Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2013), *27*, 415–420.
© Cambridge University Press 2013 0890-0604/13 $25.00
doi:10.1017/S0890060413000164

TECHNICAL BRIEF

Modeling detailed design knowledge with the extended structure–behavior–function model

YONG CHEN, JIAN HUANG, YOUBAI XIE, AND ZHINAN ZHANG
School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China

Abstract

Detailed design is often a time-consuming and experience-dependent engineering process, where various detailed design knowledge can be reused. This paper proposes a formal approach for modeling detailed design knowledge for effective reuse. An extended structure–behavior–function model is developed for representing the structural, behavioral, and functional information in various life cycle periods of a detailed design. Based on the extended structure–behavior–function model, an issue- and solution-based approach is then developed to model the detailed knowledge of a mechanical design. The proposed approach is implemented in a detailed design knowledge modeling system, with a fixture design knowledge modeling as a brief example.

Keywords: Design Rationale; Detailed Design Knowledge; Mechanical Design; Structure–Behavior–Function; Tacit Knowledge

1. INTRODUCTION

Design knowledge reuse (DKR) is an effective approach for design organizations to develop robust artifacts in short time and with low cost. However, existing computer-aided design (CAD) and product life cycle management software, which are primarily developed for designers to manage geometrical information, cannot provide effective support for DKR. Therefore, DKR has attracted considerable attention from the engineering design community in recent decades. For example, Gero (1990) has developed the function–behavior–structure model for organizing the conceptual knowledge about a design prototype; Szykman et al. (2000) have developed a web-based approach for modeling the conceptual knowledge of existing artifacts for reuse; Baxter et al. (2007) have developed an integrated DKR framework for the early design stage through bringing together best practice reuse, design rationale capture, and knowledge-based support; Goel et al. (2009) have developed the structure–behavior–function (SBF) model for representing conceptual design knowledge; and Bracewell et al. (2009) have extended the issue-based information system for capturing design rationale in a more effective manner. Witherell et al. (2010) have developed a first-order-logic-based approach to improve design knowledge management. Our previous research has developed a knowledge-based approach for reusing multidisciplinary principle solutions through design synthesis (Chen, Liu, et al., 2012).

However, existing DKR research is primarily for modeling and reusing early (e.g., conceptual) design knowledge, with very little dealing with detailed design knowledge. Here, detailed design primarily refers to a mechanical design process for determining the forms and dimension of mechanical parts. According to Pahl and Beitz (2007), detailed design should consider not only the functional and embodiment requirements generated in upstream stages but also various life cycle (e.g., manufacturing, assembling, transportation, maintenance, etc.) requirements in downstream stages. Therefore, it is a complex decision-making process that requires all life cycle knowledge from both upstream and downstream stages. In addition, since detailed design is often experience dependent, much reusable knowledge often exists. Therefore, it is of great value to study how to model and reuse detailed design knowledge.

This paper will introduce a formal approach for modeling detailed design knowledge, which is the cornerstone of our DKR research. Section 2 introduces how the original SBF model (Goel et al., 2009) is extended to represent detailed design information. Based on the extended SBF model, Section 3 then proposes a formal approach for modeling detailed design knowledge. With a fixture design case, Section 4 briefly illustrates how the proposed design knowledge modeling

Reprint requests to: Yong Chen, Room 838, School of Mechanical Engineering, Shanghai Jiao Tong University, No. 800, Dongchuan Road, Minhang District, Shanghai 200240, China. E-mail: mechenyong@sjtu.edu.cn

approach is implemented. Finally, Section 5 concludes this paper.

2. AN EXTENDED SBF MODEL

Engineering design researchers usually agree that a technical artifact is primarily related to three basic concepts: function, structure, and behavior. These three concepts and their relations can be used to model design knowledge. One of the representative works is the original SBF model for representing conceptual design knowledge (Goel et al., 2009). In this research, we have extended the original SBF model for representing detailed design information, as shown in Figure 1.

