# Open Configuration: a New Approach to Product Customization

**Linda L. Zhang**[1] and **Xiaoyu Chen**[*1, 2] and **Andreas Falkner**[3] and **Chengbin Chu**[2]

**Abstract.** State-of-the-art product configuration enables companies to deliver customized products by selecting and assembling predefined configuration elements based on known relationships. This paper introduces an innovative concept, open configuration, in order to assist companies in configuring products that correspond exactly to what customers want. Superior to product configuration, open configuration involves both predefined configuration elements and new ones in configuring customized products. As a first step, this study explains the concept of open configuration and the basic principles. It also discusses in detail the challenges involved in open configuration, such as conceptual model development, open configuration optimization, and open configuration knowledge representation.

## 1 INTRODUCTION

With the advancement of design and manufacturing technologies, customers are no longer satisfied with standardized products. They increasingly demand products that could satisfy their individual needs. As a result, companies need to timely offer customized products at affordable costs to survive [1]. With traditional design approaches, companies cannot efficiently develop customized products [2, 3]. Product configuration has been proposed to enable companies to deliver customized products at low costs with short delivery times. Product configuration has been widely applied to a variety of industries, including computer, telecommunication systems, transportation, industrial products, medical systems and services [4]. It brings companies a number of advantages in delivering required products. These advantages include managing product variety [5], shortening delivery time [6], improving product quality [7], simplifying order acquisition and fulfilment activities [8], etc.

Product configuration has received much attention from industrial and academia alike. Researchers have approached product configuration from different perspectives and have developed diverse methods, methodologies, approaches, and algorithms to solve different configuration issues and problems. In spite of the diversities among these solution tools, they are developed based on a common assumption: the configuration elements, such as components, modules, attributes, functions, and their relationships are predefined. In relation to this assumption, the products that can be configured are known in principle even if not explicitly listable [2]. In this regard, product configuration cannot deal with such products that demand new functions and components in addition to the predefined ones. In another word, it cannot configure customized products in a true sense, i.e., to the full extent that it covers all reasonable and unforeseen customer requirements.

This study proposes an innovative concept 'open configuration' in order to help companies configure such products that can meet both predefined and unforeseen customer requirements, that is, to meet customer requirements as complete as possible without making too much compromise (see Section 2). In this regard, in configuring customized products, open configuration deals with not only the addition of new configuration elements, such as functions, components, but also the modification of existing configuration elements, more specifically components. Existing component modification is to accommodate the integration of new components with the predefined ones.

In the rest of this paper, Section 2 uses a fridge configuration example to illustrate the limitation of product configuration, i.e., the product configured lie in a known range in accordance with the predefined components. Section 3 introduces the concept of open configuration, its basic principles, and its process. Section 4 sheds lights on the challenges involved in open configuration. We end the paper in Section 5 by pointing out the ongoing research that we are working on.

## 2 PRODUCT CONFIGURATION

As a special design activity, product configuration capitalizes on design results, such as components, attributes and their relationships [9, 10]. It entails such a process that based on given customer requirements, suitable components are selected from the set of predefined component types; the selected components are evaluated and further arranged into products according to the configuration constraints and rules.

Take fridge configuration as an example. Assume in this example, there are 6 component types, including *Refrigerator (R)*, *Freezer (F)*, *Freezer drawer (Fd)*, *Variable compartment (V)*, *Base (B)*, *Outer casing (O)*. Each component type is defined by a set of attributes (number, size, price) and each attribute can assume a number of values. Table 1 summarizes these component types, the attributes, and attribute values.

For example, $N_R:(1,2)$ represents the number of *Refrigerators* in one fridge can be 1 or 2; $S_R:(\text{small, medium, large, extra-large})$ indicates the component *Refrigerator* has four different sizes: small, medium, large, extra-large. Price mentioned hereinafter states the price of the configured fridge.

