

A New Knowledge Modeling Method for Product Design

Lingling Li^{1,2}, Zhigang Li^{1*}, Fenfen Zhu¹, Ligu Ma¹

¹ Province-Ministry Joint Key Laboratory of Electromagnetic Field and Electrical Apparatus Reliability, Hebei University of Technology, Hebei University of Technology

² School of Electrical Engineering and Automation, Tianjin University

* Corresponding author's E-mail: lilingling@hebut.edu.cn

Abstract: The representation and organization of knowledge is the core of key technologies of artificial intelligence, and the modeling of design knowledge is an important and difficult step in developing intelligent designing system. The Design-Mode-Oriented Model (DMOM) used in representation of design knowledge was proposed in this paper. And corresponding to DMOM, three object design models, named as function model, parametric model and case model were also proposed to represent the knowledge of object design. The information gathered from users can be classified into three kinds and each of which represents functions, technical specifications and types of product respectively. Three design modes were also established by mapping the three kinds of information to corresponding object design models. By using model transformation, design information can be processed and the solution of design can be found. This modeling method can be used in the applications of designing for most of electrical and mechanical products.

Keywords: Intelligent design; Knowledge modeling; Process designed model; Object design model; Design mode

1. Introduction

Intelligent design is a design to solve design problems and obtain design solutions through the design-typed expert system and the human-computer intelligent design system. It is a process of design knowledge automatic processing in decision-making level, a major branch derived from the artificial intelligence knowledge and knowledge engineering applications in the design field and regarded as one representative aspect in the latest development of modern design technology.

Intelligent design knowledge includes: process designed knowledge, object designed knowledge and the cognitive knowledge of designer. A new modeling method will be proposed in dealing with the above two in this article. And one part of the cognitive knowl-

edge has been integrated into the knowledge model while the other part has been reflected in the planning, controlling of design task and system reasoning.

The development of intelligent design technology fundamentally depends on the understanding of the essence of design. Since the beginning of 1960s, some industrial developed countries had already been carrying out researches. As a result, a lot of theories were brought up, the famous ones of which are: Germany's systematic product design theory, Japan's statistical theory of product design process, the former Soviet's creative design-oriented problem solving theory and so on. In 1989, Dixon, design theorist and methodologist in U.S, summarized the design theories of that period, and then proposed the four-model theory of product design. The four models are describing model, cognitive model, prescription model

and computational model. From the articles published at the Universal Design Theory International Conference held in Germany in 1998, some design theories had been further deepened, and some new ideas had been put forward. For example, the TRIZ theory was applied into product concept design, the Scenario-FBS framework model was brought up based on the Function-Behavior-Structure (FBS) framework.

In recent years, although the new theories and views [1~4] emerged one after another, a complete design theoretical system has not been established yet, and intelligent design knowledge modeling method [5~8] still needs deep exploration. As the report from the seminar about the product design theory and methodology held by the U.S Natural Science Foundation Committee in 1996 describes: design theory is in pre-theory phase. And now the situation remains.

2. The expression of the design process knowledge

Knowledge expression is a complex task. Compared with other application fields, the knowledge expression in the design field encounters greater difficulties. The reasons are: the concept expression and description of the design are usually closely related with the physical model, which makes the whole design process inseparable from the use of product model; what's more, the product model is not only a collection of geometric information, but also a lot of attributes related with the design process added. About product design process, literatures [9~12] expressed their own understandings from different views. The modeling methods of design process knowledge are different due to varied understandings of design process. At present, the design process models include: the design model based on universal design theory, the case-based model, the analysis-synthesis-evaluation model, the generation-test model, the constraint satisfaction model, the design idea model and the heart-type expression model developed on former basis.

With the focus on design mode, a model expressed the knowledge of design process was proposed in this paper, which is the design model (Design Mode-Oriented Model, DMOM) based on design mode. Consequently, three design object models compatible with the DMOM were also proposed: functional model, parametric model and case model. The three models are all conceptual models. The knowledge modeling method described in this paper is favorable when object-oriented approach is adopted.

2.1 The definition and classification of design mode

Definition 1. Design mode: the methods and process of mapping the product design requirements described by natural language to the object designed model.

The DMOM was put forward based on the following recognition:

From the view of information processing, the design process is a process which is converting input information (such as: functional requirements, design parameters) into output information (such as: design drawings, technical documents). However, the description methods of input information are varied, including: functional description, describing with product specifications, describing by listing of similar products and so on. For design tasks, the starting point is different from varied description methods. This paper treats the different described methods of the product given by users as a classified form of input information, and classifies the design process into "design mode" based on former result.

