Platform-Based Product Design and Development: A Knowledge Intensive Support Approach

Xuan F Zha, Ram D Sriram
Manufacturing System Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899, USA

Abstract

This paper presents a knowledge-intensive support paradigm for platform-based product family design and development. The fundamental issues underlying the product family design and development, including product platform and product family modeling, product family generation and evolution, and product family evaluation for customization, are discussed. A module-based integrated design scheme is proposed with knowledge support for product family architecture modeling, product platform establishment, product family generation, and product variant assessment. A systematic methodology and the relevant technologies are investigated and developed for knowledge supported product family design process. The developed information and knowledge-modeling framework and prototype system can be used for platform product design knowledge capture, representation and management and offer on-line support for designers in the design process. The issues and requirements related to developing a knowledge-intensive support system for modular platform based product family design are also addressed.

Keywords: product platform, product architecture, product family, modular design, knowledge support

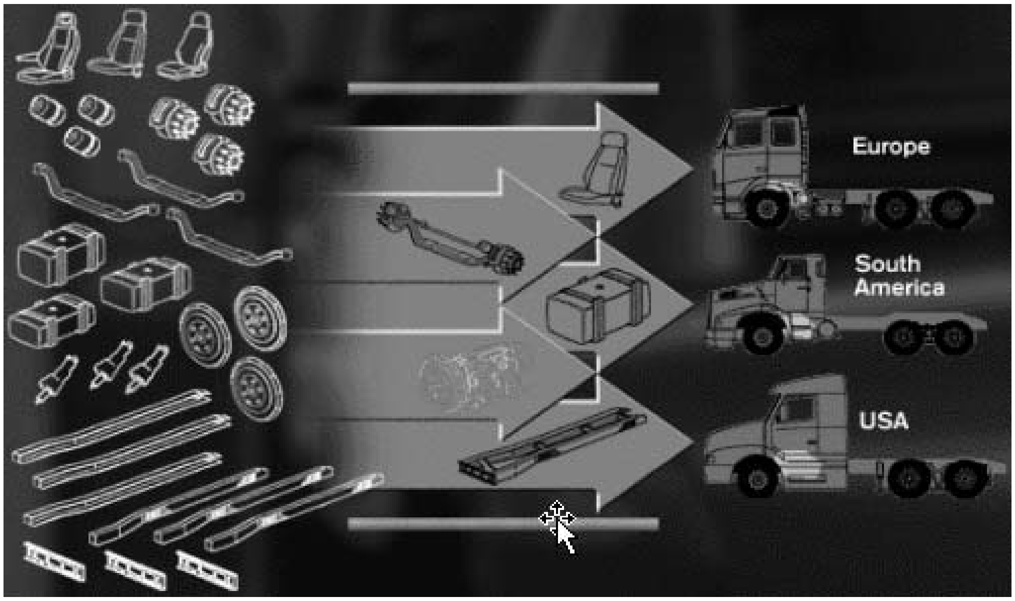
1. Introduction

Product family is a group of related products that share common features, components, and subsystems, and satisfy a variety of market niches. Product platform is a set of parts, subsystems, interfaces, and manufacturing processes that are shared among a set of products (Meyer and Lehnerd 1997). A product family comprises a set of variables, features or components that remain constant in a product platform and from product to product. The design of platform-based product family has been recognized as an efficient and effective means to realize sufficient product variety to satisfy a range of customer demands in support for mass customization (Tseng and Jiao 1998). The platform product development approach usually includes two main phases: 1) the establishment of the appropriate product platform; and 2) the customization of the platform into individual product variants to meet the specific market, business and engineering needs. The establishment, maintenance and application of the right product platform are very complex. Figure 1 gives an example of modular platform-based motor truck family design and development in volvo.

Contemporary design processes have become increasingly knowledge-intensive and collaborative (Tong and Sriram 1991a,b; Sriram 2002). Knowledge-intensive support becomes more critical in the design process and has been recognized as a key solution towards future competitive advantages in product development. To improve the product family design for mass customization process, it is imperative to provide knowledge support and share design knowledge among distributed designers. Several quantitative frameworks have been proposed for both phases in platform product development. They provide valuable managerial guidelines in implementing the platform product development approach. However, there are very few systematic qualitative or intelligent methodologies to support the product development team members to adopt this platform product development practice, despite the progress made in several research projects (Zha and Lu 2002a,b; Simpson 1998, Simpson et al. 2001, Nanda et al. 2006).

