A scenario-based approach for requirements management in engineering design
Ze-Lin Liu, Zhinan Zhang and Yong Chen
Concurrent Engineering 2012 20: 99 originally published online 16 April 2012
DOI: 10.1177/1063293X12440895
The online version of this article can be found at:
http://cer.sagepub.com/content/20/2/99

Published by:
http://www.sagepublications.com

Additional services and information for Concurrent Engineering can be found at:
Email Alerts: http://cer.sagepub.com/cgi/alerts
Subscriptions: http://cer.sagepub.com/subscriptions
Reprints: http://www.sagepub.com/journalsReprints.nav
Permissions: http://www.sagepub.com/journalsPermissions.nav
Citations: http://cer.sagepub.com/content/20/2/99.refs.html

>> Version of Record - Jun 20, 2012
OnlineFirst Version of Record - Apr 16, 2012
What is This?
A scenario-based approach for requirements management in engineering design

Ze-Lin Liu, Zhinan Zhang and Yong Chen

Abstract
In engineering design, capturing customers’ requirements exactly and transforming them into design specifications are vital to designing a quality product. However, the expressions of customer requirements are normally imprecise and ambiguous due to their linguistic origins. There is still a lack of a systematic approach for elaborating these requirements and transforming them from informal to formal. Therefore, this article provides a scenario-based, systematic approach for requirements management in engineering design. The requirements management process is conceptualized as a three-phase model, and scenarios are integrated into this model for elaborating and formally representing the requirements. A case study of a soy milk maker design is also provided to demonstrate the proposed approach.

Keywords
Customer needs, engineering design requirement, requirement management, scenario

Introduction
The engineering design process starts with some expression of needs that are to be satisfied by the creation of a product (Darlington and Culley, 2002). Therefore, capturing customer needs exactly and effectively is crucial to designing successful products in new product development (NPD) (Tseng and Jiao, 1998). Requirements management, which is defined as a process of creating, disseminating, maintaining, and verifying requirements (Fiksel and Hayes-Roth, 1993), is becoming increasingly important in engineering design today. Traditionally, there are two successive stages involved in this process, that is, identification of “the customer’s voice” and establishment of product specifications based on the information that has been elicited from the customer (Tseng and Jiao, 1998). However, there are primarily two difficulties associated with this process. One is that the requirement statements from different groups of customers are often colloquial, ambiguous, and subjective, which may lead a product designer to understand the needs based on some vague assumptions and implicit inference. Furthermore, poor understanding of the customer requirements may mislead designers into developing a wrong product. The other is that the product specifications, usually represented as a list of product requirements (Pahl and Beitz, 1996), are lacking a formal and structural representation. That is the reason why relatively little progress has been made in providing computer-based assistance for engineering requirements management.

Facing the difficulties above, existing research on requirements management in engineering design is mainly concerned with capturing customer needs in a more effective way and transmitting them into product specifications, involving requirements elicitation, analysis, and specification (Jiao and Chen, 2006). However, there is still a lack of a systematic approach for requirements management in the above field, and rarely does a formal representation scheme of product specifications exist. Correspondingly, requirements engineering (RE) has been an intense research area in the field of software and information systems. Scenarios, or use
cases, play an important role in conveying design requirements information in this field. They are informally expressed examples of interactions between the users and the designed software system, which are widely used in RE (Alexander and Stevens, 2002; Carroll, 2000; Hooks and Farry, 2001; Kaindl, 2000; Rios et al., 2006; Young, 2001). Therefore, we extend the concept of scenarios in engineering design and propose a scenario-based systematic approach for engineering requirements management. The approach is primarily composed of two parts, the development of a formal representation scheme of scenario (carrier of requirement information) and a scenario-based step-wise process model for guiding engineering requirements management.