The description of a target product is usually ambiguous when user uses functional description. It only reflects certain requirements about the principle and the external performance of the target product. For example, to design an intelligent electrical product with the following features: when the current is greater than 4A, the output circuit cuts off with automatic alarm, remote reset and so on. This kind of descriptions does not reveal the essence of the design object by only specifying some functional requirements. From the view of defining product, this method has qualitative characteristic. The design process starts from what is called functional model in this paper.

Technical parameters are the characterizations of the product performance specifications. A set of well-chosen parameters can fully reflect the characteristics of a certain type of products. For example, to design an electrical product, whose technical parameters are: the contact current of output circuit is less than 4A, dielectrics voltage-resistance is 600V, operating time is less than 2ms, control-loop circuit voltage is 27V, and contact resistance is 50mΩ. Experienced designers are able to design a certain scheme quickly, which probably meets the design requirements. This design process is called parametric model in this paper, which uses the product characteristics and design requirements as an entry point.

Different from the two methods above, if a user explicitly specifies the type of a target product with specifying certain design requirements (including functional requirements and technical indicators), the nature of

this design task is to give an optimal design solution. It could usually be accomplished by modifying, supplementing and improving an existing one. This design process is called case model in this article.

2.2 The basic ideas of DMOM

The design model (DMOM) based on design mode is to classify information of the design requirements specified by users and described with the aforementioned three different forms into three categories firstly. Based on which, we can determine specified mode of the design process and map the information to corresponding conceptual models (including the functional model, parametric model and case model) stored in the computer. Then, the conceptual design results can be obtained by a self-adapted mode transformation based on certain control strategies and reasoning algorithms. The last step is to evaluate the conceptual design and re-establish design mode if it does not meet users' requirements. The basic idea of DMOM shows in Fig.1.

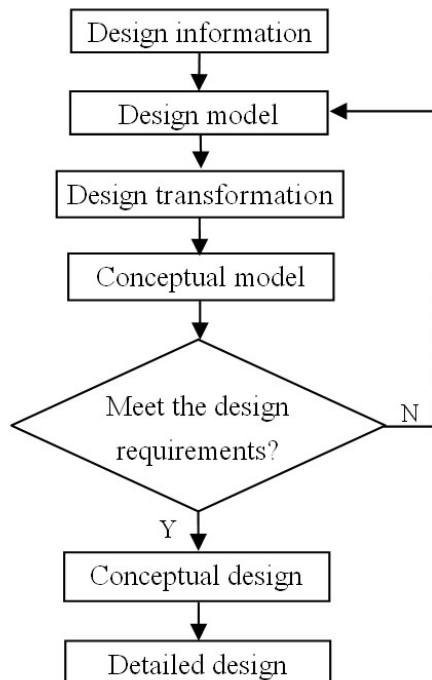


Figure 1 The design model based on design patterns-DMOM

Definition2. Mode Transformation: the transformation among different design modes.

The purposes of applying mode transformation is to process design information provided by users comprehensively on one hand, and to continue to solve design problems in case of failure of one mode on the other hand.

3. Three kinds of object designed model

3.1 The Functional model (Fm) of products

Functional model Fm can be expressed by a 5-tuple:

$$Fm = \{Fa, Aa, Ga, Ta, Sa\}$$

Where, Fa is the functional attribute information, including all kinds of possible functional requirements specified by users; Aa is the behavioural attribute information to express the function-related behavior performed by the design objects, such as the armature motion of a relay in order to achieve on-off function; Ga is the topology information of the carrier and a description of design object's geometry, connection relationship and the spatial position relationship, such as adjacency, tangency and separation; Ta is the characteristic transformation information, which describes the conditions for transformation and the dependence relationship among the different functional properties, behaviour attributes and topological structures information; Sa is the control information during the solving process to guide the solving process based on functional mode, which includes:

(1) The mapping rules from the functional model to the product conceptual design. These rules are used to achieve reasoning based on functional model and get the results of conceptual design.

(2) The mapping rules between functional model and instance. These rules are used to guide the reasoning based on instance and complete the detailed design.

