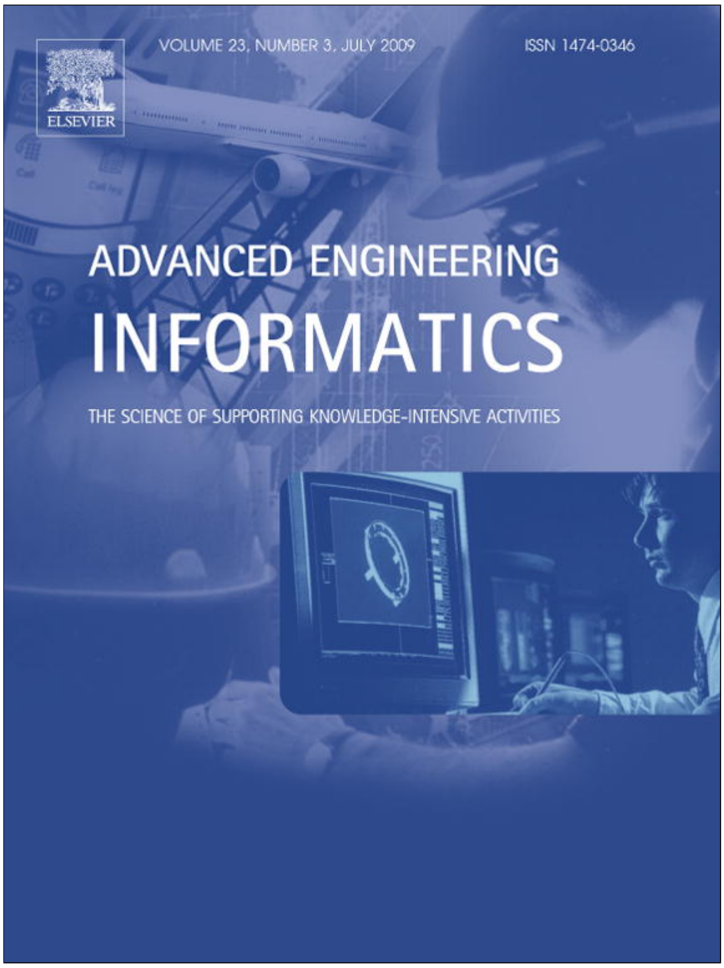
VOLUME 23, NUMBER 3, JULY 2009
ISSN 1474-0346
ELSEVIER
ADVANCED ENGINEERING
INFORMATICS
THE SCIENCE OF SUPPORTING KNOWLEDGE-INTENSIVE ACTIVITIES

Advanced Engineering Informatics 23 (2009) 294–308

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

# Understanding design activities through computer simulation

Yong Zeng [a,*], Shengji Yao [b]

[a] Concordia Institute for Information Systems Engineering, Concordia University, 1455 de Maisonneuve Blvd. West, EV007.633, Montréal, Que., Canada H3G 1M8
[b] Department of Mechanical Engineering, University of New Brunswick, 15 Dineen Dr, Fredericton, NB, Canada E3B 5A3

ARTICLE INFO

*Article history:*
Received 4 October 2007
Received in revised form 9 February 2009
Accepted 17 February 2009
Available online 26 March 2009

ABSTRACT

In this paper, a new experimental approach – computer simulation – is introduced for understanding design activities and for validating design theories. Following the generic framework of computer simulation, three main components in simulating design activities are introduced: mathematical model, simulation model, and statistical analysis. The mathematical model consists of the design governing equation and Environment-Based Design (EBD), based on which three routes are introduced to look for new design solutions: (1) formulating the design problem differently at the beginning of a design process may get quite different solutions, in which creative design could emerge; (2) extending designer's knowledge and experience can help generate more candidate solutions, and so increasing the probability of generating a good concept; (3) changing the sequence of design problem decomposition may change product requirements, and thus change the generated design concepts. By viewing mesh generation algorithms as design agents, a computer simulation environment is used to study design activities. Statistical analysis is conducted to validate quantitatively the three routes to new design solutions. The results show that computer simulation is an effective approach to studying design activities.



## 1. Introduction

During the last several decades, a variety of design theories and methodologies have been proposed, such as systematic design methodology [1,2], decision-based design theory [3], Theory of Inventive Problem-Solving (TRIZ) [4], computational design theory [5], axiomatic design [6], General Design Theory [7], Formal Design Theory [8] and Axiomatic Theory of Design Modeling [9]. These design research efforts can be classified into three major categories: philosophical, deductive and inductive.

Philosophical studies investigate design problem, design objects and design process through retrospection and speculation. The philosophical studies in design have enriched our understanding of design research and provided us a macro-perspective to study design. Yoshikawa indicated that design philosophy is the highest level of speculative thinking about the experience and manifestation of design, the role and position of design in the society, the historical evolution of the design discipline, and the foundational basis of design thinking [10]. Horváth [11] summarized that philosophy of design is often equaled to a meta-theoretical framework for design theories by which epistemological and ontological clarity could be brought in, and often to a philosophy of practice.

In addition to the philosophical approach to design research, a design theory can be developed by using two systematic approaches: deductive and inductive studies, as is shown in Fig. 1. Deductive studies attempt to establish design theories by the direct derivation from axioms whereas inductive studies aim to develop a design theory through the generalization from observations on design activities. These two approaches are taken as top–down and bottom–up strategies, respectively, in [9]. A good design theory must reflect the nature and characteristics of the design process. As stated in [9], a design theory can be verified by applying it to case studies, by comparing it to commonly accepted understanding of design properties through experiments, or by applying it to improve the design practices through managing the identified factors implied in the design process model established in design theories.

Deductive studies aim to establish the fundamental principles and theories for engineering design by identifying the common elements and disclosing the underlying order of the design process. The breakthrough and innovation in understanding and modeling the design process depend on more scientific exploration of design activities, which can provide a formal representation of the design process. This kind of research targets the general design problem and the general model of design through the axiomatic approach. The representative work in exploring mathematical approaches is the General Design Theory [7], Extended General Design Theory [12], Formal Design Theory [8] and Axiomatic Theory of Design Modeling [9]. Yoshikawa established the General Design Theory (GDT) in 1981. Later Tomiyama and Yoshikawa [12] extended the GDT in recognizing that human recognition is imperfect.

* Corresponding author. Tel.: +1 514 848 2424x5801; fax: +1 514 848 3171.
*E-mail address:* zeng@ciise.concordia.ca (Y. Zeng).

1474-0346/$ - see front matter 
doi:10.1016/j.aei.2009.02.001

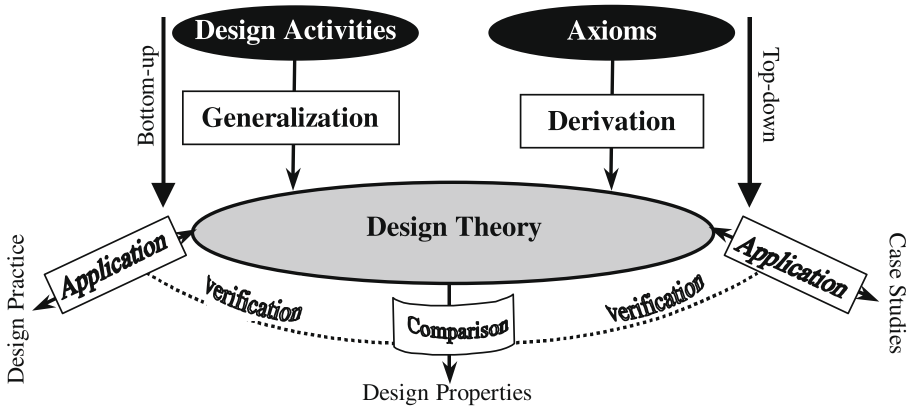


**Fig. 1.** Two main strategies of design research [9].

Tomiyama and his co-workers have applied the GDT to CAD systems and knowledge intensive engineering [13–15]. Braha and Maimon [8] proposed a formal design theory (FDT) in 1998 based on the science of the artificial proposed by Simon [16] and the pioneering work done by Yoshikawa and Tomiyama [7,12]. FDT analyzed ideal design and introduced the evolutionary process in the real design. Braha and Reich [17] further discussed a mathematical framework for describing a variety of complex and practical design processes. Based on the logic of design, Zeng and Gu developed a science-based design theory [18,19], which has been further developed into the axiomatic theory of design modeling [9].

Inductive studies attempt to generalize design theories by observing designer's design activities. Cross summarized the research methods for the study of engineering designers, which include interviews with designers, observations and case studies, protocol studies, controlled tests, simulation trials, and reflection and theorizing [20]. Protocol analysis has been used increasingly since the 1980's in investigating the process of designing and in understanding how designers design [21–23]. Protocol analysis studies a subject's mental processes in accomplishing tasks by recording their spontaneous thinking aloud as running commentary, which will be subsequently segmented into the discrete atomic mental operations [24]. Case study is a popular method used by researchers to produce new design theory, to verify and validate an existing design theory or to explain design phenomena. In contrast, controlled tests and simulation trials have rarely been used in design research.

Motivated by the success of simulation in many other fields such as manufacturing [25], construction [26], education [27], computer science [28], and human experience [29], this paper presents our research progresses in understanding design activities by using computer simulation. Computer simulation allows a user to interact with a computer-simulated environment to simulate different real situations. It is not only cost-effective, but also can be easily and quickly operated to present the real problem in many different perspectives in a virtual environment. To the authors' best knowledge, few systematic research efforts have been reported in design research focusing on computer simulation to understand design processes [30,31].

Computer simulation is the discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer, and analyzing the execution output [33]. As is shown in Fig. 2 [32], the first step in the computer simulation process is to create a descriptive model of the physical system by identifying relevant components and their relationships. Secondly, a formal model, which is often in the form of mathematical equations, is established to capture the behaviors and interactions of the system components. Thirdly, a simulation model is developed

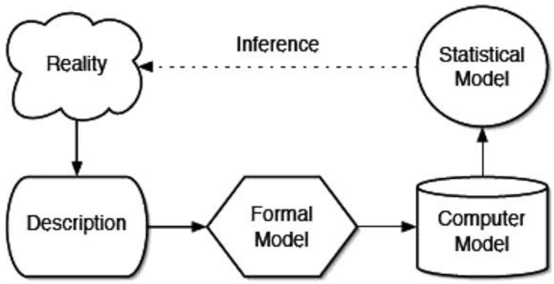


**Fig. 2.** Framework for computer simulation [32].

through computer programming. Finally, numerical experiments can be conducted to collect data for understanding the physical system through statistical analysis.