2.1. Structure

Structure refers to the physical appearance of an artifact. The structural representation here has both a composition representation and a form representation. The composition representation employs bills-of-materials to describe what entities (i.e., parts or assemblies) an artifact has, as well as the assembly relationships among them. Each entity (part or assembly) in the composition representation can be conceptualized as a binary group, <*id_name*, {*attributes*}>, where *id_name* describes the ID (identification) name of the entity, while *attributes* are physical attributes, for example, density and rigidity.

To support detailed design, the original SBF model has been extended to include the form representation, which describes the geometrical forms and the related dimensions of each part. Our form representation takes a descriptive feature-based approach, which employs geometrical features and location relation parameters between those features to represent the geometrical information of a part. Each geometrical feature is further represented with a set of parameters. Note that the form representation of a part is also associated with a parameterized CAD model to facilitate its integration with commercial CAD software.

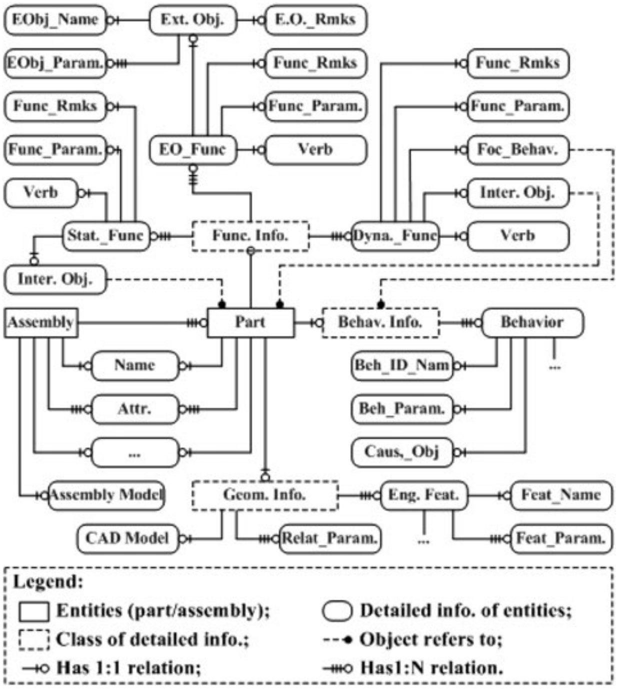


Fig. 1. An extended structure–behavior–function model.

2.2. Behavior

The behavior of an entity refers to its own state change in its life cycle periods—for example, the rotation of an axle or the deforming of a spring. Unlike the original SBF model, which usually deals with the causal behaviors for achieving a desired function (Goel et al., 2009), the extended SBF model also involves those behaviors in various life cycle (e.g., manufacturing) periods of an entity, as well as some side behaviors that are unrelated to the entity's functions. This is because designers should also consider such life cycle behaviors in detailed design.

A behavior here is quantitatively represented with a 5-ary tuple, <*id_name*, *aff_entity*, *beh_param*, *caus_obj*, *remarks*>. Here, *id_name* stands for the ID name of a behavior, *aff_entity* refers to what entity (i.e., part or assembly) the behavior is affiliated to, *beh_param* is a quantitative parameter for describing the degree of the state change, *caus_obj* describes the objects that stimulate the current behavior, and *remarks* are informal explanations. It is evident that the behavioral representation here is different from that in the original SBF model, where a behavior has been qualitatively represented as a causal state transition.

2.3. Function

Based on our recent research (Chen, Huang, et al., 2012), a function in the extended SBF model is defined as an intended relation that one entity has (i.e., subject) with another (i.e., object) in any of its life cycle periods. The concept of function here is therefore more like affordance, a novel design methodology concept developed by Maier and Fadel (2009), rather than that in the original SBF model, which primarily refers to the input–output state transitions. This affordance feature also makes our extended SBF model different from the CPM2 model (Fenves et al., 2008).