The aim of this paper is to discuss knowledge support methodologies and technologies for platform-based product family design. An integrated modular product family design process with knowledge support is explored. This process includes customer requirements modeling, product architecture modeling, product platform establishment, product family generation, and product assessment. The driving force behind this work is to develop a formal, technical approach based on the modular product design paradigm to efficiently and effectively model and synthesize a family of products (product platform and variants) which can provide increased product variety necessary for today's market.

The organization of this paper is as follows. Section 2 reviews the background and current research status related to platform-based product development and product family design. Sections 3 and 4 outline a platform-based product development model and a modular design methodology for product family design. Sections 5 and 6 discuss the module-based product family design process and discuss a knowledge support framework for modular product family design respectively. Section 7 addresses the relevant issues and technologies for implementing the knowledge-intensive support system for modular product family design. Section 8 summarizes the paper and points out the future work.



[Modules→truck platform→truck variants]

Figure 1: Modular truck family design and development (Courtesy of Volvo)

2. Literature Review

There are various approaches and strategies for designing families of products and mass customized goods reported in the literature. These techniques appear in varied disciplines such as operations research (Gaithen 1980), computer science (Nutt 1992), marketing and management science (Kotler 1989; Meyer et al 1993; Pine 1993), and engineering design (Fujita et al. 1997; Simpson et al. 1998, 2001; Ulrich et al. 1995).

Two key concepts underlie existing schemes for product family modeling: product family architecture and product family evolution. There are three kinds of approaches widely used for representing architecture and modularity of product family: 1) product-modeling language (Erens et al. 1997), 2) graphic representation (Ishii et al. 1995; Agarwal and Cagan 1998), and 3) module or building block (BB) (Tseng and Jiao 1996; Gero 1990; Fujita and Ishii 1997; Rosen 1996). The product modeling language allows product families to be represented in three domains: functional, technological, and physical. It provides an effective means for representing product variety, but offers little aid for design synthesis and analysis. In the graph structure, different types of nodes denote the individual components, subassemblies and fasteners, and the links denote dependencies between the nodes. However, it lacks the ability to model product family constraints. Although the grammar approach is conjoint with the graph representation to improve its capability of representation, graph grammars are only able to implicitly capture product architecture information and product family information by production rules (Siddique and Rosen 1999, 2001). A model specifically tailored for representation of a product family architecture is the building block model, which is derived from the concept of using modules to provide varieties. Building blocks are organized in the hierarchical decomposition tree architecture (systems, modules, and attributes) from both functional and technical viewpoints (Kusiak and Huang 1996; Jiao et al. 2000). Under the hierarchical representation scheme, product variety can be implemented at different levels within the product architecture. However, module-based product architecture reasoning systems are currently being developed from different viewpoints (Rosen 1996).

Much work done in strategic management and marketing research seeks to categorize or map the evolution and development of product families (Meyer et al. 1993; Wheelwright et al. 1989, 1992). Sanderson (1991) introduces the notion of a “virtual design” to evolve into product families. Wheelwright and Clark (1992) suggest designing “platform projects” and Rothwell and Gardiner (1990) advocate “robust designs” as a means to generate a series of different products within a single product family. These product family maps are less formal and are intended primarily for strategic management; they are actually product platforms that can be used to generate product variants to form a product family. However, none of these approaches have been formalized for design synthesis.

The basic concept of a family of products or multi-product approach is to obtain the largest set of products through a standardized set of base components and production processes (McKay et al. 1996). A key aspect in developing product families is to consider the flexibility of the assembly and manufacturing process. Stadzisz and Henrioud (1995) describe a methodology for the integrated design of product families and assembly processes through the use of web grammars (Pfaltz and Rosenfeld, 1969). The work clusters products based on geometric similarities to obtain product families so as to decrease product variability within a product family and minimize the required flexibility of the associated assembly system. It is more applicable for later design stages when more quantitative information is available.