The rest of this paper is organized as follows. Section 2 introduces a mathematical model underlying design activities, which includes the design governing equation and a design process model. Section 3 describes the simulation model used in this paper. Section 4 conducts the statistical analysis by using the data collected from the computer simulation. Section 5 concludes this paper.

## 2. Formal model of design activities

As the foundation of computer simulation of design activities, a mathematical model was proposed for understanding the factors that lead to a creative design [34] by using the axiomatic theory of design modeling [9]. This mathematical model includes the design governing equation [18] and a new design process model – Environment-Based Design (EBD) [35,36]. The design process model solves the design governing equation, based on which the factors leading to the creative design can be identified.

### *2.1. Design governing equation*

In simulation, most models describing the behaviors of the concerned physical system are mathematical in nature. In the context of design activities, the mathematical model should capture the evolution process of design. To achieve this objective, a mathematical representation of design activities is indispensable, based on which the dynamic mechanism of design evolution must be developed.

Design activities are involved with three main objects: designer, product, and environment, as is shown in Fig. 3. These three objects also interact with each other. A product is any artifact to be designed by designers whereas the environment is where the product is

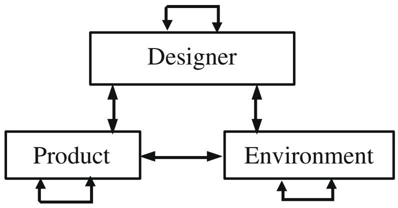


**Fig. 3.** Design activities [37].

expected to work. Theoretically, anything except the product can be seen as the product's environment. Practically, however, only the environment components that impact the product directly will be considered [35]. Take the vehicle design as an example, the vehicle is the product whereas drivers, roads, existing technologies and energy sources are the environment of the vehicle. Both the vehicle and its environment are called object. Any research into design is to investigate either those three objects or their mutual relations (particularly including the relations from each object to itself).

The axiomatic theory of design modeling is developed to represent such a structure as shown in Fig. 3. A key concept in the axiomatic theory of design modeling is the structure operation, denoted by $\oplus$, which can be defined as the union ($\cup$) of an object $O$ and the interaction ($\otimes$) of the object with itself [9].

$$\oplus O = O \cup (O \otimes O), \tag{1}$$

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 include force, movement, and system input and output. Structure operation provides a means to represent a hierarchical system with a single mathematical expression. The application of structure operation can be found in the representation of sketches and linguistic information in design [38,39].

Based on the structure operation, a product system can be defined as the structure of an object ($\Omega$) including both a product ($S$) and its environment ($E$).

$$\Omega = E \cup S, \quad \forall E, S[E \cap S = \Phi], \tag{2}$$

where $\Phi$ is the object that is included in any object.

The product system ($\oplus\Omega$) can be expanded as follows:

$$\oplus\Omega = \oplus(E \cup S) = (\oplus E) \cup (\oplus S) \cup (E \otimes S) \cup (S \otimes E), \tag{3}$$

where $\oplus E$ and $\oplus S$ are structures of the environment and the product, respectively; $E \otimes S$ and $S \otimes E$ are the interactions between the environment and the product [9]. A product system can be illustrated in Fig. 4. The product system is a part of the design activities shown in Fig. 3. Obviously, design activities can also be represented as the structure of designer, product and environment. The details are given in [9].

In the design process, any previously generated design concept can be indeed seen 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

$$\oplus E_{i+1} = \oplus(E_i \cup S_i). \tag{4}$$

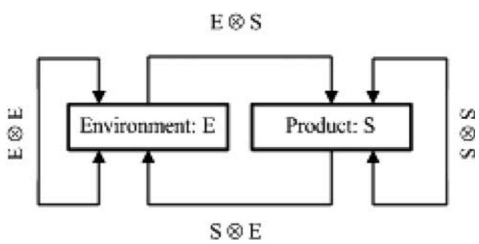


**Fig. 4.** Product system [9].

This evolution process from the design state $\oplus E_i$ to the design state $\oplus E_{i+1}$ is shown in Fig. 5 and is governed by the following design governing equation [35],

$$\oplus E_{i+1} = K_i^s(K_i^e(\oplus E_i)), \tag{5}$$

where $K_i^s$ and $K_i^e$ are synthesis and evaluation operators [18,19], respectively.

The two operators $K_i^s$ and $K_i^e$ correspond to two major phases in the design process: synthesis and evaluation. The synthesis process is responsible for proposing a set of candidate design solutions based on the design problem. It stretches the state space of design. The evaluation process is used to screen candidate solutions against the requirements in the design problem. It folds the state space of design. The interaction of both synthesis and evaluation processes gives rise to the final balanced design solutions, which can be illustrated in Fig. 6.

The design governing equation (5) is a recursive equation and is the mathematical form of the logic of design [40], which characterizes design as a process of simultaneously looking for design solutions and determining the solution evaluation criteria based on the found design solutions. This design governing equation governs design activities and underlies design processes in the same way as the differential equations do to classic engineering sciences. The design governing equation makes design problem solving as a search for fixed points under the design function $K_i^s(K_i^e(\bullet))$. Different design methodologies indeed solve the design governing equation (5) under different assumptions.

According to the recursive logic of design [40], at most stages of (conceptual) design, the evaluation operator $K_i^e$ will be determined only after a (partial) design solution is generated, which will in turn trigger new synthesis operators $K_i^s$. As a result, a small change in the initial design problem may give rise to significant differences in the final design solutions, among which creative design solutions may exist [35].

### *2.2. Design process model: Environment-Based Design (EBD)*

In solving the design governing equation (5), a new design methodology – Environment-Based Design (EBD) [35,36] – was logically derived from the axiomatic theory of design modeling [9]. As is illustrated in Fig. 7, the Environment-Based Design includes three main steps: environment analysis, conflict identification, and concept generation. These three steps work together progressively and simultaneously to generate and refine the design specifications and design solutions.

The following explains the three steps included in EBD [36]:

Step 1: Environment analysis: define the current environment system $\oplus E_i$.

$$\oplus E_i = \oplus\left(\bigcup_{j=1}^{n_e} E_{ij}\right) = \bigcup_{j=1}^{n_e}(\oplus E_{ij}) \cup \bigcup_{j_1=1}^{n_e} \bigcup_{\substack{j_2=1 \\ j_2 \neq j_1}}^{n_e} (E_{ij_1} \otimes E_{ij_2}), \tag{6}$$

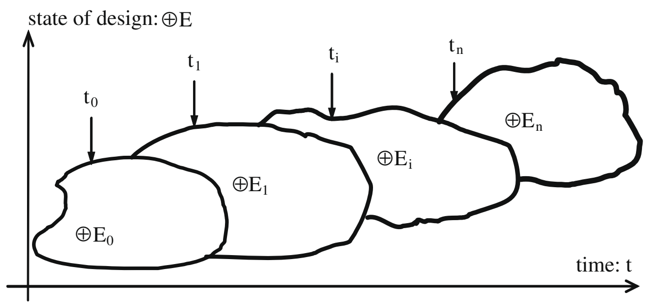


**Fig. 5.** Environment-Based Design: mathematical model [36].

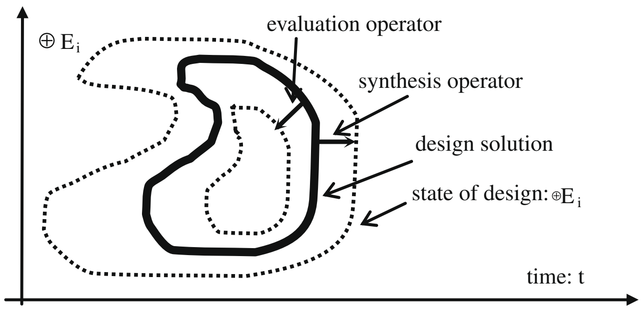


**Fig. 6.** State space of design under synthesis and evaluation operators [18,19].

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.

Step 2: Conflict identification: identify undesired conflicts $C_i$ between environment components by using evaluation operator $K_i^e$, which depends on the interested environment components.

$$C_i \subset K_i^e\left(\bigcup_{j_1=1}^{n_e}\bigcup_{\substack{j_2=1\\ j_2\neq j_1}}^{n_e}(E_{ij_1}\otimes E_{ij_2})\right). \tag{7}$$

Step 3: Concept generation: generate a design concept $s_i$ by resolving a group of chosen conflicts through a synthesis operator $K_i^s$. The generated concept becomes a part of the new product environment for the succeeding design.

$$\exists c_{ik}\subset C_i,\quad K_i^s: c_{ik}\rightarrow s_i,\quad \oplus E_{i+1}=\oplus(E_i\cup s_i). \tag{8}$$

The design process above continues with new environment analysis until no more undesired conflicts exist, i.e., $C_i = \Phi$.

### 2.3. Routes to alternative design solutions

It can be derived from the design governing equation (5) that the following three routes may lead to significantly different design solutions by changing the conditions in design requirements [41], which is necessary for the innovative and creative design:

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

The three routes given above were first observed in studying the design of finite element model [42]. The computer simulation presented later in this paper will be focused on how these three routes influence the quality of final design solutions. Other properties of conceptual design are not the concern of this paper and are discussed in [41]. The following subsections will briefly review these three routes, which will be studied in more details in Section 4 of this paper.

#### 2.3.1. Different formulation of the design problem

The formulation of a design problem is included in the environment system $\oplus E_i$ [35]. The inclusion or exclusion of an environment component $E_{ij}$ will lead to a change of $\oplus E_i$. In addition, in formulating a design problem, designers may choose to group different environment components as one assembly. This will also lead to a change of $\oplus E_i$. This can be seen from different activities between novice designers and expert designers. Since these two types of designers have quite different experience, they usually apply different methods to formulate the problem. As a result, they got different solutions. Even for the same designer, if he or she changes a perspective, the problem will be formulated in a different way and then the solution will be different. Consequently, the object $C_i$ may be changed. This changes the initial condition of the design process, which could result in significant change of design solutions.

