Integrated Design and Process Technologies, **IDPT-2003**
Printed in the United States of America, December, 2003

FORMALIZATION OF DESIGN REQUIREMENTS

Yong Zeng
Concordia Institute for Information Systems Engineering
Concordia University
1455 de Maisonneuve Blvd. W., Montreal, QC H3G 1M8, Canada
E-mail: zeng@ciise.concordia.ca
Fax: 1-514-848-3171, Tel.: 1-514-848-2424 #5801

ABSTRACT

This paper proposes a process to formalize various design requirements. The input of this formalization process is design requirements described in natural language whereas the output is the formal representation of these design requirements. The basis of this research is the axiomatic theory of design modeling (Zeng, 2002). This theory defines an engineering system and some operations on the engineering system, based on axioms of objects. Using this axiomatic theory, a formulation scheme is established to represent design requirements with the engineering system and a linguistic structure is developed to capture the language elements describing design requirements. A step-by-step formalization process is proposed to identify the engineering systems implied in the description of design requirements. A rivet setting tool design example is used to illustrate the presented notions.

INTRODUCTION

Design requirements are constraints on a product design so that the designed product can be manufactured to achieve its desired functions in its working environment. These can be motives or demands for creating a completely new product, complaints about the performance of existing products, or the failure due to malfunction of existing products. Designed products can be machines, software systems, architectures, and so on. Product environment is anything outside the product that limits the product's performances. Examples include nature and human beings. It is the source of various design requirements, such as constraints, functions, regulations, and so on. These requirements are an important part of a design problem. In engineering practice, designers and/or clients use natural language to describe various application-oriented design requirements. The terminology and jargon describing these design requirements are thus application specific. The acquisition, classification, and representation of these requirements are essential for solving design problems.

The objective of this paper is to propose a process to formalize design requirements appearing in a design. The input of the formalization process is the design requirements described in natural language while the output is the formulation of these design requirements. In contrast to existing observation and speculation-based research methods, this paper adopts the axiomatic approach to developing the formulation of design requirements. The basis of this approach is the axiomatic theory of design modeling (Zeng, 2002).

It should be noted that formulation and formalization are two mutually dependent issues. Without the context of formalization, the formulation looks aimless. Meanwhile the formalization cannot be made without the formulation being its foundation. To help readers understand this paper, the next section gives a design example to illustrate the concepts and notions introduced throughout the paper. Then, using the axiomatic theory of design modeling, a formulation of design requirements is derived and the linguistic structure of the natural language describing design requirements is established. Based on these formulation results, a step-by-step formalization process is developed to identify the formal structure implied in the description of design requirements. Example of formalizing design functions is used to show how the formalization process works. Conclusions and future directions are given in the last section.

EXAMPLE

This section adapts a rivet setting tool design example from the book by Hubka *et al* (1988) to illustrate the concepts proposed in this paper. The task of this problem is to design a tool for riveting brake linings onto brake shoes for internal drum brakes as shown in Fig. 1.

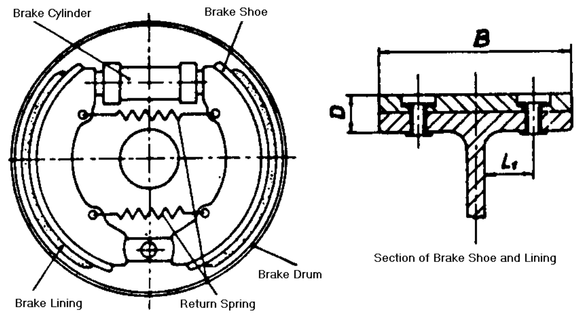
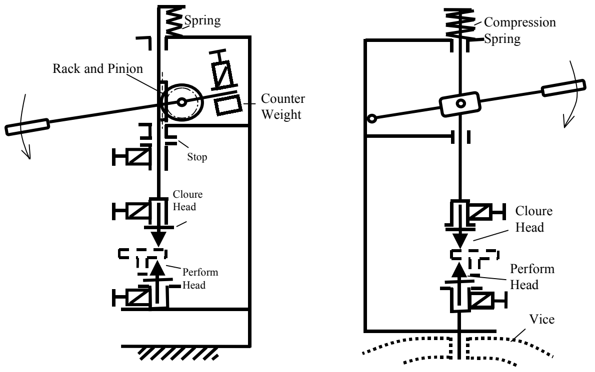


Fig. 1 Internal drum brake (Hubka et al,1988).