From the above models, function is the interface opened to the user. If using object-oriented language to develop system, in order to achieve this functional model, we must construct the object class of product function, and also define information including class name, attribute descriptions, operation description and inheritance relationship and so on. The BNF (Backus Normal Form) of product object class is defined as:

```

<Functional model> ::= { <Functional model class name>
  < Functional model type> [ <Functional model super class> ]
  <Attribute list> <Attribute table mapping rules> <Component table> <Case table> <Design process>
  [ <Description> ] }
  <Functional model class name> ::= <String>
  <Functional model type> ::= <String>
  <Attribute list> ::= <Attribute> //include the function, structure and behavior attribute
  
```

<Attribute>::=<Attribute name><Attribute type>
 <Attribute flag> [<Attribute weight>]
 <Attribute name>::=<String>
 <Attribute type>::=<String>
 <Attribute flag>::= { TRUE, FALSE} //the current state of attribute; TRUE means that the attribute is activated; FALSE means that the property is invalid.
 <Attribute weight >::=<Real number> // value range is [0,1]
 <Attribute table mapping rules>::=<The methods of achieving attribute>
 <Component table >::=<Component>
 <Component>::=<Component name><Component architecture>
 <Component architecture>::=<Component topology information table>
 <Component topology information>::=<Adjacent> | <Tangent> | <Separate>
 <Case table >::=<Case identifier>
 <Design process>::=<Functional requirements>
 <Mapping rules>
 <Functional requirements>::=<Attribute name>
 <Attribute value><Attribute value>::=<String>
 | <Values>
 <Mapping rules>::=<Rules table><Restriction table>
 <Rules table>::=<The rules mapped to product object class >
 <The rules mapped to case> // solving control information Sa
 <Rules>::=<Rules name><Premise><Conclusion>
 <Rules credibility >
 <Constraints>::=<String> | <Expression>

In functional model: (1) <Functional model class name> is the name of product object class dealing with certain kind of functional design requirements; (2) <Functional model type> represents the type of object class; (3) <Attribute list> describes all the attributes of functional model; (4) <Components table> describes the structure and geometry information related with functional model; (5) <Case table> records the total number of design examples and the case pointers, we can index an case in the case database by its pointer; (6) <Design process> describes the method to solve the design problem; (7) <Description> is used to express unstructured knowledge, those can not be expressed by using any data structure.

3.2 The Parametric model of products (Pm)

Using parameters to express users' needs is the combined effect of the carrier, function and behavior of a product, and is also the digital description of the technical characteristics of a product. Parametric model Pm is composed of the information model, geometric model and feature model form. Where,

$$\text{Information model} = \{Ta, Pa, Dc\}$$

Consistent with product category, Ta is the product type information, including all products sub-categories that parametric model Pm is able to handle.

Pa is the product technical parameter information, it is a property list whose initial value is empty and requires the user to provide. Pa is used to describe the parametric definition of the product category; but within the system, these numeric attributes associate with the function, structure and behavior of design objects through a number of mapping rules. Pa is complete for describing the product category as long as user provides all the information, then the detailed design can be completed step by step.

Dc is the design control information used to control solving process under parametric model, it is the reasoning process based on the parametric model, including: (1) The mapping rules from the technical parameters of the user's inputs to the feature model, thereby to complete the concept design of the product; (2) Finish the field knowledge of detailed design, such as: a variety of rules and methods designed for a component of the product.

Geometric model includes solid model, surface model and wireframe model, and is used to describe the geometric information of the products and their components. It is the basic information supporting the design process. Component class is the most basic design object class in the system, and it determines the function, structure and behavior of its super-class. If an object-oriented programming language is used, every information model can be linked to a geometric model respectively through a pointer.

Feature model is used to describe the typical information of the product class. Each product class has a feature model. As same as information model, feature model also contains a set of technical parameters attributes, while the attribute item (attribute name) and the number of the two models are not necessarily the same. In information model, the numeric attribute values of the list Pa need users to input, and the system will start the design process hereby. However, in practice, the description of technical perfor-

mance of a product made by user is usually not specific enough, which means that we can not ensure the completeness of information model and the accuracy of attribute values (what the user has given is usually a fuzzy interval) either, thus make design information incomplete and uncertain. At this point the feature model can make a difference.

