



HAL
open science

Configuration for mass customization: how to extend product configuration towards requirements and process configuration

Michel Aldanondo, Élise Vareilles

► To cite this version:

Michel Aldanondo, Élise Vareilles. Configuration for mass customization: how to extend product configuration towards requirements and process configuration. *Journal of Intelligent Manufacturing*, 2008, 19 (5), p. 521-535. 10.1007/s10845-008-0135-z . hal-01599442

HAL Id: hal-01599442

<https://hal.science/hal-01599442>

Submitted on 17 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Configuration for mass customization: how to extend product configuration towards requirements and process configuration

Michel Aldanondo · Elise Vareilles

Abstract In order to develop mass customization, many companies use configuration software to customize their products. Although many studies already exist about Product Configuration, Requirements and Process Configuration have not been studied in detail. As all these three aspects must be considered for mass customization, the aim of this paper is to show how Product Configuration, when considered as a constraint satisfaction problem, can be extended upstream towards Requirements Configuration and downstream towards Process Configuration. Product Configuration basics are first reviewed thanks to a constraint based approach, and an analysis of industrial configuration situations is done in order to clarify mass customization needs in terms of configuration. Then upstream Requirements Configuration and downstream Process Configuration are defined and generic models are proposed. It is shown that the proposed elements allow a global and consistent flow of configuration activities. A detailed example illustrates the different configuration problems and a discussion terminates the paper.

Keywords Configuration · Constraint satisfaction problem · Product modeling · Process modeling · Manufacturing process

Introduction

Dealing with product diversity is nowadays a strong requirement for companies fighting in a world market. This increas-

ing diversity comes from the need to be customer specific, in other word try to match the exact requirement of each customer, without including too many unnecessary elements. As the decomposition of the total demand in various demand segments is not always enough, many companies try to organize and set up mass customization techniques. In order to do so, some of them try to include configuration software (configurator) in their information system.

Most of the scientific studies completed on configuration mainly consider what can be called Product Configuration corresponding with the configuration of the product bill-of-materials. This means that the product is only considered as a set of components in a ‘physical view’. As mass customization needs to cover the management of the whole customizable product cycle from customer order to final manufacturing, it is necessary to extend the use of configuration techniques from Product Configuration to upstream Requirements Configuration and downstream Process Configuration.

The aim of this paper is therefore to show how Product Configuration, when considered as a constraint satisfaction problem, can be extended to a global configuration approach covering the cycle Requirements, Product and Process Configuration.

To this end the paper is structured as follows: the section “Product configuration and mass customization” recalls general elements relevant to Product Configuration and shows how it can be considered as a constraint satisfaction problem. The section “product configuration extension with requirements configuration” is concerned with the ‘up-stream’ extension taking into account Requirements Configuration thanks to a functional or descriptive approach of the product. The section “Product configuration extension with process configuration” deals with the ‘down-stream’ extension and considers Process Configuration thanks to a description of the process which gathers routing and

M. Aldanondo (✉) · E. Vareilles
Ecole des Mines d’Albi-Carmaux, Centre Génie Industriel,
Campus Jarlard, Albi 81000, France
e-mail: Michel.Aldanondo@enstimac.fr

E. Vareilles
e-mail: Elise.Vareilles@enstimac.fr

operations. All proposed elements are illustrated with an example that runs through out the paper and a discussion concludes the paper.

Product configuration and mass customization

Product Configuration definition is recalled and illustrated with an example. Then some discussion permits us to characterize configuration for mass customization and to introduce the needs of Requirements and Process Configuration.

Main elements of product configuration

The first sub-section recalls and discusses the definition of the Product Configuration problem and introduces the example. Then it shows how this problem can be considered as a constraint satisfaction problem (CSP) and how a configuration task can be achieved.

Product configuration problem and configurator

From all the existing studies concerning configuration, it seems that some common features defining Product Configuration could be:

- hypothesis: a product is a set of components,
- given:
 - (i) a generic model of a configurable product able to represent a family of products with all possible variants and options, that gathers:
 - (1) a set of component groups and relevant component quantities,
 - (2) a set of various constraints that restricts possible combinations of components and/or component quantity values,
 - (ii) a set of customer requirements, where a requirement corresponds with a selection of a component or a quantity of this component,
- Product Configuring can be defined as ‘finding at least one set of components that satisfies all constraints and customer requirements’.

These elements can be found and are discussed in various definitions proposed in [Mittal and Frayman \(1989\)](#), [Sabin and Weigel \(1998\)](#), [Soininen et al. \(1998\)](#) and [Aldanondo et al. \(2003\)](#). It is important to note that, according to this definition, the configuration result is a set of components or a single level bill-of-materials.

A configurator is a piece of software that assists the person in charge of the configuration task. It is composed of a knowledge base that stores the generic model of the prod-

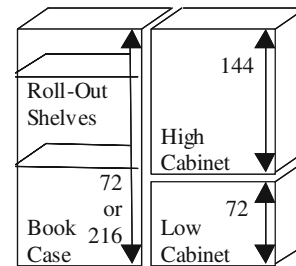


Fig. 1 Custom storage system example

uct and a set of assistance tools that help the user to find a solution or to select components. In any case, the basic common requirement, in terms of assistance, is to guarantee that the configured product is consistent (all constraints are respected) with the generic model and the requirements, during and at the end of the configuration task.

Product example

The proposed example is a simple Custom Storage System (CSS) shown in Fig. 1. Four groups of components exist in our example: the Book Case (BC) with up to four Roll-Out-Shelves (RO), the High Cabinet (HC) and the Low Cabinet (LC). The Book Case is available in two heights: 72 cm or 216 cm. All components exist in two finishes: Painted (P) or Wood (W).

Therefore the groups of components are composed as follows:

- Book Case (BC): {BC72P, BC72W, BC216P, BC216W}.
- High Cabinet (HC): {HC144P, HC144W}.
- Low Cabinet (LC): {LC72P, LC72W}.
- Roll-Out-Shelves (RO): {ROP, ROW}.

In addition, some constraints exist and reduce configuration possibilities:

- All components of any Custom Storage System (CSS) must have the same finish,
- The Book Case (BC) must be present in all configured Custom Storage Systems (CSS),
- The Low Cabinet (LC) is present according to the user requirement,
- The High Cabinet (HC) can exist if (i) a Low Cabinet (LC) exists, (ii) the Book Case (BC) is 216 cm high and (iii) if the user wants it,
- The quantity of Roll-Out-Shelves (RO) can be {1, 2} or {1, 2, 3, 4} with respect to the Book Case (BC) height = 72 or 216.

Therefore, the possible quantities of each component are:

- QBC: {1}
- QHC: {0, 1}

- QLC: {0, 1}
- QRO: {1, 2, 3, 4}

Product configuration as a constraint satisfaction problem (CSP)

CSP, defined by Montanari (1974) as a triplet $\{X, D, C\}$ where X is a set of variables, D a set of finite domains (one for each variable) and C a set of compatibility constraints (defining the possible or forbidden combinations of variable values), partially matches this problem.

Each group of components and each component quantity is associated with a variable. Each component and each component quantity value corresponds with one value of the variable. It is assumed that when the only possible component quantity value is one, there is no component quantity variable. The constraint represents the allowed or excluded combinations of components and/or component quantity values.

