



HAL
open science

Adaptability Design to Meet Dynamic Customer's Needs

Nattawut Janthong, Daniel Brissaud, Suthep Butdee

► **To cite this version:**

Nattawut Janthong, Daniel Brissaud, Suthep Butdee. Adaptability Design to Meet Dynamic Customer's Needs. IEEE International Conference on Industrial Engineering and Engineering Management (IEEM09), Dec 2009, Hong Kong, Hong Kong SAR China. hal-00413946

HAL Id: hal-00413946

<https://hal.science/hal-00413946>

Submitted on 7 Sep 2009

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.

Adaptability Design to Meet Dynamic Customer's Needs

N. Janthong^{1,2,3}, D. Brissaud¹, S. Butdee²

¹G-SCOP research laboratory, University of Grenoble, France

²Integrated Manufacturing System Research Center (IMSRC), Department of Production Engineering,

Faculty of Engineering, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand

³Thai-France Innovation Institute, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand

Abstract - The customer is the core element of business based on current market situation, even for industrial equipment which customers' requirements are now very various. It means that the inside processes of the manufacturer should adjust to customer's requirements: the company must be very flexible among other properties. The challenge is that the flexibility comes from the management of the knowledge of the company by improving the adaptive design process. A knowledge-based design process has been developed by combining the axiomatic design and case-based reasoning approaches, to enhance the performance of the design personnel and is the key factor for the success of the company.

Keywords - Knowledge-based, Customisation, Adaptable design

I. INTRODUCTION

In many business fields, customers have become very demanding to the point that they only ask for individualized products and services. These changes at the customer's level have forced companies to react by developing new operations strategies in order to secure market shares and improve profits. Furthermore, companies have realized that they have to concentrate on their strengths and core competencies while closely working with their partners if they want to effectively and efficiently respond to ever change or customer's requirements.

Related works can be split in two complementary approaches: Product configuration design and customization. Product configuration design tries to design a product from a selection of elements from a more or less set of pre-existing things: Modular design aims to develop product architecture consisting of physically detachable units/modules [1]; Platform design entails the identification of common attributes within a series of products. Those two design strategies are producer's offer oriented but the industrial issue is now individualized customer's demand oriented. Customization is an approach to meet customers' wishes [2]. It consists of modular and platform strategies dedicated to customize major functions or aesthetic qualities. All those approaches needs to formalize a priori all that should occur in product demand. A more flexible approach to answer unexpected demand and to adapt to SMEs where resources are limited is now needed.

Design adaptability refers to the adaptability in the design of a product, so that the design can be modified and adapted to produce a new product [3]. The first advantage is that the producer can adapt the same design to different requirements and produce different products. The main objective of adaptable design is to meet customers' requirements in their specificity and personalization. The second advantage comes from that design adaptability aims at reusing the same 'design' for the creation of different products. It means that adaptive design helps reduce time to develop the new product. By reusing previous designs, an engineer can reduce duration and cost of development cycle and risks on product quality and performances. Moreover, the relevant and innovative information in any discipline may also be mobilized and used to update or adapt a previous design in response to changes in technology or market preferences. The challenge is that the flexibility comes from the management of knowledge of the company by improving the adaptive design process. Hence, organizing, storing and retrieving information on previous product designs are the most important tasks in knowledge utilization to provide designers with the suitable solution that meets or almost meets to customer's needs.

Our motivation is to develop a methodology in (re)design of products for supported dynamic customer's needs applied on the design and manufacture of industrial vehicles in a Thai company. Developing this approach will give a great advantage to the company against its competitors by succeeding in developing customized products and in reducing lead time by a strong reusability of designs, components and process.

The paper aims at presenting the knowledge-based design process to (re)design products meeting customer's requirements. Section 2 will present the literature review then section 3 will develop the methodology proposed. Conclusion is given in section 4.

II. LITERATURE REVIEW

The main technique reviewed was case-based reasoning (CBR) applying on design. The basic idea of case-based reasoning is that new problems can be tackled with adapting solutions that were used to solve previous problems [2]. The practice shows that often it is more efficient to solve a problem by starting from a solution of

a previous similar problem than to generate the entire solution from scratch. Due to the mentioned properties, CBR systems have a diversity of applications in troubleshooting and in mechanical design [4] etc. The two major research issues in CBR approaches to design are the representations of design cases and the process model for recalling and adapting design cases. Representing design cases requires an abstraction of the experience into a symbolic form that the system can manipulate. Design-case recalling involves finding a relevant design experience: it is decomposed into the subtasks of indexing, retrieving, and selecting. Indexing design cases is a critical issue in CBR approach, and CBR systems suffer from an inability to distinguish between cases if indexing is inadequate. Design-case adaptation recognizes the differences between the selected design case and the new design problem, and changes the design case so that it solves the new design problem.