Functions can be classified as two types: internal functions and external functions. An internal function means that both its subject and its object belong to the artifact being designed. Internal functions can be further classified as static functions and dynamic functions. In a static function, the object is usually regarded as a static entity. For example, when the gearbox is said to have a function of containing gears, the gears are regarded as static objects. A static function can be conceptualized as a 5-ary tuple, <*sub*, *verb*, obj_{in}, *func_param*, *remarks*>. Here, *sub* denotes the subject, *verb* is a verb describing the functional relation, obj_{in} refers to the internal object (i.e., a part) that is related to the subject, and *func_param* is a functional parameter. A dynamic function

https://doi.org/10.1017/S0890060413000164 Published online by Cambridge University Press

is focused on the change of the object's dynamic behavior. It can be orally described as *to do the behavior of something*. A behavior-focused function can be conceptualized as a 6-ary tuple, $<sub, verb, obj_{in}, func\_param, foc\_behav, remarks>$. Here, the symbol *foc_behav* refers to the object's focused behavior to be processed by the function.

An external function means that its object is not a part of the artifact being designed. Similar to an internal function, an external function can be conceptualized as a 5-ary tuple, $<sub, verb, obj_{ex}, f\_param, remarks>$, where obj_{ex} refers to the external object that is acted on. Compared with existing CAD or product life cycle management tools, modeling external functions has a major advantage, that is, the external objects considered in various life cycle periods of an artifact can be captured as functional information. Therefore, it can help designers better understand a detailed design. For example, various tools for machining parts and assembling them (e.g., milling machines, wrenches, etc.) can also be modeled in external functions, which enables designers to understand how such tools influence the result of a detailed design.

3. MODELING DETAILED DESIGN KNOWLEDGE

Detailed design is primarily aimed at assigning values to the geometrical parameters of an entity. The knowledge that supports such decision-making processes is regarded as detailed design knowledge, which can also be called tacit knowledge in existing research (Haug, 2012). Just like design rationale, detailed design knowledge can also be further classified as issue knowledge and solution knowledge.

3.1. Modeling issue knowledge

It is acknowledged that issues considered during designing are the most important contextual knowledge about a design solution. They allow designers to understand and learn what issues have been considered in a design solution. In this research, detailed design issues are represented as three kinds of design constraints: structural constraints, behavioral constraints, and functional constraints.

3.1.1. Modeling structural constraints

A structural constraint usually refers to a constraint on a geometrical parameter of an entity. It may come from multiple origins (e.g., external working environments, industrial standards, and ergonomics). In addition, structural constraints can be classified as simple constraints and complex constraints.

A simple structural constraint merely deals with an allowable value range of a geometrical parameter. It can be represented as a 5-ary tuple, $\{o(p_s), v_b, u, r, rmk\}$, where p_s is the related geometrical parameter, o the object to which the geometrical parameter belongs, v_b its boundary value, u its measure unit, r the relation between the parameter and the boundary value, and *rmk* the informal remarks about the constraint. For example, assume there is a structural constraint on a car, that is, its width should be smaller than 3.5 m. This structural constraint can be represented as, $\{Car(car\_feat(w))$, 3500, "mm", "<", . . .}. Here, the entity is *Car*, which has only one complex feature, *car_feat*, with a geometrical parameter, *w*, for width.

A complex structural constraint often deals with a complex constraining relation among multiple geometrical parameters belonging to multiple entities. In this research, a complex structural constraint is represented as, $\{[o(p_s)_i], v_b, u, r, rmk\}$, where $[o(p_s)_i]$ refers to a polynomial composed of the related geometrical parameters associated with their ontological representations. How to represent a polynomial will be illustrated later.

3.1.2. Modeling behavioral constraints

A behavioral constraint refers to a constraint on the dynamic behavior of an entity to be designed. For example, a designer may impose a behavioral constraint on an economic electrical car, that is, its maximal speed should be smaller than 150 km/h. Similar to structural constraints, behavioral constraints can also be classified as simple constraints and complex constraints.

A simple behavioral constraint refers to the allowable value range of the related behavioral parameter, which can be represented with a 5-ary tuple, $\{o(p_b), v_b, u, r, rmk\}$, where o represents the subject of the behavior and p_b the corresponding behavioral parameter. For example, the behavioral constraint mentioned above can be represented as, $\{Car(v_{max})$, 150, "km/h", "<", . . .}. A complex behavioral constraint refers to the constraining relation among multiple behavioral parameters and can be represented as, $\{[o(p_b)_i], v_b, u, r, rmk\}$, where $[o(p_b)_i]$ is a polynomial composed of multiple behavioral parameters.