The Dynamic extension of the CSP, DCSP proposed by Mittal and Falkenhainer (1990), discussed in Sabin and Freuder (1999), improved by Soinenen and Gelle (1999) and frequently called now Conditional-CSP, introduces the notions of:

- Initially active variables: variables that exist in any configured product,
- Compatibility constraints: equivalent to the CSP constraints defined by Montanari,
- Activity constraints allowing the control of the existence of non initially active variable in the following ways:
 - Require: a specified value of a variable 'X' implies the existence of variable 'Y',
 - Not Require: a specified value of a variable 'X' implies the non-existence of variable 'Y'.

DCSP allows the modulation of the existence of any variable corresponding with a group of components or a component quantity.

The next sub-section shows how these elements allow the design of the generic model of the Product Configuration taken as an example.

Product generic model of the product example

With these elements, the example of section "Product example" can be modeled, as described in Fig. 2, and shows:

- Four variables corresponding with the four groups of components: BC, HC, LC, RO.

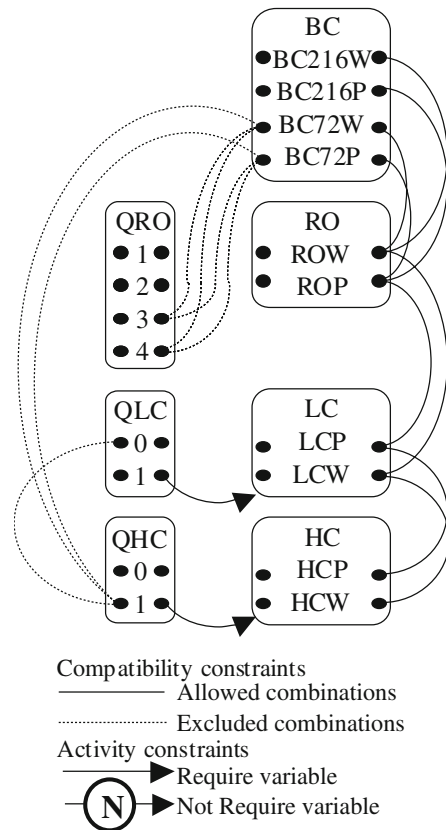


Fig. 2 Model of the example

- As the component quantity relevant to the Book Case (BC) is always one, only three variables corresponding with component quantities are necessary: QHC, QLC, QRO.
- Six compatibility constraints between the followings variables:
 - three constraints indicating the valid or allowed combinations of components with respect to the finishes 'wood' and 'painted': (BC,RO) 4 combinations, (RO, LC) 2 combinations and (LC,HC) 2 combinations,
 - one constraint excluding the RO quantity possibilities '3' and '4' with respect to the height of the book case '72': (BC,QRO) 4 combinations,
 - one constraint excluding the HC quantity possibility '1' with respect to the height of the book case '72': (BC,QHC) 2 combinations,
 - one constraint excluding the HC quantity possibility '1' with respect to the LC quantity possibility '0': (QLC,QHC) 1 combination,
- Two activity constraints implying the existence of the two component groups:
 - Low Cabinet (LC) if LC quantity = '1'
 - High Cabinet (HC) if HC quantity = '1'

Initially active or existing variables are: BC, RO, QRO, QLC and QHC. A user's requirement can be inputted on one of these variables thanks to a variable value selection. Then configuration assistance is provided by constraints propagation mechanisms relying most of the time on adaptations of arc consistency techniques (Mittal and Falkenhainer 1990). AC techniques either reduce the possible values of other variables or trigger existence of non-initially existing variables. This behavior 'user input' / 'constraints propagation' loops in an interactive way, until reaching a problem solution. A solution is found when the problem reaches a state where each existing variable has a single remaining valid value in its definition domain. By valid we mean that all constraints are respected.

This model represents a solution space of 32 configured CSS. An example of configuration process could be:

Existing variables with more than one possible value: BC, RO, QRO, QLC, QHC

- Selection of component Book Case: BC = 'BC216W'

(i) Reduced variable domain:

RO = 'ROW',

Existing variables with more than one possible value:
QRO, QLC, QHC

- Selection of component quantity High Cabinet QHC = '1'

(i) Reduced variable domain:

QLC = '1'

(ii) Activation of non-initially active variables:

Component group LC exists,
Component group HC exists,

(iii) Reduced variable domain:

LC = 'LCW'

HC = 'HCW'

Existing variables with more than one possible value:
QRO

- Selection of component quantity Roll-Out-Shelves QRO = '2'

Existing variables with more than one possible value: none

The resulting configuration is: {1 BC216W, 1 LCW, 1 HCW, 2 ROW}.

Configuration for mass customization

The previous section has recalled the basics of Product Configuration. This section discusses these elements and tries to characterize industrial situations of configuration in order to identify some configuration issues relevant to mass customization.

Characterization of configuration situations in industry

The proposed characterization is based on the following factors.

Product points of view: physical, functional and process
The generic model of Fig. 2 corresponds with what we call a 'physical' model because each variable corresponds with a group of components or a component quantity. The relevant configuration task should therefore be achieved by a person who has a deep product knowledge, because possible choices and associated user's inputs correspond with selections of component references. For a non product expert, who cannot understand what is meant by component references, it is necessary to define 'descriptive' or 'functional' properties that enable him to understand the possible choices and to input functional or descriptive requirements.

For example, the selection of the component 'BC216W' can be expressed with two descriptive properties Book Case Height = 216 cm and Book Case Finish = 'wood'.

As mass customization addresses people who are most of the time not product-expert, it is necessary to add a configuration layer that we have called Requirements Configuration that works with descriptive or functional properties. For this kind of configuration, the generic model gathers product properties and constraints. The result of the Requirements Configuration is a set of properties able to describe the configured product as we will see in section "Product configuration extension with requirements configuration".

It has been shown that the result of the Product Configuration task, or the configured product, is a set of components or a configured single level bill-of-materials. As the solution space of this problem, or the quantity of possible configured bill-of-materials, can be large (the previous example was simple and only contains 32 solutions), it is necessary to consider the configuration of the production process (including assembly and/or manufacturing) in order to achieve fast assembly and delivery required by mass customization. This requires a generic process view of the product allowing what we called Process Configuration. For this kind of configuration, the generic model gathers generic routing, generic operations, resources and constraints. The result of the Process Configuration is a set of operations able to describe the

configured routing relevant to the configured product as we will see in section “Product configuration extension with process configuration”.

Configuration dynamics Configuration Dynamics combines two aspects: the configuration task frequency and the duration of the configuration task. A high dynamics is characterized by a high configuration frequency, for example between 1 and 100 per day and very low configuration task duration, for example between 5 minutes and 1 hour. At the opposite end a low dynamics could correspond with a frequency between 1 and 10 per month and a duration between 1 day and 2 weeks.

Interactive or autonomous configuration process As seen in section “Product generic model of the product example”, configuration assistance has been introduced as an interactive behavior that corresponds with sequential cycles: ‘user input’ / ‘constraints propagation’. This behavior is interesting when it is necessary to understand the remaining possibilities step by step and therefore guide the user towards a solution.

