

Classification of Product Requirements Based on Product Environment

Zhen Yu Chen and Yong Zeng*

*Concordia Institute for Information Systems Engineering, Concordia University
1455 de Maisonneuve West, CB-410-16, Montreal, Quebec, Canada H3G 1M8*

Abstract: Effective management of product requirements is critical for designers to deliver a quality design solution in a reasonable range of cost and time. The management depends on a well-defined classification and a flexible representation of product requirements. This article proposes two classification criteria in terms of different partitions of product environment based on a formal structure of product requirements. The first criterion classifies the product requirements by partitioning product environment in terms of the product life cycle whereas the second classifies them by partitioning the product environment into natural, built, and human environments. A case study is used to show the feasibility of this approach. This research is the core of a web-based distributed product management system.

Key Words: product requirements, classification criteria, source of product requirements, product life cycle, priority level.

1. Introduction

Requirement management plays a more and more important role in product development as the complexity of products evolves, the demand for productivity rises, and the expectation for product quality increases. The missing of critical product requirements in the early product development stages may result in unnecessary rework of the product design and/or potential product malfunctions. To keep their businesses competitive in the ever increasing global market, many companies are investing in effective approaches for managing product requirements. This management deals with engineering changes [1], product life-cycle [2], customer requirements elicitation and evaluation [3], etc. A fundamental precedent for these kinds of management is the understanding of how the product requirements are generated and what kind of structure may underlie different requirements.

Compared with other areas of design research, the understanding of product requirements in particular and design problem in general is not well developed [4]. Many authors defined design from different perspectives. Taylor avoided the definition of design problem and took engineering design as the process of applying various techniques and scientific principles for the purpose of defining a device, a process, or a system in sufficient detail to permit its physical realization [5]. Asimow [6] defined engineering design as a purposeful

activity directed toward the goal of fulfilling human needs, particularly those that can be met by the technology factors of our culture. Alexander [7] focused on function and described design as the process of inventing physical things that display new physical order, organization, form, in response to function. Suh [8] attempted to provide a more formal foundation for the design process by viewing design as the creation of a synthesized solution in the form of products, processes, or systems that satisfy perceived needs through mapping between the functional requirements (FRs) in the functional domain and the design parameters (DPs) of the physical domain, through proper selection of the DPs that satisfy the FRs. It can be observed from these definitions that a design problem may consist of human needs, constraints, and functional requirements. In recent years, a great deal of effort has been made to elicit and organize the product requirements implied in a design problem [3,9–19]. Customer requirements, constraints, design tasks, design intent, design goals and objectives can be found in the description of a design problem [8,9,11–13,16,17,20–25]. It would be essential for the management of product requirements to identify the source of product requirements, based on which the product requirements can be classified.

This study aims at understanding the source and structure of the product requirements for developing effective approaches to manage the product requirements. This research is based on the observation that the management of product requirements relies on their formalization into design specifications. In formalizing the product requirements, three problems are observed: first, product requirement documents are usually written

*Author to whom correspondence should be addressed.
E-mail: zeng@ciise.concordia.ca
Figures 6–8 appear in color online: <http://cer.sagepub.com>

in natural language [26], which easily leads to ambiguous or distorted understanding of its user's original intent [27]. This makes the management of customer requirements challenging. Second, in developing a new product, the large volume of product requirements often includes different types of information. This may easily confuse and frustrate both designers and various requirement providers [28]. Third, in developing a product family from an original product, a variety of new product requirements are usually introduced [14,15]. It is difficult to predict what type of requirements may appear in the product development process. This requires that a flexible structure should be used to manage the product requirements [4].

This study will approach the aforementioned problems based on the axiomatic theory of design modeling [29]. This modeling approach is different from existing research methods in that the models are developed by following logical procedures rather than by intuition or generalization. First, a formal structure of design problem is introduced as the theoretical foundation of this research in Section 2. It is indicated that the source of product requirements is the product environment. In the context of product development, the product environment is related to the product life cycle. By classifying the players involved in the product life cycle, Section 3 provides a comprehensive description of the product environment, from which all product requirements could be derived. To support the succeeding design process, the product requirements are ranked in terms of the importance of their corresponding product environment components in Section 4. Section 5 uses a case study of rivet setting tool design to show the effectiveness of the concepts proposed in this article. The last section concludes the article and discusses some future research directions.

2. Formal Structure of Design Problem

This section provides the theoretical foundation for describing the concepts proposed in this study. A brief review of the axiomatic theory of design modeling [29] is introduced as a mathematical tool for formulating and formalizing the product requirements. Two theorems regarding the formal structure of the design problem are listed for further classification of product requirements. These two theorems are derived in another paper [4].

2.1 Review of Axiomatic Theory of Design Modeling

Axiomatic theory of design modeling is a logical tool for representing and reasoning about the object structures [29]. It provides a formal approach that allows

for the development of design theories following logical steps based on the mathematical concepts and axioms. The primitive concepts of universe, object, and relation are used in the axiomatic theory of design modeling. Their definitions can be found from the Random House Webster's Unabridged Dictionary as follows.

[Definition 1] The universe is the whole body of things and phenomena observed or postulated.

[Definition 2] An object is anything that can be observed or postulated in the universe.

It can be seen from the two definitions above that universe is the whole body of objects.

[Definition 3] 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.

$$R = A \sim B, \exists A, B, R, \quad (1)$$

where A and B are objects. $A \sim B$ is read as 'A relates to B'. R is a relation from object A to object B . Basic properties of relations include idempotent, commutative, transitive, associative, and distributive.

