



Configuration knowledge modeling: How to extend configuration from assemble/make to order towards engineer to order for the bidding process

Abdourahim Sylla, Delphine Guillon, Élise Vareilles, Michel Aldanondo,
Thierry Coudert, Laurent Geneste

► To cite this version:

Abdourahim Sylla, Delphine Guillon, Élise Vareilles, Michel Aldanondo, Thierry Coudert, et al.. Configuration knowledge modeling: How to extend configuration from assemble/make to order towards engineer to order for the bidding process. Computers in Industry, 2018, 99, pp.29-41. 10.1016/j.compind.2018.03.019 . hal-01832005

HAL Id: hal-01832005

<https://imt-mines-albi.hal.science/hal-01832005>

Submitted on 3 Oct 2018

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.



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://oatao.univ-toulouse.fr/>
Eprints ID: 19951

To link to this article:

DOI:10.1016/j.compind.2018.03.019

URL:<http://dx.doi.org/10.1016/j.compind.2018.03.019>

To cite this version:

Sylla, Abdourahim and Guillon, Delphine and Vareilles, Elise and Aldanondo, Michel and Coudert, Thierry and Geneste, Laurent
Configuration knowledge modeling: How to extend configuration from assemble/make to order towards engineer to order for the bidding process.
(2018) Computers in Industry, vol. 99. pp. 29-41. ISSN 0166-3615

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Configuration knowledge modeling: How to extend configuration from assemble/make to order towards engineer to order for the bidding process

Abdourahim Sylla^{a,b,*}, Delphine Guillon^{a,c}, Elise Vareilles^a, Michel Aldanondo^a, Thierry Coudert^b, Laurent Geneste^b

^a Université de Toulouse, IMT-Mines Albi, Centre de Génie Industriel, Albi, France

^b Université de Toulouse, INP-ENIT, Laboratoire Génie de Production, Toulouse, France

^c Ecole Supérieure des Technologies Industrielles Avancées, ESTIA Recherche, Bidart, France

ABSTRACT

The bidding process is one of the most important phases for system contractors. A successful bid implies defining and implementing attractive and realistic systems solutions that fulfill customer expectations. An additional challenge arises with the increase in systems diversity resulting from growing customization needs. As a result, for standard customizing offers, bidders find good quality support with configuration software for assemble/make-to-order situations. But when requirements exceed the standard offers, bidders need extended support to fulfill Engineering-to-Order requirements. In this context, this article shows how configuration knowledge models, which support configuration in assemble/make-to-order situations (AMTO), can be extended and used in engineer-to-order situations (ETO). Modeling is achieved assuming that the configuration problem is considered as a constraint satisfaction problem. Six key requirements that differentiate ETO from AMTO are identified and modeling extensions are proposed and discussed. An example illustrates all the contributions.

Keywords:

Bidding process
Assemble/Make-to-Order
Engineering-to-Order
Configuration software
Knowledge-based model
Constraint satisfaction problem

1. Introduction

A call for tenders is a procedure where a customer asks several potential contractors (bidders) to make different commercial offers for the development of a product or system [1,2]. It gives the customer the opportunity to compare several offers and to choose the best one in terms of price, performance and delivery date. After the reception of an invitation to submit, the bidders start their bidding process, which consists mainly of four activities: (i) analysis of the bid opportunity, (ii) elaboration of the bid solution, (iii) drawing up of the commercial bid and (iv) transmission of a bid proposal [3,4]. The contribution presented in this paper adopts the bidder's point of view and aims to support and improve the elaboration of the bid solution using a knowledge-based system.

Like Krömker et al. [5] and Yan et al. [6], we consider that a bid solution is composed of a technical part (bill of material drawing together sub-systems and components) and a delivery process (composed of key required activities and resources). But in this paper, we voluntarily restrict our proposal to the technical part, as it is more sensitive to customer requirements than the delivery process. Indeed, a non-standard requirement, such as a new feature, leading to a non-standard solution, mainly impacts the technical part of the bid. The delivery process is less impacted, as the set of required activities is assumed to be the same whatever the technical solution; we argue that only activity durations and key resources can be different between a standard solution and a non-standard one. The delivery process is thus not considered in this article.

Depending on the system diversity and the customer's requirements, two kinds of industrial situations can be identified when elaborating the technical part or solution for the bid [7–9]. The first situation is Configure-To-Order (CTO) which is intrinsically linked to Assemble-To-Order (ATO) and Make-To-Order (MTO) situations. The second situation is the Engineer-To-Order (ETO) situation. CTO refers to technical solutions (product or system) that have already been studied in detail and that are based

* Corresponding author at: Université de Toulouse, IMT-Mines Albi, Centre de Génie Industriel, Allée des sciences, Albi, 81000, France.

E-mail addresses: abdourahim.sylla@mines-albi.fr, abdourahim.sylla@enit.fr (A. Sylla), delphine.guillon@mines-albi.fr, d.guillon@estia.fr (D. Guillon), elise.vareilles@mines-albi.fr (E. Vareilles), michel.aldanondo@mines-albi.fr (M. Aldanondo), thierry.coudert@enit.fr (T. Coudert), laurent.geneste@enit.fr (L. Geneste).

on some predefined customer requirements. Most of the time in such cases, the technical solutions result from the assembly of standard sub-systems and components (ATO and MTO situations) that have been entirely defined and fully characterized [10]. In this situation, many practitioners use configuration software as a support for the elaboration of the technical part [11]. Configuration software is a knowledge-based system that, given a kind of generic model of the technical solutions, allows the bidder to instantiate or customize a specific solution according to the customer's requirements [12,13,10]. Configuration systems allow reliable and trustworthy technical solutions to be defined. But as a consequence, and even more so in a bidding context, the customer's requirements must be fulfilled only with these standard sub-systems and relevant assemblies as a standard offer.

When customer requirements cannot be fulfilled by this kind of standard offer, novel or adapted technical solutions need to be totally or partially defined, which leads to Engineering-To-Order (ETO) situations [14]. Some companies or software providers speak of "light" ETO, when standard solutions almost completely cover the requirements and just need small adaptations, or "heavy" ETO, when all standard solutions must be adapted and new ones entirely defined. For any ETO situation, the bidders may adopt two different approaches for technical solution elaboration. The first is to perform a fully detailed design, which implies studying in detail all the sub-systems, components and interfaces that may be a part of a technical solution. With this approach, the characteristics of the technical solution to the bid are very accurate and certain, but the resources, time and effort involved are considerable [15,16]. In contrast, the second approach is to perform a pre-design of the technical solutions of the bid. Only the main decompositions, characteristics or working principles for the technical solutions are defined and estimated [17]. The main advantage of this approach is that it reduces the consumption of resources, time and effort when elaborating the technical solutions. However, in contrast to the first approach, the estimation of these technical solutions is more imprecise and uncertain due to the lack of precise and complete knowledge [18,16].

Given the two previous CTO and ETO situations, our concerns are:

- (i) to consider ETO situations with a view to accepting some non-standard customer requirements,
- (ii) to adopt a pre-design approach in order to reduce bidder workload,
- (iii) to support and improve the elaboration of the technical solution with adapted configuration techniques dedicated to CTO.

In this perspective, the goal of this article is to define a generic knowledge-based model for the elaboration of technical solutions for bids in both CTO and ETO situations. Considering as a basis a generic configuration model relevant to CTO, the key contribution of this paper is to show how this generic model can be extended towards ETO situations thanks to various modeling proposals. The modeling extensions rely on identifying the set of non-standard requirements that are acceptable from the bidding company's point of view. Once this identification is made and validated, the generic configuration model is updated with the relevant ETO knowledge. In line with many authors, see Vareilles et al. [19] and Felfernig et al. [10], we consider the configuration problem as a constraint satisfaction problem (CSP) and use this framework for modeling. We need to point out that our proposals are not only dedicated to the bidding process and can be used in other design activities as well. Although this problem is quite common in design activities, the need to take into account non-standard but acceptable requirements is much more critical in the bidding process. Bidding companies have

to answer calls for tenders more and more quickly with solutions that fit the customer's requirements (due date, technical solution, cost, etc.), even if these requirements are near the limits of their technical solution catalogs. A real effort has to be made upstream to determine what technical solutions a company is able to produce to minimize the pre-design time while bidding and maximize the confidence of a company in the execution of its commitments in case of success [9].

In practice, the implications of the contribution of this article are threefold:

- (i) In general, for users of knowledge-based configuration software, it enables them to use, with minor adaptations, conventional configuration software dedicated to CTO (or AMTO) for both AMTO and ETO industrial situations.
- (ii) In the particular case of the bidding process, it enables the contractors (bidders): (a) to consider more bid opportunities and thus increase their business volume, (b) to limit the loss of resources, time and effort, especially in cases where their commercial offers are not accepted by the customers.
- (iii) At the operational level, the person in charge of the offer definition is now allowed some leeway in respect of the standard, whilst remaining within the rigor of configuration software.