An other approach is to provide all the requirements in one go, meaning that all component selections and relevant quantities that interest the user are given once. Then the configuration assistance tool tries to find, in the solution space, one or all the possible solutions that respect the constraints. This behavior, that can be time consuming, opens optimization possibilities which allow, for example the maximization of some product performance or the minimization of a solution price.

Mixed configuration processes, gathering interactive behavior followed by autonomous behavior, are of course possible.

Complexity and size of product model Industrial configuration situations can be very different in terms of model size and complexity. Large models can require up to thousands of variables while small models can be designed with only ten variables (the proposed example contains seven variables).

Complexity is delicate to quantify but roughly characterizes the degree of interdependency of variables. For example, a problem gathering ten variables and twenty constraints is often more complex than another containing twenty variables and ten constraints (the proposed example gathers seven variables and eight constraints).

Two main configuration situations

According to the previous characteristics, configuration can exist in any customer/supplier relation when defining the product object of a deal. It is nevertheless possible to identify two main classes of industrial configuration situations and to identify the one more practice in mass customization.

The first class is close to a product design activity and is mainly achieved by the design and development teams of companies. In such a case:

- Configuration dynamics is low, due to necessary meetings or phone conversations to clarify numerous requirements and some ‘trial and error’ procedures,
- Autonomous configuration is the most frequently met behavior, mainly due to the strong optimization needs,
- Size and complexity of product model are rather high.

This class of situation corresponds better with ‘business to business’ relations, meaning that both customer and supplier have a deep knowledge of product composition. Therefore, the physical point of view (Product Configuration) is the most important and the two other aspects (Requirements and Process Configuration) are most of the time not present. Two good examples of this situation can be found in [Hvam et al. \(2001\)](#) where the authors deal with the configuration of large cement factories and in [Fleischanderl et al. \(1998\)](#) where the configuration of complex switching telephone systems is considered.

The second class deals with the selling side and is conducted by the sales teams of companies. In that case:

- Configuration dynamics is high, and sometimes very high when processed on line on the web,
- Interactive configuration is the most frequently met behavior in order to guide requirements expression,
- Size and complexity of product model are rather low.

This second class of situation is mostly found in ‘business to customer’ relations, meaning that the customer is the final user of the product. As the customer doesn’t have good product knowledge, the functional point of view (Requirements Configuration) must be present in the configuration task. A high configuration dynamic implies the use of Process Configuration in order to achieve fast assembly and delivery. Typical examples of this situation can be found for example in automotive industry ([Pargamin 2003](#)) or industrial carpentry where this work was achieved [Aldanondo et al. \(2003\)](#).

The reader should be warned about this typology, because it is always possible to find an example that does not match the propositions. But, according to what can be seen in industry, the proposed general tendencies seem quite adequate.

The second class of situation is of course closer to what is called Configuration for Mass Customization. Configuration modules proposed by ERP or CRM software editors try to fulfill configuration problems of this class, they are often called ‘generic’, ‘commercial’ or ‘sales’ configurators as explained in [Haag \(1998\)](#).

In the next sections we will therefore focus our contribution on configuration for mass customization and will show how the initial Product Configuration problem can be extended upstream with Requirements Configuration and downstream with Process Configuration.

Product configuration extension with requirements configuration

This section is concerned with the ‘up-stream’ extension taking into account Requirements Configuration. As far as we know, very few authors have dealt with this idea: [Mannisto et al. \(1996\)](#) consider implicit and explicit product modeling and [Hvam et al. \(2002\)](#) consider a configuration process gathering sale and Product Configuration, but these two studies do not provide detailed modeling elements. We have explained in section “Characterization of configuration situations in industry” that configuring with a physical approach is not easy for users who are not experts in product composition. In order to enable these users to use configuration techniques, a functional or descriptive view of the product is therefore added to the physical view. As in section “Main elements of product configuration”, the configuration problem is first defined with these two product views, then the example and the constraint formalism for generic modeling are adapted and resulting models are presented and discussed.

Requirements and product configuration problem

The problem definition of section “Product configuration problem and configurator” is lightly modified with the introduction of the notion of ‘product properties’, which allows us to characterize the requirements and to introduce the descriptive view as follows:

- hypothesis: a product is a set of components,
- given:
 - (i) a generic model of a configurable product able to represent a family of products with all possible variants and options, that gathers:
 - (1) a set of component groups,
 - (2) a set of product properties,
 - (3) a set of various constraints that restricts possible combinations of components and/or property values,
 - (ii) a set of customer requirements, where a requirement corresponds with a selection of a component or a property value,
- Requirements and Product Configuring can be defined as ‘finding at least one set of components that satisfies all the constraints and the customer requirements’.

The ‘component quantity’ introduced in section “Product configuration problem and configurator” is considered in this definition as a product property. We must pointed out that component quantity is mainly a physical characteristic of the product but it is often necessary to consider it also as a requirement.

Descriptive view of the product example

The product taken as an example is absolutely the same, but a descriptive view is added. The resulting example is described in the following way:

- all component groups remain unchanged:
 - (i) Book Case (BC): {BC72P, BC72W, BC216P, BC216W},
 - (ii) High Cabinet (HC): {HC144P, HC144W},
 - (iii) Low Cabinet (LC): {LC72P, LC72W},
 - (iv) Roll-Out-Shelves (RO): {ROP, ROW},
- all quantities are now taken into account as product properties and give:
 - (i) Low Cabinet Existence (LCX): {Yes, No},
 - (ii) High Cabinet Existence (HCX): {Yes, No},
 - (iii) Quantity of Roll-Out-Shelves (QRO): {1, 2, 3, 4},
- two properties are added and characterize:
 - (i) Book Case height (BCH): {72, 216}.
 - (ii) Product Finish (PFH): {Wood, Painted}.
- all constraints remain unchanged.

Requirements and product configuration as a constraint satisfaction problem

CSP and DCSP introduced in section “Product configuration as a constraint satisfaction problem (CSP)” match this problem without any modification. Each group of components and each product property is associated with a variable. Each component and each property value corresponds with one value of the variable.

This allows the design of what we call ‘pure’ descriptive generic model of the product that can support what we call Requirements Configuration. This kind of model contains only product properties and relevant constraints.

However as the fundamental goal of configuration is to define the bill-of-materials of the configured product, this ‘pure descriptive model’ must be associated with the ‘physical model’ in order to support both Requirements and Product Configuration.

As these two models represent the same solution space with two different points of view, the constraints reducing the solution space in each view are redundant. In order to suggest some kind of independence of the Requirements Configuration from the Product Configuration, it is better to remove the redundant constraints from the physical view and to keep the constraints in the descriptive view.

