Proceedings of the ASME 2011 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2011
August 28-31, 2011, Washington, DC, USA

DETC2011-48263

ENVIRONMENT-BASED DESIGN (EBD)

Yong Zeng
Concordia Institute for Information Systems Engineering
Concordia University
1455 de Maisonneuve Blvd. West, EV07.633
Montreal, Quebec, Canada H3G 1M8
E-mail: zeng@encs.concordia.ca

ABSTRACT

This paper summarizes a design methodology - Environment Based Design (EBD) - developed over the last two decades. The EBD stems from the observation that design aims to change an existing environment to a desired one by generating a new artefact. Design starts from the environment, functions for the environment, and brings changes to the environment. This environment changing process implies the recursive evolution of design problems, design knowledge, and design solutions. Three basic activities are included in the EBD: environment analysis, conflict identification, and solution generation. In introducing the EBD, four major requirements for an effective design methodology are firstly formulated from the perspective of the nature of design, cognitive model of design, and the driving force underlying design. The mathematical, semantic and algorithmic foundations of the EBD are then presented to define methods and procedures for solving a design problem. Experimental validation and industrial applications are summarized to show the effectiveness of the EBD. Future research questions are also given in the paper.

1 Introduction

Intuitively, design is an activity that aims to change an existing environment to a desired one by creating a new artefact into the existing environment. The artefact must adapt to the goals and requirements of humans while obeying laws and rules existing in the environment from which it can never be separated [1].

Therefore, design is driven by a need or an inspiration from the existing environment (human, natural, and built). The environment was there, is there, and will always be there. Any design action changes and only changes the environment. Design starts from the environment, functions for the environment, and brings changes to the environment.

Environment-Based Design (EBD) is such a methodology that provides step-by-step procedures to guide a designer in this environment changing process. The methodology includes three interdependent activities: environment analysis, conflict identification, and solution generation. Throughout the entire design process, any newly generated design solution will be viewed as an environment component for the succeeding design activities. The design process continues until no more undesired conflict exists in the environment. A complex design problem can be decomposed through the partitioning of the environment implied in the problem description [2].

The rest of this paper is organized as follows: Section 2 formulates the major requirements for an effective design methodology, followed by Section 3 which introduces the fundamental methods included in EBD. Section 4 summarizes experimental validation and industrial applications of EBD. Section 5 concludes the paper and lists open questions for the future research.

2 Requirements on Design Methodology

Design methodology has experienced great evolution since it became a subject of academic research in the 1960's. In the early stages, influenced by the emergence of systems engineering, there was a design method movement mainly in Europe aiming to divide design activities into components that are connected logically such that design tasks can be conducted in an orderly manner [3].

Compared to the long history of design practices, the study of design theory as a scientific discipline is quite young. This study is becoming more and more important because of an increasing need of "Best Design Practice" in optimizing the available yet limited resources for the benefit of mankind. A good design cannot be achieved unless designers are armed with both profound design knowledge and a sound design methodology (explicit or implicit). As a result, a variety of design theories and methodologies have been proposed in the last several decades, such as the systematic design methodology [4, 5], decision-based design [6], theory of inventive problem-solving (TRIZ) [7], axiomatic design [8], general design theory (GDT) [9, 10], formal design theory [11], exploration based design [12, 13], total design [14], adaptable design [15], function based design [16-20], affordance based design [21], and design structure matrix (DSM) [22]. However, based on an analysis of existing design methodologies, Tomiyama concluded that most of existing abstract design methodologies have not found wide applications in industry [23].

This section will discuss the requirements for an effective design methodology, which will be critical for the industrial applications. To put it in one sentence, a design methodology can be seen as a system of procedures to support designers in design activities. Therefore, in developing a design methodology, the following issues must be taken into account:

1) What is the nature of design?
2) How does a design methodology support designers in the design activities?
3) What is the driving force behind design activities?

2.1 Nature of design problem: recursion

As was indicated by Zeng and Cheng in 1991 [1], “design is largely different from deduction, induction and abduction in that the conclusion [design solution or concept]1 of the reasoning is recursively dependent on the major premise [design knowledge] of the reasoning.” “Design is [thus] a process to simultaneously produce both the artefact [design solution or concept] and its behaviour system [design knowledge].” Therefore, the logic of design is considered to be recursive, and thus addresses “{given the environment and the function of a system}2 the reasoning process to construct the analytic reasoning of the system [design knowledge], i.e. to find the form [design solution or concept] which dominates the whole system”. This logic form was later confirmed and discussed further by Roozenburg [24].

The recursive logic of design implies that during the design process the generation and evaluation of design solutions depend on design knowledge while the kind of design knowledge that can be used for the current design is determined by the design solutions. The design research community has now generally accepted this so called co-evolutionary nature of the design process in which design problems, design solutions, and design knowledge are updated simultaneously. This can be found in an experimental validation by Dorst and Cross [25], in a computational model by Zeng and Jing [26] and by Maher and Tang [27], in a formal mathematical approach by Zeng and Gu [28], and lately in a form of design theory by Hatchuel and Weil [29]. This recursive nature of design also interprets the ill-structuredness of the design problem [30, 31].

The recursive logic of design was represented in an equation as follows [26]:

$$S = K_s\big(K_e(S)\big), \qquad (1)$$

where S is the design solution whereas K_s and K_e are synthesis and evaluation knowledge, respectively.

Along with Eq. (1), the recursive logic also implies another equation [32]:

$$R_d = K_e(K_s(R_d)), \qquad (2)$$

where R_d is the design requirement. Design requirements R_d is defined by $K_e(S)$, which is dynamically determined by S during the design process. This was represented in the following equation [28]:

$$K_e = [S \cup R_d,\ L]_{\approx}, \qquad (3)$$

where L is a collection of laws [knowledge] in the product’s working environment. $[S \cup R_d,\ L]_{\approx}$ is a collection of knowledge in L that is associated with $S \cup R_d$.

The three equations above were illustrated in Figure 1 and Figure 2, respectively. Figure 1 shows that design solutions (Eq. (1)), design problems (Eq. (2)), and design knowledge (Eq. (3)) change simultaneously and interdependently in the design process. Those three items constitute critical parts of the design space.

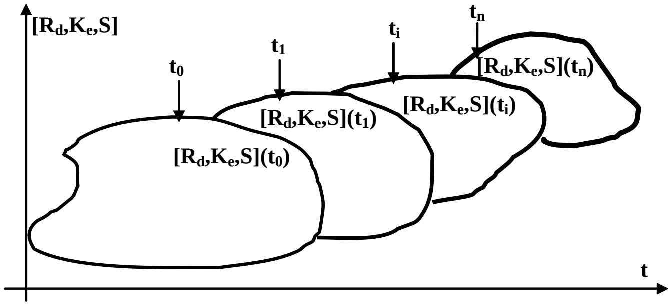

Figure 1 Co-evolution of design problems, design solutions, and design knowledge [33]

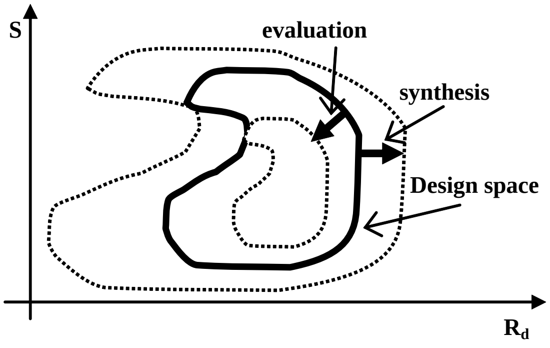

Figure 2 Design space under synthesis and evaluation [28]

Figure 2 reflects both Eq. (1) and (2), where design synthesis stretches the design space whereas design evaluation shrinks the design space. The interaction of both leads to the final design solutions.

[1] The text in [] is added by the author to relate the terminology to the contemporary research context.

[2] The brackets {} are added to the original text to keep the quotation smoothly related to its context.