3.1.3. Modeling functional constraints

Although the engineering design community usually agrees that engineering design is a function-centered activity, the role of function in detailed design has not been clarified in previous research. We have identified an axiom, the functional constraint (FC) axiom, to help designers derive quantitative constraints from functional descriptions (Chen, Huang, et al., 2012). The FC axiom is the following:

> If an entity S (subject) is said to have a function on an entity O (object), then some design parameter(s) of S will be fully or partially constrained by some design parameter(s) of O.

Here, design parameters can be functional, behavioral, or structural. The FC axiom exists and holds in each successful design case. For example, if a seat is designed to hold an adult, then its geometrical parameter, *width*, will usually be constrained by the average width of adults. Therefore, based on the FC axiom, it is possible to derive many quantitative constraints from the functional descriptions of an entity. A detailed approach for deriving functional constraints from func-

https://doi.org/10.1017/S0890060413000164 Published online by Cambridge University Press

tional descriptions can be found in our recent research (Chen, Huang, et al., 2012), and is therefore not elaborated here.

A functional constraint can be conceptualized as a 5-ary tuple, $\{p_s, p_o, r, f_s, rmk\}$, where p_s stands for a design parameter of a functional subject, p_o for a design parameter of a functional object, and f_s for the identifier(s) of the corresponding function(s). Based on the above representation, the aforementioned functional constraint on the seat can then be represented as, $\{Seat(seat\_feat(w), Adult(w), \text{“}>\text{”}, ID_{\text{to-hold-an-adult}}, \text{“}\ldots\text{”}\}$. Note that a functional constraint must be associated with the corresponding function(s) in the above representation, which at least has two advantages. One is that it can record the argument for a functional constraint, which can make the constraint easily understood by other designers. The other is that it can help a designer judge whether a functional constraint should be reused or not when the corresponding design solution has been reused.

3.2. Modeling solution knowledge

After the structural, behavioral, and functional constraints have been modeled, a critical task then emerges as how to solve these constraints to help designers determine the feasible value range of each design parameter. The constraints-solving knowledge can be called solution knowledge and can be classified as either model knowledge or computation knowledge. Represented with informal remarks, the model knowledge describes what physical model (i.e., principle) can be used to solve a constraint (e.g., a cantilever model).

The computation knowledge is a formal representation that manages the computation formulae for assigning values to design parameters. The proposed model for representing a formula is conceptualized as a 4-ary tuple, $\{p_t, P_p, r, o_c\}$. Here, p_t is a target (i.e., constrained) parameter, whose feasible value range should be determined; P_p represents a polynomial including multiple constraining parameters; r refers to the value relation between the target parameter and the polynomial; and o_c denotes the original constraint of the current formula. The polynomial (P_p) is further represented with the genetic programming approach (Koza, 1992), which can be conceptualized as a triple, $\{n_l, o, n_r\}$. Here, n_l and n_r represent the left-hand node and the right-hand node of a subtree, respectively, which can be a variable, a constant, or a mathematical operator, whereas o is the parent node of the node, which can merely be a mathematical operator. Note that each parameter in the genetic programming representation is defined in the extended SBF model.

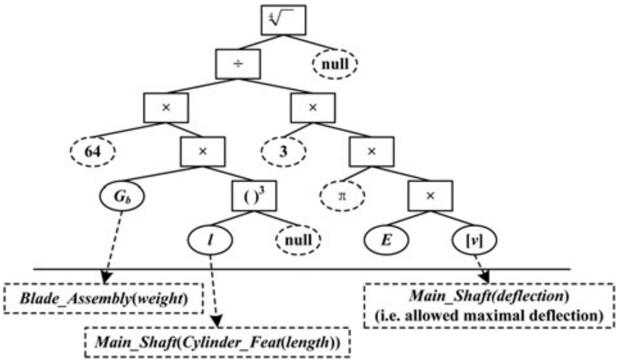


Fig. 2. A genetic programming representation case.

