A Knowledge-Based Prototype System to Support Product Conceptual Design

D. X. Cao[1], M. W. Fu[2]

[1]Hebei University of Technology, caodx@hebut.edu.cn
[2]The Hong Kong Polytechnic University, mmmwfu@polyu.edu.hk

ABSTRACT

A key step in conceptual design is to develop specifications for the required behaviors or functions and transforms the specifications into the description of component configuration. In this paper, a design synthetic approach is proposed and employed to guide the design process via behavioral reasoning and to obtain an iterative transforming process. Firstly, the functional representations and design parameters according to the design requirements of a product are presented and a behavioral matrix model by using the bond graph fundamental elements is established. Secondly, a knowledge modeling language is presented for behavioral reasoning. A design synthesis approach is described in detail in such a way to transform the matrix model for generating functional means tree and to obtain multi-solutions by artificial intelligence. A prototype system is then developed for computer-aided conceptual design. Finally, a design synthesis case of a fast clasping mechanism used in a machine center is presented to show its application.

Keywords: behavioral matrix, bond graph, functional means tree.
DOI: 10.3722/cadaps.2011.129-147

1 INTRODUCTION

In the last a few decades, computer-aided design (CAD) has dramatically changed the way that engineers design products. A marketing survey reported that computer-aided design takes above 70% share in manufacturing industry and more than 95% of CAD software systems on the marketplace are 2D or 3D drafting packages and surface or solid modeling packages [1]. It is apparent that the current CAD system is mainly used in embodiment design and detail design phases, mainly for graphical representation and geometric modeling. Using CAD software in the early design stage is still a non-trivial issue as the conceptual design is now still far from being computerized due to the intrinsic nature of the design problems [2]. The conceptual design, however, is the most crucial design phase in the whole life cycle of a product as it determines the principle schemes, component configuration and the used materials. This up-front process is critical as 20% of design activities at the stage commits to about 80% of product cost and product quality issues. With the advance of computing capacity and the availability of techniques of artificial intelligence (AI), the development of new generation computer-aided systems with more genuine design abilities to support early design. activities has become feasible [3]. This is thus a strong motivation for researchers to explore and develop conceptual design methods and computerize them in such a way to reduce engineering cost, increase product quality and shorten time-to-market. In addition, the research is also further driven by the fact that manufacturing and other downstream activities cannot make up for the poor design solutions or the developed product concepts [4]. Consequently, increasing attentions have been paid to the design activity at conceptual design level [5-6].

Many different models have been developed to describe the engineering design process from market needs to production. An effective design process model is crucial to implement conceptual design at early design stage. Suh [7] developed the Axiomatic Design Method by taking two axioms as design criteria for generation of design solutions in different domains, including customer, functional, physical and process domains. Welch & Dixon [4] transformed the product functional requirements into behavior description, and matched the physical components to behavior graph in such a way to generate multi-solutions aided by artificial intelligence. Sharpe & Bracewell [8] converted the functional structure into the functional means tree, determined the bond model, realized the system dynamic simulation and finally developed the Schembuilder software for mechatronics system from conceptual design to detailed design. Tay et al. [9] presented a systematic behavior based on the bond graph, and carried out the behavior and configuration transformation by using the genetic algorithm to generate design variables to realize system analysis, synthesis and evaluation. Gui & Mantyla [10] developed a mechanical design prototype to implement a top-down modeling system through the combination of a functional model, a connector-component structural model, and a process-oriented bond graph model.

In addition, Zhang et al. [11] presents a prototype knowledge-based system to support the synthesis of conceptual design. By using the knowledge and physical behavior from a desired function or desired behavior, a functional model that represents the causal relationships is created. When a function cannot match correspondingly to a behavior, it will be automatically decomposed into sub-functions via decomposition rules. Gero [12] proposed a design prototype schema that includes function, behavior and structure, to represent both the design objects and the related design knowledge. In conceptual design process, it is important to explore the full spectrum of available design solutions. In addition, the errors made in conceptual design stage have been proven to be extremely costly and time consuming for correction in the later product development stage. Therefore, various knowledge synthesis methods need to be developed for conceptual design solution generation.

In tandem with this, Chakrabarti & Bligh [13] developed appropriate representations and reasoning procedures for synthesising solution concepts by using a set of primary functional elements and their combination rules. These synthesis procedures can produce an extensive set of solution concepts, in terms of their topological as well as the spatial configurations, to a given design problem. Roy et al. [14] proposed a design synthesis method to guide the design process through product specification, and functional requirement representation, artifact modeling and the artifact behavior and tolerance description. In this problem-solving process, designer's knowledge and expertise are employed to implement in reasoning. Gorti et al. [16] developed an object-oriented representation for product design process, in which design operators may be updated and transformed in the design context. It supports the arrangement of design automation from manual design to automatic design with various reasoning techniques, e.g., inheritance, rules, and constraints, etc. Malmqvist [30] extended the function-means tree, which is based on the chromosome model for product modeling to concurrently document the design history. The extended function-means tree model includes functional requirement, means, objectives, and constraint objects, and solved by, alterative solutions, and requirements on, and has influence on relations.

The focus of this paper is on the strategy and methodology development to support a knowledge-based prototype system (KBPS) for conceptual design. A fundamental assumption is that the conceptual design cannot be fully automatic. However, a computer-aided system can guide designers to conduct creative design at the early stage [17]. To develop such a KBPS, it is necessary to firstly explore the related issues, such as the characterization and abstraction of conceptual design problems, the representation of design objective and other related design concepts.

In this paper, the functional representation and design parameters in preliminary design are first presented and a matrix model to incorporate an explicit behavioral reasoning for a specialized class of conceptual design problem is then developed. A knowledge structure prototype that realizes a component modeling language, a hierarchical representation of component knowledge and the component matching process is presented. The strategies for knowledge synthesis and the methods of problem solving used to implement a prototype design environment are then discussed. In addition, how to transform a behavioral matrix to produce functional means tree is also described. An artificial intelligent technique is applied to aid the computer-aided conceptual design. To verify the developed KBPS, a case study is used to verify the efficiency of the developed prototype design system.

2 FUNCTIONAL REPRESENTATIONS AND DESIGN PARAMETERS

The useage value of a product is embodied by its functions. The function is defined as the implementation of an act or carrying out a task. Researchers consider function as a transformation between the input and output of material, energy or signal [18]. For example, Pahl & Beitz [19] characterized function as a general input/output relation of a system. Function could be used for conveying design intent. This is shown in the design process developed by Kirschman et al. [20]. They presented a taxonomy of elemental mechanical functions and derived four basic types of functions related to the concepts of motion, power/matter, control and enclosure, in which each can be used with decomposition techniques.

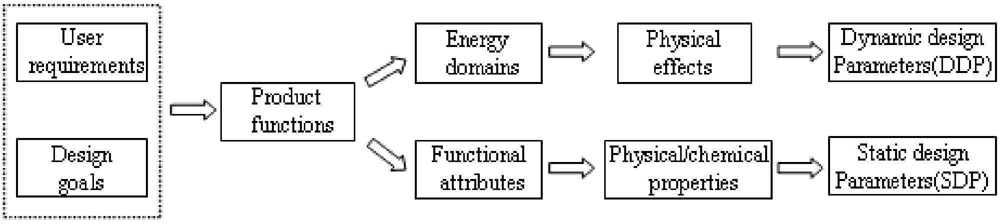


Fig. 1: Functional structure.