Based on these concepts, two axioms are defined in the axiomatic theory of design modeling.

[Axiom 1] Everything in the universe is an object.

[Axiom 2] There are relations between objects.

It can be seen from these two axioms that the characteristics of relations would play a critical role in the axiomatic theory of design modeling. We need to define a group of basic relations to capture the nature of object representation. Two corollaries of the axiomatic theory of design modeling are used to represent various relations in the universe.

[Corollary 1] Every object in the universe includes other objects. Symbolically,

$$A \supseteq B, \quad \forall A \exists B \quad (2)$$

where B is called a subobject of A . The symbol \supseteq is inclusion relation. The inclusion relation is transitive and idempotent but not commutative.

[Corollary 2] Every object in the universe interacts with other objects. Symbolically,

$$C = A \otimes B, \quad \forall A, B \exists C \quad (3)$$

where C is called the interaction of A on B . The symbol \otimes represents interaction relation. Interaction relation is idempotent but not transitive or associative. Based on Corollaries 1 and 2, the structure operation is developed.

[Definition 4] Structure operation, denoted by \oplus , is defined by the union of an object and the interaction of the object with itself.

$$\oplus O = O \cup (O \otimes O) \quad (4)$$

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

The structure operation provides the aggregation mechanism for representing the object evolution in the design process. Based on the structure operation, the concept of product system is introduced.

[Definition 5] A product system is the structure of an object (Ω) including both product (S) and its environment (E).

$$\Omega = E \cup S, \quad \forall E, S [E \cap S = \Phi] \quad (5)$$

where Φ is the object that is included in any object.

The product system ($\oplus \Omega$) can then be expanded as follows:

$$\oplus \Omega = \oplus (E \cup S) = (\oplus E) \cup (\oplus S) \cup (E \otimes S) \cup (S \otimes E) \quad (6)$$

where $\oplus E$ and $\oplus S$ are structures of the environment and product, respectively; $E \otimes S$ and $S \otimes E$ are the interactions between the environment and product. A product system can be illustrated in Figure 1.

[Definition 6] Product boundary, denoted by B , is the collection of interactions between a product and its environment.

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

There are two types of product boundary: structural boundary and physical interactions. The structural boundary (B^s) is the shared physical structure between a product and its environment. The physical interactions include actions (B^a) of the environment on the product and responses (B^r) of the product to its

environment. Therefore, product environment boundary can further be represented as

$$\begin{aligned} B &= B^s \cup B^a \cup B^r, \\ \forall B^s, B^a, B^r &[(B^s \cap B^a = \Phi) \wedge (B^s \cap B^r = \Phi) \wedge (B^a \cap B^r = \Phi)]. \end{aligned} \quad (8)$$

Since both environment and product may have components, structures $\oplus E$ and $\oplus S$ can further be decomposed into the structures of these components as well as their mutual interactions according to the definition of the structure operation. Equation (6) indeed presents a recursive structure of a product system.

2.2 Structure of Design Problem

A design problem can literally be defined as a request to design something that meets a set of descriptions of the request. Based on the axiomatic theory of design modeling, both ‘something’ and ‘descriptions of the request’ can be seen as objects and further as product systems in the context of formulating the design problem. Thus a design problem, denoted by P^d , can be formally represented as,

$$P^d = \lambda(\oplus \Omega_0, \oplus \Omega_s) \quad (9)$$

where $\oplus \Omega_0$ ($\Omega_0 = E_0 \cup S_0$, $E_0 \cap S_0 = \Phi$) can be seen as the descriptions of a request for the design, $\oplus \Omega_s$ ($\Omega_s = E_s \cup S_s$, $E_s \cap S_s = \Phi$) is something to be designed, and λ is the ‘inclusion’ relation (\supseteq) implying that $\oplus \Omega_s$ will be a part of $\oplus \Omega_0$ so that the designed product will meet the descriptions of the design. Obviously, if $\oplus \Omega_s$ is a part of $\oplus \Omega_0$, then Equation (9) is satisfied. At the beginning of the design process, $\oplus \Omega_s$ is an unknown and $\oplus \Omega_0$ is the only thing defined. The truth value of P^d is undetermined, which means the request is yet to be met.

According to Equations (6) and (7), one can have

$$\begin{aligned} \oplus \Omega_0 &= (\oplus E_0) \cup (\oplus S_0) \cup B_0, \\ \oplus \Omega_s &= (\oplus E_s) \cup (\oplus S_s) \cup B_s. \end{aligned} \quad (10)$$

Based on Equations (9) and (10), the following equation can be derived using the axiomatic theory of design modeling [4]:

$$P^d = \lambda(B_0^s, B_s^s) \wedge \lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r), \forall E, \quad (11)$$

where the symbol \wedge denotes logical ‘and’, and both B_0^i and B_s^i are defined as follows:

$$\begin{aligned} B_0^i &= (E_i \otimes S_0) \cup (S_0 \otimes E_i), \\ B_s^i &= (E_i \otimes S_s) \cup (S_s \otimes E_i). \end{aligned} \quad (12)$$

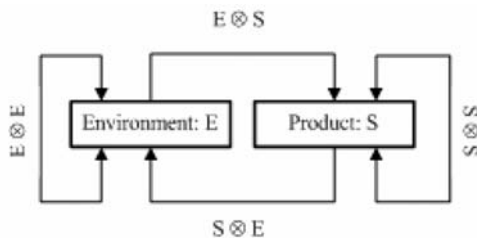


Figure 1. Product system.

