Formalization and Classification of Product Requirements Using Axiomatic Theory of Design Modeling

Zhen Yu Chen

A thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

April 2006

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-14255-3
Our file Notre référence
ISBN: 0-494-14255-3

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

ABSTRACT

Formalization and Classification of Product Requirements

Using Axiomatic Theory of Design Modeling

Zhen Yu Chen

The objective of the present thesis is to transform the customer requirement described in a natural language into a formal specification. The effective specification of product requirements is critical for designers to deliver a quality design solution in a reasonable range of cost and time. No significant research results have been reported in the literature regarding the generation of formal design specification, regardless of its important roles in product development. A good specification depends on a well-defined classification and categorization of product requirements, on flexible means of representation that can capture the various structures behind the requirements, and also on the formalization of natural language based description of the design problem.

The present thesis proposes two criteria to classify the product requirements based on the structure of the product-environment. A graphical language, named ROM, is developed to represent the formal specification of the design problem. Since majority of design problems are described in a natural language, the relationship between the formal specification and the structure of natural language is investigated in the present thesis by the formal linguistic analysis. A software prototype, ROMA, is developed to facilitate the transformation of product requirements in a natural language to the formal specifications.

An important difference of the current research from others in the same field is the application of the axiomatic approach. All the research results in the present thesis are derived from axiomatic theory of design modeling.

Acknowledgements

The first person that should be acknowledged for his roles in the present thesis is my supervisor, Dr. Yong Zeng. I owe an immense debt of gratitude to him for his knowledgeable and helpful suggestions and for his constant encouragement and support. Dr. Zeng has offered me extensive and thoughtful comments throughout my entire master's program. The comments and the patience that he has displayed are very much appreciated. I have benefited enormously from the discussions with Dr. Zeng, especially from his inspiring guidance in research methods, which will prove to be helpful in my future study and work. The outstanding organizational skills that Dr. Zeng has displayed have made a friendly, efficient and cooperative working environment for the Design Group in CIISE.

My gratitude also goes to Lan Kong, Baiquan Yan, Mingbin Chen, Shenji Yao, Bo Chen and other members in the group, who have shared all the joyful and bitter moments with me in the past two years. Their invaluable suggestions and friendship is an asset in my life. I am also grateful for Dr. R. Raut, Dr. A. Bulgak, and Dr. A. Hammad, who read my thesis and provided me with constructive suggestions. A heartfelt note of thanks goes to the professors and staffs in CIISE, who have provided a wonderful academic surroundings for me to complete this study.

My family has been especially supportive of my efforts to pursue this study, exhibiting their extraordinary patience and understanding.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and significance of requirement management

Due to the ever-changing and fast-moving market requirements, thousands of products are being introduced into the global market every day. As Ullman has indicated that there is a continuous need for new, cost-effective, and high-quality products (Ullman, 2002). For example, according to a database published by FDA (U.S. Food and Drug Administration), its Center for Devices and Radiological Health had 3353 final decision rendered issues from January to December 2004, of which the average is about 10 issues per day. This accurately represents the number of new medical devices that went onto the America market in 2004, since the FDA (Section 510(k) of the Food, Drug and Cosmetic Act) requires device manufacturers to notify the FDA of their intent to market a medical device at least 90 days in advance (FDA, 2004).

Meanwhile, demands with tight-deadlines increase pressure on designers to manage user requirements according to more efficient schedules and within less capital. In fact, it has been estimated that 85 percent of the problems with new products not working as they should, and taking too long to bring to market, or costing too much are the result of a poor design process (Ullman, 2002). Not only time and cost, but also product design quality represent a real challenge. Based on their case study research on 342 medical device failures, Dolores and Kuhn said that "Among the fault types, logic appears very high at 43%; with further details, some of these faults might fit into other classes. This class includes possible errors such as incorrect logic in the requirement specification, unexpected behavior of two or more conditions occurring simultaneously, and improper limits" (Dolores and Kuhn, 1999).

Therefore, most of product faults arise from irrational designs; furthermore, the majority of irrational designs are caused by the lack of requirement analysis or a defective design process. The quality of design can be improved by achieving a better understanding of the requirements definition process. Only reasonable requirements and rational designs are able to bring us really successful products. In the present thesis, the objective is to develop a requirement management approach in considered, ordered, efficient, and extendible way.

Being the "contract" between the supplier and the demander (these two concepts are discussed in section 5.1 of this thesis), the specification is undoubtedly important from any point of view of product life cycle. Technically speaking, the specification is the totality of the formalized requirements, which are the bridge between the user requirements and the product development. Some methods and associated tools have already been developed to make conceptual designs and detail designs. For instance, Unified Modeling Language (UML) is a telling example of a conceptual design tool. Besides, the specification generation starting from the initial requirement has received little attention compared to other stages of the design process, such as the conceptual design, the tests, which can more directly affect final products. Coupled with the development of engineering and technology, this weak but critical chain in the entire design process is becoming a bottleneck that one can not ignore.

1.2 Benefits of formalized specification

Besides meeting the needs created by numerous new products, modern industry tends to make products that are becoming more and more complex and that are growing ever larger in scale and ever more diverse in functionality. The probability that a product will cause frustration is in direct proportion to the product's complexity. Most formalized specifications based upon mathematics-like language facilitate communication and the description of problems and concepts.

An ideal specification document can help eliminate errors earlier and easier. Because it is as follow:

- Concise, only the necessary information is included;
- Unambiguous, it does not lead to more than one way of understanding for the product requirements.

A formalized specification should have the following:

- a high flexibility in order to specify requirements at various abstract levels
- a high manipulability so that the designers can develop solutions easily in the conceptual and detail design.

In some safety critical systems, formalized specification is required. For instance, in nuclear power stations, incorrect behavior may lead to hazards to a human being's health and life, or to serious environment accidents. The use of formalized specification can efficiently reduce the risks of such occurrences. Therefore, the Atomic Energy Control

Board of Canada (AECB) (Bowen and Hinchey, 1995) recommends formalized specification; the Ministry of Defense of the United Kingdom mandates it in its defence standard of MOD 00-55 (UKMOD, 1997).

So far, some formalized methods of specification have been practiced in a wide variety of design fields. In the development of software and hardware, they are useful at a number of levels of abstraction in the development process ranging from requirements capture, through to specification, design, coding, compilation and the underlying digital hardware itself (Bowen and Stavridou, 1993).

1.3 Objective

The present thesis aims to provide an effective approach to managing product requirements based on a natural language associated with new and innovative product development based on the axiomatic theory of design modeling (Zeng, 2002).

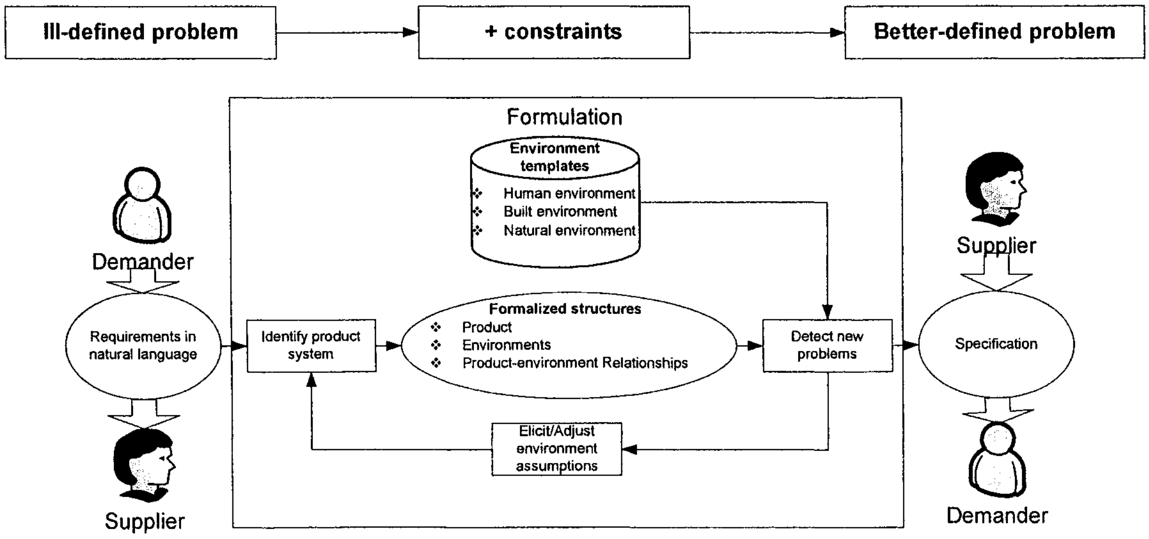


Figure 1 Problem formulation of environment-based design

A product requirement can be described in the natural language, sketches, equations, and some other forms. In the present thesis, the scope of the research object is limited in the product requirements described in natural language. The requirement described by a sketch is the research interest of Lan Kong, who is another member of our research group. The ontological representation of the product requirements have been investigating by Mingbin Chen in his master's thesis.

As shown in the top row of Figure 1, design problems are well-known as ill-defined problems (which are discussed in section 2.1); specification should be a better-defined problem after adding some constraints based on the original problem. Only in this way can the designer finally generate a satisfying solution for the problem.

The original requirements (ill-defined problems) are often provided to the supplier by the demander in the form of a natural language. During the formulation, the requirements are formalized and developed into a product system that is a formal structure including product, environments, and their relationships. In the next step, environment templates are used to the identified environments. It may lead to two situations:

1. There are some contradictions between product and environment. Therefore new design problems are generated. The designers should elicit or adjust environment assumption, and refine the product system until situation 2 occurs;
2. If there is no new design problem, the existing product system will be organized into the specification.

After the requirements have been formalized, the supplier reforms the specification in a formal structure, and submits it to the demander as the agreement between the supplier and the demander.

According to Figure 1, the formulation approach discussed in the present thesis aims at:

1. Introducing a graphic language to represent the formalized structure of a product system that is easy to understand for demanders, and facilitate to develop the design solution for the suppler.
2. Developing the prototype of a software system to aid a designer to identify a product system using the graphical language according to the requirements described in a natural language.
3. Building up a sort of environment decomposition method to help designers classify product requirements. Product requirements emerge from various sources, which depend on the "environment" according to the axiomatic theory of design modeling. Classification is one of the methods of environment decomposition, in terms of the events of the product life cycle. It is developed in order to elicit relationships on a more detailed granularity level between the product and its environment.
4. Setting up a set of rules based on the product system to identify the explicit and implicit contradictions leading to new design problems.

1.4 Challenges

In formalizing product requirements, three problems have been observed by researchers: first, the product requirement documents are usually written in a natural language (Chen, 1983), which easily leads to an ambiguous or distorted understanding of the user's original intents (Oxman, 2004). This is why product requirements are not suitable for direct computerization and management. Secondly, in developing a new product, a large number of product requirements often includes different types of information. This may easily confuse and frustrate designers and various requirement providers (Darlington and Culley, 2004). Thirdly, in developing a product family from an original product, a variety of new product requirements is introduced (Jiao and Tseng, 2004; Jiao et al., 1998). It is difficult to predict what type of requirements may appear. This kind of design problems needs the flexibility of the structure that is used to manage product requirements.

Designing is a social process, involving types of social behavior that are not easily modeled, that are difficult to evaluate, and that are outside a typical engineer's research interests. The research of design methodology is faced with these significant organizational obstacles.

Product requirements identification is intensely subjective relying on a designer's experience and knowledge. There are currently no clear systematic methodologies available. When customers find that some requirements cannot be satisfied, they must ask "Why can it not be satisfied?", "What are the criteria?" etc.

A natural language is a consummate expression developed over thousands of years, reflecting a human being's mental model. Formalizing it as a mathematical formula is almost impossible. In writing about ambiguities and inconsistencies in formal specifications in the English language, (Meyer, 1985) summarized seven "deadly sins":

- Noise: The presence of irrelevancy and unnecessary duplication which masks the basic intent of the specification
- Silence: The (unintentional) omission of parts of the intention
- Over-specification: Providing details of how the specification may be realized thereby suggesting that we employ a particular implementation which may or may not be appropriate
- Contradiction
- Ambiguity
- Forward referencing: Appealing to concepts that are defined later yet are used to make an important point early in the specification. This confuses us.
- Wishful thinking: Including some feature(s) that, despite all the goodwill in the world, cannot be realistically implemented.

1.5 Contributions

The objective of my research is the management approach of product requirements and specifications, which have received less attention than have other design stages. Nevertheless, it seems difficult to skip over this vital step in a "big" project, requirements list of which is exceedingly long. The contributions of the present thesis can be summarized as follows:

1. The product requirements include the functional requirements and the non-functional requirements. We pay more attention to the non-functional requirements than other

researchers. Non-functional requirements are as important as functional requirements in the design process. In fact, some of non-functional requirements, such as cost and schedule, have even more influence on the design process and the final design solution. These are discussed in Chapter 5.

2. The analysis of product requirements starting from a natural language is another important contribution of this research. Almost all initial product requirements are described in a natural language, which has an inherent weakness, a so-called ambiguity. We are one of the few research groups working on this problem in the design domain.
3. Unsatisfied requirements have an equally important status in the design process. Traditionally, researchers have looked at the requirements that can be satisfied, and have ignored the unsatisfied requirements. From my perspective, "satisfied" or "unsatisfied" characterizations are like tags attached to the requirements according to the designers' knowledge or experience. Because of the subjective judgment, the misunderstanding always occurs resulting in the ambiguous descriptions of requirement themselves or in a careless analysis. The unsatisfied requirements are proposed to be stored in a database associated with the reason why they can not be satisfied. They may be recalled by the movement of the environment assumptions, for example, technical advancement, or change of other environment constraints (discussed in Chapter 5).

It should be noted that all the models and methods in the present thesis are based on the axiomatic theory of design modeling (Zeng, 2002). This is different from most of the research in the literature, where observation and experimentation play a major role (Agouridas et al., 2001; Bodker, 2000; Deng et al., 2000; Gangopadhyay, 2001; Gershenson and Stauffer, 1999; Hubka and Eder, 1988; Lossack et al., 1998; McKay et al., 2001; Rounds and Cooper, 2002).

1.6 Thesis organization

Chapter 1, Introduction of the present thesis, presents the motivation, significance, objective and overview of this paper. The remainder is organized as follows:

Chapter 2, Literature Review, examines the previous research achievements dealing with the requirements analysis.

Chapter 3, Review of Axiomatic Theory of Design Modeling, is a summary of the most important theoretical foundations of this paper, which has been developed by Dr. Yong Zeng.

Chapter 4 introduces ROM language that has been developed to organize the requirements into a well defined structure. To validate the theory, a recursive object modeling analysis software prototype system is presented in Chapter 6. The majority portion of Chapter 4 and Chapter 6 was presented at the CDEN design conference in Montreal, Canada in 2004 (Chen et al., 2004). A revised and extended version has been submitted to the journal "Computers in Industry" for the consideration of publication.

Chapter 5, Classification and Categorization of Product Requirements, explains the following:

1. The product life cycle is divided into a set of events, which are design, manufacture, sale, transportation, use, maintenance, and recycle. All the requirements are classified into these seven kinds of events so that various requirement providers are able to concentrate on their respective parts, which are associated with their relevant environment.
2. All requirements are categorized into eight levels allowing the designer and decision-maker to generate or select a suitable design solution.

The research for this chapter is accepted for publication in the journal of "Concurrent Engineering Research and Applications".

In Chapter 7, Case Study, an example of rivet tool is chosen as a case study to illustrate the theory and the ROMA system.

Finally, Chapter 8 , Conclusions and Future Work, summarizes the main research result based on the present thesis, and points out future research directions.

A list of my publications during my graduate studies is provided after Chapter 8.

Chapter 2

Literature Review

As mentioned in Chapter 1, the objective of the present thesis is to do research on specification generation process starting from the product requirements. For the purpose of this research, the following goals were established:

- What is design, design problem and design process? Understanding of design, the design problem and the design process is helpful in settling the target of specification modeling.
- What is specification? The specification is the major aim of the present thesis; a correct understanding of specification is one of elementary foundations for my research direction.
- The review of the existing methods of requirements analysis aims to find an overall starting point for my research.
- The responses to a given design problem vary from person to person. There is no a criterion to judge whether a design is good or bad, therefore, the study on this problem extends the range of potential users of the method discussed into the field of general designers rather than that of some particular design groups.
- The review of the processing of natural language helps me to find the existing approaches in the language processing field.

The following subsections summarize the results from my literature review.

2.1 What is design, design problem and design process?

Many previous researchers have provided various descriptions of the term “design”: design activities are generally considered to be a form of complex problem solving (Simon, 1969); design begins with a needs-analysis (Asimow, 1962); design is a social activity (Minneman, 1991). In most design studies, the objectives usually focus on finding common characteristics from different engineering domains, within the framework of cognitive science (Goel, 1995). Therefore, design process can be regarded as a cognitive process intended to produce a solution to a design task.

Engineering design has been defined by Fielden (Feilden, 1963) as “the use of scientific principles, technical information and imagination in the definition of the mechanical structure, machine or system to perform prespecified functions with maximum economy and efficiency.”

Design problems are well-known as ill-defined and open-ended problems. An ill-defined problem, also called an ill-structured problem as distinguished from a well-structured problem, involves two critical concepts in the cognitive science. Different from a well-defined problem, such as a chess game, an ill-defined problem (a) is more complex (b) begins with the inadequate initial conditions (c) presents fewer “end” criteria. An open-ended problem does not have an optimal solution, but only a satisfying one (Simon, 1969), which may have several or many correct solutions. When provided with the same list of product requirements, different design teams produce different product solutions.

The design process varies from product to product and from industry to industry. A generic diagram of the activities that must be accomplished for all projects, is shown in

Figure 2 (Ullman, 2002). In this framework, any product must go though five phases; three of them are about the design process that are project definition, specification definition, and conceptual design. Even in the product development phase, there is the probability of refining the design solution or of canceling the entire project.

