A requirement management database system for product definition

Jianxin Jiao
The Hong Kong University of Science and Technology, Kowloon, Hong Kong

Mitchell M. Tseng
The Hong Kong University of Science and Technology, Kowloon, Hong Kong

Abstract
This paper presents a database system developed to provide a computerized environment for requirement management during the product definition phase. The scope of this database system is to facilitate and demonstrate a methodology for product definition by recognizing and adopting functional requirement patterns from previous product designs so as to address a broad spectrum of domain-specific customer requirements and organize requirement information for product specifications. The database system improves the product definition process during design and redesign efforts by integrating customer and design information all together and by reusing this information. A prototype requirement management database system is implemented on a PC platform using Microsoft Access.

1. Introduction
The first phase of the overall product realization process is product definition (Ulman, 1992). During this phase, the design team may explore any combination of customer needs, corporate objectives, product ideas, and related technological capabilities, concluding the process with a definition of the product. Usually, the product’s definition is represented as a list of product requirements, also known as product specifications or target values. This information is often a mix of quantitative values and qualitative descriptions of the product. Often, only highlights of the design are recorded for future reference, or the company may produce a formal document that must routinely undergo many amendments along with scrutiny, and may require to be signed off by many individuals (Ulman, 1992; Redmond, 1988).

Most researchers in the field and industrial designers involve themselves wholeheartedly in the process of mapping from customer needs to design solutions, i.e. the latter part of the design process subsequent to product definition. However, they are involved only spasmodically and usually highly subjectively in the first part of the design process, i.e. product definition (Redmond, 1988). As discussed by Tseng and Jiao (1997), there are formidable hindrances inherent in the product definition process, including contextual mismatching, lack of defined structures in requirements, and no structured mapping from requirements to design parameters.

Product definition has long been a tedious, time-consuming and error-prone effort enacted between customers, marketers, and designers (Figure 1). This is further compounded by the tendency for requirements to be vague and fuzzy and difficult to manage. In most cases, requirements are negotiable and conflict with each other, and tradeoffs are often necessary. In addition, in the practice of concurrent engineering, product development teams must keep track of myriad requirements derived from different perspectives on the product life cycle, including manufacturing, reliability, maintainability, and environmental safety, to name but a few. Yet, despite great advances over the past decade in computer-aided design and engineering, there has been relatively little progress in providing analogous support for requirement management. As a result, Fiksel and Hayes-Roth (1993) point out the necessity to manage requirement information during the product definition process. They defined requirement management as the process of creating, disseminating, maintaining, and verifying requirements.

The requirement management process consists of four main functions that are performed repeatedly in an iterative fashion. These are requirement elicitation, requirement analysis, requirement tracking, and requirement verification (Fiksel and Hayes-Roth, 1993; Tseng and Jiao, 1997). Requirement elicitation deals with eliciting customer needs and acquiring the voice of the customers. Requirement analysis is the process of interpreting customer needs and deriving explicit requirements that can be understood and interpreted by people and/or computer programs. Requirement tracking involves continuous interchange and negotiation within a project team regarding conflicting and changing objectives. Requirement verification embodies the procedures for determining whether or not a product design complies with a designated set of requirements.

Straightforwardly, requirement management automation will facilitate a more structured product development process, at the same time, a more effective integration of humans and machines that reduces product development costs and cycle time while improving product quality. Engineers will be able to comprehend and articulate better the many relationships and tradeoffs among product requirements and different technical approaches. It has been foreseen that, as companies review and enhance their product
development processes, they will increasingly demand various types of automated requirement management capabilities (Fiksel and Hayes-Roth, 1993). As a result, it is imperative to explore requirement management methodologies and develop computer tools to support requirement management automation.

