How to Use Configuration Software in “Less Routine Design” Situations? Some Modelling Propositions

Abdourahim Sylla[1, 2], Delphine Guillon[1, 3], Luis Garcés Monge[1, 4], Elise Vareilles[1], Michel Aldanondo[1], Thierry Coudert[2], Laurent Geneste[2]
[1]Université de Toulouse, IMT Mines Albi, Centre de Génie Industriel, Albi, France
[2]Université de Toulouse, INP ENIT, Laboratoire Génie de Production, Tarbes, France
[3]Ecole Supérieure des Technologies Industrielles Avancées, ESTIA Recherche, Bidart, France
[4] Instituto Tecnológico de Costa Rica, Cartago, Costa Rica
([1]first-name.family-name@mines-albi.fr, [2]first-name.family-name@enit.fr)

Abstract – **This paper considers the configuration of physical systems in a business to business environment (machine tool, aerospace equipment, cranes …). In this kind of business, knowledge-based configuration software are frequently used when dealing with “infinitely routine design” situations where the entire customer’s requirements can be fulfilled with standard systems. However, in “less routine design” situations where non-standard systems must be designed in order to fulfill the entire customers’ requirements, existing knowledge-based configuration software cannot be used. In fact, the configuration hypothesis state that all configured systems are assembled from standard sub-systems and components. The aim of this paper is therefore to investigate how the existing products/systems configuration hypothesis, problems’ definitions, and models can be modified or adapted in order to allow the use of configuration software in “less routine design” situations. In this purpose, first, the main differences between standard and non-standards systems are analyzed. Then, six cases of systems configuration that differentiate “less routine design” from “infinitely routine design” are identified and discussed. Finally, some Constraint Satisfaction Problems (CSP) based modeling extensions are proposed to allow the use of configuration software in these situations.**

Keywords – **Configuration software, Knowledge-based model, Constraint Satisfaction Problem, Less routine design**

I. INTRODUCTION

The current economic environment is characterized by the ever increasing demand of personalized systems from the client companies. In addition, the requirements on the performances, costs and delivery times of the systems are increasingly constrained. Therefore, in order to propose relevant systems solutions to the client companies, the supplier companies have to design customized systems in a very short period while optimizing time and resources involved in the design process [1],[2],[3].

The design of a system that fulfils the customer’ requirements is carried out using three kind of knowledge: (i) the knowledge about the customer’s requirements that are the source of the design problem, (ii) the knowledge about the potential systems solutions relevant to these requirements, and (iii) the knowledge about the design methodology [4]. Depending on the availability of these three kind of knowledge, three types of design activity have been identified in [4]: routine design, innovative design and creative design. In the context of a routine design, the relevant knowledge necessary for the design of systems solutions that fulfill the customer’s requirements are available. The design of a system in this case, consists in choosing systems solutions or to modify or adapt them slightly [4],[5].

In this paper, we consider the routine design situations. As in [6], in this article we assume that they can be further characterized from “infinitely routine design” to “less routine design”.

The “infinitely routine design” refers to the configuration problem [7]. In this situation, all possible systems solutions that are relevant to the customer’s requirements have been totally designed or predefined. The supplier has just to choose one system solution to propose to the customer. This type of “infinitely routine design” or configuration problem is encountered in many industries, including in the automotive, aeronautics or the micro-informatics sectors [8],[9]. In fact, most of the time, the systems or sub-systems solutions must be selected from a huge number of types or variants to meet specific customer’ requirements [8],[9]. Knowledge-based configuration software is very often used by the suppliers to rapidly configure systems that fulfill the customers’ requirements.

The “less routine design” refers to situations where some modifications or adaptations must be performed on existing systems solutions in order to design systems that fulfil the entire customer’s requirements [6]. For example, a customer wants a crane system composed of two sub-systems: a jib of 7 meters long and a tower of 10 meters high. The existing solutions cover the tower sub-system. However, until now, the supplier has only designed jibs of 5 and 9 meters long. Therefore, a jib of 7 meters long must be designed and integrated to the other sub-systems solutions in order to fulfil the entire customer’s requirements. In these situations, the existing configuration software cannot be used to configure the entire system. Indeed, the configuration makes the assumption that a system is assembled or defined from sub-systems and components that have been totally designed or predefined. The assembly mode of the sub-systems and components is also predefined [10],[8],[9]. As a consequence, some companies use configuration software to design the predefined parts of the system. The other parts are defined manually or using other tools such as Computer Aided Design (CAD) [11],[12],[13]. This results in additional time, resources and efforts in the design process.