The rest of the paper is organized as follows. In Section 2, knowledge-based configuration background, constraint programming and generic modeling issues relevant to CTO are presented. In Section 3, six requirements and relevant extensions of the CTO generic model towards ETO are proposed and illustrated. In Section 4, a synthesis of our proposals is made, leading to discussions and future research. Throughout the paper an example dealing with a crane system configuration problem illustrates the proposals. In the final version, a link to a configuration website will be provided in order to show the propositions' interests.

2. Configuration background and ETO issues

In this section, definition elements relevant to the configuration problem for CTO technical systems are recalled. Secondly, the Constraint Satisfaction Problem (CSP) approach used to formalize it is presented and explained. Finally, the illustrative example is introduced and modeled as a CSP.

2.1. Product and system configuration

One of the first definitions of configuration activity was provided by Mittal and Frayman [20], who described configuration activity as "a special type of design activity, with the key feature that the artifact being designed is assembled from a set of pre-defined components that can only be connected together in certain ways". The authors also consider that a component is characterized by a set of properties and ports for connecting it to other components. This definition has been adopted and discussed by many authors, especially, Soininen et al. [21], Sabin and Weigel [12], Aldanondo and Vareilles [22], Yang and Dong [23] and Felfernig et al. [10].

From previous configuration problem definitions, we consider the following one with the key elements summarized as follows:

- Hypothesis: a product is considered as a set of components
- Given:
 - (1) a generic architecture of the product that describes a family of products,
 - (2) a fixed set of component groups that are always present in any product,
 - (3) a fixed set of component groups that are optional,

- (4) a fixed set of properties that characterize either a component or a product,
- (5) a set of constraints that restrict possible combinations of components and/or property values,
- (6) a set of customer requirements, where a requirement corresponds the selection of a component or a property value.
- Objectives: The configuration of a product consists in finding at least one set of components that satisfies all the constraints and the customer requirements.

An adaptation of this problem, defined for a CTO product in the context of a CTO technical systems configuration problem, can be performed by substituting “product” and “component” with, respectively, “technical system” and “sub-system” in the above definition. With regard to this adapted problem definition, one can develop a CTO technical systems configuration model, which makes it possible to configure technical systems composed of a set of sub-systems, considering one level of decomposition (system/sub-systems). This single abstraction level hypothesis is usually sufficient if we are considering a pre-design context for bidders. However, as already noticed by Mittal and Frayman [20], Aldanondo et al. [24] and Felfernig et al. [10], neither non-standard sub-systems nor non-standard integrations between sub-systems can be identified or selected to configure a non-standard technical system. Configuration problem hypotheses require staying inside the definition of the standard solution set or standard solution catalog.

Therefore, ETO technical system configuration cannot be directly supported by a configuration model in line with previous CTO hypotheses and definitions. Consequently, these configuration definitions and hypotheses, as well as the generic models, have to be adapted to cope with ETO situations. As far as we know, there is no scientific work dealing with this objective. Although many studies have been made in the field of CTO knowledge modeling – see for example: [21,25] or [22] – much fewer concern ETO knowledge. However: [26,14,27,28] or [29] can be consulted. While each of these papers is relevant to knowledge modeling, they concern mainly either CTO knowledge or ETO knowledge and do not attempt to bridge the gap between these two modeling contexts. In this article, we extend the CTO technical systems configuration model towards ETO in order to be able to consider non-standard but acceptable customer requirements. These adaptations are the core contributions given in Section 3. In the following, the Constraint Satisfaction Problem (CSP) framework used to model the configuration problem is recalled and illustrated with the crane example.

2.2. Constraint Satisfaction Problem (CSP)

Within a CSP framework, a problem is defined as a set of constraints (C), which state relations between the problem’s

variables (V), where each variable ($v_i \in V$) can take a value only in a finite domain (D_i) [30]. Referring to the CSP definition above, Sabin and Sabin and Weigel [12], Soininen et al. [21] and others (see [10]) have shown that the product configuration problem defined in Section 2.1 can be modeled with this CSP framework.

We consider the same for systems as follow. Each sub-system group and each property is associated with a variable. A specific sub-system solution of a sub-system group or a specific value of a property corresponds to one value in the domain of the corresponding variable. The constraints represent the allowed combinations of sub-systems and/or property values. In order to be able to model optional sub-systems, Dynamic or conditional CSP (DCSP), introduced by Mittal and Falkenhainer [31], can be used. Within the DCSP framework, the notions of *active/inactive* variables and *active/inactive* constraints are added to the CSP framework. Variables and constraints are partitioned into initially active and initially inactive sets. Referring to the configuration problem of Section 2.1, the sub-systems groups that are always present in a configurable system are associated with the initially active variables whereas the optional sub-systems groups are associated with the initially inactive ones. The activation of inactive variables or constraints is performed using activity constraints that can add to the current problem the variables and constraints that were not initially active. Constraints can also be used to link sub-systems to some indicators, e.g. performance, cost, readiness or confidence, as shown in [32,33]. This assessment issue will not be considered in this article.

In this article, we use the CSP framework to extend the CTO generic model towards ETO. In Section 3, the main issues dealing with CTO and ETO are discussed and the relevant extensions are shown and illustrated on a simple example.

2.3. Illustrative example: a crane system

As an illustrative example, we consider a very simple crane system composed of only two sub-systems that are always present in any configuration (Tower and Jib) and an optional sub-system (Operator basket). This example is inspired by real situations. In the interest of clarity and understanding, we have voluntarily simplified the real-life situations in the rest of the paper.

2.3.1. Crane model

At the sub-system level, as the configuration model structure is similar for all the sub-systems, only the model relevant to the jib is detailed in Fig. 1. The Jib sub-system is characterized by two properties associated with two variables: (i) length of the jib (noted Length, with two possible values “4” or “8” meters) and (ii) stiffness of the jib (noted Stiffness, with two possible values, “low” for low-stiffness or “strong” for strong-stiffness). A sub-system

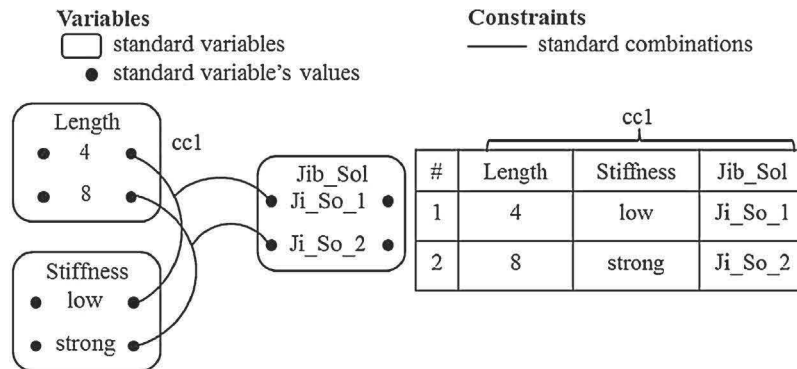


Fig. 1. Generic CTO configuration model of the jib sub-system.

group variable noted Jib_Sol (with two possible values “Ji_So_1” and “Ji_So_2”) identifies jib technical solutions. The constraint cc1 allows two possible combinations of the length and the stiffness of the jib and indicates the corresponding jib sub-system solution. According to cc1, neither an “8 m low stiffness jib” nor a “4 m strong stiffness jib” are allowed by the standard.

At the system level, the integration of the sub-systems has to be taken into account, as shown in Fig. 2.

The configuration model shows five variables:

- (1) three variables associated with the three sub-systems:
The two sub-systems that are always present in any solution:
Jib_Sol, with two possible solutions “Ji_So_1” and “Ji_So_2”, corresponding to the sub-system group for the jib,
Tower_Sol, with two possible solutions “To_So_1” and “To_So_2”, corresponding to the sub-system group for the Tower.
The subsystem that is optional:
Basket_Sol, with two possible basket solutions, small “Ba_So_1” and large “Ba_So_2”, corresponding to the sub-system group for the Basket.
- (2) one variable associated with the system property that characterizes if the optional sub-system is present:
Basket_Exist, with two possible solutions “yes” and “no”,
- (3) one variable associated with the system:
Crane_Sol, with three possible values “Cr_So_1” to “Cr_So_3”, corresponding to the final Crane system,

Three constraints, two of compatibility and one of activation, link these sub-system groups, property and system, together in order to describe the family of systems proposed in the standard:

- (1) the compatibility constraint cs1 describes the possible combination of a jib and a tower with possible relevant crane solutions:
“Ji_So_1” and “To_So_1” are compatible with “Cr_So_1”, while “Ji_So_2” and “To_So_2” are compatible with “Cr_So_2” and “Cr_So_3” (which only differ by the presence of the sub-system basket).
- (2) The compatibility constraint cs2 links the variable Crane_sol with the property Basket_exist. Cs2 allows selection of the appropriate crane technical solution.
“Cr_So_1” and “Cr_So_2” are not compatible with a basket, while “Cr_So_3” is.
- (3) The activation constraint as1 enables control of the existence of the optional sub-system group basket, Basket_Sol, according to the property value of Basket_Exist.