For example, assume the main shaft in a wind turbine has a round cross section with two design parameters, that is, diameter (d) and length (l). Due to the gravity of the turbine's blade assembly, the maximal deflection (v_m) of the shaft must be smaller than a prescribed value $[v_m]$. With the cantilever-based solution model, this deflection constraint (a behavioral constraint) can be further transformed into a structural constraint on parameter d as in

$$d > P_p = \sqrt[4]{\frac{64 G_b l^3}{3\pi E[v_m]}}.$$

Here, E is the Young modulus and G_b is the weight of the blade assembly. The above formula can be formally represented as $\{Mainshaft(cylinder\_feat(d)), \text{“}>\text{”}, P_p, \ldots\}$. Here, P_p is a polynomial, which has a genetic programming representation shown in Figure 2.

4. IMPLEMENTATION

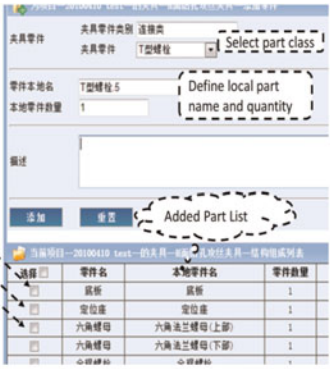
With Microsoft Visual Studio and Microsoft SQL Server, the proposed detailed design knowledge modeling approach has been implemented as a detailed design knowledge modeling system (DDKMS). A DDKMS is primarily composed of the artifact information modeling subsystem for designers to model an existing design with the extended SBF model and the design knowledge modeling subsystem for designers to model the issue knowledge and the solution knowledge. A DDKMS has been deployed in a mechanical fixture design group for test. A senior designer with about 10 years experience was invited to use a DDKMS to model the detailed design knowledge of a mechanical fixture. The primary process was as below.

First, the designer developed the structural representation for the fixture design, which primarily involved the part composition representation of the fixture and the form representation of its parts. As a result, a structural representation has been built, which comprises 18 parts and about 90 geometrical parameters. Figure 3 shows the part (composition) list and a user interface for managing it.

Second, he defined the behavioral representation for each part. As a result, he has collected some behavior(s) of the fixture parts, for example, the translation and the rotation of the double-head-bolt part, and the translation of a T-bolt part for clamping the fixture to the drilling machine. Some side behaviors have also been identified, for example, the flexible extension of the double-head-bolt part when clamping a turbocharger shell and the deflection of the base-board ([BB]) part when it is drilled. Note that such side behaviors are

https://doi.org/10.1017/S0890060413000164 Published online by Cambridge University Press

No.	Part ID Name	Part English Name	Abbr.
1	底板	Base Board	[BB]
2	定位座	Shell Seat	[SS]
3	六角法兰螺母 1	6-Ang-Flang-Nut-1	[6AFN-1]
4	双头螺栓	Double-Head-Bolt	[DHB]
5	压板	Top-Press-Board	[TPB]
6	调节螺钉	Adaptive-Pillar	[AR]
7	定位块	Pillar-Shelf	[PS]
8	调整六角螺母 1	6-Ang-Adjus-Nut-1	[6AAN-1]
9	内六角螺母 1	6-Ang-Inner-Nut-1	[6AIN-1]
10	T 型螺栓 1	T-Bolt-1	[TB-1]
…	…		



夹具零件 夹具零件类别 连接类
夹具零件 T型螺栓
零件本地名 T型螺栓.5
本地零件数量 1
描述

选择 □	零件名	本地零件名	零件数量
□	底板	底板	1
□	定位座	定位座	1
□	六角螺母	六角法兰螺母(上部)	1
□	六角螺母	六角法兰螺母(下部)	1

Fig. 3. (Color online) The part list and the user interface for managing it.

usually not captured in the original SBF model because they are unrelated to the functions of the part.

Third, the designer defined the functional information for each part. About 60 functions have been defined, which involve not only the internal functions but also the life cycle external functions. For example, the [BB] part not only has internal functions, such as supporting the shell-seat part and fixing the position of the shell-seat part, but also has some external life cycle functions, such as holding the cutter of a milling machine and occupying the operating platform of a machine. Note that such life cycle functions also cannot be captured in the original SBF model. The left part of Figure 4 shows some functions, while its right part is a user interface for defining a function shown in the left.