In order to achieve Product Configuration with respect to Requirements Configuration, these two models should be associated by other constraints. They in fact replace the redundant constraints of the physical view and provide what

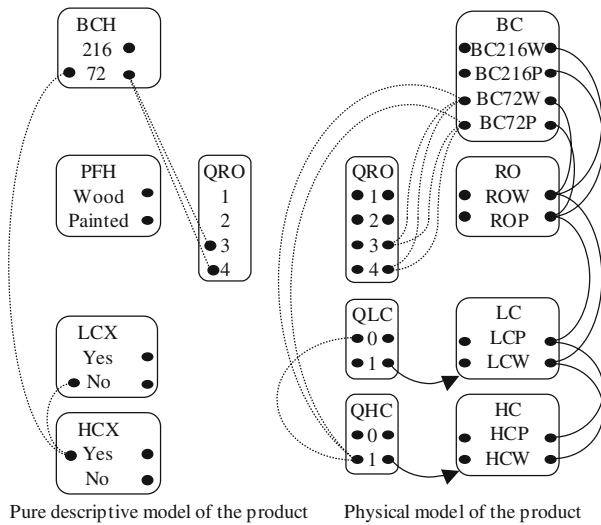


Fig. 3 Pure descriptive model and physical model

we call a ‘descriptive and physical’ model. In the study by Mannisto et al. (1996), this is addressed as an explicit model (component oriented) versus an implicit model (description oriented).

These models are shown in the next section with the example in Fig. 3.

Product generic model gathering descriptive and physical views

The ‘pure’ descriptive generic model and the Requirements Configuration process are first addressed. Then the ‘descriptive and physical’ generic model and relevant Requirements and Product Configuration process are illustrated.

Pure descriptive model and requirements of the product example

This pure descriptive model is presented in the left part of Fig. 3, while the physical model of Fig. 2 is duplicated in the right part of this same figure for comparison. The ‘pure descriptive generic model’ gathers:

- Five variables corresponding with the five product properties: BCH, PFH, LCX, HCX, QRO.
- Three compatibility constraints between the following variables:
 - (i) one constraint excluding the RO quantity possibilities ‘3’ and ‘4’ with respect to the height of the book case ‘72’: (BCH,QRO) 2 combinations,
 - (ii) one constraint excluding the HC existence possibility with respect to the height of the book case ‘72’: (BC, HCX) 1 combination,

- (iii) one constraint excluding the HC existence possibility if the LC does not exist: (LCX,HCX) 1 combination.

In the pure descriptive model the five variables are initially active. This descriptive view covers the exact same combinatory of 32 solutions. It is clear that proposed properties give a much better readability of the solution space to a user who is not a product expert. This descriptive model supports only Requirements Configuration without taking into account component selection. This is one reason for the simplicity of this descriptive model when it is compared with the physical one (right of Fig. 3).

The example of configuration process of section “Product generic model of the product example” would give with this approach:

Existing variables with more than one possible value: BCH, PFH, QRO, LCX, HCX,

- Selection of Book Case Height: BCH = ‘216’
 - (i) Reduced variable domain, none Existing variables with more than one possible value: PFH, QRO, LCX, HCX,
- Selection of Product Finish: PFH = ‘Wood’
 - (i) Reduced variable domain, none Existing variables with more than one possible value: QRO, LCX, HCX,
- Selection of High Cabinet existence: HCX = ‘Yes’
 - (i) Reduced variable domain LCX = ‘Yes’ Existing variables with more than one possible value: QRO,
- Selection of quantity Roll-Out-Shelves QRO = ‘2’ Existing variables with more than one possible value: none

The resulting configured requirements are:

{BCH = ‘216’, PFH = ‘Wood’, HCX = ‘Yes’,
LCX = ‘Yes’, QRO = ‘2’}

Descriptive and physical model of the product example

The ‘descriptive and physical generic model’, which permits us to associate Requirements and Product Configuration, is represented in Fig. 4. This model gathers together the two following model parts.

The pure descriptive generic model of section 3.4.1 without any modification (left part of Fig. 4):

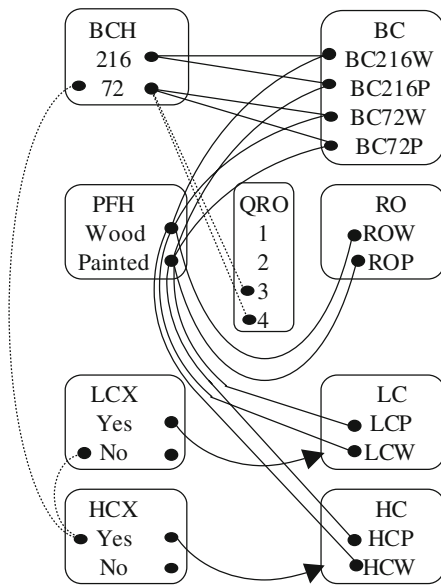


Fig. 4 Descriptive and physical model

- Five variables corresponding with the five product properties: BCH, PFH, LCX, HCX, QRO.
- Three compatibility constraints between the following variables:
 - (i) one constraint excluding the RO quantity possibilities '3' and '4' with respect to the height of the book case '72': (BCH,QRO) 2 combinations,
 - (ii) one constraint excluding the HC existence possibility, with respect to the height of the book case '72': (BC,HCX) 1 combination,
 - (iii) one constraint excluding the HC existence possibility if the LC does not exist: (LCX,HCX) 1 combination.

The physical generic model, where all constraints and component quantity variables have been removed (right part of Fig. 4):

- Four variables corresponding with the component groups: BC, RO, HC, LC.
- Eight constraints that link the two model views:
 - (i) four compatibility constraints reducing component selection with respect to finish
 - (1) one for the Book Case selection: (PFH,BC) 4 combinations,
 - (2) one for the Roll-Out-Shelves selection: (PFH,RO) 2 combinations,
 - (3) one for the Low Cabinet selection: (PFH,LC) 2 combinations,
 - (4) one for the High Cabinet selection: (PFH,HC) 2 combinations,

- (ii) one compatibility constraint selecting the correct Book Case with respect to the height of the book case (BCH,BC) 4 combinations,
- (iii) two activity constraints implying the existence of the component groups:
 - (1) High Cabinet (HC) if high cabinet existence property (HCX) = yes
 - (2) Low Cabinet (LC) if low cabinet existence property (LCX) = yes

The solution space is still the same and contains 32 solutions. The set of requirements {BCH = '216', PFH = 'Wood', HCX = 'Yes', LCX = 'Yes', QRO = '2'} generates the bill-of-materials or component set {1 BC216W, 1 LCW, 1 HCW, 2 ROW}.

Conclusion

The aim of this section was to extend Product Configuration to upstream Requirements Configuration. For this purpose a functional or descriptive modeling approach has been proposed and associated with the physical product view. It has been shown that this approach allows the consistent process of Requirements Configuration and Product Configuration. In order to be clear, this presentation considers only one level of modeling for both physical and descriptive representations. But hierarchical modeling for both models is possible and allows hierarchical function models and multi-level bill of materials as explained by Felfernig et al. (2000) or Junker and Mailharro (2003). The next section will deal with the configuration of the relevant manufacturing process.

Product configuration extension with process configuration

This section deals with the down-stream extension of Product Configuration and considers Process Configuration thanks to a description of the process gathering routings and operations. The first sub-section describes what we mean by Process Configuration, introduces the notions of routing and operation and presents a modeling short cut. Then it shows how routing configuration (section "Routing configuration") and operation configuration (section "Operation configuration") can be considered as a constraint satisfaction problem. In each section the links with the Product Configuration model will be explained.

