CASE-BASED EVOLUTIONARY DESIGN FOR MASS CUSTOMIZATION

MITCHELL M. TSENG and JIANXIN JIAO

Department of Industrial Engineering & Engineering Management
The Hong Kong University of Science & Technology, Kowloon, Hong Kong

Abstract: Reuse suggests itself as a natural technique to satisfy the increasing diversity of customers’ demands with mass production efficiency. This paper proposes a case-based evolutionary design (CBED) approach to achieving the goal of mass customization. The essence of the approach lies in a concept of the case class and a product family architecture centered design process model. Issues associated with the approach are discussed in the context of CBED for mass customization, involving case content, memory organization, indexing, retrieval, and adaptation.

Keywords: Case-based design; Mass customization; Product design; Object-oriented; Design automation.

1. INTRODUCTION

With the growing competition on product costs and increasing diversity of customers’ demands in global markets, the manufacturing industry has been under tremendous pressure to meet the seemingly conflicting goals of efficiency and product variety. This manifests the idea of satisfying individual customers’ needs with mass production efficiency, namely, mass customization, which draws a lot of attention recently [Pine, 1993; Kotha, 1996; Lee, 1994; Tseng and Jiao, 1996a]. An important step toward this new style of manufacturing will be the development and proliferation of design repositories that provide easy access to information on previous designs, new products, technological breakthroughs, and past failures. Due to the similarity/commonality over product line or among specific customized products, reuse suggests itself as a natural technique to facilitate increasingly efficient and cost effective product development. That is, a new product that reuses a previous design at some level or to some extent should be less expensive to develop than a product that is designed from scratch [Prieto-Diaz, 1993]. By reusing prior designs, an engineer can save design time and cost by leveraging off previously worked-out solutions. The information on manufacturing or field failures linked to the retrieved designs can also be used to update or adapt an old design in response to changes in technology, government regulations or market preferences [Tseng and Jiao, 1996b].

Two potential reasons for the lack of reuse, particularly at the conceptual design level, are: (1) lack of institutional memory of related product designs; and (2) the art of specialization, i.e., the belief on the part of the design team that the current product with customization is unique and thus, would not benefit from the experience of designers of similar customized products. Without an institutional memory, that is, a method of collecting and storing design experience accessible to subsequent designers, it is difficult, if not impossible, to reuse prior design experience and solutions [Chen and Lee, 1993]. Moreover, the specialization of customization emphasizes the differences between products rather than the similarities. This leads to situations in which the design team holds the belief that each new version of a customized product needs to be designed from scratch [Prieto-Diaz, 1993]. The result is that each new product evolves in isolation, with its own concepts, structures, and manufacturing processes. The customer-specific features permeate the design process from functional specification, to conceptual design, and to design details. Even if a mechanism for institutional memory were available, the customization-centered concepts detract from the ability of subsequent design activities to recognize similarities/commonalities and reuse relevant design experience to achieve the economy of scale.

Case-Based Reasoning (CBR) combines a cognitive model describing how people use and reason from past experience and a technology for finding and presenting such experiences [Domeshek and Kolodner, 1992]. The cognitive model is a memory-centered model. The basic idea is that people are good at figuring out what to do in new situations largely because they are able to remember and adapt things they did in similar previous situations [Domeshek and Kolodner, 1992]. The connection between CBR and traditional engineering design is clear and well studied [Hua et al., 1992; Sycara et al., 1992]. Given that design is an ill-structured and open-ended problem, design experience and heuristics play an important role in the design process. It is further claimed that designers generally reason from past experience rather than first principles [Maher, 1990]. Case-Based Design
(CBD) starts from complete cases, implicitly achieving tradeoffs between several functions. Cases represent specific design episodes and are reordered as descriptions of the problem and its solution, no process or justification is included in a case. In addition, as has been often discussed in the CBR literature [Domeshek and Kolodner, 1992], CBD is an experience-based method which is superior to rule-based or model-based systems in all but the most routine applications. Implicit utilization of domain knowledge enables CBD to avoid the knowledge elicitation bottleneck which is always encountered in the knowledge-based systems.

In CBD, there are several issues that are fundamental to the success of the approach [Domeshek and Kolodner, 1992]. In this research, the issues are identified and related to their instantiations in the context of the CBR approach to design for mass customization [Tseng and Jiao, 1996a]. These issues are: (1) Case content — what constitutes a case. Considering the above-mentioned specialization dilemma, it is imperative to enrich cases by reflecting their highly specialized and domain-dependent nature; (2) Comprehension problem — how to organize case memory, in particular in accordance with domain design model; (3) Indexing problem — to define the mechanism for case retrieval and best case selection; (4) Adapting problem — how can a retrieved solution be modified to the new context of the design problem; and (5) Case acquisition problem — how can cases be acquired from the domain experts and put in the case memory. A methodology that tackles these issues is proposed in this paper, namely, Case-Based Evolutionary Design for Mass Customization (CBED/MC). The remainder of this paper describes CBED/MC methodology and its treatment of the above issues, as well as considerations on mass customization.