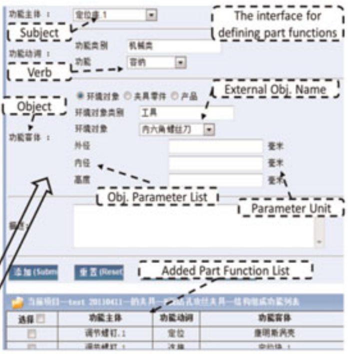
Fourth, the designer modeled the issue knowledge of a fixture. This process primarily involves building the structural, behavioral, and functional constraints on the related entities. For example, since the department standard prescribes that the thickness, *H*, of the [BB] part should be equal to 30 mm, a geometrical constraint can then be modeled as {[*BB*](*BB_Feat*(*H*)), 30, "mm," =, "Department Standard"}, where *BB_Feat* refers to the geometrical feature of the [BB] part; since the [BB] part should fulfill the functions of "semi-containing a turbocharger shell (TS)" and "supporting pillar shelf (PS)" at the same time, which are both related to the width, *B*, of the [BB] part, then another issue knowledge item can be modeled as: {[*BB*](*BB_Feat*(*B*)), [*TS*(*width*), [*PS*](*PS_Feat*(*width*))], >, [$F_{[TS]}$, $F_{[PS]}$], "The [BB] part should fulfill these two functions . . ."}, where "[*TS*(*width*), [*PS*](*PS_Feat*(*width*))]" is a list of related design parameters, *TS*(*width*) and [*PS*](*PS_Feat*(*width*)).

Fifth, the designer then modeled the solution knowledge for the fixture design. Here, the most important work is to model the computation knowledge. As a result, about 120 computational formulae have been obtained. For example, according to the issue knowledge about the width of the [BB] part mentioned before, a computation knowledge item can then be modeled as, $[BB](BB\_Feat(B)) > (0.5 \times TS(width) + [PS](PS\_Feat(width))) \times 2$; because the [BB] part has another function of occupying the operation platform of the machine, then another computation knowledge item can be modeled as $[BB](BB\_Feat(B)) < operational\ platform(width)$. Compared with the original SBF model, which does not deal with quantitative constraints, such computation knowledge is also a major extension for detailed design. Based on these two constraints, a designer can then determine a suitable value for the *width* parameter of the [BB] part. Figure 5 shows some computation knowledge items and a user interface for editing a computation knowledge item.

The results of the test are encouraging. With the aid of a DDKMS, the designer can formally model various detailed design knowledge (especially the life cycle design knowledge) about a detailed design. For example, a designer can use a DDKMS to model the knowledge about how the cutter of a milling machine can impose a functional constraint on the [BB] part, as explained before. Therefore, designers now can not only employ CAD models to keep detailed design results but also use a DDKMS to store the detailed design knowledge hiding behind those results. An interview with novice designers discloses that they are also content with a DDKMS, because they can easily learn what issues (especially those implicit issues) have been considered in

Subject	Verb	Object	Remarks
[BB]	Semi-Containing	Turbocharger shell	Main Func.
[BB]	Fixing Pos. of	[SS]	Internal Func.
[BB]	Supporting	[SS]	Internal Func.
[BB]	Fixing Pos. of	[PS]	Internal Func.
[BB]	Supporting	[PS]	Internal Func.
[BB]	Occupying	Operating platform	External Func.
[BB]	Semi-Containing	[PSB-1]	Internal Func.
[BB]	Semi-Containing	Adjustable Wrench	External Func.
…	…	…	…
[SS]	Supporting	Turbocharger shell	Main Func.
[SS]	Fixing Pos. of	Turbocharger shell	Main Func.
[SS]	Containing	[DHB]	Internal Func.
[SS]	Containing	[6-Angel Screw-1]	Internal Func.
[SS]	Containing	6-Angle Screwdriver	External Func.
	…	…	

Note: Internal Objects are surrounded by "[ ]".