In this research, the user requirements are combined with design goals, and the product functions are then determined. These functions can be decomposed further, based on their corresponding physical phenomena to function parameters in different energy domains. These variables are called dynamic design parameters (DDP), such as force, torque, displacement, etc. They conform to the specified natural laws and scientific principles. On the other hand, the functional attributes correspond to static design parameters (SDP), such as, material, color, dimension, and cost, etc. Finally, the functional structure is established and shown in Fig. 1, where, the energy domain includes mechanical, hydraulic pressure and electric systems. The physical effects contain Hooke's Law, Newton's Law, Friction Law, etc. They correspond to physical variables in a defined energy domain. For example, the parameters that Hooke's Law corresponds to its physical variables in a mechanical translation system in Tab.1 are force and displacement.

Tab. 1: Conceptual design process.

Physical effect	Energy domain	DDP input	DDP output	SDP
Hooke's Law	Energy → Elastic system → Energy	Force	Displacement	Metal/extension

Therefore, the energy flow in a mechanical system is abstracted as a transformation between effort variable (S_e) and flow variable (S_f). S_e and S_f corresponding to different energy domains are described as follows.

$$S_e=\{\text{force } (F), \text{ torque } (T), \text{ hydraulic pressure } (P), \text{ voltage } (V)\}$$
$$S_f=\{\text{velocity}(V), \text{ angular velocity}(\omega), \text{ discharge}(Q), \text{ current}(I)\} \quad (1)$$

In addition, the functional attribute (F_a) corresponding to SDP is represented as.

$$F_a = \{\text{material, color, shape, dimension, cost, ...}\} \tag{2}$$

The Functional Set (FS) can be defined as tri-nary relations, then

$$FS = \{< (S_e, S_f), F_a > | < S_e, S_f > \in \text{mechanical} \cup < S_e, S_f > \in \text{hydraulic} \cup < S_e, S_f > \in \text{electrical} \cup F_a \in SDP\}$$

In order to realize the functional requirements of mechanical product design and produce physical components, the impedance (R), capacitance (C) and inductance (I) components corresponding to different energy domains are shown in the following:

R={translation friction (f_t), rotational friction (f_r), liquid resistance (R_l), resistance (R_r)}

C={spring (k), twisting spring (E), liquid capacity (δV), electric capacity (C_e)}

L={mass (m), inertia (I), liquid inductance (δq), inductance (L_i)}

These components are combined together and the energy transformation of a physical system is described by transformer TF, gyrator GY, common effort junction "0" and the common flow junction "1". This is the fundamental theory of power bond graphs [21].

In general, a system could be decomposed into multi-energy flow subsystems for different domains, such as mechanical, electrical, and hydraulic subsystems. In system working process, there exists energy generation, consumption, storage, and transformation. Using Eq. (1), eight foundational dynamic design parameters such as S_e, S_f and their differential or integral variables [4] are considered in a multi-energy coupling system, and each functional parameter (whether efforts or flows) has six attributes shown. Therefore, S_e or S_f is a 6-tuple.

$$E_e \text{ (or } F_f) = \{\Delta, CL, ED, PV, OT, PO\} \tag{3}$$

Where, E_e (or F_f): Dynamic design parameters (DDP) effort or flow.

Δ: Time operator, it means the DDP changes along with time. For instance, the output parameter is displacement (flow), which is a fundamental parameter's (velocity) integral.

CL: Classifies, If the parameter is in motion, it is power. If the parameter is static, it is signal.

ED: Energy domains, including mechanical translation, mechanical rotation, hydraulic, and electric.

PV: Parameter value.

OT: X, Y, Z orientations.

PO: Absolute spatial positions.

According to the physical parameter attributes, any DDP can have 6×1 matrix as follows.

$$[\Delta, CL, ED, PV, OT, PO]^T \tag{4}$$

For a general mechanical product, different energy categories correspond to different physical parameters. A mechanical system contains translation and rotation, in which both sub-systems correspond to their physical parameters, while the hydraulic and electrical systems also correspond to their physical parameters. Tab.2 gives the 16 kinds of basic design variables and their attributes of an engineering system [22].

3 BEHAVIORAL MATRIX REPRESENTATION OF DESIGN PROCESS

For a mechanical product, its technical system can be described by a transforming graph as shown in Fig.2 (similar to 3-port elements of bond graphs), where the input and output parameters correspond to the effort and flow variables (8 basic parameters), and the optional parameters consist of the basic elements of the bond graphs.

By using matrix (4) and Fig.2, the input and output relations in a mechanical system can be described by the following matrix equation:

Tab. 2: Basic design parameters and their attributes of an engineering system in DDP

Design parameters	Time operators	Classifies	Energy categories	Parameter values	Orientation	Position	Effect or flow
F (force)	0	P/S	Translation	0	(0 0 0)	(0 0 0)	Effort
v (velocity)	0	P/S	Translation	0	(0 0 0)	(0 0 0)	Flow
a (acceleration)	Differential	P/S	Translation	0	(0 0 0)	(0 0 0)	Flow
S (displacement)	Integral	P/S	Translation	0	(0 0 0)	(0 0 0)	Flow
T (torque)	0	P/S	Rotation	0	(0 0 0)	(0 0 0)	Effort
ω (ang velocity)	0	P/S	Rotation	0	(0 0 0)	(0 0 0)	Flow
ε (angular)	Differential	P/S	Rotation	0	(0 0 0)	(0 0 0)	Flow
θ (ang-displacement)	Integral	P/S	Rotation	0	(0 0 0)	(0 0 0)	Flow
P (liquid force)	0	P/S	Hydraulic	0	-	-	Effort
V (volume)	0	P	Hydraulic	0	-	-	Flow
U (flow velocity)	Differential	P	Hydraulic	0	-	-	Flow
Q (flow volume)	Integral	P	Hydraulic	0	-	-	Flow
E (voltage)	0	S	Electrical	0	-	-	Effort
A (electric current)	0	S	Electrical	0	-	-	Flow
f (frequency)	0	S	Electrical	0	-	-	Flow
q (electric charge)	Integral	S	Electrical	0	-	-	Flow

$$[\text{Output parameter}] = \boldsymbol{T}\,[\text{Input parameter}] \quad (5)$$

Here

$$\boldsymbol{T} = \begin{bmatrix} \Delta & & & & & \bigcirc \\ & \text{P (or S)} & & & & \\ & & \text{ED}\rightarrow\text{ED} & & & \\ & & & \text{PV} & & \\ & & & & \text{OT}\rightarrow\text{OT} & \\ \bigcirc & & & & & \text{PO}\rightarrow\text{PO} \end{bmatrix} = \text{diag}\,[\Delta, \text{P (or S)}, \text{ED}\rightarrow\text{ED}, \text{PV}, \text{OT}\rightarrow\text{OT}, \text{PO}\rightarrow\text{PO}] \quad (6)$$

Where, T is the so-called the behavioral matrix. According to the causal relations of the bond graphs and the different input or output variables, the optional parameters and the transforming matrix of the basic elements can be obtained. In the following, the four transforming matrix T cases are discussed [23, 24].

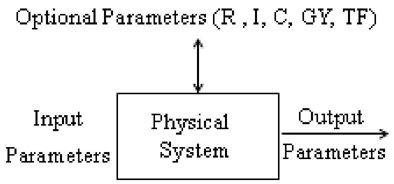


Fig. 2: Transforming graph

3.1 Input flow → Output effort

By using Eq. (5), an input flow and output effort system is described as.

$$E_o = \boldsymbol{T}_0 F_i \tag{7}$$

where, E represents effort while F represents flow; Footnote 'i' indicates the input parameters while 'o' refers to the output parameters. In addition, the resistant transform T_{0R} can change the technical system PV and consume system energy, but cannot change other attributes, in which the system causal relations depend on the input or output parameters, as shown in Fig. 3.

$$\boldsymbol{T}_{0R} = \text{diag}\,[\Delta, P \rightarrow P, ED \rightarrow ED, PV, OT \rightarrow OT, PO \rightarrow PO]$$
$$= \text{diag}\,[1, 1, 1, R, 1, 1]$$

where '1' means the attribute value, which are not changed after adding the resistant transformation.