**Literature review**

As pointed out by Tseng (Tseng and Jiao, 1998), requirements management is a preliminary design stage that is associated with a transition process from “voice of customer” (VoC) to “voice of designer” (VoD). Pahl and Beitz (1996) define a “guideline” to facilitate this transition by establishing product specifications to clarify the design task. However, in their approach, neither tools nor methods are suggested to ensure that the information of customer needs is refined in this stage. The transition is mainly achieved through communication between customers and designers, and largely depends on the designers’ experience. Many researchers are devoted to exploring approaches to elicit analysis and specify customer needs and are also seeking to develop computer-aided tools for requirements management (Baxter et al., 2008; Chen et al., 2002; Chen and Occena, 1999, 2000; Fung et al., 2003; Herlea Damian et al., 2005; Jiao and Zhang, 2005; Kano et al., 1984; Kazemzadeh et al., 2009; Kurakawa, 2004; Misra et al., 2005; Potts, 1999; Prasad, 1998, 2000; Prudhomme et al., 2003; Sutcliffe, 2003; Sutcliffe et al., 1998; Tseng and Jiao, 1998; Yan et al., 2002).

Tseng and Jiao (1998) propose to identify function requirement (FR) patterns from previous product design for addressing a broad spectrum of domain-specific customer requirements. In their model, various FRs are grouped according to the similarity among customers. A requirements management database system is also developed in their work for automatic requirements management. Prudhomme et al. (2003) articulate the notions of customers’ needs and professionals’ needs. They employ a functional analysis method to manipulate the needs of those involved in the product life cycle. Baxter et al. (2008) provide a framework for integrating design knowledge reuse and requirements management, enabling the application of requirements management as a dynamic process. Kano et al. (1984) develop a system for categorizing customer requirements into a product definition.

Quality function deployment (QFD) is widely used to convert customer requirements into technical design requirements. It utilizes a house of quality (HoQ)—a matrix to make explicit the relationships between the customer requirements and the quality characteristics (or technical attributes) of products (Prasad, 1998). Based on the conventional QFD, Prasad (2000) further develops this approach to make it suitable for work group-based engineering design. An alternate framework called concurrent function deployment (CFD) is proposed in his work. Fung et al. (2003) extend the QFD with resource constraints and allocation. Kazemzadeh et al. (2009) integrate the marketing research technology, that is, conjoint analysis and a two-stage clustering method into the QFD process.

Many methods and tools in the field of artificial intelligence are applied in requirements management for product specification development, such as the fuzzy system, neural networks, and expert systems (Chen et al., 2002; Chen and Occena, 1999, 2000; Jiao and Zhang, 2005; Yan et al., 2002). Yan et al. (2002) proposed an integrated approach on the basis of repeated single-criterion sorts and fuzzy evaluation for customer requirement elicitation. Jiao and Zhang (2005) apply the association rule mining technique to exploit the patterns of customer needs to functional requirements mappings. A strategy for acquiring customer requirement patterns is provided by Chen et al. (Chen et al., 2002). In their work, a laddering technique is employed to enable customer requirements elicitation, and an Adaptive Resonance Theory 2 (ART2) neural network is applied as a toolkit for further requirements analysis. They also develop an expert system for concurrent product design by organizing requirements information using a graph decomposition algorithm (Chen and Occena, 2000) and a knowledge hierarchical sorting process (Chen and Occena, 1999).

RE has been an intense research area in the field of software and information systems; some requirements analysis methods in this field have also demonstrated their use in engineering design (Darlington and Culley, 2002; Misra et al., 2005; Potts, 1999; Sutcliffe, 2003; Sutcliffe et al., 1998). Scenarios are widely advocated as a means of improving requirements definition in RE (Misra et al., 2005). Typical works in scenario-based RE includes the ScenIC method (Potts, 1999) and the scenario-based requirements analysis method (SCRAM) (Sutcliffe, 2003; Sutcliffe et al., 1998). The progresses in this field might help understanding design requirements in engineering design domain. Darlington and Culley (2002) compare the differences in the nature of engineering design and software engineering.
state that the formalization of design requirements in natural language can facilitate solution searching in engineering design domain, while resulting in the description of solution in the software engineering domain. Therefore, how to adjust and extend the scenario and make it suitable for requirements management in engineering design is the primary concern of this article. In this regard, there have been some works: Herlea et al. (2005) discuss a method by which the transition from informal to formal expression of the design requirements can be achieved by the paralleled refinement of scenarios and requirements. Kurakawa (2004) proposed a scenario-driven conceptual design information model based on the cognitive design problem-solving process.

The requirements management researches in engineering design reviewed above are only some typical works in this field, more works can be found in the study by Jiao and Chen (2006). However, in these existing works, there is still a lack of formal approaches or any standard for customer requirements modeling in relation to engineering design. There also rarely exists a formal representation scheme of product specifications.

