



**HAL**  
open science

## Mass customization, configuration and manufacturing

Michel Aldanondo, Khaled Hadj-Hamou, Jacques Lamothe

► **To cite this version:**

Michel Aldanondo, Khaled Hadj-Hamou, Jacques Lamothe. Mass customization, configuration and manufacturing. *Journal of Manufacturing Systems*, 2004, 33 (4), pp.381-388. hal-00452958

**HAL Id: hal-00452958**

**<https://hal.science/hal-00452958>**

Submitted on 8 Nov 2019

**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.

# Mass Customization Configuration and Manufacturing

Michel Aldanondo, Khaled Hadj-Hamou and Jacques Lamothe.  
Centre de Génie Industriel, Ecole des Mines d'Albi-Carmaux  
Campus Jarlard, 81013 Albi CT Cedex 09, France

## Abstract

Mass customization has become a main issue for many companies fighting on a world market. In order to define the customized product that fit each customer need, these companies use configuration software called configurators. Most of these configurators, mainly based on artificial intelligence techniques, are just interested in the product definition without addressing relevant manufacturing problems. The goal of this communication is to show that the same kinds of computer techniques can be used to define customized assembly operations and manufacturing routings.

Keywords: Configuration, Mass customization, Manufacturing data

## 1 INTRODUCTION

Mass customization is now a reality for many products. In order to facilitate the customer/supplier relation, many companies use configurator software to define the customized product that will fit the customer requirements. Most of these configurators, mainly based on artificial intelligence techniques, are just interested in the product definition without addressing relevant manufacturing issues. The goal of this communication is to show that the same kinds of computer techniques can be used to define customized assembly operations and manufacturing routings.

In a first section, we will briefly recall product configuration basics and introduce the need for routing configuration.

Then we will show how it is possible to add a second stage to the product configuration problem that deals with routing configuration.

We will then illustrate how classical interactive configuration process techniques can handle product routings configuration with respect to the customized product definition and the manufacturing requirements.

We will conclude on the interest of the approach in terms of errors avoidance, time to deliver decrease and cost reduction.

## 2 CONFIGURATION BASICS

### 2.1 Configuration and configurator

From all the previous works achieved concerning configuration, it seems that some common features defining configuring 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, in which a generic model is a set of components plus a set of various constraints ; and (ii) a set of customer requirements, in which each requirement can be expressed by a constraint,
- configuring can be defined as "finding at least one component set that satisfies all the constraints".

These elements can be found and discussed in the definitions proposed in [1], [2], and [3]. It is important to note that, according to these definitions, the configuration result is a set of components (or a bill-of-materials).

A configurator is 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 product and a set of assistance tools that helps the user finding a solution or selecting components. In any case, the basic common requirement, in terms of assistance, is to guarantee that the configured product is consistent with the generic model and the requirements, during and at the end of the configuration task.

### 2.2 Generic model and configuration

Most of the works recently achieved on configuration rely on propositional logic, first order logic and constraint satisfaction problem (CSP) frameworks. For generic modeling, we will use the dynamic extension of CSP proposed by Mittal et al [4]. We first present what we call the "central problem" which is handled by most configurators, give an example and corresponding DCSP model.

### Configuration Central Problem

The central problem can be defined as follows :

- h1: all components are "standard" or completely defined, it is not possible to create a new component during the configuration task,
- h2: the components are gathered in groups, each component must belong to only one group,
- h3: a group is either always present in any configured product or its existence depends on:
  - (i) the existence of other groups,
  - (ii) the selection of other components,
  - (iii) the requirements of the configurator user,
- h4: the constraints represent the allowed combinations of:
  - (i) the component selection,
  - (ii) the group existence,
- h5: the requirements of the customer correspond with:
  - (i) the selection of one component in each existing group (in some cases there must be only one remaining component that can be chosen),
  - (ii) the decision of a component group existence,
- h6: a configured product is a set of components satisfying both constraints and requirements, where one and only one component must be selected in each existing group.