#### 2.3.2. Extension of synthesis knowledge

The extension of synthesis knowledge will change the relation $\exists c_{ik}\subset C_i, c_{ik}\rightarrow s_i$. There are a few possibilities: first, more conflicts can be chosen at the same time to generate a design concept $s_i$; and secondly, the same design conflict $c_{ik}$ may be resolved by different design concept $s_i$. Both cases will update the environment system $\oplus E_{i+1}$ differently. When design conflicts are identified by analyz-

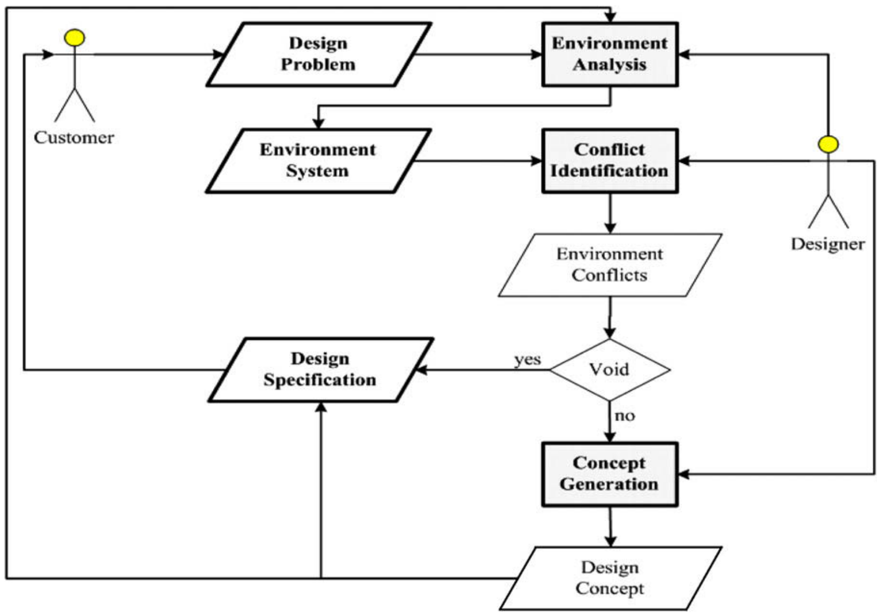


**Fig. 7.** Environment-Based Design: process flow [36].

ing the relations between environment and product, designers will use their knowledge and experience to generate some candidate solution concepts. The number and quality of the design concepts largely depend on designers' knowledge and experience. That is also a big difference between novice and expert designers. The generated concepts need to be evaluated to satisfy the specified product requirements. Novice designers often lack the ability of evaluating the generated concepts correctly, and hence finally fail to generate a good design solution. When we compare designs by a novice designer and an expert designer, we can see a big difference in the design solutions. The newly generated concepts are considered as the environment components and analyzed by combining other identified environment components for generating other design concepts. This process is represented as $\oplus E_{i+1} = K_i^s(K_i^e(\oplus E_i))$, where $K_i^s$ and $K_i^e$ are synthesis and evaluation operators, respectively. We can see that some new and different primitive products may be generated for a specified environment part by extending knowledge. Therefore, extending knowledge can help designers generate more candidate solution concepts, and so increasing the probability of generating a good concept. As a result, a new design problem is generated, which may lead to new design solutions.

#### *2.3.3. Sequence of environment decomposition*

In environment-based design theory, after the design conflicts $C_i$ are identified, there exist many ways to choose the conflict to be resolved, which is based on the decomposition of the environment. Generally, no two designers have exactly the same design knowledge, so they will use different ways to decompose the environment. Different sequences of environment decomposition may give rise to different reformulation of the design problem when designers apply their synthesis knowledge. As a result, the final solutions may be different.

## 3. Computer simulation of design activities

Section 2 has shown the descriptive and mathematical models for the computer simulation of design activities and phenomena, as is required in Fig. 2. This section addresses the third step: development of simulation model and its corresponding computer program. The simulation model indeed leads to an automatic design program, which is able to mimic the human designer's behavior.

In order to simulate the real design process, the simulation model firstly needs to use a design problem that displays the characteristics implied in common design problems. Secondly, the simulation model must show the similar phenomena appearing in common design processes. Thirdly, the chosen design problem must be easily understandable for majority of readers of the research results; therefore, not much domain knowledge should be required. Fourthly, the design problem can be automatically solved by a computerized model, which can be adjusted through various parameters and factors. Finally, evaluation criteria must be available to evaluate different design solutions for the purpose of comparison.

### *3.1. Design problem for simulation model*

#### *3.1.1. Problem description*

In this present research, the problem of automatic 2-dimensional quadrilateral finite element mesh generation is used for the computer simulation of design problem [43]. This problem aims to generate a quadrilateral finite element mesh for a single-connected plane domain with even number of boundary segments. This mesh needs to satisfy the following requirements: (1) each element is a quadrilateral; (2) the inner corner of each element should be between 45° and 145°; (3) the aspect ratio (the ratio of opposite edges) and taper ratio (the ratio of neighboring edges) of each quadrilateral should be within a predefined range; (4) the transition from a dense mesh to a coarse mesh should be smooth; (5) the structural analysis error resulted from the mesh should be minimized. Indeed, the first four requirements are resulted from satisfying the last one with the consideration of the limited computation power. It should be noted that adaptive analysis of structure [44] or mesh optimization techniques [45] have to be employed to satisfy the fifth requirement more accurately. Therefore, the rest of this paper will take into account only the first four requirements, which are illustrated in Fig. 8.

#### *3.1.2. Quantification of design requirements*

To facilitate the discussion in the rest of this paper, the design requirements for mesh generation given in Section 3.1.1 are quantified in this subsection.

*3.1.2.1. Element quality criteria.* Obviously, the best quadrilateral element is a square. Assume that there are $n$ quadrilateral elements in a finite element mesh. For each quadrilateral element $i$, the area of the quadrilateral ($s_i$) can be compared with the area of the corresponding square ($s_{i0}$) that has the same circumference as that of the quadrilateral ($s_i$). The following defines the element area ratio,

$$R_i^e = \frac{s_i}{s_{i0}}. \quad (9)$$

It is easy to show that $R_i^e \leq 1$. For the finite element mesh with $n$ elements, the mean value of all element area ratios is:

$$\bar{R}^e = \sum_{i=1}^{n} R_i^e / n. \quad (10)$$

Upper and lower limits can be set around the mean value $\bar{R}^e$ to determine how many element area ratios are within the lower and upper limits. Since the element area ratio $R_i^e$ is not greater than one, the upper limit for $\bar{R}^e$ can be taken as one. The lower limit

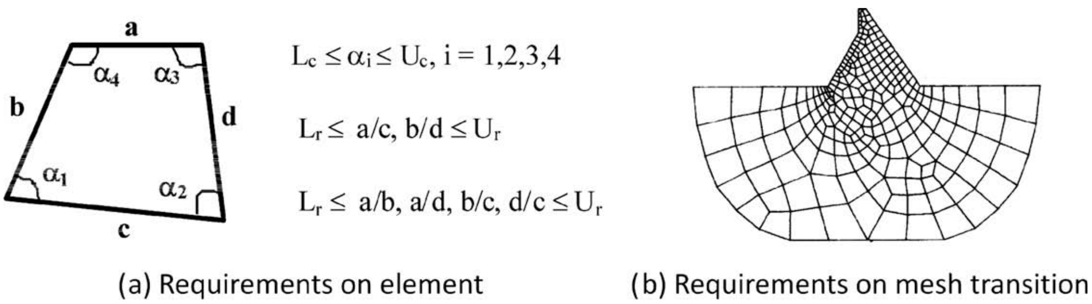


**Fig. 8.** Mesh generation requirements.

can be set by taking 30% off from the mean value $\bar{R}^e$. The percentage of the number of the element area ratios outside of the lower limit can be used to evaluate the element quality for the entire domain. The element quality criterion for the entire domain is given as

$$R^e = \frac{\text{No. of element outliers}}{n}. \quad (11)$$

A big value of $R^e$ means more elements of poor quality whereas a smaller value of $R^e$ means fewer elements out of the lower limit for the element quality.

*3.1.2.2. Transition quality criteria.* Assume that $s_i$ and $s_j$ represent the areas of an element $i$ and its neighboring element $j$, respectively. To evaluate the transition from the coarse mesh to the dense mesh, the areas of the neighboring elements can be compared, as follows:

$$R^t_{ij} = \left(\frac{S_i}{S_j}\right)^k, \quad k = \begin{cases} \frac{S_j - S_i}{|S_i - S_j|} & S_i \neq S_j \\ 0 & S_i = S_j \end{cases}. \quad (12)$$

Obviously, $R^t_{ij} \leq 1$.

The transition ratio can be used to evaluate how close the two neighboring elements are. Assume that there are $m$ pairs of neighboring elements in a generated mesh domain. Those neighboring elements must not be repeated to avoid the recalculation. Then the mean of the transition ratios in the domain is

$$\bar{R}^t = \frac{1}{m}\sum R^t_{ij}. \quad (13)$$

Similar to the index for element quality $\bar{R}^e$, the lower limit for $R^t$ is given by taking 30% off from the mean of the transition ratios. The transition quality criteria for the entire domain can be defined as the percentage of the number of transition ratios outside the lower limit in the number of the pairs of neighboring elements, namely,

$$R^t = \frac{\text{No. of transition outliers}}{m}. \quad (14)$$

Therefore, the criterion of a mesh quality can be defined as the sum of the element quality and the transition quality, as follows:

$$R = R^e + R^t. \quad (15)$$

A small value of $R$ shows a good mesh quality.

### 3.2. Appropriateness of mesh generation for simulation model

Obviously, mesh generation is a problem easily understandable for researchers across disciplines. It can be automatically solved through a computerized model. The criteria for evaluating its quality have been introduced and quantified in Section 3.1. This subsection will discuss other aspects of its appropriateness as a problem for a simulation model of design.

There are many kinds of human intelligent activities, such as forecast, diagnosis, learning, and design. There are also many different kinds of ways to distinguish those human activities. Based on the observation that human intelligent activities are mainly determined by the knowledge and the reasoning modes involved, Zeng and Cheng argued that the logics behind the activities provide demarcation between science, mathematics, and design [40]. By studying the logics underlying human intelligent activities, they proposed that recursive logic is the logic of design [40]. This logic is further confirmed by Roozenburg [46]. Design governing equation shown in (5) put the recursive logic in the form of a dynamic equation [18,35,41]. Environment-Based Design shows the process for solving the design governing equation. This subsection will analyze the mesh generation problem by going through the three main steps in the Environment-Based Design. It should be noted that the discussions below will not follow the order of the three steps given in the Environment-Based Design. Instead, we will discuss environment analysis and concept generation first before we move to conflict identification.

#### 3.2.1. Environment analysis

From Environment-Based Design theory, a design problem is implied in a product system and composed of three parts: the environment in which the designed product is expected to work, the requirements on product structure, and the requirements on performances of the designed product [35]. Table 1 shows the structure of mesh generation problem.

Since structural requirements are resulted from the consideration of performance requirements and available computational resources for existing finite element analysis systems, the structure of mesh generation problem shown in Table 1 can be further simplified as that listed in Table 2.