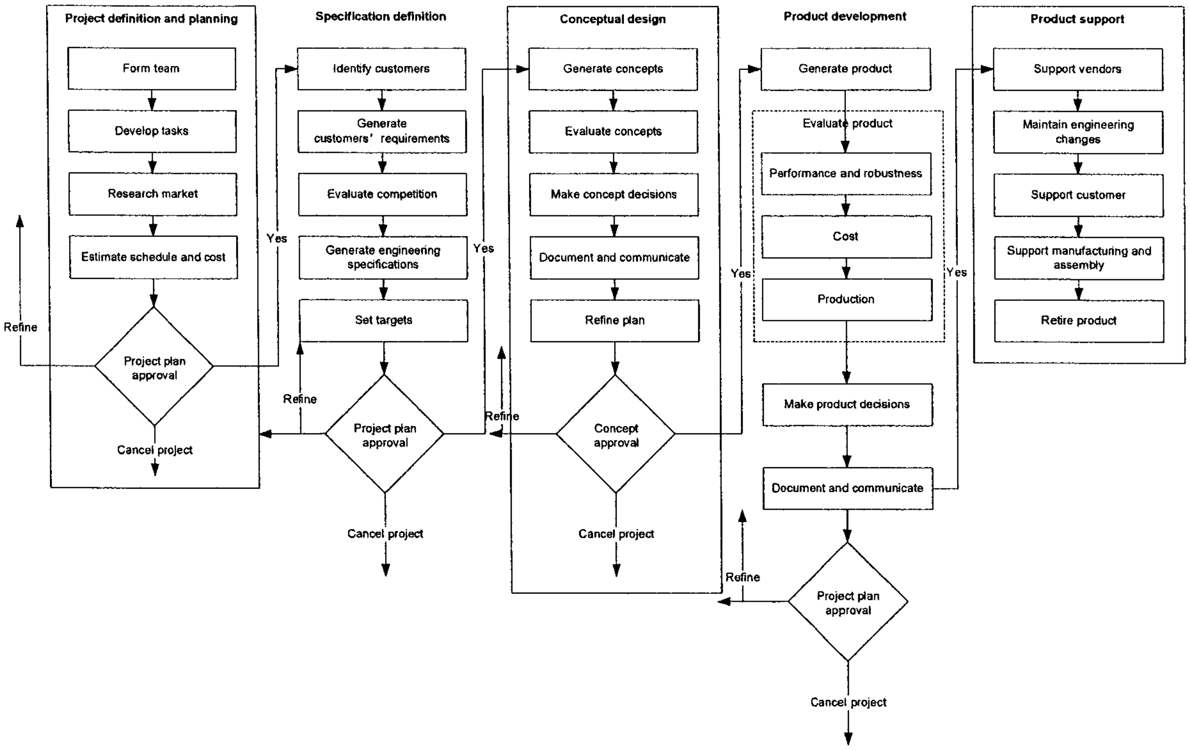


Figure 2 Ullman's generic design process in all projects (Ullman, 2002)

Pahl and Beitz have done research into the traditional design process. Their model is different from that of Ullman. Theirs is shown in Figure 3 (Pahl and Beitz, 1999). In this model, a product design process is divided into five phases, clarification of the task, concept generation, embodiment design, detail design, and physical evaluation.

Obviously, Ullman's model pays more attention on needs analysis; Pahl and Beitz's model emphasizes generation of solution and evaluation.

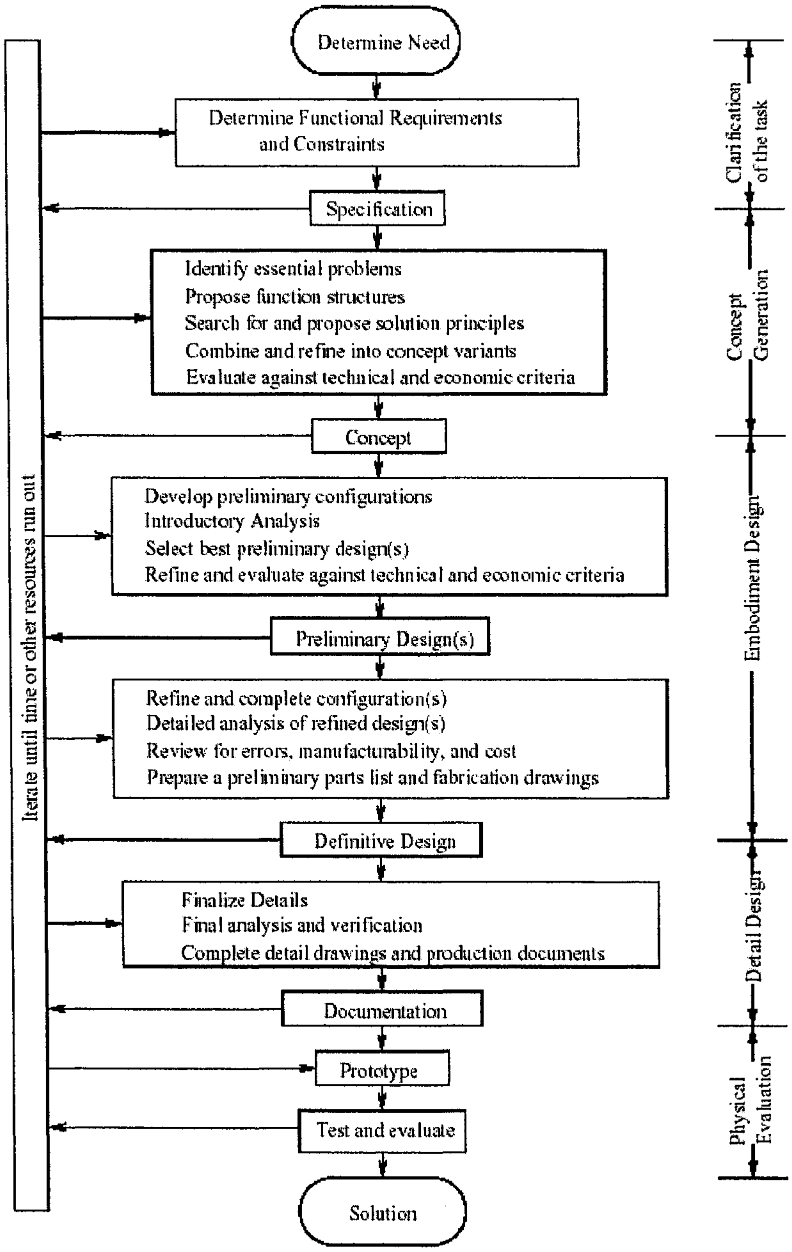


Figure 3 Pahl and Beitz's traditional design process (Pahl and Beitz, 1999)

About “design model”, Gero has conjectured that “perhaps the earliest of the widely accepted models of designing is introduced by Asimow (Asimow, 1962) who divided all the designing processes into three typical classes: analysis, synthesis, and evaluation” (Gero, 1998). The relationship among the three classes is illustrated in Figure 4.

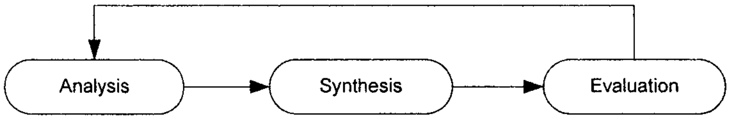


Figure 4 Analysis-synthesis-evaluation model (Asimow, 1962)

Asimow’s model was developed in subsequent research, considering that, “in the beginning of the design session that the designer not only follows evaluation by analysis but for an equal amount of time follows evaluation by synthesis. Already this behavior is different to that “predicted” by Asimov’s model. As the design session proceeds, this behavior increasingly diverges from Asimov’s model. Thus, in the last 75% of the time of the design session the predominant behavior is not that predicted by Asimov at all since it is: evaluation followed by synthesis.” (Gero, 1998). The new model of analysis-synthesis-evaluation is shown as in Figure 5 (Lawson, 1997).

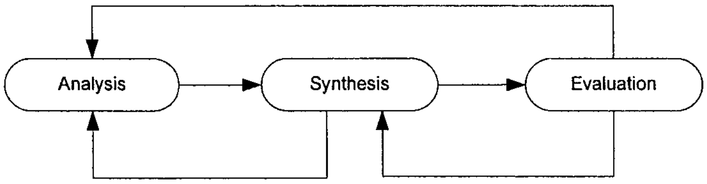


Figure 5 New model of analysis-synthesis-evaluation (Lawson, 1997)

And two more design process classes are extended from "evaluation"; they are Revision, and Implementation (Tovey, 1997).

2.2 What is specification?

The role that specification plays in the design process involves various explanations that affect different specification formalization approaches that are discussed in 2.3.

The simplest point of view interprets specification as "the list of the requirements for a project" or "the list of performance behaviors of the product".

A specification describes a system and its desired properties. The process of specification is the act of writing things down precisely. The formal specification uses a language with a mathematically defined syntax and semantics (Clarke and Wing, 1996). According to Clarke and Wing's perspective, formal methods should be described as mathematically based languages, techniques and tools for specifying and verifying safety-critical systems.

Function-structure-behavior is another critical method to describe a design target. Some researchers put forward their concepts regarding function, structure and behaviors (Hundal, 1991; Kleer and Brown, 1984; Shimomura and Takeda, 1995; Umeda et al., 1990). The intended functions are related to the purpose or intended utility of a design object. Behaviour is the totality of the properties of an object that emerge as a result of the interaction between the object's structure and its environment. The structure of a physical object is its physical embodiment, i.e. in terms of material, topology and geometry (Gero, 1990). Gero has further developed the relationship between function, behavior, and structure. It can be summarized as follows:

- *structure* exhibits *behaviour* effects *function* enables *purpose*
- *purpose* is enabled by *function* is achieved by *behaviour* is exhibited by *structure*

In summary, specification has two major functions: from the descriptive point of view, it is a tool for review or discussion, whereas from the analytic point of view, it is a mathematical model for analyzing or predicting behaviors.

2.3 Methods of requirements analysis

Some formal methods such as Z (Spivey, 1988), VDM (Bjorner and Jones, 1982; Jones, 1990), and Larch (Guttag and Horning, 1993) focus on specifying the behavior of sequential systems. States are described in terms of rich mathematical structures such as sets, relations, and functions; state transitions are given in terms of pre- and post-conditions. They are often called as Abstract Model methods. Other methods such as Communicating Sequential Processes(CSP) (Hoare, 1985), Calculus of Communicating Systems (CCS) (Milner, 1982), State charts (Harel, 1987), Temporal Logic (Manna and Pnueli, 1991), and I/O automata (Lynch and Tuttle, 1987) focus on specifying the behavior of concurrent systems.

2.4 Designers' responses to design problem

Coyne has summarized four kinds of responses to the problem of rationality(Coyne, 2005). Design tasks are usually regarded as ill-defined problems, sometimes called wicked problems (Buchanan, 1995; Kunz and Rittel, 1970; Rittel and Weber, 1973).

Some problems are defined as tame, others as wicked. The problem referred to as a tame problem is well defined, with a single goal and a set of well-defined rules to reach this goal, such as factoring a quadratic equation, traversing a maze, and solving the tower of Hanoi puzzle. Ill-defined problems are subject to redefinition and resolution in different ways over time. Ill-defined problems are not objectively given. Their formulation depends on the viewpoint of those presenting them. There is no ultimate test of the validity of a solution to an ill-defined problem. The testing of solutions takes place in some practical context, and the solutions are not easily undone.

Four kinds of responses have been proposed for design problems. They are the following: dual knowledge, pragmatic, phenomenological and narrative response. In the following sections, we discuss them separately.

2.4.1 Dual knowledge response

The initial design requirements are always described in natural languages, which use words to describe human being's emotions and feelings. To understand and satisfy what those words mean is the main task of the requirements analysis. The term "scientific rationality" is introduced to represent the trade-off between human emotion and its natural meaning from the scientific viewpoint.

The dual knowledge response does not emphasize the concept of rationality. In the real design process, the rationality has to be the balance between human emotions and feelings. However, the dual knowledge response is generally mute on the subject of how the balance is to be accomplished. A designer who employs the dual knowledge response has to be both scientist and poet.

Because there are no certain criteria governing a solution's generation and evaluation, the entire design process depends on the personal capacity of the designer or the design team. In the process of this kind of response, the evaluation is weakened by losing scale. In fact, product quality will be judged by end-users.

2.4.2 Pragmatic response

A more satisfactory response to the problem of rationality is to shift the definition of rationality so that it includes the "wicked" factors, such as value judgment, testing the context, the explanation from an authority and so on. Professional rationality cannot exist without the time-honed progress that we sometimes dismiss simply as "how we feel" about the matter at hand.

The pragmatic response asserts that a wicked problem can be described by models. However, these models stem from experiences that were accumulated thinks to a great deal of practices. Or they depend on the judgment of experts or authorities. In the process of solution generation and evaluation, experiences play main roles, even though some problem can not be measured.

2.4.3 Phenomenological

For the phenomenologist, at our core we are interpreting (hermeneutical) beings (Gadamer, 1975). Mathematics is not the language of the universe but a finely developed technique. All kind of phenomenon encountered in every-day life can be expressed in mathematical model.

Interpretation is the foundation of all being, and as such is an indeterminate, contingent, and varied foundation. Some philosophers even consider that we can explore a new language to give expression to ways of thinking outside of a rationalistic, systems-oriented frame. This new model maybe very complex because of enormous variables, but it still exists.

2.4.4 Narrative

The narrative response uncovers the underlying meaning or truth by interpreting wicked problems. A well-defined meaning structure leads to "layers of meaning", which can be used to formalize and organize design problems.

2.5 Processing of natural language

A better understanding of the requirements description can improve design quality. A few researchers have attempted to model the English language into a formal structure during the past decades. In 1983, Chen proposed a set of eleven rules for translating text from English into Entity-Relationship (ER) diagrams. (Chen, 1983)

ER diagrams are a way of displaying entity types, relationship types, and their attributes graphically. They have been widely used to create a data model, such as database schemas or object-oriented models.

Table 1 English-ER diagram translation rules (Chen, 1983)

No.	*In English*	*ER diagram*
1	A common noun (“person”, “chair”)	an entity type
2	A transitive verb	a relationship type
3	An adjective	an attribute of an entity
4	An adverb	an attribute of a relationship
5	There is(are)…X in Y	Y has(have) X
6	The X of Y is Z (if Z is a proper noun)	Y and Z = entities X = relationship between Y and Z
7	The X of Y is Z (if Z is not a proper noun)	Y = entity X = an attribute of Y Z = a value
8	The objects of algebraic or numeric	an attributes
9	A gerund	a relationship-converted entity type. (What is the relationship-converted entity type?)
10	A clause	a high-level entity type (which is abstracted from a group of interconnected low-level entity and relationship types)
11	A sentence	one and more entity type(s) + relationship

		type(s) and each entity type can be decomposed into low-level entity type(s) + relationship type(s)

The following three problems are not mentioned explicitly, but they are obvious obstacles in modeling process using entity-relationship model:

- The relationship does not have to be represented by a verb; it can be a noun (Rule 6, 7).
- According to Rule 10 and 11, an English sentence, however complex, can be iterated by simple sentences (or structures). So a complex sentence can be simplified to some simple sentences (or structures), maybe the tree-structure is a good way to present this sort of sentence.
- The pronoun and the tense are difficult to discuss. To describe the tense, a time-axis must be introduced into the ER diagrams. Thus the ER diagrams are no longer one dimensional. They must be multi-dimensional. To identify the signification of a pronoun, the historical table of all nouns cited (all nominal structures) before this pronoun (sometimes it may be cited after the pronoun) must be recorded.

Chapter 3

Review of Axiomatic Theory of Design Modeling

Axiomatic theory of design modeling is a logical tool for representing and reasoning about object structures (Zeng, 2002). It provides a formal approach that allows for the development of design theories following logical steps based on 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.

$$R = A \sim B, \exists A, B, R, \tag{3.1}$$

where A and B are objects. A~B is read as "A relates to B". R is the relation from object A to object B. The 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.

[Axiom1] 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 can be used to represent various relations in the universe.

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

$$A \supseteq B, \forall A \exists B \tag{3.2}$$

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,

$$C = A \otimes B, \forall A, B \exists C \tag{3.3}$$

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 the above Corollary 1 and 2, a new operation can be developed as follows:

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

$$\oplus O = O \cup (O \otimes O) \tag{3.4}$$

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. The definitions of this axiomatic theory can be found in (Zeng, 2002).

This chapter presents a base for representing product requirements derived using the axiomatic theory of design modeling. This base is the foundation for the formal study of product requirements. It forms a coordinate system for representing various requirements. Especially, results from this formal study will be useful for developing computer software systems that support the gathering and specification of product requirements.

In order to identify the base for representing customer requirements, some definitions are given as follows:

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

$$\Omega = E \cup S \tag{3.5}$$

According to (3.5), the product system ($\oplus\Omega$) can be represented as

$$\oplus\,\Omega = \oplus(E \cup S) = (\oplus E) \cup (\oplus S) \cup B \tag{3.6}$$

where B is product-environment boundary defined as follows:

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

$$B = (E \otimes S) \cup (S \otimes E) \tag{3.7}$$

Graphically, a product system can be represented in Figure 6.

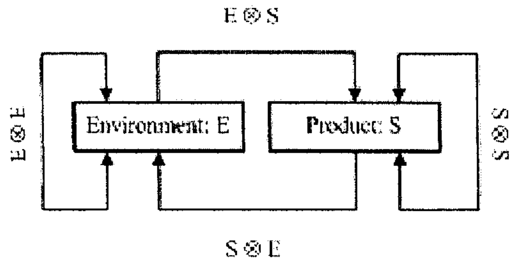


Figure 6 Product system

Product environment boundary includes structural boundary and physical interactions. The structural boundary (B^s) is the shared physical structure between the 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 the environment. Therefore, product environment boundary can be further represented as

$$B = B^s \cup B^a \cup B^r, \tag{3.8}$$

where $\cup$ denotes logical "and".

Based on the definition of product system, a design problem can be formally defined as follows.

[Definition 7] A design problem can be literally 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 can be further seen as product systems in the context of formulating design problem. Thus a design problem, denoted by P^d, can be formally represented as,

$$P^d = \lambda(\oplus\Omega_0, \oplus\Omega_s), \tag{3.9}$$

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 ($\supseteq$) impling 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 (3.9) 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 (3.6) and (3.7), we have

$$\begin{aligned} \oplus\Omega_0 &= (\oplus E_0) \cup (\oplus S_0) \cup B_0, \\ \oplus\Omega_s &= (\oplus E_s) \cup (\oplus S_s) \cup B_s. \end{aligned} \tag{3.10}$$

Since $E_i \cap S_j = \Phi, \forall i, j = 0, s$, according to Lemma 1 give in (Zeng, 2002), we have

$$P^d = \lambda(\oplus E_0, \oplus E_s) \wedge \lambda(\oplus S_0, \oplus S_s) \wedge \lambda(\oplus B_0, \oplus B_s), \tag{3.11}$$

Where the symbol $\wedge$ denotes logical "and".

Substitute (3.8) into (3.11), and according to Lemma 1 (Zeng, 2002) again, we have

$$P^d = \lambda(\oplus E_0, \oplus E_s) \wedge \lambda(\oplus S_0, \oplus S_s) \wedge \lambda(B_0^s, B_s^s) \wedge \lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r). \tag{3.12}$$

Equation (3.12) indeed can be organized into three parts:

- $\lambda(\oplus E_0, \oplus E_s)$ corresponds to requirements on product environment.
- $\lambda(\oplus S_0, \oplus S_s) \wedge \lambda(B_0^s, B_s^s)$ defines direct constraints on product.
- $\lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r)$ defines direct constraints on actions and/or responses.

Theorem of Design Problem Structure. A *design problem* is implied in a product system and 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.

This theorem can be shown in Table 2.

Table 2 Elements of design problem

Design Problem: P^d	
Product Environment	E
Performance Requirements	$\lambda(B_0^a, B_s^a) \wedge \lambda(B_0^r, B_s^r)$
Structural Requirements	$\lambda(\oplus S_0, \oplus S_s) \wedge \lambda(B_0^s, B_s^s)$

The theorem above is the foundation of the research presented in this thesis.

Chapter 4

Recursive Object Modeling Language (ROM) and English Language Processing

To formalize design problems, a formalized structure is needed to represent the problems, based on which it is possible to convert a design problem to an engineering problem that can be solved by a mathematical method.