**Understanding scenario**

The term scenario is a widely used concept in RE. In software design, it is employed to model the users of a software system and their activities, the interaction between human and computer (Alexander and Stevens, 2002; Carroll, 2000; Hooks and Farry, 2001; Kaindl, 2000; Young, 2001). This concept has been extended into the field of engineering design for conveying design requirements information. In systematic design approach (Pahl and Beitz, 1996), one method for refining and extending requirements lists is to create scenarios in task clarification phase, which sketch out the answers to “what might happen to the artifact?” and “how should the artifact react.” Kurakawa (2004) defines an artifact scenario as “the design information that represents the actions of all interested people in operating the artifact as well as future events and situations in which the artifact may be involved.”

Scenarios are narrative descriptions of activities of an artifact, involving how it might be used, who might use it or contact with it, when and where it might be used, and the reactions of the artifact in its use. Indeed, scenarios are stories, stories about things happening to an artifact or a summary of a set of real or imagined events and actions. Strictly, a scenario contains at least one specific action of the artifact. It is heavily rooted in people’s daily life experiences. Therefore, expression of requirements in scenarios makes it easier for designers to communicate with customers with a full range of needs likely to be associated with the designed artifact. Besides, scenario can also be used by designers to model the interactions between an artifact and its working environment.

A complete scenario can be specified with the following six characteristic elements: time, location, actor/agent, role (beneficiary, operator, operand, and medium), action/event, and situation.

*Time* and *location* tell when and where a scenario happens. They are the background elements to sketch a general context of a scenario. The *actor/agent* is the subject of an action or an event involved in a scenario. The subject can be human users, raw materials to be processed, or the designed artifact. Each actor is a participant and plays a *role* in the scenario, allowing designer to identify its importance in following design and establishing relations between it and other actors in the same scenario. There are primarily four roles defined here, that is, *beneficiary*, *operator*, *operand*, and *medium*. *Beneficiary* refers to a customer group who will benefit from the actions of the product in the scenario. *Operator* is the product to be designed. *Operand* is the object processed by the *Operator*. It could be energy, material, or signal. *Medium* is the physical element permeating in the working environment such as air or water. It affects the product's working process directly or indirectly, though this impact could be positive or negative. For example, assume a scenario that people use a hair dryer to dry their hair. The *beneficiary* refers to the ones who want to dry their wet hair. The electric current (*operand*) is transformed into heat by the hairdryer (*operator*), and the heat is sent out through the airflow (*medium*). *Action/event* refers to the actions and behavior of actors. Some particular actions of an artifact may cause a state change of the other associated actors. The states before and after an actor performs an action are defined as *situation*. In the above case, the electric energy is transformed into heat when it passes through the hairdryer (*situation*). Note that all these elements need not be included in every scenario; one or the other can be omitted if the general idea of the scenario can be understood without them. In this case, the values of them are denoted as “N/A” (means “null”). In addition, a complex scenario may include causal sequences of events and actions. In order to specify the initial customer needs into more detailed items for artifact design, this scenario can be divided into a series of subscenarios (also called *scene*). Each *scene* has the same elements of a complete scenario but just focuses on one episode of the initial customer needs.

Scenarios are normally expressed in natural language (informal manner) (Alexander and Stevens, 2002; Carroll, 2000; Hooks and Farry, 2001; Kaindl, 2000; Kurakawa, 2004; Prudhomme et al., 2003; Young, 2001). In order to
facilitate its use in requirements management and further the progress of computer-based assistance in this field, a formal representation scheme of scenarios is proposed in the next section.

**Formal representation of scenario**

Scenarios need to be represented in a well-structured and easily understandable manner and should be precise enough and detailed enough to support further product development. Yet no standard language exists for formal representation of a scenario, which may hinder the further progress of computer support in its applications. Therefore, a formal representation scheme is proposed in this section. The formalization is achieved through formally representing the constituent elements of a scenario, that is, how to formally represent the actors, actions, and situation. Based on the study in artificial intelligence (Russell and Norvig, 1995), the first-order logic, as a representation language, is close to a natural language and is suitable to represent objects, their states, and relations. As a result, it is employed here for the scenario formalization.