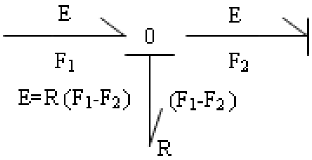


Fig. 3: Bond graph model added R transformation

According to the bond graphs, the capacitance transform $\boldsymbol{T}_{0C}$, inductance transform $\boldsymbol{T}_{0L}$ and the gyro transform $\boldsymbol{T}_{0GY}$ can be obtained and shown below.

$$\boldsymbol{T}_{0C} = \text{diag}[d/dt, 1, 1, 1/c, 1, 1]$$
$$\boldsymbol{T}_{0L} = \text{diag}[dt, 1, 1, L, 1, 1]$$
$$\boldsymbol{T}_{0GY} = \text{diag}[1, 1, ED \rightarrow ED, 1/r, OT \rightarrow OT, PO \rightarrow PO]$$

3.2 Input effort → Output flow

Based on Eq. (5), the technical system from input effort to output flow is described as follows.

$$F_o = \boldsymbol{T}_1 E_i \tag{8}$$

According to the rule of input flow → output effort, the transforming matrix $\boldsymbol{T}_{1R}$, $\boldsymbol{T}_{1C}$, $\boldsymbol{T}_{1L}$ and $\boldsymbol{T}_{1GY}$ shown are obtained in the following.

$$\boldsymbol{T}_{1R} = \text{diag}[1, 1, 1, 1/R, 1, 1]$$
$$\boldsymbol{T}_{1C} = \text{diag}[dt, 1, 1, C, 1, 1]$$
$$\boldsymbol{T}_{1L} = \text{diag}[d/dt, 1, 1, 1/L, 1, 1]$$
$$\boldsymbol{T}_{1GY} = \text{diag}[1, 1, ED \rightarrow ED, r, OT \rightarrow OT, OP \rightarrow OP]$$

3.3 Input flow → Output flow

The transformer T_{0TF} realizes energy transformation and changes the direction and position of functional parameters.

$$F_o = \boldsymbol{T}_{0TF} F_i \tag{9}$$

Where

$$\boldsymbol{T}_{0TF} = [1, 1, ED \to ED, 1/r, OT \to OT, OP \to OP]$$

3.4 Input effort → Output effort

By employing the method of Input flow→output effort, the transform of input effort → output flow are in the following

$$E_o = \boldsymbol{T}_{1TF} E_i \tag{10}$$

Where

$$\boldsymbol{T}_{1TF} = [1, 1, ED \to ED, r, OT \to OT, OP \to OP]$$

The purpose of representing the bond graph fundamental component as a diagonal matrix is to conveniently conduct design synthesis and search for multi-solutions. On the basis of the input or output parameter and different transmitting functions, the transforming matrix $\boldsymbol{T}_{0TF}$, $\boldsymbol{T}_{1TF}$, $\boldsymbol{T}_{0GY}$ and $\boldsymbol{T}_{1GY}$ can transform or transmit energy. Meanwhile, they can also change the position and direction of DDP. The transforming matrix $\boldsymbol{T}_{0L}$, $\boldsymbol{T}_{0C}$ and $\boldsymbol{T}_{1L}$, $\boldsymbol{T}_{1C}$, however, don't transmit system energy and the position and direction of DDP. But their differential and integral causal relations may affect system energy variables and frequent performance [21, 22].

4 A KNOWLEDGE STRUCTURE PROTOTYPE

Knowledge should be structured based on how it supports the reasoning needed. In this research, a formal knowledge representation is established for knowledge acquisition. The acquired knowledge should be structured into an available form according to its inference requirements. In development of knowledge-based systems, the conceptual modeling representation is crucial. In this process, the knowledge in the specific application domain is modeled and represented by using a modeling language. It is of importance for the success of a knowledge-based system that the domain knowledge should be represented correctly and completely in order to insure the data consistency and continuity [24].

In design solution generation, the domain knowledge is rather unbounded depending on the problem-solving types. It is thus difficult to provide comprehensive and complete domain knowledge. In this research, both of the frame-based and rule-based knowledge representation methods are used as it can effectively describe the knowledge by using a hierarchical knowledge representation.

4.1 Component Base Representation

Components are the basic structuring elements of a product. They implement a certain function through the interaction of several components or a single component. These components may respectively correspond to the specified configurations, such as mechanical transmission devices, electrical components as well as hydraulic components. Designers always pursue how to put these components into a repository in the light of some rules so that they may use these components efficiently to conceive mechanical product schemes at the early design stage. Roy et al. adopted the functional representation, form representation and behavioral representation to realize the component modeling [14]. They considered components can have two different varieties: either primitive or composite. The composite components are those whose internal substructure is represented explicitly by a set of much more detailed, lower-level sub-components, whereas primitive components are simple. The component signs, behavioral matrixes and performance targets are expressed with the aid of mathematics symbols. A component base with the frames is built. According to bond graph, one-port prototype components for the general engineering system are structured and shown in Tab.3. The two-port prototype components according to the mechanical and hydraulic system are given in Table 4. They can change or transform energy from one domain to another domain. Therefore, they are called transducer [21]. It is a fundamental task to describe these components formally for product conceptual design.

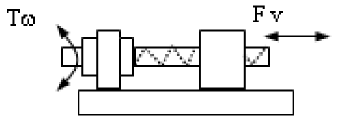
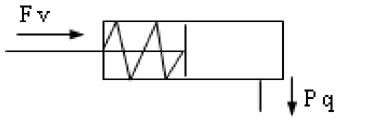
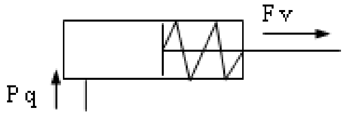
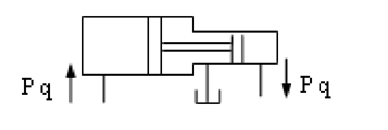
Then, a kind of component modeling language is represented as follows:

(Comp_name, Beha_matrix, Func_attribute, Eval_factor, Comp_rule) (11)

Tab. 3: One –port prototype components.

Energy Domains / Prototype Components / Component symbols	Mechanical system: Translation	Mechanical system: Rotational	Fluid system	Electrical system
T_R	Friction m	Friction	Liquid resistance R	Resistance
T_L	Friction M	Inertia I	Liquid inductance δq	Inductance
T_C	Friction k	Twisting spring E	Liquid capacitance δV	Capacitance

Tab. 4: Two-port prototype components.

Output energy Domains / Prototype component / Input Energy domains	Mechanical	Hydraulic
Mechanical		
Hydraulic		

Beha_matrix: Each component corresponds to a slot sign and different signs form a comprise symbol base. These components are generally described by common signs (such as $\boldsymbol{T}_R$, $\boldsymbol{T}_L$, $\boldsymbol{T}_C$ shown in Table 3). Some signs correspond to several behavioral matrixes, for example, a gear-drive component transform is torque→torque, it is designed as $\boldsymbol{T}_{1TF}$. If the component transform is angle velocity→ angle velocity, it is designed as $\boldsymbol{T}_{0TF}$. Both of them are put into component base. Furthermore, Beha_matrix presents the change of attributes about the component input and output DDP, such as Δ, OT, PO, as follows [4].