Equations (11) and (12) imply the following two theorems [4]:

THEOREM OF STRUCTURE OF DESIGN PROBLEM

A design problem is implied in a product system and is composed of three parts: the environment in which the designed product is expected to work; the requirements on product structure; and the requirements on performances of the designed product.

THEOREM OF SOURCE OF PRODUCT REQUIREMENTS

All the product requirements in a design problem are imposed by the product environment in which the product is expected to work.

In general, the product environment can be partitioned into a finite number of sub-environments:

$$E = \bigcup_{i=1}^n E_i, \quad \exists E_1, E_2, \dots, E_n, \quad \forall i, j, i \neq j, E_i \cap E_j = \Phi, \tag{13}$$

where n is a finite positive number. Each E_i can be an individual environment. This partition of product environment decomposes the product requirements through Equation (11) and Equation (12).

These two theorems define an invariant part of a design problem, which is the product environment. All components in a design problem can be defined through the product environment. Therefore, the product environment provides a foundation for the classification and management of the product requirements.

Particularly, since design is a recursive process of generating requirements by the demand side and satisfying it by the supply side, which is usually the designer, product environment could be defined by all the players included in the demand side, as shown in Figure 2. These players perform different functions in the product life cycle. It will be useful for the effective management of the product requirements to classify and order these requirements in terms of product life cycle.

3. Product Life Cycle

In the previous research, product life cycle is usually studied in terms of the phases that occur according to the time sequence. However, the explicit chronological order is not always helpful for identifying the different demand sides.

Based on the observation, the product life cycle is divided into seven kinds of events, which are design, manufacture, sales, transportation, use, maintenance, and recycle, as shown in Figure 3.

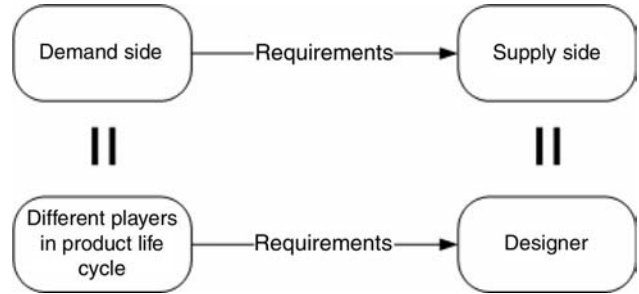


Figure 2. Demand side and supply side of design process.

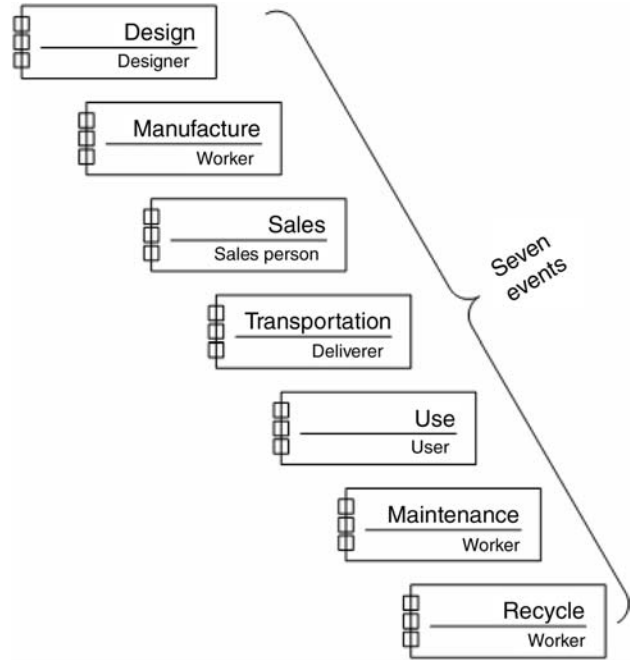


Figure 3. Seven events in product life cycle.

and recycle, as shown in Figure 3. Some of these seven events may occur simultaneously or alternatively. These seven events and their relevant actors will be discussed in this section by using examples from mechanical and software design.

3.1 Design Event

Design event, refers to the process from the generation of requirements to the acceptance of the relevant solution. In this event, the designer collects, analyzes, and satisfies the product requirements from all the players in the product life cycle.

3.2 Manufacture Event

Manufacture event, in general, is the implementation process of the design solution into a real product. In this event, manufacturing managers and workers

play major roles while other supporters may also contribute to the requirements. For instance, a product may need a special material that is difficult to purchase from the market. Design solution may have to be adjusted accordingly.

3.3 Sales Event

Sales event offers available product to the clients and customers. In most cases, sales people or marketing people provide a large number of competitive requirements to the designers, such as product price, new features of the competitors' products. All these factors may be related to the customer's purchase habit or the client's available resources and specific needs.

3.4 Transportation Event

Almost none of the software providers worry about the transportation issues. A software system of hundreds of megabytes can quickly be downloaded through high-speed Internet. Before 1994, 3.5" floppy disk was the most popular media between computers. The entire installation of Windows 3.x had to be compressed into less than 20 floppy disks. Similarly, civil engineering and mechanical engineering products, may be confronted with problems of weight, cubage, and capacity. For example, prefabricated concrete components are needed in constructing a bridge, thus their weight and size should be considered in design solution in terms of the transportation vehicles and paths they will be carried on to the destination.

3.5 Use Event

Use event refers to the process of the product being used by the end user. In this event, if the product is used to provide a service, then it is possible that both service provider and service receivers will generate requirements.

3.6 Maintenance Event

Maintenance event can be categorized into four different types in terms of different purposes: routine maintenance, corrective maintenance, perfective maintenance, and adaptive maintenance.