[1] IESEG School of Management (LEM-CNRS), Lille-Paris, France
[2] Ecole Centrale Paris (Laboratoire Genie Industriel), Paris, France
[3] Siemens AG Österreich, Vienna, Austria
[*] Corresponding author: x.chen@ieseg.fr

**Table 1.** The attributes of the fridge components.

| Component types | Number | Size | Price |
|---|---|---|---|
| Refrigerator | 1-2 | small, medium, large, extra-large | depending on size |
| Freezer | 0-1 | small, large, extra-large | depending on size |
| Freezer drawer | 0-2 | small | P(*Fd*) ( i.e., a fixed price) |
| Variable compartment | 0-1 | small | P(*V*) (i.e., a fixed price) |
| Base | 1 | standard, wide | depending on size |
| Outer casing | 1 | standard, wide | depending on size |

There are relationships among components, among attributes, and between components and attributes. For examples, $\{S_R = \text{large}, N_F = 1\} \rightarrow \{S_F = \text{small}\}$ means if one large sized *Refrigerator* and one *Freezer* are selected, the size of the *Freezer* is small; $N_{Fd} \neq 0 \rightarrow \{N_R = 2, S_R = \text{medium}\}$ states that if the component *Freezer drawer* is selected then two medium *Refrigerators* are required. The other relationships include: $\{S_R = \text{medium}, N_F = 0\} \rightarrow N_R = 2$; $\{S_F = \text{small}, S_R = \text{small}\} \rightarrow N_V = 1$; $\{S_R = \text{extra-large}, N_F = 1\} \rightarrow \{S_F = \text{extra-large}\}$; $\{S_F = \text{extra-large}\} \rightarrow \{S_B = \text{wide}, S_O = \text{wide}\}$; $\{N_V = 1, N_F = 0\} \rightarrow \{N_R = 1, S_R = \text{large}, S_V = \text{small}\}$; $\{S_R = \text{small}, N_F = 1\} \rightarrow \{S_F = \text{large}\}$.

There are four additional rules, including (1) $(N_R + N_V + N_F) \leq 3$, meaning the total number of *Refrigerator*, *Variable compartment*, and *Freezer* in one fridge should be no more than 3, (2) $N_R = 2 \rightarrow N_V + N_F = 0$, indicating if two *Refrigerators* are selected, the number of *Freezer* and *Variable compartment* is zero, (3) $N_{Fd} + N_F = 0$ representing that *Freezer* cannot be selected together with *Freezer drawer*, and (4) $N_{Fd} + N_V = 0$ indicating that *Freezer drawer* cannot be selected together with *Variable compartment*.

According to the above pre-defined components and their relationships, only 17 fridge configurations are available as possible solutions. While Fig. 1 shows 8 fridge configurations due to the space issue, different positions of components in Fig. 1.c, Fig. 1.d, Fig. 1.e, Fig. 1.f, and Fig. 1.g lead to the other 9 fridge configurations. All customized fridges to be configured based on customer requirements fall into this range of configuration solutions. (Note: Fridges from the left to the right are arranged based on the increase of price.) Take fridge f in Fig. 1 as an example to explain the components and their attributes in the configuration solution. This fridge configuration is represented as $FC_f = \{R:1, \text{small}\ ; V:1, \text{small}\ ; F:1, \text{small}\ ; B:1, \text{standard}\ ; O:1, \text{standard}\}$. It has one small *Refrigerator* on top, one small *Variable compartment* in the middle, one small *Freezer* at the bottom, one standard *Base*, and one standard *Outer casing*.

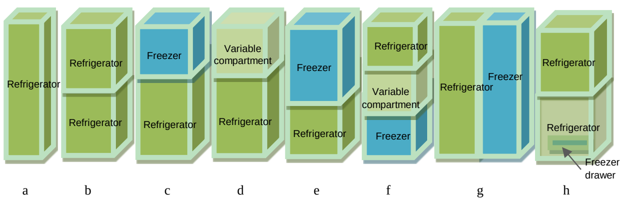


**Figure 1.** Fridge configuration solutions

Suppose the requirements from a customer include a cheaper fridge with a freezer and a large refrigerator. In accordance with these requirements, the constraints can be modeled as $\{R:1, \text{large}; N_F = 1; \min P\}$. The configured fridge must satisfy these constraints and additional rules mentioned earlier while fulfilling the customer requirements. In this regard, the constraints $\{R:1, \text{large}\}$ and $\{N_F = 1\}$ limit the possible choices to: $\{FC_c, FC_e\}$, i.e., the configuration solutions shown in Figs. 1.c and 1.e. The cost constraint $\{\min P\}$ indicating the minimal price results in the final solution to be $FC_c = \{R:1, \text{large}\ ; F:1, \text{small}\ ; B:1, \text{standard}\ ; O:1, \text{standard}\}$.