At the beginning of the mesh generation, the environment system includes two parts: the single-connected plane domain $D_0$ and the description "mesh" $M$. According to Eq. (1), this can be mathematically defined as

$$\oplus E_0 = \oplus(D_0 \cup M) = (\oplus D_0) \cup (\oplus M) \cup (D_0 \otimes M) \cup (M \otimes D_0). \quad (16)$$

In (16), since no elements have been generated, $(D_0 \otimes M) \cup (M \otimes D_0)$ means that the mesh should be in the domain $D_0$. $(\oplus M)$ includes two parts: mesh ($M$) and the relation from $M$ to itself ($M \otimes M$), which are the requirements on the quality of elements and the mesh. At this stage, the only thing that can be well defined is environment, which is the domain $D_0$.

According to the axiomatic theory of design modeling [9], the structure operation given in Eq. (1) provides a recursive definition of a hierarchical system. The complete definition of any object hangs on the primitive objects, which may change dependent on the situations and human experience. In the context of geometric modeling, the most basic components of a plane domain are boundary vertices, denoted by $v_i$, which define the domain $D_0$ as:

$$\oplus D_0 = \oplus\left(\bigcup_{i=1}^{m} v_i\right) = \left[\bigcup_{i=1}^{m}(\oplus v_i)\right] \cup \left[\bigcup_{i=1}^{m}\bigcup_{\substack{j=1 \\ j\neq 1}}^{m}(v_i \otimes v_j)\right], \quad (17)$$

**Table 1**
Structure of mesh generation problem.

| Product | Mesh |
|---|---|
| Environment | E1: A single-connected plane domain with even number of discretized boundary segments<br>E2: Existing finite element analysis systems<br>E3: Available computational resources |
| Performance requirements | R-1: The structural analysis error resulted from the mesh should be minimized |
| Structural requirements | R-2: Each inner corner of a quadrilateral should be within a predefined range<br>R-3: The aspect ratio and taper ratio of each quadrilateral should be within a predefined range<br>R-4: The transition from the coarse mesh to the dense mesh should be smooth |

**Table 2**
Simplified structure of mesh generation problem.

| Product | Mesh |
|---|---|
| Environment | A single-connected plane domain with even number of discretized boundary segments |
| Structural requirements | The smaller the quality index R is, the better quality the mesh has |

where the structure of a vertex ($\oplus v_i$) is $v_i$ since it is the most primitive entity in the context of geometric modeling. The relation between any two different vertices ($v_i \otimes v_j$) could include the distance between these two vertices or the line connecting them.

Alternatively, we can take each boundary segment or generated element as a primitive object. If we take each boundary segment, $e_i$, as a primitive object, then we have

$$\oplus D_0 = \oplus\left(\bigcup_{i=1}^{m} e_i\right) = \left[\bigcup_{i=1}^{m}(\oplus e_i)\right] \cup \left[\bigcup_{i=1}^{m}\bigcup_{\substack{j=1\\ j\neq 1}}^{m}(e_i \otimes e_j)\right], \tag{18}$$

where the structure of an edge ($\oplus e_i$) can be defined by its two vertices. The relation between any two different edges ($e_i \otimes e_j$) could include the angle between these two edges, ratio of their lengths, and parallelism.

### 3.2.2. Concept generation

The final selection of primitive objects is determined by the knowledge available or required to solve the design problem. This subsection illustrates the relation of mesh to Environment-Based Design (EBD) by using two methods: mapping element method and element extraction method. Other methods such as quad-tree and medial axis methods can be represented in the similar manner.

*3.2.2.1. Mapping element method.* The mapping element method, also known as transfinite interpolation method, first divides a plane domain into a few quadrilateral elements manually or automatically [47]. Then a predefined mesh in a unit square is mapped into each quadrilateral according to the transfinite interpolation functions for its four edges. Fig. 9 shows an example of triangular mesh generation using this method.

In this case, the primitives can be defined as mapping elements, denoted by $\xi$, based on which the plane domain $D_0$ can be represented as

$$\oplus D_0 = \oplus\left(\bigcup_{i=1}^{k} \xi_i\right) = \left[\bigcup_{i=1}^{k}(\oplus \xi_i)\right] \cup \left[\bigcup_{i=1}^{k}\bigcup_{\substack{j=1\\ j\neq 1}}^{k}(\xi_i \otimes \xi_j)\right]. \tag{19}$$

The mapping rule, denoted by $r_{\text{m}}$, can be formally described as below:

$$r_{\text{m}} : \xi_i \rightarrow s_i, \quad \forall \xi_i \subset D_0, \tag{20}$$

where $s_i$ is finite element mesh corresponding to the quadrilateral $\xi_i$. This rule corresponds to Eq. (8) for concept generation.

*3.2.2.2. Element extraction method.* Fig. 11 shows the process of an element extraction method developed based also on the recursive logic of design [43]. This element extraction method repeatedly generates an element within the domain using some predefined "if–then" rules until the whole domain is filled with required elements. There are three basic rules to extract an element from the domain shown in Fig. 10. The solid lines represent three basic boundary features, which are, respectively, denoted by $b_1$, $b_2$, and $b_3$, and the dashed lines are the added lines to form an element based on the boundary features. The rule #2 is the most commonly used for dealing with the boundary features.

The boundary features shown in Fig. 10 can be formally represented as an object, denoted by $F_b$, as follows:

$$F_b = \bigcup_{i=1}^{3} b_i. \tag{21}$$

Each boundary vertex can be related to one of these three boundary features. In this context, the primitives for the plane domain $D_0$ are vertices as is given in (17). By applying those three rules, a finite element mesh can be generated as given in Fig. 11. It can be seen from the figure that in each step, the generated finite element mesh $S_{i+1}$ is a combination of the previous mesh $S_i$ and the newly generated element $s_i$; the updated plane domain $D_{i+1}$

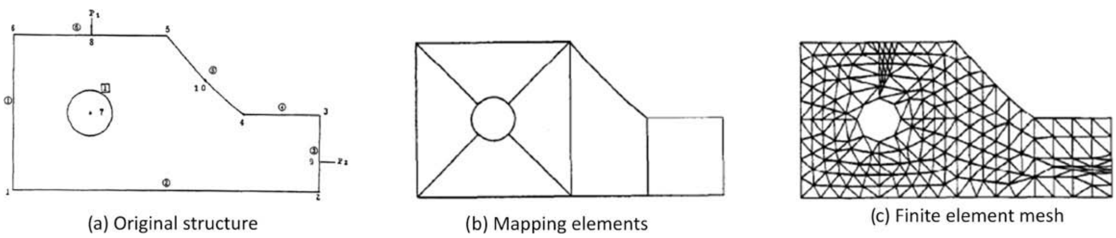


**Fig. 9.** Mapping element method: example [48].

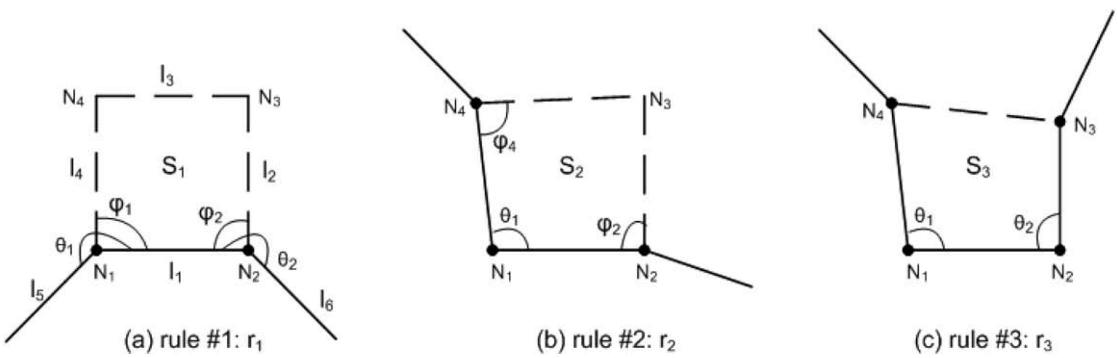


**Fig. 10.** Three basic rules for element extraction.

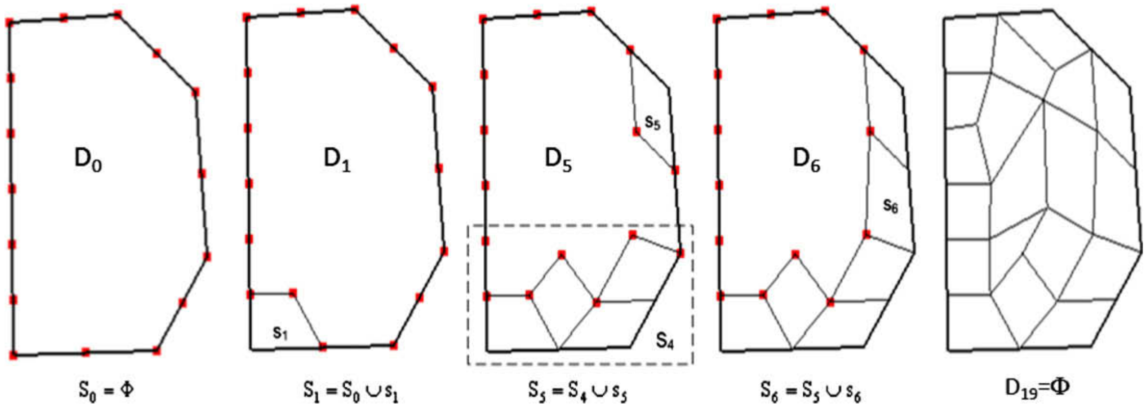


**Fig. 11.** Quadrilateral mesh generation through element extraction.

is resulted from removing the newly generated element $s_i$ from the previous plane domain $D_i$. This process will continue until the entire plane domain is filled with required elements.

The three rules and the mesh generation process shown above can be formally described as below:

$$r_j : v_i \rightarrow s_i, \quad S_{i+1} = S_i \cup s_i, \quad D_{i+1} = D_i / s_i, \quad \forall v_i \in D_i \exists r_j \subset R, \qquad (22)$$

where $r_j$ is one of the element extraction rules $R$, given in Fig. 10; $v_i$ is a part of domain boundary $D_i$; $s_i$ is the element generated around $v_i$; $S_i$ is the generated finite element mesh; symbols ∪ and / are operations "union" and "difference", respectively. This rule corresponds to Eq. (8) for concept generation.

The mapping element method for mesh generation can be viewed as simulating experienced designer's problem-solving process whereas the element extraction method can be taken as simulating the novice designer's design process. The rules (knowledge) in mapping element method deal with more complex situations than those in Fig. 10. Obviously, element extraction method can be used to generate elements within a mapping element.