### Example of a configuration Central Problem

Our central problem example is a simplified Custom Storage System. Three groups exist in our example: the Book Case (BC), 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: {BC72P, BC72W, BC216P, BC216W}.
- High Cabinet: {HC144P, HC144W}.
- Low Cabinet: {LC72P, LC72W}.

In addition, some constraints are existing on those components:

- The assembly must be with the same finish,
- The Book Case must be presented in all configured Storage Systems.
- The High Cabinet exists if and only if a Low Cabinet exists and the Book Case is 216 cm high.

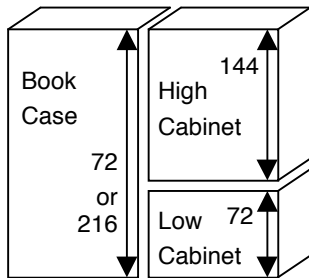


Figure 1. Central problem Example

### Generic model of the central Problem

CSP, defined by Mackworth in [5] 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 constraints (defining the possible combinations of variable values), matches partially this problem. Each group of components is associated with a variable. Each component corresponds with one value of the variable.

The constraint (solid lines of figure 2) represents the allowed combinations of components.

DSCP, proposed in [4] ads notions of:

- Initial variables: variables that exist in any configured product.
- Compatibility constraints: equivalent to the CSP constraints defined by Mackworth.
- Activity constraints: allowing to control the variable existence in the following ways: (i) Require: a specified value of a variable "x" implies the existence of variable "y", (ii) Always Require: any value of a variable "x" (or "x" exists) implies the existence of variable "y", (iii) Not Require : a specified value of a variable "x" implies the non existence of variable "y", (iv) Always Not Require : any value of a variable "x" (or "x" exists) implies the non existence of variable "y".

With these elements, the example of figure 1 can be modeled as described in figure 2.

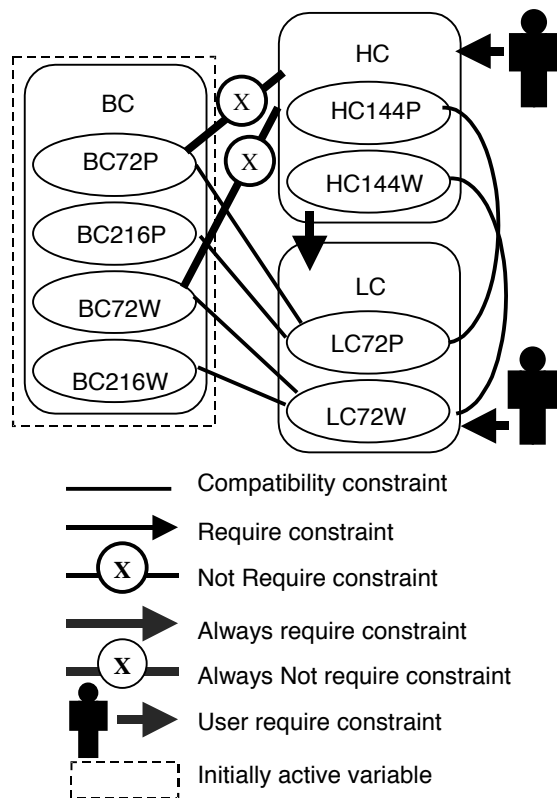


Figure 2. Central problem DCSP modeling

### Configuration processing

Configuration is interactively done with the user. Each time the user either selects a value for a variable or decide about a variable existence, the configurator propagates the requirement with respect to the various constraints as shown in figure 3.

In the upper part of figure 3, the user decides about the existence of a low cabinet (or variable LC existence) and selects the wood finish (selection of component LC72W). The configurator propagates this result (according to the single finish constraint) and removes all the variable values corresponding with painted components.

In the lower part of figure 3, the user selects the smaller book case (BC72W). The configurator removes the High Cabinet variable (according to the high cabinet existence constraint).