2.3.2. CTO illustrative scenario

As an illustrative scenario of the configuration, let us consider the following customer requirements: “We need a heavy capacity

jib-crane with a large operator basket.” This requirement is translated into the following inputs “A crane with a strong stiffness jib and a large operator basket”:

Input 1: stiffness of the jib equals strong is chosen
⇒ Stiffness = “strong”

As the Stiffness variable has been reduced, the constraints cc1 and cs1 are filtered. Cc1 implies that (1) the length equals 8 (length=“8”) and (2) that the jib solution equals Jib_Sol_2 (Jib_Sol=“Ji_So_2”). Cs1 implies that the tower solution equals To_So_2 (Tower_Sol=“To_So_2”) and therefore that the crane solution is either Cr_So_2 or Cr_So_3 (Crane_Sol=“Cr_So_2” or Cr_So_3”).

Input 2: the value yes for the optional sub-system basket is chosen
⇒ Basket_Exist = “yes”

As this variable has been valued, the activity constraint as1 activates the sub-system group relative to the sub-system basket (Basket_sol) and the constraint cs2 is filtered. Cs2 implies that the crane solution equals Cr_So_3 (Crane_Sol=“Cr_So_3”).

Input 3: the value Ba_So_2 for the optional basket is chosen as it corresponds to a large one
⇒ Basket_Sol = “Ba_So_2”

Once this variable has been valued, nothing else can happen and the configuration process is over. As a result the CTO technical solution proposed to the customer corresponds to Cr_So_3 and gathers the three sub-systems: Ji_So_2, To_So_2 and Ba_So_2. For sake of clarity, the optional standard sub-system (Operator basket) is omitted in the rest of the paper.

2.4. Synthesis

In this section, Configuration-To-Order basics have been recalled and the constraint modeling framework used to model the problem has been presented and illustrated with an example. Finally, some issues that differentiate Engineering-To-Order requirements from configure-to-order requirements have been introduced. The next section details these differences and proposes modeling extensions.

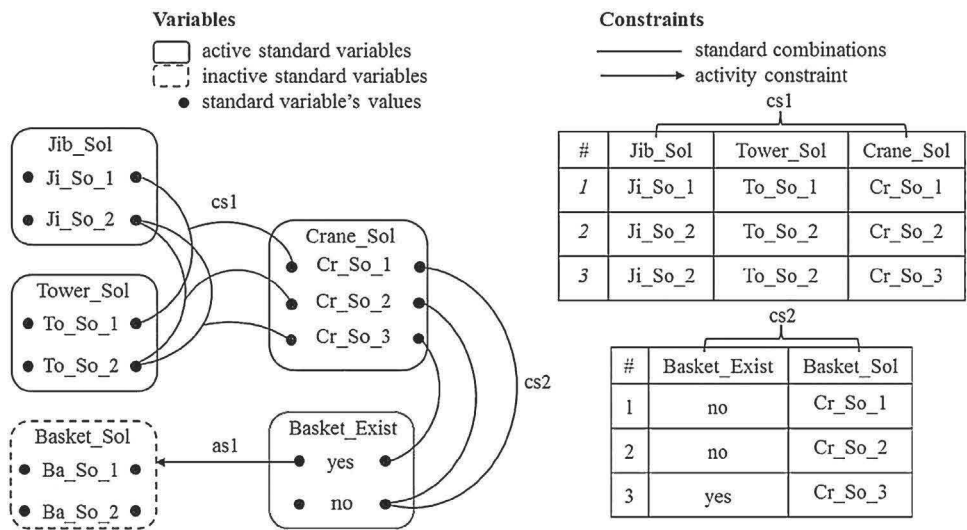


Fig. 2. Generic CTO configuration model of the crane system.

3. Proposals: ETO configuration problem and model updates

In this section, we first present and discuss the main issues dealing with CTO and ETO assembling. We propose and illustrate six constraint-based modeling extensions in order to handle ETO situations with configuration techniques.

3.1. Main issues in CTO and ETO gathering

In the following sub-sections, we first identify, define and illustrate six key required extensions to bridge CTO and ETO. We then discuss them and deduce some consequences dealing with configuration knowledge management for ETO situations. We finally propose some definition elements for CTO-ETO configuration.

3.1.1. CTO – ETO differences: six key requirements

The study of various Engineering-To-Order (ETO) industrial situations and many discussions with configuration practitioners of the configuration community have led us to identify and propose six cases that bridge the gap between CTO and ETO generic modeling. Each of these cases corresponds to real situations that any bidding company may face. For each of them, a specific kind of customer requirement cannot be fulfilled by standard sub-systems and CTO system modeling. Two types of cases are considered: those that are relevant to sub-system properties (cases from 1 to 3) and those that are relevant to sub-systems themselves (cases from 4 to 6). We want to warn the reader that these two types of cases have common roots which induce similarities. We could have proposed more abstract requirement definitions with more abstract modeling extensions. But this would have required abstract definitions and extensions plus instantiations for both property and sub-system levels. We choose to promote easy understanding, even if it may be a little longer, rather than genericity that could be much harder to follow. We therefore keep the six requirements that are shown in Fig. 3. All of them are shown in Fig. 3. In Fig. 3 that depicts the six cases, the highest horizontal flow shows CTO configuration that respects all the standard definitions from left to right: (i) standard property values, (ii) standard property value combinations (arrow that identifies sub-system solution), (iii) standard sub-system solutions, (iv) standard sub-system solution integrations (arrow that identifies system solution) and (v) standard system solutions. Lowest level flows and associated arrows situate the six ETO non-standard cases that require extensions. These six cases are defined as follows:

- **Case 1: non-standard combination of standard property values that leads to a non-standard sub-system solution.** In this first case, in order to fulfill the customer's requirements, the values of at least two standard properties, which were previously

incompatible, have to be chosen simultaneously. For instance, in the crane example, if we consider the sub-system jib presented in Fig. 1, such a requirement could be "A 4 meters strong stiffness jib is required", which is not possible with the current model.

- **Case 2: non-standard property values that lead to a non-standard sub-system solution.** In this second case, the customer requires a property value which is outside the standard values. For instance, in the crane example, if we consider the sub-system jib presented in Fig. 1, such a requirement could be "A 6 m low stiffness jib is required.", while the value "6" is outside the standard length property values and not present in the current model.
- **Case 3: non-standard property that leads to a non-standard sub-system solution.** In this third case, the customer requires a new property which does not belong to the standard. For instance, in the crane example, if we again consider the sub-system jib presented in Fig. 1, such a requirement could be "A 4 m low stiffness jib with a U-shape section is required". The new and non-standard property "section shape", undefined in the current model, needs to be added to the configured system with its value "U-shape".
- **Case 4: non-standard integration of standard sub-system solutions that leads to a non-standard system solution.** In this fourth case, the customer's requirements lead to the need to integrate standard sub-system solutions which have never yet been integrated together. For instance, in the crane example, if we consider the sub-systems Jib and Tower as presented in Fig. 2, such a requirement could correspond to the need to integrate a Jib "Ji_So_1" and a tower "To_So_2", which is not possible with the current model.
- **Case 5: non-standard integration of standard and non-standard sub-system solutions that leads to a non-standard system solution.** In this fifth case, very close to case 4, the customer's requirements lead to the need to integrate standard sub-system solutions with non-standard ones (resulting from cases 1, 2, 3). For instance, in the crane example, if we consider the sub-systems Jib and Tower as presented in Fig. 2, such a requirement could lead to the need to integrate a non-standard jib, corresponding to the following requirement: "A 4 m strong stiffness jib is required.", and a standard tower "To_So_2", which is not possible with the current model.
- **Case 6: non-standard sub-system that leads to a non-standard system solution.** In this sixth case, the customer's requirements lead to the need for a new sub-system which does not belong to the standard sub-systems catalog. Consequently, it must be designed or bought and then integrated. For instance, in the crane example, such a requirement could be "A rotation stop control sub-system that allows maximum angle+/- 10° is required", with the new non-standard sub-system "rotation stop" undefined in the current model.

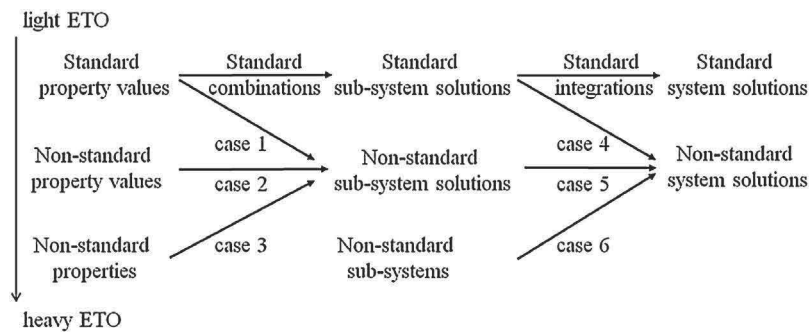


Fig. 3. Summary of the six ETO cases.