Time operator of input design variable Δ: Rotation, it is concerned with time.
Time operator of output design variable Δ: Rotation, it is also concerned with time.
OT's change: (180 0 0)
PO's change: (1.0, 1.0, 0.0)

Func_attribute: the functional attributes corresponding to SDP include material, color, shape, cost, dimension, etc. They are stored in database as the form of rules.
Eval_factor: It represents performance evaluation, including motion precision, energy efficiency, manufacture cost, structure compactness, and the complexity. Each factor value is limited within 0-1, which is relative to the other prototype components in the knowledge base. All these factors are assigned subjectively based on designer's experience [25].
Comp_rule: These rules are the IF-THEN conditional sentences which consist of a series of conditions and conclusions. Some rules can be treated as a rule subset to be attached as an independent design

object, such as matching rules, solving rules, inferring rules, etc. They control and decide the result of the system inference.

This modeling language representation can describe not only the structure characteristics of components, but also their behavior characteristics. It is thus suitable for the engineering design environment. Its data is taken as the kernel of the system. The scheme design, the knowledge-based system, and the configuration design are operated by directly visiting the database. It can easily solve the problem of data sharing among the systems.

4.2 Hierarchical Frame Structure

In the frame representation, all the information is embodied by objects. The object may be regarded as a concrete physical component or its function. The complex knowledge can be represented by a hierarchy of a chained structure. A hierarchical frame is used to help designers express component knowledge in a natural form, which can then be automatically converted into a tree structure.

An object-oriented frame structure adopts two ways for linkage including tree structure and chain structure. With the combination of the tree and chain, the data structure facilitates the searching and maintaining the database. In order to realize an object-oriented database, the tree node and slot node must be defined. The tree is suited for the data of a hierarchic structure. The hierarchic relations among the frame classes are formed by tree structures. Every node of the tree represents a frame class. The linkage between the tree nodes uses the chain linking approach. The node is defined as follows:

```
Define FMtreenode
Comp_*frame_name;
Fmtree_*parentnode;
Fmtree_*childnode;
Fmtree_*rootnode;
FC_*slotnode_list;
```

A frame corresponds to a tree-like data structure whose elements are called slots. The slots have names and accommodate various information contents. In the slots of a frame, the simple values are assigned by comparing with other frames or the slot value is computed from other information. The slots may also be left unfilled and have a default value, or be filled by inference. With the aid of inference, the frame class may inherit the values from its super-class frame. The slot node must be inserted in the frame nodes. The slot node is defined as follows:

```
Define FCslotnode
Comp_*slotname;
FC_*valueclass;
FC_*inheritancevalue;
FC_*attributevalue;
```

A frame corresponds to a component, while a component corresponds to several slots and each slot is of several values and different attributes. Therefore, the mechanical, electrical and the hydraulic component frames could be structured into the frame network structures. Frames are very useful in handling the complex knowledge structures by providing flexibility and accessibility to design knowledge. An integrated frame network structure is used to realize knowledge acquisition, knowledge transformation and functional reasoning [26].

An object not only includes meta-data, but also operating rules. In the object-oriented data model, the objects and their relations can be directly represented by the concepts of attribute classes, sub-classes and super classes. In a frame class, the structure is provided by inheritance tree with linking nodes. It can be located by using a frame through the subclass or upper class. Slot provides the data structure which stores the special attribute message of a frame class, including relation slot, attribute slot, method slot, and rule slot. With the aid of inference, the inherence value and the default value can be inherited from the classes or upper class, and passed on to the subclasses.

Therefore, the frame class of component base is described by using the object-oriented technique as follows.

```
Class Frame_name {
Beha_matrix (tim_operator, para_class, ener_domain, para_value, para_orient, spat_position);
        Slot_ structure {
```

```
            Time_operator (in_variable, out_variable);
            Ener_domain (Translation, Rotational, Hydraulic, Electrical);
            Phys_effect (Newton's law, Revnolds's law, …);
            Orient_matrix (SParallel, OParallel, Perpendicular, General );
            Const_matrix (Continuity, Linearity, Reversibility);
        }
    Func_attribute {
        Slot_value (Materials, Shape, Color, Cost, … )
        }
    Eval_factor {
        Slot_value (Mot_precision, Ene_efficiency, Man_cost, Str_compactness, Des_complexity );
        }
    Comp_rule {
        If (Comp_attribute is resistant, AND it works in Trans_system)
        THEN (Carry out T_L, T_C and T_TF TRANSformation)
        …
        }
}
```

4.3 Matching Process and Operation

As a special attribute message of an object is kept in a slot, an object-type can easily be defined according to the attributes of slots. It is thus convenient to define the properties of the object-type. For example, a gear has the moment of inertia which is defined by its axis's orientation and position. A gear-drive is composed of a pair of gears. By using the component modeling language, the component type can be easily described. Therefore, the properties of objects can be completely compared by these conditions to realize matching. The issue, however, is how well the object matches the object-type. Gelder [27] considered the object matching conditions of the object-type, which may match the object-type better than an object matching only by a few conditions of the object-type. An object matching all conditions of the object-type is called a prototype object. Some languages like semantic networks, frames, object-oriented modeling language are all used in matching operations. Zhang *et al* [11] adopted a distinct solution search strategy through scanning behavior base to search for the behavior matching. If no matching behavior is found, the desired function will be automatically decomposed into sub-functions by means of domain specific decomposed rules. Chiou & Kota [28] devised a matrix representation that not only captures the nature of an object, but also enables the automatic decomposition of a given task into simple sub-tasks. Meanwhile, they adopted two-level matching operations to search for the candidate components. The first level is used to describe the function of components and the direction of I/O axis, while the second level is used to describe the features and OPV. If no feasible solution is matched, the original object will then be decomposed into a series of sub-functions.

If the object has the properties satisfying the definition of the object type, the object matches the object type and the component is consequently selected. Experience indicates that a lot of design knowledge can be represented in this abstract schema of matching an object to an object type. In KBS development, the definition of an object type needs to be modeled and the knowledge about the object to be stored in database in order to implement the matching. The matching process is described as follows.

- To distinguish the object function based on the component modeling language, such as, behavioral matrix and attributes.
- To abstract the object prototype characters and attributes, such as, input/output time operator of design parameters, energy domain, DDP/SDP attributes.
- To define the object-type by using the component modeling language, some Eval_factor and Comp_rule can be used to reduce the potential candidates.
- To match the behavioral matrix to object-type for determining candidate matrix symbols based on Eq.(7)-(10), some rules are used to select the suitable behavioral matrixes.

• If an object matches the object-type that corresponds to a component frame class, the means is determined and the component is selected from the component base. Otherwise, it needs to decompose the object into sub functions and to conduct the next level matching operations, as shown in Fig. 4.