### 3.2.3. Conflict identification

For the simplicity of discussions, the rest of this paper will use only element extraction method. According to Eq. (7), environment conflicts are resulted from interaction between environment components. During the entire mesh generation process, the conflict between the complexity of boundary conditions and the demand for accurate solutions would drive the progress of the mesh generation. At each step of the element extraction, an extra conflict lies in the good quality of the generated element and the good quality of the remaining plane domain for easy generation of the succeeding elements. Fig. 12 shows three examples of environment conflicts corresponding to the three basic rules listed in Fig. 10, respectively. More examples can be found in [43].

Fig. 12(a) shows that a good quality element $N_1N_2N_3N_4$ can be generated around vertex $N_1$; however, this element would be too close to another vertex P on the boundary, which will make the succeeding mesh quality very poor. The same is true to the example in Fig. 12(b). For the example in Fig. 12(c), $N_1N_2N_3N_4$ would be a good element according to rule #3 defined in Fig. 10 if the remaining angle $\theta'_3$ and $\theta'_4$ are not too small. Under certain situations, rule #2 has to be applied. These conflicts can be identified by evaluating the quality of the generated elements against the predefined requirements. They can be resolved by either through some redesign rules or more advanced concept generation rules that can resolve these potential conflicts in advance. This is again a difference between experienced and novice designers.

It can be seen from the discussions above that mesh generation embodies the basic nature of a generic design problem from problem formulation to the solution process. A mesh generation algorithm can be viewed as a design agent whose responsibilities are to formulate, identify, and resolve conflicts in the environment and whose knowledge can be evolved like human designers. Some of this design agent's capabilities are given by human beings while automatic evolution is also possible [49].

## 3.3. Computer simulation environment

The element extraction algorithm introduced in Section 3.2 was re-implemented in the Visual C++ 2003. This system includes an interface for the computer simulation of design activities, which is shown in Fig. 13.

With this interface, we can generate different design solutions by (1) simulating to formulate design problem differently, (2) simulating to apply different synthesis knowledge, and (3) simulating different environment decomposition sequences. Section 4 will show our experimental results by using this system.

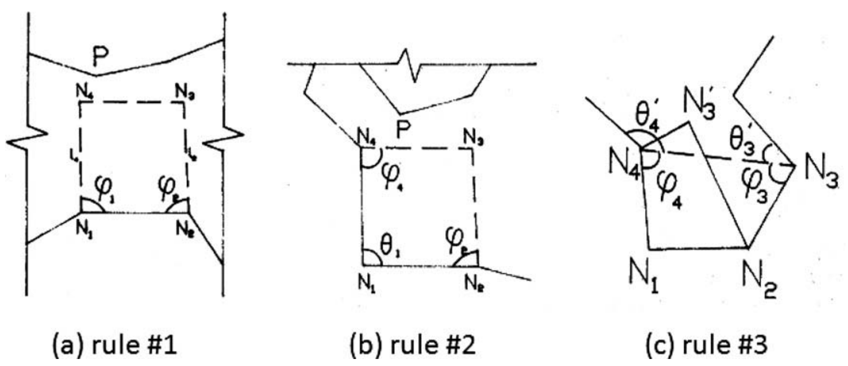


**Fig. 12.** Conflicts between a generated element and the remaining environment: examples.

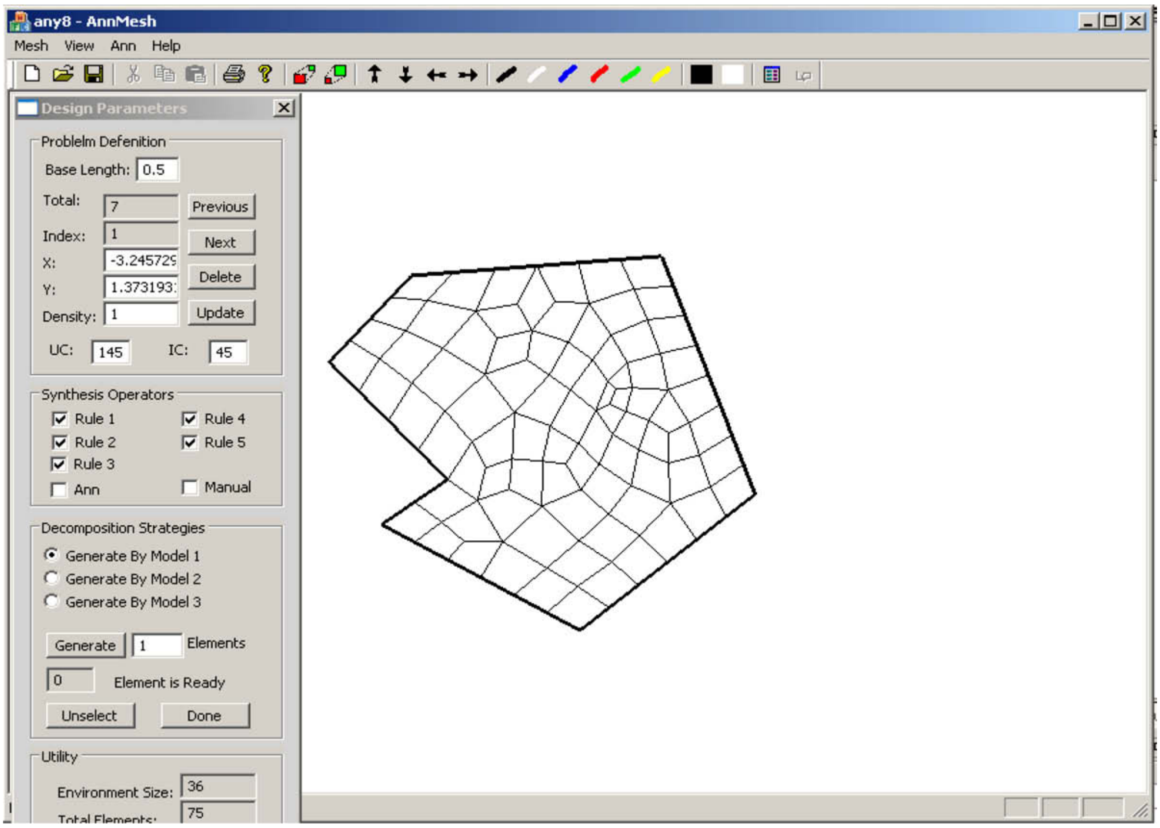


**Fig. 13.** The interface of the mesh generation system.

## 4. Statistical analysis of design phenomena through computer simulation

In this section, based on the simulation tool introduced in Section 3, the statistical analysis of design activities will be conducted to validate and refine the theoretical results obtained from Section 2. For the mesh generation problem, the product requirements are the quality requirements for the generated mesh. In the simulation model presented in this paper, the mesh elements are generated one by one. Each element is generated based on the interaction between the previous generated elements and the environment. The newly generated element is then evaluated for the generation of the next element. This evolution process continues until the domain is filled with elements.

Fig. 14 shows the changes of the mesh quality with the increase of the number of generated elements. At the beginning of the mesh generation, the generated elements have relatively good quality since the boundary feature is not complex. When more elements are generated, the boundary feature becomes more complex and the previously generated elements also affect the quality of the succeeding elements. Thus the mesh quality becomes worse when more elements are generated. This process of the mesh generation corresponds to the general design process. In a design process, a design concept at the beginning can be generated relatively easily based on one product requirement. When more product requirements are considered, the design concept has to be changed, merged or modified to accommodate other design concepts. The final design solution is resulted from the interaction between evaluation and synthesis of the design concepts.

As was indicated in Section 2.1, the design governing equation implies that a small change in the initial condition of the design problem may give rise to huge differences in the final design solutions, among which creative designs may exist. In Section 2.3, three routes are introduced that could affect the initial conditions of a design problem. In this section, we will use the developed mesh generation program to examine quantitatively how the final solutions may be affected by those three routes: formulating the design problem differently, changing the sequence of decomposing the design problem and extending synthesis knowledge.

### 4.1. Formulation of design problem

As is shown in Section 3.2, different mesh generation methods are resulted from different formulation of the plane domain $D_0$, based on different primitive objects. This paper is focused only on the element extraction method, so the formulation of the design problem lies on the initial conditions defining the plane domain $D_0$, which is characterized by its vertices and boundary segments. The boundary segments are further defined by the element base length, the density requirement on the segments around each vertex, and the vertex to start the mesh generation. In the program of mesh generation system, the default values for base length, density and starting vertex are 0.5, 1, and vertex 1, respectively. If the base length is smaller than 1, there are more points on the initial boundary and the boundary segments are smaller. If the density is greater than 1 at a specified vertex, there are more boundary points around the vertex and accordingly the mesh around the vertex is denser. The first element is generated around the starting vertex. Any change in the base length, the density at each vertex, and the starting vertex could affect the final mesh.

To study different formulation of the mesh generation design problem, we take the base length, the density at each vertex and the starting vertex as three independent variables to study the effect on the final mesh. We assume that the three variables have no interaction. The final mesh quality is the dependent variable on the three independent variables. We consider four levels of changes of