As a matter of the fact, all of the existing design methodologies aim to solve the Eq. (1), (2), and (3) under different assumptions. *An effective design methodology should help designers jump out of the recursive loop between design problems, design knowledge, and design solutions*.

2.2 Support of design activities: creativity

Due to the difficulties resulting from the recursive nature of design, effective design methodologies are indispensable for guiding designers, particularly new designers, in dealing with the complexity appearing in their design activities.

There are two great challenges faced by the effort to develop effective design methodologies. The first one lies in the contradiction between the determinism in the methodology and the randomness in creativity. In other words, how can a design methodology, which is fundamentally based on logic, lead to creative solutions, which are random and unpredictable? Zeng [32, 34] pointed out that the recursive logic of design, as reflected in Eq. (1), (2), and (3), implies a nonlinear dynamic mechanism, which can be chaotic under certain conditions. As a result, design solutions can be highly sensitive to its initial conditions. Furthermore, due to the recursive logic again, the initial condition – the design problem – may keep changing according to the previously generated design solution. Therefore, design solutions can be highly unpredictable because of the constant change of the initial conditions rooted in the recursive logic of design. The following lists three basic paths that may change the initial conditions of a design process:

1) Formulating the design problem differently,
2) Extending synthesis knowledge, and
3) Changing the sequence of environment decomposition.

A computer simulation was used to show quantitatively how the three paths above may contribute to the generation of significantly different design solutions [35].

Therefore, an effective design methodology should lead to both routine and creative design naturally.

The second challenge in developing an effective design methodology lies in two contrasting facts. On the one hand, a natural design process is one in which designers can flexibly and freely explore various avenues to achieve design goals [25, 36-38]. On the other hand, any design methodology implies a set of well structured steps for solving a design problem. This contradiction between the natural design process and the structure embedded in a design methodology triggers mental stresses on designers when applying the design methodology during the design process.

To develop a more effective design methodology, we have adopted the Yerkes-Dodson (Y-D) law [39] to design creativity [40]. The Yerkes-Dodson (Y-D) law shows that there is an inverse U-shape correlation between arousal and performance. According to the Y-D law, a medium level of arousal results in the best performance. In the context of innovative and creative design, we have made the following hypothesis:

Design creativity may happen when a designer is in a state of medium level of arousal/stress.

By applying the hypothesis above to design methodologies, it can be assumed *that an effective design methodology should help a designer maintain his/her mental stress at an optimal level during the design process*, as shown in Figure 3.

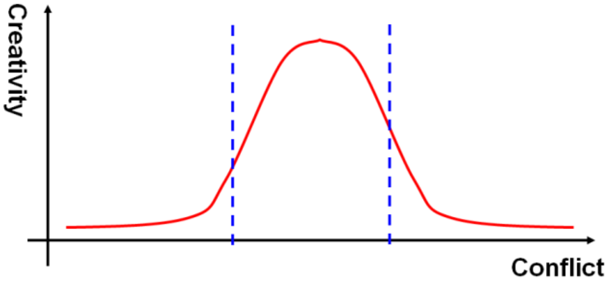

Figure 3 Design creativity vs. Conflict between design methodology and the natural design process

Therefore, in order for a design methodology to be useful and applicable to practices of innovative and creative design, the methodology must lead designers naturally to a medium arousal zone by taking into account the designer's cognitive model.

The author and his research group has been studying the quantification of the designer's cognitive processes, particularly the designer's mental stress, though linguistic information [40], body language [41], as well as eye movement and EEG systems [42, 43].

2.3 Driving force behind design: conflict

From the engineering point of view, the driving force underlying the design evolution is critical for an effective design methodology. The driving force determines the direction of the design activities and when the design is completed.

The Theory of Inventive Problem Solving (TRIZ), developed by Altshuller and his school in the former USSR, indicates that two factors, which are ideation and contradiction, drive design - the evolution of technical systems [7, 44]. It is a human-oriented knowledge-based systematic methodology of inventive problem solving, which includes a series of powerful tools to detect the search for solutions to engineering problems [45]. TRIZ relies on deep knowledge and provides possibilities for effective solutions of difficult problems. The central concept in TRIZ is contradiction.

The axiomatic design theory provides general principles for a good design, which must follow two axioms – axiom of independence and axiom of information. The two axioms indeed define the ideation for design.

The EBD takes the undesired conflicts in the existing environment as the driving force of design. When there is no undesired conflict, the design activity stops [34].

Taking conflict as the driving forces behind design evolution may seem to be exclusive of many existing design methodologies. For example, human centered design takes the needs of the end user as the driving force whereas function based design uses function as the fundamental driver of design.

However, in EBD, there are human, natural, and built environments. Human needs come from conflicts between human and the rest of environment. Function is indeed a solution to resolve a conflict.

The driving force is the condition for the evolution of design, which starts, sustains, and stops the design process.

2.4 Summary

In summary, an effective design methodology should satisfy the following requirements:

1) to help designers jump out of the recursive loop between design problems, design knowledge, and design solutions,
2) to lead to both routine and creative design naturally,
3) to help a designer maintain his/her mental stress at an optimal level during the design process, and
4) to include naturally the conditions for the evolution of the design.

3 EBD Methodology and its Fundamentals

The development of the EBD methodology has gone through four phases. The first phase was the discovery of the recursive logic of design in 1991 [1]. Design reasoning was defined as a recursive one that generates simultaneously and recursively design problems, design solutions, and design knowledge starting from the environment of an artefact. This recursive logic of design was applied in an automatic finite element mesh generation algorithm [46].

The second phase was marked by a formal mathematical model of design [26, 28, 33], where the recursive logic of design was formalized in a set-theoretic formalism [47, 48]. The design governing equation was proposed to reflect the recursive dynamic mechanism of design. A formal descriptive model of design was developed, which describes how design requirements are transformed into design concepts through design knowledge, how the requirements are redefined by the generated design concepts, how the design knowledge is redefined by the design concepts and the reformulated design requirements. The design process is formalized into one that finds the fixed points for the design governing equation in the design space defined by design requirements, design concepts, and design knowledge.

The third phase is based on the definition of design problems, design solutions, and design knowledge in the structure of product environment [2, 32, 34, 49, 50], with the support of a new mathematical formalism [49] for modeling design activities. This new formalism naturally integrates all the concepts that represent design into the structure of environment. Those concepts include design requirements, product functions, product behaviours, design solutions, etc [51]. As a result, the design process is mathematically formulated into an environment evolution process.

Based on the descriptive design model developed in the first three phases, the EBD has been made a prescriptive design model through a few critical methods. The first one is the Recursive Object Model (ROM) [52] which represents a text in a graph that has three types of relations: constraint, connection, and predicate. The second is a question asking algorithm based on ROM for eliciting product requirements [53]. The third is the structure of design conflict [54]. Others include algorithms for conflict identification and resolution as well as for environment decomposition.

The rest of this section will put those fundamentals in the context of EBD methodology.

3.1 EBD: mathematical formulation

The recursive logic of design implies a fundamental question to any design methodology: function (requirements) first or form (solution) first? This “chicken-egg” like problem leads to a great difficulty for all the function-based design approaches – where is the functional structure without having a solution in mind? [2].

A fundamental concept in EBD is that function, form, requirements, solutions, knowledge, and other relevant design information are all present in each and every state of design. They trigger each other’s evolution during the design process. Mathematically, a challenge is how to embody all of those types of heterogeneous information in a homogeneous mathematical representation? To this end, the author has developed a new mathematical representation in the Axiomatic Theory of Design Modeling [49].

Mathematically, Axiomatic Theory of Design Modeling (ATDM) is different from set theory in that there is no structural hierarchy; hence, membership operation ($\in$) does not exist anymore. Following this theory, a formal model of design could be derived to represent the syntactic structure of hierarchical evolving design objects and the dynamic design process. A major operation in ATDM is structure operation, denoted by $\oplus$, which is defined as the union ($\cup$) of an object O and the interaction ($\otimes$) of the object with itself.

$$\oplus\, O = O \cup (O \otimes O), \qquad (4)$$

where $\oplus$O is the structure of the object O. Everything in the universe can be seen as an object. Interactions between objects are also objects. Examples of interaction include force, movement, and system input and output. Structure operation provides a means to represent a hierarchical system with a single mathematical expression.