Routine maintenance is needed to recalibrate the product to remove or clean the waste or to replace the parts of the product that are consumable or wearable. For example, the engine oil change in vehicle maintenance, fuse change of electric stove, etc.

Corrective maintenance [30] happens when the product has problems against the product requirement. Examples include the service package or an upgrade patch package from the software providers. The recalls

by automobile manufacturers also fall into this category.

Perfective maintenance [30] helps to improve the product performances or add new functions to the product. Some customized features belong to this aspect, such as the installation of a CD player in a car.

If the environment in which a product works changed, the product may need to go through an adaptive maintenance [30], which refers to the adaptation made to help the product to suit the new environment. For example, a software product ported to a new compiler, operating system, or hardware.

3.7 Recycle Event

This event happens when the product reaches its retirement or removal of the product installation is needed.

In correspondence to the structure of the design problem, the above seven events represent seven individual environments, which are design, manufacture, sales, transportation, use, maintenance, and recycle. They are denoted by E_{ds} , E_{mf} , E_{sl} , E_{tp} , E_{us} , E_{mt} , and E_{rc} , respectively. Hence,

$$E = E_{ds} \cup E_{mf} \cup E_{sl} \cup E_{tp} \cup E_{us} \cup E_{mt} \cup E_{rc}.$$

These seven environments partition product requirements in terms of Equations (11) and (12).

4. Level of Requirements

As can be seen from Section 3, the source of product requirements is various and the number of requirements for a single product may be huge. It is usually challenging to design a product to satisfy all the requirements. Hence, it is necessary to rank all the requirements so that designers can easily know which requirements have higher priority.

In Figure 4, the product requirements are categorized into eight levels: natural laws; social laws and regulations; technical limitation; cost, time, and human resource; basic functions; extended functions; exception control level; and human-machine interface. In this pyramid-like model, those requirements at the lower levels have higher priority in developing a design solution, and those meeting the requirements at the highest level are called high usability products.

All products must be built based on the natural laws and rules. In the second place, a designer must follow social laws, regulations, and other mandatory criteria. Then, the designer takes technical limitations into consideration when design solutions are formulated, after which the designer has to make sure that the budget, schedule, and human resource demand are

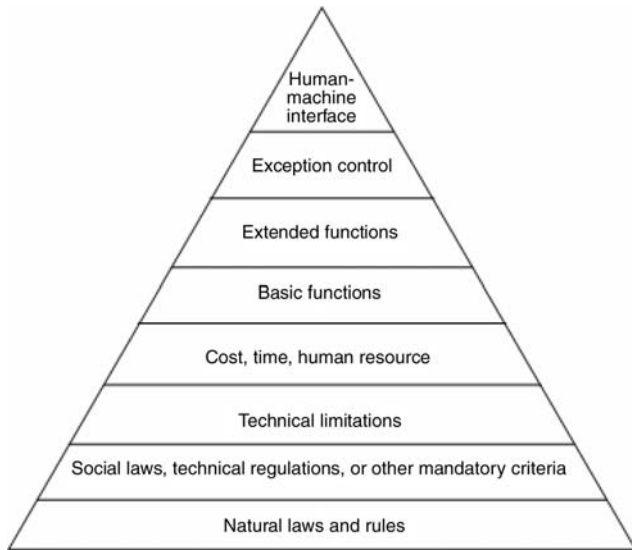


Figure 4. Eight levels of requirements.

within an acceptable range. These four levels of requirements are the basic conditions for a product to be born and exist physically in its environment. On the basis of satisfying these four levels of requirements, designers may start to realize basic functions. After all basic functions are achieved, designers can start to consider extended functions, exception control, and human-machine interfaces.

The four lowest levels are objective requirements that are almost impossible to be changed. In other words, natural laws, social laws, and technical limitations are usually not changed for a product. Compared with these lower three levels, the fourth level possesses certain degrees of flexibility such as the adjustment of capital, schedule, and human resource and has an acceptable scale. With the paucity of financial, time, and human resources, it may happen that some of the basic functional requirements cannot be fully satisfied, and have to be removed from the basic functions level, and be realized later in the extended functions level or above. As a result, those requirements in the highest four levels are not intrinsic qualities of product requirements. The distribution of higher-level requirements relies on the capacity to satisfy the lower-level requirements in the design process. Under different circumstances defined by the lower four product requirements, different requirements will be put in the level of basic functions, extended functions, exception control, and human-machine interface.

In this model, higher-level requirements can be considered after lower-level requirements are satisfied. Basically, this pyramid-like model can be divided into two major groups: non-functional requirements, and functional requirements. The lower four: natural law and rules; social law, regulations; technical

limitations; cost, time, and human resource level are usually nonfunctional requirements. The upper four: basic functions, extended functions; exception control; and human-machine interface are usually functional requirements. The eight levels are discussed as follows.

4.1 Natural Laws and Rules

All products are parts of the nature from which they can never be separated [31]. Any product is not able to escape from natural law; that is why the perpetual motion machine can never be made true.

4.2 Social Laws and Technical Regulations

When a design solution is being developed, if there exist relevant social laws, technical regulations, or other mandatory criteria, they should be observed first. Here are two typical examples. Since 1998, Canada has required all new cars sold to have daytime running lights. That means that the car should be running with its headlights on regardless of the sight condition. An electric and electronic product used in North America must be designed to use a 120 V power supply, though it must use 220–240 V in some other countries. Electronic products sold to these regions should be designed to suit this regulation.