The additional information regarding this design problem includes: the user of the tool is the car mechanic. The hand force, foot force, and the working height should follow ergonomic standards. The use of the tool should conform to related industry safety standards. The service life of the tool should be around 5 years. The tool should be easy for transportation and maintenance. The tool will be manufactured in a specific workshop, which has specified equipment. The cost of the tool cannot be over $190.00. Fig. 2 gives two examples of the final design concepts.



a) b)

Fig. 2 Design concepts (adapted from Hubka et al,1988).

The theory presented in the rest of this paper will formulate design requirements included in the above problem description.

FORMULATION OF DESIGN REQUIREMENTS

This section will derive the formal syntax and semantics of design requirements using the axiomatic theory of design modeling.

Review of the Axiomatic Theory of Design Modeling

Axiomatic theory of design modeling includes two groups of axioms: axioms of objects and axioms of the human thought (Zeng, 2002). This paper will only use the axioms of objects. These two axioms state that everything in the universe is an object and that there are relations (denoted by $\otimes$) between objects. The relation from an object to itself is formally called the relation on the object. Informally, the universe is the whole body of things and phenomena observed or postulated. An object is anything that can be observed or postulated in the universe. Two important notions are established from this axiomatic theory: structure operation and range operation. They provide aggregation and generalization mechanisms for representing the object evolution in the design process. These two operations are defined as follow:

[D1] *Structure operation*, denoted by $\oplus$, is defined by the union of an object and the relation of the object to itself.

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

$\oplus O$ is the structure of the object O.

[D2] *Range operation*, denoted by Θ, is defined by the intersection of an object and the relation of the object to itself.

$$\Theta\, O = O \cap (O \otimes O). \tag{2}$$

ΘO is the range of the object O.

Other operations such as $\subseteq$, $=$, $\cup$, and $\cap$ are also defined in Zeng (2002). The following rules hold for these operations in the context of aggregation and generalization:

$$A \otimes (B \cup C) = (A \otimes B) \cup (A \otimes C), \tag{3}$$
$$(A \cup B) \otimes C = (A \otimes C) \cup (B \otimes C),$$
$$A \otimes (B \cap C) = (A \otimes B) \cap (A \otimes C),$$
$$(A \cap B) \otimes C = (A \otimes C) \cap (B \otimes C),$$
$$\Theta(A \cup B) = (\Theta\, A) \cup (\Theta\, B),$$
$$\Theta(A \otimes B) = (\Theta\, A) \otimes (\Theta\, B),$$
$$\Theta(\oplus A) = \oplus(\Theta\, A).$$

Engineering System

An engineering system can be divided into two parts: product and its environment. For the example shown in Fig. 2, the product is made up of components such as spring, stop, counter-weight, rack and pinion, as well as closure and perform heads. The environment includes the natural environment such as gravity field, the ergonomic environment such as the forces that the user may impose, the spatial environment such as the physical properties of the brake assembly, the financial environment such as the price of each component, the manufacturing environment such as the available manufacturing tools, and so on. Generally speaking, everything except the product itself can be seen as its environment. Let

$$\Omega = E \cup S, \tag{4}$$

where product and its environment are denoted by S and E, respectively. Then an engineering system, $\oplus\Omega$, is formally represented as

$$\oplus\Omega = \oplus(E \cup S) = (\oplus E) \cup (\oplus S) \cup (E \otimes S) \cup (S \otimes E), \tag{5}$$

where $\oplus E$ and $\oplus S$ are structures of the environment and product, respectively.

There are two types of relations between product and environment. One is the structural relation such as geometric contact. Another is the physical interaction such as action and response. For the rivet setting tool design, examples of the first type include the connections between the tool and the brake assembly, the contacting geometry between the car mechanic's hands and the handler of the tool. In the context of ergonomic environment, the interactions include the force acted on the handler by the car mechanic's hand or foot. Denoting the union of all possible relations by B, we have

$$B = (E \otimes S) \cup (S \otimes E). \quad (6)$$

Substituting Equation (6) into (5), we have

$$\oplus\Omega = \oplus(E \cup S) = (\oplus E) \cup (\oplus S) \cup B. \quad (7)$$

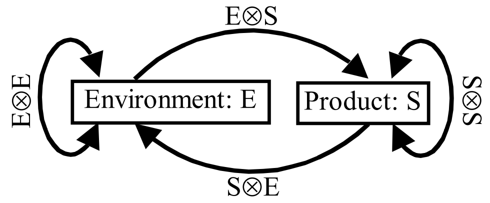


Fig. 3 Engineering system.

Graphically, an engineering system can be represented in Fig. 3. Any information about an engineering system should be derivable from or defined based on Equation (7).