Main elements of process configuration

Very few authors have considered this problem. Hvam et al. (2002) showed the interest of Process Configuration in the global enterprise view, but did not provide any constraint

modeling elements. Some modeling elements can be found in [Aldanondo et al. \(2004\)](#).

The process is associated with what we call a ‘Process Routing’. A process routing is a set of ‘Operations’ linked with ‘Anteriority constraints’. An operation is defined by a set of ‘Required Resources’ that allows the operation to be processed, each required resource is characterized by a ‘Resource quantity’.

These elements allow us to characterize and introduce the Process Configuration view or routing configuration as follows:

- hypothesis: a routing is a set of operations linked with anteriority constraints, an operation is a set of resources characterized by a required quantity,
- given:
 - (i) a generic model of a configurable routing able to represent a family of production processes with all possible variants and options, that gathers:
 - (1) a set of operations,
 - (2) a set of resources with a required quantity,
 - (3) a set of various constraints that restricts possible combinations of operations, resources and required quantities,
 - (ii) a set of inputs, where an input corresponds with a selection of an operation, a resource or a quantity value,
- routing configuring can be defined as ‘finding at least one set of operations with relevant sets of pairs (resource, quantity) that satisfies all the constraints and the inputs’.

As Process Configuration will be achieved with respect to Product Configuration, the existence of an operation in a configured process routing can depend on the configured product. As an operation is defined by a set of required resources, it is necessary to modulate the existence of the set of required resources. In order to represent these existence constraints in an easy to understand model, a modeling short cut, inspired by the Composite CSP extension proposed by [Sabin and Freuder \(1996\)](#) is defined here. This short cut triggers the existence of a CSP sub model.

Let us consider a sub-model SM, gathering:

- a set of variables initially existing in SM: SM_V1 , {variables $sm_v1.i$ },
- a set of variables initially not existing in SM: SM_V2 , {variables $sm_v2.j$ },
- a set of compatibility constraints between variables $\in SM_V = SM_V1 \cup SM_V2$
- a set of activity constraints conditioning the existence of the variables belonging to SM_V2 written as follows:

$$\forall sm_v2.j \in SM_V2: (vk = Cste-k) \rightarrow sm_v2.j, \text{ with } vk \in SM_V$$

When the existence of the sub-model SM is conditioned by a variable value, for example variable $SM-X = \text{‘Yes’}$, all variables of sub-model SM should be constrained as follows:

- the variables $sm_v1.i \in SM_V1$ become initially not existing and should be conditioned by the activity constraint:

$$\forall sm_v1.i \in SM_V1: (SM-X = \text{‘Yes’}) \rightarrow sm_v1.i,$$

- the activity constraints acting on variables $sm_v2.j \in SM_V2$ are cumulated with the constraint $SM-X = \text{‘Yes’}$, as follows:

$$\forall sm_v2.j \in SM_V2: (SM-X = \text{‘Yes’}) \wedge (vk = Cste-k) \rightarrow sm_v2.j, \text{ with } vk \in SM_V$$

We call this modeling ‘short cut’, that allows an uncomplicated representation, a ‘sub-model-activation-constraint’. [Fig. 5](#) shows the representation with the short cut on the left and the model with the detailed constraints on the right.

Routing configuration

We deal first with routing configuration modeling elements, then the routing model of the example of section “Product configuration extension with requirements configuration” is presented in detail.

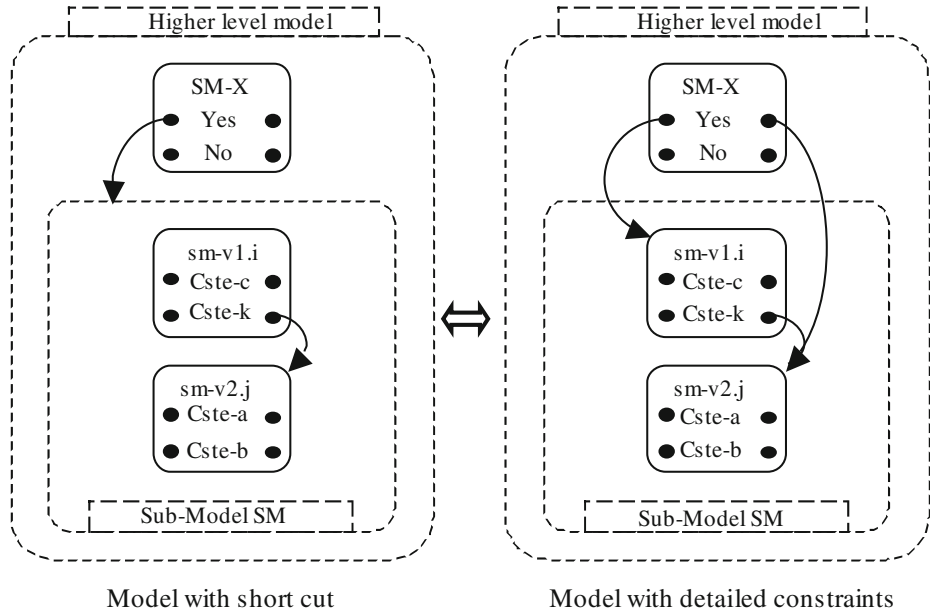
Routing configuration as a constraint satisfaction problem

A ‘Generic-Operation’ variable is associated with each generic operation (Op). A generic operation is a group of operations, where a group can contain a single operation. According to the configured product, an operation can be selected among the group of operations defined in the generic operation.

The representation of the sequencing of generic operations, with generic anteriority constraints, is achieved with the ‘port’ notion (Po) early proposed by [Mittal and Frayman \(1989\)](#) for Product Configuration. This notion allows representing links between components, ‘port variables’ are associated to components and constraints reduce connection possibilities between components through these ports.

Therefore, we propose to associate some ‘Operation-Port’ (OpPo) variables to each generic operation. These Operation-Port variables are either ‘Previous’ (OpPo–) or ‘Next’ (OpPo+), meaning that the values of each of these variables are respectively the ports of next (OpPo+) or previous (OpPo–) operation-ports that can be connected.

Fig. 5 Sub-model activation constraint



A generic operation that has no previous port variable (resp. next port variable) is at the beginning (resp. the end) of the routing.

A generic operation that has more than one previous port variables (resp. next port variables) represents some kind of ‘AND convergence’ (resp. ‘AND divergence’) of the production process (assembly or disassembly for example).

Generic-Operation variables and Operation-Port variables are gathered in what we call a ‘generic operation sub model’. According to the configured product data, which corresponds to the input of routing configuration, the existence of a generic operation sub model can be triggered by a sub-model-activation-constraint.

Operation models, or detailed models showing resources requirements are also considered as sub-models (object of the next section). Therefore, each variable corresponding with a generic operation (Op) triggers an operation sub model.

Routing configuration model of the product example

The routing configuration model presented in Fig. 6 is relevant to the Custom Storage System (CSS) product model of Fig. 4. Routing description elements and the corresponding model are presented simultaneously.