Each attribute in feature model has its own attribute value. However, these attribute values are not real ordinary numbers, but fuzzy numbers or fuzzy interval. The set of fuzzy numbers roughly describes the common features of a certain product, such as the scope of power, volume and movement time of solid state relay etc. When the solution of a design problem is assumed to be a solid state relay, the set of fuzzy numbers will impose a series of fuzzy constraints and limitations on this assumed solution. Through these fuzzy constraints, you can determine whether to choose the type of solid state relay as an appropriate solution or not. This way of solving problems is similar to fuzzy pattern recognition, so at this time each feature mode can be regarded as a typical model, thereby the problem can be seen as a mode to be recognized. When the problem is recognized as a feature model, the product type and the common features of this product represented by this model are the conceptual design solution of the target product. The attributes of the feature model entirely depend on the technical parameter attributes that the user mentioned in the product design requirements. Some properties may not be in the attribute list of Pa, but they could be in the attribute group of the feature model. The attribute value (fuzzy number or fuzzy interval) of each attribute in feature model can be directly given by the experts, or obtained through searching a case from case database. The later one is known as the learning based on cases and analogy, and a kind of machine self-learning method. The discussion of this kind of problem is not our article's concern.

The BNF of product object class of parametric model is defined as:

$\langle \text{Parametric model} \rangle ::= \{ \langle \text{Parametric model class name} \rangle \langle \text{Parametric model type} \rangle [\langle \text{Parametric model super-class} \rangle] \langle \text{Attribute list} \rangle \langle \text{Attribute table mapping rule} \rangle \langle \text{Data list} \rangle \langle \text{Component table} \rangle \langle \text{Case table} \rangle \langle \text{Design process} \rangle [\langle \text{Description} \rangle] \}$

Where, (1) $\langle \text{Parametric model class name} \rangle$ is the name of product object class for dealing with parametric design requirements; (2) $\langle \text{Parametric model type} \rangle$ expresses the product type of a object class; (3) $\langle \text{Attribute list} \rangle$ describes all features of the in-

formation model in parametric model; (4) $\langle \text{Attribute table mapping rule} \rangle$ is the method of achieving attribute; (5) $\langle \text{Data list} \rangle$ records a variety of data required in parametric model; (6) $\langle \text{Components table} \rangle$ expresses geometric information related with the parametric model, which is used to describe the geometric model; (7) $\langle \text{Case table} \rangle$ records the total number and the case pointers related with the feature model; (8) $\langle \text{Design process} \rangle$ describes the solution strategy of the geometric model.

3.3 Case model (Cm)

Case model is a high level abstraction based on case database. A practical case contains two types of information: (1) information related with design conditions, such as the technical requirements, functional description and behavioral description of the product gathered from the user. These information describes the attributes related with design tasks and design requirements; (2) information describing design results, such as the generated methods, the application range of case, the structural parameters described the geometrical features and so on. These two types of information constitute a complete case model. Therefore, the case model can be defined as the conceptual model based on the design conditions and the design results.

The differences and connection between instance and the case model are: instance is the case model, whose abstract structure is filled with the certain attribute values, and it is the design result that satisfies certain design requirements and a realization of case model; while the case model is a general abstraction of all instances.

Instance and case model can be expressed as the following 3-triple:

$$\text{Case} = \{ \text{PS}, \text{FS}, \text{OS} \}$$

Where, PS is the parameter set, and the parametric description of all attributes (features) of the product. FS is a functional description set, describing the functional properties of the case. In a practical case, it includes the behavior, function and the relationship with the other components. OS is an entity set, and is a description of the geometric attributes and material attributes of the case.

In case model, PS, FS and OS sets are not filled with the specific content.

4. The organization and management of the design object knowledge

A product super-class (such as motors, electrical appliances and machine tools, etc.), can always be divided into several subclasses according to a certain principle, and normally each subclass can be divided furthermore. A product subclass without subclasses can be broken down into several components, and similarly each component can also be broken down into several parts. The hierarchy and decomposition like these constitute a tree, which describes the hierarchical division process of “product- parts- components”. All the nodes of the tree can be classified into three major layers (products layer, parts layer and components layer).

Functional model can only be divided into products layer. All functional models are grouped together, organized and managed in accordance with the topology structure.

Parametric model and case model can be divided all the way down into parts layer.

Be noted that, in the products layer, parametric model = information model + geometric model + feature model; beginning with the components layer, parametric model = information model + geometric model. In other words, feature model is only for the concept design of product, providing guidance and support to the parametric model in products layer.

All parametric models and case models were organized according to the tree structure. In this way, we can conveniently manage and use knowledge in favor of improving the reasoning efficiency of the system.