Design Requirements: Syntax

[D3] A *constraint*, denoted by c, defines a relation between an object, denoted by x, and a range of this object, Θx, within which the object can change. Symbolically,

$$c = \lambda(x, \Theta x), \quad (8)$$

where λ is a relation from x to Θx.

[D4] *Design requirements*, denoted by R^d, are constraints on the engineering system to be designed. Symbolically,

$$R^d \subseteq \lambda(\oplus\Omega, \Theta(\oplus\Omega)). \quad (9)$$

where ⊕Ω is called the constrained engineering system while Θ (⊕Ω) the constraining engineering system.

According to Equations (3) and (5), we have

$$\oplus\Omega = (\oplus E) \cup (\oplus S) \cup B, \quad (10)$$

$$\Theta(\oplus\Omega) = (\oplus(\Theta E)) \cup (\oplus(\Theta S)) \cup \Theta B.$$

Substituting Equation (10) into Equation (9) and considering the definition of constraint [D3], we get

$$R^d \subseteq \lambda(\oplus E, \oplus(\Theta E)) \cup \lambda(\oplus S, \oplus(\Theta S)) \cup \lambda(B, \Theta B). \quad (11)$$

The above derivation process holds since λ is a relation, which obeys the rules defined in Equation (3).

Since the environment is predefined in a design problem, i.e., $E = \Theta E$, the first item in Equation (11) can be eliminated.

$$R^d \subseteq \lambda(\oplus S, \oplus(\Theta S)) \cup \lambda(B, \Theta B). \quad (12)$$

Equation (12) is the basic formulation of design requirements. It gives a uniform syntactic representation of various design requirements. The next subsection will study its semantics in the context of engineering design through the classification of design requirements.

Design Requirements: Semantics

From the product life cycle point of view, any product design must take into account a number of requirements regarding functionality, safety, manufacturability, assembly, testing, shipping, distribution, operation, services, re-manufacturing, recycling and disposal (Gu and Sosale, 1999). They are necessary functions and task-specific constraints, which can be listed under the following headings (Pahl and Beitz, 1988): geometry, kinematics, forces, energy, material, signal, safety, ergonomics, production, quality, control, assembly, transport, operation, maintenance, costs, and schedules. A proper classification of these design requirements is essential for organizing design knowledge and the design process.

In terms of Equation (12), design requirements can be classified with respect to the object to be constrained or the object imposing constraints. Obviously, there are only two objects being constrained: product structure (⊕S) and interactions between the product and its environment (B). They are called *structural requirements* and *performance requirements*, respectively. This is shown in Fig. 4. It should be noted that geometric contact between the product and its environment is not a design requirement, but will be taken into account in another paper.

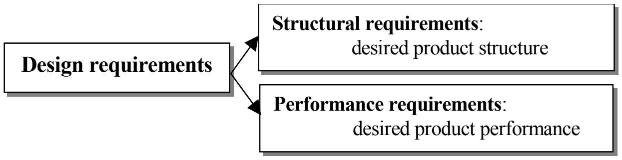


Fig. 4 Classification of design requirements.

For the rivet setting tool design, "the working height should follow ergonomic standards" is a structural requirement whereas "the tool should conform to related industry safety standards" is a performance requirement.

In engineering practice, the environment can usually be divided into different types and/or parts. Suppose

$$E = \bigcup_{i=1}^{n} E_i = E_1 \cup E_2 \cup \cdots \cup E_i \cup \cdots \cup E_{n-1} \cup E_n, \quad (13)$$

where n is a finite positive number. Substituting Equation (13) into (6), we have

$$B = \bigcup_{i=1}^{n} B_i = \bigcup_{i=1}^{n} ((E_i \otimes S) \cup (S \otimes E_i)). \quad (14)$$

Moreover, substituting Equations (14) into Equation (12), then applying the definition of constraint [D3], we have

$$R^d \subseteq \lambda(\oplus S, \oplus(\Theta S)) \cup (\bigcup_{i=1}^{n} \lambda(B_i, \Theta B_i)). \quad (15)$$

Equation (15) classifies design requirements based on the environment imposing the constraints. Still for the example of rivet setting tool design, corresponding to different types of environment, we have ergonomic requirements, manufacturing requirements, financial requirements, etc.

The above two perspectives of classifying design requirements can be combined as is shown in Fig. 5. This figure means that each design requirement, originated from an environment element, imposes constraints on either a structural or a performance aspect of an engineering system.

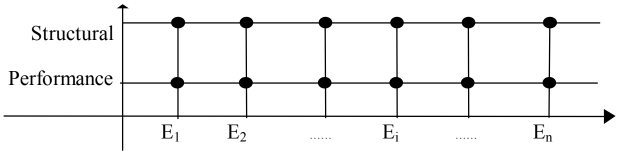


Fig. 5 Refined classification of design requirements.