The concept of case based reasoning can be defined in the way to organize information or data, and this concept is applied to either 'idea', innovation or any other kinds of information that is to be stored and used somehow afterwards.

However, in industrial products designs are quite difficult to represent as a well-structured list of features. Therefore, there is a need to develop a method that clearly determines industrial products design requirements. Such a method that rigorously defines the design requirements could be axiomatic design among others.

Axiomatic design defines design as the creation of synthesized solutions in the form of products, processes or systems that satisfy perceived needs through mappings between Functional Requirements (FRs) and Design Parameters (DPs) [5]. A fundamental aspect of the mapping process is the idea of decompositions through zigzagging. The design progresses from a higher, abstract level down to a more detailed level. These results in the formation of design hierarchies in the FRs and DPs are similar in nature to standard product functional and structural hierarchies. Thus it can identify which parts of the design structure are used to perform specific functions.

To facilitate (re)designing industrial product, this paper combines the axiomatic design and case-based reasoning approaches. The case based reasoning is used as a general framework for the reuse of design purpose and applied when a similar function is required. The axiomatic design principle is used for creating cases by analyzing existing products which FRs and DPs were decomposed. These FRs and DPs are utilized as case index and case representation in case library. It is also used for creating design database or design library by identifying relations between FRs and possible DPs of each component in design library. The information content is used for evaluating design solutions (DPs) from design library or design database composed of various components to fulfill new functional requirements which do not exist yet in the case library. The design with satisfied independence axiom provides the sequence to modify DPs in the adaptation process.

III. DESIGN METHODOLOGY

The core of the reuse design principle is structured in two main stages, preceded by the expected design description and followed by the capitalization of the work as shown in Fig 1.

The first stage deals with a research of similarity between the new case to be designed and all the previous cases that were capitalized in a library. The similarity search leans on the comparison of the matching of the values of the different requirements in the cases and an evaluation of the gap. If an exact matching is found, the new case is considered to be identical of the previous one and consequently that the solution of the previous case can be considered to be the solution of the new case. If no case matches the new one, the system tries to propose to designers cases that seems close to the problem to be dealt with. When one or several are considered to be close, based on designers' experience, the solution of one of those cases is considered to be the referent solution and the designers will improve it to meet the new requirements (knowledge-based adaptation of the product definition and adaptation of the business matters that are considered to be the second stage of the methodology). When no case can be considered as reference, the search strategy is unsuccessful and the designers have to manually design a new product.

The second stage deals with the adaptation of the referent solution to create the new one when a referent solution has been decided. A solution associated with a case and stored in the library, includes data about the manufacturing process and consequently on time and cost. These referent data are adapted to evaluate their values for the new case.

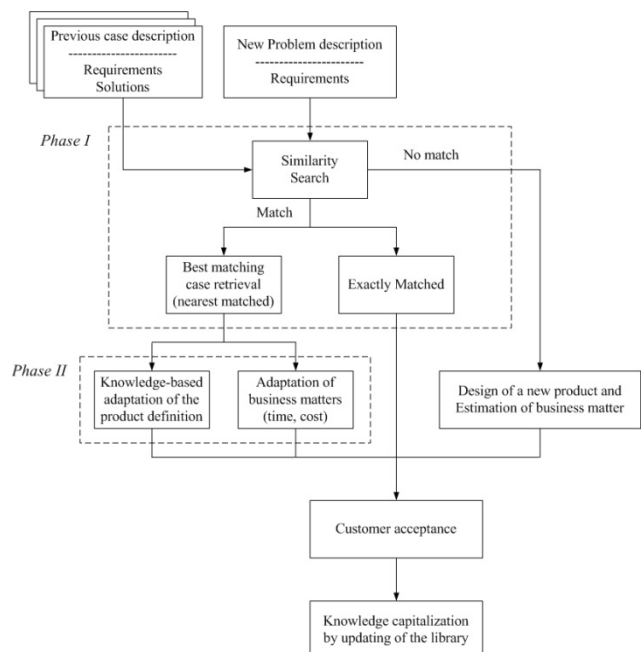


Fig. 1. Design reuses principle

This adaptation is very quick and enables engineers to negotiate the manufacturing contract with the customer as soon as possible and before time and cost have been deployed in the company to develop the new solution. When a new design has been done, it is stored in the library to be capitalized and eventually reused as reference for a new design.