The configuration is over and the result is the set composed of the two components BC72W and LC 72W.

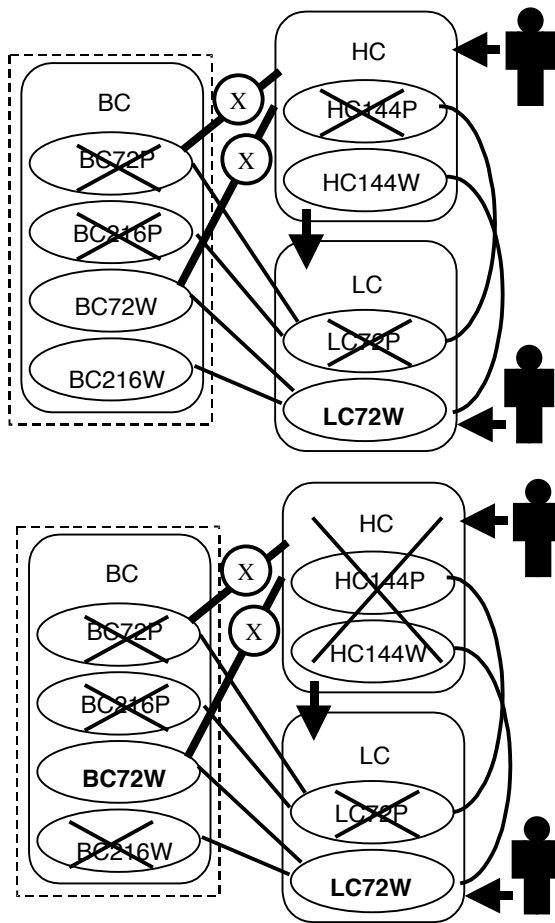


Figure 3 – Interactive configuration process

### 2.3 Routing configuration need.

The previous section has shown how the configuration process assisted by a configurator can be achieved. It is clear that the result of this process consists in the definition of the bill-of-materials of the product that corresponds with the customer requirements while respecting the generic model.

Once the product is defined, the assembly process need to be defined with respect to configured product. In our example, this can mean for example

- According to the kind of finish (wood or painted), the facility used change,
- According to the presence of high and low cabinet in the configured product, a final assembly operation may exist or not.

Therefore a clear need for routing configuration can be identified. But in our case, as all components are standard (hypothesis h1) routing configuration can just deal with assembly, finishing and packaging. Of course this limit does not exist and configuration is not limited to standard component.

The elements presented in section 2.2 were rather simple in order to permit comprehension, many extensions of the central problem has been identified, a complete survey of these extension has been proposed by Aldanondo in [6]. One of these extensions is to be able to configure product with parametric or tailored component.

For example when you arrange a piece of furniture in an old house, you need to have a specific size which rarely corresponds to the standard offer and may not match the "hole in the wall". In order to capture this market, many companies propose product with customization possibilities that are not restricted to standard component assembling, for more information see the survey of Brown in [7].

Therefore a parametric or tailored component is characterized by numerical tailoring attributes with a continuous definition domain defining the range of possible values.

For these tailored components it is necessary to derive for each component instantiation (with specific dimensions) a specific routing and sometimes specific NC machine programs.

For our storage system this can correspond, for example, with the following added requirements:

(i) the height of the Book Case and the High Cabinet can be tailored in a certain range:

$$- 72 \leq BC\_Height \leq 216,$$

$$- 50 \leq HC\_Height \leq 144.$$

(ii) The High Cabinet can exist if :

$$- BC\_Height \geq 72 + 50 = 122$$

(iii) When the High Cabinet exists, its top must be at the same height than the top of the Book Case. Therefore a constraint must specify that:

$$- BC\_Height = HC\_Height + 72.$$

The resulting configuration model could be the one shown in figure 4, where descriptive variables replace component group for an easier reading. It must be noticed that figure 4 possible configurations are exactly the same compared with figure 2 solutions, except of course tailoring or parametric possibilities.