As such, we have the *Theorem of Design Requirements*:

> Design requirements can be divided into structural requirements and performance requirements. Structural requirements are constraints on the product structure while performance requirements are constraints on the product performance. These requirements can be decomposed in terms of the product environment in which the product is expected to work.

Based on this theorem, an algebraic structure can be established to represent design problems.

FORMALIZATION OF DESIGN REQUIREMENTS

In engineering applications, design requirements are usually described through natural language. It is not feasible to force designers to describe design requirements in the form of constraints on product structure or product performance as are shown in Equations (13) and (15). Hence, it is essential to establish a step-by-step process to formalize design requirements described by natural language into that of the formulation of Equation (15), if the formal specifications are ever to be applied to support the design problem solving. This process is called the formalization of design requirements. To achieve the above objective, the following two questions need to be answered:

1) What is the linguistic structure of the natural language describing the design requirements?
2) How to represent the linguistic structure of design requirements within the engineering system?

This section consists of three subsections. The first subsection defines basic linguistic elements. The second subsection discusses linguistic structure of design requirements. The third subsection presents the process of formalizing design requirements. In the following discussions, all statements related to natural language are cited from Turner (1971) either directly or indirectly. Readers should also note that the word ‘object’ in the context of grammar is not the same as the object in the axiomatic theory of design modelling. It will be self-evident in the context in most cases.

Linguistic Elements

Language is a symbolic system. By common agreement among its users, its symbols (letters and words) stand for ideas in the mind or objects in the environment, or they fulfill certain structural functions in the language pattern. To compare with the axioms of objects in the axiomatic theory of design modeling (Zeng, 2002), the informal definition of object is provided again: an object is anything that can be observed or postulated in the universe. It is obvious that ‘ideas in the mind and objects in the environment’ correspond to objects in the universe.

In the context of engineering applications, different than in that of imaginative literature, engineers directly refer to objects in engineering systems rather than use metaphors. This provides a foundation to formulate design requirements. In describing design requirements, sentences are the basic construct carrying the complete meaning. A sentence consists of one or more clauses capable of presenting a complete thought in a manner, which is logically and grammatically acceptable. Clauses are usually formed by linguistic elements such as nouns, verbs, adjectives, adverbs, pronouns, articles, etc. For the sake of brevity, this paper will only formally discuss nouns and verbs. As a matter of fact, major research results in the design requirement modeling only use these two elements to define functions (Pahl and Beitz, 1988; Lin and Chen, 2002). The present approach can be applied to more complex situations where adjectives, adverbs, and other linguistic elements appear in describing design requirements.

A noun is a word used to name a person, place, thing, quality, idea, or action. In a sentence, it tells who or what did the action or was acted upon by the verb. Any noun in a natural language names an object in the universe. In the context of engineering systems, a noun names either of product, environment, or the four types of relations shown in Fig. 3.

A verb is a word used to indicate the action from/to/on an object or the state of an object. Verbs showing actions include “move”, “change”, “rivet”, etc. Verbs indicating states include “is”, “are”, “has”, “have”, “run”, etc. There are four principal verb types: auxiliary, linking, intransitive, and transitive. Examples of auxiliary verbs include ‘can’, ‘do’, ‘may’, ‘shall’, etc. They shade the meaning of the main verb in some desired manner. The linking verb links a noun to a complement to form a complete sentence. An intransitive verb is one which is able to serve by itself as the predicate of a sentence; no complement or object is required. A transitive verb cannot act alone as a predicate; an object is needed to complete the sentence.

The basic English sentence takes the pattern: subject + predicate. The predicate may be only a verb or a verb plus other elements, such as complement, direct object, indirect object, and objective complement. On the basis of the predicate structure, there are five basic sentence patterns:

1) Subject + intransitive verb
2) Subject + linking verb + subjective complement

3) Subject + transitive verb + direct object
4) Subject + transitive verb + indirect object + direct object
5) Subject + transitive verb + direct object + objective complement

Linguistic Structure of Design Requirements

No matter how complicated design requirements might be for a design problem, they can be ultimately structured into a set of sentences, which assume the given five patterns. On the other hand, as is shown in Fig. 5 and Equation (12), any design requirement can be formulated as a constraint on certain part of engineering system. Therefore, if the relationship between those five sentence patterns and the engineering system can be established, then all design requirements can be logically formalized. It is the objective of this subsection to map plain English sentences describing the design requirements into the engineering system described by the formal symbols.