Towards this end, this paper presents a database system developed to provide a computerized environment for requirement management during the product definition phase, namely the requirement management database (RMDB) system. The scope of this database system is to facilitate and demonstrate a methodology for product definition by recognizing functional requirement (FR) patterns, noted as the PDFR methodology. The PDFR methodology adopts FR patterns from previous product designs to address a broad spectrum of domain-specific customer requirements and to organize requirement information for design specifications. The RMDB system is an implementation of the PDFR methodology to improve the product definition process during design and redesign efforts. The prototype RMDB system is implemented on a PC platform using Microsoft Access.

In the next section, the background research leading to the PDFR methodology is presented upon which the requirement management database system is based. In Section 3, the system design issues involved in the development of the RMDB system are described, along with an existing database technologies review. The software selection and the RMDB system implementation are also discussed in Section 3. In Section 4, an example of product definition in designing power supply products is presented with a focus on the usage of the RMDB system. In Section 5, a plan for future work is presented and finally in Section 6 the paper is concluded.

2. Requirement management methodology

Approaches to defining product specifications by capturing, analyzing, understanding, and projecting customer needs, sometimes called the Voice of the Customer (VoC), have received a significant amount of interest from both academia and practitioners in recent years (Fung and Popplewell, 1995; Fiksel and Hayes-Roth, 1993). A method used for transforming the VoC to product specifications is developed by Ofuji et al. (Shoji et al., 1993), in which semantics methods, such as the KJ method (affinity diagram) and MPM (multipickup method), are applied as the basis for discovering underlying facts from affective language. Kano et al. (1984) develop systematics to categorize customer needs for product definition. Towards this end, marketing researchers emphasize customer profiling by applying regression analysis to compare customer characteristics to determine their overall rankings in contribution towards profitability (Jenkins, 1995). Marketing research techniques include focus groups, one-on-one interviews, and similarity-dissimilarity attribute rankings (Griffin and Hauser, 1992). While these types of activities are helpful for discovering the VoC, it is difficult for them to obtain design information because marketers do not know what engineers need to know. They have shortcomings in facilitating the synchronization of marketing and engineering to develop product definitions coherently.

In the engineering community, total design (Pugh, 1991) points out the importance of understanding and interpreting market information and encapsulating these in a comprehensive and thorough product design specification. Quality function deployment (QFD) (Clausing, 1994) joins both marketing and engineering efforts using a very basic tool of the matrix, namely the house of quality (HoQ). While QFD excels in converting customer information to design requirements, it is limited as a means of actually discovering the VoC (Hauge and Stauffer, 1993). To empower QFD with marketing aspects, Fung and Popplewell (1995) propose to pre-process the VoC prior to its being entered as customer attributes into the HoQ. In pre-processing, they adopted an affinity diagram (KJ method) to categorize and the analytic hierarchy process (AHP) to prioritize the customer requirements. Fukuda and Matsuura (1993) also propose to prioritize the customer’s requirements by AHP for concurrent design.

In summary, most approaches assume product development starts from a clean sheet of paper. In practice, however, most new products evolve from existing products. There is little attention paid to evolutionary product design in terms of product definition.
Historical data, the product evolution path, and feedback from customers on current products are often considered only implicitly, if not ignored. As a result, product design seldom has the opportunity to take advantage of the wealth of customer requirement information in existing products. Furthermore, with a shortened product life-cycle, expensive investments in product development, and the proliferation of product varieties, the existing approaches are often constrained by the schedule deadline and lack of objectivity in defining product specifications.

In order to improve the product definition process during design and redesign efforts, we develop a systematic approach to product definition through recognizing functional requirement patterns from existing products (noted as PDFR; Tseng and Jiao, 1997). These patterns help modify product designs based on historical data about customers’ needs through underlying patterns of functional requirements (FR) as well as patterns of their interrelationships.

In PDFR methodology, FR patterns consist of FR topology, FR classification, and FR templates. The terminological FR variables and the interrelationships among them are referred to as FR topology, which implicates the decomposition hierarchy of FR variables and a taxonomy of FRs. FR topology enhances the design taxonomy (Hauge and Stauffer, 1993), which aims at assisting customer requirement acquisition, in that FR topology is more specialized to reflect product domain-specific requirements. FR classification manifests itself through and corresponds to a spectrum of product families with different sets of requirements. FR templates help product redesign by providing a history of past designs, thus allowing for the easy reuse and inheritance of design knowledge.