In this section, before describing the scenario representation scheme, a product case is presented, which will also be used in the following sections of the article as an example. This product is a soy milk maker. A simplified schematic diagram of it is shown in Figure 1. It is an electromechanical device that is mainly used to produce homemade soy milk quickly and conveniently. The raw materials are soaked soybeans and water. Electric energy is taken as the power supply of this device. After a series of processing steps, the output will be the boiled soy milk.

**Representing actors and their states**

Actors are the subjects of actions in scenarios. An actor \( A \) in a certain state \( S \) can be depicted by several properties \( P_i \) \((i = 1, 2 \ldots n)\) of it; these are represented as \( P_i(A) \). And if more than one property is used to describe the state, these properties should be connected with the symbol “\( \land \)” that is, \( P_1(A) \land P_2(A) \land P_3(A) \ldots \)

For example, in the soy milk maker case, the soaked soybeans (actor), before being processed by the machine, are in a form of complete particles. This initial state of the soybeans is described as “soaked (Soybeans) \( \land \) complete (Soybeans).”

**Representing action/event**

Action is the movement or behavior of an actor that would cause a state change of the actor or some other associated actors in the same scenario. This interaction relationship can be represented by the predicate: \( \text{action} (P_r, (A)) \), meaning that \( P_r \) exerts an action upon an object \( A \). In product design, the actions of a designed product (key actions) are always the top concern of designers. Therefore, the product is taken as the default subject of the key action of a scenario. And the action of \( P_r \) upon \( A \) can be simplified as \( \text{action} (A) \). If multiple objects are impacted by this action, the expression should be: \( \text{action} (A_1, A_2 \ldots A_n) \). For example, one particular action of the soy milk maker (\( P_r \)) is “mix the soaked soybeans with cold water.” This phrase can be formalized as “\( \text{mix} \) (soaked (Soybeans), cold (Water)).” The word in bold is the key verb to express the particular action of this product. In this sentence, the product is the subject of this action, the soybeans (\( A_1 \)) and water (\( A_2 \)) are the objects processed by the product.

**Representing situation**

Situation indicates the state change of some actors after a particular action. For one situation, there are at least two states involved, that is, the state before and after an action. A state is composed of some actors, their associated properties, and relations (see Figure 2, left). If actor \( A \) and actor \( B \) are in relation \( R \), then the first-order logic form is \( R(A, B) \). A one-way arrow “\( \Rightarrow \)” is used to denote the state transition in a situation. For example, a script in natural language about a situation that occurs during soybean processing is “In a mixture...”
of soybeans and water, the soy milk maker breaks the soaked soybeans into pieces. Then, the crushed soybeans and water are fully mixed.” The action verb “break” here is used as a watershed to divide the two states, which are derived from this description, formulated as follows (see Figure 2, right).

Representing constraints

The representation of actors, actions, and situations forms the framework of a scenario. In order to elaborate the requirements in scenarios, more details of customer needs must be taken into account. Therefore, besides identifying the characteristic elements of scenarios, a variety of constraints can be derived from the customer needs as supplements to enrich the scenario description.

In present approach, the constraint on \( x \) is simplified as \( C(x) \), where the \( x \) might be an actor with its properties or a particular action. In the aforementioned soy milk maker case, the container inside the product is used to keep the mixture of soybeans and water. So it must be made of waterproof material for rust resistance. The statement requires that the container must be waterproof. In first-order logic, they are expressed as:

1. \( \text{waterproof}(\text{material(Container)}) \)
2. \( \text{resistance}(\text{material(Container)}, \text{rust}) \)

These are constraints on the properties of actors (product or its components). In addition, there are also constraints on some actions of actors, as follows:

3. \( \text{noise}(\text{break}(\text{soaked(Soybeans)})) \leq 80\text{(dBA)} \)
4. \( \text{vibration}(\text{break}(\text{soaked(Soybeans)}), \text{Machinebody}) \)

Constraint (3) denotes that during the breaking process of soybeans, the noise made by the machine should be restricted to 80 dBA or less. Meanwhile, the vibration of the machine body should be avoided (in constraint (4)). By comparing the constraint cases, there are two categories of constraints that can be distinguished here, that is, qualitative constraints and quantitative constraints. The qualitative constraints describe certain desired properties or state of actors or actions. Their expressions are first-order logic sentences. The constraints (1), (2), and (4) are in this category. In contrast, the quantitative constraints are constructed with some measurable properties, meaning that the properties are within a range of values in an interval (pair of numbers indicating the lower and the upper boundaries). The interval can be closed or not closed. So this category of constraints can be formulated as equations or inequalities over the properties of actors. The constraint (3) is a quantitative constraint.