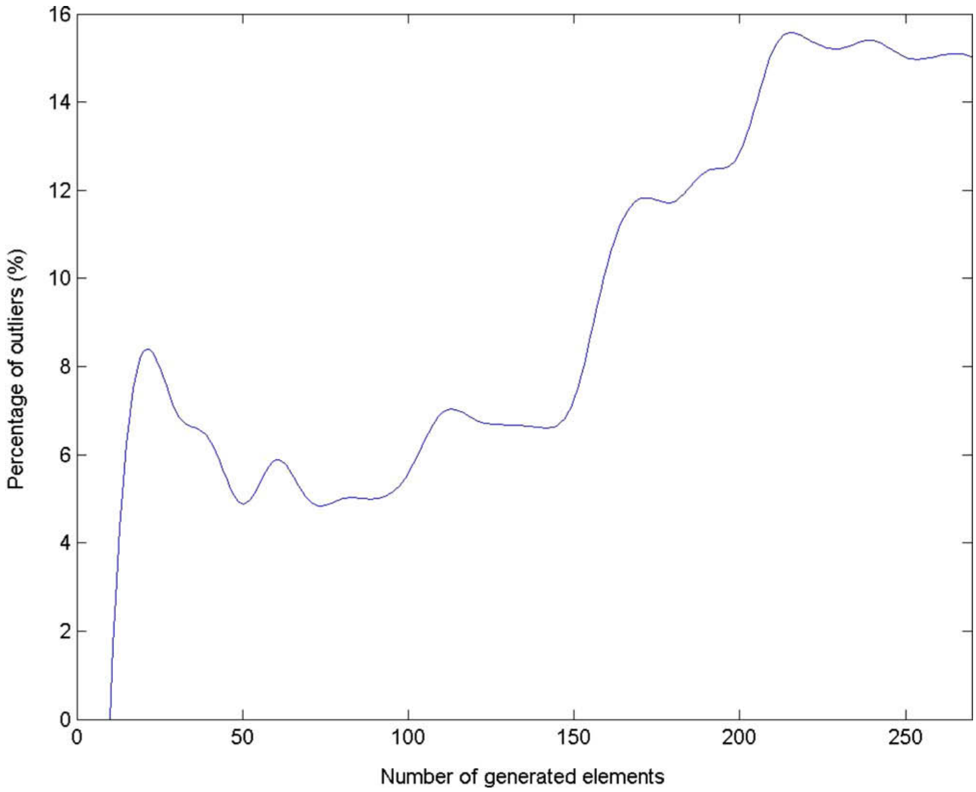


**Fig. 14.** Changes of mesh quality during mesh generation.

the three independent variables. The base length changes between 0.3, 0.4, 0.5 and 0.6; the density changes from 0.8, 1, 1.2 and 1.4; and the starting vertex changes from the first index to the fourth index. The index number is the sequence of the vertices when forming the initial boundary. In this paper, 10 randomly generated quadrilateral domains are used for mesh generation. For each quadrilateral domain, to consider the combination of the effects of the three variables, we applied the orthogonal array to formulate different initial conditions for the domain. The orthogonal array is a method normally required in a full factorial experimental design. For $q$ independent parameters to be identified with $p$ levels, the number of samples required in the orthogonal array is $q*(p-1)+1$ and the orthogonal array is $L_{(q(p-1)+1)}(p^q)$. This number is apparently smaller than the complete sample number of $p^q$, so the method of orthogonal array helps to reduce the number of samples significantly. For 3 variables with 4 levels, the orthogonal array, $L_{10}(4^3)$ and an additional 6 samples are adopted to generate 16 samples for each quadrilateral domain as shown in Table 3.

Then these samples with different initial conditions are used to generate mesh and the generated mesh is evaluated by using the criteria introduced in Section 3.1.2. We take the mesh quality value of the sample 1 as the standard one. Then we calculate the percentage of the deviations of the mesh qualities of the 16 samples from the standard one based on the following equation.

$$\text{Percentage of deviations} = \frac{|\,\text{mesh quality of sample } i - \text{mesh quality of sample 1}\,|}{\text{mesh quality of sample 1}} \times 100\%.$$

**Table 3**
The sixteen samples for different problem formulation.

| Sample | Base length | Density at vertex 1 | Starting vertex |
|---|---|---|---|
| 1 | 0.3 | 0.8 | 1 |
| 2 | 0.3 | 1.0 | 2 |
| 3 | 0.3 | 1.2 | 3 |
| 4 | 0.3 | 1.4 | 4 |
| 5 | 0.4 | 0.8 | 2 |
| 6 | 0.4 | 1.0 | 1 |
| 7 | 0.4 | 1.2 | 4 |
| 8 | 0.4 | 1.4 | 3 |
| 9 | 0.5 | 0.8 | 3 |
| 10 | 0.5 | 1.0 | 4 |
| 11 | 0.5 | 1.2 | 1 |
| 12 | 0.5 | 1.4 | 2 |
| 13 | 0.6 | 0.8 | 4 |
| 14 | 0.6 | 1.0 | 3 |
| 15 | 0.6 | 1.2 | 2 |
| 16 | 0.6 | 1.4 | 1 |

**Table 4**
Three way of ANOVA table.

Analysis of Variance

| Source | Sum Sq. | d.f. | Mean Sq. | F | Prob>F |
|---|---|---|---|---|---|
| Base length | 1.25188e+007 | 3 | 4.17295e+006 | 1.37 | 0.2555 |
| Density | 1.19743e+007 | 3 | 3.99143e+006 | 1.31 | 0.2745 |
| Starting point | 1.17525e+007 | 3 | 3.91749e+006 | 1.28 | 0.2826 |
| Error | 4.58296e+008 | 150 | 3.05531e+006 | | |
| Total | 4.94542e+008 | 159 | | | |

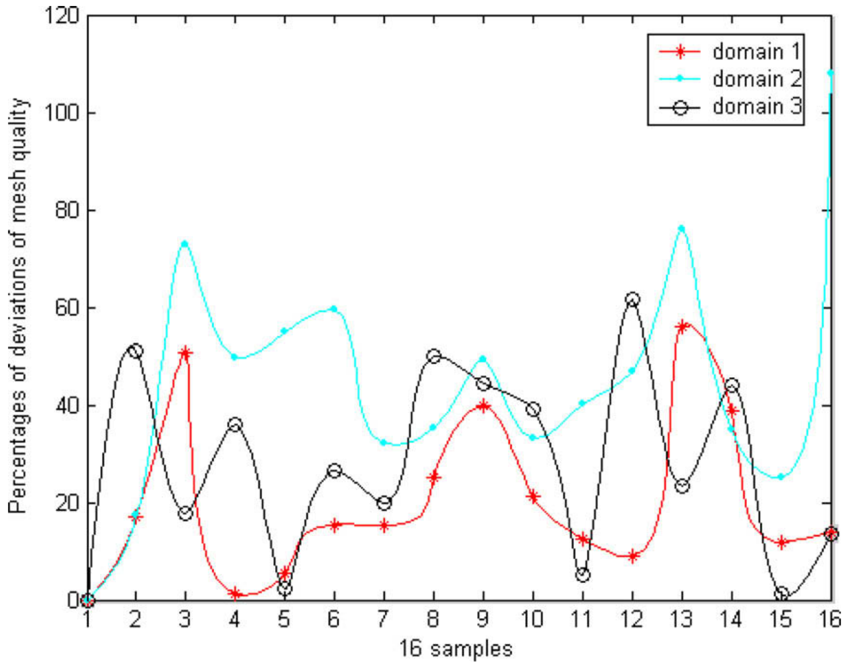


**Fig. 15.** Deviations of mesh quality for three domains.

The three-way of variance was performed with the base length (four levels), density at vertex 1 (four levels) and starting vertex (four levels) as the independent variables and the percentage of deviations of mesh quality as the dependent variable. The level of significance is chosen at 0.05 as a matter of convention. As shown in Table 4, the ANOVA results show the base length ($F = 1.37 > p = 0.2555$), density ($F = 1.31 > p = 0.2745$), and the starting vertex ($F = 1.28 > p = 0.2826$) have significant effects on the quality of the generated mesh. Fig. 15 is the percentage deviations of the mesh quality for three different domains. There are 16 different ways of formulating the initial conditions. We can see that different initial conditions for generating mesh for the three domains make the final mesh quality different. That means small changes of formulating the design problem may lead to various changes on the results of the design solution. It can be implied that formulating the design problem differently may increase the chance of generating creative design solutions.

### *4.2. Sequence of decomposition of design problem*

The decomposition of the mesh generation problem is to decompose the current boundary for generating an element around a reference point from the boundary. A reference point is where we start to decompose the boundary and extract an element, but not every point on the existing boundary can be taken as a reference point. In our study, if the angle at a boundary point is between 45° and 135°, it may be taken as a reference point. Once a new element is generated, the next reference point needs to be selected. The sequence of decomposition of design problem in the example of mesh generation is to find a way to identify a reference point. Various criteria can be used to identify the next reference point. Fig. 16 shows two examples.

The method in Fig. 16(a) always chooses the first valid reference point next to the current one from the updated boundary.

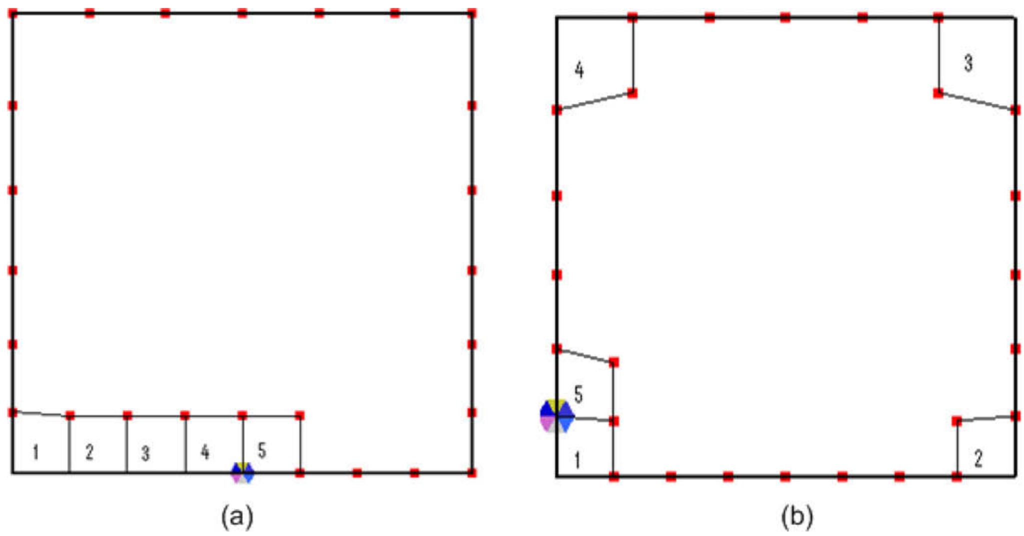


**Fig. 16.** Finding reference point: (a) newer boundary first; (b) older boundary first.

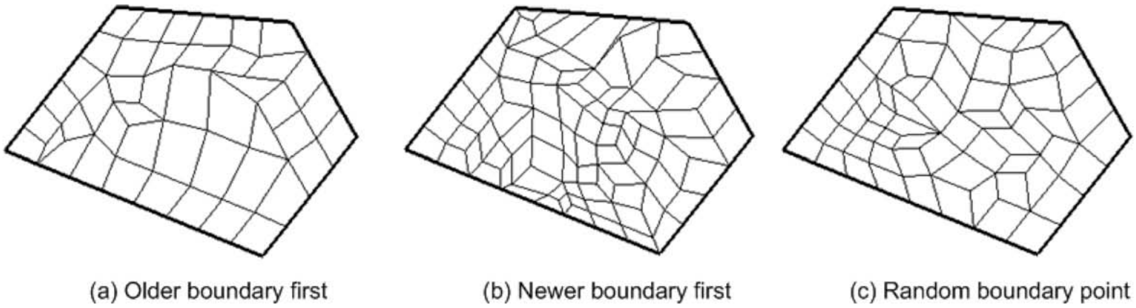


**Fig. 17.** Three different ways of decomposition of mesh generation problem.

**Table 5**
One way of ANOVA table.

ANOVA Table

| Source | SS | df | MS | F | Prob>F |
| --- | --- | --- | --- | --- | --- |
| Columns | 0.01417 | 2 | 0.00708 | 4.93 | 0.015 |
| Error | 0.03883 | 27 | 0.00144 | | |
| Total | 0.053 | 29 | | | |

In another method, the updated boundary will not be processed until all the valid reference points in the previous boundary have been processed. This is shown in Fig. 16(b). In Fig. 16(b), the first four elements are generated from the previous boundary and element five is generated in the new boundary since no more points meet the requirements of a reference point on the previous boundary. If no reference point has been found, then a quadrilateral element will be extracted around the shortest boundary segment. A third method is to randomly choose a valid reference point from the updated boundary for generating an element. Using the three different methods to find reference points to decompose the same domain, different meshes can be generated as shown in Fig. 17.