In addition, in order to improve the reasoning efficiency, some areal knowledge in the design can be encapsulated in the design object model rather than stored in a knowledge repository. For example, in the parametric models of the parts layer and components layer, we can only encapsulate the domain knowledge related with these parts and components. When such knowledge in a design is needed, you do not have to search the entire knowledge repository. In other words we could avoid excessive consumption of the system time. Knowledge repository is used to store meta-knowledge and the planning and to control knowledge of design tasks and the technical standards of product, etc.

5. The expression of cognitive knowledge-the planning and system reasoning of design task

The planning, controlling and solving process based on design mode is a process of information organizing, input, transformation, processing and output, which is a control tactics based on the realization of design mode.

The planning of design task is to decompose design task based on design model. In this paper, the task planning based on design mode decomposes the design task according to the following two ways:

(1) Product function decomposition. Quickly determine the type of product and complete the conception design by using the characteristic of the larger grading of function description.

(2) Product structure decomposition. Determine the geometric parameters of product parts and components and complete the detailed design by using the knowledge implicated in system.

The development of system based on the knowledge model in this paper must apply the two-stage control strategies. There are three objective inference engines under the source inference engine, which start reasoning based on the three kinds of design object model respectively. The three kinds of reasoning are known as the functional reasoning, parametric reasoning and case reasoning.

The product design information given by users can be divided into three categories: (1) Functional attribute information, which are the requirements of the function and behavior characteristics of product; (2) Technical parameters information, which are the requirements of the technical features of product; (3) Product type information, which is the requirement of product type. Based on the three categories information, source inference engine starts from three objective inference engines respectively. The system's reasoning process is shown in Fig. 2.

Both of functional reasoning and parametric reasoning can get the conceptual design solution. But from the view of the definition product, the functional description is in a higher level than the parametric description, so it is appropriate to start the functional reasoning first. Parametric reasoning is started based on the results of functional reasoning, which means that the matching of the feature model would be implemented in the various subclasses of the product class determined by functional reasoning at first. At this moment, design mode switches from functional model to parametric model.