Scenario-based requirements management process

This section introduces how scenarios are employed in the process of engineering requirements management. As shown in Figure 3, the requirements management is a stepwise process that can be conceptualized as a three-phase model (requirements management process model (RMPM)). The starting point of this model is the initial statements of customer needs, and the ending point is the design specifications that are represented by formalized scenarios. Between these two points lies a wide gap that has to be bridged during the preliminary design stage. In this model, the gap is divided into smaller ones, thus reducing the difficulty and complexity of developing the design specifications on the basis of customer needs statements. Three primary phases are included in this division to enhance the ease of modeling the requirements management process: elicitation phase, the decomposition phase, and the formalization phase. The details of each phase will be introduced in the following paragraphs of this section.

Elicitation phase

The first phase is the customer needs elicitation (in Figure 3, top left), which emphasizes a transformation that converts customer verbatim constructs into explicit and objective customer requirements (Jiao and Chen, 2006). The initial customer needs statements are normally qualitative and tend to be vague, partial, and subjective due to their linguistic origins. Sometimes, the statements from one customer group might have considerable conflict with another. Therefore, in this phase, experienced engineering designers should try to determine the requirements as concretely and as completely as possible, including implicit ones, distinguish mandatory and desirable requirements, and look for contradictions between these requirements. This would be an iterative process that is mainly achieved through interviews between customers and designers.
After elicitation, the refined statements of customer needs are often an overview of what customers want and an overall functionality of the artifact to be designed. Subsequently, various scenarios need to be created to further specify the requirements. During this process, the requirement statements will likely be decomposed into different scenarios.

**Decomposition phase**

In this phase, the scenarios begin to be integrated into the requirements management process. There are two reasons for the creation of scenarios: one is to further specify the elicited customer needs statements through identifying the characteristic elements (introduced in section “Representing situation”) of the scenario. Here, the identification is done by inquiry, that is, asking several distinct types of questions by designers.

- **When/where.** These questions ask about the time and location constraints on the event in a scenario, that is, when and where the scenario happens.
- **Who.** These questions request confirmation about the actors/agents who will participate in a scenario. This means that both the product and its associated environmental elements should be clarified by designers in the requirements management process.
- **What is.** These questions ask about the role an agent plays and its actions in a scenario.
- **How.** These questions request further clarification of the agents’ state change process in a scenario after some actions are taken, that is, how the actions are performed.

The other reason for creating scenarios is to structurally interpret the elicited statements through identifying the relationships between these scenarios. There are two kinds of relations distinguished here, that is, the sequential relation (SR) and the concurrent relation (CR). Scenarios in SRs refer to those scenarios that happen sequentially, meaning that the event that occurs in a latter scenario is conditional to the event occurring in its former one. Hence, this is a causal sequence. For example, in the soy milk production, the scenario of “breaking soybeans” occurs after the scenario of “mixing soaked soybeans with water.” Scenarios in CRs indicate that one scenario happens during the occurrence of another. For example, in the soy milk cooking process (as scenario A), the boiled liquid may overflow from the container (as scenario B). These two scenarios are concurrent scenarios. Anyway, the scenarios in relations of SR or CR can be summarized as interrelated scenarios. They are strongly associated with the state transformation process of the operand (energy, material, or signal) in the use of product. Besides, there are also scenarios with no direct correlations, for example, the installation and transportation of the soy milk maker are both product-usage scenarios but have no direct relations between them and the scenarios of the soy milk production process.

**Formalization phase**

In the present approach, the formalization of requirements is achieved by formally representing various scenarios. The technology used for formal representation of scenarios has been introduced in section “Formal representation of scenario.”

After the formalization phase, the scenario-based requirements management process is finished. The three sequential phases indicate a transition process from initial customer needs statements to the final formalized design specifications. During this process, the results of a latter phase are built based on the foundation of its former phase. In this way, the formal product specifications can be traced back to the initial natural language statements. Therefore, a clear mapping between them is assured.

