exploration process is based in the emergent short term forecasting approach proposed in [11].

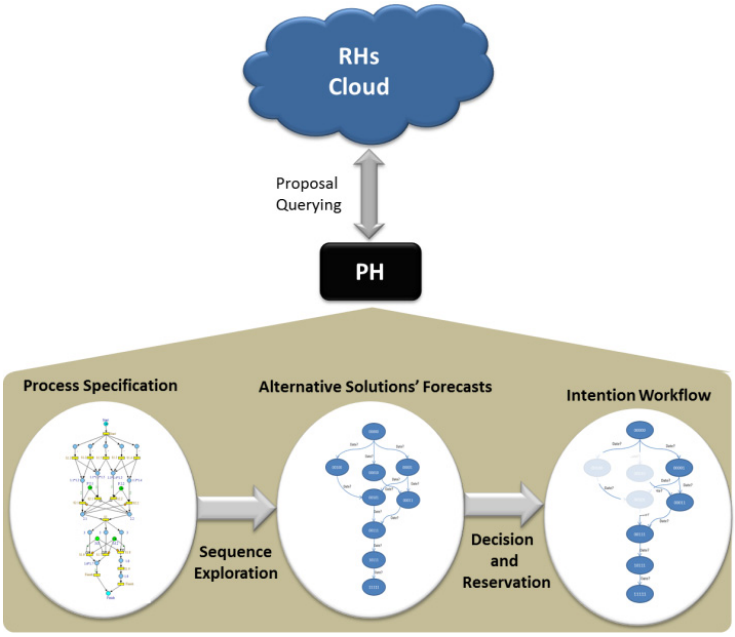

Fig. 8. Workflow Exploration and Selection[1]

In this way, the exploration process concludes with an arborescence of possible solutions according to the systems present state. Later on, by the use of graph search algorithms, the evaluation of the best solution can be done for their reservation and subsequent execution as presented above.

Continuous Exploration

One of the main requirements for SoHMS is a good reactivity to production disturbances. For this reason the PH enters in a reactive mode that enables it to re-start the exploration for new possible workflows, again in terms of service sequence, provider and time of execution, which could have a better performance than the original solution proposed. This re-exploration for new solutions can be done in a periodic manner and/or in an event-driven manner starting from the present production state:

[1] The state diagram arborescence shown in this image does not correspond to the Petri-Net in the example. This was simplified for visual reasons.

- *Periodic Re-Exploration*: The exploration of new alternatives is made every fixed period of time in order to detect changes in the system, that are not directly related to the PH in question. Such changes have an indirect impact in the product production plan and aren't directly notified to the PH's to trigger exploration. Examples are: changes of intention of other PH, breakdowns of RHs out of the PHs holarchy (non-contractor RHs), etc.
- *Event-Driven Re-Exploration*: Exploration is triggered by the notification of a disturbance involving one of the holons directly associated to the PH production plan (holons belonging to the same holarchy). In this case if an associated RH suffers a disturbance, the PH can immediately start exploration of new feasible alternatives.

5 Conclusion and Perspectives

Using Petri-Nets and the concept of Services in an HMS, creating a SoHMS brings several advantages to the control system. The Petri-Nets, thanks to its great expressiveness of the synchronous and asynchronous aspects between manufacturing tasks, allows the PH to explore all possible production solutions. This advantage adds flexibility to the system as it is not limited by the production specification but by the constraints inherent to the product's production interdependencies. Equally important, they express with great simplicity a potential explosion of production trajectories that would be difficult to model otherwise, as such arborescence depends on the combinatorial nature of manufacturing services and on contractor selection. On the other hand, the inclusion of the concept of services, giving rise to the SoHMS, introduces a clear and unified way to describe manufacturing tasks as means to determine accurately a RH's manufacturing capability, based on the task nature more than on the resources' model. This facilitates the introduction of different resource technologies as it is independent of the technology used. Moreover, it welcomes the customer involvement in the manufacturing specification and imports the advantages of service reutilization (as in product families) for cost reduction and faster design to production time. Finally, the short-term forecasting approach for the exploration of production alternatives represents a step for augmenting the systems productivity and bringing it close to optimality, by augmenting the vision of the system for the whole production lifecycle of the product.

On future work, more detail will be added on the reactive mechanism for the re-exploration of new solutions due to disturbances. For such, also social behaviour rules will be defined in order to avoid chaotic interactions in the system [11].

Work is also to be done in defining the RH model and behaviour algorithms for scheduling local resources such as to maximize their utilization or other criteria. Due to the potential explosion of alternative solutions, the exploration of all these can be time consuming and difficult to compute. In such case, machine learning algorithms could help in identifying those trajectories that give the best results according to the evaluation criterion used and limit the exploration of new solutions to a reasonable number around these solution areas.

References

1. Jiao, J., Simpson, T.W.: Product family design and platform-based product development: a state-of-the-art review. Journal of intelligent Manufacturing 18, 5–29 (2007)
2. Child, P., Diedrich, R., Sanders, F.H., Wisniowski, S.: Sloan Management Review. SMR forum: The Management of Complexity 33, 73–80 (1991)
3. Molina, A., Rodriguez, C.A., Ahuett, H., Cortes, J.A., Ramirez, M., Jiménez, G., Martinez, S.: Next-generation manufacturing systems: key research issues in developing and integrating reconfigurable and intelligent machines. International Journal of Computer Integrated Manufacturing 18(7), 525–536 (2005)
4. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.: Reference architecture for holonic manufacturing systems: PROSA. Computers in Industry 37, 255–274 (1998)
5. Meyer, M., Lehnerd, A.P.: The power of product platform- building value and cost leadship. Free Press, New York (1997)
6. Martinez, M.T., Favrel, J., Fhodous, P.: Product family manufacturing plan generation and classification. Concurrent Engineering: Research and Applications 8(1), 12–22 (2000)
7. Simpson, T.W., Maier, J.R.A., Mistree, F.: Product platform design: Method and application. Research in Engineering Design 13(1), 2–22 (2001)
8. Jiao, J., Tseng, M., Duffy, V., Lin, F.: Product Family Modeling for Mass Customization. Computers and Industrial Engineering 35(3-4), 495–498 (1998)
9. Grönroos, C.: Service Management and Marketing. A customer relationship management approach, 2nd edn. Wiley, Chichester (2001)
10. Nagorny, K., et al.: A service- and multi-agent-oriented manufacturing automation architecture. Computers in Industry (2012), http://dx.doi.org/10.1016/j.compind.2012.08.003
11. Valckenaers, P., Karuna, H., Saint Germain, B., Verstraete, P., Van Brussel, H.: Emergent short-term forecasting through ant colony engineering in coordination and control systems. Advanced Engineering Informatics 20(3), 261–278 (2006)
12. Cassandras, C., Lafortune, S.: Introduction to Discrete Event Systems. Springer Science (2008)
13. Mendes, J.M., Leitão, P., Restivo, F., Colombo, A.W.: Process Optimization of Service-Oriented Automation Devices Based on Petri Nets, pp. 274–279. IEEE (2010), doi:978-1-4244-7300-7
14. Wong, C.Y., McFarlane, D., Zaharudin, A.A., Agarwal, V.: The Intelligent Product Driven Supply Chain. In: IEEE Systems Man and Cybernetics, Hammammet, Tunisia (2002)