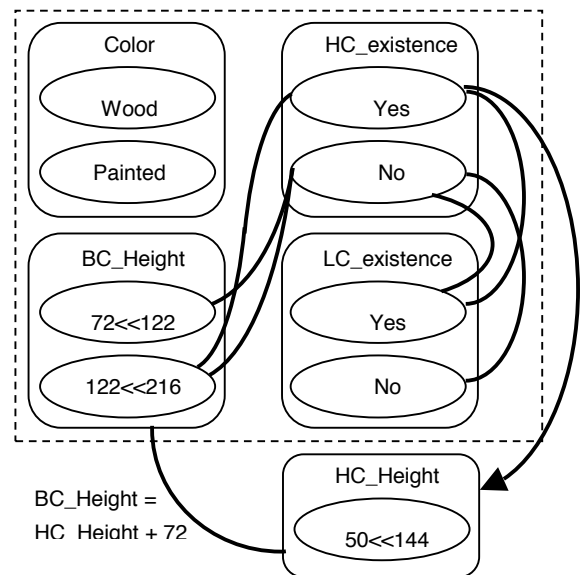


Figure 4 – Model with tailored component

### 2.4 Conclusion

This section has shown configuration basics and introduced the need for routing configuration. The next section will deal with routing configuration.

### 3 ADDING A ROUTING CONFIGURATION STAGE

#### 3.1 Modeling generic routings

In order to use configuration main ideas for process design, it is necessary to define a generic model of the product routing.

A generic routing can be defined as follow :

- (i) a generic routing is a set of generic operations,
- (ii) generic operations are linked with anteriority constraints,
- (iii) existence of each generic operation can be modulated by constraints
- (iv) each generic operation is defined by a set of variables corresponding for example with : facility location, resources names, duration....

For our example, a generic model of the routing for the whole storage system could be the one presented in figure 5. Three kinds of resources exist :

- cutting machine : small one and large one,
- assembly facility : small one and large one,
- finishing facility : wood or paint

and cutting duration may vary according to product finish.

The arrows just represent anteriority constraints for comprehension and should not be considered as DCSP anteriority constraints.

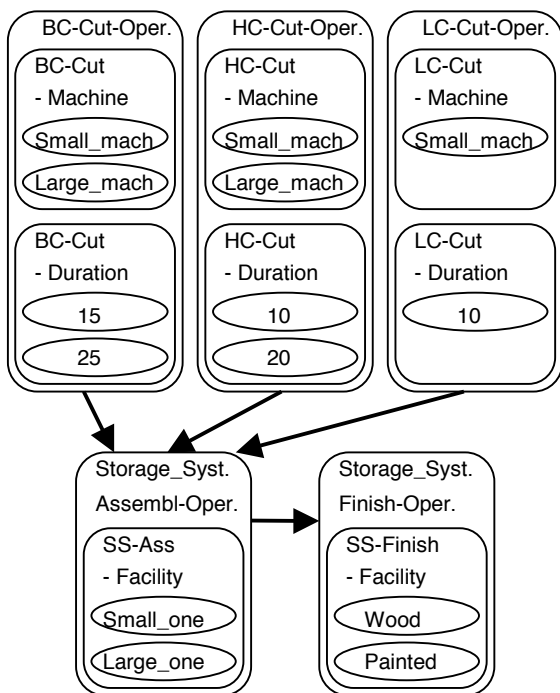


Figure 5 – Generic routing model

#### 3.2 Linking routing and configuration models

Once product configuration is achieved, routing configuration can be processed. As routing depends of configured product characteristics, it is necessary to link the two models.

The operations that exist for any configured product are identified as DCSP initially active variables and the links between the two models correspond with DCSP compatibility and activity constraints.

The two operations that always exist are “BC-Cut-Oper.” And “Storage\_Syst.-Finish-Oper.”. All the other operations exist according to configuration variable values.