Requirements belonging to this level must be satisfied. A product whose performances conflict with this level of requirements has to be redesigned or discarded.

4.3 Technical Limitations

Due to various technical constraints in different contexts, considerations should be given to the technical limitations. On the other hand, some requirements put forward by the demand side may not be able to be realized with the capacity of the available technologies. Hence, adjustments in the design solutions have to be made. The following gives three examples.

Before Windows 3.x came into being around 1990, development of a software product using GUI was almost impossible to be realized. At that age, Macintosh seemed to be the only choice, if end users were eager to use icon-based operation or start application by clicking the mouse.

Before the release of Windows 95, the programmer had to handle the memory allocation very carefully, because of the limitation of 64 kB.

Before robust CAD/CAM systems were available, automobiles with complex shapes were difficult to make.

4.4 Cost, Time, and Human Resources

From the business perspective, cost, time, and human resources are primarily considered after the aforementioned three levels of requirements are satisfied [32]. To achieve more profits, almost all the enterprises would associate their product development with acceptable cost, reasonable time schedule, and appropriate investment of human resources. This association happens throughout the entire product development phase including the design, manufacture, and maintenance.

4.5 Basic Functions

Basic functions are those functions that necessitate the products to work for specific purposes. The definition of the ultimate basic functions is realized through constant negotiation between the supply side and the end user. Generally speaking, basic functions are set up at the early model (version) of the product. Basic functions should not be sacrificed for the privilege of other functions.

4.6 Extended Functions

To facilitate users in the use of the product besides basic functions, some extended functions are added to products. Those auxiliary functions help the product to meet the various demands of different users. For example, the style and layout of a product are designed in such a way that users of different tastes can choose their favorite models.

4.7 Exception Control

Exception cases have to be considered so that no serious disasters would happen and damages could be under control. This level of requirements is extremely important when the reliability of a product is vital in its use. The recovery mechanism of a database is a typical example of good exception control. An exceptional power cut will cause disastrous consequence to a running database. Fortunately, a large database, such as Oracle, Sybase, provides recovery mechanisms to rebuild the entire database from zero. A few data may be lost, rather than have nothing. A multiple-engine airplane is another example.

4.8 Human–Machine Interface

Requirements at this level introduce a high usability product to users. As defined by ‘The International Engineering Consortium’, human–machine interface (HMI) is where people and technology meet. The ISO

9241 standard defines three components of quality of use: effectiveness, efficiency, and satisfaction [33].

Effectiveness – ‘Does the product do what the user requires? Does it do the right thing?’

Efficiency – ‘Can the users learn the HMI quickly? Can they carry out their tasks with minimum expanded effort?’

Satisfaction – ‘Do users express satisfaction with the product? Does the new product reduce stress?’

Obviously, effectiveness refers to the requirements at basic and extended function levels. The other two, efficiency and satisfaction, refer to the human–machine interface level.

These eight levels of product requirements are also related to product environment. In this case, the product environment can be partitioned into natural, built, and human environments. The highest four levels of product requirements come from the human environment. They serve the purposes of human use of the product. The lowest level of product requirements comes from natural environment. The rest is the result of built environment. Denote natural, built, and human environments by E^n , E^b , E^h , respectively, then:

$$E = E^n \cup E^b \cup E^h.$$

5. Case Study

5.1 Example: Rivet Setting Tool Design

A rivet setting tool design example is adapted from the book by Hubka et al. [34] to illustrate the concepts proposed in this study. The task of this problem is to design a tool for riveting brake linings onto brake shoes for internal drum brakes. The problem is described as follows:

The tool rivets brake linings onto the brake shoes. The user of this tool is a car mechanic. The hand force, foot force, and working height should follow ergonomic standards. The use of this tool should conform to the related industry safety standards. The service life of this tool should be around 5 years. The tool should be easy for transportation and maintenance. It will be manufactured in a specific workshop that has specified equipments. The cost of this tool cannot be over \$190.00.

5.2 Requirements Classification

This is a simple requirement description. The entire requirement document has only eight sentences, but the content is inclusive. They will be analyzed sentence by sentence as follows:

The tool rivets brake linings onto brake shoes.

- belongs to **[Use event]**. It describes the basic function of the target product **{Basic functions level}**.

The user of the tool is a car mechanic.

- belongs to **[Use event]**. It defines that end users are technicians – car mechanics who may have some professional skills in using this tool. **{Human-machine interface level}**.

The hand force, foot force, and working height should follow ergonomic standards.

- belongs to **[Use event]**. It points out three aspects of ergonomic standards the target product should follow to make end user more comfortable. **{Human-machine interface level}**.

The use of this tool should conform to the related industry safety standards.

- belongs to **[Use event]**. The related industry safety standards can be regarded as a kind of mandatory criterion, sometimes associated with the labor law. **{Law and regulations level}**.

The service life of this tool should be around 5 years.

- belongs to **[Use event]**. If 5 years is the common service life required of any product, then this requirement can be classified into **{Basic functions level}**. Otherwise, it may be regarded as **{Extended functions level}**.

The tool should be easy for transportation and maintenance.

- belongs to **[Transportation event, maintenance event]**. These two requirements exceed basic functions level, primarily, they are **{Extended functions level}**. If the absence of these two functions will influence marketing performance, they may be lowered into **{Basic functions level}**.

As a matter of fact, the distribution of higher-level requirements relies on the lower-level requirements. In this case, to achieve the goal of profit, the division of basic functions and extended functions may vary.

The tool will be manufactured in a specific workshop, which has specified equipments.