As only predefined elements are involved, product configuration fails to provide customized products in a true sense or provides these products which can meet unforeseen customer requirements. Take the above fridge configuration as an example. Suppose that the requirements from another customer include any of the following:

- a fridge consisting of only one medium refrigerator,
- a fridge consisting of 2 freezers,
- an outer casing with a special color, and
- a cheaper fridge to be moved easily and with at least one freezer drawer.

In general, the first two requirements violate some predefined constraints (although the first one requires a new - lower - type of outer casing as a side-effect); the last two introduce new concepts. In more detail, the third requirement requires a new attribute value for the component outer casing. The last one is more complex. A part of it, i.e., being cheaper and with one freezer drawer, can be fulfilled by the predefined functions and components, while the rest cannot be fulfilled by the available functions, thus calling for a new function: ‘to be movable’. This new function, in turn, needs new components, such as ‘wheels’, ‘brakes’, etc., which are necessary for delivering this function. Because of the lack of these components, product configuration can provide the customer with one of the fridges shown in Fig. 1 without satisfying all his requirements. The customer, thus, has to accept this fridge by making compromise (e.g., accept a cheapest fridge with a freezer drawer, which cannot be moved easily).

## 3 OPEN CONFIGURATION

In order to help companies configure customized products that correspond exactly to what a customer requires, this paper puts forward the concept of open configuration. The basic principle and general process of open configuration are introduced below.

### 3.1 Open configuration concept

Built on top of product configuration, open configuration is to configure customized products to meet customer requirements in a true sense. Similar as product configuration, it utilizes design results, selects components, and arranges the selected components according to constraints and rules. In extension to product configuration, it involves new component design, more specifically the specification of functions and the selection of the corresponding components. In addition, it deals with the modification of the predefined components, which allows the integration of new configuration elements.

### 3.2 Open configuration overview and process

Open configuration involves two types of knowledge: predefined knowledge and dynamic knowledge. Predefined knowledge relates to predefined functions, components, and relationships; dynamic knowledge is associated with newly defined elements. In relation to these customer requirements, which can be fulfilled by the predefined functions (i.e., Type Ⅰ requirements in Fig. 2), the corresponding components are selected, while for these requirements, which cannot be fulfilled by the predefined functions (i.e., Type Ⅱ requirements in the figure), new functions and corresponding components are specified. The specification of these new configuration elements contributes to the extension of the dynamic knowledge. The relationships among the predefined elements and the newly defined elements are specified as well. This specification contributes to the interaction between the predefined knowledge and the dynamic knowledge. By respecting the constraints embedded in both the predefined and dynamic knowledge, all necessary components are selected, modified, and arranged into a customized product.

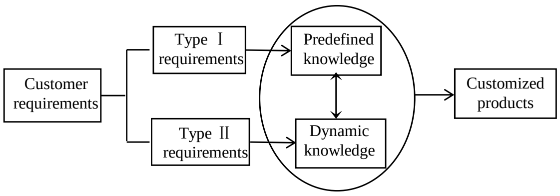


**Figure 2.** Open configuration overview

In more detail, suppose that given customer requirements are valid, complete and do not conflict with one another. These requirements are evaluated first to determine whether or not they can be fulfilled by the available configuration elements (i.e., functions and components). According to the evaluation results, these requirements are classified into Type Ⅰ and Type Ⅱ requirements. Fig. 3 summarizes this process.

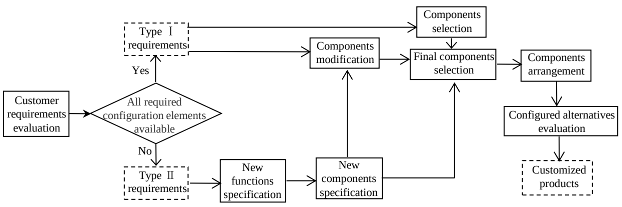


**Figure 3.** Open configuration process

For Type Ⅱ requirements, new functions are specified and all possible components which can realize these functions are subsequently determined. Also specified are the relationships among functions, among components, and between functions and components. This process contributes to the extension of the dynamic knowledge. For Type Ⅰ requirements, all possible components are selected from the predefined ones. In addition, to be compatible with the newly introduced components, some predefined components are modified by respecting constrains and rules embedded in the predefined and dynamic knowledge. This process reflects the interaction between the dynamic and predefined knowledge. From the modified components, newly introduced components, and selected predefined components, suitable components are further selected for forming configuration alternatives, which can meet customer requirements. In the selection, consistency and compatibility evaluations might be carried out. The selected components are arranged into product configuration alternatives by following the product structure described in the dynamic and predefined knowledge. These configuration alternatives are further evaluated under certain criteria. Based on the evaluation results, the optimal one or multiple are suggested to customers.