2. CBED/MC METHODOLOGY

CBED/MC is based on the assumption that access to examples of existing designs and experiences from related products will facilitate reuse. Thus, its case base consists of solutions and examples from past design experiences. The specialization issue aforementioned further proposes that effective decision support also requires a unifying product platform through which to view, extract, and integrate past experiences. The product platform supports CBED/MC at the back-end to provide designers with a set of concepts and common solutions to specify the current design that emphasizes its similarities/commonalities to past designs so as to keep the economy of scale.

2.1. Product family architecture. CBED/MC organizes information around features common to the whole set of product designs by means of its product platform, namely, Product Family Architecture (PFA) [Tseng and Jiao, 1996a] (Fig. 1). The development of PFA acts as an essential link in design for mass customization to provide context coherent integration. The merits of PFA is judged by its capability to cover a group of individual product requirements. In other words, a good PFA possesses an inherent structure that can generate a group of products to satisfy a spectrum of customers’ needs while keeping mass production efficiency. On the other hand, mass customization can only be achieved if customers’ requirements are effectively captured and addressed in design, manufacturing and services. PFA plays an important role in instantiating the necessity to be customer-focused and product-driven in manufacturing enterprise. That is PFA should be convenient for the sales people to bring out the customers’ requirements and in the mean time, be conducive to facilitating the design, manufacturing and service. In addition, in light of CBD, a case base acts as the driving force to case-based reasoning mechanism. The quality of the case content in a case base is decisive to the overall CBD system capability. As in a knowledge base in expert systems applications, special efforts should be made in the construction of case base, in particular to reflect the highly specialized nature of domain design problems. In addition, going back to the aforementioned case acquisition issue in CBD, it is doubtful that all the previous designs have the same priorities to be considered as design examples and are stored in the case base. That is only those good designs in some sense can be selected as cases, while a design considered to be inappropriate should be discarded in constructing a case base. As a result, CBED/MC employs PFA as the basis for constructing the case base so as to guarantee that only good designs in terms of mass customization become the candidates for future reuse.

The PFA has two decomposition hierarchies: one a functional and the other structural [Tseng and Jiao, 1996a]. A product is composed of parts which operate together to achieve a goal. In our research into mass customization [Tseng and Jiao, 1996a], the structural hierarchy of PFA consist of a collection of building blocks (or modules) with network-like topological inter-relationships to figure out end products according to configuration rules. Main issues associated with PFA structural hierarchy include (a) building blocks in design and production, (b) network structure of inter-relationships among building blocks, (c) configuration rules for synthesizing end products, and (d) the economic evaluation. The functional hierarchy of PFA resembles a
taxonomy of product functional requirements (FRs) which captures diverse customer needs and points to various sets of product design objects grouped together with a recognizable purpose. The holistic view adds a high level system perspective on the whole product families. The back-end structural view comprises the components for configuring end products which are manipulated by the functional view at the front end.

2.2. Case-based evolutionary design based on PFA. Evolutionary design is a frequently adopted technique in practice to enhance reusability, where products already have an installed base or existing product families [Tseng and Jiao, 1996b]. With the explication of underpinning PFA, evolutionary design can be considered as a routine design, in which the structure of a product is predetermined, and as a result, the task of design is actually to acquire customization requirements, select correct alternative building blocks and determine relative parameters. Apparently, the CBR paradigm is best applicable to this problem domain. Facing the new paradigm of business competition, we developed PFA-based approach to mass customization [Tseng and Jiao, 1996a]. PFA provides a generic architecture to capture and utilize commonalities, within which each new product instantiates and extends so as to anchor future designs to a common product line structure. When PFA is applied, case-based evolutionary design (CBED) not only unburdens the knowledge base from keeping variant forms of the same solution, but also models the design process of a class of products that can widely variegate designs based on individual customization requirements within a coherent framework. As a result, CBED/MC adopts PFA as the basis for modification to meet specific customer needs while maintaining the integrity of the product family and the continuity of the infrastructure, hence, leveraging existing design and manufacturing investments.