Due to the capacity of human cognition and the scope of an application, a group of primitive objects can always be defined as objects that cannot or need not to be further decomposed [49, 52]:

$$\oplus O_i^a = O_i^a, \exists n, i = 1, \ldots, n. \qquad (5)$$

where O_i^a is a primitive object whereas n is the number of primitive objects.

As was discussed earlier in the paper, in the design process, any previously generated design concept can be treated as an environment component for the succeeding design. As a result, a new state of design can be defined as the structure of the old environment (E_i) and the newly generated design concept (S_i), which is a partial design solution.

$$\oplus\, E_{i+1} = \oplus\, (E_i \cup S_i). \qquad (6)$$

The environment can be divided into three kinds: natural, built, and human [34]. Built environments are the artefacts designed and created by human beings whereas the human environment includes all of the human beings but particularly the human users of an artefact. Starting from a textual definition of a design problem, it was derived that the structure of the design problem can be represented in Table 1 [34], where S is the product; E is the product environment; λ is a constraint relation; all the B's constitute the boundary between the product and its environment. The boundary B has two kinds: structural and causes/effects.

Table 1 Structure of design problems [34]

Design problem: P^d	
Product Environment	E
Performance Requirements	$\lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r)$
Structural Requirements	$\lambda(\oplus S_0, \oplus S_s) \wedge \lambda(B_0^s, B_s^s)$

Since the boundary B is a relation between the product S and its environment E and the product is not known in advance in design, the design problem is defined by the only known information – environment E.

The environment structure, which is $\oplus E_i$, includes the description of the design solution at design stage i, the design requirements for the design stage i+1, the relevant design knowledge, and other design information [34, 51]. As a result, Figure 1 can be replaced by Figure 4, which is more mathematically neat and intuitively meaningful.

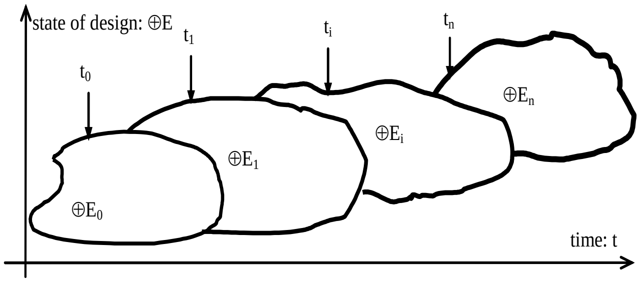

Figure 4 EBD: evolution of environment [34]

Following Axiomatic Theory of Design Modeling, the EBD methodology can be derived [49], which includes three activities – environment analysis, conflict identification, and solution generation as follows:

- Environment analysis: define the current environment system $\oplus E_i$.

$$\oplus E_i = \oplus \left(\bigcup_{j=1}^{n_e} E_{ij} \right)$$

$$\oplus E_i = \left(\cup_{j=1}^{n_e} (\oplus E_{ij}) \right) \cup \left(\cup_{j_1=1}^{n_e} \cup_{\substack{j_2=2 \\ j_2 \neq j_1}}^{n_e} \left(E_{ij_1} \otimes E_{ij_2} \right) \right) \qquad (7)$$

where n_e is the number of components included in the environment E_i at the i^{th} design state; E_{ij} is an environment component at the same design state. It should be noted that decisions on how many (n_e) and what environment components (E_{ij}) are included in E_i depend on designer's experience and other factors relevant to the concerned design problem. A roadmap is provided to facilitate this process in [55].

- Conflict identification: identify undesired conflicts C_i between environment relationships by using evaluation operator K_i^e.

$$C_i \subset K_i^e (\cup_{j_1=1}^{n_e} \cup_{\substack{j_2=1 \\ j_2 \neq j_1}}^{n_e} \left(E_{ij_1} \otimes E_{ij_2} \right). \qquad (8)$$

- Solution generation: generate a design solution s_i by resolving a group of chosen conflicts through a synthesis operator K_i^s. The generated solution becomes a part of the new product environment for the succeeding design.

$$\exists C_{ik} \subset C_i, K_i^s : c_{ik} \rightarrow s_i, \oplus E_{i+1} = \oplus (E_i \cup s_i). \qquad (9)$$

As illustrated in Figure 5, the three activities work together to update environment and its internal relationships to solve a design problem. The design process continues with new environment analysis until no more undesired conflicts exist, i.e., $C_i = \Phi$.

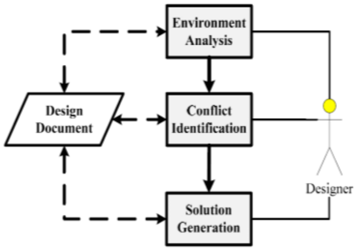

Figure 5 EBD: process model

Since environment is the source of design requirements, design knowledge, and design solutions, EBD provides an effective approach for the designers to jump out of the recursive loop between them. Hence, the first requirement for an effective design methodology is satisfied in EBD.

Furthermore, as was validated in [35], creative design may happen naturally when designers follow the EBD, which meets the second requirement for EBD to be an effective design requirement.

Finally, since the EBD process is driven by undesired conflicts in the current environment structure, the condition for the evolution of design is naturally included. This shows the fourth requirement for an effective design methodology holds for EBD.

It must be noted that though TRIZ and EBD both take conflict as a central concept, they are two different methodologies. The difference goes along the entire design process. TRIZ includes three kinds of contradictions (conflicts): administrative, technical, and physical. Administrative contradiction is out of the scope of the TRIZ methodology [7]. TRIZ bases its technical contradiction on 39 engineering parameters; as a result, it cannot deal with software design, policy design, and many other non-engineering problems. Furthermore, TRIZ does not have systematic method to identify contradictions, which demands highly of the engineer's expertise in TRIZ methodology. Finally, TRIZ is empirically developed without a formal and solid mathematical foundation, which limits its application to single or low number of contradictions. EBD, however, is a general design methodology that can be applied to any design problem. Conflict identification and resolution follows well defined procedures, as will be shown in the next few sections.

Nevertheless, the fact that TRIZ was generalized from practices and the EBD is derived mathematically strengthens the thesis that design is driven by conflict. The two methodologies can learn from each other and benefit each other.

3.2 EBD: semantic foundation

Corresponding to Figure 1 and Figure 4, the EBD process can also be illustrated in Figure 6. On the one hand, a designer may have to remove some information from the current design state in order to identify the real intent behind the current design problem. On the other hand, the designer may also add information to complete the design state so that the existing undesired conflicts can be properly eliminated.

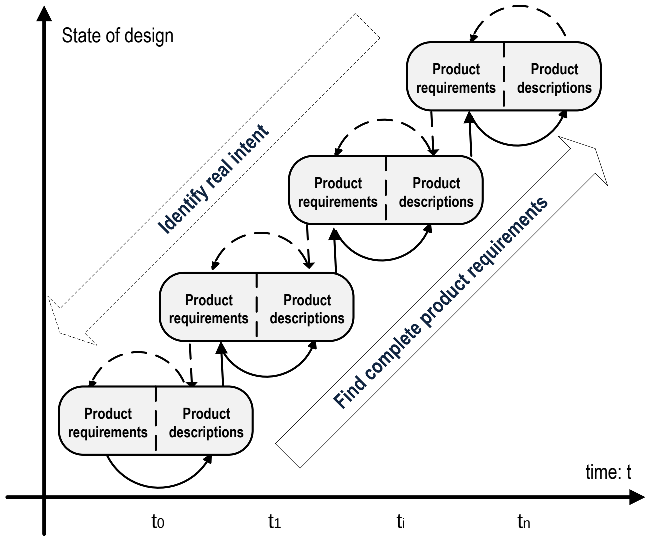

Figure 6 Evolution of the design process [53]

Semantics matter throughout the entire design process. Firstly, a designer must understand the design problem. In EBD, this is through environment analysis and conflict identification. Secondly, when one creates a design through solution generation, a meaning is indeed assigned to the artefact that he/she creates in the context/environment of the created artefact. Finally, one must interpret the meaning of the design to the artefact's environment. This interpretation may be involved in the protection of intellectual properties implied in the design. This interpretation is also important in training the users of the artefact. Furthermore, this interpretation may mean that the artefact will carry out a function that it was not intended for by its original designer(s). In this semantics centered environment based design process, if one design carries a significantly surprising new meaning, then this design is called creative. The importance of semantics in design can be seen from [56-58].