3.1.2. Discussion of requirements and knowledge modeling consequences

For each of these cases, the bidding company has never before carried out the engineering work that would enable him to propose a non-standard solution to the customer. This does not mean that such sub-systems could not be developed, produced and integrated with these characteristics. It simply means that at the time when the standard solution set or catalog was defined, these combinations of properties or sub-systems were not studied. Most of the time, this is because it was thought that the standard catalog represents the vast majority of the customers' requirements. As customers need and want increasingly personalized products, many companies more and more frequently have to propose non-standard or out-of-range system solutions in order to fulfill non-standard requirements. Of course, a bidding company cannot simply accept any requirement, so each one has to decide which non-standard requirement is acceptable. This induces a three-level requirement characterization scale: standard, non-standard acceptable and non-standard non-acceptable.

In the introduction, we spoke of "light" versus "heavy" ETO. With the proposed cases, we can slightly refine "heavy ETO" characterization as: (i) Case 3 "ETO heavier" than Case 2 "ETO heavier" than Case 1 and (ii) Case 6 "ETO heavier" than Case 5 "ETO heavier" than Case 4. Furthermore, the six previous cases can of course be combined in many ways, enforcing the "heavy" ETO characterization.

As this problem is increasingly encountered in the bidding process and refers to ETO situations, our goal is to extend CTO constraint configuration models towards ETO in order to be able to consider the six previous cases as non-standard but acceptable customer's requirements.

Before modeling proposals, an important and critical issue dealing with knowledge modeling and coding in configuration for CTO and ETO situations must be discussed and clarified.

In Configure-To-Order situations, the product knowledge necessary for the configuration software set-up is defined by an expert team that usually includes experienced people from the sales, manufacture, and design departments. Generally speaking, these expert teams talk and reach a compromise about a standard offer defining what can be designed, manufactured and put on the market. Then the configuration software is set-up according to this standard offer and the bidder must respect this standard.

In Engineer-To-Order situations, a standard offer is always present, but these expert teams must also decide on the non-standard but acceptable requirements. This means, with respect to previous cases, deciding which of the following non-standard aspects can be accepted: (i) combination of standard property values, (ii) property values (iii) property, (iv) integration of standard sub-system solutions, (v) integration of standard and non-standard sub-system solutions and (vi) sub-system. The examples that illustrate each extension will assume that the non-standard but acceptable requirements are those that illustrate the case description of Section 2.4. Furthermore, when such non-standard requirements are considered, we assume that it is the responsibility of the bidding company expert team to consider their feasibility, i.e. their ability to be developed technically and economically [34]. It is important to note that (1) it is not always easy to identify which potentially non-standard requirements may be required by customers or to know how to evaluate their feasibility *a priori* and (2) the evaluation of this feasibility relies on the skills and expertise of the expert team and its own subjective point of view. This means that the knowledge set up in the configuration software will not

contain any constraint that can affect non-standard bidder choice. For example, when configuring a jib sub-system, if a non-standard but acceptable requirement means that a new property (case 3) can be added, the bidder can add whatever (s)he wants as a property. (S)he can add "section shape" but also "color" or whatever other property is deemed necessary. Once the system is configured with all these non-standard specificities, if the customer accepts the offer, all the relevant engineering activities must be achieved in order to define, manufacture and assemble the non-standard system solution. Then if this new system solution provides satisfaction to both customer and bidder, the knowledge relevant to this non-standard solution will be input into the configuration and, as a result, this non-standard description will become a standard one.

3.1.3. Towards a definition of the CTO-ETO Configuration Problem

Definition elements for CTO-ETO system configuration problems, which highlight the notion of standard and non-standard specificities, can now be derived from the product and system definition of Section 2.1 as follow.

- Hypothesis: a technical system is considered as a set of sub-systems
- Given:
 - (1) a generic architecture that represent a family of standard technical systems
 - (2) a fixed set of standard sub-system groups that are always present in any technical system,
 - (3) a fixed set of standard sub-system groups that are optional in technical systems,
 - (4) a fixed set of standard properties that characterizes either a standard system or a standard sub-system,
 - (5) a set of constraints that restrict possible combinations of standard sub-systems and/or standard property values,
 - (6) a set of customer requirements, where a requirement corresponds to a selection of a sub-system or a property value.
- Given additional features that will lead to the ETO system solutions:
 - non-standard combination of standard property values,
 - non-standard property values,
 - non-standard properties,
 - non-standard integration of standard sub-system solutions,
 - non-standard integration of standard and non-standard sub-system solutions,
 - non-standard sub-system.
- Objectives: The configuration of a technical system consists in finding at least one set of sub-systems (standard or non-standard) that satisfies all the constraints and the customer's requirements.

3.2. Extension of CTO configuration model towards ETO

In this section, we present constraint-based modeling extensions that fulfill the previous definition of a CTO-ETO configuration problem. Each of the six previously identified ETO cases is considered separately. For each one, first the extension of the knowledge-based model is described; secondly, the impacts on the constraint-based model are presented and illustrated using the crane example. A first sub-section deals with the requirements relevant to the sub-system properties and a second sub-section addresses those relevant to the sub-systems themselves.

3.2.1. Standard and Non-standard Properties leading to Non-standard Sub-systems

Case 1: Non-standard combinations of standard property values.

Model extension.

In order to allow some combinations of standard property values which were not permitted by the standard offer, we need to broaden the model by linking non-standard combinations of standard property values to a sub-system group. In terms of sub-system deductions, these non-standard value combinations are only compatible with a non-standard sub-system solution.

Therefore, the constraint-model extensions consist in adding two elements:

- (i) a non-standard sub-system value in the domain of the variable sub-system,
- (ii) as many non-standard constraint tuples as needed to link the non-standard combinations of standard property values to the new non-standard sub-system value.

Crane example.

The configuration model presented in Fig. 4 below depicts an example of the addition of one non-standard acceptable combination of two standard property values. This configuration model is derived from the generic CTO jib configuration model presented in Fig. 1. In this case, the non-standard acceptable requirement “A 4 m strong stiffness jib is required” needs to be taken into account. In order to do so:

- (i) the non-standard jib sub-system solution noted “**Ji_So_NS_AC**” is added to the domain of the variable Jib_Sol.
- (ii) this new value is linked to the non-standard acceptable combination (4 m, strong) as shown in Fig. 4. See line N°3 of constraint cc1.

The term “**Ji_So_NS_AC**” is used to enable the bidders to know that this sub-system Jib solution is **Non-Standard (NS)** and results from a non-standard **Acceptable Combination** of standard property values (**AC**). Referring to case 1 mentioned in sub-Section 2.4, this modification allows a crane with a 4-m jib with a strong stiffness to now be configured.

Case 2: Non-standard property values.

Model extension.

In order to define sub-system solutions with characteristics that exceed the domain of the standard property values, non-standard property values can be added to the domains of the properties. These non-standard values correspond to “non-standard acceptable” requirements. In terms of sub-system solution deductions, these non-standard values are only compatible with a non-standard sub-system solution.

Therefore, the constraint-model extensions consist in adding three elements:

- (i) non-standard values to the domain of some standard properties,
- (ii) a non-standard sub-system solution,
- (iii) constraint tuples that link the non-standard property values and the non-standard sub-system solution.

Crane example.

The configuration model presented in Fig. 5 below depicts an example of addition of non-standard numerical property values to the domain of a standard property. This configuration model is derived from the generic CTO jib configuration model presented in Fig. 1. The difference is that non-standard property values (any value between “4” and “8” meters) are added to the domain of the length of the jib (see “**J4,8[m]**” in Fig. 5). All other values outside [4,8] are not acceptable requirements. In order to do so:

- (i) the non-standard values “**J4,8[m]**” are added to the domain of the property jib length as non-standard acceptable requirements,
- (ii) the non-standard sub-system solution, “**Ji_So_NS_AV**”, is added in the domain of the sub-system group Jib,
- (iii) The non-standard lengths are linked with new tuples to the two possible values of the jib stiffness (“low” and “strong”) and the corresponding previous non-standard jib sub-system solution is noted “**Ji_So_NS_AV**” (see lines n°3 and n°4 of the constraint cc1 in Fig. 5).

The term “**Ji_So_NS_AV**” is used to enable the bidders to know that this sub-system solution is **Non-Standard (NS)** and results from a non-standard **Acceptable Value** in one of the properties (**AV**). Referring to case 2 mentioned in sub-Section 2.4, by using this modification a crane with a 6-m jib with a low stiffness can now be configured.

Case 3: Non-standard property

Model extension.