# 4 CHALLENGES INVOLVED IN OPEN CONFIGURATION

In accordance with the involvement of new configuration elements, open configuration changes the basic assumptions and reasoning processes of product configuration. In this regard, there are a number of potential challenges involved in open configuration. Due to the page limitation, this paper discusses five of these challenges, including open configuration modeling, system design and development, open configuration solving, open configuration optimization, and open configuration knowledge representation.

## 4.1 Open configuration modeling

Open configuration modeling addresses the modeling of open configuration knowledge and the reasoning mechanism for using the configuration knowledge. The modeling of open configuration knowledge is to model configuration elements, constraints, and rules. It involves two kinds of knowledge: predefined knowledge and dynamic knowledge. A product model and corresponding functional architectures should be developed for defining and further classifying the two different types of knowledge. The modeling of the reasoning mechanism is to shed light on (1) how new functions are specified, (2) how new components are determined, and (3) how components are selected and arranged into products.

In open configuration modeling, the components and functions are characterized by their attributes, while the inter-connections among the components are represented by connections and ports. The modeling of the dynamic knowledge needs to take into account the fact that new functions and components are added based on the unforeseen customer requirements. Thus, its modeling involves newly-added concepts, constraints, and rules. The modeling of the predefined knowledge needs to consider these predefined components, modified components, and their relationships. The interaction between predefined knowledge and dynamic knowledge needs to be modeled as well.

Open configuration modeling is more sophisticated than configuration modeling due to the involvement of the dynamic knowledge. In this regard, it is interesting to see whether or not these techniques which are suitable for modeling product configuration (e.g., Unified Modeling Language (UML), Alloy, and generative Constraint Satisfaction Problem (CSP) [11]) can be used to model open configuration. If these techniques are feasible, how can they be modified or adjusted to model open configuration. If these techniques are not feasible, new modeling formalisms and constructs are to be developed.

## 4.2 System design and development

System design and development for open configuration refers to the design and development of the computer information system to implement open configuration, i.e., open configurators. Open configurators consist of a customer input module which deals with customer requirements evaluation, open configuration knowledge bases, reasoning and evaluation mechanisms, optimization and diagnosis mechanisms, and an output module which communicates the configuration results with users. Different from product configurators, open configurators involve two knowledge bases: a knowledge base for the predefined knowledge and the other for the dynamic knowledge. Joint reasoning mechanisms between the two knowledge bases are required, which mainly associate with interacting and integrating elements from the two knowledge bases. For the dynamic knowledge base, new elements design modules are needed to develop and maintain this knowledge base. The new elements design modules include the module for specifying new functions with respect to the requirements, the module for selecting new components to fulfill new functions and the module for interfacing with the predefined elements. For the predefined knowledge base, different from product configurators, there need to be a modification module for modify existing components to be compatible with the new ones.

In designing and developing open configurators, the techniques should have the ability to model dynamic knowledge and the interaction between dynamic knowledge and predefined knowledge. In this regard, the available system design techniques for product configuration may need to be modified in designing and developing open configurators.

## 4.3 Open configuration knowledge representation

Open configuration knowledge representation entails the effective organization of open configuration knowledge, including the predefined and dynamic knowledge. It logically uniforms the open configuration knowledge and enables the utilization of the knowledge in different configuration tasks.

The representation of open configuration knowledge includes the representation of predefined components, relationships, constraints and rules; the representation of newly-added components, relationships, constraints and rules; and the representation of the constraints and relationships between predefined knowledge and newly added knowledge. From the experience of the knowledge representation for product configuration, open configuration should be considered as both a classification problem (i.e., capturing the aspects of taxonomy and topology) and a constraint satisfaction problem (i.e., capturing the aspects of constraints and resource balancing). Considering the dynamic and indeterminate feature of open configuration, it might be potentially challenging to capture different aspects of open configuration knowledge (e.g., taxonomy, topology, constraints, and resource balancing) in one model. Further studies may try to design new models (or sub models to be embedded in the available tools) separately on each aspect and joint them together to represent the knowledge.

