Research on Cyclic Mapping Model and Solving Approach for Conceptual Design

S. ZHANG
School of Information
Zhejiang University of Finance & Economics,
Hangzhou 310018
CHINA
zs760914@sina.com

Abstract: - An ideal conceptual design model should support multi-level innovative design through rational mapping layer and mapping relationship. In this paper, five existing conceptual design models are reviewed from this perspective, and a new model which supports the alternation of cyclic mappings among functional decomposition functional solving and combination of solutions in turn is presented. Then the knowledge representation scheme for principle solving and the feature-based scheme for interface representation are discussed. Finally a cyclic solving approach to conceptual design is put forward, and an interface matrix is applied to facilitate computer processing.

Key-Words: - Computer aided design; Conceptual design model; Knowledge representation scheme; Design catalogue; Feature model; Interface matrix;

1 Introduction

According to the definitions given by Paul and Beitz [1], the conceptual design can be mainly divided into two stages:(1) functional decomposition: the overall function of a product is decomposed into many sub-functions to build the product's functional structure; (2) functional solving: principle solutions of each sub-function will be derived from functional solving, then those principle solutions will be composed into the solutions of overall function to form the design proposal. It is generally agreed that the two stages proceed in order. The standpoint reflects part of thinking process of conceptual design, it is rational, but there still exist two problems, and they are as follows:

(1)Separating from functional solving will lead to an uncertainty of the granularity of functional decomposition. The designers are unable to estimate what level of the granularity of functional decomposition will make function decomposed to be solved successfully.

(2)Function is the abstract and the summary of design requirements. At the right beginning of the conceptual design, much related information of function is indefinite, qualitative and incomplete. It is not easy to automatically decompose the abstract function with the current artificial intelligence technology.

The above-mentioned problems make current research on the conceptual design automation into some troubles. The author believes that both functional solving and functional decomposition in the conceptual design incorporate features of relative independent ability and process in order as well as features of reciprocal causation and alternation and interaction. The problems arising in the conceptual design automation will be alleviated when these two stages are integrated, and a further study will be made in this paper.

2 The existing conceptual design models

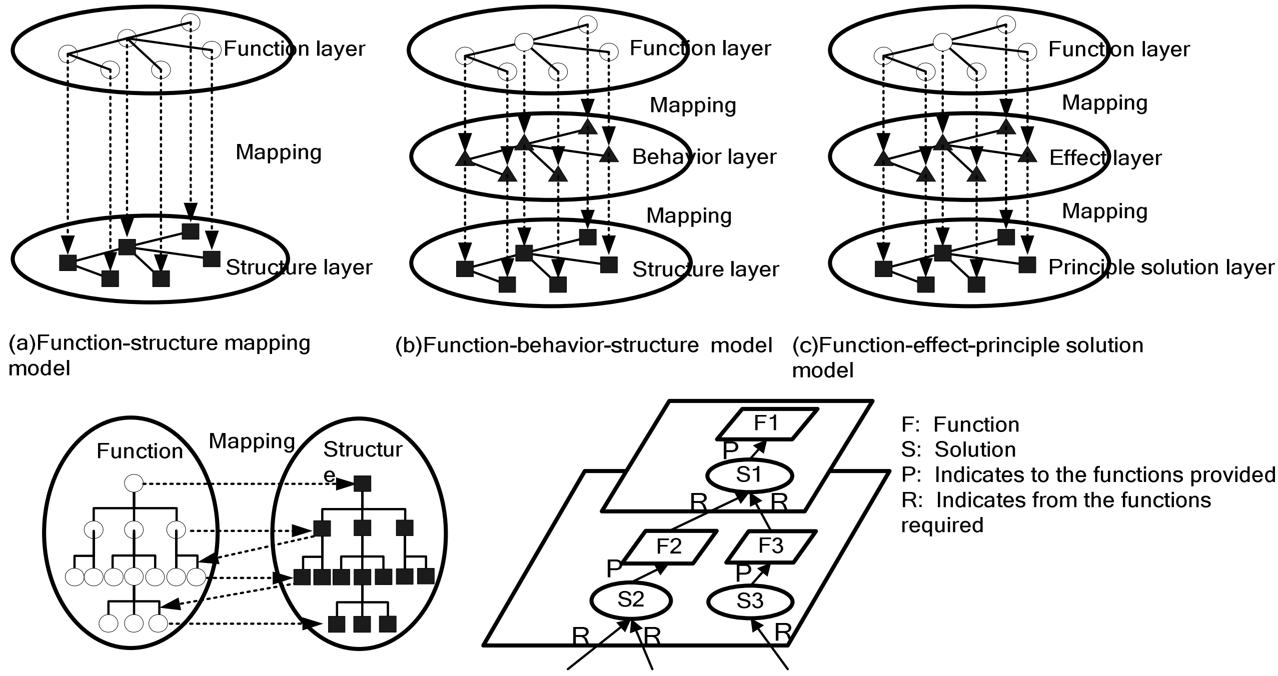
The conceptual design model is a formal representation as well as a regular description for the conceptual design process. It is also a direct embodiment of design thinking. With the research and development of the design methodology, people can have a general understanding and grasp of the conceptual design process, and they can try to establish an operable process model at all levels of the design process. Over the years, due to the extensive research on the conceptual design model which is based on layered mapping, there have been emerged a number of representative layered mapping models.