The design methodology has been developed based on reuse design principle afore mentioned. The methodology as shown in Fig. 2 is based on the assumption that the designers do not need to design products from scratch every time. They go through their ability to access to existing designs from related products and components then to revise them to fulfill specific customers' needs.

The function structure and the physical structure of products from past design experiences were stored in case library. Moreover, the design library kept the designs which included components information and their function definition, which came from supplier's standard catalogue. Both case library and design library were utilized to create suitable design solutions to achieve new functionalities.

The process started by the comparison of new customers' requirements and constraints to function structures and physical structures of existing products that perform similar requirements and constraints. The result is that functions can be separated in product functions that have already been developed in existing designs, and add-on functions that did not exist and require to be fulfilled through the designs process. To achieve that, add-on functions are decomposed in terms of functional requirements; physical solutions are retrieved by comparison to other products of the family and by searching in designs database and standard components library. The retrieval process based on functionality and other specifications is accomplished by the aid of inference engine. Both rules and cases are necessary for the reasoning process. Then, adaptations of the design are needed to re-configure and integrate components to achieve the new design. Thus, product architecture, platforms, modules as well as functional and physical structures are the main drivers to create the case base. The adaptation process needs to follow the most suitable sequence.

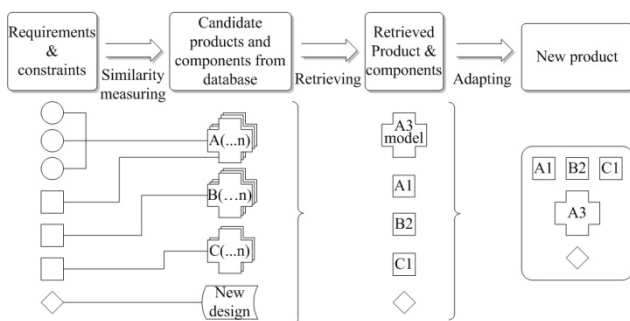


Fig. 2. Design methodology

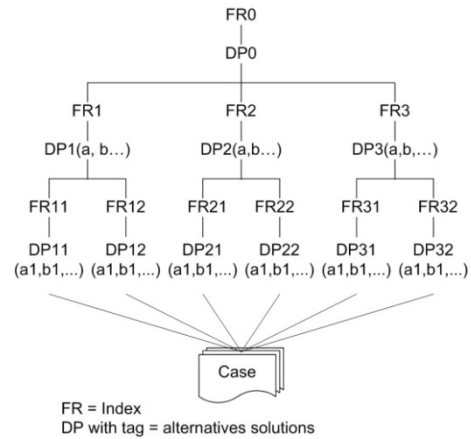


Fig. 3. An index structure in case library

A. Case Representation

The basic idea is to organize specific cases, which share similar properties under a general structure. The scheme of a case consists of four parts, including customer's requirements, customers' constraints, functional requirements and design parameters. The case is represented in terms of a design hierarchy in each of the domains: functional and physical. The hierarchical structures in the FR-domain and the DP-domain correspond to customers' requirements. An advantage of this representation is that it allows a case to be accessed on its whole or by its parts when a new problem must be solved. Similar cases at appropriate levels of abstraction are retrieved from the case base and the solutions from these cases are combined and refined; the constraints can be used to guide adaptation.

B. Case Indexing

Case indexing involves assigning indices to cases for their quick and easy retrieval from case library. Axiomatic design decomposition principles are used to determine the indexing of both design cases and their solutions as shown in Fig 3. This indexing structure scheme also allows the composition of different case pieces to create a new solution. It shows that there are many different ways to satisfy the FRs. FR skeleton sets can be generated for each of the design cases in case library. Each step down in the hierarchy represents a refinement of the unit design. It helps distinguish between cases which lead to efficient case matching and retrieving.

C. Case retrieval

As afore mentioned, when new customers' requirements and constraints are given, similar historical design cases are searched, matched and retrieved. The result is that two major functions are classified, namely product functions from existing products in case library that have similar functions according to problem inputs and add-on functions that are not on the retrieved existing

products. Thus, the case retrieval process includes two phases – (i) similarity matching of product functions and (ii) similarity matching of add-on functions. Each phase relies on achieving two goals: finding a similar case set and finding the most similar case in this set.

In the first phase, the similarity matching of product functions as shown in Fig 4, finds the similar case set from customers' requirements (CA_i) that are compared to product function hierarchy of each case ($Case_i(FR_i)$).