## 4.4 Open configuration solving

Open configuration solving relates to the development and application of algorithms or other tools to solve open configuration problems. In solving an open configuration problem, the problem needs to be modeled first with respect to customer requirements and configuration rules. To solve this model, algorithms need to be developed subsequently.

In the situation that customer requirements demand new functions, the dynamic knowledge will be specified. The modeling of open configuration problem will associate with the interaction between the customer requirements and two types of knowledge (predefined knowledge and dynamic knowledge). The main difficulties are (1) the modeling of new function specification, (2) the modeling of new components selection according to the customer requirements, (3) and the modeling of the interaction between new components and selected existing components. After modeling an open configuration problem, suitable algorithms need to be developed to solve the model. Because of the differences between product configuration and open configuration and the corresponding differences between a product configuration model and an open configuration model, these algorithms, which are suitable for product configuration solving, may not be applicable for open configuration solving. Thus, new algorithms are to be developed.

## 4.5 Open configuration optimization

During each step of open configuration, optimal functions, components and structures need to be specified from a number of alternatives. The dynamic feature of open configuration increases the degree of difficulty in optimizing the new functions, new components, and the interaction between new components and predefined ones. In this regard, an explicit optimization mechanism needs to be developed.

In accordance with the open configuration process discussed earlier, the optimization mechanism should evaluate the configuration elements at three levels. In the first level, the mechanism should evaluate all the possible function alternatives for fulfilling Type II requirements and decide on the optimal ones. This optimization might be based on, e.g., the performance and completeness of these function alternatives. In the second level, the mechanism should evaluate all the possible component alternatives for delivering the determined new functions and decide on the optimal ones. This optimization may take into account, e.g., the compatibility among the new components and the interaction with predefined components. In the third level, the mechanism should evaluate all the product configuration alternatives and decide on the optimal ones. This optimization may consider, e.g., product reliability.

# 5 CONCLUSION

In response to the limitation of product configuration, this paper proposed open configuration to help design customer-driven product in a true sense. It introduced the concept and process of open configuration. It also discussed several challenges involved in open configuration. Currently, we are working on the formulation of open configuration. In the formulation, new components, relationships among new components, and relationships between new components and existing components will be defined and modeled. This formulation is to rigorously define open configuration and shed light on the reasoning behind open configuration.

## REFERENCES


[1] M. Heiskala, K.S. Paloheimo, and J. Tiihonen, *Mass customization of services: benefits and challenges of configurable services*, 206-221, Proceedings of Frontiers of e-Business Research, Tampere, Finland, 2005.

[2] D. Sabin and R. Weigel, 'Product configuration frameworks-a survey', *IEEE Intelligent Systems and Their Applications*, **13(4)**, 42-49, (1998).

[3] S. Schmitt and R. Bergmann, *Applying case-based reasoning technology for product selection and customization in electronic commerce environments*, 42-48, Proceedings of the 12th International Bled Electronic Commerce Conference, Bled, Slovenia, 1999.

[4] A. Trentin, E. Perin, and C. Forza, 'Product configurator impact on product quality', *International Journal of Production Economics*, **135(2)**, 850-859, (2012).

[5] C. Forza and F. Salvador, 'Managing for variety in the order acquisition and fulfilment process: the contribution of product configuration systems', *International Journal of Production Economics*, **76(1)**, 87-98, (2002a).

[6] A. Haug, L. Hvam, and N.H. Mortensen, 'The impact of product configurators on lead times in engineering-oriented companies', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **25(2)**, 197-206, (2011).

[7] A. Trentin, E. Perin, and C. Forza, 'Overcoming the customization-responsiveness squeeze by using product configurators: Beyond anecdotal evidence', *Computers in Industry*, **62(3)**, 260-268, (2011).

[8] C. Forza and F. Salvador, 'Product configuration and inter-firm co-ordination: an innovative solution from a small manufacturing enterprise', *Computers in Industry*, **49(1)**, 37-46, (2002b).

[9] S. Mittal and F. Frayman, *Towards a generic model of configuration tasks*, 1395-1401, Proceedings of the 11th International Joint Conference on Artificial Intelligence, Detroit, USA, 1989.

[10] D. Brown, 'Defining configuration', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **12(4)**, 301-305, (1998).

[11] A. Falkner, A. Haselböck, G. Schenner, and H. Schreiner, 'Modeling and solving technical product configuration problems', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **25(2)**, 115-129, (2011).