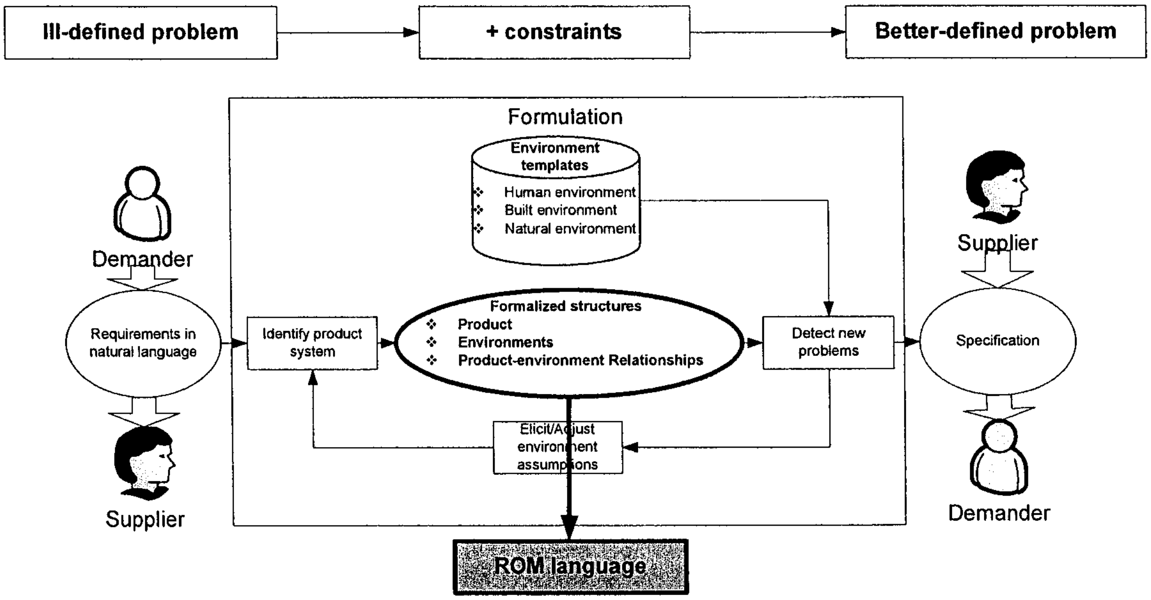


Figure 7 Formalized structure in formulation

A Recursive Object Modeling (ROM) language is proposed to illustrate the modeling components graphically in order to formalize design problems using the axiomatic theory of design modeling. This chapter introduces the ROM components related to the formalization of product requirements.

4.1 Recursive object modeling (ROM)

4.1.1 Real-entity

$\boxed{O_r}$ e.g. $\boxed{\text{book}}$

Word in a solid box represents a real-entity that is as follow:

- A concrete noun, which names a thing that one can perceive through ones physical senses: touch, sight, taste, hearing, or smell. This kind of objects can be defined as $\oplus O_r = O_r \cup (O_r \otimes O_r)$.
- The concepts that can be measured. For example, height, width, length, capacity, day, month, year, etc. They may be properties of an object.
- A proper noun, which represents the name of a specific person, place, or thing. It is always written with a capital letter.
- Numerical value, such as, 23 and 198.

Technically, the real-entity presents an unambiguous concept that is unnecessary to be more decomposed.

$\boxed{O_r}^{c}$ e.g. $\boxed{\text{book}}^{c}$

A solid box with a capital "c" means the copy of a real-entity that is totally the same as the real-entity itself. When a real-entity is referred in two different diagrams, or when a real-entity has too many relationships with other objects, or when a real-entity has a long distance to other objects, this symbol is recommended to be used in order to increase the readability of the diagram.

4.1.2 Virtual-entity

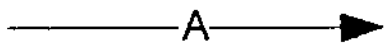
$\boxed{O_V}$ e.g. $\boxed{\text{red}}$

Word in a dashed box represents a virtual-entity, which is as follow:

- An abstract noun, which refers to states, events, concepts, feelings, qualities, etc., that have no physical existence, and is the opposite of a concrete noun, for example, “freedom”, “happiness”, and “idea”.
- An adjective, which modifies a noun. It describes the quality, state, or action that a noun refers to.
- An adverb, which modifies the meaning of a verb to describe when/where/the way the action is done or an adjective, another adverb.
- A helping verb (also known as auxiliary verb), which is used with the verb in order to create the many sorts of tenses available in English. A helping verb may be translated to time flags in time sequence diagram discussed in the following.
- Conditions and phrases or clauses.

A virtual-entity can be developed into more detailed meanings.

4.1.3 Action-relationship

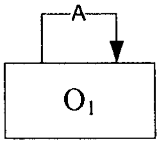


A solid arrow represents an action relationship.

- A transitive verb indicates an action performed to O_2 at the arrow side by O_1 at the start side;

$$A = O_1 \otimes O_2$$

- An intransitive verb does not take a direct object.



$$A = O_1 \otimes O_1$$

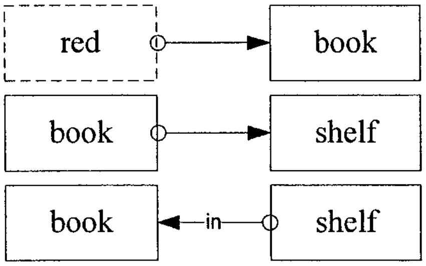
4.1.4 Modification-relationship

A solid arrow line with a circle at the line's start end represents a relationship that is indicated by prepositions and other modified relationships in the English language.

A read book

the book shelf

A book in the shelf



4.1.5 Equivalence-relationship

A solid arrow with an equal mark represents an equivalence relationship. The objects at the two ends of this kind of symbol have the exchangeable meaning.

——=——►

Figure 8 means that the "transportation action" is the equivalence of the action relationship "move". (This figure can also be seen in section 5.1.4.)

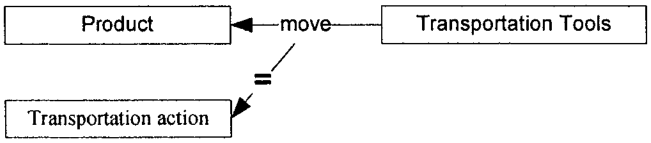


Figure 8 Example of equivalence-relationship

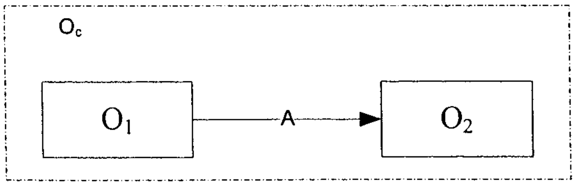
4.1.6 Status-change-relationship

- - - - - - - - - - -►

The above symbol represents a status change of an object caused by an unknown, or an undeveloped, or a complicated, or an uncared-for relationship. The symbol is a dashed line with an arrow end.

4.1.7 Composite-object

A composite-object is represented by a dashed box with other objects in it.



$$O_c = O_1 \cup O_2 \cup A$$
$$\because A = O_1 \otimes O_2$$
$$\therefore O_c = O_1 \cup O_2 \cup (O_1 \otimes O_2)$$

The combination of a composite-object and an equivalence-relationship allows the design problem to be extended from a simple requirement to a more complex description. The combination enhances the expressivity of the ROM language.

For example, a "transportation action" can be extended to a composite-object shown in Figure 10.

4.1.8 Sequence relationships

We have defined five relationships for the ROM time-sequence diagram. They are lifetime line, lifetime endpoint, lifetime period, period start point, and period end point.

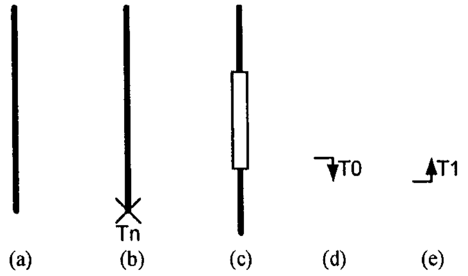


Figure 9 Components of ROM time-sequence diagram

Figure 9(a) is a lifetime line that is a vertical line under an object, defined in the axiomatic theory, that could be a real-entity, a virtual-entity, an action-relationship, a modified-relationship, or a composite-object.

A lifetime endpoint in Figure 9(b) is a check marker right above a time point variable T_n. The time point must be used with a lifetime line to end an object's lifetime. It means that the object does not exist after this time point.

A lifetime period in Figure 9(c) refers to the duration of a relationship. The start point of a lifetime period can be specified by Figure 9(d) at time T_0, and end point specified by Figure 9(e) at time T_1. The lifetime line without an endpoint means that the object is still in its life time.

Figure 10 is an example of the application of the sequence relationships and the composite-object.

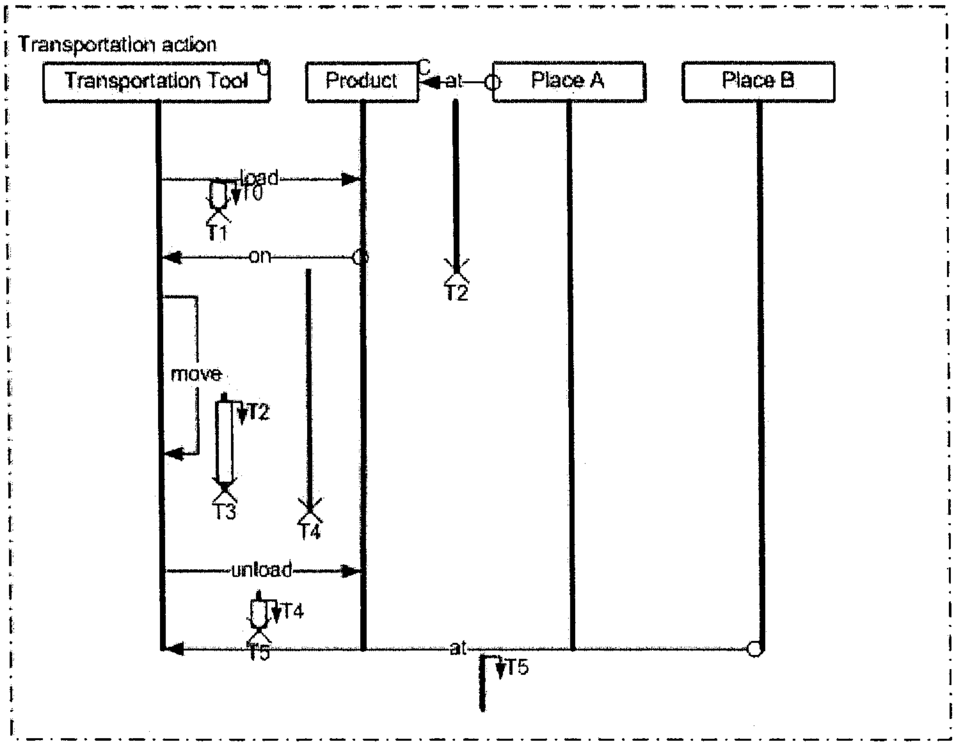


Figure 10 Example of composite-object and sequence relationships

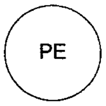
According to its definition in Merriam-Webster dictionary, “transportation” is “means of conveyance or travel from one place to another”. A typical transportation action is extended to several entities and sorts of relationships in Figure 10. The entire process can be descried as follow:

- At the initial states, a product is located at place A.
- The product is loaded onto the transportation tool. The “load” action starts at the time point T0, and ends at the time point T1. When this action is done, the relationship between the “product” and “transportation tool” can be stated as “the product is on the transportation tool”.
- The transportation tool starts to move out of place A at the time point T2 and the relationship “at” between the “product” and the “place A” is changed to “the product is not at place A anymore”. Therefore, the relationship “at” has its lifetime endpoint at the time point T2.
- The transportation tool keeps moving until it arrives at the destination place B at the time point T3.
- The product is unloaded from the transportation tool at place B at the time point T4 while the relationship “on” between the “transportation tool” and the “product” ends. Therefore, the relationship “on” has its lifetime endpoint at the time point T4.
- When the “unload” action is finished at the time point T5, the “product” arrives at “place B”, the relationship “product is at place B” starts at the time point T5.

It should be noted that the ROM sequence diagram is different from the time diagram of UML in Rational Rose. In UML, the relationship does not have life time because the relationship is a member variable or function associated with an object. In axiomatic theory of design modeling, the relationship has its life time because it is defined as a kind of object (Zeng, 2002). The relationship can emerge or disappear if the status of an object changes.

4.1.9 Environment

The capital "PE" surrounded by a circle represents the environment related to a real-entity, virtual-entity, or composite-object connected with it.



4.2 Design problem using ROM language

Based on the above conventions, the theorem of design problem structure can be graphically illustrated in Figure 11 using the ROM diagram.

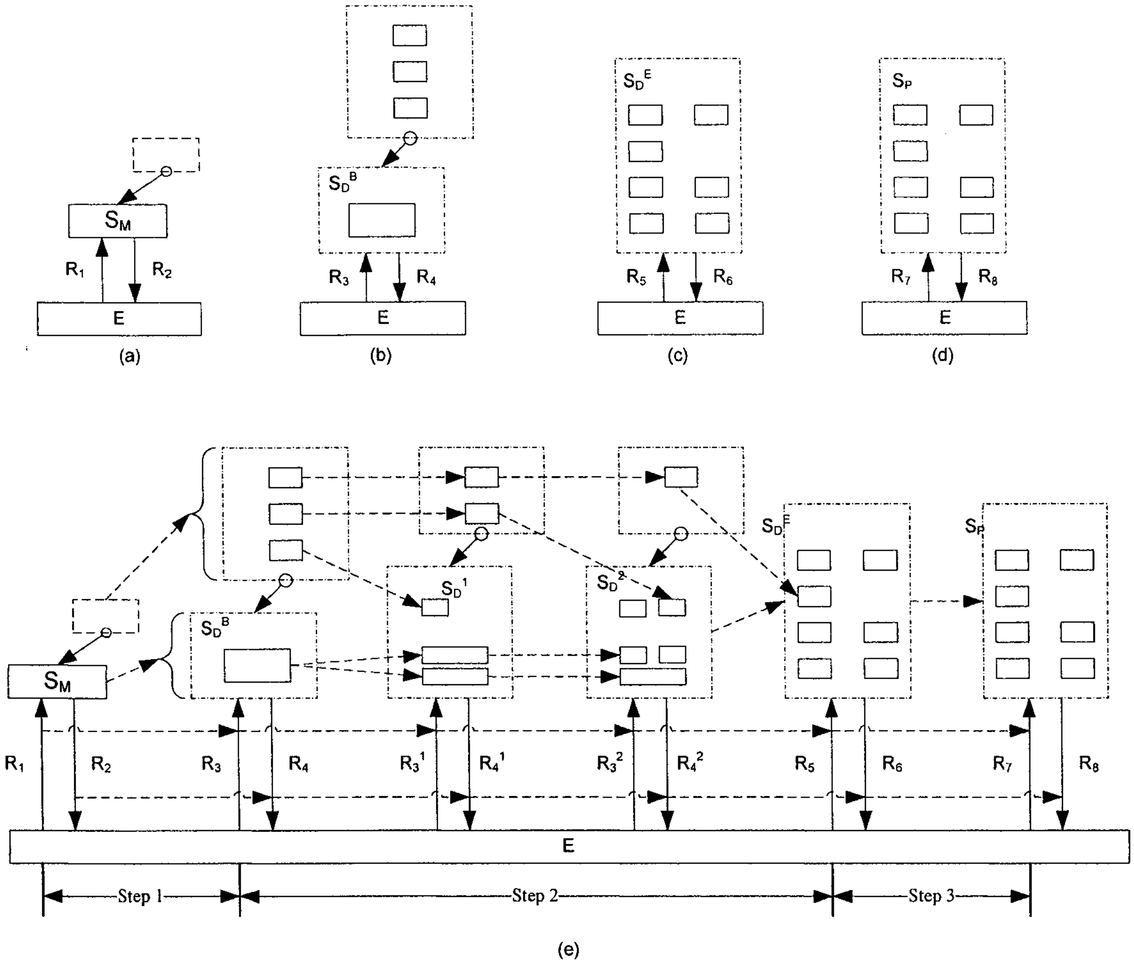


Figure 11 Evolution of design problem structure

Figure 11(a) is the original description for a design problem in a natural language, which reflects the demanders' mental model of a product. Usually, a simple target product is named in this mental model. And the rest parts are full of the plentiful adjectives and adverbs, such as "easy", "cheap", "fast", which depict demanders' desire.

Figure 11(e) shows the evolution of a design problem structure. In step1, the designer can form an initial design model (Figure 11(b)) according to the original requirements and his/her professional knowledge. The initial design model (Figure 11(b)) has more concrete concepts than demanders' mental model.

Step 2 is the design process. Most of the abstract requirements represented by virtual-entities (abstract nouns, adjectives and adverbs) are concretized to the real-entities step by step, and be converted to the structure constraints until a final solution (Figure 11(c)) is generated.

Step 3 reflects the manufacture process in which the final solution is made into a real product (Figure 11(d)).

The design process can be explained as a process that the virtual-entities are converted to the real-entities. A high quality design solution depends on building up a quality object recursive model with the well-development of the virtual-entities and the well-understanding of the real-entities.

In section 4.3, we address the linguistic foundation of the recursive object model from the English language.

4.3 Linguistic structure of product requirements

4.3.1 English words and sentences

The English language has eight traditional parts of speech in its grammar: noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection. For the sake of brevity, the present paper will only formally discuss nouns and verbs.

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. Pronouns are used as replacements or substitutes for nouns and noun phrases in a sentence.

A verb is a word used to indicate the action from/to/on an object or the state of an object. There are four principal verb types: helping, linking, intransitive, and transitive. Helping verb shades the meaning of the main verb in some desired manner. The linking verb links a noun to a complement to indicate the state of the noun. The intransitive verb itself indicates a state of a noun. The transitive verb shows actions from one object to another.

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:

Subject + intransitive verb

Subject + linking verb + subjective complement

Subject + transitive verb + direct object

Subject + transitive verb + indirect object + direct object

Subject + transitive verb + direct object + objective complement

4.3.2 Formal structure of linguistic components

No matter how complicated the user requirements might be for a design problem, they can be ultimately structured into a group of sentences, which assume the above five patterns. On the other hand, as shown in Figure 6 and Equation (3.12), any design requirement can be formulated as a constraint on certain part of product system.

Therefore, if the relationship between those five sentence patterns and the product system can be established, then all design requirements can be logically formalized. It is the objective of this subsection to map the plain English describing the design requirements into the product system described by the formal symbols.

As in Figure 6, there are six objects in a product 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 product system, here gives again 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 product system.

4.3.2.1 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 a constrain 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.

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

$$P_{lv} \subseteq O_1 \otimes_{lv} O_2, \tag{4.1}$$

4.3.2.2 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.