We used the three different ways of decomposing the mesh boundary on 10 randomly generated domains based on the same initial conditions. One way ANOVA is used to evaluate the effect of the three different decomposition methods on the final generated mesh quality. The ANOVA results, as in Table 5, show that the effect is significant ($F = 4.93 > p = 0.015$). Fig. 18 shows the comparisons of the mesh qualities of 10 different mesh domains using three decomposition methods. We can see that method (a) can generate better mesh quality than methods (b) and (c). This is because method (a) takes into account more boundary information by choosing all possible reference points from the original boundary and updates the original boundary with all newly generated elements. However, with method (b), the local boundary will become bad with the appearance of a bad element and continues to generate bad elements; and method (c) is to randomly choose a reference point on the boundary and it cannot guarantee to generate good elements. In the mesh generation problem, method (a)

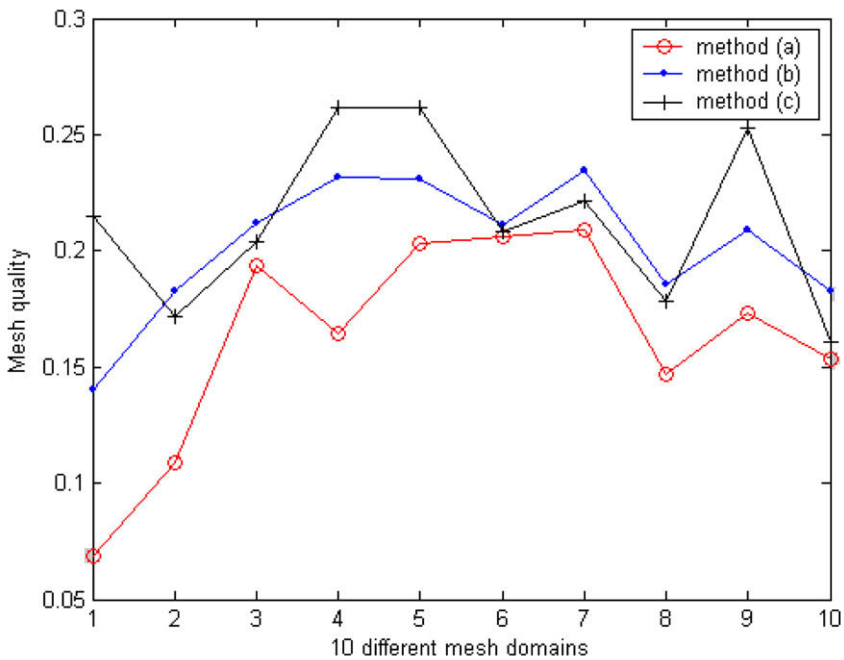


**Fig. 18.** Comparisons of mesh qualities using three decompositions methods.

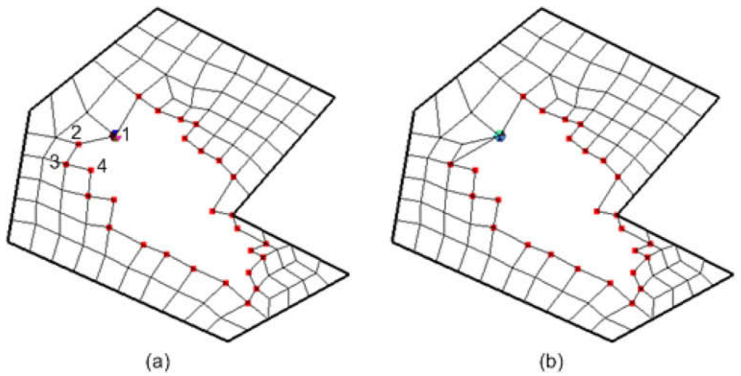


**Fig. 19.** Comparisons of mesh results using different rules.

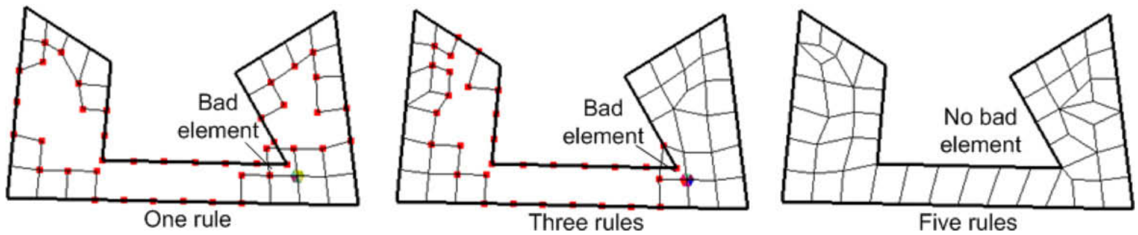


**Fig. 20.** Comparisons of bad elements using different rules.

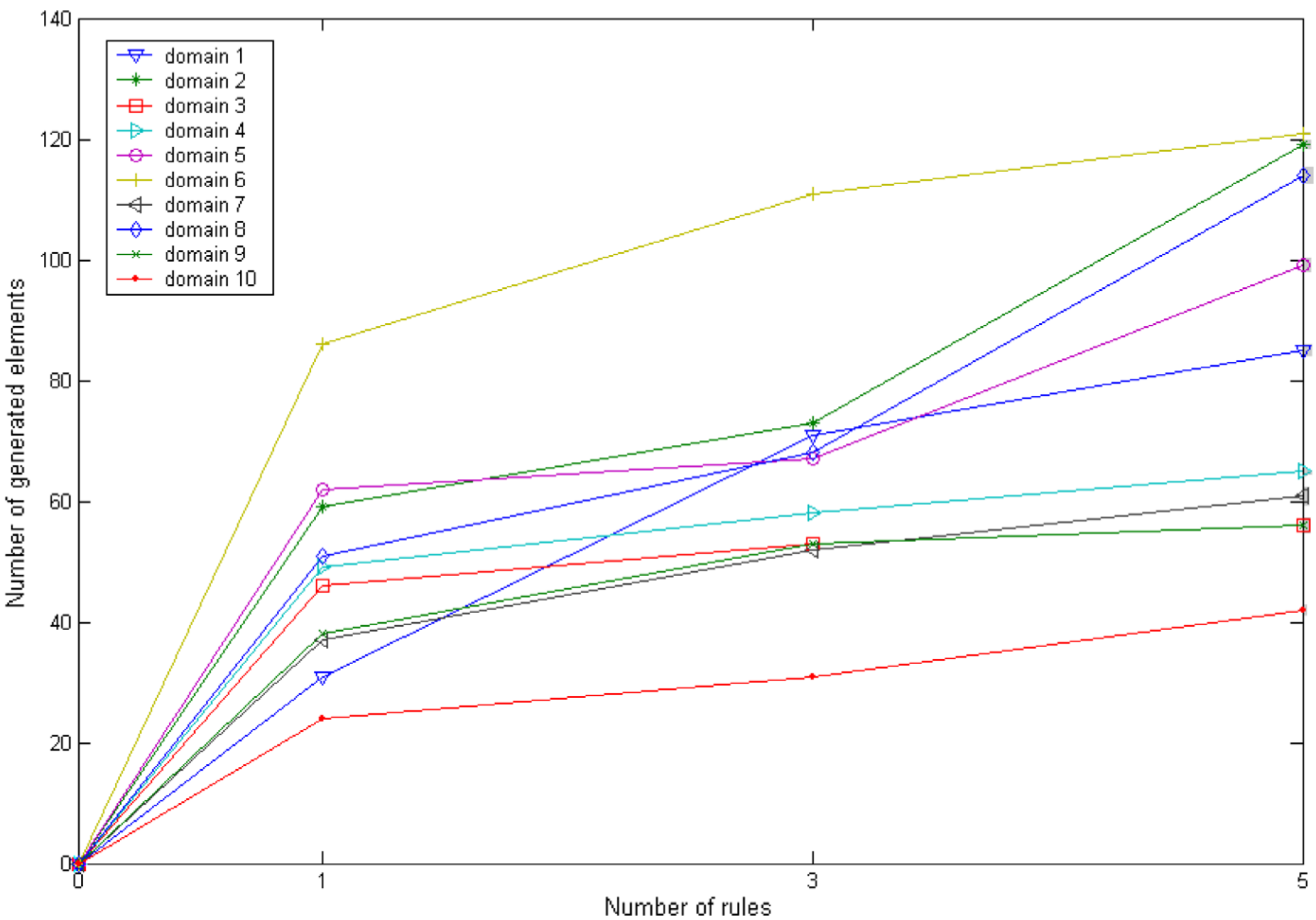


**Fig. 21.** Number of generated elements using different rules for 10 domains.

can be a good way for generating high quality mesh. Therefore, changing the sequence of decomposition of design problem may lead to different design solutions.

### 4.3. Synthesis knowledge

Zeng and Cheng [43] developed five ''if–then'' rules to resolve the conflicts appearing in the mesh generation process through an interactive trial-and-error process based on the three basic element extraction rules. These five rules represent synthesis knowledge for generating an element. For the five heuristic rules, each rule is developed based on the special features of the boundary, so the mesh can be generated better with combining more heuristic rules. Fig. 19 shows an example using different rules to generate mesh elements. In Fig. 19(a), around the boundary points, 1, 2, 3, 4, (b) gives the result of a bad element using three rules, but (c) shows a good element using five rules. With the appearance of the bad element, the succeeding elements cannot be generated well.

The five rules used in the mesh generation design problem are different and each can represent different designers with different synthesis knowledge. Among these rules, rule 5 is an essential rule to generate an element. So we take one rule (rule 5), three rules (rule 3, rule 4 and rule 5), and five rules to generate mesh elements on the same domain, respectively. We selected 10 domains with complex boundary features and applied one rule, three rules and fives rules on the 10 domains for mesh generation. We found out more good elements can be generated when increasing the number of the rules. If the bad elements are generated using fewer rules, the succeeding mesh elements cannot be completed well. Fig. 20 shows the generated elements using one rule, three rules and five rules, respectively.