The various links are going to be analyzed in the next sub-sections.

#### BC-Cut-Operation

The model links are shown in figure 6. As this operation always exist; there is no activity constraint. The machine used depends of the Height of the Book Case and the operation duration depends on the required kind of finish of the product. Thus, four compatibility constraints link the configuration model and this first operation.

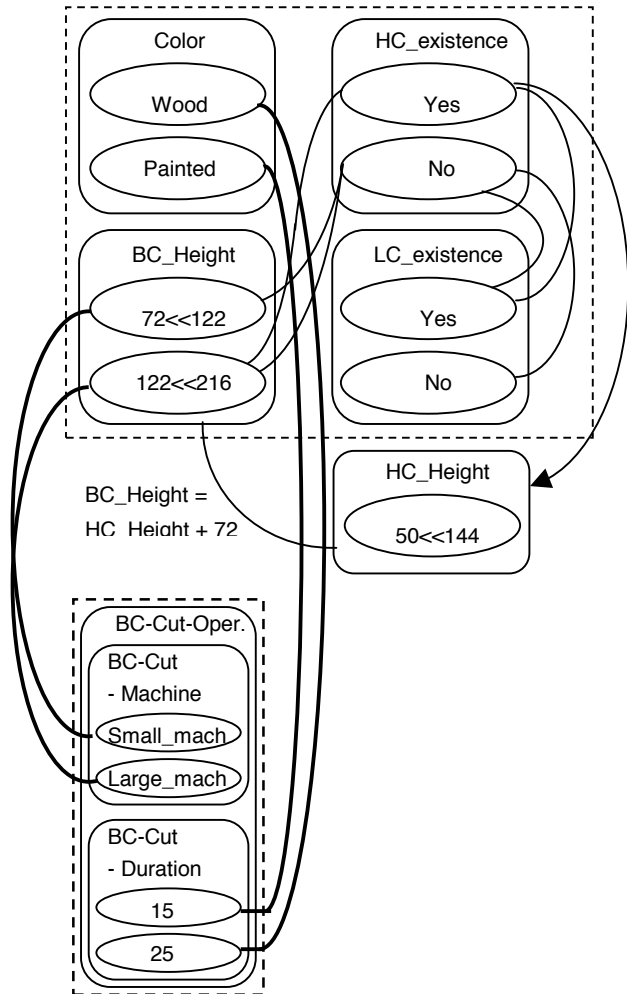


Figure 6 – BC-Cut-Operation model links.

#### HC and LC-Cut-Operation

As represented in figure 7, these two operations exist according to two configuration variables. Therefore two activity constraints modulate the existence of these operations.

As the low cabinet is a standard component, there is no machine choice. The small size of this component allowed to propose an average duration whatever the LC finish is. This operation is in fact a standard operation without customizable possibility.

For the high cabinet, the two machine can be used, and the selection is achieved by the routing designer. As for the BC, the duration of the HC-cutting operation depends of the finish.