For symbolic attributes, the simplest similarity measure is to score 1 for equality and 0 for inequality as follows:

$$sim(CA_i, Case_i(FR_i)) = \begin{cases} 1 & \text{if } CA_i = Case_i(FR_i); \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Thus, a set of cases from the case base that are similar to the current input case is equal to the intersection of (CA_i) and ($Case_i(FR_i)$) as follows :

$$\{P\} = CA_i \cap Case_i(FR_i) \quad ; \{P\} = \text{set of relevant products}$$

After all similar cases are found, a mechanism to find the most similar case in this set is needed. The input attribute value are used to compare the function attributes of each retrieved case. Then, the $sim(CA_i, Case_i(FR_i))$ can be calculated by:

$$sim(CA_i, Case_i(FR_i)) = 1 - \frac{(CA_i - Case_i(FR_i))}{system\ range(FR_i)} \quad (2)$$

where: $CA_i - Case_i(FR_i)$ is the difference between the feature values of the input and the retrieved case and, $system\ range(FR_i)$ is the range based on the capacity of the producer. Then to turn a normalized distance function into a similarity measure, its value subtracts from 1.

Practically designers must translate customer's requirements to product functions and customer's constraints to attribute values related to product functions. Most of the time, the customer wishes specifying parts of the solution in addition to its design problem only (sometimes the initial wish is in terms of solution prescription and a significant joint work is necessary to extract the real design problem as well). It means that identifying the DPs (then giving values to attributes when necessary) related to the design process is a crucial part of the process.

Thus, to find the most similar case in this set, the input constraints (CS_i) are used to compare the design parameters of each retrieved case. Then, the $sim(Cs_i, Case_i(DP_i))$ can be calculated by:

$$sim(Cs_i, Case_i(DP_i)) = \sum_{i=1}^n 1 - \frac{(Cs_i - Case_i(DP_i))}{system\ range(DP_i)} \quad (3)$$

where: $Cs_i - Case_i(DP_i)$ is the difference between the feature values of the input and the retrieved case and

$system\ range(DP_i)$ is the range which each DP can satisfy FR based on the capacity of the producer. Then to turn a normalized distance function into a similarity measure, its value subtracts from 1. The set of cases is ranked by these similarity scores and the retrieved case is this with the highest similarity score.

The total similarity measurement is calculated as follows:

$$Sim(I, R) = \frac{\sum_{i=1}^n w_i \times sim(f_i^I, f_i^R)}{\sum_{i=1}^n w_i} \quad (4)$$

where: $sim(f_i^I, f_i^R)$ is $sim(CA_i, Case_i(FR_i))$ and/or $sim(Cs_i, Case_i(DP_i))$, w_i is weighted value of attribute i . The highest total similarity score represented the most similar case in the set in which it will be retrieve as the solution to solve the input problem.

In the second phase, the add-on functions are the customer's attributes (CA_i) that did not match ($Case_i(FR_i)$) in the first phase. There are two possibilities for the remaining CAs. The first one is when the function does not exist in the retrieved cases but could be in other product families: the system is called to search in the other product libraries by the same procedure as in the first phase.

The second one is when the function does not exist at all in any cases of the database: the producer never did this function before for any product they did. A new design of the function of the product must be created. The add-on library and the designs database include mechanical parts, electrical parts, software modules etc. These add-on components are defined as pairs of FR and DP for single component and hierarchy of FR and DP in case of assembly components.

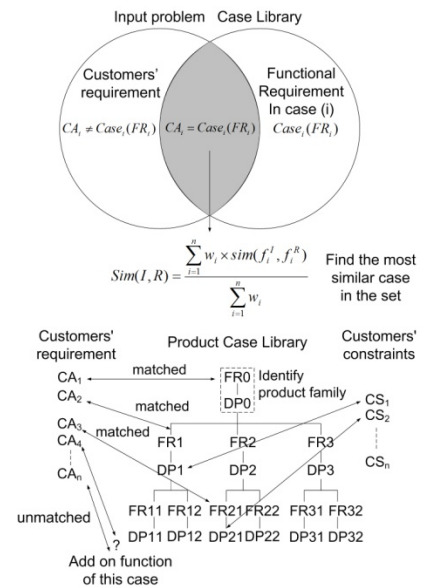


Fig. 4. Case retrieval based on similarity of product function matching

Similar to matching CA_i with $Case_i(FR_i)$ in the first phase, $Design_i(FR_i)$ are defined to distinguish the sources of information between case library for reuse design and design library for new design. The components in design database were evaluated to find a solution that satisfies the add-on function. If a CA corresponds to one function, a solution can be found by $sim(CA_i, Design_i(FR_i))$. If a CA corresponds to a hierarchy of functions the solution of the CAs can be found by $sim(CA_i(FR_i), Design(FR_i))$. Thus, matching a CA_i with $Design_i(FR_i)$ is as follows :