In order to support the processing of semantics in design, Zeng proposed a Recursive Object Model (ROM) [52], which also echoes the recursive logic of design [1]. ROM provides the foundation of representation for EBD whereas the recursive logic is the backbone of design reasoning. Though ROM was originally developed to deal with linguistic information, it is being extended to process other design information as well.

Table 2 Recursive Object Model (ROM) [52]

Type		Symbol	Description
Object	Primitive Object	$\boxed{O}$	Everything in the universe
	Compound Object	$\boxed{\boxed{O}}$	An object that includes at least 2 other objects
Relations	Constraint Relation	●─ $\boxed{\xi}$ →	A descriptive or limiting relation
	Connection Relation	- - $\boxed{\iota}$ - ->	To connect two objects that do not constrain each other
	Predicate Relation	─ $\boxed{\rho}$ →	An object's action on the other or an object's states.

ROM can be seen as a refined representation of environment structure by including three types of interaction operation $\otimes$: constraint, predicate and connection. The ROM includes two kinds of objects, which are primitive and compound objects. Table 2 shows the graphic symbols in the ROM.

The following subsections will introduce ROM-based approaches for each of its three activities in the EBD.

3.3 EBD: algorithmic foundation

3.3.1 Environment analysis

The objective of environment analysis is to identify the environment in which the desired product is to work. According to the EBD, the environment includes its components and the relationships between those components. To move in either of the two directions as shown in Figure 6, a designer must ask the right questions [53]. Eris also highlighted the importance of the right question in design by posing the following questions: "1) Can an effective decision making process be performed without having high quality information? 2) Can high quality information be acquired and generated without performing an effective inquiry process?" [57].

In environment analysis, two kinds of questions (inquiries) will be asked. The first is generic questions for the clarification and extension of the meaning of the design problem whereas the second is domain specific questions for implicit design information related to the current problem. The ROM, as a linguistic tool in design, is used as the foundation for generating the questions.

Generic questions

The generic questions aim to help designers better understand the design problem through linguistic analysis using ROM. Table 3 lists the procedures for generic question asking.

Table 3 Procedure for generic question asking

Step 1: Generate the ROM diagram for the design problem.
Step 2: Ask a question using the rules given in Table 4 and templates in Table 5.
Step 3: Find answers to the question.
Step 4: Generate the ROM diagram for the answer and merge it back to the original ROM diagram.
Step 5: Repeat Step 1-4 until no more questions.

Table 4 Rules for generic questions [53]

Rule 1: Before an object can be further defined, the objects constraining them should be further refined.
Rule 2: An object with the most undefined constraints should be considered first.

Domain specific questions

The objective of asking domain specific questions is to collect the information that would have significant influence on the design problem. The information will be collected without imposing any assumptions on the structure of the final design solutions. Therefore, collected information is composed by the domain relevant environment components and their relationships, which can be defined without the knowledge about design requirements and design solutions.

Table 5 Templates for generic questions [53]

	Conditions	Question
T1	For a concrete, proper, or abstract noun N	What is N?
T2	For a noun naming a quantity Q of an object N, such as height, width, length, capacity, and level	How many / much / long / big / … is the Q of N?
T3	For a verb V	How to V? Or Why V?
T4	For a modifier M of a verb V	Why V M?
T5	For an adjective or an adverb A	What do you mean by A?
T6	For a relation R that misses related objects	What (who) R (the given object)? Or (the given object) R what (whom)?

As a generic design methodology, no domain specific template will be required. Instead, a roadmap is proposed as guidance to facilitate the identification of complete environment components and their relations for the concerned design stage [53, 55]. This roadmap categorizes product environment in terms of two criteria. One criterion partitions product environment based on the product lifecycle whereas the other classifies the product environment into natural, built, and human. Figure 7 illustrates a case where there are seven events in the product lifecycle. For each event in the lifecycle, the environment components can be classified into a pyramid with the natural environment in the base and the human environment at the top.

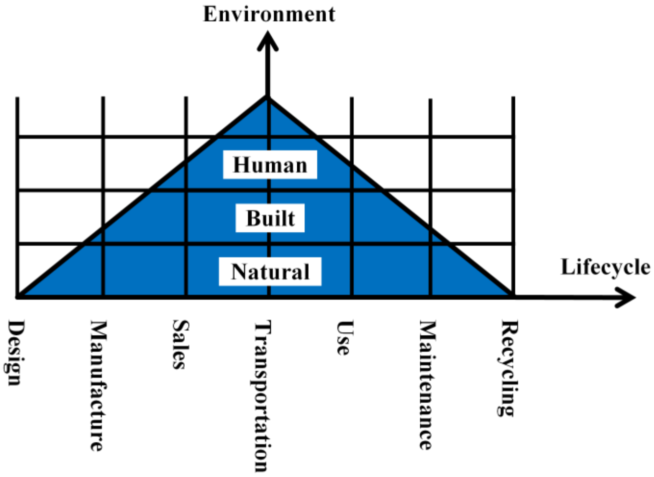

Figure 7 Roadmap for domain related environment: an example

Corresponding to the roadmap as shown in Figure 7, Table 6 gives the procedure for identifying environment components related to the concerned domain.

Table 6 Procedure for asking domain specific questions

Step 1: Ask and answer the question: what is the lifecycle of the product to be designed?
Step 2: For each event included in the lifecycle, ask and answer the question: what are the relevant components for natural, built, and human environments for this event?
Step 3: Generate the ROM diagram for each answer and merge them back to the original ROM diagram.
Step 4: Apply the procedure for generic question asking.

Through environment analysis, a complex ROM diagram will usually be generated, which is a graphic representation of Eq. (7). Figure 8 shows such an example for a new medical device design [59].

The rules for generating ROM diagrams from text can be found from [52]. Methods and algorithms to extract semantics from ROM diagrams can be seen from [60, 61].

3.3.2 Conflict identification

Contradiction, which can be seen as conflict, is a central concept in TRIZ methodology [7]. Three conflicts (contradictions) were identified: administrative, technical, and physical. Administrative contradiction is a situation where “something is required to make or receive some result to avoid the undesirable phenomenon, but it is not known how to achieve the result.” “The administrative contradiction itself is provisional, has no heuristic vlue, and does not show a direction to the answer.” Technical contradiction, which represents a conflict between two subsystems, is a situation where an action is simultaneously useful and harmful. “Physical contradiction is a situation where a given subsystem should have property A to execute a necessary function and property non-A or anti-A to satisfy the conditions of a problem.” [44] Different approaches are adopted in TRIZ for different kind of contradictions, most of which are for engineering domain due to its dependency on engineering parameters.

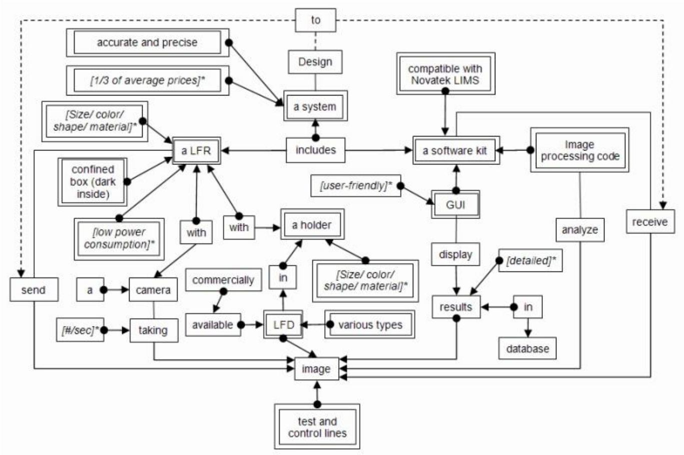

Figure 8 ROM diagram for a medical device design [59]

Conflicts are also considered as the driving force in the EBD process. By proper identification of all those conflicts, a designer will be able to develop satisfactory solutions for the design problem. A conflict is composed of three basic elements: two competing objects and one resource object that the former two objects contend for [54]. Figure 9 shows the structure of a conflict.