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

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

$$P_{iv} \subseteq O(t_1) \otimes_{iv} O(t_2), \tag{4.2}$$

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

4.3.2.3 Transitive verb

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

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 a transitive verb, P_{tv} for the sentence pattern 3, 4, and 5, we have

$$P_{tv} \subseteq O_1 \otimes_{tv} O_2, \tag{4.3}$$

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

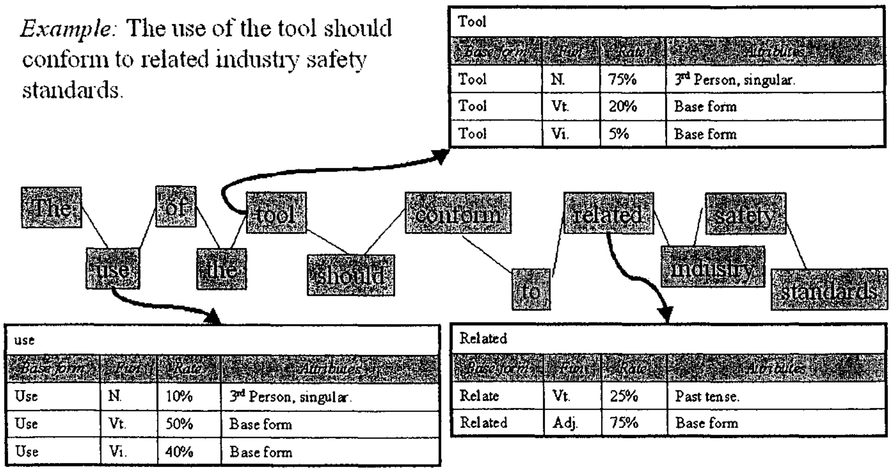
4.4 Formal structure of product requirements

This section will address three major steps in the formalization process of user requirements described in natural language. The three major steps are lexical analysis, syntactical analysis, and structure analysis.

4.4.1 Lexical analysis

Language is a symbolic system that is built up on words. Lexical analysis is the first step in understanding a language. The purpose of this analysis is to identify the lexical property of words forming a design problem. It has four main steps: decomposition of sentence, reduction of word, determination of part of speech, and identification of phrase. The four steps are explained as follows.

At the stage of decomposition, a sentence is decomposed into single words with possible lexical attributes. Figure 12 gives such an example.



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

Tool

Base form	Fun	Rate	Attributes
Tool	N.	75%	3rd Person, singular.
Tool	Vt.	20%	Base form
Tool	Vi.	5%	Base form

use

Base form	Fun	Rate	Attributes
Use	N.	10%	3rd Person, singular.
Use	Vt.	50%	Base form
Use	Vi.	40%	Base form

Related

Base form	Fun	Rate	Attributes
Relate	Vt.	25%	Past tense.
Related	Adj.	75%	Base form

Figure 12 Decomposition of sentence

The inflection of word is reduced to base form. 1) Nouns have two forms, singular and plurality. 2) Verbs have seven forms: base form, infinitive, past simple, past participle, present participle, present simple and third person singular. 3) Adjectives and adverbs have three forms: base form, comparative degree and superlative degree. The lexicon server should be able to reduce them to the base form.

Part of speech of each word is determined. According to English Grammar (Schmidt, 1995), there are eight parts of speech including verb, noun, adjective, adverb, pronoun, preposition, conjunction, and interjection. For verb, noun, and adjective, they can be further classified. To analyze English language more accurately, we have classified the parts of speech of words into 17 types.

Since some words may have multiple attributes, ambiguities exists in deciding part of speech for a word. For example, word "use" can be a noun as well as a verb. However, in most cases, the attribute of a word can be determined by using some simple rules and its context. For example, an article is followed by a noun after ignoring the modifier. This rule can be expressed as a regular expression:

article (adjective phrase | noun phrase)* noun.

Phrases are identified from words by using rules such as:

- If a phrase has the form "noun A of noun B", it is a noun phrase in which the noun A is the head word, the noun B is a modifier.
- If a phrase has the form "helping verb main verb", it is a verb phrase, and the main verb is the head word.

Considering the above two rules, we can find that “the use of the tool” is a noun phrase and “should conform to” is a verb phrase from the example shown in Figure 12.

4.4.2 Syntactic analysis

The objective of the syntactic analysis is to identify the pattern of a sentence that describes a design problem. For the purpose of our study, an English sentence is analyzed based on an English grammar library called Gram-Lib. The English grammar library is composed of a set of basic sentence patterns that describe the relationship between subject, predicate, object of a sentence. The syntactic analysis has two main steps: decomposition of complex sentence and identification of sentence patterns. These two main steps can be explained as follows.

In the step of decomposition of complex sentence, a sentence may be decomposed into some basic sentence patterns. According to the principle “One sentence, one predicate”, a complex sentence is decomposed into a set of phrases and clauses which has and only has one verb. Furthermore, each simple sentence should be contained in the Gram-Lib. If some sentences cannot be analyzed, Gram-Lib should be extended.

Each decomposed sentence is compared against basic patterns until a relevant one is found. Then, the sentence pattern is identified.

Table 3 Example: English sentence pattern

Phrases	Attribute	Function
The **use** of the tool	*Common Noun*	*Subject*
should **conform** to	*Transitive Verb*	*Predicate*
related industry safety **standards**	*Common Noun*	*Object*

Table 3 shows the parts of the sentence example given in Figure 12. It corresponds to the pattern of "Subject + Linking verb + Noun/Pronoun" in Gram-lib.

4.4.3 Structural analysis

Structural analysis is the final step toward the generation of the formal structure for a design problem described by natural language. Since the pattern of a sentence is identified after syntactic analysis, the objective of structure analysis becomes the mapping of syntactic elements into the objects in an ROM model. The structure analysis consists of two steps. The first step is to map each simple sentence to an ROM diagram. The second step is to identify product, environment, performance requirements, and structural requirements from the ROM diagram.

In the context of product systems, a noun names either of product, environment, or the relations between them; a verb names the type of relation involved in a sentence.

Chapter 5

Classification and Categorization of Product Requirements

The designer is rarely able to satisfy all product requirements. It is therefore important to establish the proper choices to satisfy most of the reasonable requirements and find the solutions (Akin, 1986)

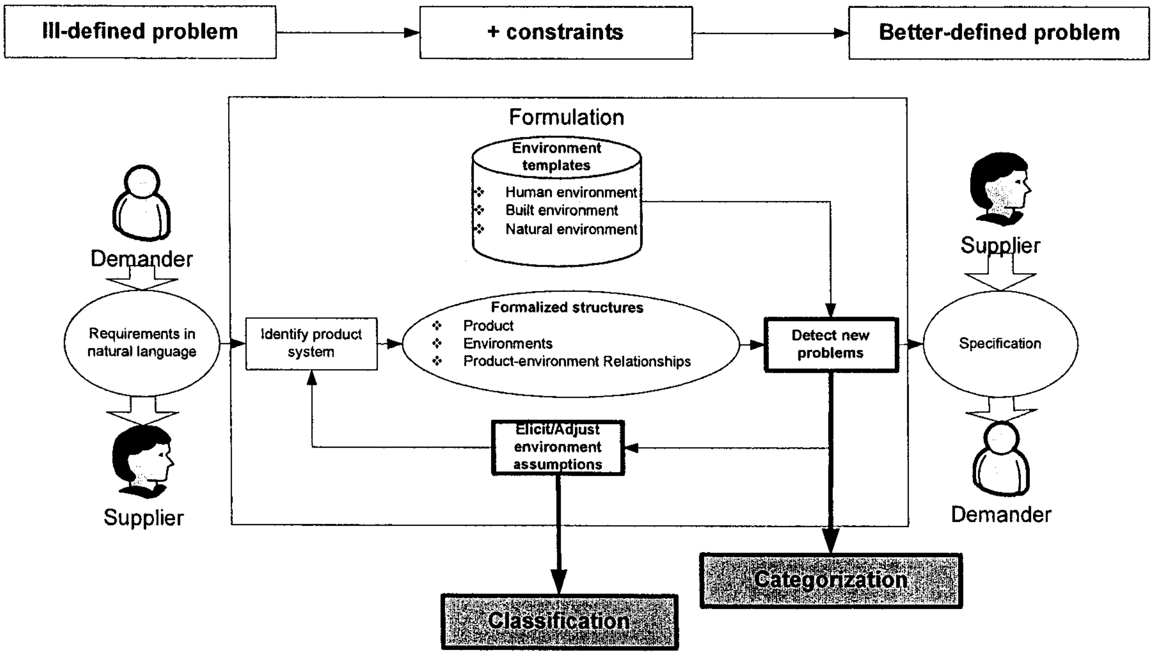


Figure 13 Classification and categorization in formulation

Two tasks in formulating the product requirements are discussed in this chapter as shown in Figure 13:

- Eliciting or adjusting environments assumptions by classifying the events of product life cycle according to the difference of environments related to the events.

- Detecting the new problems that emerge from the contradictions of the existing requirements, and by ordering and categorizing them into eight level constraints.

5.1 Product life cycle

Design is a repetitive process of generating the requirements by the demand side/demander and satisfying it by the supply side/supplier. As shown in Figure 14, the initial requirements usually stem from the demand side, which are the different actors or players in the product life cycle. And these requirements will be satisfied by the supply side.

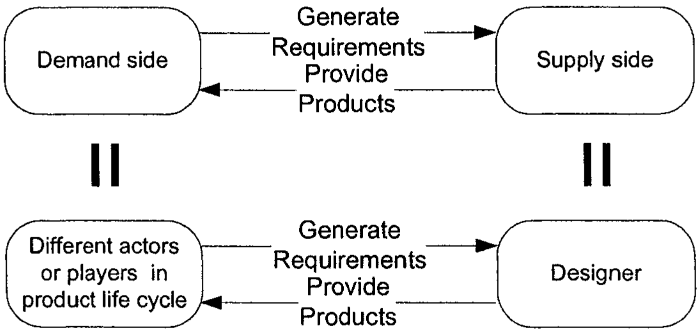


Figure 14 Demand side and supply side of design process

In the previous research, the 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 different demand sides.

Based on our observation, the product life cycle is divided into seven kinds of events in terms of environment's difference, which are design, manufacture, sales, transportation, use, maintenance, and recycle, as shown in Figure 15. These events are building blocks of entire product life cycle. At any time point of product life, one or some of these seven events may occur simultaneously or alternatively. We believe that occurrence of any event is driven by various requirements. These seven events and their relevant actors are discussed in this chapter by using examples from mechanical and software design.

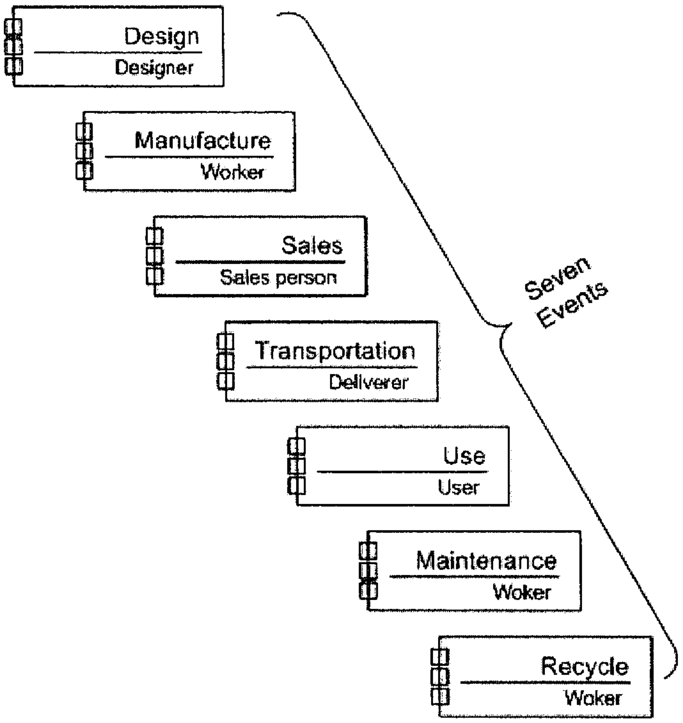


Figure 15 Seven events in product life cycle

Each subsection includes:

- a general description of the event
- a ROM diagram based on a typical example of the event; the ROM symbols used in this chapter have been introduce in Chapter 4.
- a product-environment structure diagram proposed by Zeng (Zeng, 2002)
- a brief explanation of the product-environment system based on the given example. In this part, the environment is decomposed into three types that are natural

environments, built environments, and human environments. Denote natural, built, and human environments by E_n, E_b, and E_h, respectively, we have:

$$E = E_h \cup E_b \cup E_n$$

5.1.1 Design event

Design event, refers to the process from the generation of requirements to the acceptance of a relevant solution. The input of design event is product requirements and design constraints implied in the environment, whereas its output is a satisfying solution. Design event can be triggered by any product requirements that arise from all the players of other events in the product life cycle. The ROM diagram of a typical design event is shown in Figure 16.

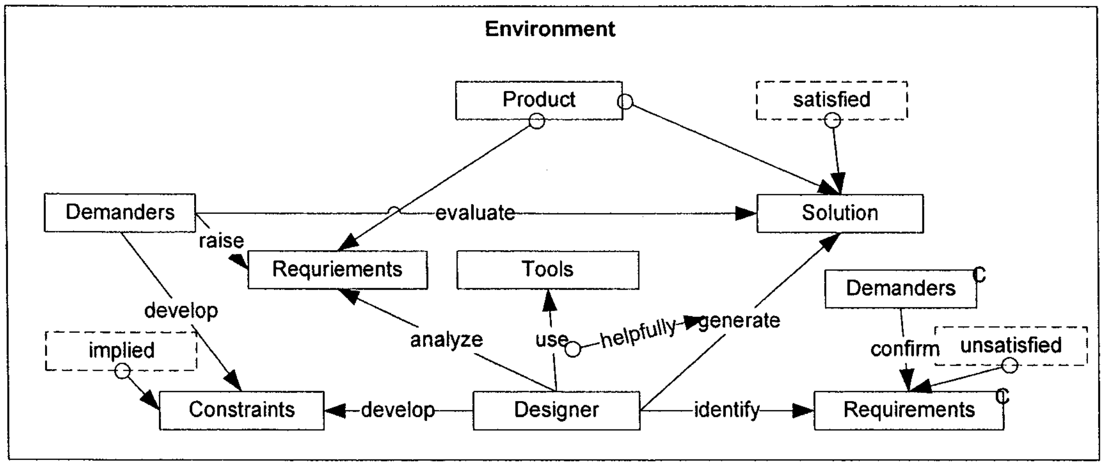


Figure 16 ROM diagram of design Event

In design event, the designer collects, analyzes, and satisfies the product requirements. The product requirements may be raised by the demanders. Cooperating with the demanders, the designer analyzes the product requirements, and develops some implied constraints by decomposing environment with the help of the designer tools, such as ROMA system, Rational Rose, AutoCAD, Solidwork, etc. Eventually, a satisfying product solution is generated accompanied with the identified unsatisfied requirements.

The entire design process can be donated as:

$$P_{Design} \subseteq A \otimes R_{Solution}, \tag{5.1}$$

where, A consists of two parts: requirements and implicit constraints. We have

$$A = A_{Requirements} \cup A_{constraints} \tag{5.2}$$

Moreover, substituting Equation (5.2) into Equation (5.1)

$$P_{Design} \subseteq (A_{Requirements} \cup A_{constraints}) \otimes R_{Solution} \tag{5.3}$$

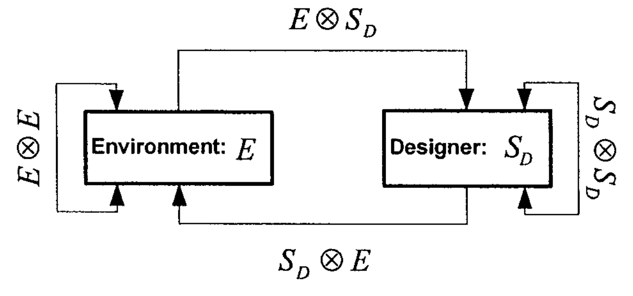


Figure 17 Environment-designer structure

As shown in Figure 17, the relationship between the designer and the environment can be explained as following:

$E = E_h \cup E_b \cup E_n$:

E_h - the demanders of the product requirements who are the players in the entire product life cycle discussed in the following sections

E_b - a design methodology, or a design tools, or a design solution

E_n - the natural environment related to the design event

$E \otimes E$ - (a) the relationship between the material supply and the market; (b) the evaluation to the design solution; (c) the confirmation to unsatisfied requirements by demanders

$S_D \otimes S_D$ - (a) the designer's knowledge; (b) the designer capacity

$E \otimes S_D$ - the generation process of product requirements to the designer.

$S_D \otimes E$ - (a) analyzing the requirements; (b) the development of the implied constraints by the designer; (c) the generation of a satisfied solution; (d) the identification of the unsatisfied requirements.

5.1.2 Manufacture event

Manufacture event, simply put, is the implementation process of a design solution into a real product. The input of the manufacture event is the raw material required in the design solution, and its output is a real product. In this event, the manufacturing workers or operators 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. The design solution may have to be adjusted accordingly.

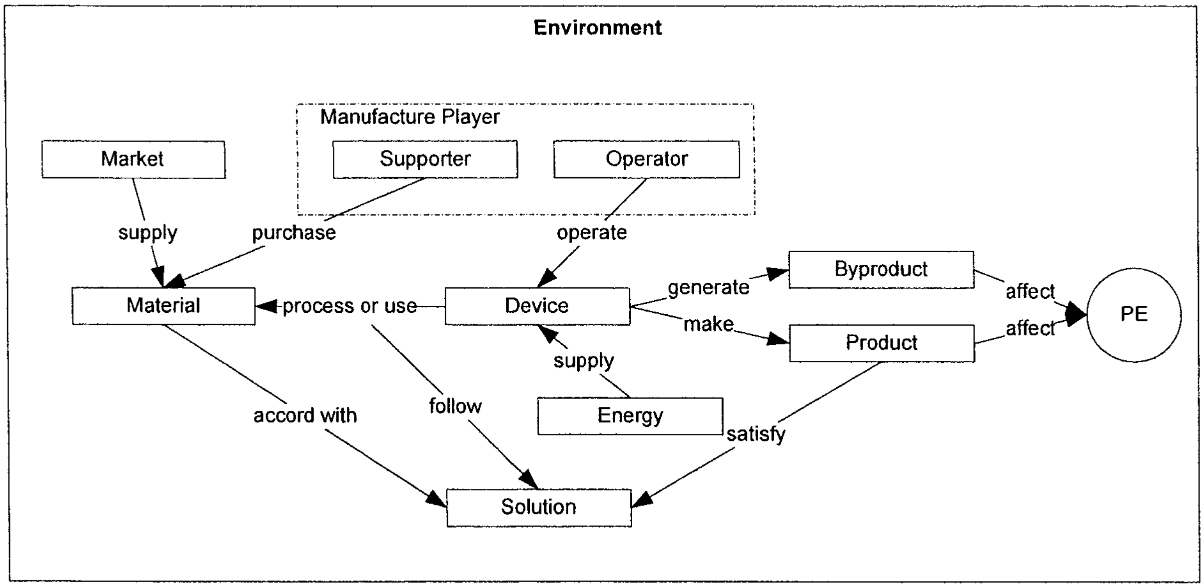


Figure 18 ROM diagram of manufacture Event

A typical manufacture event is illustrated in Figure 18. The workers or operators drive the manufacture device supplied by energy to process the raw material purchased from the market to make a real product, while some byproducts may also be generated. The raw materials should conform to the design solution and the manufacture process should follow the instruction of the design solution.