Four Generic-Operations variables are proposed:

- Op-BCRO for cutting and assembling the book case and the Roll-Out-Shelves with a single value {Ph1},
- Op-HCLC for cutting and assembling the high and the low cabinet with a single value {Ph2},
- Op-CSS for assembling the whole product with a single value {Ph3},

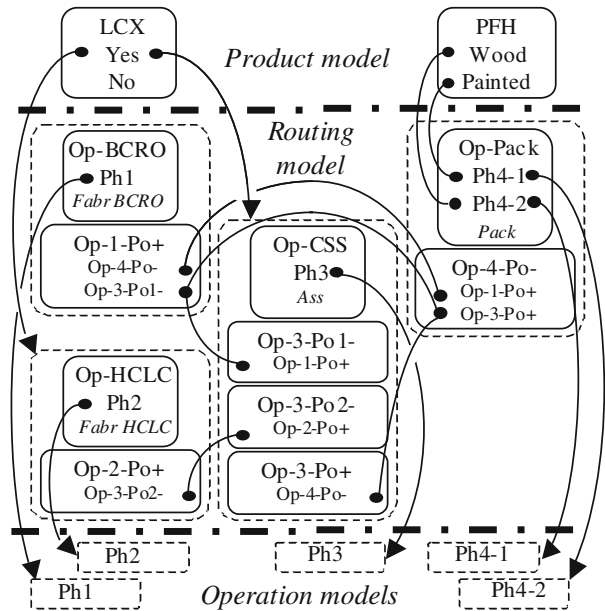


Fig. 6 Process routing model

- Op-Pack for packaging the whole product with two values {Ph4-1, Ph4-2}.

Each generic operation contains a single operation except the packaging where two operations, Ph4-1 and Ph4-2, are different according to the finish of the product. The two generic operation Op-BCRO and Op-Pack always exist in the routing while the two others, Op-HCLC and Op-CSS, exist only if the lower cabinet is present in the configured product.

Operation-ports variables.

- For cutting and assembling the book case and the Roll-Out-Shelves, Op-BCRO:
 - (i) one 'next' port (Op-1-Po+), with two possible values:
 - (1) 'Op-4-Po-', previous port of Op-Pack,
 - (2) 'Op-3-Po1-', previous port of Op-CSS,
 which show that Op-BCRO can be connected to Op-CSS, or directly to Op-Pack if there is no low cabinet (LCX = 'NO'). This shows an 'OR divergence' possibility for 'next' operation anteriority constraints.
- For cutting and assembling the high and the low cabinet, Op-HCLC:
 - (i) one 'next' port (Op-2-Po+), with a single possible value:
 - (1) 'Op-3-Po2-', previous port of Op-CSS,
 which shows that it must be connected to Op-CSS.
- For assembling of the whole product, Op-CSS:
 - (i) two 'previous' ports for Op-CSS (Op-3-Po1-) and (Op-3-Po2-) with respectively a single possible value for each:
 - (1) 'Op-1-Po+', next port of Op-BCRO,
 - (2) 'Op-2-Po+', next port of Op-HCLC,
 These two ports show an 'AND convergence' possibility for 'previous' operation anteriority constraints,
 - (ii) one 'next' port (Op-3-Po+) for Op-CSS, with a single possible value:
 - (1) 'Op-4-Po-', previous port of Op-Pack,
 which shows that it must be connected to Op-Pack
- For packaging of the whole product, Op-Pack:
 - (i) one 'previous' port (Op-4-Po-) for Op-Pack, with two possible values:
 - (1) 'Op-1-Po+', next port of Op-BCRO,
 - (2) 'Op-3-Po+', next port of Op-CSS,
 which represents that Op-Pack can be connected to Op-CSS or directly to Op-BCRO if the low cabinet does not exist. This shows an 'OR convergence' possibility for 'previous' operation anteriority constraints.

The constraints between the port variables only show the compatibility of connections. Two kinds of compatibility can exist:

- the first is rather obvious and indicates that if an operation op-i is before an operation op-j then operation op-j is after operation op-i. In our example if Op-HCLC is before Op-CSS, then Op-CSS is after OP-HCLC.

- the second one is less obvious and indicates that if an op-i is before op-j then op-k must be before op-l (op-j and op-k can be the same operation). In our example if Op-BCRO is before Op-CSS then Op-CSS is before OP-Pack.

The Generic-Operation variables and Operation-ports variables are gathered in four generic operation sub models.

Routing configuration inputs as links with the Product Configuration model.

The two generic operation sub models Op-HCLC and Op-CSS, exist only if the lower cabinet exists in the configured product. This is described with the two sub-model-activation-constraints between product and routing models, in the upper left part of Fig. 6: (LCX = 'Yes' → Op-HCLC) and (LCX = 'Yes' → Op-CSS).

The packaging operation selection (Ph4-1 or Ph4-2) with respect to the finish of the product is achieved thanks to a compatibility constraint between the two variables PFH and Op-Pack (upper right part of Fig. 6).

Links with the operation configuration model

Each generic operation (Ph1, Ph2, Ph3, Ph4.1 and Ph4.2) is linked to a generic operation sub model thanks to a sub-model-activation-constraint in the lower part of Fig. 6.

As the number of 'OR divergence or convergence' is rather small (2), this model represents only the four following routings:

- if the lower cabinet is not present in the configured product:
 - Ph1 → Ph4-1, with a paint finish,
 - Ph1 → Ph4-2, with a wood finish,
- if the lower cabinet is present in the configured product:
 - (Ph1 and Ph2) → Ph3 → Ph4-1, with a paint finish,
 - (Ph1 and Ph2) → Ph3 → Ph4-2, with a wood finish.

This combinatory will increase when operation diversity is taken into account. The customer's requirements taken as an example at the end of the section "Product generic model gathering descriptive and physical views", correspond with the previous routing {BCH = '216', PFH = 'Wood', HCX = 'Yes', LCX = 'Yes', QRO = '2'} and would give the following configured routing:

- Op-BCRO = 'Ph1'
 - Op-1-Po+ = 'Op-3-Po1-': meaning Ph1 then Ph3
- Op-HCLC = 'Ph2'
 - Op-2-Po+ = 'Op-3-Po2-': meaning Ph2 then Ph3
- Op-CSS = 'Ph3'
 - Op-3-Po1- = 'Op-1-Po+': meaning Ph3 after Ph1

Op-3-Po2- = 'Op-2-Po+' : meaning Ph3 after Ph2
 Op-3-Po+ = 'Op-4-Po-' : meaning Ph3 then Ph4-2

- Op-Pack = 'Ph4-2'

Op-4-Po- = 'Op-3-Po+' : meaning Ph4-2 after Ph3

Operation configuration

Modeling elements are first discussed, then a piece of the model of the example is shown in detail.

Operation configuration as a constraint satisfaction problem

An operation is described with a set of required groups of resources (Re) where a set can contain at least a single group of resources. Each required group of resources is associated with a numerical quantity (Qt). A group of resources gathers different resources that can achieve the same kind of processing but with different characteristics (size or speed for example).

An operation is therefore a set of couples of variables {(group of resources, quantity required)}. When there is no 'quantity' variable, it is assumed that a single unit of resource is required. The variables belonging to this set of couples are gathered in what we call an 'operation sub model'. Compatibility constraints can link these variables in order to modulate some associations of resources and/or quantities.

Operation configuration inputs as links between the operation model and the product model:

- Any variable of each operation sub model, group of resources or quantity required, can be linked thanks to compatibility constraints with the product model.