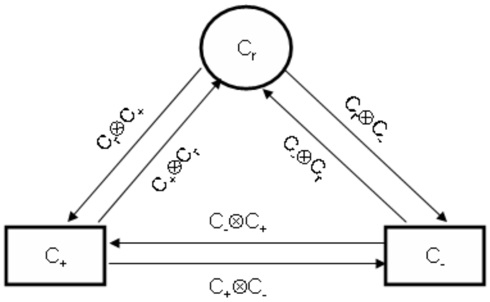

Figure 9 Structure of design conflict [54]

From the developed ROM diagram, there will be conflicts or potential conflicts between the relationships, according to Eq. (8). Currently, we have developed three rules for identifying potential conflicts from a ROM diagram, as shown in Figure 10 and Table 7. Application of those rules will result in a structure of conflict as shown in Figure 9.

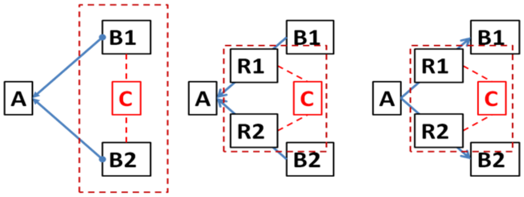

Figure 10 Three rules for conflict identification from a ROM diagram

Table 7 Rules for identifying potential conflicts

Rule 1:	If an object has multiple constraints, then potential conflict exists between any pair of constraining objects.
Rule 2:	If an object has multiple predicate relations from other objects, then potential conflict exists between a pair of those predicate relations.
Rule 3:	If an object has multiple predicate relations to other objects, then potential conflict exists between a pair of those predicate relations.

Design conflicts can be identified by iteratively applying the three rules to the ROM diagram for a design problem together with the semantics of the constraints or predicates. However, it must be noted that the rules listed here are not inclusive and complete. Experiments and investigations are required to develop more robust rules.

It can be seen that the conflicts used in EBD differ from those in TRIZ in that only physical conflict will be left for solution generation. Environment analysis process could transform administrative contradiction into technical ones which are in turn located as physical ones from the ROM diagram.

3.3.3 Solution generation

In generating design solutions, a design process was divided into two parts: atomic design and the recursive resolution of a complex design problem [1]. Atomic design is where knowledge is available or can be systematically discovered to resolve a conflict. Hence, it depends on the designer’s experience and background. Obviously, an experienced designer can have a more complex atomic design than a novice one may have. Recursive resolution of a complex design problem can be conducted through environment decomposition, which will identify the key conflict to start with.

Due to the limitation of space, we will briefly summarize the concepts in solution generation. The details will be provided in a separate paper.

Atomic design

Atomic design is supported by a set of atomic requirements, atomic solutions, and atomic design knowledge [26]. Indeed, these atomic objects are a part of the subject of study of the ontological basis of engineering. The atomic design problem can be solved by applying an exploration based model proposed by Smither [12, 13]. Design creativity may occur when the set of atomic design knowledge is extended [32, 35, 62].

The input for the atomic design process is a design conflict, which is equivalent of physical contradiction in TRIZ. Therefore, the four separation heuristics from TRIZ can be applied: separation in space, separation in time, separation upon condition, and separation between parts and whole [7]. In addition, new components can be added as an approach to addressing the separation upon condition.

The strategy above will result in one of the two situations: 1) adjustment of existing design parameters need to be optimized; and 2) new component needed to connect unsolved/connected physical effects in the ROM diagram. For the details, please refer to Section 4.1 in [28]: primitive design.

It must be noted, however, the methods developed so far are all heuristic. It is the author's belief that this is one of the critical stages for design creativity to happen. One advantage of the EBD is to guide designers to rapidly identify this focus so that creative solutions may be generated.

Recursive resolution

The core of the recursive resolution strategy consists of a requirement decomposition operator and structure recomposition operator. The recomposition operator embodies the approaches to dealing with the conflicts between the newly generated solution and the existing solutions and problems, which leads to the reformulation of the original design problem [26]. Those operators were further refined in [28], based on which an environment decomposition based method was proposed [2]. The environment decomposition process is updated in Table 8 by using the ROM.

Table 8 Procedure for solution generation based on environment decomposition

Step 1:	Find the critical environment component from the ROM diagram
Step 2:	Resolve the conflict associated with the critical environment component by applying the atomic design process
Step 3:	Update the ROM diagram
Step 4:	Repeat the EBD process with the updated ROM diagram

In implementing the procedure in Table 8, the following question must be firstly answered: where to extract the element? The procedure given in Table 9 addresses this issue.

Table 9 Procedure for finding the critical environment component

Step 1:	For each conflict object, if it is not semantically constrained by another conflict object in the ROM diagram, then take it as a candidate starting environment component.
Step 2:	For all the candidate conflict objects, calculate the number of semantic constraints on that object.
Step 3:	Choose the candidate object with the greatest number of semantic constraints as the starting environment.

3.4 Summary

In summary, the EBD provides designers a sense of direction by guiding them collect the necessary and sufficient information for a design task, by supporting them determine the focus at each stage of design, by helping them decomposing a complex problem into atomic ones, and by investigating potential solutions for each atomic design problem. This could help designers manage their mental stress in solving a design problem, which increases their chances to take advantage of their creative potentials. This qualitatively shows that EBD could be an effective approach with respect to the third requirement for effective design methodology.

4 Validation of EBD

A major problem faced by all the design methodologies is how to validate them. Experimental validation and case studies, are two most widely accepted approaches for testing a design methodology [63]. Successful industrial applications are also good support for an effective design methodology.

This section summarizes a few of such validation of the EBD.

4.1 Industrial applications

4.1.1 Design of an automatic finite element mesh generation algorithm

This problem is extracted from two industrial projects: one for radar structure analysis and the other for sewage system design. The goal of this problem is to generate a two-dimensional quadrilateral mesh from a predefined piece-wise boundary, satisfying certain requirements and following the element extraction method as shown in Figure 11 [46, 64].

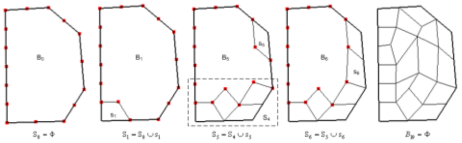

Figure 11 Quadrilateral mesh generation by element extraction [64]

The element extraction method generates the finite element mesh element by element along the boundary of the domain. Each element will be generated on the basis of one boundary segment. Three basic element extraction rules can be defined to cover all the possible situations [46]: 1) adding three new edges; 2) adding two new edges; and 3) adding one new edge. They are shown in Figure 12.

The best element is a square; therefore, the elements extracted during the mesh generation process should resemble a square as much as possible. However, if an element is generated with the best possible quality, then the quality of the future elements will deteriorate in most of cases. This is a major conflict in quantifying the three rules – under which conditions should each rule be triggered and how to determine the parameters for generating an element. In order to generate good enough elements that also leave room for the future elements, robust element extraction rules must be designed through experiments. An ANN based approach was thus developed for this purpose [64].

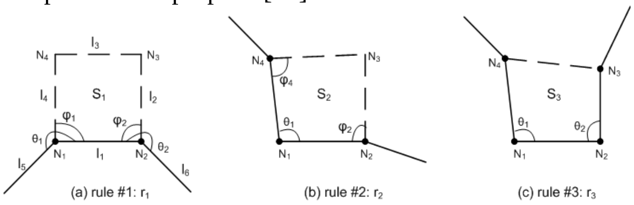

Figure 12 Three basic element extraction rules [46]

This problem was used as a platform for computer simulation of design, where routes leading to new design were quantitatively studied [35].