The aim of this paper is to investigate how the existing configuration hypothesis, problems' definitions, and models can be modified or adapted in order to extend the use of configuration software towards "less routine design" situations. In the section 2, relevant products/systems configuration background, including problems definitions and Constraint Satisfaction Problems (CSP) knowledge modelling, are recalled. In section 3, the main differences between standard and non-standards systems are analyzed. Then, six cases of systems configuration that differentiate "infinite routine design" from "less routine design" are identified and discussed. Some Constraint Satisfaction Problems (CSP) based modeling extensions that consider the six cases of "less routine design" are also proposed.

II. PRODUCT/SYSTEM CONFIGURATION IN "INFINITE ROUTINE DESIGN" SITUATIONS

A. Configuration problem definition

Since the first configuration problems defined by Mittal [14], many products configuration problems have been defined in the scientific literature [8],[9],[15]. According to the problems, different aspects of a product are considered, especially the physical, descriptive, and functional aspects [8],[9],[15]. Among all these definitions, we consider the key elements proposed in [14],[15]. They are presented as follows:

Assumption: a system is considered as set of sub-systems
Given:

- each system or sub-subsystem is characterized with a predefined set of attributes which have predefined domains,
- the attributes can be either descriptive (length, power for instance) or key performance indicators such as the cost,
- the sub-systems that have the same characteristics constitute a family of sub-systems,
- the possible combinations or assembly of sub-systems and/or attributes values are predefined with a set of constraints,
- a customer's requirements corresponds to the selection of a sub-system or attributes values.

Objectives: The configuration consists in finding at least one set of sub-systems that satisfy all the constraints and customer's requirements.

As you can see in this configuration problem definition, only systems and sub-systems that totally have been designed or predefined are considered. This is the common point between the configuration problems and models encountered in the scientific literature. They all assume the following hypothesis [8],[9],[14],[15],[16],[17]: (i) a configured product or system is assembled from predefined sub-systems or components, and (ii) the assembly mode is also predefined. As a consequence, these definitions and models are not suitable to the "less routine design" situations where some sub-systems are not totally designed. In this article, in section 3.1, we propose some adaptations of this definition to the "less routine design" situations. In the part B, we introduce the CSP-based modelling framework that is used to model systems configuration knowledge. We also present an example of system configuration in "infinitely routine design" situations.

B. CSP based knowledge modelling

In the scientific literature, the CSP (Constraint Satisfaction Problem) is the most commonly formalism used to formalize configuration knowledge. It gathers three elements : (i) a set variables, (ii) a finite domain for each variable, and (iii) a set of constraint that establishes relationships between variables [18]. Referring to the configuration problem previously defined, a CSP-based configuration model is defined as follow [7],[13],[15]: each sub-system family and each attribute is associated to a variable. A specific sub-system or attribute value is then a value in its corresponding variable domain. The constraints are used either to specify acceptable combinations of sub-system solutions and/or attribute values. For example, in the Fig. 1, the sub-system jib is associated to the variable "Jib solution". Its descriptive attributes are associated to the variables "Length" and "Stiffness". The length of the jib has two possible values "5 meters" and "9 meters" which represents its domain. The constraints are represented with the full line. They link the attributes' values to their corresponding sub-systems' solutions. Using this model in a configuration software, if the customer' requirements correspond to these solutions; the supplier can configure rapidly at least one solution that cover all the requirements. However, if the customer's requirements exceed these solutions, the supplier cannot exploit this model in a configuration software to configure a crane system solution that covers all the requirements, even if the supplier is able to design, produce or assemble and deliver that solution.

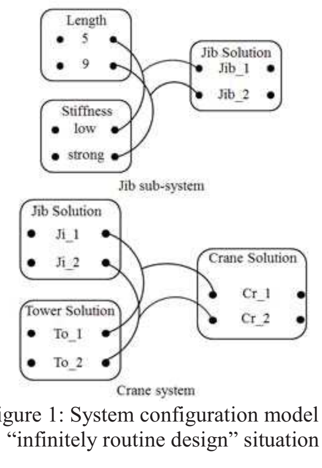

Figure 1: System configuration model in "infinitely routine design" situation

In the next section, we propose some modifications or adaptations to the existing configuration problems' definitions and models in order to allow the use of configuration software in "less routine design" situations.