From the perspective of the layer of mapping, the mapping models can be divided into direct mapping model[2,3,8,9,10] and indirect mapping model[4-7]. Direct mapping model is represented by the function

-structure mapping model (Figure 1a), and illustrates that it can be directly mapped with the direct relationship between its function and structure. However, its limitations lie in the fact that its function is merely the reference of its structure, while neglecting the support to the creative thought as well as a wide range of innovative approaches and multi-level innovation activities in the mapping process. Therefore how to support the brainstorming activities in the function-structure mapping process has become a key issue. Indirect mapping model illustrates that the function and structure are the product descriptions of two different perspectives respectively, which are characterized by the existence of a qualitative difference, so there are difficulties in achieving this mapping directly. Therefore, an intermediate link is required for the interconnection of the functional description and the structural description. Gero, Umeda, took behavior as a bridge which links the function and the structure, and put forward a Function-Behavior-Structure (FBS) model (Figure 1b), holding the opinion that the function is the action to the behavior, and the behavior is a relatively objective description of the change of the product structure or the state of the object which arose from the change of time, which can be indicated by a sequence of change. They tried to explain how the structure fulfills its function via behavior, which shows the support of multi-level innovation activities to some extent. The behavior mentioned here is to highlight the technology sequence or action sequences needed to fulfill certain functions, however, it can not in-depth reflect or explain the fundamental reason that function is realized by the structure, and have some limitations. Feng et al. regarded effect as an intermediate link between the function and principle solution, and put forward the function-effect-principle solution model (Figure 1c), which illustrates that the function epitomizes the design tasks and requirements, the effect describes the basic mechanism of functional realization, and the principle solution describes the realization structure of the effect-it is the embodiment of the effect. When we are making conceptual design for products, we should first identify the function, and then convert functional requirements into effect description, and finally into principle solution. The model is designed in line with the designer's brainstorming process and it is conducive to the innovation of the principle solution, but it is not applicable to certain occasions in which effect and working principle are not obvious.

From the perspective of the direction of mapping, the mapping models can be divided into one-way mapping model[2-7] and the to-and-fro mapping model[8-10]. One-way mapping model emphasized the solving of the function-structure is mapped by way of one-way, one-pass and layered mapping, and therefore such kind of model only supports the solving and the mapping from the functional layer to the other layers, while neglecting the reverse one. Due to this reason, the to-and-fro mapping model is generally only applicable to the conceptual problems with single function. The to-and-fro mapping model is represented by the zigzag function-structure mapping model (Figure 1d) and the Freeman and Newell's model (Figure 1e), which illustrates that the solving of the function-structure can only be realized by repetitive mapping. Generally speaking, the mapping layer of such kind of model is simple, and most of them are direct mapping solving, so they can not fully indicate the cause-and-effect relationship between function and structure. In addition, such models do not put forward operable methods when dealing with the reversed mapping relationship between structure and function. For example, the Freeman-Newell model merely divides the expression of the principle solution into three parts: that is, description of a principle solution, the function that the principle solution can offer and the pre-function that must be fulfilled in advance. However, it is very difficult for the designer to determine the pre-function of an isolated principle solution. Therefore, such knowledge base of principle solution is often hard to build.

From the above analysis we can see that the existing models of conceptual design have their own rationality respectively, but as a whole, there are still three problems: 1) the division of layer for mapping is not rational enough. The existing models can express the relationship that the function is realized by the structure, but it can not fully express the inherent reasons, leading to information faults between the function and structure. 2) the basic requirement of the conceptual design is the diversity of solution path as well as the multiple solutions of design scheme. The existing models lack the support of multi-level innovation activities, and then different levels of creative thinking can not be fully reflected on them in the design process. 3) the existing mapping models are single, mostly one-way, one-pass, and layer by layer, which only support the solving and the mapping from the functional layer to the other layers, while do not in the reverse process. Therefore, they are generally only applicable to the conceptual problems with single function.



(a)Function-structure mapping model (b)Function-behavior-structure model (c)Function-effect-principle solution model

(d)Zigzag function-structure mapping (e)Freeman-Newell model

Fig.1 The existing conceptual design models

3 Cyclic mapping model for conceptual design of complex function product

[Definition] complex function: a number of different and (or) same sub-functions fulfill the complicated function according to a certain relationship and synergistic action.