**An illustrative case**

A soy milk maker is a small kitchen appliance that automatically cooks soy milk for family members. This section reports a case study conducted for the soy milk maker design to practice the proposed scenario-based requirements management methodology. In order to compare our approach, a design specification (in requirements list) for the same case is also presented in Appendix 1.

**Elicited customer needs statements**

As mentioned in section “Scenario-based requirements management process,” the start point of requirements management is often a large collection of natural language requirements statements. The elicitation of initial customer requirements is achieved through comprehensive interviews with customers. The communication process is beyond the concern of this article, and thus, it is not given in detail here. Following are the elicited customer needs statements (in italic):

*Fresh soy milk is a daily drink for many people, but getting up early in the morning to purchase soy milk from street vendor is an unpleasant thing. Thus, customers require a small appliance to make high quality soy milk at home for family members (3-4 people). The soy milk making process should be simple and safe. This process must be finished within a short time (30 min), and the cleaning of the device after use should also be taken into account. Because it is designed for...*
**Scenarios — Soy milk maker**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
</table>
| I. | Fully mix the soaked soybeans with certain amount of cold water.  
*The amount of soybeans is about 75-85 g for one time, and water is between 1000-1600 ml.  
The mixing process should be less than 3 min.* |
| II. | Break the soaked soybeans in water, then the soy milk dissolved in the water.  
*In the breaking process, the vibration of the machine should be prevented, and the noise should be restricted to 80 dBA or less.  
The breaking process should be finished within 10 min.* |
| III. | Fully heat the Dissolved soy milk and the strained soybean residua.  
*Ambient temperature around the machine should be controlled within 15°C-28°C.  
The temperature of the outer wall of the machine should be kept under 40°C for user protection.  
The heating process should be about 15 min at least to ensure the soy milk is fully cooked.* |
| IV. | Separate the hot soy milk and the soybean residua.  
*The hot drink should be prevented from being contaminated by some parts of the machine.  
The process should be less than 2 min.* |
| V. | Following the user manual, a user can assemble the product conveniently.  
The product should be partly pre-assembled, main components (assembly) can be mounted by a customer. |
| VI. | The boiling soy milk overflows from the machine.  
*In the cooking process, a mechanism is needed to stop the boiling soy milk from overflowing.* |
| VII. | User cleans the product (these parts contact with soy milk in the cooking process) with water.  
*Moreover, the product should be cleaned.*  
*The electric control part is not directly contact with the soy milk cooking part.* |
| VIII. | A user moves the machine all by himself/herself in his/her house.  
*One person is sufficient for transportation.* |

**Additional specifications:** Budget-priced (max. 150 S) ...

*Figure 4. Informal scenarios graph of soy milk maker.*

*Requirement decomposition*

Based on the elicited statements of customer needs in the last subsection, various scenarios are created to decompose the statements into many slices, as shown in Figure 4. The creation of a scenario is achieved by identifying its characteristic elements, following the inquiry-based way introduced in section “The decomposition phase.” As shown in Figure 4, scenario(I) (abbreviated as S(I)), S(II), S(III), S(IV), and S(VI) are relevant scenarios that are directly associated with the soy milk production process. Among them, the S(I), S(II), S(III), and S(IV) are in SRs, dividing the soy milk processing into four sequential steps. The bold verb in each of these four scenarios denotes one action of the soy milk maker. S(III) and S(VI) are in CR. The remaining scenarios (S(V), S(VII), and S(VIII)) are about the installation, cleaning, and transportation of...
the machine, respectively. They are relatively independent. In addition, the specifications associated with the event happening in a scenario are the sentences in italics, as restrictions on the scenario. Ultimately, customer requirements are not only multifaceted but are also intrinsically open ended; more scenarios can be
added to this graph. However, there is no real sense that all the requirements can be represented by scenarios. For instance, the budget price of the designed product is an economic requirement that is hard to assign to a particular scenario, so it is listed as an additional specification at the bottom of this scenario graph.