III. PROPOSITIONS

In this section, we propose some elements that allow to extend the use of configuration software from "infinitely routine" design towards "less routine design" situations. For this purpose, like Myrodia et al. [12], Aldanondo et al. [15] and other authors, we distinguish : the sub-systems, integrations and systems that have been totally designed or predefined as standard elements, and those that have not been totally designed or predefined as non-standard ones. In the part A of this section, we analysis the main differences between standard and non-standard systems that allow to identify six cases that differentiate the configuration of systems in "less routine design" from "infinitely routine design" situations. In the part B, using a simple example, we show how a configuration model relevant to "infinitely routine design" can be adapted and extended towards "less routine design" situations.

A. Differences between "infinitely" and "less" routine design

In this part, an analysis of the characteristics of standards and non-standards systems has allowed us to identify the main characteristics which permit to distinguish them. These characteristics rely on: the descriptive attributes of the sub-systems and systems, and the sub-systems that compose the systems. These two elements (descriptive attributes and sub-systems) may: (i) be standard or non-standard, (ii) take standard or non-standard values/instances, and (iii) be the object of standard or non-standard associations/integrations. On this basis, we will talk about standard systems configuration (a configuration in a "infinitely routine design" situation) when all elements, all values or instances, and all associations or integrations are standard. In contrast, for any other case, we will talk about non-standard systems configuration. Thus, the presence of a non-standard feature implies a case of non-standard systems configuration (a configuration in a "less routine design" situation). This analysis has leaded us to identify six cases that differentiate the configuration in "less routine design" from "infinitely routine design". They represent the different cases of systems configuration in "less routine design" situation. Three cases concern the sub-systems and three relate to the systems. They are presented in Fig. 2 and are described in the following.

The three cases at the sub-system level are:

Case 1: Non-standard association of standard values for the descriptive attributes. This happens when two or more descriptive attributes values that have never been associated together to configure a sub-system have to be associated in order to fulfil customer' requirements. For example, in the Fig. 1, a jib with "5 meters" long and "strong" stiffness is required by a customer.

Case 2: Addition of non-standard values for a descriptive attribute. This happens when a non-standard value must be considered for a descriptive attribute in order to fulfill customer's requirements. For example, a customer wants a jib with "11 meters" long.

Case 3: Addition of non-standard attribute for a sub-system. In this case, a non-standard attribute must be added to configure a sub-system that fulfills customer's requirements. For example, a customer asks for a jib with a specific "shape".

The three cases at the system level are:

Case 4: Non-standard integration of standard instances or solutions for the sub-systems. This happens when two or more sub-systems solutions that have never been integrated together to configure a system, must be integrated to fulfil customer' requirements. For example, the jib "ji_1" and the tower "To_2" must be integrated to fulfil a customer's requirements in the Fig. 1.

Case 5: Addition of a non-standard instance or solution for a sub-system. This happens when a non-standard sub-system solution must be considered for a sub-system in order to fulfill customer's requirements. For example, a customer wants a tower different from "To_1" and "To_2".

Case 6: Addition of non-standard sub-system to a system. In this case, a non-standard sub-system that has never been considered in a system must be added to configure a system that fulfills the customer's requirements. For example, a customer wants a control cabin.

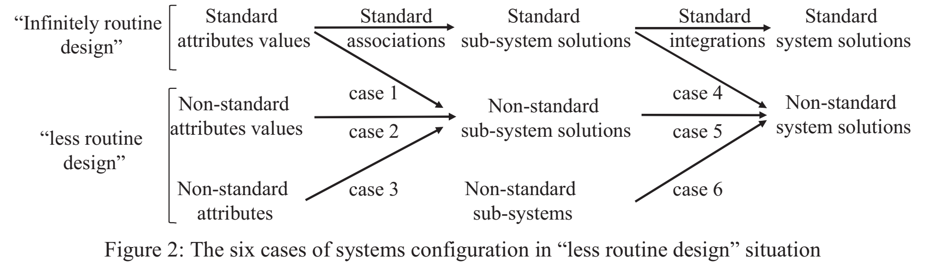

Figure 2: The six cases of systems configuration in "less routine design" situation

In each of these six cases, all the standard solutions that constitute the diversity of systems (options and variants), formalized in a generic model, do not cover the entire customer's requirements. Non-standard systems must be configured. However, as the knowledge related to these non-standard systems is not formalized in a generic model, they cannot be exploited in a configuration software to configure a non-standard system relevant to the customer's requirements.