Conceptual design is a typical ill-defined problem solving process. In the initial stage of conceptual design, due to incomplete information of the function, we can not obtain complete functional structure through the abstract functional decomposition. In fact, the relationship between the generation of the functional structure and functional solving is alternating and reciprocal causation. Based on the above analysis, the author of this paper put forward the cyclic mapping model for conceptual design of complex function product, and it is shown in Figure 2. The model contains three types of mapping patterns, a total number of 12 kinds of mapping relationships, corresponding to the three sub-processes of complex function solving: 1) the top-to-bottom decomposition mapping pattern: when the problem to be solved is rather complicated, the decomposing mapping will be done first to reduce the complexity. 2) solving mapping pattern from left to right: by function→ effect→ working principle→ abstract structure(F→E→P→S) layers mapping or cross-layer mapping to seek access to the effect solution, principle solution and abstract structure solution whose sub-functions are to be solved. 3) the right-to-left derivative mapping pattern: to eliminate the interface mismatch that arose from the combination of sub-functional solution . That is, when the related sub-functional solution makes a combination to form the overall function on the effect layer, the working principle layer and the structure layer, interface mismatch may be produced, and at this time the additional sub-function is required to be derived to eliminate the collision. In the above-mentioned three types of mapping patterns, the solving mapping is the core and the main body, and the decomposition mapping is the preparation for the smooth progress of solving mapping, while the derivative mapping is the supplement for the imperfection of the solution that is obtained from the solving mapping; the above-mentioned three types of mapping are both end to end, alternating, as both reciprocal causation, and they form three different layers of cyclic mapping in the effect layer, the working principle layer and the abstract structure layer (as shown in the dotted line of figure 3), eventually generating a full functional structure and derives the corresponding overall structure solution.

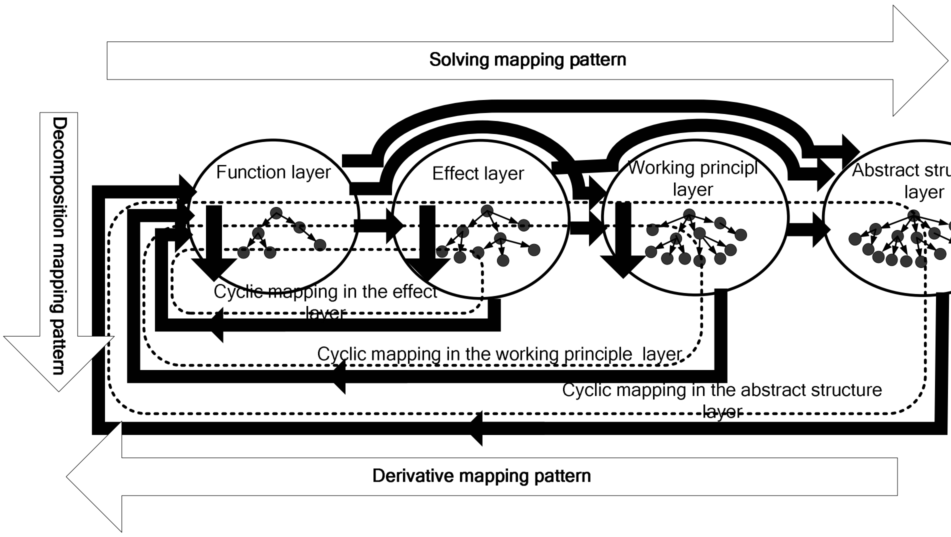


Fig.2 The cyclic mapping model for conceptual design

4 Key technology of cyclic solving for conceptual design

4.1 A knowledge representation scheme for principle solving

In essence, the mapping process of the conceptual design is a knowledge-based problem-solving process, so the quality of the knowledge base has a great impact on it. Results demonstrated that, most of knowledge that is required for the principle solving comes from the physical effects or the working principle which have been found, and most of them have gotten successful engineering applications. The innovative principle solving is actually repetition as well as re- engineering of those knowledge and examples that are already in existence. As early as four decades ago, some German scholars, Roth, Koller, Pahl and Beitz, etc. had began to get down to systemic codification, expression and application of principle knowledge by their hands, and the results they got had provided an important reference for the establishment of the principle solving knowledge base. The mid-and late 90s, Feng et al. launched a study on design catalogues and developed a number of design catalogues of principle solution with typical functions. Taking into account the complexity and ill-defined feature of the conceptual design knowledge, it will be difficult to extract and describe such knowledge directly by using computer. The author made an in-depth research on the knowledge acquisition of principle, and proposed the "design catalogue→feature model→feature frame→class and object" knowledge representation scheme for principle solving (Figure 3). The author also took the separation function, the driving function and the transmission function, etc. as examples, collected more than 100 effects catalogues and over 2000 working principle catalogues and abstract structure catalogues, and then developed a design catalogue which includes 500 universal typical solutions by listing, abstracting and making classification. In addition, the author used the feature model of principle solving for the complex function to make hierarchical reorganization onto the design catalogue knowledge, and used the feature-frame method as well as object-oriented programming to convert it into the computer-oriented knowledge base that have successfully solved the existing problems of knowledge representation to principle solving for the complex function.