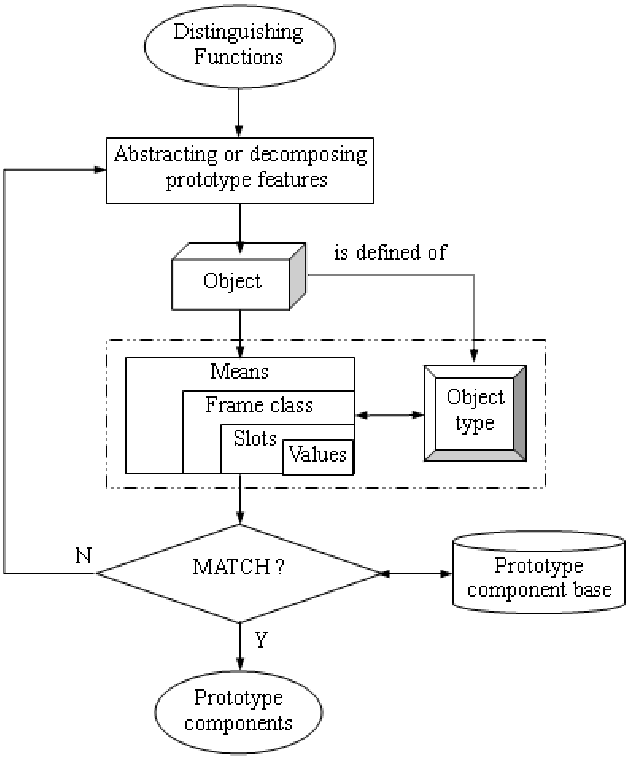


Fig. 4: The matching process of the knowledge-based system.

5 A KNOWLEDGE-BASED PROTOTYPE SYSTEM REALIZATION

The key task to realizing a knowledge-based system is to find a suitable process representation for design solution generation. The existed approaches, such as design catalogue, decision tree, objective tree, functional means tree, etc., have been used by most of the researchers in engineering design. They can generate the effective design solutions with the aid of various inference techniques [29]. Therefore, it is important to develop an efficient knowledge modeling representation and reasoning methods for design decision-making.

5.1 Representation of Functional Means Tree

The functional means tree is the direct representation of functional decomposition. It consists of fundament elements that contain functions and means. It also belongs to a kind method of AND/OR tree representation. The functional solutions that may be realized by one or more means are commonly called as means. The sub-function is realized by two means that they form an OR relation, as shown in Fig.5 (a). Means is embodied by one or more sub-functions, and is completed by two sub-functions that they form an AND relation in Fig. 5(b). Malmqvist [30] extended the functional means tree further and included the constraints and objectives as shown in Fig.6 (a).

In this paper, the functional means tree is first represented based on functions and means. Its function is then extended into an effort-flow transformation and the means is embodied by one or more behavior matrixes, i.e., effort-flow transformation is realized by the bond graph fundament components (means), as shown in Fig.6 (b).

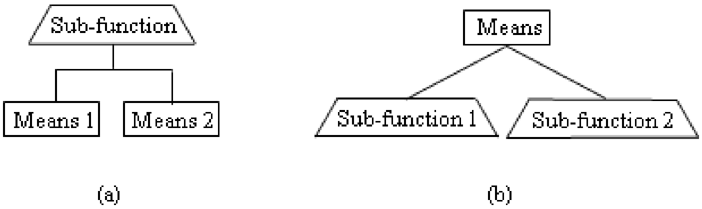


Fig. 5: The basic structural block of functional means tree.

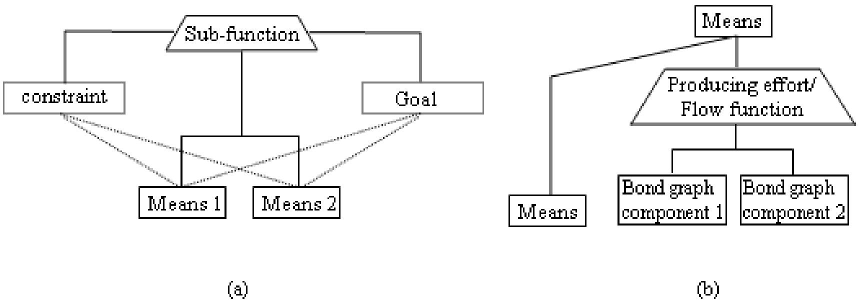


Fig. 6: The basic structural block of functional means tree after extension.

5.2 Generation of Design Schemes

The process of behavioral reasoning is the course that the functional means tree is established step by step. The method of top-down is adopted to decompose functions layer by layer. The overall function is decomposed into the means that may be realized, and the means is also decomposed further and embodied by sub-functions. Eventually, the means is obtained in correspondence with a series of entities, that is, the fundament components. Function and means include different layers, and they are linked with line segments between layers and layers. For the bottom layer determination, Sturges [31] considered when the sub-functions are decomposed into the supported function should stop. The supported function is a general part, process or sub-function structure, and so on. Stone et al. [32] applied the functional basis as an approach to identifying the product architecture. They gave the functional model a common vocabulary and identified a stopping point for decomposition by specifying that the function and flow words are chosen from a certain level. In this paper, the bottom layer means corresponds to the components in knowledge base.

At first, it is the key step to generate design parameters after the design requirements are abstracted, which include the function parameters, input/output parameter relations and their attributes in a mechanical system. It is then to determine the behavioral matrix model as Eqs. (7), (8), (9) and (10). One or more behavioral transforming matrixes are added into the system to satisfy function requirements. Sometimes behavioral matrix arrangement does not match the actual mechanical system components. This needs designer to add the behavioral matrix into the system and/or decompose the original transforming matrix further until the basic mechanical components. There are the close relations among the control strategies, inference engine, and the method of knowledge representation. As the knowledge base is composed of a series of frame classes, the tree inference chain is used as the controlling strategies to design the inference engine. To generate the behavioral matrix automatically, the artificial intelligent techniques can help to decrease and restrict the quantity of behavioral matrix for scheme design. Fig.7 shows a prototype system framework for conceptual design. The defined knowledge base contains the basic transforming matrix, matching rules and evaluation rules. They represent the knowledge on frame-based and rule-based. The component base contains all the common mechanical components, electrical components and hydraulic pressure components, etc. At the same time, every component corresponds to different symbols, behavioral matrixes and the performance targets. However, some different components could correspond to the same one behavioral matrix, and they may be stored by the dynamic data structure. They are also represented by the knowledge on frame-based. In the process of system running, the behavior matrix attribute is transformed continuously based on the procedure of effort→flow→effort →flow □□□. Once taking up a transformation, the inference engine will use the knowledge base to index corresponding to the physical components and make some decisions. If the behavior matrix cannot index corresponding to the physical components, go by dispatching disposal or add a transforming matrix by a designer. After the reasoning is finished, a functional means tree is produced, as shown in Fig. 8 [22].