2.3. The concept of case class. Case class is the class of cases, which describes the common structures of a set of cases and the shared knowledge about case analysis, evaluation and modification. The case class consists of three parts, i.e., CC = (C_A, C_M, C_K). The common attribute part (C_A) summarizes the information on the structure of cases. Design prototype is obtained by inheriting all the common attributes (C_A) of the corresponding case class (CC), while its attribute values are mapped from the retrieved similar case by the indexing model (C_M), or deducted from the generalized design knowledge (C_K). The case class is the abstraction of the previous cases in the case space. From an object-oriented viewpoint, every case is an instance of the case class, and design is thus a process to instantiate the case class as the design prototype and assign values to each attribute of this design prototype. The mapping between attributes and their values reflects the class-member relationship between a case class and the cases represented by this case class. A case class corresponds to many cases that possess common attributes, and vice versa. The generalized knowledge model deals with the expression of generalized knowledge used to analyze, evaluate, and modify the design prototype in order to satisfy current design requirements. The case indexing model (C_M) helps a designer quickly retrieve a similar case from a set of design cases represented by a case class. In a broader sense, C_M classifies the design cases according to the similarities between them in terms of their satisfaction with design requirements. Instead of searching the whole case space, C_M divides the case space into classes, thus reduces the searching space for similar case retrieval so that a higher retrieval efficiency is achieved.

2.4. Case class centered design process model. As illustrated in Fig. 2, the process of case class centered design involves mappings between the requirement space and the case space, and the case class. A case class retrieves the analogical cases from the case space, and maps to the design prototype instantiated by this case class. Then, the generalized knowledge model is invoked to analyze the design prototype. If the design prototype satisfies all design goals and constraints, it is treated as the final solution. If only part of design goals and constraints are met, then the design prototype needs to be modified recursively until all the requirements are resolved. In case no analogical case can be suggested, the generalized knowledge model is invoked to generate a new solution for the current design. The new solution is then stored in the case space as a new case. The case class will subsequently induce the case space with new cases so as to enhance its ability to case representation and design problem solving.

3. MEMORY ORGANIZATION SCHEMA

In CBED/MC, case memory is organized through introducing the concept of case class, which groups domain objects that share some aspects into conceptual classes. Thus the aforementioned comprehension problem is addressed by the appropriate classification rule sets, one for each conceptual class. The class-member relationships between a case class and its corresponding cases enable the design cases to be organized in an
object-oriented fashion. In CBED/MC, the memory model is divided into two components, viz., case class memory and design case memory. Conforming to PFA composition, CBED/MC employs two decomposition hierarchies, viz., a functional hierarchy and a structural hierarchy. The functional hierarchy reflects the front-end view of a product design, which is composed of a functional requirement (FR) topology. The terminological FR variables and the interrelationships among them are referred to as FR topology, which implicates the decomposition hierarchy of FR variables and the taxonomy of FRs [Tseng and Jiao, 1996]. An example of FR topology for power supply design is given in Table 1. At the back-end, a design is organized into a partonomic hierarchy in which each node is described by the most discriminating design parameters (DPs) and/or structural attributes. For example, the structural information on a power supply can be decomposed into a set of subcomponents as partially shown in Fig. 3. Using this approach to case memory representation, only those constraints that are specific to the context of the design situation are stored in the case, more general constraints are stored in the case class.

### 3.1. Case representation

CBED/MC stores designs in inheritance and partonomic hierarchies in the form of case class memory and case memory. The representation of the case class and the cases is based on the decomposition model and the object-oriented paradigm. Due to the hierarchical nature, the frame-based data structure with value inheritance is convenient to represent design information at both the component and system level. The abstract data type and class-instance inheritance supported by the object-oriented paradigm largely facilitates finding the interrelation in design information and storing different levels of generalized design knowledge. CBED/MC adopts a kind of object-oriented frame-based knowledge representation model [Jiao et al., 1994] to represent the case class. Frame trees are used to represent the decomposition structure of the case class. A frame is considered as an object, whose slots are used to describe attributes of the object. Meanwhile, the knowledge dealing with an object is packaged by the object body in the form of rules and/or data-driven methods so as to facilitate the management and maintenance of the knowledge. Being a member of the case class, each case has the same decomposition structure as that of its parent case class, while special values should be assigned to its attribute slots corresponding to the nodes in the partonomic decomposition tree.