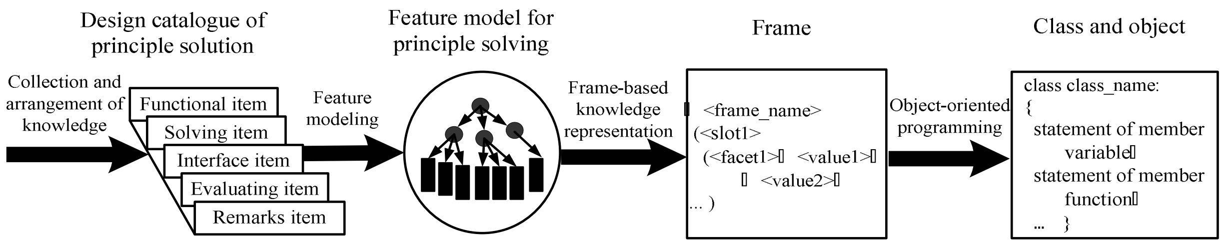


Fig.3 The knowledge representation scheme for principle solving

4.1.1 Design catalogue of principle solution

The design catalogue of principle solution is the knowledge base for carrying out principle solving as well as the knowledge source of principle solving system, so its structure is bound to adapt to the process of principle design. The author has put forward an improved design catalogue of principle solution which is composed of functional item, solving item, interface item, evaluating item and remarks item, and the structure of the design catalogue is shown in figure 4. Functional item is the identification and the reference of the design catalogue, and is the knowledge for providing functional guidance. Solving item describes mapping relationship among the function→effect→working principle→abstract structure, and reflects the multiplicity of solutions and innovation of design. Functional item and solving item correspond respectively to the four-stage sub-catalogues which include functional sub-catalogue, effect sub-catalogue, work principle sub-catalogue and abstract structural sub-catalogue and are the critical knowledge for the principle solving. Interface item, evaluating item and remarks item are the knowledge of carrying out the combination of solution, the decision-making and application respectively, leading to the final result that is merged into one or more optimal and feasible solution. From this viewpoint, it can be seen that, compared to the previous principle design catalogues which were single-function-oriented, the one for complex function provides the support for the entire process of the principle design.

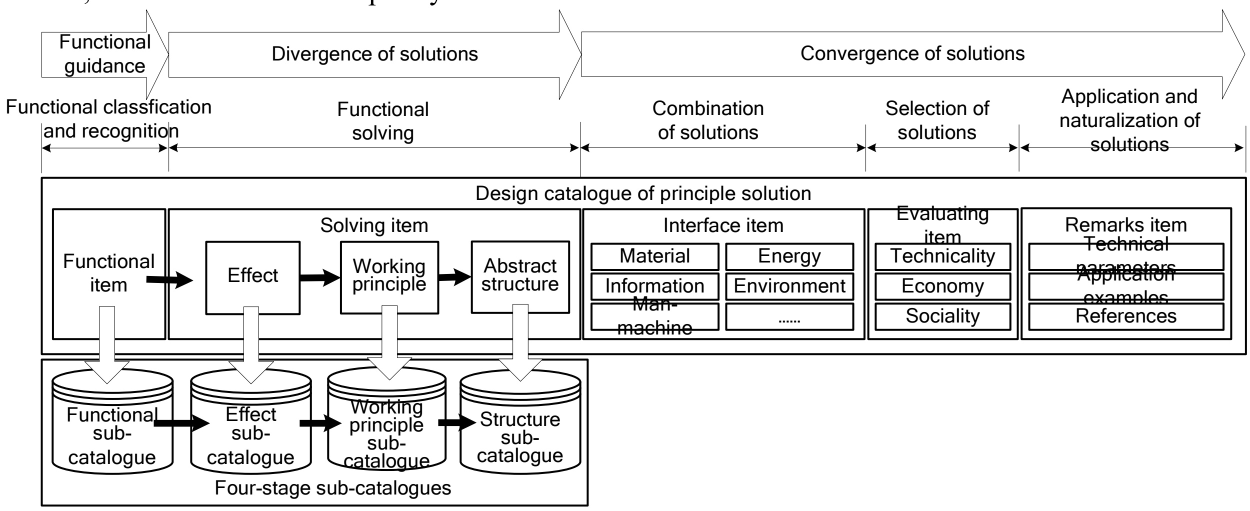


Fig.4 The structure of design catalogue of principle solution

4.1.2 Feature model for principle solving

The design catalogue contains the most relevant knowledge of principle solving, but its form and content come from manual abstraction and manual analysis, e.g. from some figures, texts, diagrams, forms and a combination of empirical data, so the design catalogue-based knowledge lacks systematic method, thus inhibiting the further computer-based processing. In order to enable the design catalogue to be better applicable to computer-oriented principle solving, we must make systematic and hierarchical processing on its system through feature model.

Corresponding to the design catalogue of principle solution, the feature model for principle solving includes the descriptive knowledge of solutions, as well as the knowledge of principle solving, combination of solutions, evaluation of solutions and application of solutions. The former is object knowledge, while the latter is process knowledge. Here, the author puts forward feature model for principle solving of complex function product, and it is shown in Figure 5. The feature model contains two parts: the object feature and the process feature. The object feature makes a description and explanation to those objects, such as function, effect solution, working principle solution and abstract structural solution, which are involved in the process of the principle solving. The object feature is composed of functional feature base, effect solution feature base, working principle feature base and structure solution feature base. The functional feature illustrates the purpose of the solution, the effect solution feature makes a rough description of the working mechanism of the solution, the working principle feature makes a further detailed description of the working mechanism of the solution from the perspective of acting force, acting motion and acting surface, and the structure solution feature embodies and materializes the effect and the working principle.

The process feature makes a description of the key steps of the principle solving, and is composed of solving feature base, interface feature base, evaluating feature base and case feature base. The solving feature describes the mapping relationship among function, effect, working principle and abstract structure, and is the knowledge for the principle solving. The interface feature describes the interface matching relationship among the solution and its related solution and other external objects, and is the knowledge for the combination of solutions. The interface feature is the main focus of the current paper. The evaluating feature describes the evaluation index of the solution from these perspectives of technicality, economy and sociality, and is mainly used for the evaluation as well as the selection of the solutions. Finally, the case feature describes the application of the solutions in the existing products, and is mainly used in the application and the naturalization of solutions.