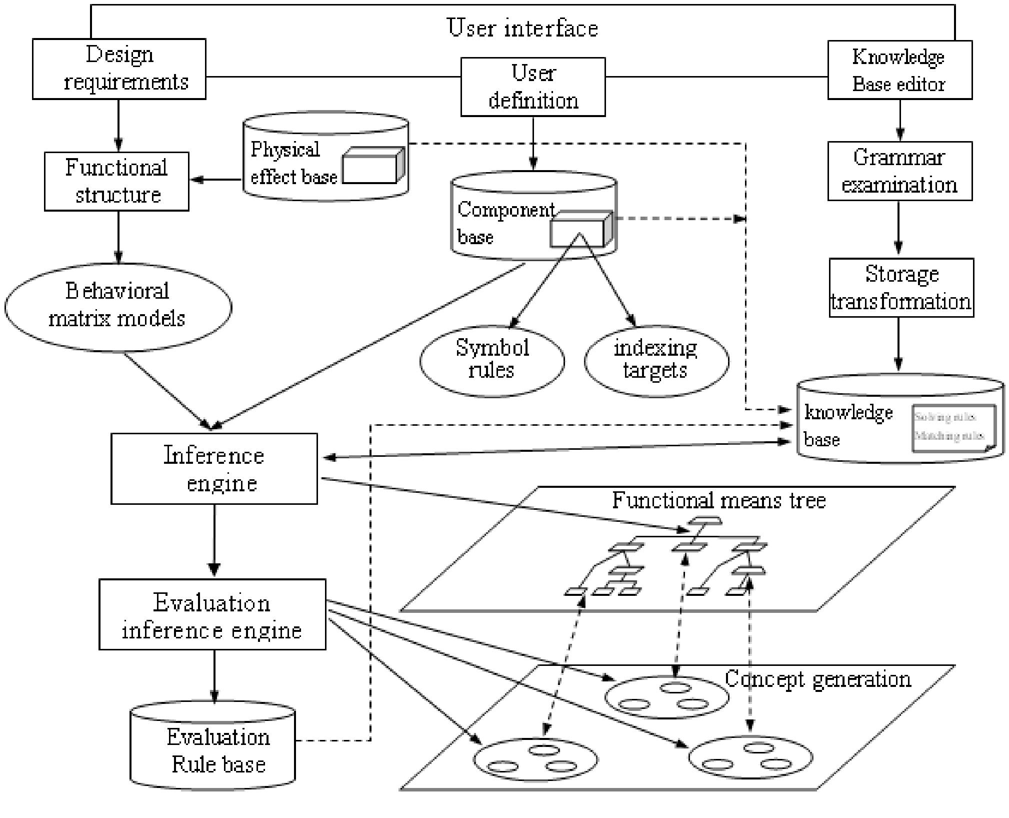


Fig. 7: Prototype conceptual design system.

In the knowledge base with the representation of object-oriented frame structure, the tree structure is composed of the frame classes. Each frame class is connected through the pointer of the parent node and the pointer of the child node in the frame class node. Actually, it gives an inference chain, which means that the inference can proceed only according to the tree chain of the knowledge base frame structure. It forces the inference to proceed in a certain range, which subsequently reduces search space. As an example, Fig. 8 shows the functional means tree structure for the knowledge base in which each means corresponding to components represents a frame class. During the process of inference running, an object (input effort *T*/Output effort *F*) is decomposed into four frame classes ($\boldsymbol{T}_{1R}$, $\boldsymbol{T}_{1GY}$, $\boldsymbol{T}_{1C}$, $\boldsymbol{T}_{1TF}$) which are located at the first level. Each frame class is decomposed further into sub-functions or sub-frame classes. Only the lower layer pointer is needed to judge if its child nodes (such as T_{0L} and T_{0TF} located at the third level) satisfy the matching conditions without considering other frame classes (such as T_{1C} and T_{1GY} located at the third level). If the first level T_{1GY} meets the matching conditions, the rules of slot T_{1GY} are used in inference further, until one or several results are obtained. If the inference failure occurs (e.g. in the seventh level T_{1R}) it will trace to its upper frame class T_{1TF}. T_{1TF} will then choose another child frame class T_{1GY} to make inference, but not other frame classes. If all the child frame class inferences fail in T_{1TF}, it will continue to trace to the upper layer based on its pointer of the parent node. According to the effect of the tree inference chain, the unconcerned frame classes are removed from the current searching space so that the searching space and tracing range are further reduced, and the solving process is accelerated.

The reasoning process is implemented by the inference chain of the frame class tree. In the knowledge base, calculating and processing are embedded in the rules. In the inference process, not only the matching of object and object type must be considered, but also the calculation and scheme comparison be finished. In Fig. 8, the painting gray rectangles represent the root nodes while the painting half-gray rectangles denote the tree nodes in the functional means tree. Different branch of the tree corresponds to different design solutions. In order to evaluate multi-solutions and decide the main performances of each solution, the evaluation inference engine and the evaluation rules are used to appraise the synthetic targets of the solutions, such as cost, motion precision, complexity, structure compactness, and the performance targets (the exact value of time and frequency response, and so on). If the depth search approach is first adopted to index functional means tree, three branches with the smallest evaluation factors will be generated [23]. Finally three ideal solutions are given, and the best one is selected as shown in Fig. 8 (a), (b) and (c).

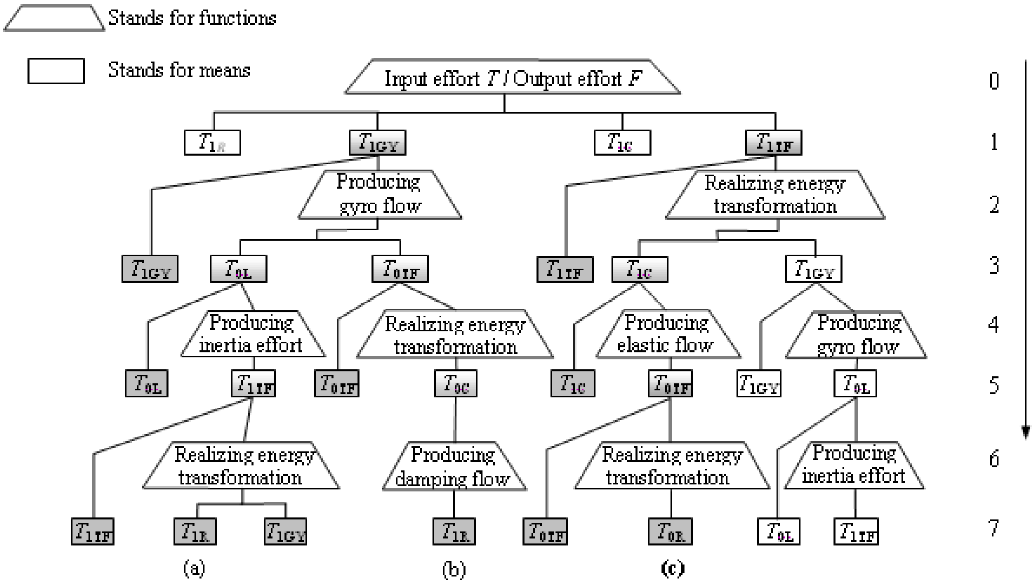


Fig. 8: Functional means tree.

5.3 Man-machine Interface

A prototype system is developed based on the above presented technology. The purpose of the prototype is to implement the method of knowledge synthesis in a computer-aided design environment, and the developed system can be used to aid conceptual design. In addition, the knowledge-based system shell is also developed to support the user interface for data input and feedback of design process.

A man-machine interface is provided by the developed system. The used modeling approach which adapts to obtain different domains of knowledge ensures the validity of the designed prototype system. The users can edit the contents of database in the system, which includes the knowledge base and the dynamic database. The process can be partly realized by the user operating on the main interface called as *Designbuilder* system as shown in Fig. 9. Moreover, the operating environment provides different menus, dialogs, alternate controlling buttons, etc. Therefore, it facilitates to generate principle schemes and provides the user-friendly interface.

In the following, the main functions of the prototype system are highlighted.

(1) Menu files have the general functions like the other design software. They are mainly involved in the text file operation of the frame classes and the acquisition of new design knowledge from the designer. They can help designer maintain correct knowledge syntax, and perform the consistency check on the updated knowledge base.

(2) Design parameter input includes 16 kinds of design variables, design requirements as well as prototype component choices (such as transducers). They provide a man-machine friendly interface.

(3) Design operations include browsing energy category and physical parameter changes during the process of design reasoning, generating functional means tree, transforming automatically into bond graph, producing simulation curves and implementing configuration design.