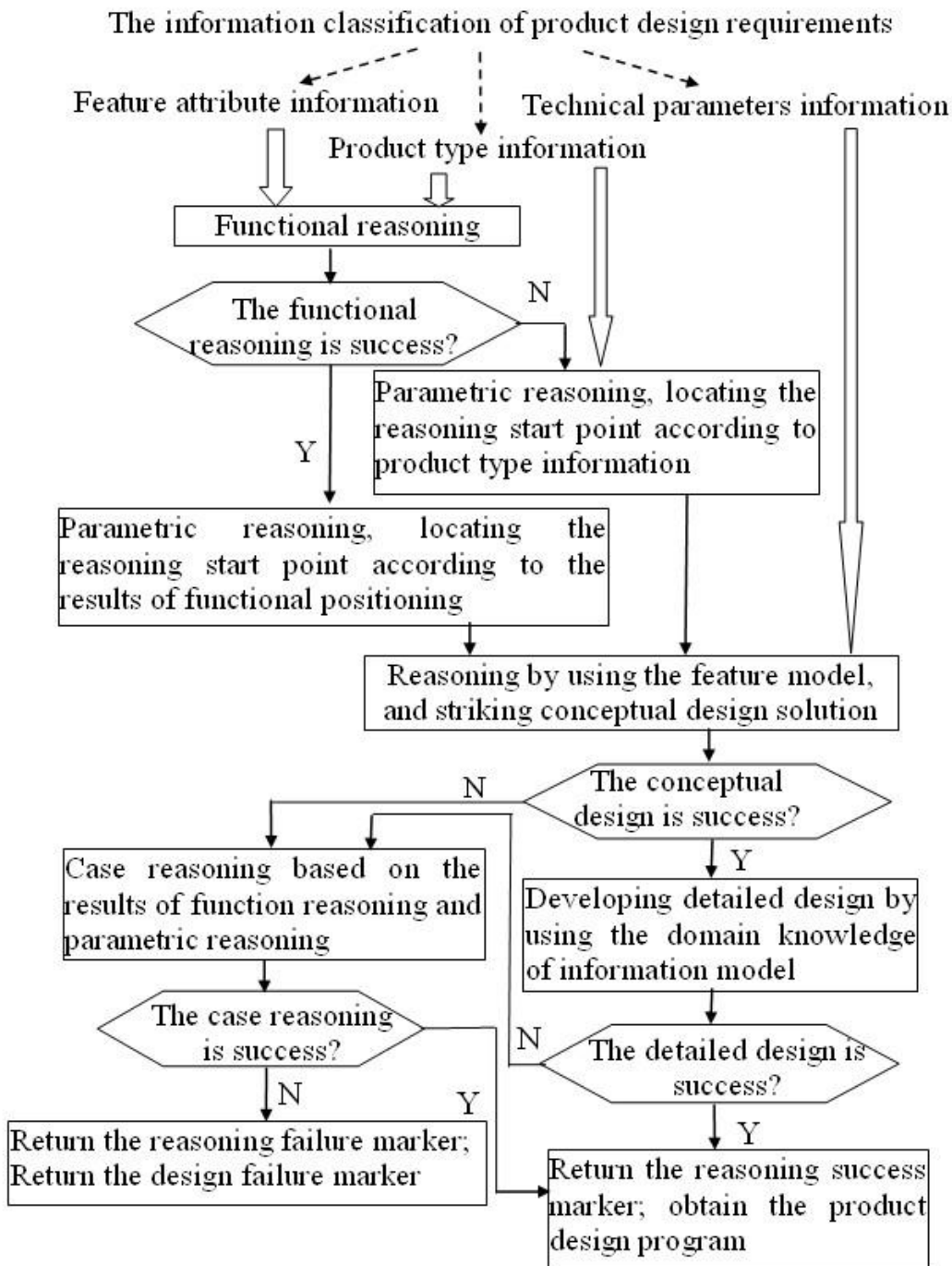


Figure 2 Design mode converter supporting DMOM

Although the comprehensive ability of parametric design model in solving a problem is strong and a large number of fields knowledge are encapsulated in the parametric models of products layer, parts layer and components layer, the parametric reasoning still has the possibility of interrupting when the design information given by user is incomplete. If so, case reasoning starts from the breaking point. And the result of case reasoning will be returned to the parametric model at this point, which makes the design process under the parametric mode resume. At this point, mode transformation conducts back and forth between the parametric mode and case mode.

Assumption: under the parametric design mode, the target design problem and the feature model of electromagnetic relay match successfully (the product type is identified as the electromagnetic relay). But parametric reasoning is interrupted due to the insufficient design information in the process of armature designing. At this moment, the measures taken by the system and reasons of taking such measures are described in the following contents.

Each parametric model defines an <case table>; <case table> defines a case and its identifier. All the identifiers are pointer types, pointing to the entry address of the various cases in the case repository. For the armature of the relay electromagnetic system, the <case table> of whose parametric model lists the accessible entries of all cases belongs to the same type stored in the case repository. After obtaining a case identifier, we can get the armature case through the following CasefindIndex method, and the design mode will switch from parametric mode to case mode. Then the system can get the geometry size of the armature by modifying the case, and the result will be returned to the parametric model of the armature. It is a process that the design model switches from the case mode to parametric mode.

The new design of armature is coded in system as a new case. While the cases are increasing in the system, and the case identifiers are increasing with them through the ReviseIndex.

```
Public // Keyword, public property
Property CaseIndex: TCase read CasefindIndex write
ReviseIndex;
// Define the case of armature as case class; define
CasefindIndex as a process of retrieving case; define
ReviseIndex as a process to modify case.
Procedure DesignArmaturepart (Sender: TDesign-
Method);
// Design process of armature
```

The design process of armature is an internal process in the process of solving parametric model, mainly to determine the geometric size of the armature according to the current design result of relay.

Functional model also defines an <case table>, whose role is similar to the <case table> in parametric model. Mode transformation carries out back and forth between the functional mode and case mode when the technical parametric information given by user is empty.

As the design model (DMOM) is based on design mode, the adaptive transformation among the different design modes is shown in Fig. 3.

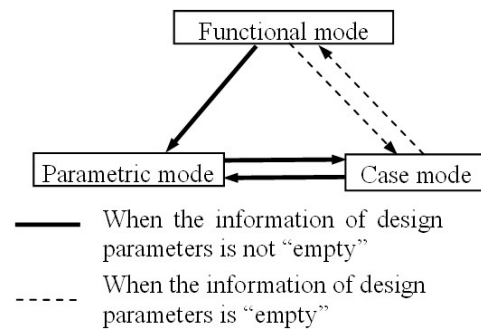


Figure 3 Design mode converter supporting DMOM

6. Case study

When users open the design program, whose interface showed in Fig. 4 can be seen.