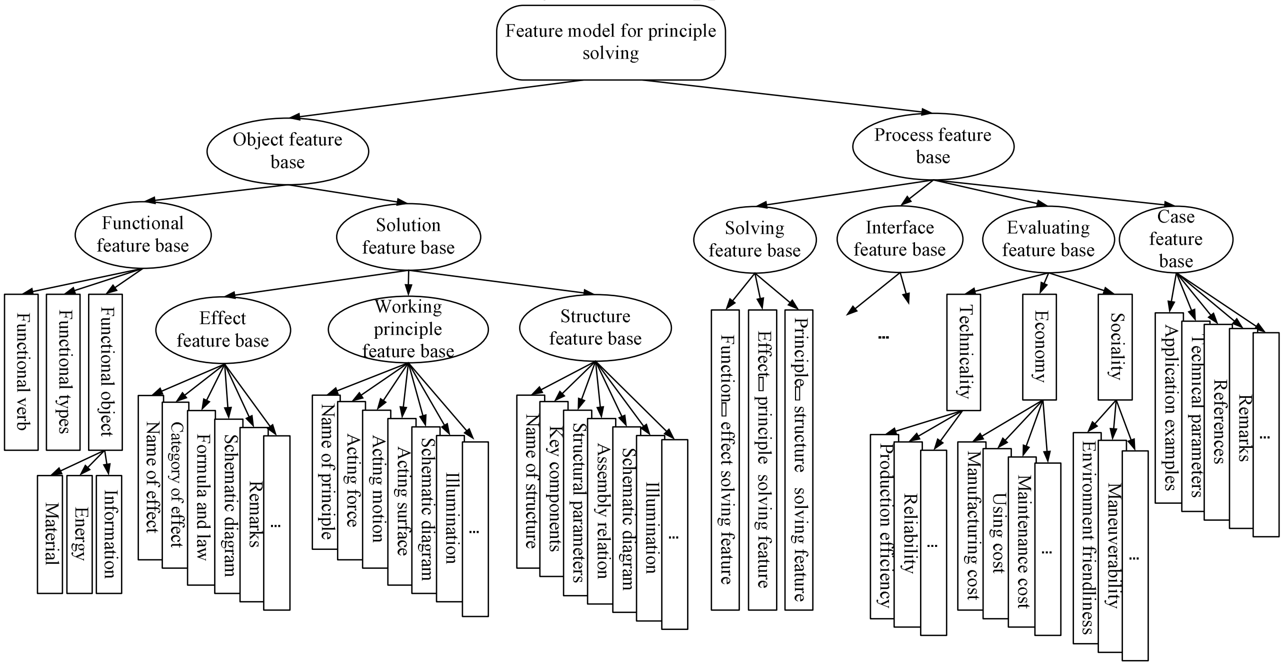


Fig.5 The feature model for principle solving

4.2 The feature-based scheme for interface representation

If we regard the solution as a black box (Figure 6) without taking its internal structure into account but only the interaction between the solution and the external object, then the interaction will constitute its interface.

[Definition] interface: the description of the interaction between the solution and the external objects (including the associated sub-functional solutions, the acting objects, the environment and the human).

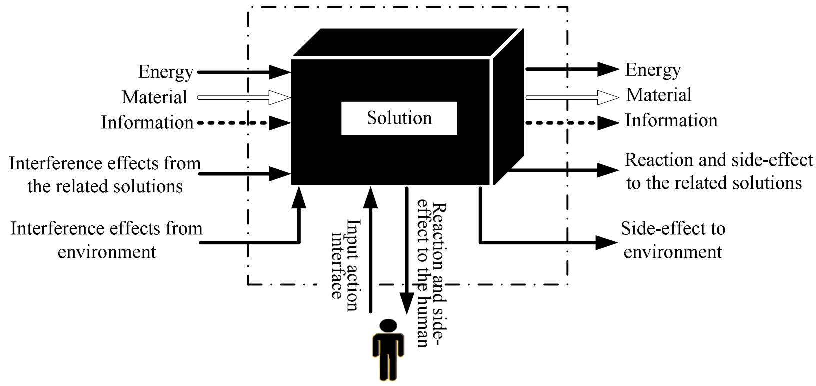


Fig.6 Interaction between the solution and the external object

One of the most important preconditions for supporting the entire process of the conceptual design is to describe the interfaces between different sub-functional solutions, so as to acquire the matching knowledge for the combining the principle solutions. Here, a feature-based scheme for interface representation as shown in figure 7 is put forward in this paper. The interfaces could be divided into acting object interface, associated solution interface, environmental interface and man-machine interface according to objects of interface. According to the acting direction (the solution is taken as reference object) of interface, the interfaces could be divided into input interface (such as the input object interface, interference effect interface, the input action interface, which is shown in Figure 6) and output interface (such as the output objects interface, reaction interface, side-effect interface, which is shown in Figure 6). According to both their advantages and disadvantages, the interfaces could be divided into advantageous interface (Such as the input action interface, reaction interface) and disadvantageous interface (such as the interference effect interference, side-effect interface).