- belongs to **[Manufacture event]**. The manufacturing of the tool cannot be performed in a general

Table 1. Event property of the example.

Manufacture event

- *The tool will be manufactured in a specific workshop, which has specified equipment.*

Sales event

- *The cost of this tool cannot be over \$190.00.*

Transportation event

- *The tool should be easy for transportation.*

Use event

- *The tool rivets brake linings onto brake shoes.*
- *The user of the tool is the car mechanic.*
- *The hand force, foot force, and working height should follow ergonomic standards.*
- *The use of this tool should conform to the related industry safety standards.*
- *The service life of this tool should be around 5 years.*

Maintenance event

- *The tool should be easy for maintenance.*
-

workshop, because it needs specified equipments. **{Technical limitations level}**.

The cost of this tool cannot be over \$190.00.

- belongs to **[Sales event]**. The price of this tool will not exceed \$190.00 so that it is competitive in price in the market. **{Cost, time, human resource level}**.

The above classification is depicted in Tables 1 and 2, respectively.

Obviously, original requirement descriptions are too vague and inadequate to make further analysis, such as, ‘...easy for transportation and maintenance.’, ‘...conform to the related industry safety standards.’ etc. According to different events listed in Table 1, the detailed requirements have to be specified by interviewing relevant demander, and a separate specification documentation needs to be generated for confirmation.

5.3 Formal Structure of Product Requirements

To support the specification and refinement of product requirements, the first step is to identify the product system implied in a design problem described by a natural language. The following graphic symbols are used to represent objects and relations included in English.

A word surrounded by a solid line box represents a concrete-entity that equals nouns in English:

$$\boxed{O} \quad \oplus O = O \cup (O \otimes O).$$

An arrow with a solid line represents an action relationship that equals transitive verbs in English:

Table 2. Requirements level of the example.

Law and regulations level

- The use of this tool should conform to the related industry safety standards.

Technical limitations level

- The tool will be manufactured in a specific workshop, which has specified equipment.

Cost, time, human resource level

- The cost of this tool cannot be over \$190.00.

Basic functions level

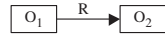
- The tool rivets brake linings onto brake shoes.
- The service life of this tool should be around 5 years.

Extended functions level

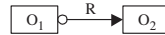
- The tool should be easy for transportation and maintenance.

Human-machine interface level

- The user of the tool is the car mechanic.
- The hand force, foot force, and working height should follow ergonomic standards.



$$R = O_1 \otimes O_2.$$



$$R = O_1 \otimes O_2$$



$$O_1 = O_2 \otimes O_2$$

An arrow with a solid line attached by a circle at the other end represents a modification relationship that is indicated by prepositions, participles, and other relationship words in English.

A word surrounded by dash-line box represents an abstract entity that equals adjectives, adverbs, or helping verbs, etc. Usually, an abstract entity is used with a modification relationship flag.

Table 3. Formalization of product requirements.

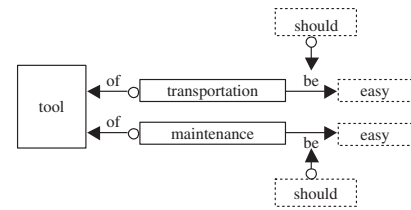
#1	The tool rivets brake linings onto brake shoes.	Basic function level use event	
#2	The user of the tool is the car mechanic.	Human-machine interface level use event	
#3	The hand force, foot force, and the working height of the tool should follow ergonomic standards.	Human-machine interface level use event	
#4	The use of the tool should conform to related industry safety standards.	Law regulations level use event	
#5	The service life of the tool should be around 5 years.	Basic function level use event	

(Continued)

Table 3. Continued.

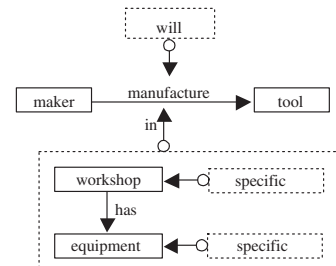
#6 The tool should be easy for transportation and maintenance.

Cost, time, human resource level transportation event maintenance event



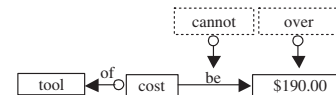
#7 The tool will be manufactured in a specific workshop, which has specified equipment.

Technical limitations level manufacture event

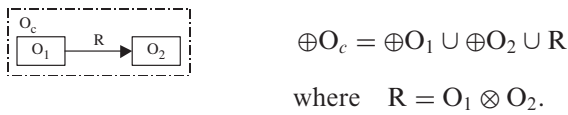


#8 The cost of the tool cannot be over \$190.00.

Cost, time, human resource level sales event



A dash-dot line box represents a composite object, which consists of other kinds of objects.



Using the symbols above, the product requirements for this problem can be formalized sentence by sentence as in Table 3.

A linguistic analysis system has been developed to generate the diagrams shown in Table 3 [35]. The input of this system is a textual product requirements document whereas its output is a product-system represented by the symbols defined at the beginning of this section. Figure 6 lists the result from analyzing the seven requirements in Table 3.

Figure 7 shows the result of classification of requirements sorted by events.

Figure 8 depicts the result of classification of requirements sorted by priority.