As in Fig. 3, there are six objects in an engineering system: product, environment, two relations between the product and the environment, one relation on the product, and one relation on the environment. These six objects correspond to nouns in a sentence. To associate verbs to the engineering system, here again is the definition of the relation in the context of axiomatic theory of design modeling (Zeng, 2002): a relation is an aspect or quality that connects two or more objects as being or belonging or working together or as being of the same kind. Relation can also be a property that holds between an ordered pair of objects. Obviously, the first part of this definition corresponds to the linking verb whereas the second part the transitive and intransitive verbs. The following will formally associate the three types of verbs describing design requirements to the engineering system.

1) Linking verb

In describing a design requirement, a linking verb connects two nouns. It links the first noun to a complement, which is also a noun or a noun phrase, to form a complete sentence. In this case, the complement can be seen as the range defining the requirement while the first noun can be seen as the object being constrained. The requirement "the service life of the tool should be around 5 years" is such an example.

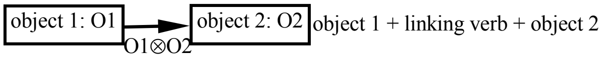


Fig. 6 Linking verb.

The object can be a part of the product or the environment. Using $\otimes_{lv}$ to represent the relation corresponding to the linking verb, p_{lv} for the sentence pattern 2, we have

$$p_{lv} \subseteq O1 \otimes_{lv} O2\,. \qquad (16)$$

2) Intransitive verb

An intransitive verb only involves one object and has the form 'noun verb", so it describes a relation on itself, which is a state of the object.

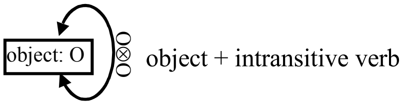


Fig. 7 Intransitive verb.

"Spring deforms" is such an example. The intransitive verb can usually be viewed as a relation between two states of an object, as is shown in Fig. 8.

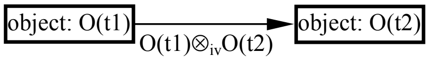


Fig. 8 Intransitive verb.

Using $\otimes_{iv}$ to represent the relation corresponding to an intransitive verb, p_{iv} for the sentence pattern 1, we have

$$p_{iv} \subseteq O(t1) \otimes_{iv} O(t2)\,. \qquad (17)$$

where O(t) is an object with the time t as its part.

3) Transitive verb

In the case of transitive verbs, the verb or the verb together with its direct object constitutes a relation between two objects.

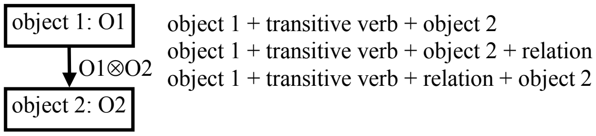


Fig. 9 Transitive verb.

An example of such a sentence pattern is "the rivet setting tool put rivets into an assembly of the brake lining and shoe".

Using $\otimes_{tv}$ to represent the relation corresponding to an transitive verb, p_{tv} for the sentence pattern 3, 4, and 5, we have

$$p_{tv} \subseteq O1 \otimes_{tv} O2\,. \qquad (18)$$

In describing a design requirement, the above five sentence patterns either describe the constraining or the constrained engineering system.

Process of Formalizing Design Requirements

As was indicated in the introduction of this paper, the input of the formalization process is the design requirements described by natural language while the output is the formulation of these design requirements. To transform the design requirements from the first representation to the second, the procedure given below should be followed:

1) Identify nouns in each design requirement and make each noun an object.
2) Identify the verb in each design requirement and make the verb a relation in terms of the three verb forms.

3) Uncover objects implied in describing the design requirement, including environment, subject and/or object missed in the design requirement.
4) Assign the objects and relations to either constraining or constrained engineering system, except the linking verb between these two systems.
5) Assign the relation corresponding to the linking verb of these two engineering systems to the relation λ in Equation (9).
6) The above processes repeat for all design requirements.

This formalization process is summarized in Fig. 10. The implementation will be introduced in a separate paper.