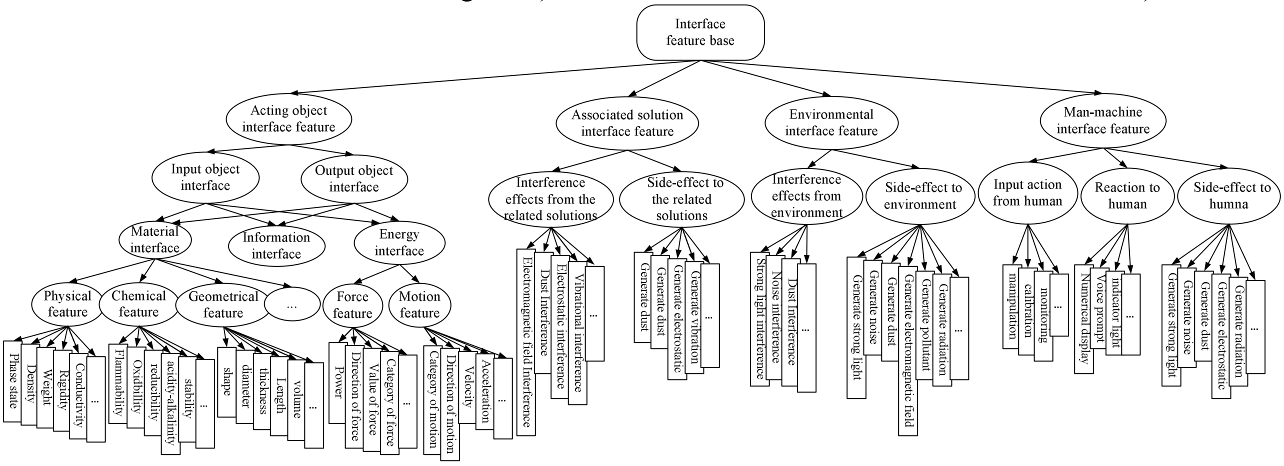


Fig.7 The feature-based scheme for interface representation

We take the solution of ozone generation by corona discharge as example (Figure 8), and its main interface is shown in table 1.

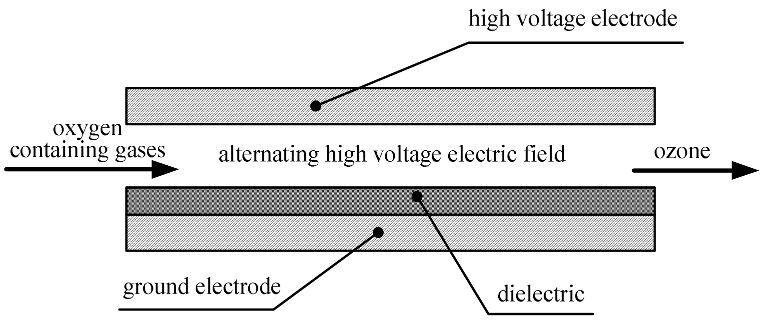


Fig.8 Schematic diagram of ozone generation by corona discharge

Table 1 The interface of the solution of ozone generation by corona discharge

Category of interface	Name of interface	Description of interface	Classification according to the acting direction	Classification according to the advantages and disadvantages
Input materiel interface	Name of material	Pure oxygen or air	Input interface	Advantageous interface
Input materiel interface	Motion status	Flowing air	Input interface	Advantageous interface
Output materiel interface	Name of material	Ozone	Output interface	Advantageous interface
Input power interface	Category of power	Alternating high voltage electricity	Input interface	Advantageous interface
Side-effect interface to environment	Heat	Generate a large number of heat	Output interface	Disadvantageous interface

Side-effect interface to the human	Ozonetoxicity	Harm the human body's respiratory system	Output interface	Disadvantageous interface
Side-effect interface to the related solutions	Electromagnetic interference of	Alternating high voltage electric field has an impact on the stable operation of the low-voltage controlled circuit	Output interface	Disadvantageous interface
…	…	…	…	…

4.3 Cyclic solving approach to conceptual design based on interface mapping

Based on the cyclic mapping model for conceptual design, the author presents an interface matching-based cyclic solving strategy for conceptual design and its steps are as follows:

Step 1: First of all, study on the input object features and output object features of conceptual design requirement, and make them to match with the acting object feature in the knowledge base of functional prototype to reason out the aim sub-function.

Step 2: To obtain a number of solutions of the aim sub-function through the solving of the function→effect→working principle→abstract structure.

Step 3: Check one by one the suitability of interface between sub-function and the external object (including the associated sub-function solutions, acting object, environment and human), and check out whether there is interface mismatch. If the interface matches perfectly, then it means that there is no need to derive new sub-functions; otherwise, it means that new sub-functions are necessary to be derived at the place where the interface mismatch happens to convert the interface, so as to eliminate the interface collision.

Step 4: Check one by one those interface features that produce collisions, and make them to match with the acting object feature in the knowledge base of functional prototype to reason out new sub-functions, and then continue to the step 2.

The process of conceptual design will not be ended until all interface collisions are eliminated in the same way as what have been mentioned above.

4.4 Interface matrix

The above analysis has shown that, the key of the cyclic solving approach to conceptual design lies in checking the interfaces among various sub-functional solutions to find the unmatched interfaces. In order to facilitate computer processing, we use a matrix method in this paper to represent and process the interface mapping in a formal manner.