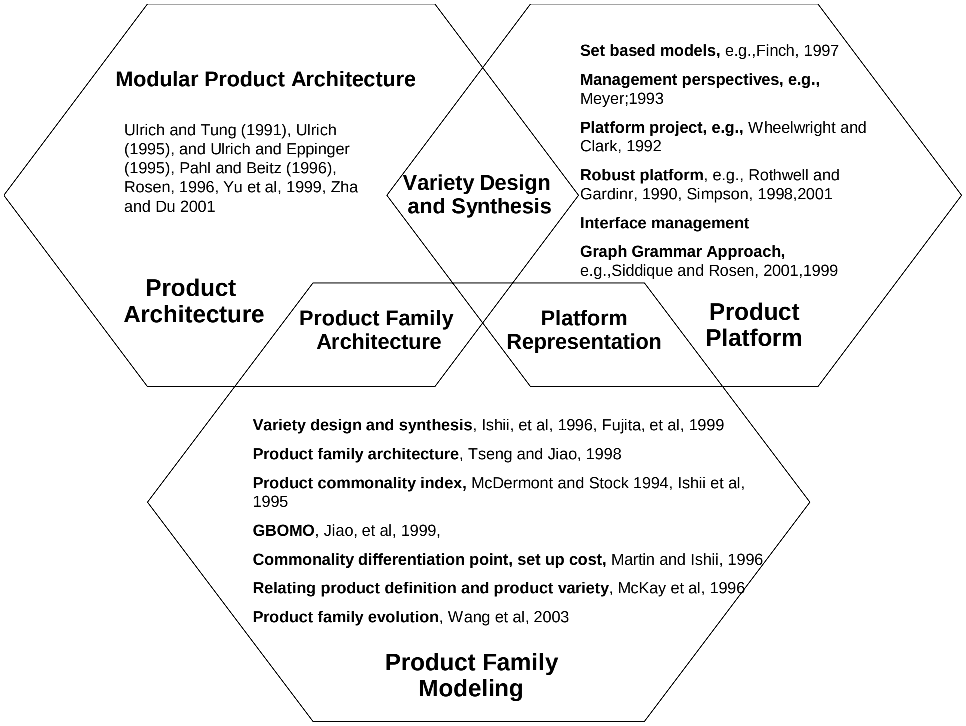

Modular Product Architecture
Ulrich and Tung (1991), Ulrich (1995), and Ulrich and Eppinger (1995), Pahl and Beitz (1996), Rosen, 1996, Yu et al, 1999, Zha and Du 2001
Product Architecture
Variety Design and Synthesis
Set based models, e.g.,Finch, 1997
Management perspectives, e.g., Meyer;1993
Platform project, e.g., Wheelwright and Clark, 1992
Robust platform, e.g., Rothwell and Gardinr, 1990, Simpson, 1998,2001
Interface management
Graph Grammar Approach, e.g.,Siddique and Rosen, 2001,1999
Product Platform
Product Family Architecture
Platform Representation
Variety design and synthesis, Ishii, et al, 1996, Fujita, et al, 1999
Product family architecture, Tseng and Jiao, 1998
Product commonality index, McDermont and Stock 1994, Ishii et al, 1995
GBOMO, Jiao, et al, 1999,
Commonality differentiation point, set up cost, Martin and Ishii, 1996
Relating product definition and product variety, McKay et al, 1996
Product family evolution, Wang et al, 2003
Product Family Modeling

Figure 2: Overview of related work on platform-based product family design and development

Tseng and Jiao (1996, 1998) developed a set of approaches entitled "Design for Mass Customization (DFMC)" with an emphasis on how to "set up a rational product family architecture in order to conduct family-based design, rather than design only a single product." The family-based DFMC approach groups similar products into families based on functional requirements, product topology or manufacturing and assembly similarity. Accordingly, it provides a series of steps to formulate an optimal product family architecture. Their work is also more applicable in the later stages of design, particularly once the system architecture has been established. Gonzale-Zugasti (2000) proposes a four-step interactive process model for designing a platform-based product family: design requirements and models (e.g., functional requirements, and design constraints, etc), platform design, variants design, and platform evaluation, re-negotiation, and iteration.

The most important characteristics that have been stressed in the literature for designing product families are modularity (Chen et al. 1994,1996; Martin and Ishii 1996; Sanderson 1991; Ulrich and Tung 1991), commonality and reusability (Collier 1981,1982; McDermott et al. 1994), and standardization (Lee and Tang 1997; Ulrich and Eppinger 1995). The concept of functional modularity should be incorporated with the requirements of product families from the product life cycle perspective. Ulrich and Tung (1991) give a summary of different types of modularity. Chen et al. (1996) describe a family of products as a "family of designs" that conforms to a given ranged set of design requirements and recommend designing product families by changing a small number of components or modules. Ishii and his team (Ishii et al. 1995; Martin and Ishii 1996; Chang and Ward 1995) emphasize the computational approaches for product variety design, including representation, measurement and evaluation of product varieties. "Design for Variety" refers to product and process designs that meet the best balance of design modularity, component standardization, and product offering. Uzumeri and Sanderson (1995) emphasize flexibility and standardization as a means for enhancing product flexibility and offering a wide variety of products. McDermott (1994) and Collier (1981) stress commonality across products within a product family as an effective means to provide product variety. Ulrich (1995) and Ulrich and

Eppinger (1995) investigate the role of product architecture and the impact on product change, product variety, component standardization, product performance, and product development management.