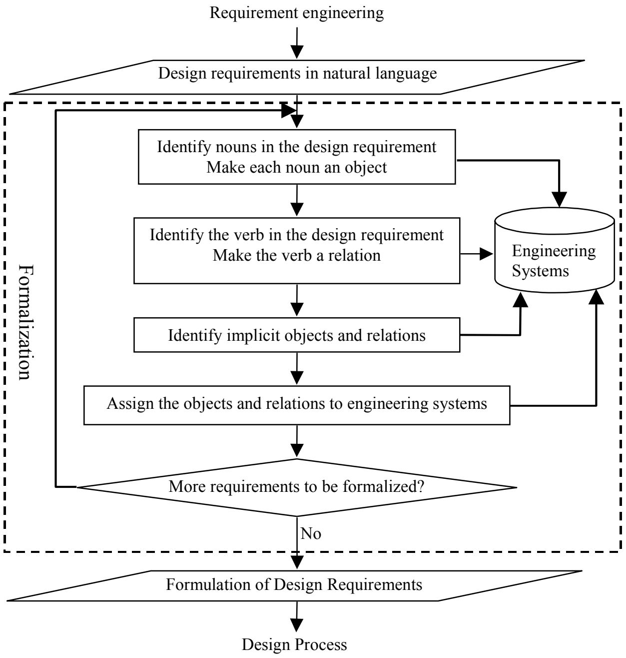


Fig. 10 Formalization process of design requirements.

EXAMPLE: FORMALIZATION OF FUNCTIONS

This section will use the formalization process in Fig. 10 to formalize design functions. A design function usually states what a product can or should do. Among many approaches to modeling functions (e.g., Deng *et al*, 2000; Chittaro and Kumar, 1998; Lossack *et al*, 1998; Gui and Mantyla, 1994; Rane and Issac, 1990; Grabowski and Benz, 1989; Pahl and Beitz, 1988), two major function models are generally accepted in the design research community. One defines a function as a relation between the input and output of energy, material, and information (Pahl and Beitz, 1988). Another represents a function in the form of verb-noun phrase (Miles, 1989). For example, in terms of input-output model, one of the gear's functions is "to transmit one force to another" whereas according to the verb-noun model the same function is "to transmit force". This will be the focus of the next two subsections. This section will show that both can be formalized following the procedures in Fig. 10.

Input-output model

Fig. 11 gives an example of a function of the rivet setting tool represented with the input-output function model. Here, the input and output are defined with respect to the function verb "riveting". The input and output are two states of the brake assembly.

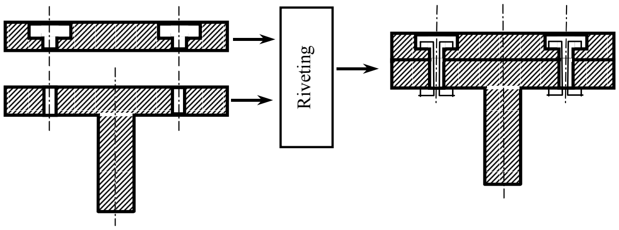


Fig. 11 Function of rivet setting tool (Hubka *et al*,1988).

This functional requirement corresponds to the relation defined by transitive verbs in Fig. 9. By comparing it with the engineering system in Fig. 3, three major differences can be identified: 1) the function is defined centred on a function verb while the engineering system is defined centred on product and environment, which are nouns; 2) the use of the function verb is artificial, subjective, and domain dependent. For the gear example, the verb 'transmit' can be replaced by 'change to' and many others without altering its meaning. This replacement does not bring in much new insight into the original function. Instead, it makes it a challenging subject to study the ontology of function representation (Gershenson and Stauffer, 1996; Kumar and Upadhyaya, 1997). On the other hand, the replacement of product in the engineering system often brings in new opportunity leading to different designs. For the gear example again, "gear" can be replaced by "belt" or "pulley". These are new alternatives for the transmitting or the changing of the force; 3) the input and output for a function verb can be anything that may or may not be physically dependent on the verb. In the engineering system, the input and output for the product and environment must be related to the product ($S\otimes E$, $E\otimes S$, $S\otimes S$, and $E\otimes E$).

Following the procedures illustrated in Fig. 10, the function given in Fig. 11 can be represented using the notion of engineering system:

1) The nouns in this description include: brake assembly ($\chi(t)$) of linings ($\chi_l(t)$), shoes ($\chi_s(t)$), and rivets ($\chi_r(t)$) before and after riveting.

$$\chi(t) \supseteq \chi_l(t) \cup \chi_s(t) \cup \chi_r(t)\,. \tag{19}$$

2) The verb in this function description is "riveting", which can be defined as a relation in Equation (20).

$$\text{riveting} \subseteq \chi(t_b) \otimes \chi(t_a)\,. \tag{20}$$

where t_b and t_a represent the time before and after riveting.