(4) Design inference involves inference engine, functional decomposition as well as multi-solution generation. The inference engine may choose forward chaining or backward chaining. The inference cannot be executed before the knowledge-based file is loaded. The inference results are passed dynamically to the application program of the graphic drawing, such as MatLab, AutoCAD.

(5) Scheme choice and evaluation include the comparison and selection of a solution over alternative solutions. This activity involves the application of domain expertise as well as the use of evaluation criteria. If the users are not satisfied with the solutions, they may press the modification button to revise the s selected olution, such as adding behavioral matrix or deleting behavioral matrix.

6 A CASE STUDY

A fast clasping mechanism is a sub-system of a fixture used in machine centers. The original clasping mechanism, which is used in machining centers as a subsystem of a fixture, is a screw mechanism. The mechanism is operated manually. The speed for clasping and releasing workpiece is slow and not suitable for the mass production [33]. The users hope that a new product can be developed for the fast clasping and releasing operations. Generally, the fast clamping needs to be driven by hydraulic pressure, pneumatic or electromagnetism jig [34]. In addition to generating greater clamping force, the output parameter should be surely effort (clamping force), and the input parameter should also be effort (operator's force, hydraulic force, atmospheric pressure, and so on). Therefore, the designed systematic model should be in the format of Eq. (8) or Eq. (10). Supposing the time variable and the inertia of system are not concerned and the liquid is not compressed. The input design parameter torque and the output design parameter force are selected in the main interface of *Designbuilder*, as shown in Fig. 9. These parameters belong to DDP and the attribute 'energy flow' should be selected. In the process of problem solving, the system needs users to select a transducer (as shown in Fig. 10) in order to go on the reasoning further. With the inference strategy of mixed control, the functional means tree as shown in Fig.11 can be got (totally 7 kinds of solutions corresponding to Fig. 8), just (a), (b) and (c) are three kinds of relative ideal schemes, in which (c) behavioral matrixes are put in order as follows.

$$\boldsymbol{T}_{1TF} \rightarrow \boldsymbol{T}_{1C} \rightarrow \boldsymbol{T}_{0TF} \rightarrow \boldsymbol{T}_{0R}$$

Where, $\boldsymbol{T}_{TF}$ stands for hydraulic cylinder. In order to facilitate operator's manual input effort, a screw mechanism is adopted so that the modulation transform $\boldsymbol{T'}_{1TF}$ should be added before it. In addition, an effort variable S_e needs to be added. The resistant transform T_{1R} and capacitance transform $\boldsymbol{T}_{0C}$ are also needed. $\boldsymbol{T}_{0C}$ is an elastic effort of screw mechanism, and $\boldsymbol{T}_{1R}$ represents the damping flows of screw mechanism. $\boldsymbol{T}_{1TF}$ and $\boldsymbol{T}_{0TF}$ stand for hydraulic cylinder respectively. They can increase the hydraulic pressure along with the rate of area A_0/A_1. Furthermore, the screw mechanism $\boldsymbol{T'}_{1TF}$ can amplify the displacement along with the difference between screw pitch t_0 and t_1. Therefore, the extended system behavioral matrix order is represented as follows.

$$\boldsymbol{T}_{0C} \rightarrow \boldsymbol{T}_{1TF} \rightarrow \boldsymbol{T}_{1R} \rightarrow \boldsymbol{T}_{1TF} \rightarrow \boldsymbol{T}_{1C} \rightarrow \boldsymbol{T}_{0TF} \rightarrow \boldsymbol{T}_{0R}$$

After extending the above system behavioral matrixes through using protocol criteria, the bond graph model is showen in Fig. 10. In addition, Fig. 11 (a) gives the configuration design based on Fig. 8(c) and Fig. 11 (b) shows the shape of fast clasping mechanism. This product is already under production with a small quantity in a company. It greatly shortened tooling time and thus improved productivity.

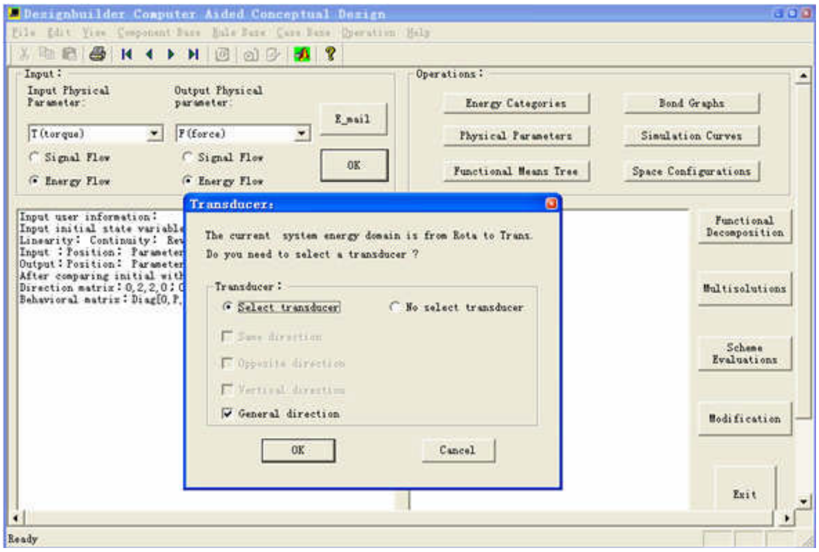


Fig. 9: Transducer selection dialogue window.

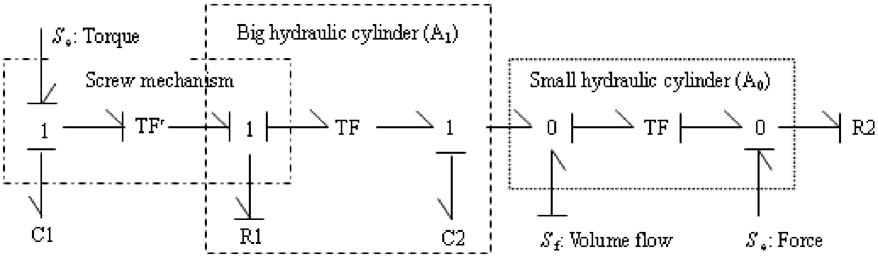


Fig. 10: Bond graph model.

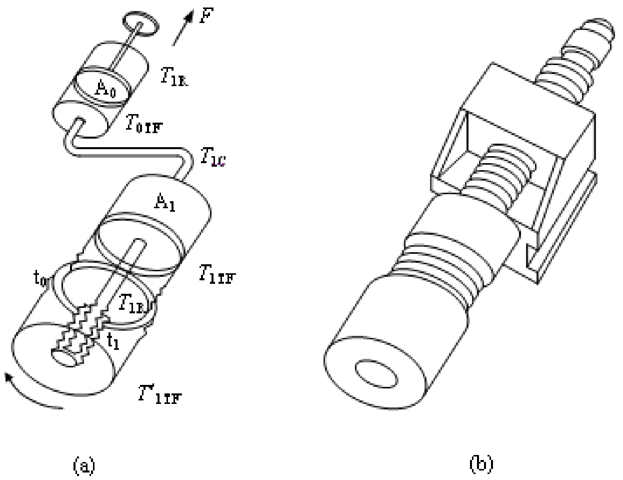


Fig. 11: A scheme of fast clasping mechanism of machine center.