Operation configuration inputs as links between the operation model and the routing model:

- As explained at the end of section "Routing configuration as a constraint satisfaction problem", the existence of each operation sub model is triggered by a 'sub-model-activation-constraint' that allows configuration of the necessary operations.

Example of operation configuration model

These elements can be used for modeling each generic operation of our CSS routing model example. Three operation sub models of (i) the cutting and assembly operation of BCRO or Ph1, and (ii) the packaging of CSS, or Ph4-1 and Ph4-2, are explained and shown in Fig. 7. Operation configuration inputs or constraints between product model, routing model and operation models are also illustrated.

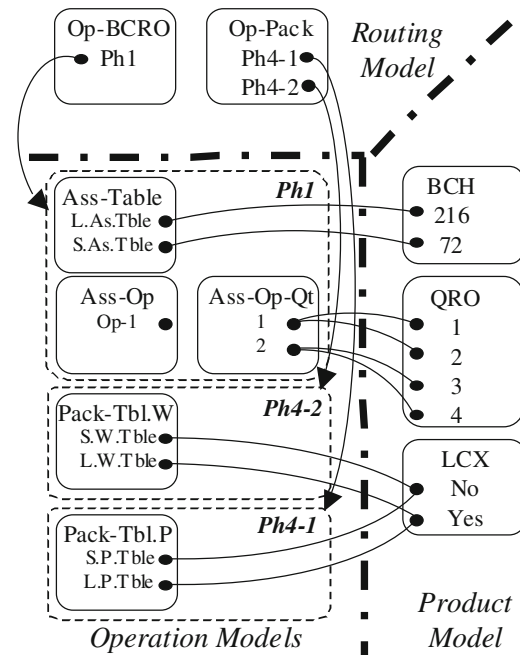


Fig. 7 Operation sub models

Generic operation cutting and assembling BC-RO (Ph1)
 This operation requires two groups of resources:

- Assembly Table (Ass-Table) that gathers:
 - a large table (L.As.Tble),
 - a small table (S.As.Tble).

They are respectively used when the book case height is 216 cm or 72 cm. This is represented thanks to the compatibility constraint between the variables (Ass-Table) and (BCH), between the product model and the operation model. The required quantity of resource is always one.

- Assembly operator (Ass-Op) that contains:
 - a single competency (Op-1).

The possible required quantities of resource (Ass-Op-Qt) are:

- one, if the quantity of Roll-Out-Shelves (QRO) equals 1 or 2,
- two, if the quantity of Roll-Out-Shelves (QRO) equals 3 or 4.

This is shown thanks to the compatibility constraint between variables (Ass-Op-Qt) and (QRO), between the product model and the operation model.

Generic operations packaging the whole product (Ph4-1 and Ph4-2)

Operation Ph4-1 for paint finish requires a single group of resources:

- Packaging table for paint finish (Pack-Tbl.P) containing two resource tables:

- (i) a large one (L.P.Tble),
- (ii) a small one ((S.P.Tble).

They are used respectively if the low cabinet (LC) belongs to the product or not. This is represented by the compatibility constraint between the variables (Pack-Tbl.P) and (LCX), between the product model and the operation model.

Operation Ph4-2 for wood finish is modeled in a similar way with one group of resources:

- Packaging table for wood finish (Pack-Tbl.W) containing two resource tables:

- (i) a large one (L.W.Tble),
- (ii) a small one ((S.W.Tble).

The same kind of constraints exists.

These three operation sub-models are linked with the routing model thanks to 'operation-activation-constraint':

- Op-BCRO = 'Ph1' → Ph1,
- Op-Pack = 'Ph4.1' → Ph4.1,
- Op-Pack = 'Ph4.2' → Ph4.2.

When the constraints of the product model are not taken into account, the number of different configured operations is 4 for operation Ph1 and 2 for operations Ph4-1 and Ph4-2. If we assume only 1 configured operation for both Ph2 and Ph3, the combinatory of each routing described in the end of section "Routing configuration model of the product example" is:

- Ph1 → Ph4-1: $4*2=8$
- Ph1 → Ph4-2: $4*2=8$
- (Ph1 and Ph2) → Ph3 → Ph4-1: $(4*1) 1*2=8$
- (Ph1 and Ph2) → Ph3 → Ph4-2: $(4*1) 1*2=8$

Therefore, the solution space of the process model covers 32 different routings. We should pointed out that if each Ph2 and Ph3 model had represented two configured operations (instead of one), the global combinatory of the process model would reach 80.

When the constraints of the product model are taken into account, the solution space decreases from 32 to 12. Only three instances of operation Ph1 remain valid and only one for Ph4-1 and Ph4-2. This gives the following solution space for the process model when the whole product model is considered:

- Ph1 → Ph4-1: $3*1=6$
- Ph1 → Ph4-2: $3*1=6$

- (Ph1 and Ph2) → Ph3 → Ph4-1: $(3*1) 1*1=3$
- (Ph1 and Ph2) → Ph3 → Ph4-2: $(3*1) 1*1=3$

The customer requirements taken as an example at the end of the section "Routing configuration model of the product example" {BCH = '216', PFH = 'Wood', HCX = 'Yes', LCX = 'Yes', QRO = '2'} and the configured routing would give the following configured operations Ph1 and Ph4-1.:

- Ph1

Ass-Table = 'L.As.Tble', because BCH = '216'

Ass-Op = 'Op-1' with Ass-OP-Qt = '1', because QRO = '2'

- Ph4-1

Pack-Tbl.P = 'L.P.Tble', because LCX = 'Yes'.

Conclusion

We have shown in this section how Process Configuration can be achieved with respect to product model. The approach has been proposed with a single level decomposition of the process (Routing/Operations), but of course it is possible to consider multi-level decomposition as for example Macro-operation/Operation/Instructions, but resources would only appear at the lowest level instructions in this example. The constraint based formalism proposed in section "Product configuration and mass customization" has been re-used without any modification.

Discussion and conclusion

This last section summarizes our study, discusses some operating issues and concludes.

The aim of this paper has been to show how Product Configuration concepts, when considered as a satisfaction problem, can be extended to the cycle Requirements, Product and Process Configuration in order to assist mass customization development.

The first section brought to mind Product Configuration basics and introduced constraint based modeling elements. The modeling framework uses the Dynamic CSP extension or Condition CSP. A characterization of industrial configuration situations has been proposed. It has allowed proposing two main tendencies for the industrial configuration problem: 'Selling situation' and 'Design situation'.

The second section extended Product Configuration to upstream Requirements Configuration. The functional or descriptive point of view that can be taken when considering the product is first introduced. Then this functional approach

has been merged with the physical one in order to have a consistent configuration of both Requirements and Product.

The third section extended previous results towards downstream Process Configuration. A two-levels modeling method combining a routing level and an operation level has been proposed but we have pointed out that this method is also valid for multi-level process decompositions. The notion of port, frequently used for component location in Product Configuration, has been used with success for modeling the anteriority constraints between operations. Manufacturing operations have been modeled as a simple set of resources, more detailed modeling is of course possible. The links between the process model (gathering routing and operation models) and the product model have been identified and permit Process Configuration with respect to the configured product.