4.1.2 Design of a vision based curve reconstruction algorithm

The objective of the problem is to connect a set of unlabelled points into curves that look natural to human vision. An example is shown in Figure 13.

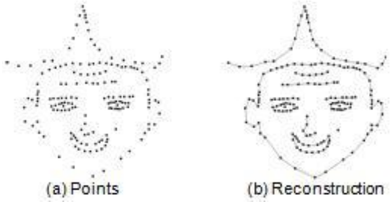

Figure 13 Curve reconstruction problem [65]

Through environment analysis, we identified the two criteria for connecting the points: 1) human eyes tend to connect closest points, and 2) human eyes tend to connect points into smooth curve(s). These two rules are shown in Figure 14.

The conflict happens when the connection of two nearest points does not result in a smooth curve or when the smooth connection of two points results in visually unnatural curves. This is shown in Figure 15.

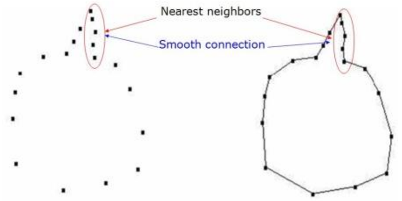

Figure 14 Two rules for curve reconstruction

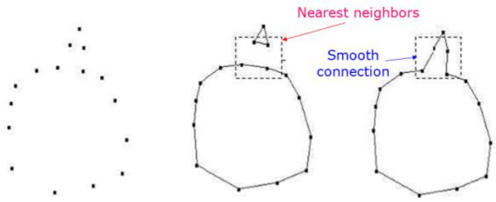

Figure 15 Conflict in curve reconstruction

This conflict between nearness and smoothness was solved from two approaches. The first approach considered only the nearness property, which means that any connection between closest points will satisfy the smoothness property. The proposed approach works perfectly if the points meet the necessary and sufficient sampling conditions [65].

The second approach integrated both criteria and resolves the conflicts by setting priority to either rule with a user-defined parameter according to human visual perception [66].

4.1.3 Design of PLM process

In this project, a few aerospace companies wanted to have a new design chain management (DCM) process model that can support the secure collaboration between partners. In developing the DCM model, we started from an informal definition of DCM given by Twigg [67]. Following the EBD methodology, an environment analysis of the definition leads to a wheel model of PLM system as shown in Figure 16, which was further refined into a model for DCM in Figure 17 [68]. The recursiveness of the development can be seen from the model.

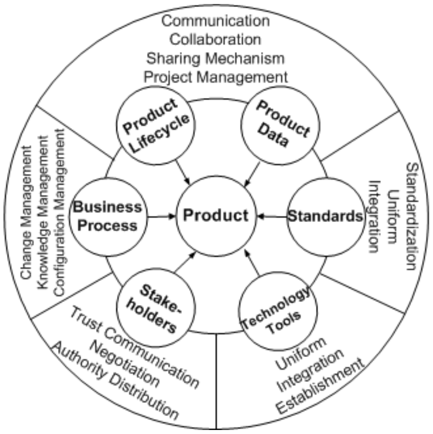

Figure 16 PLM Wheel Model [68]

By refining the relationship between product data and product lifecycle, it is revealed that the collaboration between suppliers and the developer needs data sharing while the developer may want to maintain its competitive edge by securing some key product information. This issue of information protection was transformed into a conflict defined by Rule 2 in Table 7, as shown in Figure 18.

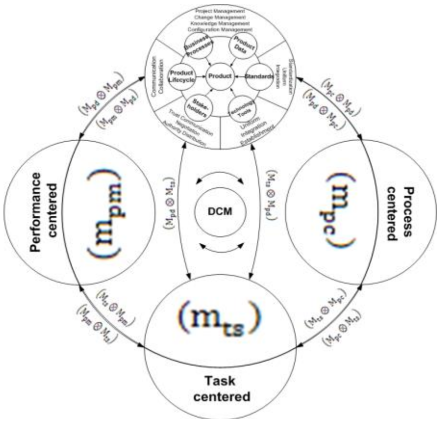

Figure 17 Development for conceptual model of DCM [68]

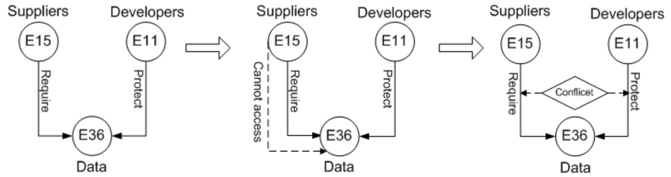

Figure 18 Conflict between developers and suppliers [68]

Further analysis of this conflict has led us to formulate a new kind of IP leakage problem: leakage through malicious redesign. A conceptual model for modeling and mitigating the IP leakage risk has thus been developed [69].

4.2 Experiment: protocol analysis

As was discussed in Section 2.4, one of the requirements for an effective design methodology is that design creativity should happen naturally when designers follow such a methodology and that designers should be in a state of medium mental stress to have a greater chance to be creative. These two requirements imply that an effective design methodology must follow a descriptive design model. Part of our experimental efforts has been to validate that the EBD is also a descriptive model of design.

In one of our experiments, we adopted the design problem used by Dorst and Cross [25] by rephrasing it in a way that is more understandable to our subjects. This design problem is “to design a litter-disposal system for the passenger compartment of a train. This system should be convenient for the passengers to deposit garbage and meanwhile it is easy for the cleaners to collect the garbage.” Seven graduate students, respectively from mechanical engineering, electrical engineering, and computer engineering, were taken as the subjects in our experiment.

The EBD was able to be applied to segment and encode all the protocol data from different subjects [70]. Furthermore, detailed analysis of all those seven subjects has revealed an EBD process underlying their design activities [71]. The differences in their performance lie in their ability to identify correctly the environment components and their relationships as well as the conflicts between environment relationships [70, 71].

5 Conclusions and Future Research Issues

5.1 Conclusions

The EBD is a logical and recursive process that aims to provide designers the right direction for solving a design problem. It includes three activities: environment analysis, conflict identification, and solution generation, which can be carried out simultaneously and recursively.

The EBD stems from the observation that design aims to change an existing environment to a desired one by generating a new artefact. Design starts from the environment, functions for the environment, and brings changes to the environment. This environment changing process implies the recursive evolution of design problems, design knowledge, and design solutions.

Four major requirements for an effective design methodology are formulated from the perspective of the nature of design, cognitive model of design, and driving forces underlying design. They are: 1) to help designers jump out of the recursive loop between design problems, design knowledge, and design solutions; 2) to guide both routine and creative design naturally; 3) To help designers maintain his/her mental stress to an optimal level during the design process; and 4) To include naturally the conditions for the evolution of the design.

Fundamental methods supporting the EBD are also introduced. The core of those fundamentals is ROM – Recursive Object Model, based on which environment analysis, conflict identification, and solution generation are conducted. Procedures for each design activity are presented.

Finally, experimental validation and industrial applications are summarized to show the effectiveness of the EBD.

5.2 Future research issues

The development of an effective design methodology is a challenging yet rewarding mission. Many research issues are critical for more efficient application of EBD to complex industrial problems. The following lists some of those issues:

- Integration of EBD, TRIZ, and Axiomatic Design. This integration and collaboration is ongoing between the author’s research group and two research groups respectively focused on those two methodologies. Experiments have been conducted on those three methodologies.
- Relations to other methodologies. It will also be interesting to look into the relation to other design theory and methodology. For example, C-K theory bases itself on the recursiveness of design and the interdependence of design concept and design knowledge, which is the core of recursive logic of design [1], from which the EBD was started.
- Conflict resolution. Currently, most conflict resolution research is focused on collaboration and optimization. How this could be done at a conceptual level for technical problems would be an interesting application of EBD.
- Environment decomposition. The existing ROM based algorithm for environment decomposition is still primitive. More robust algorithms are required.
- ROM for other types of information. ROM was developed for processing textual information. Its extension to geometric, physical, and engineering relations is a critical issue. The link between ROM and existing conceptual models needs to be developed.
- Complex ROM diagram. Though ROM is effective for the collection of the right information, identification of conflicts, and solution generation, it is not convenient for human designers to draw and to manipulate when the diagram becomes big. A robust software tool is needed. The lessons for the growth of finite element method could be applied to ROM’s applications to

highly complex engineering design problems. This is indeed a challenge for all of the design methodologies targeting the conceptual design problems.