**Scenario formalization**

In contrast to Figure 4, Figure 5 illustrates the scenario graph after formalization. In each formalized scenario, the one-way arrow “⇒” indicates the state change in a situation. The change is usually caused by a product action, which is described by a predicate sentence with a bold verb. The contents in italic are the constraints in action, which is described by a predicate sentence with a guideline for designers to acquire customer requirements, that is, requirements management approach can be mainly summarized into the following aspects. First, requirements management is performed as a stepwise refinement process in our approach. The RMPM (in Figure 3) provides a guideline for designers to acquire customer requirements and subsequently refine these requirements step by step. Second, scenarios play a role as information carrier to represent the specified customer requirements, that is, the informal scenarios in natural language can assist designers to clarify customer needs, and the formal scenarios represent requirements in a certain level of abstraction, which can facilitate design solution searching. The formal format is also helpful for further development of computer-aided tools in engineering requirements management. Third, the scenarios are created with a primary concern for the working process of designed products. So the requirements generated in this process are interspersed into several interrelated scenarios, rather than the causal form in a requirements list. The correlations of these requirements are thus established.

In conclusion, this article provides a scenario-based systematic approach for the identification, elaboration, and specification of engineering design requirements. The requirements management process is conceptualized as a three-phase model, and the applicability of scenarios is also shown in the model. In this approach, the concept of scenarios is extended on the basis of scenarios known in the field of RE. Furthermore, a comprehensive view of scenario is introduced on the aspects of concept, purpose, content, and form. A formal representation scheme of scenarios is also developed for further advancement of computer support in engineering requirements management. A case study on the requirements management for a soy milk maker design illustrates the operations in the present approach.

**Funding**

This research is supported by Natural Science Foundation of China (grant nos. 50975173 and 50933004), Ministry of Science and Technology of China (grant no. 2008AA04Z108), and Science and Technology Commission of Shanghai Municipality (grant no. 09QA1402800).

**Acknowledgment**

We are grateful to the reviewers and the editors for their constructive suggestions. We specially thank Mr Reynolds for improving the quality of this article.

**References**


Appendix 1 Design specifications of soy milk maker

List of requirements: soy milk maker

1. Geometry:
   (a) Consideration of the maximum capacity of soy milk production for one time 1200–1800 mL (for 2–3 people to drink)
   (b) Consideration of the length of a adult user’s palm and arm (portable)

2. Kinematics:
   (a) Speed range of build-in motor to break soaked soybeans in a container
   (b) The machine body should stand stable during the working process, vibration control
   (c) The soy milk production process should be finished in a short time (within 30 min)

3. Weight:
   (a) Empty: maximum of 3.5 kg
   (b) Dry soybeans: 80 ± 5 g
   (c) Water and soybeans: 1300–1600 mL

4. Energy:
   (a) Electric motor + 220 V AC or electric motor + accumulator
   (b) 30-min duration of use per time, 1–10 times of use per day
   (c) Ambient air temperature during working: 15°C–28°C

5. Material:
   (a) Waterproof, suitable for kitchen use
   (b) Free from rust, moisture resistant
   (c) Consideration of temperature raise during soy milk production
   (d) Food safe, no contamination to the soy milk
   (e) Easily cleaned, hard for soy milk residue to attach on the container
   (f) Materials against minor collision

6. Safety:
   (a) Overflow prevention
   (b) Electric power leakage prevention

(continued)
Appendix 1 (continued)

List of requirements: soy milk maker

7. Ergonomics:
   (a) Portable handle
   (b) Noise control (maximum of 80 dBA noise development)
   (c) Simple operating and control elements

8. Construction:
   (a) Complete delivery, possibly only partly preassembled, and container/control panel/handle/filter possibly to be mounted by a user

9. Transportation:
   (a) One person is sufficient for transportation

10. Maintenance:
    (a) Easily cleaned with water
    (b) No wearing parts, if easily replaceable
    (c) Easily disassembled to reload raw materials

11. Costs:
    (a) Budget-priced (maximum of $150)

12. Recycling:
    (a) Enable separation of waste disposal, environment friendly

Biographies

Ze-Lin Liu is a PhD candidate of Institute of Engineering Design, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai. He has a major in mechanical design. His research interests are knowledge-based system, artificial intelligence in product/engineering design, and supporting tools.

Zhinan Zhang is a postdoctoral candidate in Shanghai Jiao Tong University, China. He also received his PhD degree in this university. His research interests include design theory and methodology, knowledge flow, and supporting tools. Also he has more than 3 years experience on engine tribology.

Yong Chen is an associate professor of Institute of Engineering Design, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai. He received his PhD degree in Zhejiang University. His research interests include conceptual design, artificial intelligence in product/engineering design, and knowledge-based system.