In reviewing prior work, we found that several quantitative frameworks have been proposed for product family design. They provide valuable managerial guidelines in implementing the overall platform-based product family development. The overview of related research on platform-based product design and development can be summarized as shown in Figure 2. There are generally two approaches for product family design. One is the top-down approach that adopts platform-based product family design (Simpson 1998, Simpson et al. 2001). The other is the bottom-up approach which implements family based product design through re-design or modification of constituent components of the product. The former one is the current dominant research approach. The research and development work is mainly in the realm of academics and does not provide support for knowledge-based processes. There are very few systematic quantitative or intelligent methodologies that support product development team members to adopt the platform-based product development practice, despite the progress made in several research projects (Zha and Lu 2002a,b, Zha et al. 2004, Wang et al. 2003, Simpson et al. 2001). Much of their work lays a solid foundation for this research on the knowledge supported platform product development. The approach advocated in this work is for companies to realize a family of modular products that can be easily modified, configured and quickly adapted to satisfy a variety of customer requirements or target specific market niches with knowledge support.

3. Platform-Based Product Design and Development

A product family may have its origin in a differentiation process of a base product or in an aggregation process of distinct products. The product family has most impacts on a firm's ability to efficiently deliver large product variety and has profound implications for subsequent product development activities. The product family design process is tightly linked to issues of importance to the entire enterprise: product change, product variety, component standardization, product performance, manufacturability, and product development management. An effective platform for a product family can allow a variety of derivative products to be created more rapidly and easily (cost and time savings), with each product providing the features and functions desired by a particular market segment (Simpson 1998, Simpson et al. 2001).

An interactive process for designing a platform-based product family was summarized in (Gonzale-Zugasti 2000). Figure 3 shows an overview of the interactive process applied to cellular phone family design. The steps in the product family design process shown in Figure 3 are described in more detail below:

(1) Design requirements and models (e.g. customer requirements, functional requirements, and design constraints). The first step is to construct mathematical models that connect the process models, design choices to the performance indices for products in a family. Design process models are descriptions of the sequence of activities that take place in the design process. They are often drawn in the form of flow diagrams, with feedback showing the iterative returns to the earlier stages. These would include performance, as well as cost models and would also incorporate revenue and competition models in the case of commercial products.

(2) Platform design. With design requirements and models, the design team can create a set of individually designed products as a baseline case against which platform-based variants can be compared. Based on these individually designed products, the representatives from the design team or subsystem experts can explore the commonalities of the design and decide on the common platform. The decision is made based on the similarity of the requirements, the flexibility of the subsystems involved, and other concerns such as availability of resources, manufacturability and assemblability, and schedule constraints.

(3) Variants design. Once a platform is generated, a portion of the design will be handed over to the individual design teams who can complete and optimize the design of their respective products by adjusting the variant variables.

(4) Platform evaluation, re-negotiation, and iteration. The new designs form an alternative product family, which can then be compared to the baseline case of individually designed products or to other platform-based alternatives in terms of technical performance, cost, risk, etc. If the platform-based family is not acceptable, it may be necessary to renegotiate the platform choices and iterate through the design loop to arrive at an adequate family design.