3) The objects implied in this function description include the objects that can be used as the subject of the verb “riveting”, which is the “rivet setting tool” (denoted by ($\chi_t(t)$), and the environment in which the system works (denoted by E).

$$\Omega_f = \chi(t) \cup \chi_t(t) \cup E. \tag{21}$$

4) By apply the structure operation in Equation (7) to Equation (21), the engineering system ($\oplus\Omega_f$) corresponding to this function is established. This is shown in Equation (22) and Fig. 12.

$$\begin{aligned}\oplus\Omega_f = {} & (\oplus\chi(t)) \cup (\oplus\chi_t(t)) \cup (\oplus E) \cup \\ & (\chi(t) \otimes \chi(t)) \cup (\chi_t(t) \otimes \chi_t(t)) \cup (E \otimes E) \cup \\ & (\chi(t) \otimes \chi_t(t)) \cup (\chi_t(t) \otimes \chi(t)) \cup (\chi(t) \otimes E) \cup \\ & (E \otimes \chi(t)) \cup (\chi_t(t) \otimes \mathrm{E}) \cup (\mathrm{E} \otimes \chi_t(t)).\end{aligned} \tag{22}$$

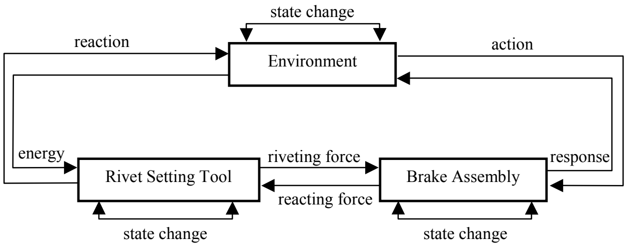


Fig. 12 Representation of a function of the rivet setting tool using the engineering system.

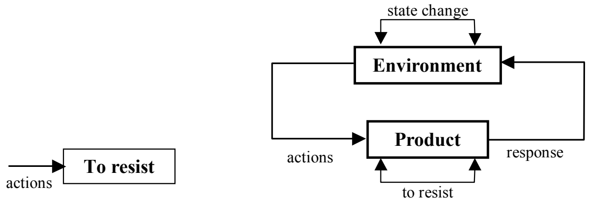
5) Equation (22) has defined the range of the engineering system to be designed. Assume that the desired system to be $\oplus\Omega$ and that the relation λ to be “is an instance of”, then the design requirement corresponding to this function is

$$r^d = \lambda(\oplus\Omega, \oplus\Omega_f). \tag{23}$$

Verb-noun model

In the “to verb-noun” function model, the verb is transitive. In terms of the definition of the verb, transitive verbs can be divided into two categories: passive and active. Passive verbs accept actions while active verbs initiate actions. Correspondingly, there are two types of functions: passive and active (Hubka *et al*, 1988). Passive functions show how a product should respond to external actions by accepting or allowing the actions whereas active functions show how a product acts on other products.

Passive functions have the pattern “to resist external actions passively”. It is shown in Fig. 13a). The actions may come from human being, other external agents, or other parts of the product. To represent it using the engineering system, the involved objects and relations need to be identified following the procedures given in Fig. 10. It can be transformed into the form shown in Fig. 13b), which can be read as “this type of product can resist, accept, or allow xx action by giving xx response”.



a) b)

Fig. 13 Passive function.

For example, one of rivet’s functions is “to deform under external forces. This function is illustrated in Fig. 14.

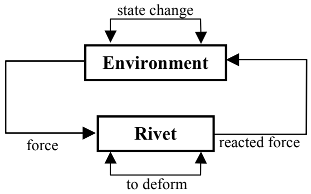


Fig. 14 A function of the rivet.

Active functions have the pattern “to act on the surrounding products”. Two explicit objects can be identified from the pattern. The general form of this function can be represented as in Fig. 15.

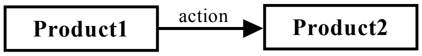


Fig. 15 Pattern of active function.

By adding implicit objects implied in the definition of function, the engineering system corresponding to the function can be recovered as in Fig. 16.

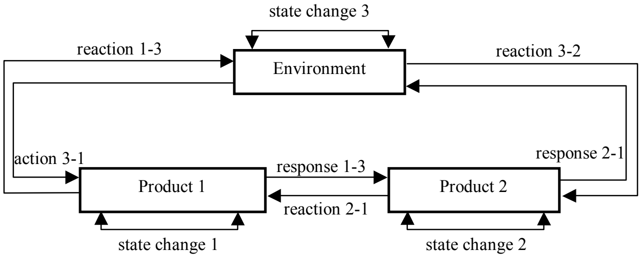


Fig. 16 Active function.

Still for the rivet example, it has another function “to connect components”, as in Fig. 17.

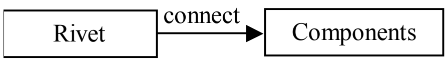


Fig. 17 “Connect” function of rivet.