In Checking the suitability of solutions, we need to view four types of interfaces: the input interface that is required by the solution (recorded as $S_{(s-r)}$), the input interface that is provided by the external objects (including the associated sub-functional solutions, the acting objects, the environment and the human) (recorded as $S_{(o-p)}$), the output interface that is provided by the solution (recorded as $S_{(s-p)}$), as well as the output interface that is required to be offered by the external objects (recorded as $S_{(o-r)}$). Herer we take $S_{(s-r)}$ as an example. $S_{(s-r)}$ can be expressed as the boolean vector which is made up of the elements that all consist of “1”. Since the dimensionality of the boolean vector is the number of elements of the $S_{(s-r)}$, any element in the set can be represented as a unit vector. Any one of the subset of $S_{(s-r)}$ can also be represented as a boolean vector which can be achieved by “or” calculation to the unit vector. Suppose $S_{(s-r)}$ consists of four interface elements (recorded as $s_{(s-r)(i)}$□i=1~4□), then $S_{(s-r)}$=[1 1 1 1]. Any one element of $S_{(s-r)}$ can be represented as the unit vector, such as $\{s_{(s-r)(2)}\}$=[0 1 0 0]. And any of the subset can also be represented as the boolean expression vector, for example, the interface set which consists of $s_{(s-r)(1)}$ □ $s_{(s-r)(2)}$ □ $s_{(s-r)(4)}$ can be expressed as

$\{s_{(s-r)(1)}\}$□$\{s_{(s-r)(2)}\}$□$\{s_{(s-r)(4)}\}$=[1 0 0 0] OR [0 1 0 0] OR [0 0 0 1] = [1 1 0 1] (1)

Similarly, interface sets such as $S_{(o-p)}$□$S_{(s-p)}$□$S_{(o-r)}$ can also be represented as the above-shown boolean vector . Therefore, the interface mapping relationship can be stored through an interface matrix, and the row vector of the matrix is made up of the interface of the solution interface (ie, $S_{(s-r)}$□$S_{(s-p)}$)) , and its column vector is made up of the external object interface (that is, $S_{(o-p)}$□$S_{(o-r)}$) composition. Therefore the interface matrix can be represented as

$$
(S_{(s-r)} ?\ S_{(s-p)})\ ?(S_{(o-p)} ?\ S_{(o-r)}) = \begin{array}{cc}
\begin{array}{cccccccccc} s_{(s-r)(1)} & \cdots & s_{(s-r)(i)} & \ddots & s_{(s-r)(p)} & s_{(s-p)(1)} & \cdots & s_{(s-p)(m)} & \cdots & s_{(s-p)(x)} \end{array} & \\
\left[\begin{array}{ccccc|ccccc}
a_{11} & \cdots & a_{1i} & \cdots & a_{1p} & & & & & \\
\vdots & & \vdots & & \vdots & & & & & \\
a_{j1} & \cdots & a_{ji} & \cdots & a_{jp} & & & O_{qx} & & \\
\vdots & & \vdots & & \vdots & & & & & \\
a_{q1} & \cdots & a_{qi} & \cdots & a_{qp} & & & & & \\
\hline
 & & & & & b_{11} & \cdots & b_{1m} & \cdots & b_{1x} \\
 & & & & & \vdots & & \vdots & & \vdots \\
 & & O_{yp} & & & b_{n1} & \cdots & b_{nm} & \cdots & b_{nx} \\
 & & & & & \vdots & & \vdots & & \vdots \\
 & & & & & b_{y1} & \cdots & b_{ym} & \cdots & b_{yx}
\end{array}\right] &
\begin{array}{c} s_{(o-p)(1)} \\ \vdots \\ s_{(o-p)(j)} \\ \vdots \\ s_{(o-p)(q)} \\ s_{(o-r)(1)} \\ \vdots \\ s_{(o-r)(n)} \\ \vdots \\ s_{(o-r)(y)} \end{array}
\end{array} \quad (2)
$$

In this chart, $s_{(s-r)(i)}$□$s_{(s-p)(m)}$□$s_{(o-p)(j)}$□$s_{(o-r)(n)}$ is the vector element of the interface set $S_{(s-r)}$□$S_{(s-p)}$□$S_{(o-p)}$□$S_{(o-r)}$ respectively. a_{ji} and b_{nm} are the matrix element that restore the result of interface mapping, and their values are as follows:

$$
a_{ji}\text{□ } b_{nm} = \begin{cases} 0: & \text{If the row interface and the column interface do not} \\ & \text{belong to the same type interface} \\ 1: & \text{If the row interface and the column interface belong to the} \\ & \text{same type interface, and they get a successful mapping.} \\ -1: & \text{If the line interface and the row interface belong to the} \\ & \text{same type interface, but they get an unsuccessful mapping.} \end{cases} \quad (3)
$$