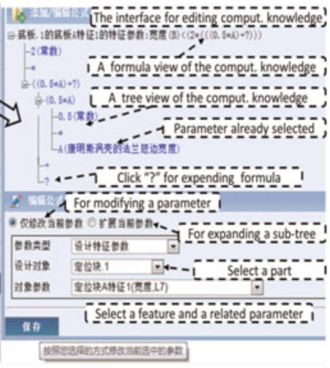


功能主体： 定位座.1
功能类别 机械类
功能动词： 功能 容纳
● 环境对象 ○ 夹具零件 ○ 产品
环境对象类别 工具
功能宾体： 环境对象 内六角螺丝刀
外径 毫米
内径 毫米
高度 毫米
描述：

选择 □	功能主体	功能动词	功能宾体
□	调节螺钉.1	定位	康明斯涡壳

Fig. 4. (Color online) The function list and the related management user interface.

Symbol	Computation Knowledge Examples	The Related Issue Knowledge
H	*H* = 30 mm	Issue-S: Department standard
B	*B* > (0.5*TS(width) + [PS](PS_Feat(width)))*2	Issue-F: F[1]&F[5]
A	*A* > TS(length)+[PS](PS_Feat(length)) + [SB-1](SB_Feat(disp)) + [SB-2](SB_Feat(disp))	Issue-F: F[1]&F[5]&F[...]
D	*D* = OP(s-width)	Issue-F: F[6]
C	*C* = (*B*-*D*)/2	Issue-S: centered constraint
W	*W* = [KSB](KSB_Feat(diameter))	Issue-F: containing [KSB]
L	*L* > 1.5 * [KSB](KSB_Feat(diameter))	Issue-F: allow [KSB] to move
L1	*L1* > 0.5 * TS.width	Issue-F: F[1]
d	*d* = [SS](SS_Feat(c4hc-diam))	Issue-F: F[2] (c4hc: the circle of 4-holes' center)
M	*M* = [DHB](DHB_Feat(diam))	Issue-F: contain [DHB]
…	…	…



底板.1的底板A特征1的特征参数:宽度(B)<(2*(((0.5*A)+7)))
-2(常数)
-*
-((0.5*A)+7)
-(0.5*A)
-0.5(常数)
-*
-A(康明斯涡壳的法兰翅边宽度)
-+
-?
● 仅修改当前参数 ○ 扩展当前参数
参数类型 设计特征参数
设计对象 定位块.1
对象参数 定位块A特征1(宽度,L7)
按照您选择的方式修改当前选中的参数

The interface for editing comput. knowledge
A formula view of the comput. knowledge
A tree view of the comput. knowledge
Parameter already selected
Click "?" for expending formula
For modifying a parameter
For expanding a sub-tree
Select a part
Select a feature and a related parameter
编辑公式
保存

Fig. 5. (Color online) The solution knowledge and the managing user interface.

https://doi.org/10.1017/S0890060413000164 Published online by Cambridge University Press

a detailed design and how these issues are addressed with related solution models. This can largely shorten the time for a novice designer to learn how to design a regular mechanical fixture.

5. CONCLUSIONS

This paper attempts to develop a formal approach for modeling detailed design knowledge for effective reuse. The primary works are as follows.

First, the original SBF model has been extended to represent the structural, behavioral, and functional information in detailed design. A notable feature of the extended SBF model is that it allows designers to employ external functions to capture various life cycle objects that should be considered in a detailed design.

Second, based on the extended SBF model, an issue- and solution-based approach has been developed to model detailed design knowledge. It enables designers to model the knowledge about what issues (especially those implicit life cycle issues) have been considered and the knowledge about how they have been addressed with suitable solution models. This approach at least has two salient features. One is that it provides a method to derive quantitative functional constraints from qualitative function descriptions with the FC axiom. The other is that a genetic programming-based approach has been employed to represent the computation knowledge so that designers can input quantitative design constraints freely, rather than rely on knowledge engineers to encode them. Since such detailed design knowledge also belongs to tacit knowledge, the proposed approach also demonstrates that tacit knowledge can also be modeled in a structured manner.