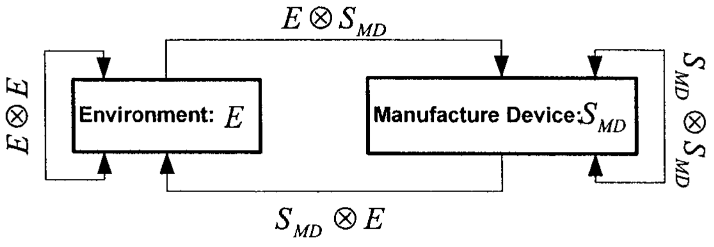


Figure 19 Environment-manufacture device structure

$E = E_h \cup E_b \cup E_n$:

E_h - manufacture players, such as operators or workers and kinds of supporters

E_b - product, raw material, design solution, energy, and byproduct

E_n - natural environment related to the manufacture event, in this case, for example, market

$E \otimes E$ - (a) the material supply from market; (b) the material purchase by supporters; (c) the requirements for the material according to the design solution; (d) the conformance of final product to its design specification; (e) the influence of a product and byproducts to natural environment, for example, environment pollution

$S_{MD} \otimes S_{MD}$ - manufacture capacity of the device. For example, the machining precision of machine tools belongs to this kind of relationship.

$E \otimes S_{MD}$ - (a) the use of the raw materials by the device; (b) the supply of energy; (c) the operation of the device by operators. It belongs to the relationship of environment to manufacture device that represents the input of manufacture device.

$S_{MD} \otimes E$ - The relationship of device to environment represents the manufacture process. And the result of this relationship is a “real product”.

5.1.3 Sale event

A sale event offers available products to clients and customers. In most cases, the sales people or marketing people provide a large number of the 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. The typical sales event can be illustrated in Figure 20.

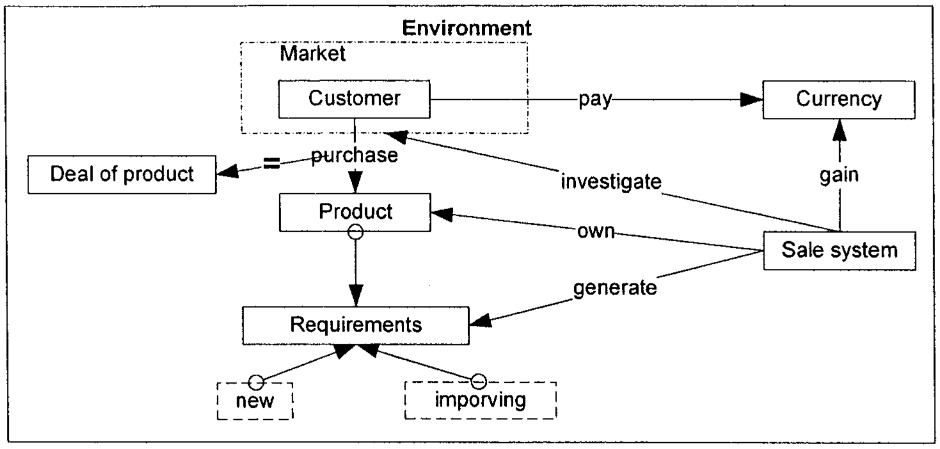


Figure 20 ROM diagram of sale event

The purchase action is a time sequence process as shown in Figure 21.

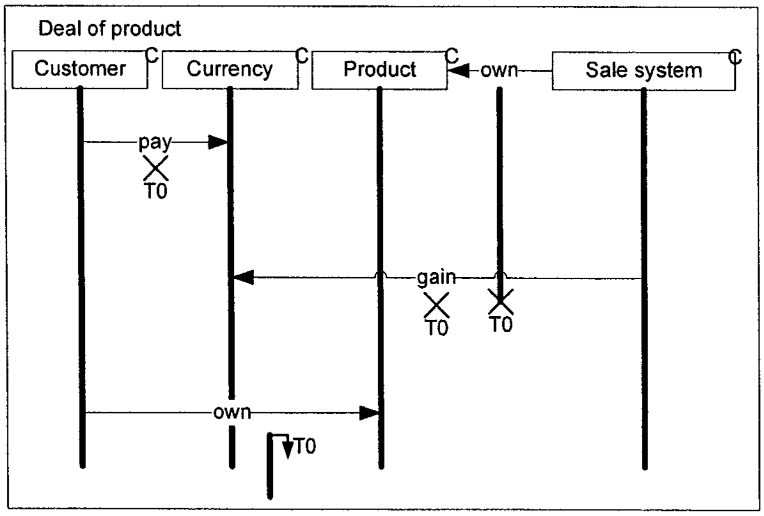


Figure 21 ROM sequence diagram of purchase action

At the beginning of the deal, the sale system has the ownership to the product. After the customer makes the payments, this ownership is transferred to the customer of the product.

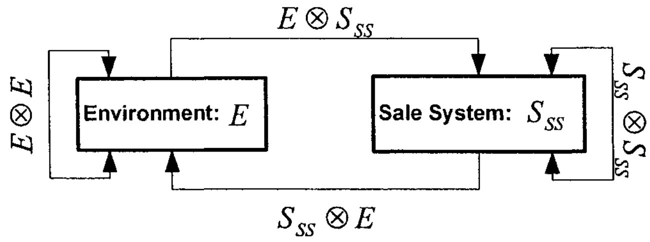


Figure 22 Environment-sale system structure

$E = E_h \cup E_b \cup E_n$:

E_h - (a) customers; (b) other players involving in sale events, such as pre-sale engineers

E_b - (a) product; (b) currency; (c) new or improved product requirements

E_n - the natural environment related to the sale event, such as market.

$S_{ss} \otimes S_{ss}$ - Making a proper market policy is indicated in the relationship on sale system itself.

$E \otimes E$ - (a) decision-making in purchasing a product; (b) payment of customer for the product; (c) getting the ownership of product after a successful deal.

$E \otimes S_{ss}$ - (a) the gain of the sale system from the customer; (b) the change of the product ownership after a successful deal.

$S_{ss} \otimes E$ - (a) the investigation to the market; (b) the generation of the new or updated product requirements.

5.1.4 Transportation event

Almost none of the software providers worry about the transportation issues. By far, the software system of hundreds of mega-bytes can be quickly downloaded through the high-speed Internet or published by CDs. Before 1994, 3.5" floppy disks were the most popular media between computers. The entire installation of Windows 3.X had to be compressed into less than twenty floppy disks. Similarly, the civil engineering and mechanical engineering product may be confronted with the 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 the design solution in terms of the transportation vehicles and paths though which they will be carried to the destination. A typical transportation event is illustrated in Figure 23.

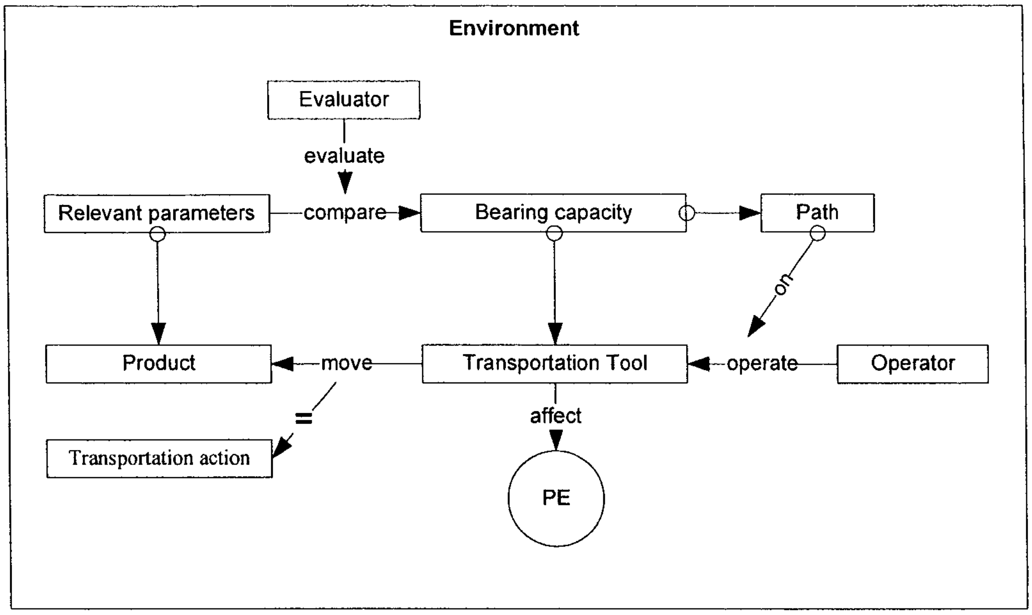


Figure 23 ROM diagram of transportation event

A transportation plan is made by observing and evaluating the following facts:

- Relevant parameters of payload, such as, chemical characteristic of dangerous goods, dimension, etc;
- The loading capacity of a transportation tool, for instance, its allowable bearing capacity;
- The bearing capacity of passed path, for instance, the bridge's maximum loading, maximum width of the road, the clearance height of culverts or overpass bridges, the throughput of a computer network, etc.

A transportation action can be described as shown in Figure 24.

At source place A, the product is loaded onto a transportation tool. The transportation tool moves the product by the planned path. The product is unloaded from the transportation tool when they arrive at the destination place B. After the transportation action, the environment of this system is changed from place A to place B.

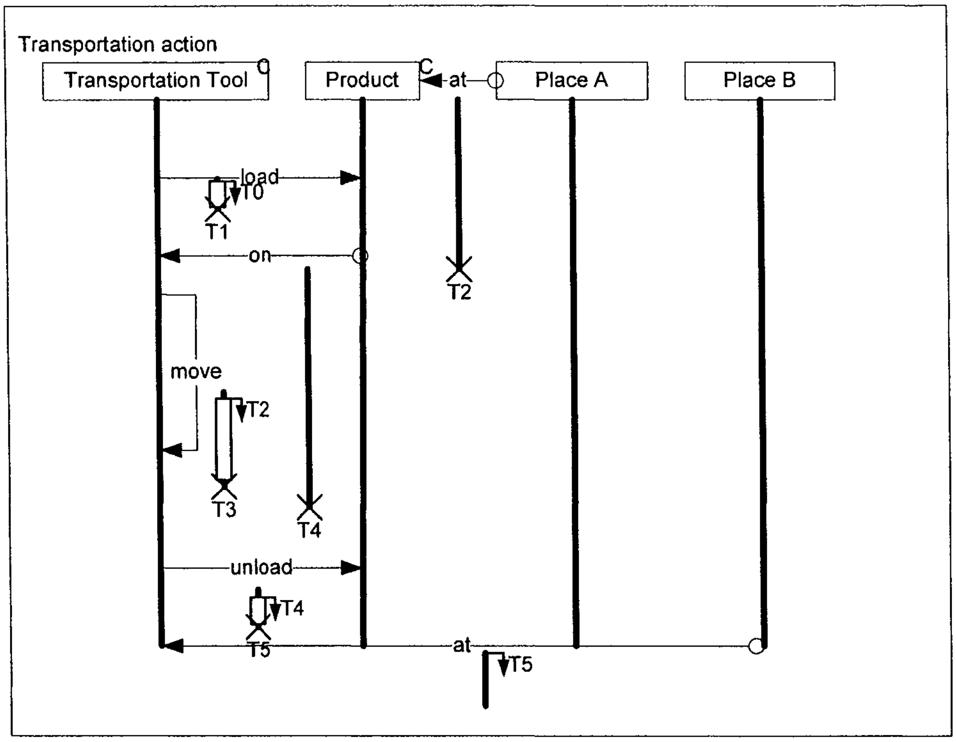


Figure 24 ROM sequence diagram of transportation action

The detailed description of Figure 24 can be found in section 4.1.8.

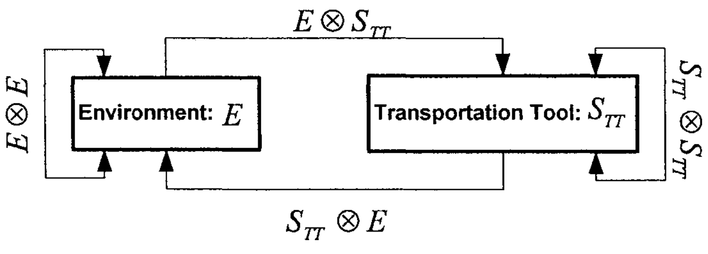


Figure 25 Environment-transportation tool structure

$E = E_h \cup E_b \cup E_n$:

E_h - (a) evaluator who is responsible to evaluate the transportation plan; (b) operator driving the transportation tools

E_b - (a) the product; (b) the passed path

E_n - (a) the source place A; (b) the destination place B

$S_{TT} \otimes S_{TT}$ - (a) the maximum loading capacity of a transportation tool; (b) the movement of the transportation tool carrying the product

$E \otimes E$ - (a) relevant to the parameters the product's transportation; (b) the bearing capacity of passed path; (c) the comparison of product parameters; (d) the loading capacity of the path; (e) the evaluation to the comparison result by evaluator

$(E \otimes S_{TT}) \cup (S_{TT} \otimes E)$ - (a) the process of loading product to transportation tools at source place A; (b) the process of unloading product from transportation tools at destination place B

5.1.5 Use event

Use event refers to the process of the product being used by the end user. The use event is paid more attention by designers and end users than other events, because it occurs most frequently during the product life cycle. Its environment is also the most complicated from product to product. Therefore, only the common circumstance is discussed in the present thesis.

What the design and manufacture event create is a product's physical structure. When the product is put into its use event environment, the ideal situation is that the desired inputs cause the desired performances (output), and these performances can be used by the end user.

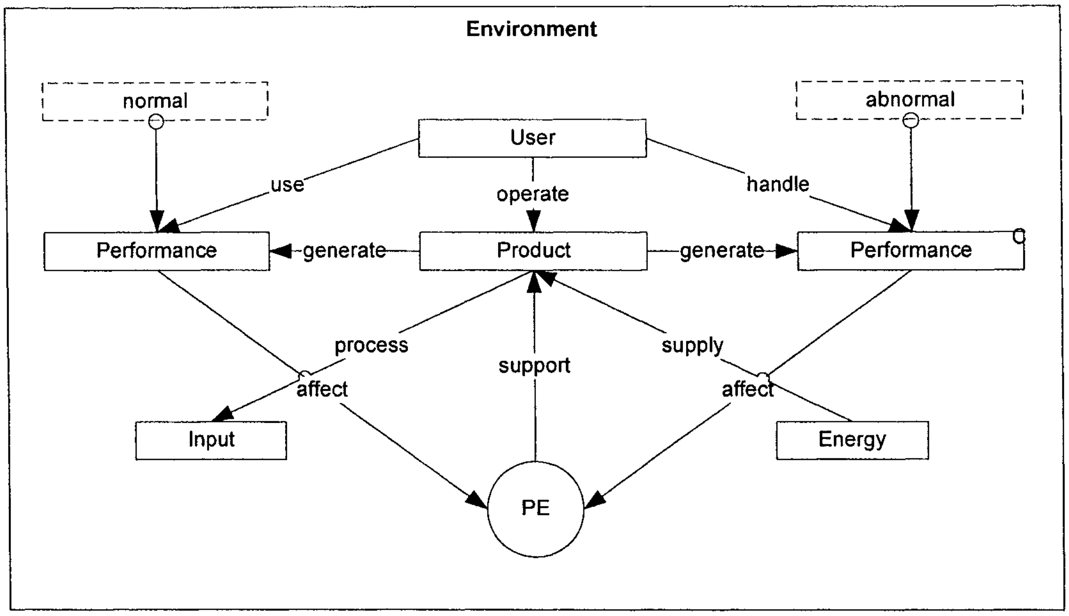


Figure 26 ROM diagram of use event

Figure 26 tells us that a user operates a product to process an input with the energy supply, generate normal performances in desired cases, or the abnormal performances due to an exceptional input or the use event environments.

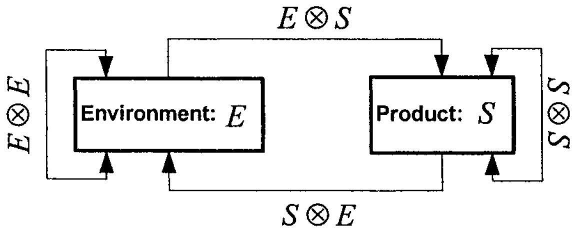


Figure 27 Environment-product structure

$E = E_h \cup E_b \cup E_n$:

E_h - Most commonly, (a) end user of product; (b) if the product is used to provide a service, then it is possible that both service provider and service receivers will generate requirements.

E_b - (a) input; (b) energy supply, in some simple mechanical tools energy supply is same as its input. For instance, a screw drive, and a rivet tool in the case study subject of the present thesis.

E_n - relevant natural environment, including temperature, place, and so on.

$S \otimes S$ - (a) the product's physical structure; (b) the desired performance of the product

$E \otimes S$ - (a) the energy supply; (b) the input of the product system

$S \otimes E$ - the generation of performance

5.1.6 Maintenance event

A maintenance event can be categorized into four different types in terms of different purposes: routine maintenance, corrective maintenance, perfective maintenance, 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 (Schach, 2004) happens when the product has problems against the product requirement. Examples include the service package or an upgrade patch package from software providers. The recalls by automobile manufacturers also fall into this category.

Perfective maintenance (Schach, 2004) 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* (Schach, 2004), 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 other hardware platforms.

A typical maintenance event is shown in Figure 28.