$$sim(CA_i, Design_i(FR_i)) = \begin{cases} 1 & \text{if } CA_i = Design_i(FR_i); \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

or

$$sim(CA_i(FR_i), Design_i(FR_i)) = \begin{cases} 1 & \text{if } CA_i(FR_i) = Design_i(FR_i); \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The solution to satisfy each FR must be evaluated by minimizing the information content of the design based on Suh's axiomatic design principle as follows:

$$I = \log \left(\frac{\text{systemrange}}{\text{commonrange}} \right) \quad (7)$$

$$I_{total} = \sum_1^n I_i \quad (8)$$

In our work, we found that designing is a complex task and it is unreasonable to expect a case base to contain all the possible design cases. It is the reason why our methodology combines case-based reasoning to initiate an appropriate design due to past experience and axiomatic design rules to provide this design with the new functions needed. This provides a combined advice that better satisfies design constraints and compatibility requirements compared to only CBR system.

D. Case Adaptation

If an exact matching case is found from the case retrieval process, its design can be used for the new order without any modification. Otherwise, an adaptation process is invoked to detect the discrepancies between the most similar case and the new order, and to reconcile the discrepancies by adapting the past design to the new situation. The adaptation knowledge is usually represented as rules. The adaptation rules specify, under a certain situation, how to modify the value of a feature, or how to insert or delete certain features of the case representation in order to generate a solution for the new problem. According to axiomatic design principles when the relationships between FRs and DPs is uncoupled design, the set of adaptation rules can be easily and automatically selected by the system to make effect on similar old cases and to produce the new one. Uncoupled design occurs when each FR is satisfied by exactly one DP. The resulting matrix is diagonal and the design equation has an exact solution. The selection of adaptation rules is done easily by comparing the conflicting differences between the new problem and the current retrieved case.

In addition, the sequence of applying the adaptation rules is also important because when the design matrix is lower triangular the resulting design is decoupled, which means that a sequence exists, where the FRs can be satisfied by adjusting DPs in a certain order. This is a very important finding, as the design process is determined to a great extent by this sequence. If the case adaptation process does not follow the sequence specified by the triangular design matrix, the system appears to be very complex, which is defined as the imaginary complexity [6]. While the traditional approach of case based reasoning does not specify how to consider the sequence to adjust parameters of old case features, so there is no clear way to guaranty the correct sequence to apply adaptation rules. The axiomatic design principles can help designers to make decision in order to adapt the old case to solve the new case without randomly manner to satisfy the desired system function.

IV. CONCLUSION

This paper has presented the concept of combining axiomatic design and case-based reasoning to assist the design process of evolving systems of industrial products. The company knowledge can be stored then reused and integrated with various technologies from design database and generate new functionalities for improving the existing products. Function and physical decompositions are the basic methods to represent cases and are also used to define indexes in case library as well as used to determine new designs when no case exists in case library. It also supports design engineers to achieve the adaptable design by the defined sequence of the adaptable process when the design is decoupled. However, the quality of the design solution depends on the set of FRs and DPs in the case and add-on library. Design engineers must carefully decompose the set of FRs and DPs in existing product functions and add-on component library functions that will be further reused.

REFERENCES

- [1] Gu, P., Hashemian, M., Sosale, S. (1997). An integrated design methodology for life cycle engineering: *Annals of CIRP*, 1997, Vol46/1, pp 71-74.
- [2] Tseng, MM., Du, X. (1998). Design by customers for mass customization products: *Annals of CIRP*, Vol 47/1, 1998, pp 103-106.
- [3] Gu, P., Hashemian, M., Nee, AYC. (2004). Adaptable design: *Annals of CIRP*, 2004, Vol 53/2
- [4] Watson, I., "Case-based reasoning is a methodology not a technology", *Knowledge-Based Systems*, Vol 12, 1999, pp. 303-308.
- [5] Suh, N.P., *The Principles of Design*, Oxford University Press, 1990.
- [6] Suh, N.P., Complexity in Engineering, *Annals of the CIRP*, Vol53/1, 2004.