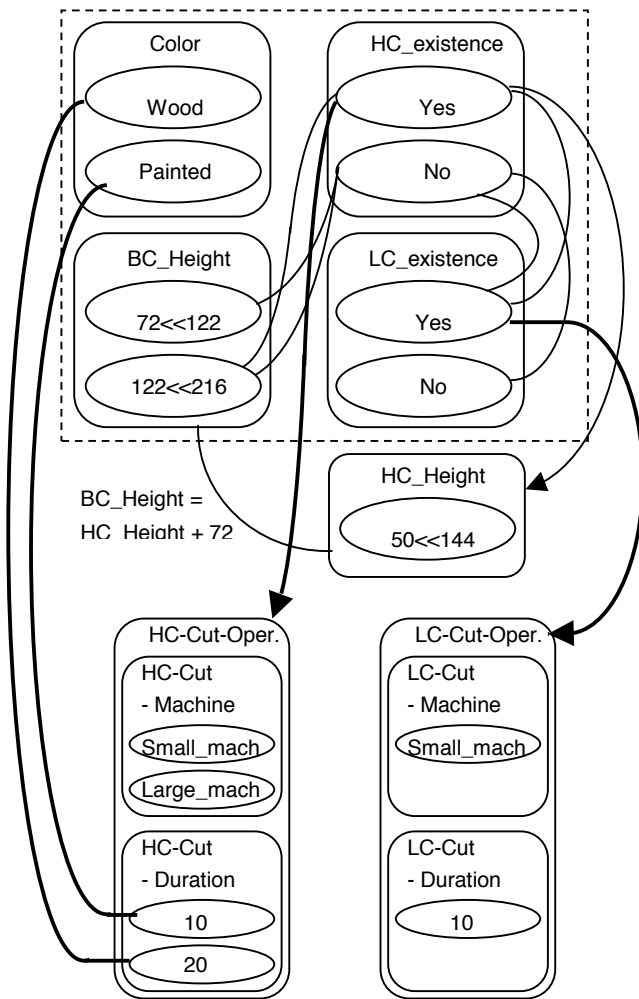


Figure 7 – HC and LC-Cut-Operation model links.

#### Finish and assembly Operation

The links are represented in figure 8.

The finish operation always exists and the facility used depends of the kind of finish selected during product configuration.

The assembly operation exists if at least the low cabinet is present in the configured product. An activity constraint modulates therefore the existence of the assembly operation. If the assembly gathers only BC and LC the small assembly facility is used, if HC is present in the product the large facility must be used. Two compatibility constraints specify these requirements.

#### Conclusion

We have seen in this section how routing generic model and product configuration generic could be link. The next section will discuss configuration achievement of both product and routing configuration.

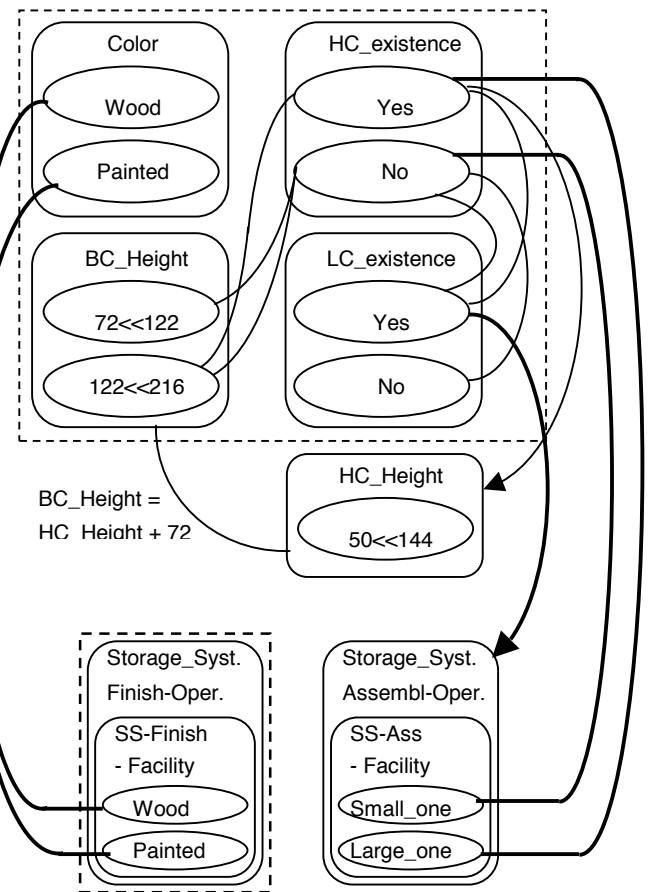


Figure 8 – Finish and assembly Operation model links.

## 4 PRODUCT AND ROUTING CONFIGURATION

### 4.1 Configuring product and routing

We are going to see how the interactive product configuration process presented in section 2.2 and illustrated in figure 3 can be completed with routing configuration.