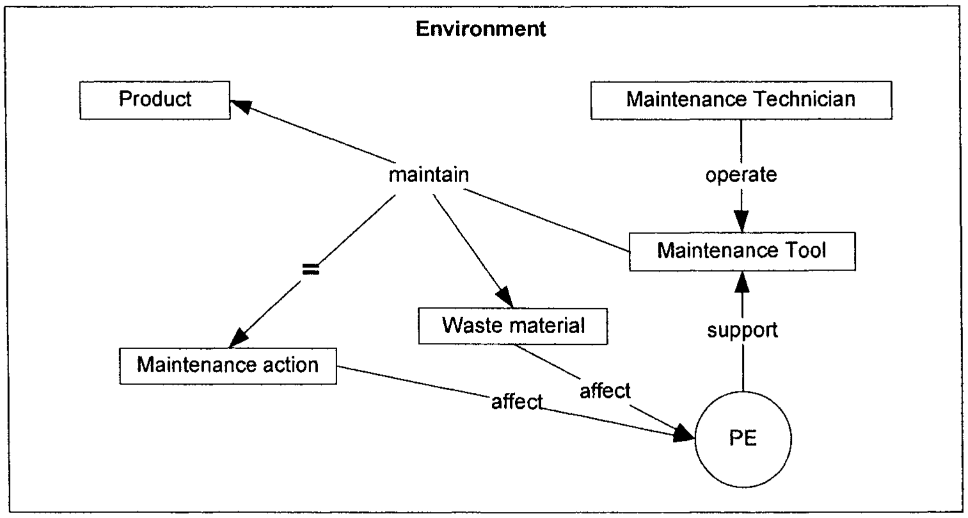


Figure 28 ROM diagram of maintenance event

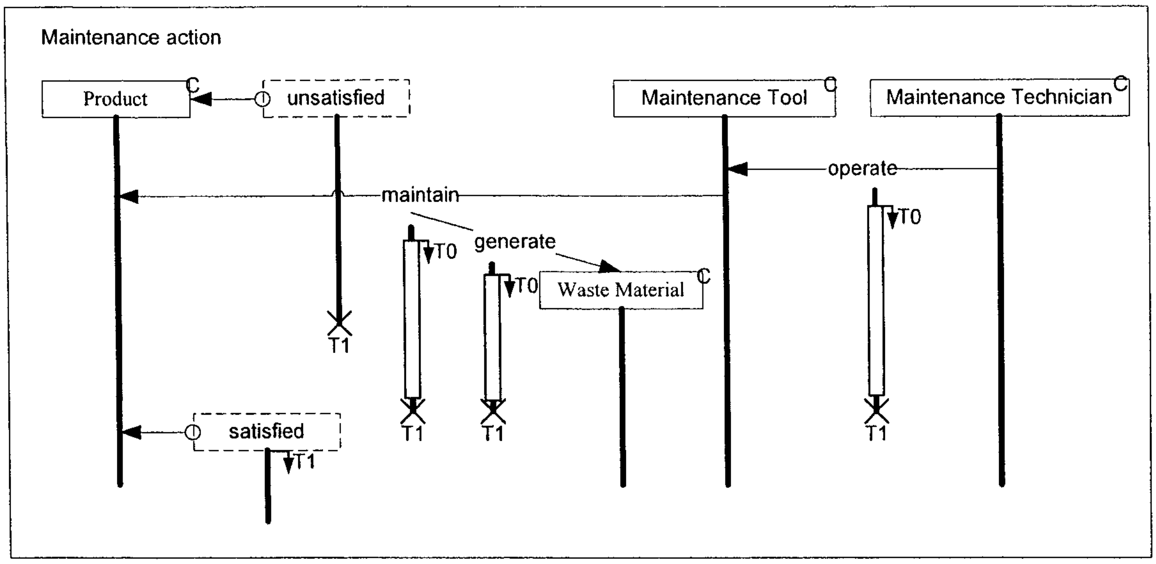


Figure 29 ROM sequence diagram of maintenance action

As a general rule, the maintenance event adjust the performance from a unsatisfied or a predictable unsatisfied status (routine maintenance) to a more satisfied status, meanwhile some maintenance processes may generate waste materials that affect the environment furthermore.

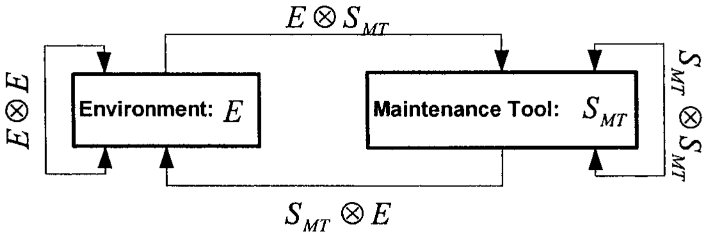


Figure 30 Environment-maintenance tool structure

$E = E_h \cup E_b \cup E_n$:

E_h - the maintenance technician

E_b - (a) the product; (b) other supports from environment

E_n - relevant natural environments

$S_{MT} \otimes S_{MT}$ - the maintenance tool's working capacity, for example, power, precision of the tool.

$E \otimes E$ - (a) the professional skill of the technician; (b) the physical structure of the product

$E \otimes S_{MT}$ - (a) the technician's operation to the maintenance tool; (b) the power/energy supply. For example, if the maintenance event happens on a wild field, the power supply may come from a car's engine, sometime even be human power.

$S_{MT} \otimes E$ - (a) the maintenance process on the product; (b) the generation of the waste materials.

5.1.7 Recycle event

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

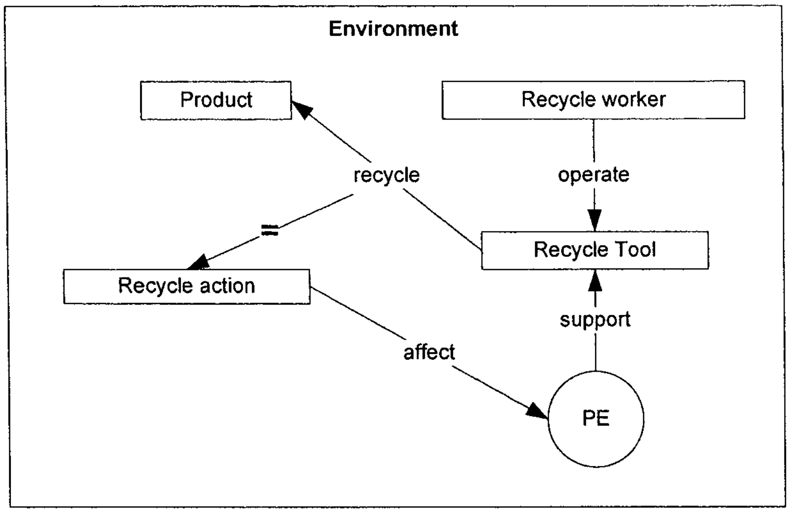


Figure 31 ROM diagram of recycle event

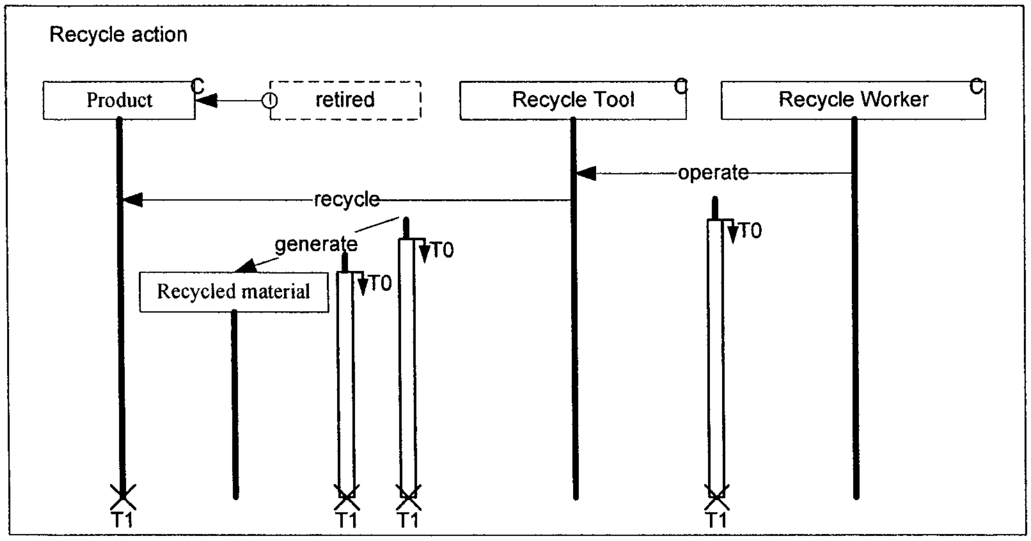


Figure 32 ROM sequence diagram of recycle action

A recycle action is that a recycle worker operates tool to remove the retired product from its installed environment. This process may generate some recycled materials which may pollute natural environment, like hydro resource, air, etc. Using of recyclable material in design event will be helpful to reduce increasing deterioration of the environment.

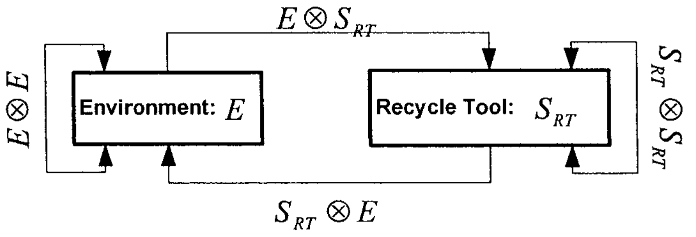


Figure 33 Environment-recycle tool structure

$E = E_h \cup E_b \cup E_n$:

E_h - recycle workers

E_b - (a) retired product; (b) other support from environment

E_n - relevant natural environment

$S_{RT} \otimes S_{RT}$ - recycle tools' process capacity

$E \otimes S_{RT}$ - (a) workers' operation to recycle tools; (a) the power/energy supply

$S_{MT} \otimes E$: (a) the recycle process on the retired product; (b) the generation of recycle material

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

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

5.2 Level of requirements

As can be seen from last section, 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 34, the product requirements are categorized into eight levels: natural laws; social law 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.

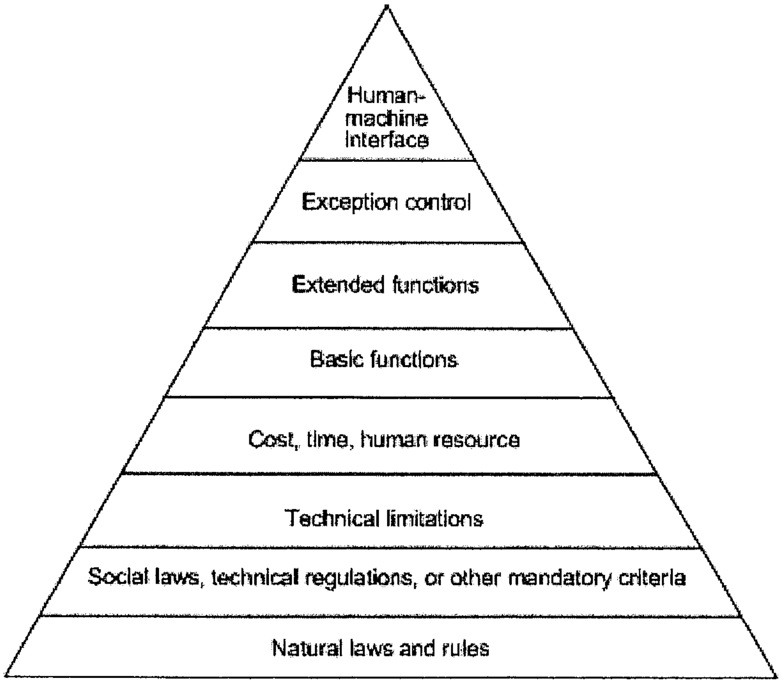


Figure 34 Eight levels of requirements

All products must be built based on 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 designer has to make sure that the budget, schedule and human resource demand are 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 focus on realization of the basic functions. After all basic functions were achieved, designers can start to consider the 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, the 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 as the adjustment of capital, schedule and human resource 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 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, will we put different requirements 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 non-functional requirements. The upper four: basic functions, extended functions, exception control, human-machine interface are usually functional requirements. The following will discuss these eight levels, respectively.

5.2.1 Natural laws and rules

All products are parts of the nature from which they can never be separated (Zeng and Cheng, 1991). Any product is not able to escape from natural law; that is why the perpetual motion machine can never be made true.

5.2.2 Social laws and technical regulations

When a design solution is being developed, if there are 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 in 120 V power supply, though in 220~240 V for some other countries. Electronic products sold to these regions should be designed to suit this regulation.

Requirements belonging to this level must be satisfied. In other words, the product has to be redesigned or discarded when its performances conflict with this level.

5.2.3 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 available technologies, adjustments in the design solutions have thus 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 seems to be the only choice, if end users were eager to use icon-based operation or start application by clicking mouse.

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

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

5.2.4 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 (Adolph, 1996). 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.

5.2.5 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.

5.2.6 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.

5.2.7 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 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 entire database from zero. A few data may be lost, rather than have nothing. Multiple-engine airplane is another example.

5.2.8 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 (Consortium).

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 form human environment. They serve for 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.

Chapter 6

Software Prototype for Recursive Object Modeling Analysis: ROMA

The quality of design can be improved by a better understanding of the requirements definition process. In Chapter 4, a formalized graphic language, ROM, is developed to represent the formalized structure, and the foundation of natural language process is also discussed. In Chapter 5 the classification and categorization methods are introduced to elicit/adjust environment assumptions, and detect new design problems. As shown in Figure 35, a software prototype system ROMA is presented in this chapter as a tool to help designer to build a formalized product system represented by the ROM language from a design problem described in natural language.

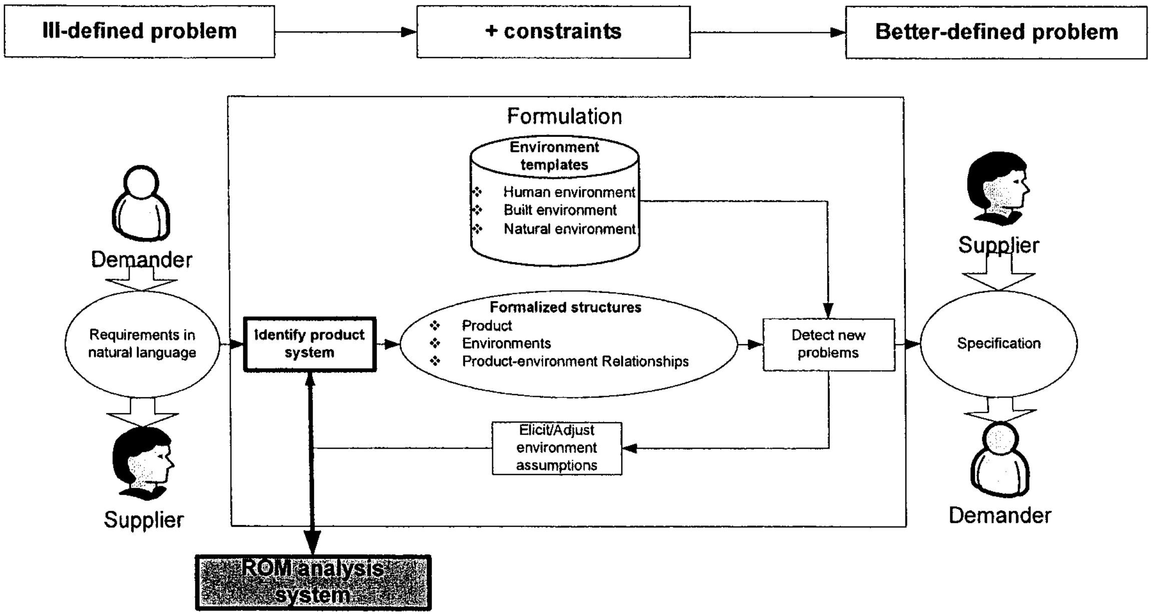


Figure 35 ROM analysis system in formulation

6.1 Problem formulation

The central task in solving a design problem is to specify product requirements. An ROM diagram is used to present the elements of requirements and the relationships between them.

We have been developing a software system which can automatically translate product requirements described by natural language into an ROM diagram. The input of this software system is product requirements described in English and the corresponding ROM diagram is the output. In this chapter, we will present the architecture of prototype system and some results from formalizing product requirements.

6.2 System architecture

We have implemented a software system named ROMA (ROM analysis system). This system contains four modules: ROM Client, Lexicon Server, Syntax Server, and Knowledge-Base Server. The architecture of this system is shown in Figure 36.

ROM Client is the user interface of whole system. It does not involve core logical operations, whereas it invokes these operations from servers. The two main tasks of ROM Client are: to display the ROM diagram which had been analyzed by other modules, and to provide an interactive terminal that enable end users to participate in decision-making in the case of emerging ambiguity of various solutions. To a group of users, they are able to run ROM Client application individually to process respective requirement documents, and build the ROM diagram which belongs to them.

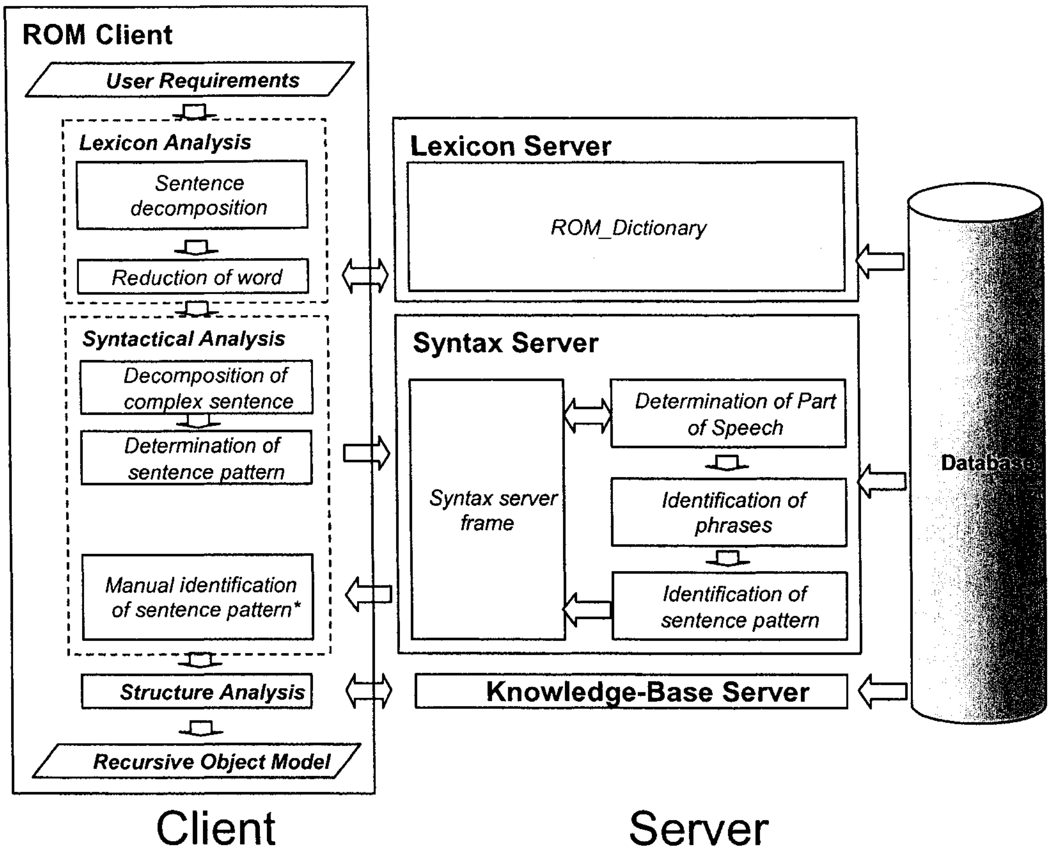


Figure 36 Architecture of ROMA

Lexicon Server is responsible for the reduction of single word, returning search results from ROM_Dictionary to ROM Client. ROM_Dictionary is the kernel of Lexicon Server, which includes 67181 base-form words, and 15141 infected form words at present. To improve query performance, over 80 thousand words are saved in a two dimension hash table. Meanwhile, for the purpose of running on multiple platforms, the Lex-Server is divided into three layers: core layer, interface layer and application layer (Figure 37). All core operations are placed in the core layer, such as: loading the dictionary data from file or database, searching word from the dictionary, reducing the inflected word to base form, etc. An example of search results is shown in Figure 42.