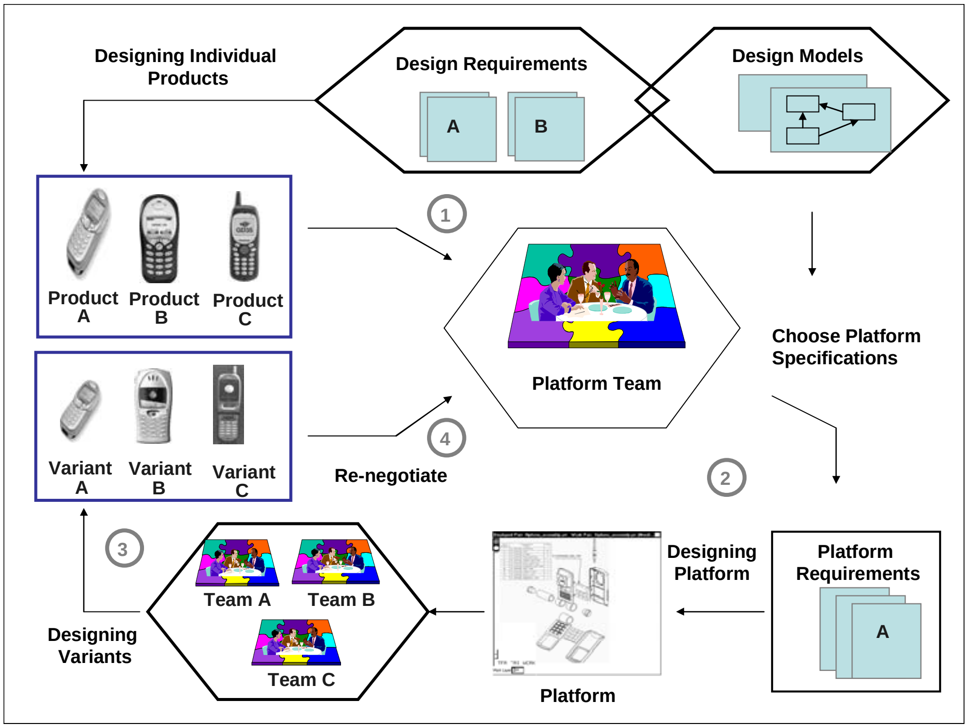


Figure 3: Platform-based product family design implementation

4. Product Platform and Product Family Modeling

Within a platform-based design and development strategy, there are different ways to create a product family: the integral approach and the modular approach. As such, there are two categories of product platforms: integral platform and modular platform. The integral platform is a single, monolithic part of the product shared by all the products in the family. Although it seems to be a restrictive type of platform, real examples exist, such as the telecommunications ground network for interplanetary spacecraft described in (Gonzale-Zugasti 2000). The single common platform is an integral part of each variant and it cannot be replaced by a different piece or module. The modular platform is more general, in which the product is divided into modules that can be swapped by others with different sizes or functionality to create variants. Modular systems provide the ability to achieve product variety through the combination and standardization of components. Within a modular platform, a set of modules is reused across the product family. Companies usually have a set of modules already designed for previous products that could be reused, as well as the resources to design new versions of the same modules or modules with new functionality. In addition, there exists the possibility of purchasing modules from existing catalogs, or even outsourcing the design of new ones.

The modular platform-based product family design and development process advocated in this research generates a re-configurable product platform that can be easily modified and upgraded through the addition, substitution, and exclusion of modules to realize module-based product family. Therefore, the focus of discussion in this section is on modular product family modeling, product platform generation, and product family evaluation. The detailed module-based product family design process will be discussed in the next section.

4.1 Product Family Architecture Modeling

A product family architecture represents the conceptual structure and logical organization of product families from viewpoints of both customers and designers. A well-developed product family architecture can provide a generic architecture to capture and utilize commonality, within which each new product instantiates and extends so as to anchor future designs to a common product line structure. Thus, the modeling and design of product architectures is critical for mass customizing products to meet differentiated market niches and satisfy requirements on local content, component carry-over between generations, recyclability, and other strategic issues.

The modeling and representation scheme used in this research is to combine recent developments in product representation (e.g., Fujita and Ishii 1997; Zha and Du 2001; Rosen 1996) into a hybrid approach. The hybrid approach hierarchically decomposes product families into products or systems, modules, and attributes, as shown in Figure 4. Under this hierarchical representation scheme, product variety is implemented at different levels within the product architecture. Discrete Mathematics and Matrix Algebra are used as a formal foundation for configuration design of modular product architectures. Based on the hybrid representation, a knowledge support product module reasoning system is developed. Details will be discussed below.