6. Conclusions

In this study, two criteria are proposed to classify the product requirements. The first criterion classifies product requirements by partitioning product environ-

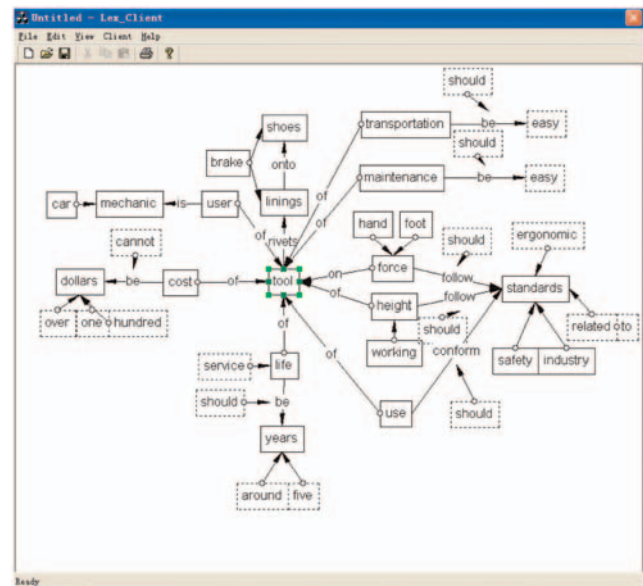


Figure 6. Product system implied in the design problem.

ment in terms of the events in the product life cycle whereas the second classifies them by the priority levels of product requirements through partitioning product environment into natural, built, and human environments. By taking the classification as a foundation, a

Sentence	Event	Level
The tool rivets brake linings onto brake shoes.	[Use Event]	(Basic functions level)
The user of the tool is the car mechanic.	[Use Event]	(Human-machine interface level)
The hand force on the tool should follow ergonomic standards.	[Use Event]	(Human-machine interface level)
The foot force on the tool should follow ergonomic standards.	[Use Event]	(Human-machine interface level)
The working height of the tool should follow ergonomic standards.	[Use Event]	(Human-machine interface level)
The use of the tool should conform to related industry safety st.	[Use Event]	(Law Regulations level)
The service life of the tool should be around five years.	[Use Event]	(Extended functions level)
The transportation of the tool should be easy.	[Transportation Event]	(Extended functions level)
The maintenance of the tool should be easy.	[Maintenance Event]	(Extended functions level)
The tool will be manufactured in a specific workshop, which h.	[Manufacture Event]	(Technical limitations level)
The cost of the tool cannot be over one hundred dollars.	[Sale Event]	(Cost, time, human resource level)

Sentence	Event	Level
The use of the tool should conform to related industry safety s.	[Use Event]	(Law Regulations level)
The tool will be manufactured in a specific workshop, which h.	[Manufacture Event]	(Technical limitations level)
The cost of the tool cannot be over one hundred dollars.	[Sale Event]	(Cost, time, human resource level)
The tool rivets brake linings onto brake shoes.	[Use Event]	(Basic functions level)
The service life of the tool should be around five years.	[Use Event]	(Extended functions level)
The transportation of the tool should be easy.	[Transportation Event]	(Extended functions level)
The maintenance of the tool should be easy.	[Maintenance Event]	(Extended functions level)
The user of the tool is the car mechanic.	[Use Event]	(Human-machine interface level)
The hand force on the tool should follow ergonomic standards.	[Use Event]	(Human-machine interface level)
The foot force on the tool should follow ergonomic standards.	[Use Event]	(Human-machine interface level)
The working height of the tool should follow ergonomic stand.	[Use Event]	(Human-machine interface level)

Figure 8. Classification of requirements sorted by priority.

rivet setting tool design is used as an example to illustrate the concepts presented in this paper. Results from a software prototype we have developed to formalize and formulate customer requirements are given to show how this research may be used as a core for a requirement management system.

Based on the work presented in this paper, a web-based distributed design software system is under development for the smooth and seamless management of product requirements throughout the product development process. Ontology is being developed to define a template for describing product environment and potential relationships between the product and its environment. The authors are also integrating a sketch-based conceptual design system [36] into this system to deal with the geometric information appearing in the description of design problem.

Acknowledgment

This work is partially supported by NSERC (Grant number RGPIN 298255).

References

1. Tavcara, J.E. and Duhovnikb, J.E. (2005). Engineering Change Management in Individual and Mass Production, *Robotics and Computer-Integrated Manufacturing*, **21**(3): 205–215.
2. Kiritsis, D., Bufardi, A. and Xirouchakis, P. (2003). Research Issues on Product Lifecycle Management and Information Tracking using Smart Embedded Systems, *Advanced Engineering Informatics*, **17**(3–4): 189–202.
3. Chen, C.H., Khoo, L.P. and Yan, W. (2002). A Strategy for Acquiring Customer Requirement Patterns using Laddering Technique and Art2 Neural Network, *Advanced Engineering Informatics*, **16**(3): 229–240.
4. Zeng, Y. (2004). Environment-based Formulation of Design Problem, *Transaction of SDPS: Journal of Integrated Design and Process Science*, **8**(4) (in press).
5. Taylor, E.S. (1959). M.I.T. Report, MIT Press, Cambridge, MA.
6. Asimow, M. (1962). *Introduction to Design*, Englewood Cliffs, NJ: Prentice-Hall.
7. Alexander, E.R. (1979). The Design of Alternatives in Organizational Contexts: A Pilot Study, *Administration Science Quarterly*, **24**: 382–404.
8. Suh, N. (1990). *The Principles of Design*, Oxford University Press.
9. Agouridas, V., Baxter, J., de Pennington, A. and McKay, A. (2001). On Defining Product Requirements: A Case Study in the UK Health Care Sector, In: *Proceedings of ASME 2001 Design Engineering Technical Conferences and Computer and Information in Engineering Conference*, Pittsburgh, Pennsylvania, pp. DETC2001/DTM-21692.
10. Chittaro, L. and Kumar, A.N. (1998). Reasoning About Function and Its Applications to Engineering, *Artificial Intelligence in Engineering*, **12**: 331–336.
11. 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**: 36–52.
12. Gershenson, J.K. and Stauffer, L.A. (1999). A Taxonomy for Design Requirements from Corporate Customers, *Research in Engineering Design*, **11**: 103–115.
13. Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S. and Wood, K.L. (2002). A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts, *Research in Engineering Design*, **13**: 65–82.
14. Jiao, J., Tseng, M., Duffy, V.G. and Lin, F. (1998). Product Family Modeling for Mass Customization, *Computers & Industrial Engineering*, **35**(3–4): 495–498.
15. Jiao, J. and Tseng, M. (2004). Customizability Analysis in Design for Mass Customization, *Computer Aided Design*, **36**(8): 745–757.
16. Lossack, R.S., Umeda, Y. and Tomiyama, T. (1998). Requirement, Function and Physical Principle Modeling as the Basis for a Model of Synthesis, Computer Aided Conceptual Design'98, In: *Proceedings of the 1998 Lancaster International Workshop on Engineering Design*, pp. 165–179.
17. McKay, A., de Pennington, A. and Baxter, J. (2001). Requirements Management: A Representation Scheme for Product Specifications, *Computer-Aided Design*, **33**: 511–520.