- Ontology for relationship/environment. Ontology for environment and environment relationships is important for more efficient application of EBD.
- Routes/stimuli to creativity. Stimuli in EBD application should be tested to validate its support for design creativity.
- Mental stress vs. creativity. Quantitative experimental studies need to be conducted to model the relationship between mental stresses and creativity.

ACKNOWLEDGMENTS

In developing the Environment-Based Design (EBD) over the last two decades, I have been blessed to have the support from numerous teachers, students, and colleagues as well as many companies and funding agencies. This research would be impossible without their help and support. I am especially indebted to Profs. H.B. Liu, J. H. Cao, G. D. Cheng, P. Gu, J. Ronsky, D. Xue, A. Ertas, F. Haghighat, N. Esmail, and R. Dssouli for their support, encouragement, and inspiration. I am also very thankful to my former and current students J. Jing, Z. Chen, B. Yan, B. Chen, S. Yao, M. Wang, S. Li, L. Chen, T. A. Nguyen, W. Liu, X. Sun, S. Tan, Y. Li, R. Pop, A. Moroz, H. K. Bani Milhim, and X. Cao among many others, for their contributions to the development of EBD. I owe a lot to my colleagues and collaborators P. Winkelman, H. Mehdi, N. Schemenauer, Z. Li, A. Pardasani, V. Gupta, F. Zhou, C. Conrardy, D. Tate, R. De Guio, J. Vareille, and C. Eric for their stimulating discussions. Financial supports from NSFC (National Science Foundation of China) (1994-1996) and NSERC (Natural Sciences and Engineering Research Council of Canada) (2004-present), Canada Research Chair Program (2004-present), CRIAQ, Concordia University (2004-present), and Institute of Logistics Engineering (1992-1996). The following organizations and companies have supported the application of EBD to their product/process development and service design through research contracts: Nanjing 14th Research Institute, Novatek International Inc., Mecanica Solutions Inc., Pratt & Whitney Canada Corp., Bombardier Inc., CAE Inc., CMC Electronics Inc., Rolls- Royce Canada Limited., and the City of Edmonton.

I also thank the anonymous reviewers of this paper for their invaluable comments and suggestions.

REFERENCES

[1] Zeng, Y. and G.D. Cheng, *On the logic of design.* Design Studies, 1991. **12**(3): p. 137-141.

[2] Zeng, Y. and P. Gu. *An environment decomposition-based approach to design concept generation*. in *Proceedings of International Conference on Engineering Design'01*.525-532. 2001.

[3] Cross, N., ed. *Developments in design methodology*. 1984, Wiley, Chichester.

[4] Pahl, G., W. Beitz and K. Wallace, *Engineering design: A systematic approach*. 1988: Springer-Verlag.

[5] Hubka, V. and W. Eder, *Theory of technical systems*. 1988: Springer-Verlag.

[6] Hazelrigg, G.A., *Systems engineering: An approach to information-based design*. 1996: Prentice Hall.

[7] Altshuller, G.S., *Creativity as an exact science: The theory of the solutions of inventive problems*. Studies in cyberbetics, ed. F.H. George. 1984: Gordon and Breach Science Publishers.

[8] Suh, N., *The principles of design*. 1990: Oxford University Press.

[9] Yoshikawa, H., *General design theory and cad system*, in *Man-machine communication in cad/cam, proceedings IFIP WG 5.2 working conference*, T. Sata and E.A. Warman, Editors. 1981, North Holland: Amsterdam. p. 35-38.

[10] Tomiyama, T. and H. Yoshikawa, *Extended general design theory*, in *Design theory for cad, proceedings ifip wg5.2 working conference on design theory for cad*, H. Yoshikawa and E.A. Warman, Editors. 1987, Elsevier Science Publishers B. V.: Amesterdam. p. 95-130.

[11] Braha, D. and O. Maimon, *A mathematical theory of design*. 1998: Kluwer Academic Publishers.

[12] Smithers, T., A. Conkie, J. Doheny, B. Logan, K. Millington, and M.X. Tang, *Design as intelligent behavior: An AI in design research programme.* Artificial Intelligence in Engineering, 1990. **5**(2): p. 78-109.

[13] Smithers, T. and W.O. Troxell, *Design is intelligent behaviour, but what's the formalism?* Artificial Intelligence in Engineering, Design, Analysis and Manufacturing, 1990. **2**(4): p. 89-98.

[14] Nagarajan, R.P., S.J. Passey, P.L. Wong, M.C. Pritchard, and G. Nagappan. *Performance measures and metrics for collaborative design chain management*. in *10th International Conference on Concurrent Enterprising*.14-16 June 2004. 2004. Escuela Superior de Ingenieros, Seville, Spain.

[15] Gu, P., M. Hashemian, and A.Y.C. Nee, *Adaptable design.* CIRP Annals - Manufacturing Technology, 2004. **53**(2): p. 539-557.

[16] Bhatta, S.R. and A.K. Goel. *A functional theory of design patterns*. 294-300. 1997. San Francisco, CA, USA: Morgan Kaufmann Publishers.

[17] Gero, J.S. and U. Kannengiesser, *The situated function-behaviour-structure framework.* Design Studies, 2004. **25**(4): p. 373-391.

[18] Stone, R.B. and K.L. Wood, *Development of a functional basis for design.* Journal of Mechanical Design, 2000. **122**(4): p. 359-370.

[19] Umeda, Y., H. Takeda, and T. Tomiyama. *Function, behavior, and structure*. in *AIENG'90 Applications of*

AI in Engineering.177-193. 1990: Computational Mechanics Publications and Springer-Verlag.

[20] Umeda, Y. and T. Tomiyama, *Functional reasoning in design*. Expert, IEEE [see also IEEE Intelligent Systems and Their Applications], 1997. **12**(2): p. 42-48.

[21] Maier, J.R.A. and G.M. Fadel, *Affordance based design: A rational theory for design*. Research in Engineering Design, 2009. **20**(1): p. 13-27.

[22] Steward, D., *The design structure matrix: A method for managing the design of complex systems*. IEEE Transactions on Engineering Management, 1981. **EM-28**(3).

[23] Tomiyama, T., *Design methodologies: Industrial and educational applications*. CIRP Annals - Manufacturing Technology, 2009. **58**: p. 543-565.

[24] Roozenburg, N.F.M., *On the logic of innovative design*, in *Research in design thinking*, N. Cross, K. Dorst, and N.F.M. Roozenburg, Editors. 1992, Delft University Press. p. 127-138.

[25] Dorst, K. and N. Cross, *Creativity in the design process: Co-evolution of problem-solution*. Design Studies, 2001. **22**(5): p. 425-437.

[26] Zeng, Y. and J. Jing. *Computational model for design*. in *Fourth International Conference on Computer-Aided Design and Computer Graphics*.638-643. 1996. Wuhan, China: SPIE.

[27] Maher, M. and H. Tang, *Co-evolution as a computational and cognitive model of design*. Research in Engineering Design, 2003. **14**: p. 47-63.

[28] Zeng, Y. and P. Gu, *A science-based approach to product design theory part I: Formulation and formalization of design process*. Robotics and Computer-Integrated Manufacturing, 1999. **15**(4): p. 331-339.

[29] Hatchuel, A. and B. Weil, *A new approach of innovative design: An introduction to c-k theory*, in *International Conference On Engineering Design ICED 03* 2003: Stockholm.