The section 2.2 product configuration correspond with the following decisions when the models of section 3 are considered :

- Decision of LC existence :  
LC\_existence = Yes
- Selection of the wood finish :  
Color = Wood
- Selection of a bookcase with a small Height, for example :  
BC\_Height = 100  
HC can not exist, and only the “No value” can and must be selected for HC\_Existence  
HC\_existence = No

We are going to analyze each of these product configuration decision and follow the constraint propagation on the routing generic model.

- LC\_existence = Yes  
(i) figure 7 activity constraint (LC\_existence=yes => LC-Cut-Oper. Existence = true)  
implies : **LC-Cut-Oper exists.**

(ii) figure 8 activity constraint (LC\_existence=yes => Storage\_syst.-Assembl-Oper. Existence = true)

implies : **Storage\_syst.-Assembl -Oper exists.**

- Color = Wood

(i) figure 6 compatibility constraint (Color=wood, BC-Cut-Oper-Duration = 25)

implies that : **BC-Cut-Oper-Duration = 25**

(ii) figure 7 compatibility constraint (Color=wood, HC-Cut-Oper-Duration = 20)

implies : **HC-Cut-Oper-Duration = 20**

but this variable is not existing at this time in the configuration.

(iii) figure 8 compatibility constraint (Color=wood, Storage\_syst.-Finish-Oper-Facility = wood)

implies : **SS.-Finish-Oper-Facility = wood**

- BC Height = 100

(i) figure 6 compatibility constraint (BC\_Height ∈ [72,122], BC-Cut-Oper.-machine = small\_mach)

implies : **BC-Cut-Oper.-machine = small\_mach**

- HC existence = No

(i) figure 8 compatibility constraint (HC\_existence = No, SS-Assembl-Oper-Facility = small\_one)

implies : **SS.-Assembl-Oper-Facility = small\_one**

The resulting configured routing is shown in figure 9.

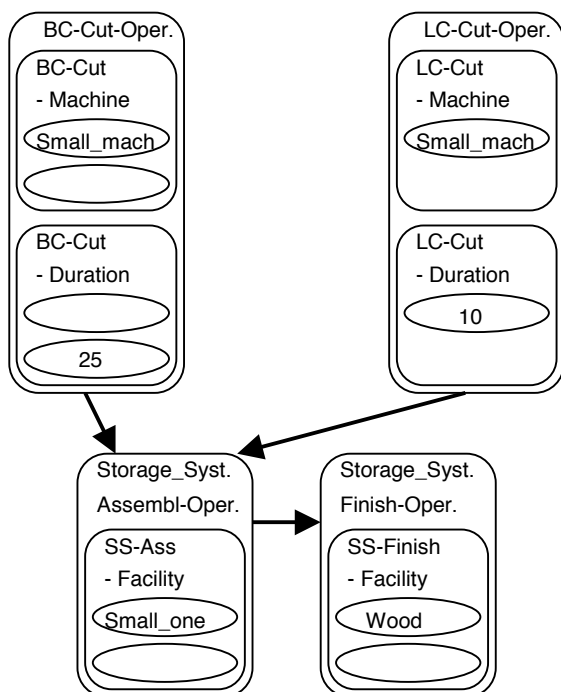


Figure 9 – Configured routing

## 4.2 Discussion

The proposed configuration approach is thus composed of two distinctive process : the product configuration and the routing configuration. These two steps correspond in fact with two generic models. These two generic models are most of the time designed by different persons belonging to different departments of the supplier company, most of the time marketing and design teams for product configuration model and production team for routing models. This decomposition has some interest and some draw-backs.

## Model consistency

It must always be possible for any configured product to configure a routing. Therefore each time a modification is made on the product model, the routing model and links must be checked in order to guarantee consistent result. This is not an easy task and many scientists are working on model diagnosis as felfering [8] or Sabin[9].