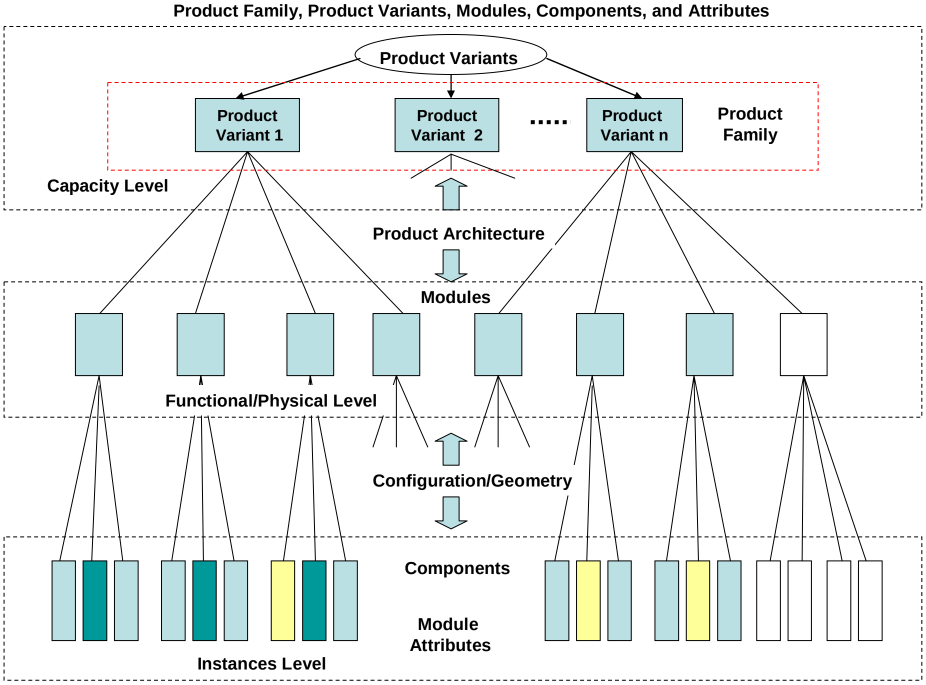


Figure 4: Product families, modules, and attributes

4.2 Product Family Generation and Optimization

Product family is generated through configuration design, in which a family of products can widely vary the selection and assembly of modules or pre-defined building blocks at different levels of abstraction so as to satisfy diverse customer requirements (Tseng and Jiao 1996,1998; Fujita et al. 1998,1999). The essence of configuration design is to synthesize product structures by determining what modules or building blocks are in the product and how they are configured to satisfy a set of requirements and constraints. There are many approaches to address module assembly and configuration design, such as assembly incidence matrix and genetic algorithms (Chen et al. 1999; Zha and Du 2001; Brown 1998; Legar 1999).

In this research, the structured genetic algorithm-based (sGA) (Dasgupta and McGrego 1994) representation and evolutionary design scheme are employed for product family generation through modules configuration, as shown in Figure 5. The sGA product representation uses regulatory genes that act as a switch to turn genes on (active) and off (passive). Each gene in higher levels acts as a switchable pointer that has two possible targets: when the gene is active (on) it points to its lower-level target (gene), and when passive (off) it points to the same-level target. At the evaluation stage only the expressed genes of an individual are translated into the phenotypic functionality, which means that only the genes that are currently active contribute to the product, hence to the fitness of the product. The passive genes do not influence fitness and are carried along as redundant genetic material during the evolutionary process. Therefore, the utilization of the sGA approach to product families can be summarized as follows. 1) Genes represent modules that are either active or passive, depending on whether or not they are part of the product family architecture. 2) A family of products (variants) relies on the addition or subtraction of modules meeting customer requirements. 3) Product variants in the family could be evaluated by alternating different "active" and "passive" modules. A product family would thus correspond to product variants that have different active and passive combinations of modules.