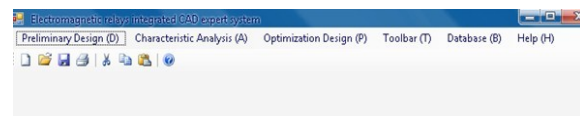


Figure 4 Design mode converter supporting DMOM

In this interface, the system menu gives the preliminary design, features analysis, optimization and other design options. This paper discusses the preliminary design of the product. When user enters the relevant design information, design process starts. If the user wants to suspend it temporarily, all information can be saved into a file during the process. In order to resume the process, you only need to open the corresponding file. Therefore, the user's input and the status information of the process are recorded in a design file.

When a user begins a new design task, the design requirements input interface shown in Fig. 5 pops up.

Design requirements interface contains two entries: function requirements and technology requirements. Among which, the function requirements input interface offers a number of functional requirements for a

user to choose; and the technology requirements input interface lists some parameters and product types for a user to choose.

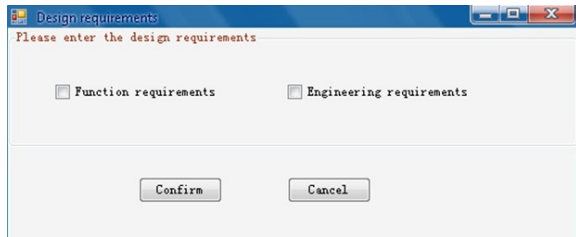


Figure 5 Design requirements interface

In the design requirements interface, if the user selected design requirements including functional requirements, the function selection input interface shown in Fig. 6 will appear.

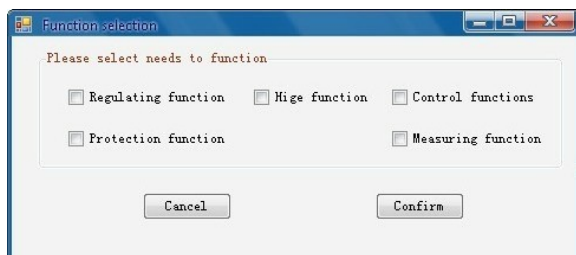


Figure 6 Function selection interface

The function selection interface is constructed by options. Users only select the functions the design object should have, whereas parameters are not necessary. If users do not choose any functions, the system will cancel the options of function requirements stored in internal records, after which the task planning no longer starts any process associated with function requirements. Process such as function reasoning can only be triggered when at least one function is selected. Therefore, users' selections have direct effects on the task planning and design control process.

The technology requirements input interface shown in Fig. 7 is a multi-input interface. Users can choose the required parameters based on their own interests, product type selection is also in this interface. The technology requirements interface also controls several other optional input interfaces, including input control circuit, output control circuit, and other technology requirements and so on.

The product type selection is a single input interface. The user can select any product type listed in Fig. 7. If user does not select, the system will choose the type of the product to be designed based on function reasoning; otherwise, some menus associated with

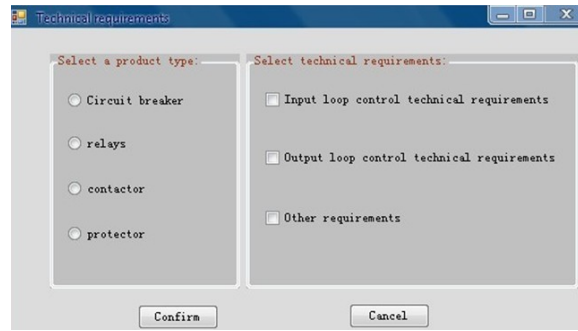


Figure 7 Technical requirements interface

the selected type will pop up. For example, if a user selects the relay as the type of the product to be designed, some of the interfaces related to relay will pop up immediately.

When a relay product is chosen, the interface of other technology requirements shown in Fig. 8 will pop up. In this interface, users could specify the necessary input items of the design, such as media pressure endurance, the maximum ambient temperature, etc; meanwhile, some optional requirements, such as movement time and release time, could also be specified in this interface.

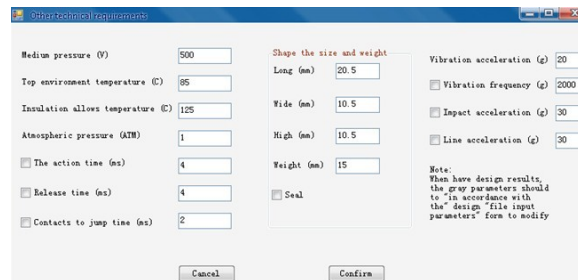


Figure 8 Other's technical requirements interface

After the user's input is given, based on which the system would plan all procedures and initiate the reasoning process to get the preliminary design. Under the condition shown in Fig. 8, output interface of the preliminary design of the relay is illustrated in Fig. 9.