The PDFR methodology is divided into two phases, i.e. the FR pattern recognition phase and the FR pattern adoption phase. The FR pattern recognition phase is a preparatory stage in which FR patterns are extracted from historical data regarding existing designs. The FR patterns, in essence, represent a set of FRs for a spectrum of product offerings in a company. Such a generic representation of FRs can then be used to develop product specifications for a new design in the FR pattern adoption phase.

Figure 2 summarizes the procedures of the PDFR methodology.

3. Requirement management database system

Recognizing the importance of improving customer needs elicitation, generating product specifications, and managing this information, much of the development efforts of the PDFR methodology are geared towards organizing the customer and design information in such a way that it brings together marketing and design engineering in the product definition phase. As a methodology of organizing specifications for engineering, PDFR facilitates the storage and retrieval of customer requirements and product specifications. FR patterns allow for easy reuse of design knowledge by providing a design history. Thus, when filled with appropriate information, they accommodate design teams with the knowledge necessary for defining the requirements for a new product.

3.1 Domain information management

The requirement management database (RMDB) is a demonstration and implementation of PDFR methodology to organize customer and design information effectively for product definition. The tasks of the RMDB include integrating customer requirements and product specification, generating product specifications that will satisfy customer
needs, and making the information available for downstream design activities. RMDB employs FR classification patterns to organize various products and a functional decomposition hierarchy to represent customer needs. FR patterns are extracted from product repositories and consist of the FR topology and FR templates corresponding to different FR classifications.

To help in the elicitation of customer needs, FR topology is used to prompt the marketer and the customer for their interactive question probe, which alleviates the domain knowledge acquisition bottleneck in the product definition process. To assist design engineers to define product specifications, customer needs should be presented in an organized and systematic way. A FR template conforms to a product family and provides a class-member relationship between a product family and a new design. That is, a new product specification can be defined through instantiating the related class template for specific customer needs. Therefore, the integration of the customers, the marketer, and the design engineer can be supported by underlying FR patterns. All FR patterns, used by marketing to probe the customer, are also used to organize customer requirements and design specifications. By
doing so, marketing and design engineering share customer information in the identical format and in a structural manner. A typical product development adopting RMDB is resembled in Figure 3. In such a model, both the marketer and design engineer use the same FR patterns to elicit and organize customer needs.

3.2 Data model and development software

Popular data models in database management systems include hierarchical, network, relational, and object-oriented. The hierarchical data model allows the user to represent each one-to-many relationship by a parent-child designation. In the hierarchical model, many-to-many relationships can be implemented only in a clumsy way, which often results in a redundancy stored data (Elmasri and Navathe, 1994). In addition, the operations of insertion and deletion become very complex as a result of strict hierarchical ordering. In practice, there often exist many-to-many relationships between customer needs and design attributes in product definition. Therefore, RMDB cannot be adequately supported by the hierarchical model. Like the hierarchical model, a network schema can directly represent one-to-many relationships. Unlike in the hierarchical model, a record type can be child of more than one parent. However, in addition to the complexity of the network model, there is a restriction on the configuration of many-to-many relationships in a network model, in which a record type that is the child for a relationship cannot also be the parent for a relationship. The result is a schema that has only shallow levels (Elmasri and Navathe, 1994) and is therefore not applicable to RMDB. The relational model has been popular owing to its simple tabular data format and convenient support to nonprocedural requests and data independence. Although the object-oriented approach promises to reduce the time and cost involved in software development, it is not yet fully developed to handle complex engineering design tasks (Law et al., 1990). Recently, the trend is to extend the relational model using an object-oriented approach that provides certain features lacking in the relational model (Law et al., 1990). Engineering applications like product design involving teamwork can benefit from the object-oriented extensions without sacrificing the advantages of the relational model. The enhanced entity-relationship (EER) model (Elmasri and Navathe, 1994) incorporates important concepts from the object-oriented approach into the entity-relationship (ER) model to represent more complex requirements of engineering applications.