In order to define specific sub-system solutions which require a specific description, some non-standard properties can be added to the sub-system configuration model. The addition of a non-standard property corresponds to “non-standard acceptable” requirements. In terms of sub-system solution deductions, a non-standard property can only be compatible with a non-standard sub-system solution.

Therefore, the constraint-model extensions, relevant to each non-standard property, consist in adding six elements:

- (i) one variable to capture the non-standard property name,
- (ii) one variable to capture the non-standard property value,

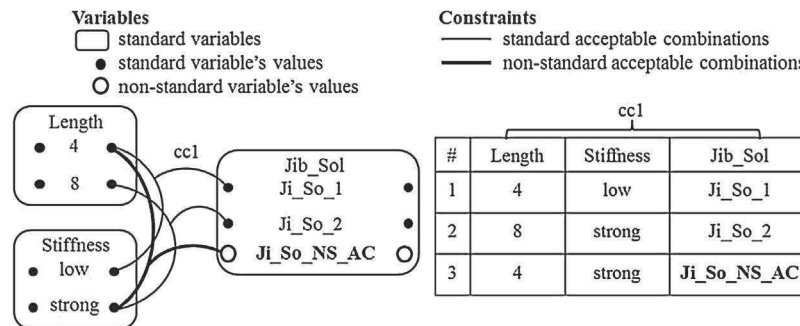


Fig. 4. Non-standard combination of standard property values.

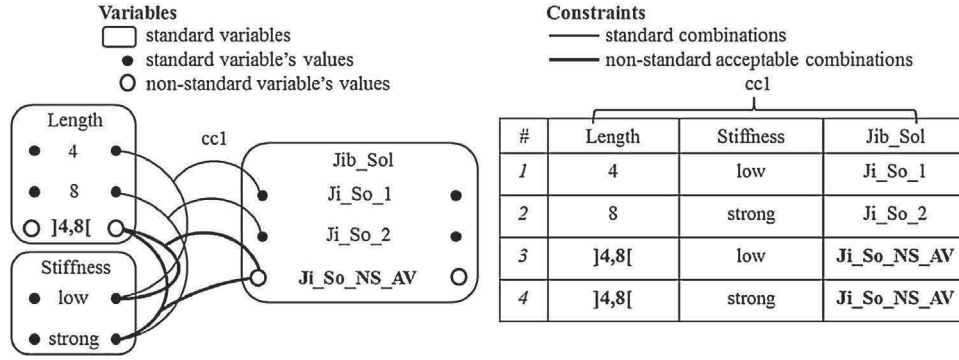


Fig. 5. Non-standard property values.

- (iii) one variable to capture the requirement relevant to the existence (existence flag) of the two previous variables relevant to non-standard property description,
- (iv) one activity constraint that activates the two previous variables,
- (v) one non-standard sub-system solution associated with this non-standard property,
- (vi) one compatibility constraint linking the existence flag and the sub-system solution.

Crane example.

The configuration model presented in Fig. 6 below depicts an example of the addition of a non-standard property. This configuration model is derived from the generic CTO jib configuration model presented in Fig. 1. It shows how a non-standard property can be added to the model to meet a specific customer requirement. In order to do so:

- (i) one variable, noted **NS_prop_name**, is added with a symbolic domain allowing the capture of any kind of character string,
- (ii) one variable, noted **NS_prop_value**, is added with a symbolic domain allowing the capture of any kind of character string,

- (iii) one variable, noted **NS_prop_exist**, is added with a Boolean domain “yes” or “no” allowing the capture of the existence requirement of the non-standard property,
- (iv) one activity constraint **ac1**, {NS_prop_exist = “yes”} => activation of {NS_prop_name, NS_prop_value}
- (v) the non-standard sub-system solution, “**Ji_So_NS_AP**”, is added in the domain of the sub-system group Jib, variable Jib_Sol
- (vi) one compatibility constraint **cc2**, that associates the existence flag and sub-system solution (NS_prop_exist, Jib_Sol) and shows that the positive flag can only be associated with the non-standard sub-system “Ji_So_NS_AP”.

The term “**Ji_So_NS_AP**” is used so that the bidders will know that this sub-system solution is **Non-Standard (NS)** and results from a non-standard **Acceptable Property**

(AP). Referring to case 3 mentioned in Section 2.4, this modification allows a crane with a 4-m jib with a low stiffness and a “**U_shape**” to now be configured: (1) Jib_Length = “4”, (2) Jib_Stiffness = “low”, (3) NS_prop_exist = “yes”, (4) NS_prop_name, = “jib shape”, (5) NS_prop_value, “U shape”, (6) Jib_Sol = “Ji_So_NS_AP”.

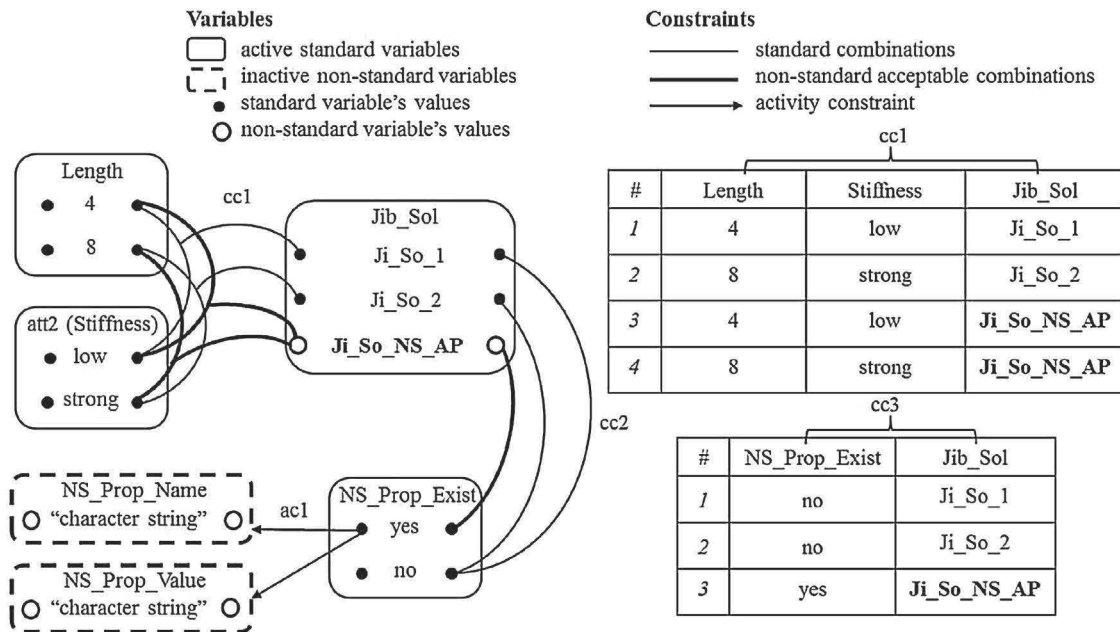


Fig. 6. Non-standard property.

Combining non-standard ETO cases at sub-system level

The three previous cases can be associated. This allows non-standard sub-systems to be defined and configured, which can potentially assemble a standard CTO offer with non-standard: case 1 or case 2 or case 3 or cases 1 and 3, or cases 2 and 3. Two combinations are missing non-standard case 1 and 2 and non-standard cases 1, 2 and 3, because as soon as there is a non-standard property value (case 2), this requires a non-standard combination of property values (case 1). Sub-system solutions relevant to cases 1 and 3 are noted with “**NS_AC_AP**” while those relevant to cases 2 and 3 are noted with “**NS_AV_AP**”. It is very important for the bidders to easily understand why the sub-system is non-standard. Indeed, by knowing the root causes, the bidders can better estimate the time needed to design such a new sub-system and assess the cost indicators.

Crane example.

The configuration model presented in Fig. 7 below shows these five possibilities for non-standard jib sub-system solutions. The initial standard solution Ji_So_1 and Ji_So_2 are merged with five non-standard Jib solutions that are:

- (i) Ji_So_NS_AC” for case 1
- (ii) “Ji_So_NS_AV” for case 2
- (iii) “Ji_So_NS_AP” for case 3
- (iv) “Ji_So_NS_AC_AP” for cases 1 and 3
- (v) “Ji_So_NS_AV_AP” for cases 2 and 3

All modeling extensions of Figs. 4–6 are merged in Fig. 7. The configuration process can start either by selecting the standard values for the standard property or by selecting non-standard ones.

3.2.2. Standard and non-standard sub-systems leading to non-standard technical systems

Case 4: Non-standard integration of standard sub-system solutions.

Model extension.

This case is very similar to case 1, dealing with non-standard combinations of property values. In order to allow the integration of standard sub-systems which were not permitted by the standard offer, we need to broaden the model by linking non-standard integrations of standard sub-system solutions to the technical system. The addition of a non-standard integration

corresponds to a “non-standard acceptable” requirement. In terms of technical system deductions, these non-standard sub-system integrations are only compatible with a non-standard technical solution.