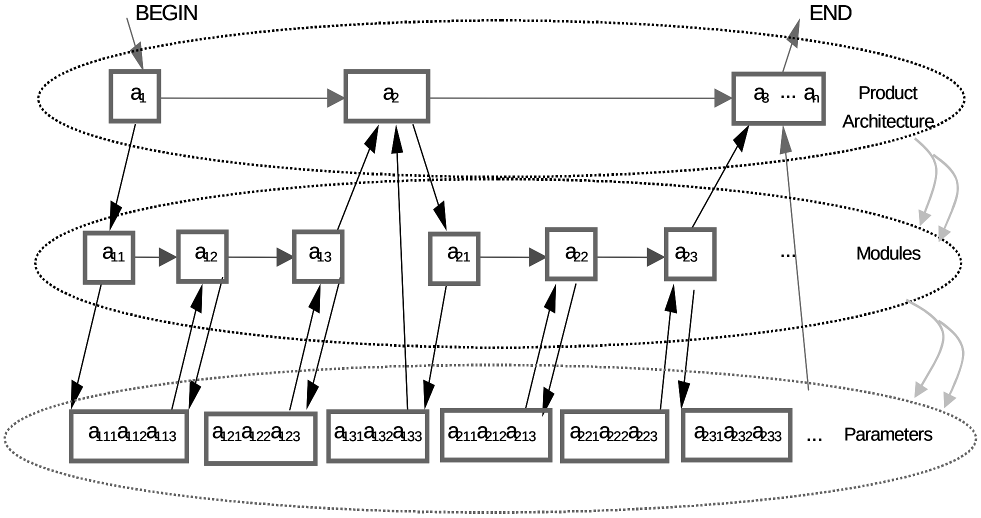
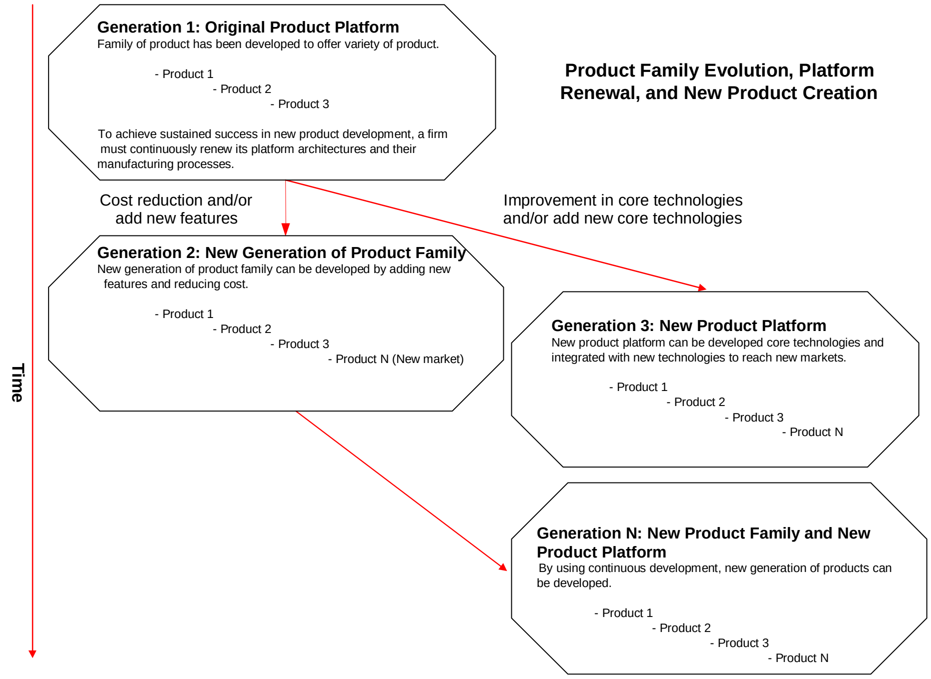


Figure 5: Structured GA for product design implementation



Product Family Evolution, Platform Renewal, and New Product Creation

Generation 1: Original Product Platform

Family of product has been developed to offer variety of product.

- Product 1
- Product 2
- Product 3

To achieve sustained success in new product development, a firm must continuously renew its platform architectures and their manufacturing processes.

Cost reduction and/or add new features

Improvement in core technologies and/or add new core technologies

Generation 2: New Generation of Product Family

New generation of product family can be developed by adding new features and reducing cost.

- Product 1
- Product 2
- Product 3
- Product N (New market)

Generation 3: New Product Platform

New product platform can be developed core technologies and integrated with new technologies to reach new markets.

- Product 1
- Product 2
- Product 3
- Product N

Generation N: New Product Family and New Product Platform

By using continuous development, new generation of products can be developed.

- Product 1
- Product 2
- Product 3
- Product N

Figure 6: Platform extension and evolution

A typical application example can be seen in the modular robot family which is a collection of interchangeable modules (e.g., link and joint units) that can be assembled into many different types and configurations (see Section 6.22). The robot variety (family) is implemented through combining the standardized modules at different levels within the generic product/robot architecture, i.e., a platform to satisfy task/ customer requirements.