Therefore, in order to allow the construction of generic models that gather knowledge related to both standard and non-standard systems, a definition of standard and non-standard system configuration problem is proposed in the following. It includes standard and non-standard elements.

Assumption: a system is considered as set of sub-systems;

Given:

- each system or sub-subsystem is characterized with a standard or non-standard set of attributes which have standard or non-standards values in their domains,
- the attributes can be either descriptive (length, power for instance) or key performance indicators such as the cost,
- the sub-systems that have the same characteristics constitute a family, they can be standard or non-standard,
- the possible combinations or assemblies of sub-systems and/or attributes values are defined with a set of constraints, the combinations can be standard or non-standard,
- a customer's requirements corresponds to the selection of a sub-system or attributes values.

Objectives: The configuration consists in finding at least one set of standard and/or non-standard sub-systems that satisfy all the constraints and customer's requirements.

Based on this definition, in the part B of this section, we propose some modelling approaches that enable to extend existing configuration models relevant to "infinitely routine" design towards "less routine design" situations.

B. CSP-based Modelling approaches for systems configuration in "less routine design" situations

For each of the six cases of configuration of systems in "less routine design" listed in the part A of this section, we have proposed some modifications on the existing configuration models in order to include knowledge related to non-standard elements in the generic models. These modifications include changes to the variables and their domain (the set of possible values), as well as changes to the constraints that bind them. In this article, we only present the extension for the case 1 at the sub-system level and the case 5 at the system level. The same example used for the configuration of system in "infinitely routine design" is used. The model is presented in the Fig. 3. This model is a very simple one. The aim is to show how a configuration model relevant to "infinitely routine design" situation can be modified and extended towards "less routine design".

At the upper level of the Fig. 3, the sub-system model (case 1) is presented. The same variables as for the configuration model in "infinitely routine design" situation are kept. The main differences are:

- a non-standard sub-system instance or solution "Ji_NS" is added to the domain of the "Jib Solution", this enable the supplier to know that this solution has not been totally designed yet.
- a constraint is added for the non-standard association of standard values; it links the values "5 meters" of the attribute "length", the value "strong" of the attribute "stiffness" and the non-standard solution "Ji_NS";

At the lower level of the Fig. 3, the system model (case 5) is presented. The same variables as for the configuration model in "infinitely routine design" situation are also kept. The main differences are:

- a non-standard sub-system instance or solution "Ji_NS" is added to the domain of the "Jib Solution", it results from the modification made at the sub-system level;
- a non-standard system instance or solution "Cr_NS" is added to the domain of the "Crane Solution"; as for the sub-system, it enables the supplier to know that this system has not been totally designed yet;
- two constraints are added for the non-standard integrations of : "Ji_NS" and "To_1", and "Ji_NS" and "To_2".

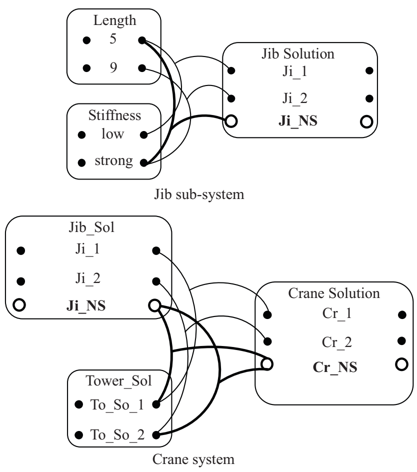

Figure 3: Systems configuration model in "less routine design" situation

IV. CONCLUSION AND FUTURE RESEARCH

In this article, we have studied the configuration of physical systems in the context of business to business environment where a supplier has to propose a system solution to a client company in a very short period while optimizing time, resources and efforts involved. The aim of the article was to propose some solutions in order to extend the use of configuration software from “infinitely routine design” towards “less routine design” situations.

For this purpose, first, we have shown why the existing configuration hypothesis, problems’ definitions and models are not adapted for systems configuration in “less routine design” situations. Then, we have analyzed the main differences between standard and non-standard systems. This has allowed us to identify six cases of systems configuration that differentiate the configuration of systems in “infinitely routine design” from “less routine design” situations. The six cases represent the different situations of systems configuration in “less routine design”. This is the main contribution of this article. As far as we know, no scientific work has proposed a formalization of the differences between systems configuration in “infinitely routine design” and “less routine design” situations. Finally, based on these six cases and the configuration background, we have proposed a definition and some CSP (Constraint Satisfaction Problems) modelling approaches for systems configuration problems in “infinitely routine design” and “less routine design” situations. A simple example is used to illustrate the propositions.