Third, a DDKMS has been developed based on the proposed approach, which has also been implemented in a fixture design group for test. It is shown that a DDKMS allows a designer to formally model structure-, behavior-, and function-related detailed design knowledge, which usually remains implicit behind the CAD models. Based on the modeled design knowledge, it is also convenient for (novice) designers to understand how design results (i.e., CAD models) have been achieved, that is, what issues have been considered and how they are addressed, and to reuse them in an effective way.

ACKNOWLEDGMENTS

This research is supported by the Science and Technology Commission of Shanghai Municipality, China (No. 09QA1402800), the Natural Science Foundation of China (Nos. 50975173, 50935004, and 51205247), and the Ministry of Science and Technology of China (No. 2008AA04Z108).

REFERENCES

Baxter, D., Gao, J., Case, K., Harding, J., Young, B., & Cochrane, S. (2007). An engineering design knowledge reuse methodology using process modeling. *Research in Engineering Design 18*, 37–48.

Bracewell, R., Wallace, K., Moss, M., & Knott, D. (2009). Capturing design rationale. *Computer-Aided Design 41(3)*, 173–186.

Chen, Y., Huang, J., & Xie, Y. (2012). Using part functions to capture various lifecycle requirements in detailed design. *Proc. Int. Conf. Design Computing and Cognition*, College Station, Texas, June 2012.

Chen, Y., Liu, Z., & Xie, Y. (2012). A knowledge-based framework for creative conceptual design of multi-disciplinary systems. *Computer-Aided Design 44(2)*, 146–153.

Fenves, S.J., Foufou, S., Bock, C., & Sriram, S.D. (2008). CPM2: a core model for product data. *Journal of Computing and Information Sciences in Engineering 8*, 014501.

Gero, J.S. (1990). Design prototypes: a knowledge representation schema for design. *AI Magazine 11(4)*, 26–36.

Goel, A.K., Rugaber, S., & Vattam, S. (2009). Structure, behavior and function of complex systems: the structure, behavior and function modeling language. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 23*, 23–35.

Haug, A. (2012). The illusion of tacit knowledge as the great problem in the development of product configurators. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 26(1)*, 25–37.

Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Maier, J.R.A., & Fadel, G.M. (2009). Affordance-based design: a relational theory for design. *Research in Engineering Design 20*, 13–27.

Pahl, G., & Beitz, W. (2007). *Engineering Design: A Systematic Approach*. London: Springer-Verlag.

Szykman, S., Racz, J., Bochenek, C., & Sriram, R.D. (2000). A web-based system for design artifact modeling. *Design Studies 21*, 145–165.

Witherell, P., Krishnamurty, S., Grosse, I.R., & Wileden, J.C. (2010). Improved knowledge management through first-order logic in engineering design ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 24*, 245–257.

Yong Chen is an Associate Professor of engineering design in the School of Mechanical Engineering at Shanghai Jiao Tong University. He received his bachelor and PhD degrees from Zhejiang University. Yong joined Shanghai Jiao Tong University as a postdoc in 2004. He has been a visiting scholar in the IMPACT Laboratory at the University of Southern California for 1 year. Dr. Chen is interested in many aspects of engineering design research, in particular conceptual design, creative design, design knowledge reuse, and computer-aided design.

Jian Huang is currently a PhD student in the School of Mechanical Engineering at Shanghai Jiao Tong University. He received his bachelor degree in 2008 from Wuhan University. Jian's research interests include knowledge-based design and computer-aided design.

Youbai Xie is a Professor in the School of Mechanical Engineering at Shanghai Jiao Tong University. He is also the Director of the Institute of Modern Design at Shanghai Jiao Tong University. Youbai is an academician of the Chinese Academy of Engineering. His research interests include engineering tribology, design science, and knowledge management. He has over 200 publications in Chinese and international journals.

Zhinan Zhang is currently a postdoc in the School of Mechanical Engineering at Shanghai Jiao Tong University, from which he received his PhD degree in 2011. Dr. Zhang's research interests include design theory and methodology, knowledge management, and engineering tribology.

https://doi.org/10.1017/S0890060413000164 Published online by Cambridge University Press