Therefore, the constraint-model extensions consist in adding two elements:

- (i) a non-standard technical system solution value in the domain of the technical system variable,
- (ii) as many non-standard constraint tuples as needed to link the non-standard integrations of standard sub-systems to the new non-standard technical system solution.

Crane example.

The configuration model presented in Fig. 8 below depicts an example of the addition of one non-standard acceptable integration of two standard sub-system solutions. This configuration model is derived from the generic crane configuration model presented in Fig. 2. In this case, the non-standard acceptable technical requirement “We need to integrate To_So_2 with Ji_So_1” needs to be taken into account. In order to do so:

- (i) a non-standard crane system solution noted “**Cr_So_NS_AI**” is added to the domain of the variable Crane_Sol.
- (ii) this new value is linked to the non-standard acceptable integration thanks to the addition of the non-standard constraint tuple (To_So_2, Ji_So_1) of the constraint **cs1** (To_Sol, Jib_Sol, Crane_Sol), as shown in line n°3 of Fig. 8.

The term “**Cr_So_NS_AI**” is used to enable the bidders to know that this technical solution is **Non-Standard (NS)** and results from a non-standard **Acceptable Integration** of standard sub-systems solutions (**AI**). Referring to case 4 mentioned in Section 2.4, this modification means that a crane composed of the sub-systems To_So_2 and Ji_So_1 can now be configured.

Case 5: Non-standard integration of standard and non-standard sub-system solutions.

Model extension.

This case is very similar to case 4, the only difference being that now a non-standard sub-system solution (resulting of cases 1, 2 and/or 3) can also be integrated to provide a non-standard technical system solution. In terms of technical system deduction,

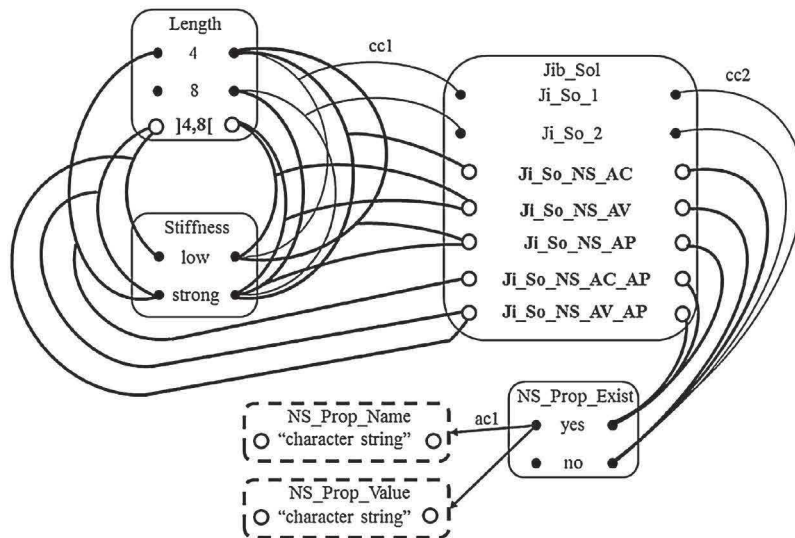


Fig. 7. Combination of the non-standard ETO cases at sub-system level.

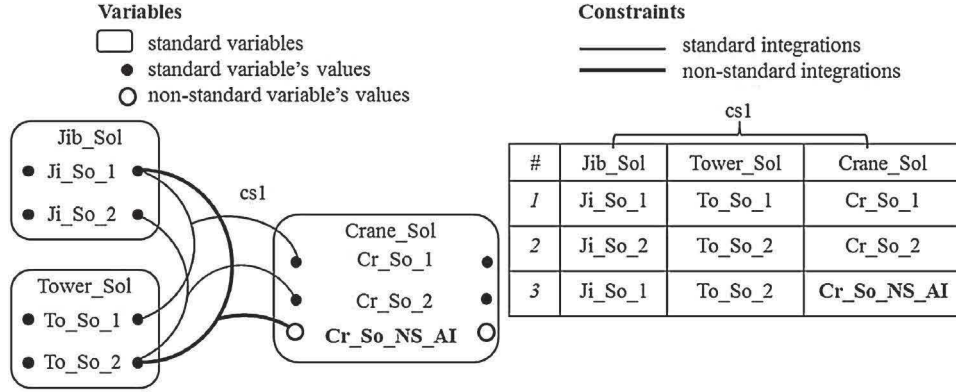


Fig. 8. Non-standard integration of standard sub-system solutions.

as in case 4, these non-standard integrations of standard and non-standard sub-systems are only compatible with a non-standard technical solution.

Therefore, the constraint-model extensions consist in adding two elements:

- a non-standard technical system solution value in the domain of the technical system variable,
- as many non-standard constraint tuples as needed to link the non-standard integrations of standard and non-standard sub-systems to the new non-standard technical system solution.

Crane example.

The configuration model presented in Fig. 9 below depicts an example of the addition of one non-standard acceptable integration of a standard with a non-standard sub-system. This configuration model is derived from the generic CTO crane configuration model presented in Fig. 2. In this case, the non-standard acceptable technical requirement “We need to integrate To_So_2 with a 4 m strong stiffness jib” needs to be taken into account. In order to do so:

- a non-standard crane system solution noted “**Cr_So_NS_NI**” is added to the domain of the variable Crane_Sol.
- this new value is linked to the non-standard acceptable integration by means of the addition of the non-standard constraint tuple (To_So_2, Ji_So_NS_AC) of the constraint **cs1** (To_Sol, Jib_Sol, Crane_Sol) as shown in lines n°4 and 5 of Fig. 9.

The term “**Cr_So_NS_NI**” is used to enable the bidders to know that this technical solution is **Non-Standard (NS)** and results from a **Non-standard acceptable Integration** of standard and non-

standard sub-systems solutions (**NI**). Referring to case 5 mentioned in Section 2.4, this modification means that a crane with a standard sub-system solution To_So_2 and non-standard sub-system solution Ji_So_NS_AC can now be configured.

Case 6: Non-standard sub-system.

Model extension.

This case is very similar to case 3, which describes adding a non-standard property. In order to define specific system solutions that require a specific new function, some non-standard sub-systems can be added to the system configuration model. The addition of a non-standard sub-system corresponds to “non-standard acceptable” requirements. In terms of system-solution deductions, as in cases 4 and 5, a non-standard system can only be compatible with a non-standard technical solution.

Therefore, the constraint-model extensions relevant to each non-standard sub-system consist in adding six elements:

- one variable to capture the non-standard sub-system name,
- one variable to capture the non-standard sub-system description,
- one variable to capture the requirement relevant to the existence (existence flag) of the two previous variables relevant to non-standard system description,
- one activity constraint that activates the two previous variables,
- one non-standard system solution associated with this non-standard sub-system,
- one compatibility constraint linking the existence flag and the system solution.

Crane example.

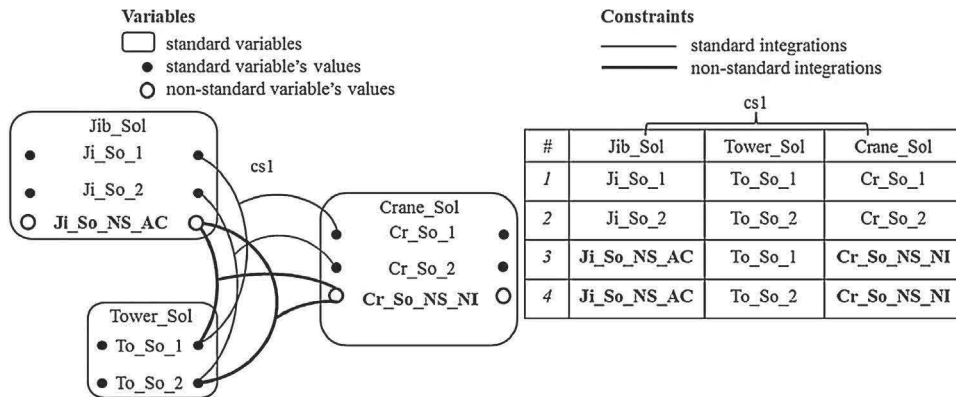


Fig. 9. Non-standard integration of standard and non-standard sub-system solutions.