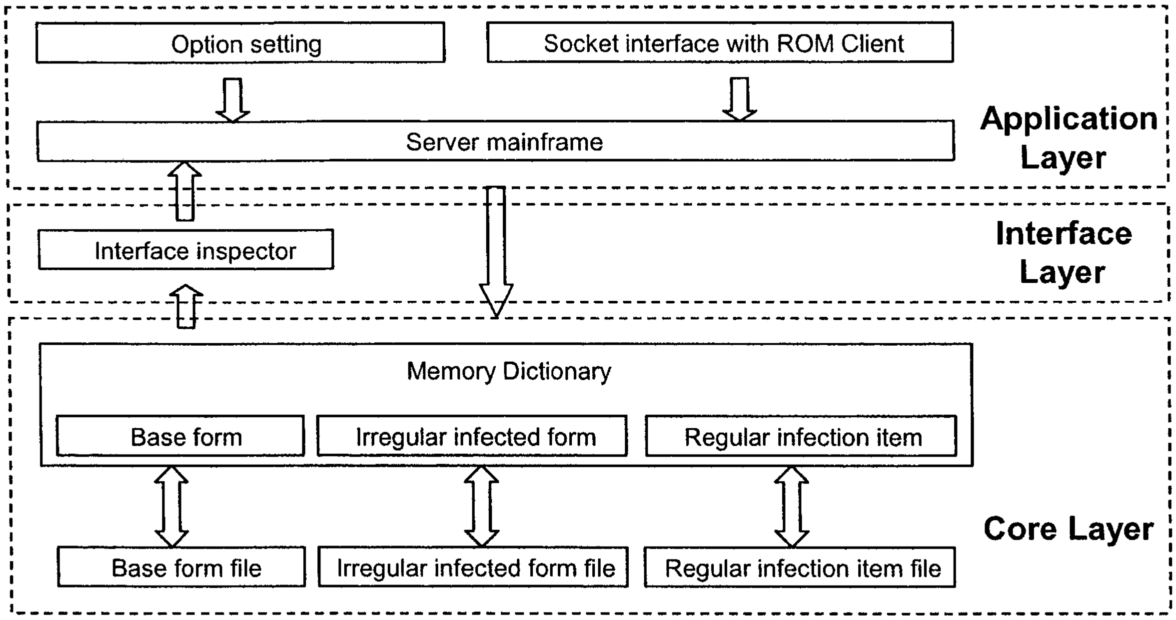


Figure 37 Structure of lexicon server

Syntax Server is responsible for the analysis of sentence structure, mapping the sentence structure into ROM structure, and making them into XML data package, then sending the package to ROM Client. In this module, we adjust the technology of nested invoking between C++ and Prolog Figure 38. This kind of bidirectional invoking can benefit from each other. C++ has higher - flexibility, performance and friendly user interface. On the other hand, Prolog has powerful capacity in logical deduction. The core of Syntax Server consists of three infrastructures: determination of part of speech, identification of phrase, and identification of sentence pattern as is shown in Figure 36. All of them are implemented with Prolog. The results of syntax analysis are complex. To organize data structure efficiently, XML technology is introduced as data exchange formation between Syntax server and ROM Client. The content of XML data package is shown in Figure 43.

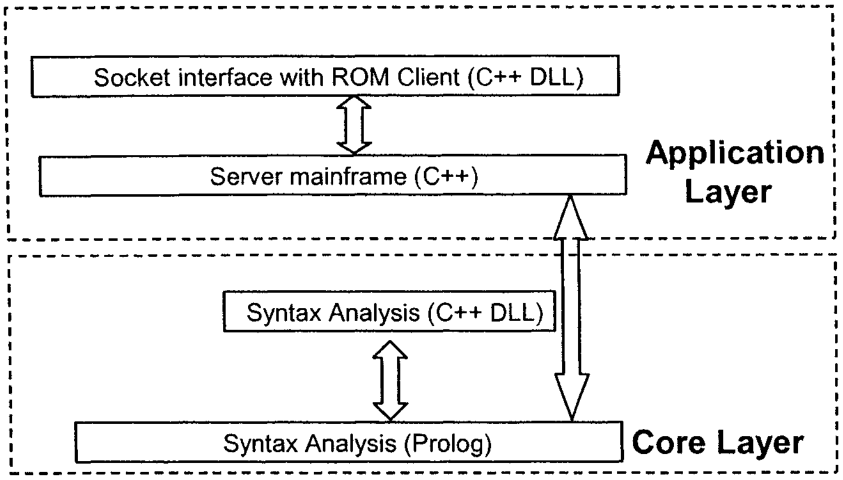


Figure 38 Structure of syntax server

The objective of Knowledge-Base Server is to process the sentence for which the pure syntactical analysis does not work. This part is still under development.

Since the foundation of this software system is the axiomatic theory of design modeling, a data structure is developed to describe the objects in this theory, as is shown in Figure 39.

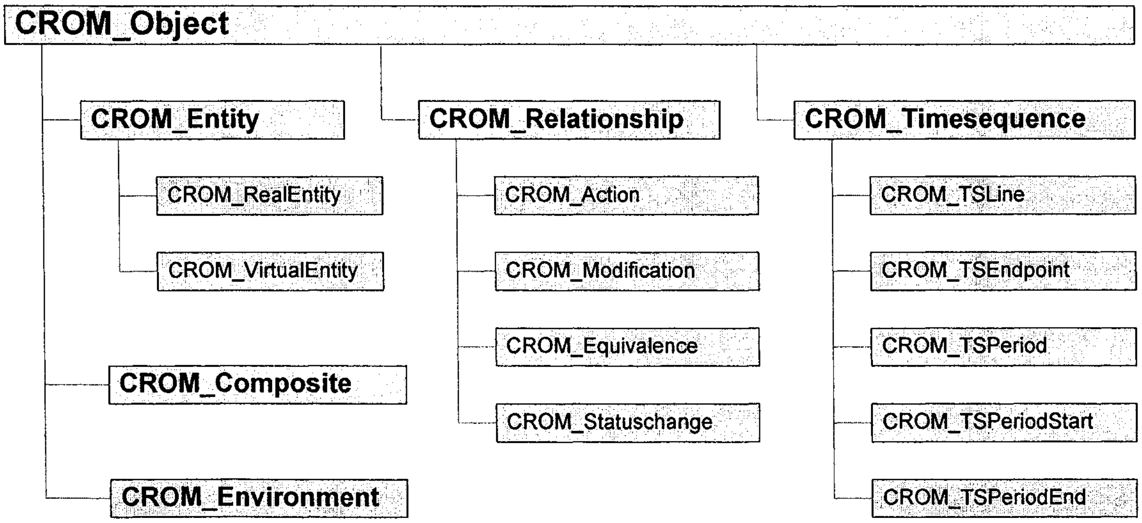


Figure 39 Hierarchichal structure of ROM objects

This hierarchical architecture describes essential elements of the axiomatic theory of design modeling (Zeng, 2002), such as the object and the relationship between objects in ROMA. The CROM_Object is an abstract class of all objects in the axiomatic theory of design modeling (Zeng, 2002). It has five subclasses: CROM_Entity, CROM_Relationship, CROM_Composite, CROM_Timesequence, and CROM_Enviroment.

- CROM_Entity has two subclasses: (a) the CROM_RealEntity defined in section 4.1.1 comes from noun in English Grammar; (b) the CROM_VirtualEntity defined in section 4.1.2 comes from an adjective, adverb etc.
- CROM_Relationship represents all kinds of relationship between the instance objects of CROM_Entity or CROM_Relationship. In this structure, relationship is categorized into actions (defined in 4.1.3), modifications (defined in 4.1.4), equivalences (defined in 4.1.5), and status-changes (defined in 4.1.6). Generally speaking, actions come from verbs, and modifications come from sorts of relationship between modifiers and modified objects.
- CROM_Composite is defined in section 4.1.7, which can be decomposed into CROM_Entity and CROM_Relationship at a lower level of granularity.
- CROM_Timesequence has five subclasses that are defined in section 4.1.8. They are (a) the CROM_TSLine representing a lifetime line; (b) the CROM_TSEndpoint representing a lifetime endpoint; (c) the CROM_TSPeriod representing a lifetime period; (d) the CROM_TSPeriodStart representing the start

point of a lifetime period; (e) the CROM_TSPeriodEnd representing the end point of a lifetime period.

- CROM_Enviroment (defined in section 4.1.9) usually is the default object of a product system.

Chapter 7

Case Study

A rivet setting tool design example is adapted 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 Figure 40.

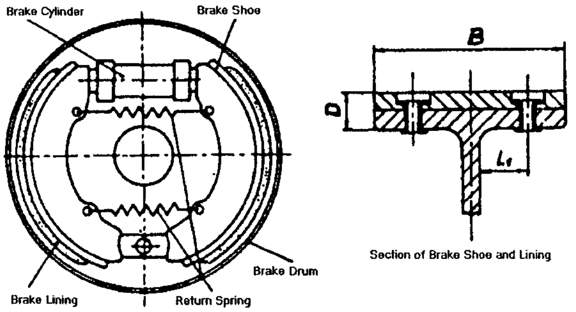


Figure 40 Internal drum brake (Hubka et al., 1988).

The additional information regarding this design problem includes: 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. The tool will be manufactured in a specific workshop, which has specified equipment. The cost of this tool cannot be over $190.00.

According to theorem of design problem structure, rivet setting tool design problem is given in Table 4.

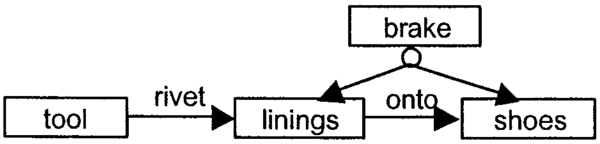
Table 4 Design problem: rivet setting tool design.

Rivet Setting Tool Design	
Product Environment	R-E1. Nature. R-E2. Mechanics. R-E3. Manufacturing shop. R-E4. Transportation facilities. R-E5. Brake shoe and lining.
Boundary Requirements	R-R1. To rivet brake linings onto brake shoes. R-R2. The hand and foot forces should follow ergonomic standards. R-R3. The use of tool should conform to related industry safety standards. R-R4. The tool can be manufactured in the specific workshop. R-R5. The service life of tool will be around 5 years. R-R6. The tool should be easy for transportation and maintenance. R-R7. The cost of tool cannot be over $190.00
Structural Requirements	R-R8. The working height should follow ergonomic standards

7.1 Requirements classification

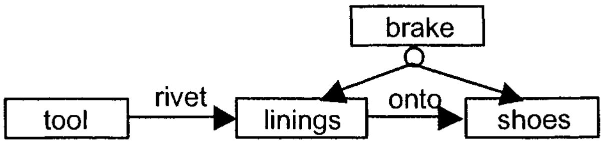
This is a simple requirements description. The entire requirement document has eight sentences, but the content is abundant. In this section, each sentence is mapped into a ROM diagram; classified into its corresponding events; categorized into relevant requirements levels.

The tool rivets brake linings onto brake shoes.



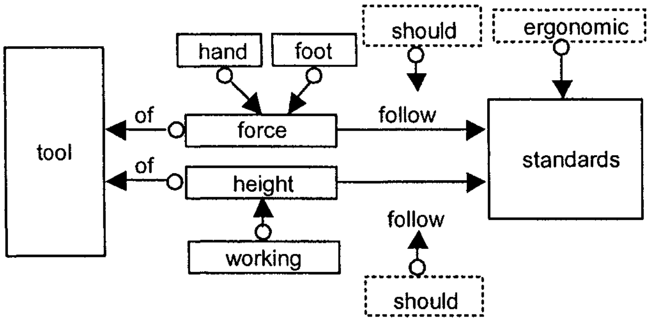
- *belongs to [Use Event]. It describes the basic function of the target product (Basic functions level).*

The user of the tool is a car mechanic.



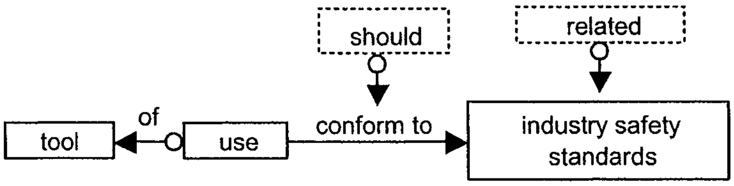
- *belongs to [Use Event]. It defines that end users are technicians – car mechanics who may have some professional skills in using this tool. (Human-machine interface level).*

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



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

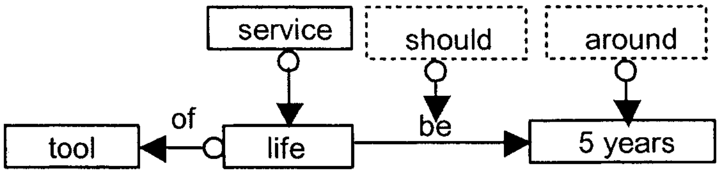
The use of this tool should conform to the related industry safety standards.



- *belongs to [Use Event]. The related industry safety standards can be regarded as a kind of mandatory criterion, sometimes associated with*

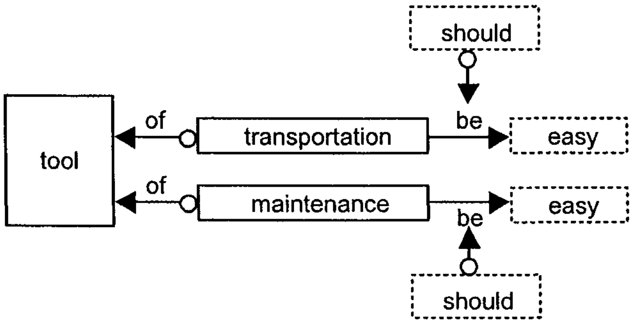
Labor Law. (Law Regulations level).

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



- *belongs to [Use Event]. If five-year is common service life of congener product, this requirement can be classified into (Basic functions level). Otherwise, it may be regarded as (Extended functions level).*

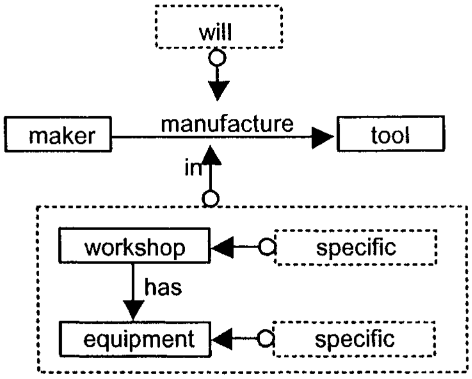
The tool should be easy for transportation and maintenance.



- *belongs to [Transportation Event, Maintenance Event]. These two requirements exceed basic functions level, basically, they are (Extended functions level). If the absence of these two functions will influence marketing performance, they may be lowered into (Basic functions level}.*

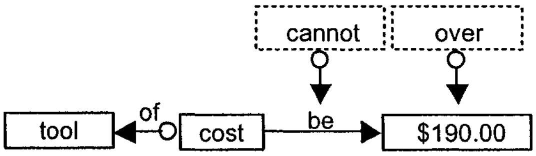
As mentioned in section 5.2, the distribution of higher-level requirements relies on the lower-level requirements. In this case, to achieve the goal of profit, the division of basic functions and extended functions may vary.

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



- *belongs to [Manufacture Event]. The manufacture of the tool cannot be performed in a general workshop, because it needs specified equipments. (Technical limitations level).*

The cost of this tool cannot be over $190.00.



- *belongs to [Sale 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 Table 5 and Table 6, respectively.

Table 5 Event property of example

Manufacture Event

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

Sale 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.*

Table 6 Requirements level of example

Law 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.*

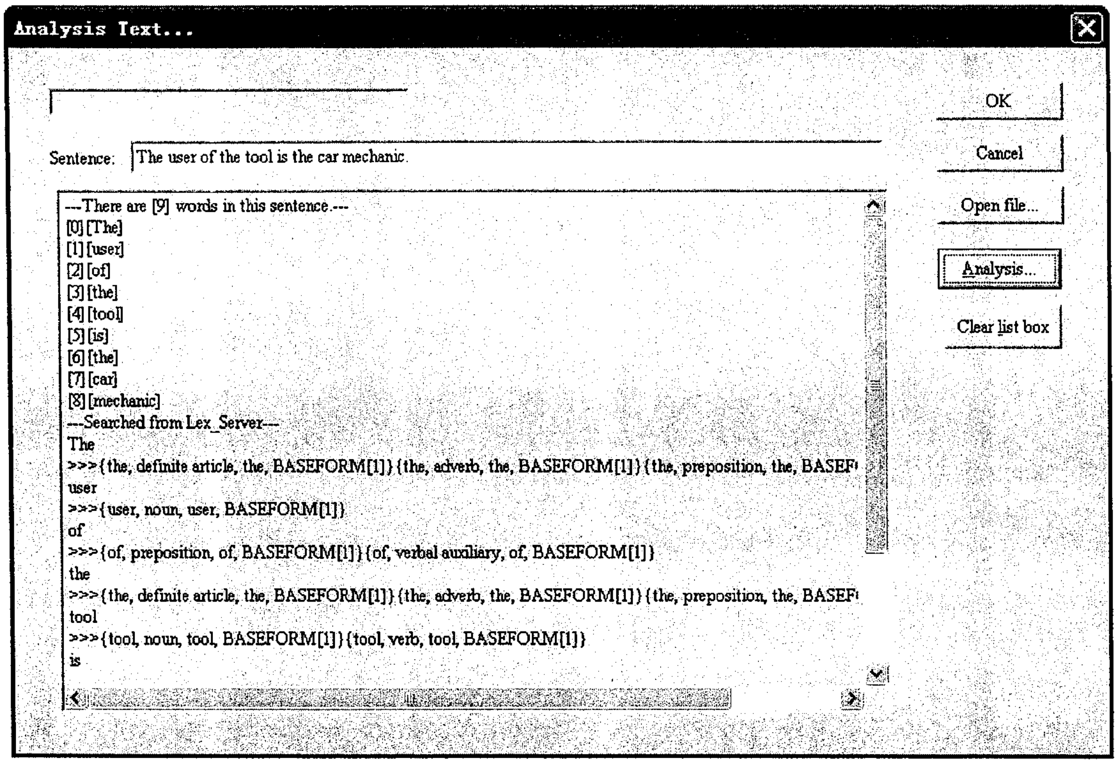
Human-machine interface level

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

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 5, the detailed requirements can be specified by interviewing relevant demander, and separated specification documentation is generated for confirming.

7.2 Experimental results

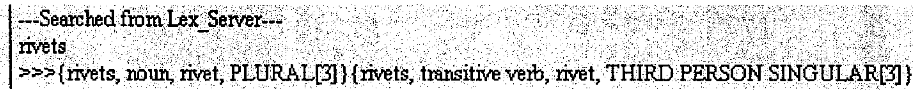
Firstly, a user requirement document is separated into sentences and each sentence is decomposed into a series of single words. Each single word is reduced in Lexicon Server, and sent back to ROM Client. When we send a single sentence as input, Figure 41 shows the process of decomposing sentence and reducing single word. Figure 42 shows the result from Lexicon Server.