This function can be formalized into the engineering system in Fig. 18.

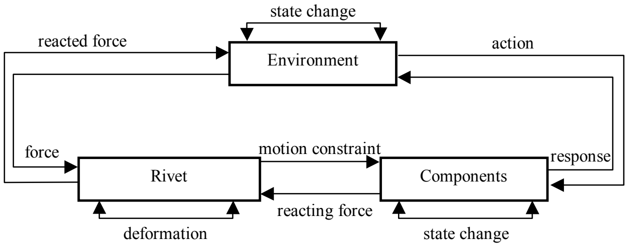


Fig. 18 Active function of rivet.

The above examples show how a design function can be formalized into an engineering system using the formalization process established in this paper.

CONCLUSION

In this paper, a formalization process is derived to transform design requirements described with natural language into their formulations. This process provides a step-by-step approach to establishing the engineering system implied in design requirements. Examples from formalizing design functions have shown the usefulness of this approach. Since the basis of this derivation process is the axiomatic theory of design modeling established earlier by the author, the work presented in this paper can also be seen as a validation of the axiomatic theory.

At the core of the formalization process are: 1) the formal definition of design requirements as constraints on an engineering system; 2) the linguistic structure of the natural language describing design requirements. The definition of design requirements involves two engineering systems: the constraining and the constrained. The former imposes constraints on the later. The formalization process establishes a partial formulation of these two systems based on the design requirements by identifying all objects and their relations implied in the design requirements.

Further research includes the formulation of design problems, decomposition strategies in the design problem solving, design requirement conflict resolution, and ontology for design requirements representation. In addition, it is also important to investigate the relationship between the presented work and the entity-relationship model proposed by Chen (1983).

ACKNOWLEDGEMENTS

The author thanks P. Gu, J. Neelamkavil, J. Dickinson, W. Shen, V. Gupta and A. Pardasani for their suggestions and comments during the process of generating the ideas presented in this paper.

REFERENCES

Chen, P., *English sentence structure and entity-relationship diagrams.* Information Science, 1983. **29**: p. 97-149.

Chittaro, L. and Kumar, A. N., 1998, Reasoning about function and its applications to engineering, *Artificial Intelligence in Engineering*, 12, 331-336.

Deng, Y. M., Tor, S. B., and Britton, G. A., 2000, Abstracting and exploring functional design information for conceptual mechanical product design, *Engineering with Computers*, 16, pp. 36-52.

Gershenson, J. K. and Stauffer, L. A., 1996, The validation of a taxonomy for manufacturability design requirements, *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, August 18-22, 1996, Irvine, California, 96-DETC/DTM-1514.

Grabowski, H. and Benz, T., 1989, Implementing the design methodology, in *IFIP WG5.2 Workshop on Intelligent CAD*, pp. 26-29, North-Holland, The Netherlands.

Gu, P. and Sosale, S., 1999, Product modularization for life cycle engineering, *Robotics and Computer Integrated Manufacturing*, 15, pp. 331-339.

Gui, J. and Mantyla, M., 1994, Functional understanding of assembly modeling, *Computer Aided Design*, 26, pp. 435-451.

Hubka, V., Andreasen, M. and Eder, W., 1988, *Practical Studies in Systematic Design*, Butterworths.

Kumar, A. N. and Upadhyaya, S. J., 1997, Component-ontological representation of function for reasoning about devices, *Artificial Intelligence in Engineering*, 12, pp. 399-415.

Lin, L. and Chen, L. C., 2002, Constraints modeling in product design, *Journal of Engineering Design*, 13, pp. 205-214.

Lossack, R. S, Umeda, Y. and Tomiyama, T., 1998, Requirement, function and physical principle modeling as the basis for a model of synthesis, in A. Brawshaw and J. Counsell (eds.), *Computer Aided Conceptual Design'98, Proceedings of the 1998 Lancaster International Workshop on Engineering Design*, pp.165-179.

Miles, L. D., 1989, *Techniques of Value Analysis and Engineering*, Eleanor Miles Walker.

Pahl, G., Beitz, W., 1988, Engineering design: A systematic approach, Spinger-Verlag.

Rane, P and Issac, J. R., 1990, Functionality: the key concept towards integration of the product cycle, in *Proceedings of the 1991 ASME International Computers in Engineering Conference and Exhibition*, 1, pp.317-326.

Turner, R., 1971, *Grammar Review for Technical Writers*, Rinehart Press.

Zeng, Y., 2002, Axiomatic theory of design modeling, *Journal of Integrated Design and Process Science*, 6(3), pp. 1-28.