7. Conclusion

In this paper, based on the entire product design field, the intelligent design knowledge modeling method is discussed. And the purpose of which is to explore a knowledge modeling method that has a certain commonality in the intelligent design of the majority electromechanical products, rather than only for a specific product. Among them, the model used for expressing the knowledge of design process is general, which

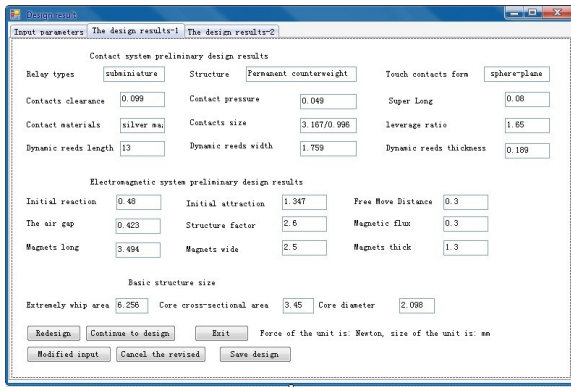


Figure 9 Interface of design results

is the design model (DMOM) based on design mode. Although the functional model, parametric model and case model used to describe the knowledge of design objects are associated with certain specific field knowledge, the models themselves still have a good applicability in the definition and information composition for most products. At present, some methods have already been applied in the expert system of electrical product design.

8. Acknowledgments

This work was supported by the National Natural Science Foundation (No. 60771069), the Hebei Province Natural Science Foundation (No. F2010000151) and China Postdoctoral Science Foundation (No. 201-00470778).

References

- [1] Kokotovich Vasilije, Purcell. Terry. "Mental synthesis and creativity in design: an experimental examination", In: *Design Studies*, vol. 21, no. 5, pp. 437-449, 2000.
- [2] Bonnardel, N.. "Towards understanding and supporting creativity in design: analogies in a constrained cognitive environment", In: *Knowledge-Based Systems*, vol. 13, no. 7, pp.505-513, 2000.
- [3] Hsiao Shih-Wen and Chuang Jiun-Chau. "A reverse engineering based approach for product form design", In: *Design Studies*, vol. 24, no. 2, pp. 155-171, 2003.
- [4] Rosenman M. A.. "An Exploration into Evolutionary Models for Non-Routine Design", In: *Artificial Intelligence in Engineering*, vol. 11, no. 3, pp. 287-293, 1997.
- [5] Li Lingling, Zhang Yong, Lv Qing and Liu Fengguo. "Intelligent Design of Investment Casting Mold for Turbine Blade Based on Hybrid Reasoning Method", In: *Low Voltage Apparatus*, no.5, pp. 10-14, 2007.
- [6] Zhang Xiangjun, Gui Changlin. "Gene Models in Intelligent Computer-aided Design", In: *Chinese Journal of Mechanical Engineering*, vol. 37, no. 2, pp. 8-11, 2001.
- [7] Finnie Gavin and Sun Zhaohao. "R5 model for case-based reasoning", In: *Knowledge-Based Systems*, vol. 16, no. 1, pp. 59-65, 2003.
- [8] Lingling Li. "The Study on the Construction Method of Design-Oriented and Evaluation-Oriented Expert System for Relay", In: *Tianjin, Hebei University of Technology*, 2004.
- [9] Umeda Y. and Tomiyama T.. "Functional Reasoning in Design", In: *IEEE Expert*, vol. 12, no. 2, pp. 42-48, 1997.
- [10] Zhao Zhen, Lu Shi-jun, Wang Ai-chen and Peng Ying-hong. "Knowledge-Based Integrating Modeling Method for Stamping Product and Process", In: *Journal of Shanghai Jiaotong University*, Vol.37, No.2, pp.153-156, 2003.
- [11] Jiang Ruisong, Zhang Dinghua, Wang Wen-hu and SHI Hua-jie. "Intelligent Design of Investment Casting Mold for Turbine Blade Based on Hybrid Reasoning Method", In: *Foundry*, Vol.57, No.9, pp. 916-919, 2008.
- [12] Gayretli A. and Abdalla H. S.. "An Object-oriented Constraints-based System for Concurrent Product Development", In: *Robotics and Computer-Integrated Manufacturing*, no. 15, pp. 133-144, 1999.