[30] Reitman, W.R., *Heuristic decision procedures, open constraints, and the structure of ill-defined problems*, in *Human judgments and optimality*, M.W. Shelley and G.L. Bryan, Editors. 1964, Wiley: New York. p. 282–315.

[31] Simon, H.A., *Structure of ill-structured problems*. Artificial Intelligence, 1973.

[32] Zeng, Y., *Axiomatic approach to the modeling of product conceptual design processes using set theory*, in *Department of Mechanical and Manufacturing Engineering*. 2001, University of Calgary: Calgary, Alberta, Canada. p. 229.

[33] Zeng, Y. and P. Gu, *A science-based approach to product design theory part II: Formulation of design requirements and products*. Robotics and Computer Integrated Manufacturing, 1999. **15**(4): p. 341-352.

[34] Zeng, Y., *Environment-based formulation of design problem*. Transaction of SDPS: Journal of Integrated Design and Process Science, 2004. **8**(4): p. 45-63.

[35] Zeng, Y. and S.J. Yao, *Understanding design activities through computer simulation*. Advanced Engineering Informatics, 2009. **23**: p. 294-308.

[36] Akin, O. and C. Akin, *On the process of creativity in puzzles, inventions, and designs*. Automation in Construction, 1998. **7**: p. 123-138.

[37] Dasgupta, *Creativity in innovation and design*. 1994.

[38] Wharton, J.G. *Cognition and creativity: Creativity, imagination and the objectivist epistemology*. 1999 [cited; Available from: www.axiomatic.net/cc.pdf.

[39] Broadhurst, P.L., *Emotionality and the Yerkes-Dodson law*. Journal of Experimental Psychology, 54, 345-352, 1957.

[40] Zhu, S., S.J. Yao, and Y. Zeng. *A novel approach to quantifying designer's mental stress in the conceptual design process*. in *ASME DETC/CIE*.DETC2007-35887. 2007. Las Vegas, Nevada, USA.

[41] Tang, Y. and Y. Zeng. *Quantifying designer's mental stress in the conceptual design process using kinesics study*. in *International Conference on Engineering Design, ICED'09* 2009. Stanford University, Stanford CA, USA.

[42] Nguyen, T.A. and Y. Zeng. *Analysis of design activities using EEG signals*. in *2010 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conferences*. DETC2010-28477. 2010. Montreal, Canada.

[43] Petkar, H., S. Dande, R. Yadav, T.A. Nguyen, and Y. Zeng. *A pilot study to access designer's mental stresses using eye gaze system and electroencephalogram (EEG)*. in *29th International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. DETC2009-86542. 2009. San Diego, USA.

[44] Savransky, S.D., *Engineering of creativity: Introduction to TRIZ methodology of inventive problem solving*. 2000: CRC Press.

[45] *The theory of inventive problem solving*. World Patent Information, 1996. **8**(4): p. 250.

[46] Zeng, Y. and G. Cheng, *Knowledge based free mesh generation of quadrilateral elements in two dimensional domains*. Microcomputers in Civil Engineering, 1993. **8**(4): p. 259-270.

[47] Zeng, Y. and P. Gu. *A set-theoretic model of product design requirements and products*. in *Proceedings of the 12th International Conference on Engineering Design*.1117-1120. 1999. Munich, Germany.

[48] Zeng, Y. and P. Gu. *A set-theoretic model of design process*. in *Proceedings of the 12th International Conference on Engineering Design*.1179-1182. 1999.

[49] Zeng, Y., *Axiomatic theory of design modeling.* Transaction of SDPS: Journal of Integrated Design and Process Science, 2002. **6**(3): p. 1-28.

[50] Zeng, Y., *Environment based design: Process model*. 2004, Concordia Institute for Information Systems Engineering, Concordia University: Montreal. p. 40.

[51] Yan, B. and Y. Zeng, *A formal representation of technical systems*, in *Global perspective for competitive enterprise, economy and ecology, Proceedings of the 16th ISPE international conference on concurrent engineering*, S.-Y. Chou, A. Trappey, J. Pokojski, and S. Smith, Editors. 2009, Springer. p. 465-476.

[52] Zeng, Y., *Recursive object model (ROM) - modeling of linguistic information in engineering design.* Computers in Industry, 2008. **59**(6): p. 612-625.

[53] Wang, M. and Y. Zeng, *Asking the right questions to elicit product requirements.* International Journal of Computer Integrated Manufacturing, 2009. **22**(4): p. 283 - 298.

[54] Yan, B. and Y. Zeng. *On the structure of design conflicts*. in *The 12th World Conference on Integrated Design & Process Technology* 2009. Alabama.

[55] Chen, Z.Y. and Y. Zeng, *Classification of product requirements based on product environment.* Concurrent Engineering: Research and Applications, An International Journal, 2006. **14**(3): p. 219-230.

[56] Aurisicchio, M., R.H. Bracewell, and K.M. Wallace. *Characterising in detail the information requests of engineering designers*. in *Proceeding of IDETC'06, ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference* 2006. Philadelphia, Pennsylvania, USA.

[57] Eris, O., *Effective inquiry for innovative engineering design*. 2004, Boston: Kluwer Academic Publishers.

[58] Georgiev, G.V., T. Taura, A. Chakrabarti, and Y. Nagai. *Method of design through structuring of meanings*. in *Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Technical Conferences & Computers and Information in Engineering Conference*. DETC2008-49500. 2008. Brooklyn, New York, USA.

[59] Tan, S., Y. Zeng, and A. Montazami, *Environment-based design of medical devices: A case study*, in *Biomedical engineering: Health care systems, technologies and techniques*, S. Suh and M. Tanik, Editors. 2011, Springer: New York.

[60] Chen, L. and Y. Zeng. *Automatic generation of UML diagrams from product requirements described by natural language*. in *ASME IDETC*. 86514. 2009. San Diego, USA.

[61] Chen, Z., S. Yao, J. Lin, Y. Zeng, and A. Eberlein, *Formalization of product requirements: From natural language descriptions to formal specifications.* International Journal of Manufacturing Research, 2007. **2**(3): p. 362 – 387.

[62] Zeng, Y., *Knowledge-based finite element model design: Theory and implementations*, in *Research Institute of Engineering Mechanics*. 1992, Dalian University of Technology: Dalian. p. 109.

[63] Blessing, L.T.S. and A. Chakrabarti, *DRM, a design research methodology*. 2009, London: Springer.

[64] Yao, S.J., B.Q. Yan, B. Chen, and Y. Zeng, *An ANN based element extraction method for automatic finite element mesh generation.* Expert System with Applications, 2005. **29**(1): p. 193–206.

[65] Zeng, Y., T.A. Nguyen, B. Yan, and S. Li, *A distance-based parameter free algorithm for curve reconstruction.* Computer-Aided Design, 2008. **40**(2): p. 210-222.

[66] Nguyen, T.A. and Y. Zeng, *VICUR: A human-vision-based algorithm for curve reconstruction.* Robot. Comput.-Integr. Manuf., 2008. **24**(6): p. 824-834.

[67] Twigg, D., *Managing product development within a design chain.* International Journal of Operations & Production Management, 1998. **18**(5): p. 508 - 524.

[68] Sun, X., Y. Zeng, and W. Liu, *Formalization of design chain management using environment-based design (ebd) theory.* Journal of Intelligent manufacturing, 2011. **Accepted**.

[69] Zhang, D.Y., Y. Zeng, L. Wang, H. Li, and Y. Geng, *Modeling and evaluating information leakage caused by inferences in supply chains.* Computers in Industry, 2011. **62**(3): p. 351-363.

[70] Yao, S. and Y. Zeng, *Preliminary study of cognitive model of designer's creativity by using formal protocol analysis*, in *The future of product development*. 2007, Springer Berlin Heidelberg. p. 165-174.

[71] Feng, D. and Y. Zeng. *Validation of environment-based design (EBD) through protocol analysis experiment*. in *Proceedings of International Conference on Engineering Design* 2009. Stanford.