The end-users of RMDB are marketers and design engineers, therefore a PC-based platform is considered. The same consideration is given to the Windows operating system owing to its popularity. The RMDB prototype system is implemented using Microsoft Access database management tool (User's guide, 1994). Microsoft Access is a relatively easy to use software package system for the Windows environment with powerful database management capabilities. As a virtual programming tool with strengths in relational database applications, it provides event-driven mechanisms to build graphical user
interface easily. Its floating toolbar, along with cue cards and powerful wizards simplify many programming tasks, resulting in a professional looking program.

### 3.3 Functional analysis

Requirement analysis is the first step of the database life-cycle. RMDB attempts to support three sets of users: the customers who select product offerings from a company by specifying their wants, the marketers who play an important role in bridging the customers and the engineers to acquire and refine customer requirements, and the engineering designers who define product specifications according to elaborated customer requirements and manufacturing considerations.

The functional modeling of RMDB is shown in Figure 4, described using IDEF0.

#### 3.4 Database design

Data modeling is an important step following requirement analysis to help the database designer conceptualize the entities and their relationships. The ER model for the RMDB system is shown in Figure 5 with major entities and their relationships. The subsequent logical design aims at obtaining a representation that uses as efficiently as possible the facilities for structuring data and modeling constraints available in the logical design (Elmasri and Navathe, 1994). In the logical design of the RMDB system, the conceptual data schema (ER model) is translated into a logical schema (Figure 6) tailored to the specific database management system (Microsoft Access). All these issues are implemented during the physical database design by using Microsoft Access database software. The RMDB system interacts with the user through various views or forms. The first form, as shown in Figure 7, is the main menu of a prototype RMDB system. The menu structure can be defined according to user identification and task analysis. The menu structure of the RMDB system embodies four sets of tasks as shown in Figure 8. They are product selection, order processing, product specification, and DB maintenance, conveying the customer view, marketing view, engineering view, and system view on the RMDB system, respectively.

### 4. RMDB for power supply design

Power supplies possess a special trait in product definition common to industrial products. That is, in addition to the general hindrance shown in Figure 1, the product...
Definition is a cooperative effort among the customer, the marketer and the engineer. Unlike many consumer products, it seldom can be developed by the customer or engineer alone. By applying the PDFR methodology, the RMDB system alleviates the difficulties inherent in the product definition for power supply design. The screenshot in Figure 9 illustrates the elicitation of customer needs with the aid of FR topology so as to help customers articulate requirement information in completeness. FR templates offer a structured mechanism for defining product specifications, as shown in Figure 10.

5. Future work

Design can be described as a mapping process from customer requirements in the functional domain to design solutions in the physical domain (Suh, 1990). RMDB captures requirement information in the functional domain to facilitate product definition. The subsequent design mapping can also be supported by the PDFR methodology if design information can be appropriately represented and organized in the database. That is, the requirement management (RM) database, as shown in Figure 3, is extended to act as an engineering database, along with a comprehensive product repository. In such a way, a concurrent design environment can be established within a coherent framework.

Moreover, in evolutionary design, how to find a similar case for a new design is often experience dependent and mostly by trial and error. Case-based reasoning technique has proved promising in finding a similar design for accommodating the new design context, in which the PDFR methodology can perform as a fundamental mechanism for case memory organization.

6. Summary

This paper presents a domain independent requirement management database system for organizing requirement information during product definition. It is based on a methodology for requirement management which provides a framework to integrate customer and design information and to reuse such information. A prototype RMDB system is implemented on the PC platform using Microsoft Access database software. An application of RMDB system to power supply design is reported to illustrate the feasibility and the potential of the RMDB system.

References


---

**Figure 10**
Defining product specifications based on FR templates

**Figure 11**
Screen shot of product specification report