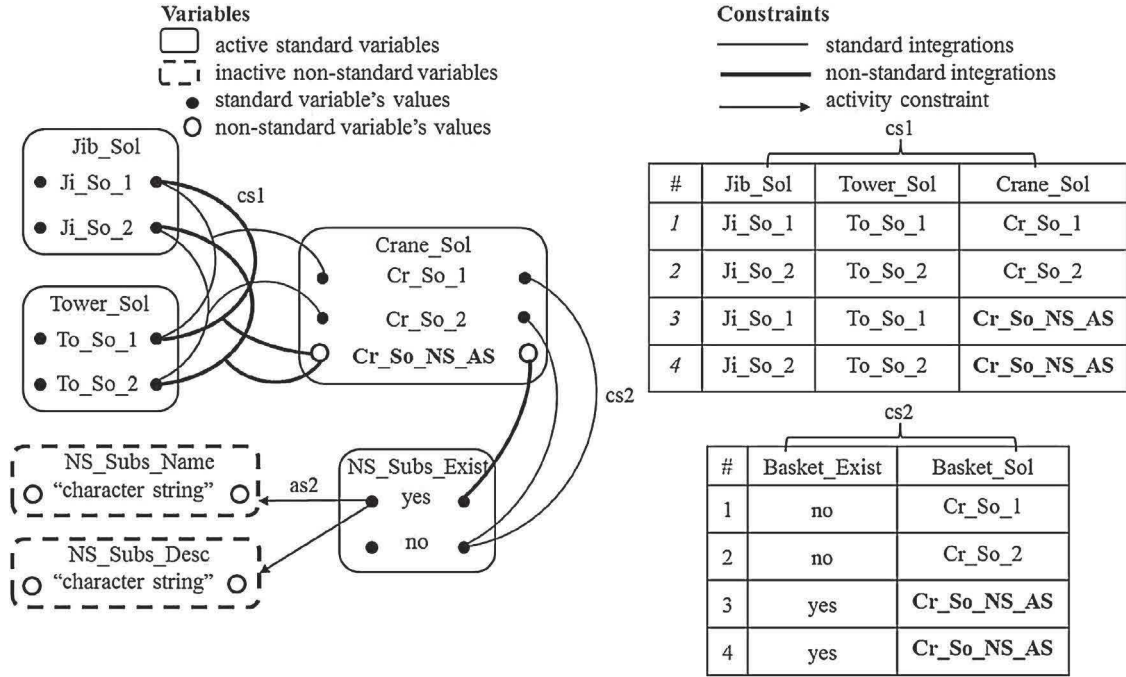


Fig. 10. Non-standard sub-system.

The configuration model presented in Fig. 10 below depicts an example of the addition of a non-standard sub-system. This configuration model is derived from the generic CTO crane configuration model presented in Fig. 2. It shows how one non-standard sub-system can be added to the model to meet a specific customer requirement. In order to do so:

- one variable, noted **NS_subs_name**, is added with a symbolic domain allowing the capture of any kind of character string capturing the type of the sub-system,
- one variable, noted **NS_subs_desc**, is added with a symbolic domain allowing the capture of any kind of character string capturing the description of the sub-system,
- one variable, noted **NS_subs_exist**, is added with a Boolean domain "yes" or "no" allowing capture of the existence requirement of the non-standard sub-system,

- one activity constraint **as2**, {NS_subs_exist = "yes"} => activation of {NS_subs_name, NS_subs_desc}
- the non-standard system solution, "**Cr_So_NS_AS**", is added in the domain of the crane system, variable Crane_Sol
- one compatibility constraint **cs2**, which associates the existence flag and system solution (NS_subs_exist, Crane_sol) and shows that the positive flag can only be associated with the non-standard sub-system "Cr_So_NS_AS".

The term "**Cr_So_NS_AS**" is used to enable the bidders to know that this system solution is **Non-Standard (NS)** and results from a non-standard **Acceptable Sub-system (AS)**. Referring to case 6 mentioned in Section 2.4, this modification allows a crane with a 4-m jib with a low stiffness and a "**rotation stop**" additional sub-system to now be configured: (1) Jib_Length = "4", (2) Jib_Stiffness = "low", (3) NS_subs_exist = "yes", (4)

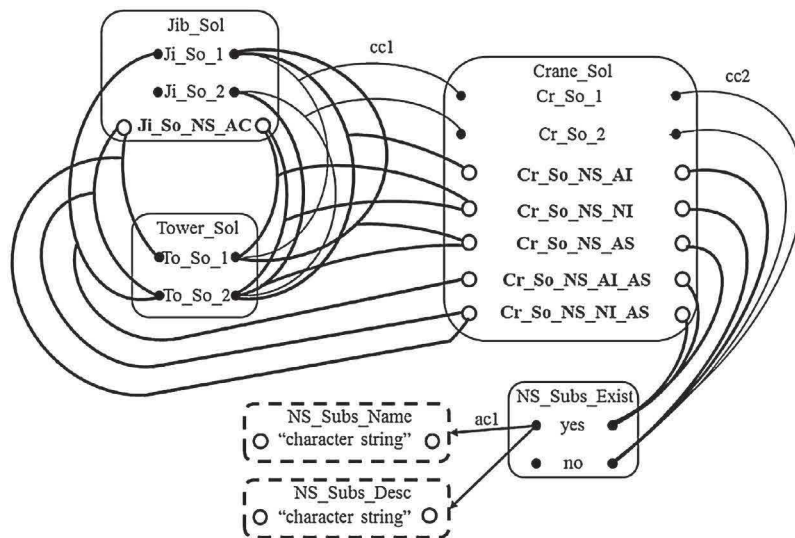


Fig. 11. Combination of the non-standard ETO cases at system level.

NS_subs_name, = "rotation stop", (5) NS_subs_desc, "Max +/- 120°", (6) Crane_sol = "Cr_So_NS_AS".

Combining non-standard ETO cases at system level

As with the sub-system level, the models of the three previous cases can be associated. The resulting model gathers the configuration models and potentially allows a standard CTO offer to be assembled with non-standard: case 4 or case 5 or case 6 or cases 4 and 6 or cases 5 and 6. Two combinations are also missing non-standard case 4 and 5 and non-standard cases 4, 5 and 6, because as soon as there is a non-standard sub-system solution (case 5), this requires a non-standard integration of sub-system solutions (case 4). System solutions relevant to cases 4 and 6 are noted with "NS_AI_AS" while those relevant to case 2 and 3 are noted with "NS_NI_AS".

Crane example.

The configuration model presented in Fig. 11 below shows these five possibilities for non-standard crane system solutions. The initial standard solution Cr_So_1 and Cr_So_2 are merged with five non-standard crane solutions, which are:

- "Cr_So_NS_AI" for case 4
- "Cr_So_NS_NI" for case 5
- "Cr_So_NS_AS" for case 6
- "Cr_So_NS_AI_AS" for cases 4 and 6
- "Cr_So_NS_NI_AS" for cases 5 and 6

All modeling extensions of Figs. 8–10 are merged in Fig. 11.

4. Conclusion

Answering a call for tender has become a real challenge today, with bidders and companies having to cope first with a very tight time period between the date of the call and the closing date for the application, and secondly, with ever more specific customer requirements, which are not completely within their capabilities and need some engineering work. Therefore, companies that nowadays run Configuration-to-Order (CTO) software must deal more and more frequently with out-of-range requirements that lead to non-standard solutions. This pushes them towards Engineer-to-Order (ETO) situations. In this article, we focus on the technical part (bill of materials of systems grouping sub-systems) of the bid solution. As a consequence, we propose a generic knowledge model supporting the elaboration of technical solutions to a bid for both CTO and ETO situations. In order to do so, we have based our proposals on the framework most commonly used by the CTO configuration community, which is the Constraint Satisfaction Problem or CSP framework, and we have extended it from CTO to ETO situations.

We started by recalling configuration problems and constraint approaches and then introduced a very simple crane system to illustrate a configuration problem. This very simple example is used throughout the article to highlight the needs of ETO while bidding, as well as the advantages and the implementation of our proposals.

Secondly, six cases that bridge the gap between CTO and ETO situations have been identified. Each of them results from a specific customer requirement which cannot be directly fulfilled by the company's standard catalog and CTO modeling. Therefore, some engineering work is needed and this leads to a non-standard technical solution. In order to reach this solution, companies first have to identify which specific non-standard requirements are acceptable and which are not. It is only after this step that the knowledge-based model can be enriched with non-standard but acceptable specificities. Consequently, we have extended the configuration definition to an ETO situation by adding the notions of "standard" and "non-standard": (i) combination of property

values, (ii) property values, (iii) property, (iv) integration of sub-system solutions, (v) integration of standard and non-standard sub-system solutions, and (vi) sub-systems. These notions and relevant cases have been situated on a global ETO schema (see Fig. 3) that shows the diversity of all the possible non-standard configuration flows. They have been summarized and grouped as follows:

- Standard and non-standard properties leading to non-standard sub-system solutions:
 - Case 1: Non-standard combinations of standard property values (AC)
 - Case 2: Non-standard property values (AV)
 - Case 3: Non-standard property (AP).
- Standard and non-standard sub-systems leading to non-standard technical system solutions:
 - Case 4: Non-standard integration of standard sub-system solutions (AI)
 - Case 5: Non-standard integration of standard and non-standard sub-system solutions (NI)
 - Case 6: Non-standard sub-system (AS).