18. Rounds, K.S. and Cooper, J.S. (2002). Development of Product Design Requirements using Taxonomies of Environmental Issues, *Research in Engineering Design*, **13**: 94–108.
19. Chen, C.H., Khoo, L.P. and Yan, W. (2005). PDCS-A Product Definition and Customisation System for Product Concept Development, *Expert Systems with Applications*, **28**(3): 591–602.
20. Hubka, V. and Eder, W. (1988). *Theory of Technical Systems*, Springer-Verlag.
21. Altshuller, G.S. (1984). *Creativity as an Exact Science: The Theory of the Solutions of Inventive Problems*, Gordon and Breach Science Publishers.
22. Braha, D. and Maimon, O. (1998). *A Mathematical Theory of Design*, Kluwer Academic Publishers.
23. Lin, L. and Chen, L.C. (2002). Constraints Modeling in Product Design, *Journal of Engineering Design*, **13**(3): 205–214.
24. Pahl, G. and Beitz, W. (1988). *Engineering Design: A Systematic Approach*, Springer-Verlag.
25. Yoshikawa, H. (1981). General Design Theory and CAD System, In: Sata, T. and Warman, E.A. (eds), *Man-Machine Communication in CAD/CAM, Proceedings Ifip Wg 5.2 Working Conference*, North Holland, Amsterdam, pp. 35–38.
26. Chen, P. (1983). English Sentence Structures and Entity-Relationship Diagram, *Information Sciences*, **29**: 127–149.
27. Oxman, R. (2004). Think-Maps: Teaching Design Thinking in Design Education, *Design Studies*, **25**: 63–91.
28. Darlington, M.J. and Culley, S.J. (2004). A Model of Factors Influencing the Design Requirement, *Design Studies*, **25**(4): 329–350.
29. Zeng, Y. (2002). Axiomatic Theory of Design Modeling, *Transaction of SDPS: Journal of Integrated Design and Process Science*, **6**(3): 1–28.
30. Schach, S.R. (2004). *Object-Oriented and Classical Software Engineering*, McGraw-Hill.
31. Zeng, Y. and Cheng, G.D. (1991). On the Logic of Design, *Design Studies*, **12**(3): 137–141.
32. Adolph, W.S. (1996). Cash Cow in the Tar Pit: Reengineering a Legacy System, *IEEE Software*, **13**: 41–47.
33. Consortium, T.I.E. The Human-Machine Interface (HMI), <http://www.iec.org>.
34. Hubka, V., Andreasen, M. and Eder, W. (1988). *Practical Studies in Systematic Design*, Butterworths.
35. Chen, Z.Y., Yao, S.J. and Zeng, Y. (2004). A Systematic Approach for the Specification of User Requirements, *The Inaugural CDEN Design Conference*, Montreal, Canada.
36. Zeng, Y., Pardasani, A., Antunes, H., Li, Z., Dickinson, J., Gupta, V. and Baulier, D. (2004). Mathematical Foundation for Modeling Conceptual Design Sketches, *Transactions of the ASME: Journal of Computing and Information Science in Engineering*, **4**(2): 150–159.

Mr. Zhen Yu Chen



Mr. Zhen Yu Chen is a Master's student in the Concordia Institute for Information Systems Engineering at Concordia University, Montreal, Canada. His research interests include requirements engineering, service engineering, and image processing. He received his BEng. degree from Zhejiang University of Technology in

1994. Prior to his graduate studies, he worked as a software developer and project manager in the telecommunication and manufacturing industries in China for 10 years.

Dr Yong Zeng



Dr Yong Zeng is an Associate Professor in the Concordia Institute for Information Systems Engineering at Concordia University, Montreal, Canada. He is Canada Research Chair in Design Science. He received his BEng. degree in structural engineering from the Institute of Logistical Engineering and MSc and PhD degrees

in computational mechanics from Dalian University of Technology in 1986, 1989, and 1992, respectively. He received his second PhD degree in design engineering from the University of Calgary in 2001. His research interests include the science of design, requirements engineering, computer-aided design and manufacturing, computational geometry, and finite element modeling.