7 CONCLUSIONS

Although the knowledge based system has been extensively applied in engineering design for several decades, it has not yet got a great success in preliminary design. The key issue is the difficulty to formalize design process modeling and knowledge modeling at the conceptual design stage. The paper proposed a behavioral matrix for conceptual design process, and presented a knowledge modeling language for describing the structure and behavior characteristics of components. A prototype system is developed by using an object-oriented technique and its interfaces are user-friendly. The proposed design processes are well suitable for description of product design requirements, behavioral matrix modeling, and design problem solving as well as multi-solution generation. The developed knowledge-based system shell and environment is an efficient elicitation approach and technique for solving design problems at the preliminary stage. The knowledge modeling language facilitates the cooperation between the knowledge-based system and the database for design inference. In addition, different transducer choices are provided to produce a certain impact on the result of design problem solving. How to improve the intelligence of prototype system to adapt to the general mechanical product preliminary design as well as how to realize a controllable design of system in order to avoid missing an opportunity for solving good schemes are our research direction in the future.

ACKNOWLEDGEMENTS

This research is sponsored by the National Nature Science Foundation of China (NSFC) under the grant of No.50775065 and the Nature Science Foundation of Hebei Province, P.R. China under the grant of No.E2008000102.

REFERENCES

[1] Bradley, H. D.; Maropoulos, P. G.: A relation-based product model for computer-supported early design assessment, Journal of Materials Processing Technology, 76, 1998, 88-95.
[2] Hsu, W.; Liu, B.: Conceptual design: issues and challenges, Computer-Aided Design, 32(14), 2000, 849-850. doi:10.1016/S0010-4485(00)00074-9
[3] Zhao, W.; E1-Gizawy, A. S.; Billatos, S. B.:1996, A knowledge-based system methodology for conceptual design of mechanical systems, 1996 ASME International Mechanical Engineering Congress and Exposition, Atlanta, GA. November 17-22.
[4] Welch, R. V.; Dixon, J. R.: Guiding Conceptual Design Through Behavioral Reasoning, Research in Engineering Design, 6(3),1994, 169-188. doi:10.1007/BF01607277

[5] Hsu, W.; Woon, I. M.: Current research in the conceptual design of mechanical products, Computer-Aided Design, 30(5), 1998, 377-389. doi:10.1016/S0010-4485(97)00101-2

[6] Cross, N.: Engineering Design Methods John Wiley & Sons, 1991

[7] Suh, N.P.: Axiomatic design: Advances and Applications, Oxford University Press, New York, 2000

[8] Sharpe, J. E.; Bracewell, R. H.: The use of bond graph reasoning for the design of interdisciplinary schemes, International Conference on Bond Graph Modeling and Simulation, 1995, pp. 116-121.

[9] Tay, E. H.; Flowers, W.; Barrus, J.: Automated generation and analysis of dynamic system designs, Research in Engineering Design, 10(1), 1998, 15-29. doi:10.1007/BF01580267

[10] Gui, Jin-Kang; Mantyla, Martti: Functional Understanding of Assembly Modeling Computer-Aided Design, 26(6), 1994, 435-451.

[11] Zhang, W. Y.; Tor, S. B.; Britton, G. A.: A prototype knowledge based system for conceptual synthesis of the design process. Int. J. Adv. Manuf. Technol, 2001, 17, 549-557.

[12] Gero, J. S.: Design Prototypes: A Knowledge Representation Schema for Design, AI Magazine, 11 (4), 1990, 26-36.

[13] Chakrabarti, A.; Bligh T. P.: An approach to functional synthesis of mechanical design concepts: theory, application and emerging research issues, Artificial Intelligence for Engineering Design Analysis and Manufacturing (AIEDAM), 10 (4), 1996, 313-332. (Special Issue)

[14] Roy, U.; Pramanik, N.; Sudarsan, R.: Function-to-form mapping: model, representation and applications in design synthesis, Computer-Aided Design, 2001, 33, 699-719.

[15] Akagi, S.; Fujita, K.: Building an expert system for engineering design based on the object-oriented knowledge representation concept, Journal of Mechanical Design, 112 (2), 1990, 215-222. doi:10.1115/1.2912595

[16] Gorti, S. R.; Gupta, A.; Kim, G. J.: An object-oriented representation for product and design process, Computer-Aided Design, 30(7), 1998, 489–501. doi:10.1016/S0010-4485(97)00087-0

[17] Zhang, W. Y.; Tor, S. B.; Britton, G. A.: EFDEX: A knowledge-based expert system for functional design of engineering systems, Engineering with Computers, 17(4), 2001, 339-353.

[18] Hubka, V.; Eder, W. E.: Theory of technical systems, Springer-Verlag, Berlin, 1988.

[19] Pahl G.; Beitz W.: Engineering Design: a systematic approach (M), revised 2nd edition, Springer-Verglag, London, 1996.

[20] Kirschman, C. F.; Fadel, G. M.: Classifying functions for mechanical design, Transactions of the ASME Journal of Mechanical Design, 120(4), 1998, 475-490. doi:10.1115/1.2829176

[21] Karnopp, D. C.; Margolis, D. L.; Rosenberg, R. C.: System Dynamics: A Unified Approach. 3rd ed., John Wiley & Sons, New York, 2000.

[22] Cao, Dongxing: Study on key technologies of the process modeling and multi-agent collaboration for the product conceptual design, PhD thesis, 2004, Hebei University of Technology

[23] Cao, Dongxing; Tan, Runhua; Yuan, Caiyun: Conceptual design of mechanical product based on functional decomposition, Chinese Journal of Mechanical Engineering (in Chinese), 37(11), 2001, 13-17. doi:10.3901/JME.2001.11.013

[24] Su, K. W.; Hwang, S. L.: Liu, T. H.: Knowledge architecture and framework design for preventing human error in maintenance, Expert Systems with Applications, 19, 2000, 219-228.

[25] Welch, R. V.: 1991, Conceptual design of mechanical systems, 3rd International Conf. on Design Theory and Methodology, ASME, USA, p.61-68.

[26] Yang, R.; Xiao, R.: 2004, An integrated approach to design knowledge acquisition and transformation and its implementation, Proceedings of International Conference on Manufacturing Automation, Oct. 26-29, Wuhan, China, pp.365-373.

[27] Gelder, J. D.: 1997, Conceptual modeling of building regulation knowledge, Artificial Intelligence in Engineering, 11, pp.273-284. doi:10.1016/S0954-1810(96)00043-X

[28] Chiou, S. J.; Kota, S.: Automated conceptual design of mechanisms. Mechanism and Machine Theory, 34 (3), 1999, 467-495. doi:10.1016/S0094-114X(98)00037-8

[29] Zha, X. F.; Lim S. Y. E.; Fok S. C.: Development of expert system for concurrent product design and planning for assembly. The International Journal of Advanced Manufacturing Technology, 15, 1999, 153-162. doi:10.1007/s001700050052

[30] Malmqvist, J.: Improved function-means trees by inclusion of design history information, Journal of Engineering Design, 8(2), 1997, 107-117. doi:10.1080/09544829708907955

[31] Sturges, R. H.; O'Shaughnessy, K.; Reed, R. G.: A systematic approach to conceptual design, Concurrent Engineering: Research and Applications, 1(2), 1993, 93-105.
[32] Stone, R.; Wood, K.: A heuristic method for identifying modules for product architectures. Design Studies, 2000, 21, 5-31. doi:10.1016/S0142-694X(99)00003-4
[33] Kumar, A. S.; Subramaniam, V.; Teck, T. B.: Conceptual design of fixtures using machine learning techniques, International Journal of Advanced Manufacturing Technology, 16(3), 2000, 176-181.
[34] Roy, U.; Liao, Jianmin: Application of a blackboard framework to a cooperative fixture design system, Computers in Industry, 1998, 37, 67-81. doi:10.1016/S0166-3615(98)00068-2