### 3.2. A layered case indexing method

CBED/MC adopts a layered case indexing method, which consists of two sequenced sets of indices. namely case class indices (CCIM) and the case indices (CI). Thus the case indexing model (CCIM) can be expressed by CCIM = (CCRI, CI). Case class indices are constructed based on the functional decomposition hierarchy of PFA, i.e., FR classifications. They are intended to support various categories of case memory. Each category is indexed by FR classification hierarchy, which is developed by using a conceptual clustering technique [Tseng and Jiao, 1996b]. By case class indices, design cases are organized in various classifications from more general levels to more specific levels. Traversing the FR classification hierarchy to match the requirements of a new design problem leads directly to a list of relevant cases. As another sort of indexing representation, the case indices are extracted from the attribute nodes of the partonomic hierarchy of PFA, and used to access subcases of design cases so that they can be directly retrieved where appropriate.

### 3.3. Case retrieval through index elaboration

Case retrieval strategies for design problem solving must make explicit the most relevant match rather than the closest match [Maher, 1990]. This manifests the superficial similarity problem in CBR, as pointed out by Hua et al. [1992]. One solution to this problem is called index elaboration [Kolodner, 1984], which is a process that transforms the given specifications of a new problem into a more appropriate set of specifications. CBED/MC adopts a two-phase iterative process that makes use of index elaboration by the layered indexing method proposed above (Fig. 4). In order to achieve index elaboration, the case retriever performs incremental retrieval. In the first phase, a set of appropriate design case classes are retrieved using the case class indices, which include the initial specifications of a new design problem. The case

### Table 1. The FR topology for power supplies

<table>
<thead>
<tr>
<th>FR0: Universal low power AC/DC power supplies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DESCRIPTIVE LEVEL</strong></td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>FR1: Operating range</td>
</tr>
<tr>
<td>FR2: Power level</td>
</tr>
<tr>
<td>FR3: Operating condition</td>
</tr>
<tr>
<td>FR4: Mechanical requirement</td>
</tr>
</tbody>
</table>

*This table is truncated-given due to space limitations.*

![Fig. 3. A partonomic hierarchy for power supplies](image-url)
class retrieved via the initial specifications is used to identify more discriminating and critical features of the new design problem. In the second phase, the elaborated specifications or the salient features are used as case indices to search case memory in order to identify the most relevant cases. Correspondingly, case retrieval occurs at two levels in index elaboration. The first operates in the case class memory space and the second in the design case memory space. Once a case class is retrieved, it provides a space for elaborating the indices.

3.4. Case selection and similarity measure. The retriever typically finds more than one case or subcase. The purpose of the selector is to rank the set of retrieved cases to identify the most relevant case. In order to rank the retrieved cases, a measure of similarity must be developed. A weighted count of matching requirements or partial design solution attributes is used to provide a basis for ranking. The rank of an individual matching case is determined by assigning a numerical value to each feature and then producing a total ranking considering all features. This approach is widely deployed in CBR [Maher, 1990].

4. CASE ADAPTATION

Within PFA framework, CBED/MC has two ways in which adaptation knowledge can be formulated: (1) by categorizing the cases as instances of case classes which can simply be reinstantiated; (2) by providing specific adaptation knowledge which modifies aspects of cases instead of regenerating or reinstantiating them. Such adaptation knowledge could be stored with the cases in the case memory, and be kept separately from generalized domain knowledge stored in case classes. CBED/MC stores specific adaptation knowledge with each case, consisting of two parts: (1) configuration knowledge, expressed as constraints on the values of the parameters. Constraints can be defined, such as \( V_{out} = V_{in}(R_1 + R_2) \), or restrictions, such as \( V_{ass} > V_{mix}(V_{in} + N_{out}) \). Constraints are used to adapt the parameters of the retrieved case; (2) topological knowledge, expressed by a set of topological change rules specific to the case. A topological change rule might propose to change to another case class.

5. CONCLUDING REMARKS

As the natural technique to achieve mass customization, reuse can be best approached by CBR paradigm. PFA provides a unifying product platform to cover diverse customer needs while keeping the economy of scale. In CBED/MC, PFA plays an important role in acquiring highly-specialized domain knowledge and organizing previous design cases. With the introduction of the case class, case memory stores designs in inheritance and partonomic hierarchies, which enables cases to be represented in an object-oriented fashion. Case classes conform to the product classifications in PFA, while cases correspond to specific products. This class-member relationships facilitate the layered indexing and retrieval by index elaboration. The adaptation problem can thus be solved by opportune traversals through the set of conceptual classes. Based on the methodology presented, future efforts will be geared towards implementing a CBED/MC system.

Acknowledgments: This work is partially supported by Computer Products Asia-Pacific Ltd. (Power Conversion, Hong Kong) under grant CPI 95/96.EG01, the HKUST Research Infrastructure Grant (RI 93/94 EG08), and Hong Kong Research Grant Council (HKUST 566/94E).

REFERENCES