Sentence: The user of the tool is the car mechanic.

```
---There are [9] words in this sentence.---
[0] [The]
[1] [user]
[2] [of]
[3] [the]
[4] [tool]
[5] [is]
[6] [the]
[7] [car]
[8] [mechanic]
---Searched from Lex_Server---
The
>>>{the, definite article, the, BASEFORM[1]}{the, adverb, the, BASEFORM[1]}{the, preposition, the, BASEF
user
>>>{user, noun, user, BASEFORM[1]}
of
>>>{of, preposition, of, BASEFORM[1]}{of, verbal auxiliary, of, BASEFORM[1]}
the
>>>{the, definite article, the, BASEFORM[1]}{the, adverb, the, BASEFORM[1]}{the, preposition, the, BASEF
tool
>>>{tool, noun, tool, BASEFORM[1]}{tool, verb, tool, BASEFORM[1]}
is
```

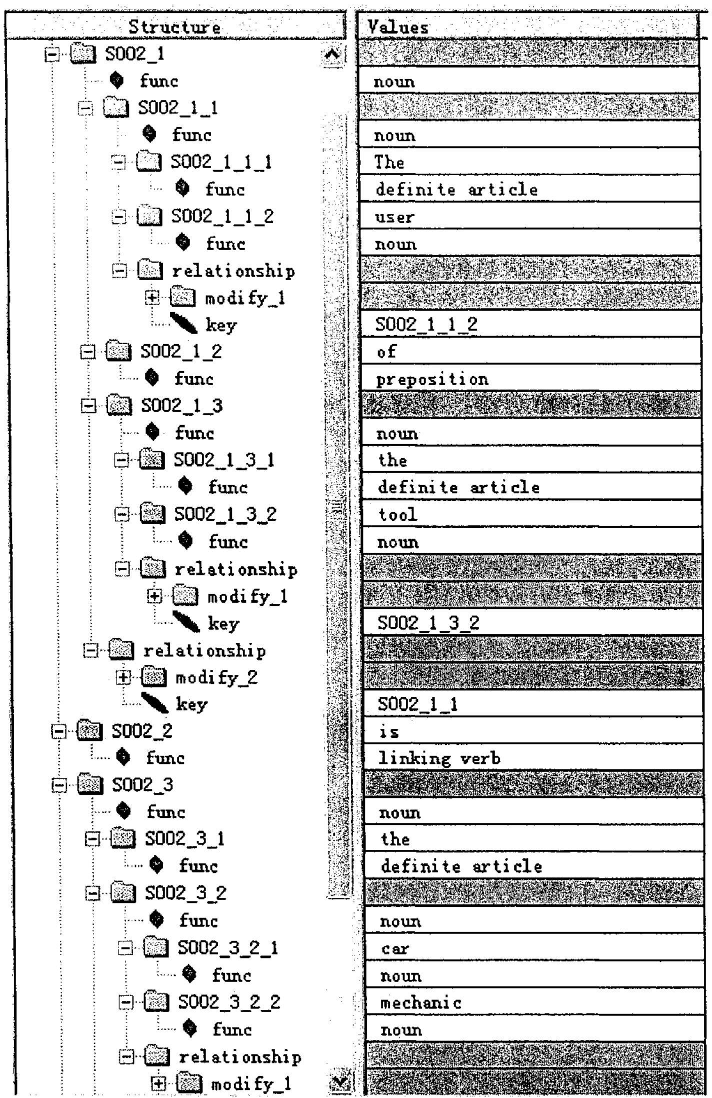
Figure 41 Analysis result of single sentence



```
---Searched from Lex_Server---
rivets
>>>{rivets, noun, rivet, PLURAL[3]}{rivets, transitive verb, rivet, THIRD PERSON SINGULAR[3]}
```

Figure 42 Result of lexicon analysis

ROM Client sends Syntax Server a data package, which has only one sentence including the result of lexicon analysis. Each sentence is analyzed in Syntax Server, and the outcome of this step is sentence structure, which is organized into XML package shown in Figure 43.



```
Structure                         Values
S002_1
    func                          noun
    S002_1_1
        func                      noun
        S002_1_1_1                The
            func                  definite article
        S002_1_1_2                user
            func                  noun
        relationship
            modify_1
            key                   S002_1_1_2
    S002_1_2                      of
        func                      preposition
    S002_1_3
        func                      noun
        S002_1_3_1                the
            func                  definite article
        S002_1_3_2                tool
            func                  noun
        relationship
            modify_1
            key                   S002_1_3_2
    relationship
        modify_2
        key                       S002_1_1
S002_2                            is
    func                          linking verb
S002_3
    func                          noun
    S002_3_1                      the
        func                      definite article
    S002_3_2
        func                      noun
        S002_3_2_1                car
            func                  noun
        S002_3_2_2                mechanic
            func                  noun
        relationship
            modify_1
```

Figure 43 Result of syntax analysis

Based on the outcome of the syntax analysis, the ROM Client generates the ROM diagram in Figure 44. In this version, we have not been able to analyze passive sentences, and this part will be the future job. Figure 44 lists the result from formulizing seven requirements for Table 4.

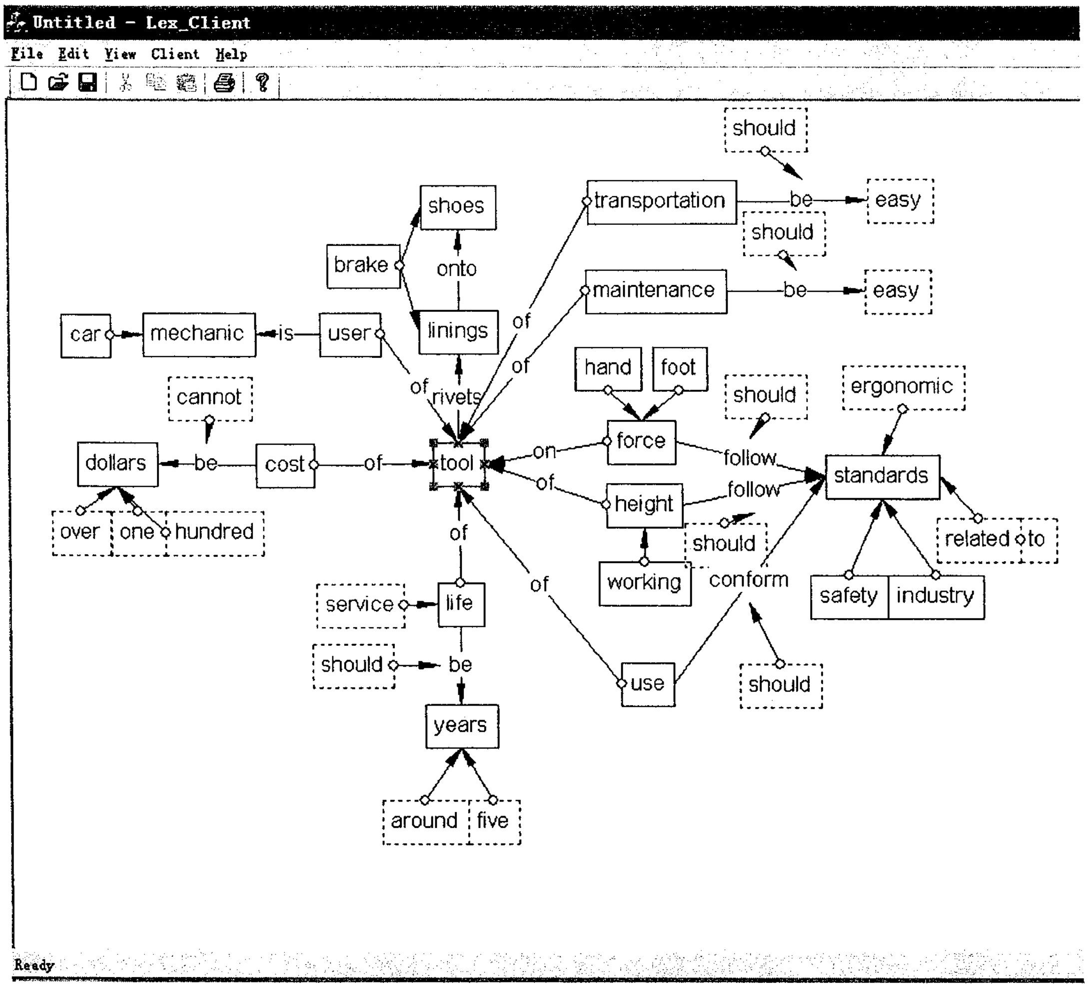


Figure 44 ROM diagram of example

It can be seen from Figure 44 that tool is the product to be designed. Environment includes: “car mechanic”, “standards”, “brake shoes” and “brake linings”, etc. The infection between the product and environment include: “rivet”, “force”, “cost”, “life”, etc. Some objects do not seem to fall into any category at this stage, such as “transportation” and “maintenance” because no corresponding environment has been explicitly stated in the problem description. This is up to the proceeding design process for the clarification of the design problem.

Chapter 8

Conclusions and Future Work

The modern enterprises are facing growing pressure from the market. This pressure is usually transferred to the product development, and is eventually reflected in the product design. A cost-effective and high-quality product design largely depends on a better understanding of product requirements, which usually starts from customers and ends with design specifications.

The present thesis aims to provide an effective approach to managing product requirements. The proposed approach transforms customer requirements described in a natural language into formalized specifications based on the environment-based formulation of design problem (Zeng, 2004). In this present thesis, four major tasks in the transformation process are investigated, which are representation of the formalized product system, classification of product requirements, categorization of design problem, and identification of product system from natural language based description of design problem.

The results from achieving the tasks above are as following:

1. A graphic language, named ROM, is developed to represent the formalized structure, which can be used in modeling the entire design process. (Chapter 4)
2. A classification criterion is proposed to elicit and adjust environment assumptions, according to the different human and built environments underlying

the events throughout the product life cycle. This criterion is used to identify the critical constraints on the design from the environment assumption. (Chapter 5)

3. A categorization method is established to detect new design problems by finding conflicts existing in the identified product system. This categorization method ranks product requirements into eight levels of priority. (Chapter 5)
4. A software prototype, ROMA, is designed and implemented as a tool to help designers transform a design problem described in natural language into a formalized product system represented by the ROM language. (Chapter 6)

A case study about a rivet setting tool design is used to illustrate the concepts proposed in the present thesis. (Chapter 7)

The future work includes an environment ontological template, which can be taken as a knowledge-base for various engineering design problems. The ROMA system can be extended by including a knowledge-base server and an upgraded ROM dictionary. The ROM language itself can be improved by considering more complex situations.

Publications

Publications in Refereed Journals

1. **Z. Chen** and Y. Zeng (2005), Classification of product requirements based on product environment, *Concurrent Engineering Research and Applications: an International Journal*, 2005 (accepted on April 14, 2005).

2. **Z. Chen**, S. Yao, Y. Zeng, and A. Eberlein (2005), Formalization of product requirements: from natural language descriptions to formal specifications, *Computers in Industry* (under review).

3. M. Chen, **Z. Chen**, L. Kong, and Y. Zeng (2005), Gathering and classification of product requirements for medical devices design, *Transactions on SDPS: Journal of Integrated Design and Process Science, Vol. 9, No.4, pp.*61-70.

Publications in Refereed Conferences

1. **Z. Chen** and Y. Zeng (2005), A new approach to managing product requirements, *Advanced Manufacturing Technology'05*, London, Canada, May 16-17, 2005.

2. **Z. Chen**, S. Yao, and Y. Zeng (2004), A systematic approach for the specification of customer requirements, *Inaugural CDEN Design Conference*, McGill University, Montreal, Quebec, July 29-30, 2004.

3. M. Chen, **Z. Chen**, L. Kong, and Y. Zeng (2005), Gathering and classification of product requirements for medical devices design, *The Eighth World Conference on Integrated Design & Process Technology*, Beijing, China, June 12-17, 2005.

Bibliography

Adolph, W.S., 1996. Cash cow in the tar pit: reengineering a legacy system. IEEE Software, 13: 41-47.

Agouridas, V., Baxter, J., de Pennington, A. and McKay, A., 2001. On defining product requirements: a case study in the UK health care sector, Proceedings of ASME 2001 Design Engineering Technical Conferences and Computer and Information in Engineering Conference, Pittsburgh, Pennsylvania.

Akin, O., 1986. Psychology of architectural design. Pion, London, 205 pp.

Asimow, M., 1962. Introduction to design. Prentice-Hall, Englewood Cliffs, N.J.

Bjorner, D. and Jones, C.B., 1982. Formal specification and software development. Prentice-Hall International series in computer science. Prentice/Hall International, 501 pp.

Bodker, S., 2000. Scenarios in user-centered design-setting the stage for reflection and action. Interacting with Computers, 13(1): 61-75.

Bowen, J. and Stavridou, V., 1993. Safety-critical systems, formal methods and standards. Software Engineering Journal, 8(4): 189-209.

Bowen, J.P. and Hinchey, M.G., 1995. Seven more myths of formal methods. IEEE Software, 12(4): 34-41.

Buchanan, R., 1995. Wicked problems in design thinking. In: V. Margolin and R. Buchanan (Editors), The idea of design. MIT Press, Cambridge, MA, pp. 3-20.

Chen, P., 1983. English sentence structures and entity-relationship diagram. Information Sciences, 29: 127-149.

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.

Clarke, E.M. and Wing, J.M., 1996. Formal methods: state of the art and future directions. ACM Computing Surveys, 28(4): 626-643.

Consortium, T.I.E., The Human-machine interface (HMI).

Coyne, R., 2005. Wicked problems revisited. Design Studies, 26(1): 5-17.

Darlington, M.J. and Culley, S.J., 2004. A model of factors influencing the design requirement. Design Studies, 25(4): 329-350.

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.

Dolores, R.W. and Kuhn, D.R., 1999. Lessons from 342 medical device failures, 4th IEEE International Symposium on High-Assurance Systems Engineering. IEEE Computer Society, Washington, D.C, USA, pp. 123-131.

FDA, 2004. Information on Releasable 510(k)s. U.S. Food and Drag Administration.

Feilden, G.B.R., 1963. Engineering Design. Report of Royal Commission - HMSO, London.

Gadamer, H.G., 1975. Truth and method. Seabury Press, New York.

Gangopadhyay, A., 2001. Conceptual modeling from natural language functional specifications. Artificial Intelligence in Engineering, 15: 207-218.

Gero, J.S., 1990. A locus for knowledge-based systems in CAAD education. In: M. McCullough, W.J. Mitchell and P. Purcell (Editors), The Electronic Design Studio: Architectural Education in the Computer Era. MIT Press, Cambridge, MA, USA, pp. 516.

Gero, J.S., 1998. Towards a model of designing which includes its situatedness. In: H. Grabowski, S. Rude and G. Grein (Editors), Universal Design Theory. Shaker Verlag, Aachen, Germany, pp. 47-56.

Gershenson, J.K. and Stauffer, L.A., 1999. A taxonomy for design requirements from corporate customers. Research in Engineering Design, 11: 103-115.

Goel, V., 1995. Sketches of thought. The MIT Press, Cambridge, MA, 320 pp.

Guttag, J.V. and Horning, J.J., 1993. Larch: languages and tools for formal specification. Texts and Monographs in Computer Science. Springer, 250 pp.

Harel, D., 1987. Statecharts: A visual formalism for complex systems. Science of Computer Programming, 8: 231-274.

Hoare, C.A.R., 1985. Communicating sequential processes. Prentice-Hall International series in computer science, 256. Prentice/Hall International.

Hubka, V., Andreasen, M. and Eder, W., 1988. Practical studies in systematic design. Butterworths.

Hubka, V. and Eder, W., 1988. Theory of technical systems. Springer-Verlag.

Hundal, M.S., 1991. Conceptual design of technical systems, NSF Design and Manufacturing Systems Conference. Society of Manufacturing Engineers, Michigan, pp. 1041-1049.

Jiao, J. and Tseng, M., 2004. Customizability analysis in design for mass customization. Computer Aided Design, 36(8): 745-757.

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.

Jones, C.B., 1990. Systematic software development using VDM. Prentice Hall international series in computer science. Prentice Hall, 333 pp.

Kleer, J.D. and Brown, J.S., 1984. A qualitative physics based on confluences. Artificial Intelligence, 24(1-3): 7-83.

Kunz, W. and Rittel, H., 1970. Information science: on the structure of its problems. Information Storage Retrieval, 8: 95-98.

Lawson, B., 1997. How designers think. Architectural Press, 352 pp.

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 (Editors), Computer Aided Conceptual Design'98, Proceedings of the 1998 Lancaster International Workshop on Engineering Design, pp. 165-179.

Lynch, N.A. and Tuttle, M.R., 1987. Hierarchical correctness proofs for distributed algorithms. MIT Technical Report: 137-151.

Manna, Z. and Pnueli, A., 1991. The temporal logic of reactive and concurrent systems: specification. Springer, 448 pp.

McKay, A., de Pennington, A. and Baxter, J., 2001. Requirements management: a representation scheme for product specifications. Computer-Aided Design, 33: 511-520.

Meyer, B., 1985. On formalism in specifications. IEEE Software, 2(1): 6-26.

Milner, R., 1982. A calculus of communicating systems, 92. Springer-Verlag, New York, 260 pp.

Minneman, S.L., 1991. The social construction of a technical reality: empirical studies of group engineering design practice. Ph.D. Dissertation Thesis, Stanford University.

Oxman, R., 2004. Think-maps: teaching design thinking in design education. Design Studies, 25: 63-91.

Pahl, G. and Beitz, W., 1999. Engineering design : a systematic approach. Springer, 544 pp.

Rittel, H. and Weber, M., 1973. Dilemmas in a general theory of planning. Policy Sciences, 4: 155-169.

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.

Schach, S.R., 2004. Object-oriented and classical software engineering. McGraw-Hill, 608 pp.

Schmidt, H., 1995. Advanced English grammar. Pearson ESL, 384 pp.

Shimomura, Y. and Takeda, H., 1995. Representation of design object based on the functional evolution progress model, Design Engineering Technical Conference. ASME, pp. 351-360.

Simon, H., 1969. The sciences of the artificial. MIT Press, Cambridge, MA, 58 pp.

Spivey, J.M., 1988. Understanding Z: a specification language and its formal semantics. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, New York, 144 pp.

Tovey, M., 1997. Styling and design: intuition and analysis in industrial design. Design Studies, 18(1): 5-31.

UKMOD, 1997. Requirements for safety related software in defence equipment. 00-55/Issue 2, Ministry of Defence of U.K.

Ullman, D.G., 2002. The mechanical design process. McGraw-Hill Science/Engineering/Math, 432 pp.

Umeda, Y., Takeda, H., Tomiyama, T. and Yoshikawa, H., 1990. Function, behaviour, and structure. In: J.S. Gero (Editor), Applications of artificial engineering in engineering V. Computational Mechanics Publications, Southampton.

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

Zeng, Y., 2004. Environment-based formulation of design problem. Transaction of SDPS: Journal of Integrated Design and Process Science.

Zeng, Y. and Cheng, G.D., 1991. On the logic of design. Design Studies, 12(3): 137-141.