Then we count the number of the elements generated before the appearance of a bad element using one rule, three rules and five rules on the 10 different domains, which can be shown in Fig. 21. We can see that with more number of rules used in mesh generation, more good elements can be generated. Therefore, extending the synthesis knowledge can increase the chance of getting good design solutions.

## 5. Conclusion

In this paper, a new experimental approach – computer simulation – is developed for understanding design activities. Following the generic framework of computer simulation, this paper introduces the following three main components in simulating design activities: mathematical model, simulation model, and statistical analysis.

The mathematical model is based on the design governing equation derived from the recursive logic of design and axiomatic theory of design modeling. Environment-Based Design (EBD) is introduced as the approach to solving the design governing equation. It can be followed from the mathematical model of design activities that three routes may lead to new design solutions: (1) formulating the design problem differently at the beginning of a design process may get quite different solutions, in which creative design could emerge; (2) extending designer's knowledge and experience can help generate more candidate solutions, and increase the probability of generating a good concept; (3) changing the sequence of design problem decomposition may change product requirements, and thus change the generated design concepts.

In order to develop the computer simulation model, finite element mesh design is used as a design example. It is shown that mesh generation satisfies basic requirements for developing a model for simulating design activities. Two different kinds of mesh generation algorithms (mapping element method and element extraction method) are used to demonstrate that mesh generation resembles generic design in problem formulation and problem-solving process. The algorithm can be viewed as a design agent whose responsibilities are to formulate, identify, and resolve conflicts in the environment and whose knowledge can be evolved like human designers.

By using the simulation environment founded on the mathematical model of design, statistical analysis is conducted to validate quantitatively the three routes to new design solutions. Approaches from experimental design are used to quantify how the final mesh quality may be affected by the factors related to those three routes. The experiment results show that changes in problem formulation, problem decomposition, and synthesis rules can significantly change the final design solutions. Increasing the possibilities of generating new design solutions may increase the chance of getting creative design solutions.

The computer simulation approach proposed in this paper provides a new dimension for studying design activities. Our future work includes the integration of machine learning strategies into the system to investigate more systematically how new synthesis knowledge may play a role in new product design. A generic design system based on Environment-Based Design (EBD) is under development, which will provide an environment to experiment with more design problems. Comparisons with our on-going research in cognitive studies in design will also be conducted.

## Acknowledgments

This work is partially supported by NSERC (Grant number RGPIN 298255). The authors thank Mr. B. Yan and Mr. B. Chen for implementing the mesh generation system used in this paper. We are also grateful to the anonymous reviewers for their invaluable comments, which have helped us greatly in revising this paper.

## References

[1] V. Hubka, W. Eder, Theory of Technical Systems, Springer-Verlag, 1988.
[2] G. Pahl, W. Beitz, Engineering Design: a Systematic Approach, Spinger-Verlag, 1988.
[3] G.A. Hazelrigg, Systems Engineering: an Approach to Information-Based Design, Prentice Hall, 1996.
[4] G.S. Altshuller, Creativity as an exact science: the theory of the solutions of inventive problems, in: F.H. George (Ed.), Studies in Cyberbetics, Gordon and Breach Science Publishers, 1984.
[5] J. Gero, M.L. Maher, A framework for research in computing, http://www.arch.usyd.edu.au/john/publications, 1997.
[6] N. Suh, The Principles of Design, Oxford University Press, 1990.
[7] H. Yoshikawa, General design theory and CAD system, in: T. Sata, E.A. Warman (Eds.), Man-Machine Communication in CAD/CAM, Proceedings IFIP WG 5.2 Working Conference, North Holland, Amsterdam, 1981, pp. 35–38.
[8] D. Braha, O. Maimon, A Mathematical Theory of Design, Kluwer Academic Publishers, 1998.
[9] Y. Zeng, Axiomatic theory of design modeling, Transaction of SDPS: Journal of Integrated Design and Process Science 6 (3) (2002) 1–28.
[10] H. Yoshikawa, Design philosophy: the state of the art, Annals of the CIRP 38 (2) (1989) 579–586.
[11] I. Horváth, A treatise on order in engineering design research, Research in Engineering Design 15 (3) (2004) 155–181.
[12] T. Tomiyama, H. Yoshikawa, Extended general design theory, in: H. Yoshikawa, E.A. Warman (Eds.), Design Theory for CAD, Proceedings IFIP WG5.2 Working Conference on Design Theory for CAD, Elsevier Science Publishers B.V., Amesterdam, 1987, pp. 95–130.
[13] M.S. Erden, H. Komoto, T.J. van Beek, V. D'Amelio, E. Echavarria, T. Tomiyama, A review of function modeling: approaches and applications, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 22 (02) (2008) 147–169.
[14] T. Tomiyama, From general design theory to knowledge-intensive engineering, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 8 (4) (1994) 319–333.
[15] T. Tomiyama, Intelligent computer-aided design systems: past 20 years and future 20 years, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 21 (01) (2007) 27–29.

[16] H. Simon, The Sciences of the Artificial, third ed., The MIT Press, 1996.
[17] D. Braha, Y. Reich, Topological structures for modeling engineering design processes, Research in Engineering Design 4 (4) (2003) 185–199.
[18] Y. Zeng, P. Gu, A science-based approach to product design theory. Part I: formulation and formalization of design process, Robotics and Computer Integrated Manufacturing 15 (4) (1999) 331–339.
[19] Y. Zeng, P. Gu, A science-based approach to product design theory. Part II: formulation of design requirements and products, Robotics and Computer Integrated Manufacturing 14 (4) (1999) 341–352.
[20] N. Cross, K. Dorst, N.F.M. Roozenburg (Eds.), Research in Design Thinking, Delft University Press, 1992.
[21] M. Eckersley, The form of design processes: a protocol analysis study, Design Studies 9 (2) (1988) 86–94.
[22] Y. Jin, P. Chusilp, Study of mental iteration in different design situations, Design Studies 27 (1) (2006) 25–55.
[23] L.A. Stauffer, D.G. Ullman, Fundamental processes of mechanical designers based on empirical data, Journal of Engineering Design 2 (2) (1991) 113–125.
[24] http://www.informatics.susx.ac.uk/books/computers-and-thought/gloss/node1.html.
[25] S. Fujii, H. Tsunoda, A. Ogita, Y. Kidani, Distributed simulation model for computer integrated manufacturing, in: Proceedings of the 26th Conference on Winter Simulation, Society for Computer Simulation International, Orlando, FL, USA, 1994.
[26] H. Li, Z. Ma, Q. Shen, S. Kong, Virtual experiment of innovative construction operations, Automation in Construction 12 (5) (2003) 561–575.
[27] V. Marinov, I. Andonov, Virtual experiment as an educational aid in tribology of cutting, Computer Methods in Applied Mechanics and Engineering 147 (3–4) (1997) 323–327.
[28] C.J. Atman, M.E. Cardella, J. Turns, R. Adams, Comparing freshman and senior engineering design processes: an in-depth follow-up study, Design Studies 26 (4) (2005) 325–357.
[29] J. Takatalo, G. Nyman, L. Laaksonen, Components of human experience in virtual environments, Computers in Human Behavior 24 (1) (2008) 1–15.
[30] G. Kampisand, L. Gulyas, Emergence out of interaction: developing evolutionary technology for design innovation, Advanced Engineering Informatics 20 (3) (2006) 313–320.
[31] K.M. Saridakis, A.J. Dentsoras, Soft computing in engineering design – a review, Advanced Engineering Informatics 22 (2) (2008) 202–221.
[32] P.J. Sanchez, Fundamentals of simulation modeling, in: Proceedings of the 2007 Winter Simulation Conference, Society for Computer Simulation International, Washington DC, 2007.
[33] P.A. Fishwick, Simulation Model Design and Execution: Building Digital Worlds, first ed., Prentice Hall, Upper Saddle River, NJ, USA, 1995.
[34] Y. Zeng, B.Q. Yan, B. Chen, S.J. Yao, A theoretical and experimental study on design creativity, in: The Inaugural CDEN Design Conference, Montreal, Canada, 2004.
[35] Y. Zeng, Environment-based formulation of design problem, Transaction of SDPS: Journal of Integrated Design and Process Science 8 (4) (2004) 45–63.
[36] Y. Zeng, Environment-Based Design: Process Model. Concordia Institute for Information Systems Engineering, Concordia University, Montreal, 2004. p. 40.
[37] Y. Zeng, A formal design science, in: The Inaugural CDEN Design Conference, Montreal, Canada, 2004.
[38] Y. Zeng, Recursive object model (ROM) – modeling of linguistic information in engineering design, Computers in Industry 59 (6) (2008) 612–625.
[39] Y. Zeng, A. Pardasani, H. Antunes, Z. Li, J. Dickinson, V. Gupta, D. Baulier, Mathematical foundation for modeling conceptual design sketches, Transactions of the ASME: Journal of Computing and Information Science in Engineering 4 (2) (2004) 150–159.
[40] Y. Zeng, G.D. Cheng, On the logic of design, Design Studies 12 (3) (1991) 137–141.
[41] Y. Zeng, Axiomatic Approach to the Modeling of Product Conceptual Design Processes Using Set Theory, Department of Mechanical and Manufacturing Engineering, University of Calgary, Calgary, Alberta, Canada, 2001, p. 229.
[42] Y. Zeng, Knowledge-Based Finite Element Model Design: Theory and Implementations, Research Institute of Engineering Mechanics, Dalian University of Technology, Dalian, 1992, p. 109.
[43] Y. Zeng, G. Cheng, Knowledge based free mesh generation of quadrilateral elements in two dimensional domains, Microcomputers in Civil Engineering 8 (4) (1993) 259–270.
[44] B. Emmanuel, C. Patrice, Adaptive mesh refinement for the control of cost and quality in finite element analysis, Finite Element in Analysis Design 41 (15) (2005) 1413–1440.
[45] M.K. Patrick, Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II: a framework for volume mesh optimization and the condition number of the Jacobian matrix, International Journal for Numerical Methods in Engineering 48 (8) (2000) 1165–1185.
[46] N.F.M. Roozenburg, On the logic of innovative design, in: N. Cross, K. Dorst, N.F.M. Roozenburg (Eds.), Research in Design Thinking, Delft University Press, 1992, pp. 127–138.
[47] K. Ho-Le, Finite element mesh generation methods: a review and classification, Computer Aided Design 20 (1) (1988) 27–38.
[48] G. Cheng, Y. Zeng, Strategies for automatic finite element modeling, Computers & Structures 44 (4) (1992) 905–909.
[49] S. Yao, B. Yan, B. Chen, Y. Zeng, An ANN based element extraction method for automatic finite element mesh generation, Expert System with Applications 29 (1) (2005) 193–206.