In order to illustrate the proposed elements, a detailed example runs through out the paper and show how the three configuration views can be articulated. The constraint based approach permits to clearly identify the different links that exist between the three different configuration views. Furthermore, constraint propagation possibilities allow to quickly identify the consequences of any configuration decision on all configuration variables opening powerful 'what if' configuration strategies. Some solution space calculations are provided in order to underline the potential possibilities of the proposed concepts.

Given the proposed methodology that has been designed to handle the whole configuration process starting with Requirements Configuration and finishing with Process Configuration, some operating issues are interesting to discuss. The first one concerns how the proposed methodology works. The second one deals with the workload or duration necessary to deploy a full configuration solution. The last one considers some impacts of configuration techniques on the product development process.

The proposed methodology has been designed and used in detailed during the deployment of a configuration software in a company producing industrial carpentry products (around 100 selling points, 20 factories and 20.000 components). For this case the three configuration views (functional, physical and process) have been fully modeled and inputted in the software and configuration has been working for five years. The main key result of configuration in the mass customization strategy of this company was the improvement of error avoidance. Given these interesting results, we have been working on various industrial cases and identify two main problems: (i) the presence of numerical variables and numerical constraints that can not be properly handled by constraint based configurators, (ii) the difficulty to identify people for designing generic models. Besides these two problems, the proposed elements have worked rather well.

After the first success in the deployment of the proposed methodology, we try to gather and analyze some deployment

data in order to be able to estimate the workload and duration of the deployment our approach. Mixing estimation techniques results obtained in product design as those proposed by (Bashir and Thomson 1999) and some regression analysis we have been able to identify an order of magnitude estimation of the deployment workload that is detailed in (Aldanondo et al 2004). Globally, configuration software deployment is roughly comparable with ERP deployment, it is very difficult to deploy our method and a configuration solution in less than a year, and the average duration is between one and two years. A very strong influence comes from the level of organization of the company and the knowledge or skill of the actors involved in the deployment project.

Simultaneously with the company level of organization, the product development process has a strong influence on the feasibility and interest of a configuration solution. Product need to be designed with a modular approach or a platform organization. An other key factor that strongly modulates the interest of configuration techniques lies in the stability of the diversity proposed to the customer and the stability of the technical solutions or product components. For both, each time a modification occurs, it is necessary to actualize the configuration models.

From the previous discussion, it is clear that management and organization are important factors for the success of the deployment of configuration techniques in order to achieve mass customization. But modeling and processing product diversity with configurators is the other complementary key issue. For this last issue, the main interest of the methodology described in this paper lies in the constraint based approach that allows a global consistent configuration process from the requirement until product manufacturing. Furthermore, constraints are open to optimization possibilities, preferences management and 'what if' behavior. Some commercial configuration software's operate with these approaches, but not many scientific studies dealing with this aspect have been done. We hope to stimulate research in this area in order to have a global workflow concerning the customizable product cycle. Following Process Configuration, another domain where configuration can become an interesting issue is packaging or distribution configuration.

References

- Aldanondo, M., Hadj-Hamou, K., & Lamothe, J. (2004). Mass customization, configuration and manufacturing. *CIRP Journal of Manufacturing Systems*, 33(4), 381–388. (WISU-Verlag Aache).
- Aldanondo, M., Hadj-Hamou, K., Moynard, G., & Lamothe, J. (2003). Mass customization and configuration: Requirement analysis and constraint based modeling propositions. *Integrated Computer-Aided Engineering*, 10(2), 177–189. (IOS Press).
- Aldanondo, M., Moynard, G., & Hadj-Hamou, K. (2004). Workload estimation formulae for the deployment of commercial configurators. *Proceedings of the International Conference on Economic,*

- Technical and Organizational Aspects of Product Configuration Systems* (pp. 119–128). Copenhagen, Denmark.
- Bashir, H. A., & Thomson, V. (1999). Metrics for design projects: A review. *Design Studies*, 20(3), 263–277.
- Felfernig, A., Friedrich, G., & Jannach, D. (2000). UML as domain specific language for the construction of knowledge-based configuration systems. *International Journal of Software Engineering and Knowledge Engineering*, 10(4), 449–469.
- Fleischanderl, G., Friedrich, G., Haselböck, A., Schreiner, H., & Stumptner, M. (1998). Configuring large systems using generative constraint satisfaction. *IEEE Intelligent Systems*, 13(4), 59–68.
- Haag, A. (1998). Sales configuration in business processes. *IEEE Intelligent Systems*, 13(4), 78–85.
- Hvam, L., Riis, J., & Malis, M. (2002). A multi-perspective approach for the design of configuration systems. *Proceedings of the ECAI 2002 Workshop on Configuration* (pp. 56–62). Lyon, France.
- Hvam, L., Riis, J., Malis, M., & Hansen, B. (2001). Reengineering of the quotation process—application of knowledge based systems procedure for building product models. *Proceedings of the 2001 International Conference on Industrial Engineering and Production Management* (Vol. 1, pp. 242–248). Quebec City, Canada.
- Junker, U., & Mailharro, D. (2003). The Logic of ILOG (J) configurator: Combining constraint programming with a description logic. *Proceedings of the IJCAI 2003 Workshop on Configuration* (pp. 13–20). Acapulco, Mexico.
- Mannisto, T., Peltonen, H., & Sulonen, R. (1996). View to product configuration knowledge modeling and evolution. *Proceedings of the AAAI 1996 Workshop on Configuration* (pp. 111–118). AAAI Press.
- Mittal, S., & Falkenhainer, B. (1990). Dynamic constraint satisfaction problems. *Proceedings of the 9th National Conference on Artificial Intelligence AAAI* (pp. 25–32). Boston, USA.
- Mittal, S., & Frayman, F. (1989). Towards a generic model of configuration tasks. *Proceedings of IJCAI 1989* (Vol. 2, pp. 1395–1401). Detroit, USA.
- Montanari, H. (1974). Networks of constraints: Fundamental properties and application to picture processing. *Information Sciences*, 7, 95–132.
- Pargamin, B. (2002). Vehicle sales configuration: The cluster tree approach. *Proceedings of the ECAI 2002 Workshop on Configuration* (pp. 35–40). Lyon, France.
- Sabin, D., & Freuder, E. (1996). Configuration as composite constraint satisfaction. *Proceedings of the AAAI 1996 Workshop on Configuration* (pp. 28–36). AAAI Press.
- Sabin, M., & Freuder, E. (1999). Detecting and resolving inconsistency and redundancy in conditional constraint satisfaction problems. *Proceedings of AAAI 1999 Workshop on Configuration* (pp. 90–94). Orlando, Florida.
- Sabin, D., & Weigel, R. (1998). Product configuration frameworks—a survey. *IEEE Intelligent Systems*, 13(4), 42–49.
- Soininen, T., & Gelle, E. (1999). Dynamic constraint satisfaction in configuration. *Proceedings of AAAI 1999 Workshop on Configuration* (pp. 95–100).
- Soininen, T., Tiihonen, T., Männistö, T., & Sulonen, R. (1998). Towards a general ontology of configuration. *AIEDAM*, 12(4), 357–372.