## Technical data management

An interest of this two step configuration approach is to permit to separate clearly product and process definition. Each company teams involved in design and manufacturing is responsible of a piece of model and part of the configuration process. Therefore technical data management and some work-flow set up can easily be deployed to guarantee a smooth technical data organization.

## Production data and NC program configuration

We have seen how product routing can be configured with respect to product configuration. The same kind of ideas could be used to configure NC program for machine or to generate information sheets containing various process information. Therefore a third configuration step could be added to product and routing configuration.

## 5 CONCLUSION

Configuration software presents major interests for mass customization deployment.

Generic modeling and constraint propagation allow to guarantee, during and at the end of the configuration process, that the configured object (product or routing) is error free (as far as the generic model is good). This is a very significant interest, because mistakes or errors are the great fear of companies that propose product with customization possibilities.

Constraint propagation allows to see very quickly (almost in real time) the consequences (for both product and routing definition) of each product configuration decision. Therefore, any customer question can be answered very rapidly and the definition of the configured product matching any specific requirement can be done in a very low cycle time. L.Hvam et al reported in [10] that the customized product definition cycle time has reduced from 5/20 days to 1/2 days through the deployment of a configuration.

Error free configured product and quicker configured product definition tend to give confidence in person working in sale/design/manufacturing department of companies offering customization possibilities. This is of importance when you consider that the financial margin gained on a customizable product is in average three or four time larger that the one obtained with a standard product.

These two main improvements (errors and cycle time) permit enormous cost reduction. When, for example, up to 90% of manual customization definitions are not transformed in real customer command, it is clear that huge amount of energy, time and money is lost. If configurator utilization can divide this “energy” by five, cost reduction are really significant.

Product configuration software modules exist now in any major ERP system (Oracle, SAP, Baan...). Some of them have good capabilities for manufacturing data configuration (routing, information sheets, tooling). This shows the interest of major software companies for configuration.

## 6 REFERENCES

- [1] Mittal, S., and Frayman. F., 1989, Towards a generic model of configuration tasks, Int. Joint Conf. Artificial Intelligence, Morgan Kaufmann, Los Altos, California, pp. 1395-1401.
- [2] Kühn, C., 1999, Requirements for Configuring Complex Software-Based Systems, AAAI Workshop on Configuration, Orlando, USA, pp 11-16.
- [3] Aldanondo, M., Moynard, G., Hadj-Hamou, K. and Lamothe J., 2000, Product generic modeling elements for configuration, International conference on Engineering and Production Management (IEPM), Quebec City, Canada, Vol n°1, pp 578-587.
- [4] Mittal, S., and Falkenhainer, B., 1990, Dynamic Constraint Satisfaction Problems, Proc. Ninth National Conf. on Artificial Intelligence AAAI, Boston, USA, pp 25-32.
- [5] Mackworth, A.K., 1992, Consistency in networks of relations, Artificial Intelligence, vol. n°8, pp 99-118.
- [6] Aldanondo, M., Moynard, G. and Hadj-Hamou, K., 2000, General configurator requirements and modeling elements, ECAI Workshop on Configuration, Berlin, Germany, pp 1-6.
- [7] Brown, D.C., 1998, Defining Configuring, AI EDAM (Artificial Intelligence for Engineering Design, Analysis and Manufacturing), vol. 12, n°4, pp 301-306.
- [8] Felfering, A., Friedrich, G., Dietmar, J. and Stumptner, M., 2000, Exploiting structural abstraction for consistency based diagnosis of large configurator knowledge bases, ECAI Workshop on Configuration, Berlin, Germany, pp. 23-28.
- [9] Sabin, M. and Freuder, E.C., 1999, Detecting and resolving inconsistency and redundancy in conditional constraint satisfaction problems, AAAI Workshop on Configuration, Orlando, USA, pp 90-94.
- [10] Hvam, L., Malis, M., Hansen B. and Riis J., Reengineering of the quotations process - application of knowledge based systems, IEPM 2001 Conference– August 2001 Quebec, pp242-251.