For each of these cases, we have shown how the CTO knowledge-based model can be extended or updated with all non-standard specificities. Each extension has been illustrated using the crane example. Furthermore, it has been shown that different modeling extensions could be cumulated in order to take into account different non-standard but acceptable requirements present in the same industrial situation. For each root cause leading to an ETO configuration, we have paid attention to making sure that bidders are aware of the reasons for the ETO configuration by using specific notation proposals in the non-standard solution: (i) AC for acceptable combination, (ii) AV for acceptable value, (iii) AP for acceptable property, (iv) AI for acceptable integration of standard sub-system solutions, (v) NI for acceptable integration of standard and non-standard sub-system solutions (vi) AS for acceptable sub-system.

All these proposals tend to show that, in terms of modeling and in terms of constraint filtering, it appears highly feasible to extend the use of constraint-based configuration software from Configuration-to-Order toward Engineer-to-Order. As stated in Section 2, we did not find any significant previous work that attempted to combine CTO and ETO knowledge modeling issues. We hope that this first proposal will stimulate work in the scientific configuration community. As regards configuration software providers, we have seen some configuration-aiding software that proposes a kind of "escape gate" for addressing some ETO issues, which would allow a conventional CTO standard configuration process, coupled with some kind of open design for non-standard requirement processing. We hope that our proposals will contribute to a better understanding of this kind of "escape gate" and therefore provide a better structure to CTO-ETO configuration systems. The genericity of our proposals allows them to be applied in the bidding process as well as in other design activities. To conclude on the practical benefits of our proposals, we are convinced that they can genuinely improve practitioners' bidding methods: (i) by allowing them to define more bids while reducing the time, effort and resources allocated, and (ii) by giving them controlled freedom compared to a catalog of standard solutions.

In our future research, we now need to extend our proposals to the delivery process by identifying and drawing the links between non-standard systems and their delivery processes. We also have to test the applicability and scalability of our proposals using larger real cases. Particularly with complex systems, it is not always easy to identify which new features will be required and to know how to

evaluate their feasibility *a priori*. On a more conceptual side, we also think that the six cases presented here can be considered as a good basis for proposing a more formal definition of the heaviness/lightness level that characterizes Engineer-to-Order situations.

References

- [1] G. Arslan, M. Tuncan, M.T. Birgonul, I. Dikmen, E-bidding proposal preparation system for construction projects, *Build. Environ.* 41 (10) (2006) 1406–1413.
- [2] R. Sonmez, B. Sözen, A support vector machine method for bid/no bid decision making, *J. Civ. Eng. Manage.* 23 (5) (2017) 641–649.
- [3] R. Chahal, A.R. Ghomari, An Approach for a Bidding Process Knowledge Capitalization, in world academy of science, *Eng. Technol.* 13 (2008).
- [4] A. Leśniak, E. Plebankiewicz, Modeling the decision-making process concerning participation in construction bidding, *J. Manage. Eng.* 231 (2) (2013) 4014032.
- [5] M. Krömker, K.D. Thoben, A. Wickner, An infrastructure to support concurrent engineering in bid preparation, *Comput. Ind.* 33 (97) (1997) 201–208.
- [6] W. Yan, M.C. Pritchard, C.H. Chen, L.P. Khoo, A strategy for integrating product conceptualization and bid preparation, *Int. J. Adv. Manuf. Technol.* 29 (5) (2006) 616–628.
- [7] B. Chandrasekaran, Generic tasks in knowledge based reasoning: high level building blocks for expert system design, *IEEE Expert* (3) (1986) 23–30.
- [8] J. Olhager, Strategic positioning of the order penetration point, *Int. J. Prod. Econ.* 85 (2003) 319–321.
- [9] A. Sylla, E. Vareilles, M. Aldanondo, T. Coudert, K. Kirytopoulos, Customer/supplier relationship: reducing uncertainties in commercial offers thanks to readiness, risk and confidence considerations, *Adv. Mech. Des. Eng. Manuf.* (2017) 1115–1122.
- [10] A. Felfernig, L. Hotz, C. Baglay, J. Tiihonen, Knowledge-Based Configuration From Research to Business Cases, (2014) .
- [11] A. Myrodiia, K. Kristjansdottir, L. Hvam, Impact of product configuration systems on product profitability and costing accuracy, *Comput. Ind.* 88 (2017) 12–18.
- [12] D. Sabin, R. Weigel, Product configuration frameworks- a survey, *IEEE Intell. Syst.* 13 (4) (1998) 42–49.
- [13] P.-A. Yvar, A CSP approach for the network of product lifecycle constraints consistency in a collaborative design context, *Eng. Appl. Artif. Intell.* 22 (6) (2009) 961–970.
- [14] L. Rivest, A. Desrochers, A. Brie, Adaptive generic product structure modelling for design reuse in engineer-to-order products, *Comput. Ind.* 61 (2010) 53–65.
- [15] O. Willner, J. Gosling, P. Schönsleben, Establishing a maturity model for design automation in sales-delivery processes of ETO products, *Comput. Ind.* 82 (2016) 57–68.
- [16] A. Sylla, E. Vareilles, T. Coudert, M. Aldanondo, L. Geneste, Readiness, feasibility and confidence: how to help bidders to better develop and assess their offers, *Int. J. Prod. Res.* 55 (23) (2017) 7204–7222.
- [17] G. Pahl, W. Beitz, *Engineering Design A Systematic Approach*, Springer Science & Business Media, 2013.
- [18] J. Dantan, N. Gayton, A.J. Qureshi, M. Lemaire, A. Etienne, Tolerance analysis approach based on the classification of uncertainty (aleatory/epistemic), 12th CIRP Conference on Computer Aided Tolerancing (10) (2013) 287–293.
- [19] É. Vareilles, M. Aldanondo, A. Codet De Boisse, T. Coudert, P. Gaborit, L. Geneste, How to take into account general and contextual knowledge for interactive aiding design: towards the coupling of CSP and CBR approaches, *Eng. Appl. Artif. Intell.* 25 (1) (2012) 31–47.
- [20] S. Mittal, F. Frayman, Towards a generic model of configuration tasks, *Proc. Elev. Int. Jt. Conf. Artif. Intell.* 2 (1989) 1395–1401.
- [21] T. Soeninen, J. Tiihonen, T. Männistö, R. Sulonen, Towards a general ontology of configuration, *Ai Edam* 12 (4) (1998) 357–372.
- [22] M. Aldanondo, E. Vareilles, Configuration for mass customization: how to extend product configuration towards requirements and process configuration, *J. Intell. Manuf.* 19 (5) (2008) 521–535.
- [23] D. Yang, M. Dong, Applying constraint satisfaction approach to solve product configuration problems with cardinality-based configuration rules, *J. Intell. Manuf.* (2013) 99–111.
- [24] M. Aldanondo, K. Hadj-hamou, G. Moynard, J. Lamothe, Mass customization and configuration: requirement analysis and constraint based modeling propositions, *Integr. Comput. Aided Eng.* 10 (2003) 177–189.
- [25] A. Felfernig, G. Friedrich, D. Jannach, Conceptual modeling for configuration of mass-customizable products, *Artif. Intell. Eng.* 15 (2001) 165–176.
- [26] Z. Siddique, J.A. Ninan, Modeling of modularity and scaling for integration of customer in design of engineer-to-order products, *Integr. Comput. Aided Eng.* 13 (2) (2006) 133–148.
- [27] F. Elgh, Modelling and management of product knowledge in an engineer-To-Order, *International Conference on Engineering Design, ICED11* (August) (2011).
- [28] Z. Miaofen, S. Shaohui, G. Youping, C. Guojin, Research on the configuration design of the ETO product based on knowledge and thing characteristic table, *International Conference on Applied Informatics and Communication ICAIC* (2011) 273–281.
- [29] Y. Kristianto, P. Helo, R. Jianxin, A system level product configurator for engineer-to-order supply chains, *Comput. Ind.* 72 (2015) 82–91.
- [30] U. Montanari, Network of constraints: fundamental properties and applications to picture processing, *Inf. Sci.* 7 (1974) 97–132.
- [31] S. Mittal, B. Falkenhainer, Dynamic constraint satisfaction problems. Proceedings of the 9th, *National Conference on Artificial Intelligence AAAI* (1990) 25–32.
- [32] É. Vareilles, M. Aldanondo, P. Gaborit, Evaluation and design: a knowledge-based approach, *Int. J. Comput. Integr. Manuf.* 20 (7) (2007) 639–653.
- [33] P. Pitiot, M. Aldanondo, E. Vareilles, Concurrent product configuration and process planning: some optimization experimental results, *Comput. Ind.* 65 (4) (2014) 610–621.
- [34] E. Vareilles, T. Coudert, M. Aldanondo, L. Geneste, J. Abeille, *System Design and Project Planning: Model and Rules to Manage Their Interactions*, *Integrated Computer-Aided Engineering*, 22, IOS Press, 2015, pp. 327–342.