In order to facilitate the matrix processing, we partition the matrix into four small pieces of matrix, respectively, which are recorded as $S_{(s-r)}\times S_{(o-p)}$□$S_{(s-p)}\times S_{(o-r)}$□$S_{(s-p)}\times S_{(o-p)}$□ $S_{(s-r)}\times S_{(o-r)}$, and

$$
S_{(s-r)} ? S_{(o-p)} = \begin{array}{cc}
\begin{array}{ccccc} s_{(s-r)(1)} & \cdots & s_{(s-r)(i)} & \cdots & s_{(s-r)(p)} \end{array} & \\
\left[\begin{array}{ccccc}
a_{11} & \cdots & a_{1i} & \cdots & a_{1p} \\
\vdots & & \vdots & & \vdots \\
a_{j1} & \cdots & a_{ji} & \cdots & a_{jp} \\
\vdots & & \vdots & & \vdots \\
a_{q1} & \cdots & a_{qi} & \cdots & a_{qp}
\end{array}\right] &
\begin{array}{c} s_{(o-p)(1)} \\ \vdots \\ s_{(o-p)(j)} \\ \vdots \\ s_{(o-p)(q)} \end{array}
\end{array}
\qquad
S_{(s-p)} ? S_{(o-r)} = \begin{array}{cc}
\begin{array}{ccccc} s_{(s-p)(1)} & \cdots & s_{(s-p)(m)} & \cdots & s_{(s-p)(x)} \end{array} & \\
\left[\begin{array}{ccccc}
b_{11} & \cdots & b_{1m} & \cdots & b_{1x} \\
\vdots & & \vdots & & \vdots \\
b_{n1} & \cdots & b_{nm} & \cdots & b_{nx} \\
\vdots & & \vdots & & \vdots \\
b_{y1} & \cdots & b_{ym} & \cdots & b_{yx}
\end{array}\right] &
\begin{array}{c} s_{(o-r)(1)} \\ \vdots \\ s_{(o-r)(n)} \\ \vdots \\ s_{(o-r)(y)} \end{array}
\end{array} \quad (4)
$$

As there is not interface mapping relationship among the $S_{(s-p)}$ and $S_{(o-p)}$, $S_{(s-r)}$ and $S_{(o-r)}$, $S_{(s-p)}\times S_{(o-p)}$ and $S_{(s-r)}\times S_{(o-r)}$ are matrix which all consist of "0" and recorded as O_{qx} and O_{yp}. This shows that the interface determination only needs to be done among $S_{(s-r)}$ and $S_{(o-p)}$, $S_{(s-p)}$ and $S_{(o-r)}$. Upon completion of all the interfaces mapping, $S_{(s-r)}\times S_{(o-p)}$ and $S_{(s-p)}\times S_{(o-r)}$ are obtained, and their interface mapping results are stored in the matrix elements. By removing all row interfaces and column interfaces whose matrix elements consist of "1", we can obtain two interface conflict matrix, which are recorded as $S'_{(s-r)}\times S'_{(o-p)}$ and $S'_{(s-p)}\times S'_{(o-r)}$. Next, we will view all the row interfaces and column interfaces whose elements consist of "1" in the interface conflict matrix to derive new sub-functions, which is to eliminate the interface mismatching.

5 Conclusion

By analyzing both the advantages and disadvantages of the existing solving models for conceptual design, a cyclic mapping model for conceptual design, which supports functions, effect, working principle and abstract structure cross-mapping is put forward in this paper. The model offers a cyclic mapping mechanism which supports the alternation of functional decomposition□functional solving and combination of solution with the help of the reasonable mapping layer and rich mapping relationship, thus generating a full functional structure and derives the corresponding abstract feasible structural solution. On this basis, the key technology such as knowledge representation scheme for principle solving, feature-based scheme for interface representation, cyclic solving approach to conceptual design, interface matrix have been elaborated in this paper, which lays down a good foundation for the realization of the conceptual design automation.

Acknowledgement. This work was supported in part by 863 Fund of China(2008AA04Z108) and Zhejiang Provincial Education Department's Project of China (20070838).

References:

[1] Pahl, G., Beitz, W.: Engineering design- a systematic approach. Lodon: Springer-Verlag (1996)

[2] Kota, S., Lee, CL.: A functional framework for hydraulic systems design using abstract design building blocks. In: Proc. Of the ASME International Computers in Engineering Conference, Boston (1990)

[3] Bracewell, R. H., Sharpe, J. E. E.: Functional descriptions used in computer support for qualitative scheme generation- schemebuilder. AI EDAM 10 (1996) 333-346

[4] Qian, L., Gero, J. S.. Function-behavior-structure path and their role in analogy-based design. AI EDAM 10 (1996) 289-312

[5] Ranta, M., Mantyla, M., Umeda, Y., et al.: Integration of functional and feature-based product modeling- the IMS/GNOSIS Experience. Computer Aided Design 28 (1996) 371-381

[6] Yan, M.: Representing design knowledge as a network of function, behavior and structure. Design Studies 14 (1993) 314-329

[7] Feng, P. E., Xu, G. R.: Feature modeling based on design catalogues for principle conceptual design. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 10 (1996) 347-354

[8] Suh, N. P.: The principles of design. London: Oxford University Press (1990)

[9] Freeman, P., Newell, A.: A model for functional reasoning in design. In: Proceeding of the Second International Joint Conference on Artificial Intelligence, London, (1971) 621-640

[10] Amaresh, C., Thomas, P. B.: A scheme for functional reasoning in conceptual design. Design Studies 22 (2001) 493-517