As a future research, we intend to test the applicability of our proposals on a larger case of systems configuration. We also intend to extend the configuration of processes relevant to “infinitely routine design” towards “less routine design”.

ACKNOWLEDGMENT

The authors would like to thank all ANR OPERA partners for their implications in our research work.

REFERENCES

[1] M. Krömker, K. D. Thoben, and a. Wickner, “An infrastructure to support concurrent engineering in bid preparation,” *Comput. Ind.*, vol. 33, no. 97, pp. 201–208, 1997.

[2] A. Sylla, E. Vareilles, M. Aldanondo, T. Coudert, and K. Kirytopoulos, “Customer / Supplier Relationship : reducing Uncertainties in Commercial Offers thanks to Readiness , Risk and Confidence Considerations,” in *Advances on Mechanics, Design Engineering and Manufacturing*, 2017, pp. 1115–1122.

[3] J. p Cannon and C. Homburg, “Buyer-Supplier Relationships and Customer Firm Costs,” *J. Mark.*, vol. 65, no. 1, pp. 29–43, 2018.

[4] B. Chandrasekaran, “Generic Tasks in Knowledge Based Reasoning: High Level Building Blocks for Expert System Design,” *IEEE Expert*, vol. 1(3), pp. 23–30, 1986.

[5] P. Pitiot, M. Aldanondo, E. Vareilles, P. Gaborit, M. Djefel, and S. Carbonnel, “Concurrent product configuration and process planning, towards an approach combining interactivity and optimality,” *Int. J. Prod. Res.*, vol. 7543, no. December 2014, pp. 1–18, 2012.

[6] A. Sylla, E. Vareilles, T. Coudert, M. Aldanondo, and L. Geneste, “Readiness , feasibility and confidence : how to help bidders to better develop and assess their offers,” *Int. J. Prod. Res.*, 2017.

[7] D. Sabin and R. Weigel, “Product configuration frameworks- a survey,” *IEEE Intell. Syst.*, vol. 13, no. 4, pp. 42–49, 1998.

[8] A. Felfernig, L. Hotz, C. Baglay, and J. Tiihonen, *Knowledge-based configuration From Research to Business Cases*. 2014.

[9] L. L. Zhang, “Product configuration : a review of the state-of-the-art and future research,” *Int. J. Prod. Res.*, vol. 52, no. 21, pp. 6381–6398, 2014.

[10] S. Mittal and F. Frayman, “Towards a generic model of configuration tasks,” *Proc. Elev. Int. Jt. Conf. Artif. Intell.*, vol. 2, pp. 1395–1401, 1989.

[11] A. Haug, L. Hvam, and N. H. Mortensen, “Reducing variety in product solution spaces of engineer-to-order companies : The case of Novenco A / S,” *Int. J. Prod. Dev.*, vol. 18, no. 6, pp. 531–547, 2013.

[12] A. Myrodia, K. Kristjansdottir, and L. Hvam, “Impact of product configuration systems on product profitability and costing accuracy,” *Comput. Ind.*, vol. 88, pp. 12–18, 2017.

[13] A. Sylla, E. Vareilles, T. Coudert, M. Aldanondo, L. Geneste, and Y. Beauregard, “ETO Bid Solutions Definition and Selection Using Configuration Models and a Multi-Criteria Approach,” in *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2017, pp. 1833–1837.

[14] S. Mittal and F. Frayman, “Towards a generic model of configuration tasks,” in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1989, vol. 2, pp. 1395–1401.

[15] M. Aldanondo and E. Vareilles, “Configuration for mass customization: How to extend product configuration towards requirements and process configuration,” *J. Intell. Manuf.*, vol. 19, no. 5, pp. 521–535, 2008.

[16] T. Soininen, J. Tiihonen, M. Tomi, R. Sulonen, and T. Männistö, “Towards a general ontology of configuration,” *Aiedam*, vol. 12, no. 4, pp. 357–372, 1998.

[17] A. Günter and C. Kühn, “Knowledge-Based Configuration – Survey and Future Directions –,” in *German Conference on Knowledge-Based Systems*, 1999, pp. 47–66.

[18] U. Montanari, “Network of Constraints: Fundamental Properties and Applications to Picture Processing,” *Inf. Sci.*, vol. 7, pp. 97–132, 1974.