4.3 Product Family Evolution Representation

Product family evolution maps or catalogs are intended for strategic management and can be used as mechanisms to generate product variants to form a product family (Wheelwright and Sasser 1989). First, the market segmentation grid is used to facilitate identifying platform leveraging strategies in a product family (Meyer et al. 1997). The major market segment serviced by products is listed horizontally in a market segmentation grid and the vertical axis reflects different tiers of price and performance within each market segment. Similar to the work in (Simpson 1998), the market segmentation grid is applied to identify module-based product platform scaling opportunities from overall design requirements. As a qualitative approach, the beachhead method is most helpful to identify and develop a common platform within a product family. Here, we use the product family map or catalog to trace the evolution of a product family, as shown in Figure 6.

4.4 Product Family Evaluation for Customization

The product customization stage aims at obtaining a feasible architecture of the product family member through reasoning over product family module space according to customer requirements (Meyer et al. 1997). There are two steps involved in this stage. First, customer requirements such as function, assembly, and reuse need to be converted to constraints (Suh 1990). Then, the reasoning is performed at two levels: namely, module and attribute levels, to determine feasible product family member architecture.

In order to evaluate a family of products for mass customization, suitable metrics are needed to assess the appropriateness of a product platform and the corresponding family of products (Krishnan and Gupta 2001). The metrics should also be useful for measuring the various attributes of the product family and assessing a platform's modularity. With respect to the process of a modular platform-based product family design and customization, the evaluation of a product family can be viewed from three different level perspectives: product platform, product family and product variant. The product variant level evaluation is actually the same as or similar to the individual product design evaluation. Various traditional design evaluation approaches are applicable, and the metrics for this level evaluation include cost, time, assemblability, manufacturability, etc. The platform and family level evaluations are focused on the overall benefit of product family development. The metrics at these levels reflect that the main goal of designing products/families is to maximize the benefits to the company. Thus, they can be used to monitor the platform and product family development. It is related to the impact a Research and Development (R&D) project has to platform component revenue and investments into resources. Meyer and Lehnerd (1997) describe several metrics to measure the performance of product families in general: 1) platform efficiency and platform effectiveness to measure R&D performance, focused on platforms and their follow-on product variants within a product family, 2) cycle time efficiency, i.e., elapsed time to develop a derivative product compared with the elapsed time to develop the platform, 3) technological competitive responsiveness, i.e., tracking the degree to which a firm has beaten its competitors to the market place with new features or capabilities in its products, and 4) profit potential, i.e., targeting the profitability of derivative products by examining gross margins.. Other platform-related strategies to minimize product variety are described in (Krishnan and Gupta 2001; Jiao and Tseng 1998; Sanderson 1991). These metrics do not explicitly tell management when to create a new platform. However, they provide a rich context to determine when product platforms should be replaced and what to expect from new products based on these new platforms. In this research, the following two metrics have been used in platform-based family level evaluation (Simpson 1998):

(a) Market efficiency (η_M) embodies a tradeoff between the marketing and the engineering design to provide the least amount of variety so as to satisfy the greatest amount of customers, i.e., target the largest number of market niches with the fewest products.

(b) Investment efficiency (η_I) embodies a tradeoff between the manufacturing and the engineering design to invest a minimal amount of capital into machining and tooling equipment while still being able to produce as large a variety of products as possible.

Therefore, they can be represented by the following two equations, respectively:

$$\eta_M = N_{tm}/N_M \tag{1}$$

$$\eta_I = C_m/N_v \tag{2}$$

where, N_{tm} and N_M are the number of the targetable market niches and the total market numbers, respectively; C_m and N_v are the manufacturing equipment costs and the number of the product varieties, respectively. Of course, a tradeoff also exists between the market efficiency and the investment efficiency as an increase in the investment efficiency through a decrease in product variety can cause a decrease in the market efficiency.

5. Module-Based Product Family Design Process

As shown in Figure 4, product variety can be implemented at different levels within the product architecture. From the aspect of product design, component standardization through a modular architecture has clear advantages in the areas of cost, product performance and product development. Decomposing the problem into modules and defining how modules are related to one another creates the model of a design problem. The module-based product family design process, as shown in Figure 7, is achieved through the following steps (Zha and Lu 2002a,b; Zha and Sriram 2004):

(1) The requirement analysis and modeling for a product (family) is carried out both from the customer and the designer viewpoints using design function deployment (DFD) and Hatley/Pirbhai technique (Sivaloganathan et al 2001; Rushton & Zakarian, 2000). A function-function interaction matrix is generated.

(2) The combination of heuristic and quantitative clustering algorithms is used to modularize the product (family) architecture, and a modularity matrix is constructed.
(3) All modules in the product (family) are identified through the modularity matrix, and the types/functions of all these modules can be further identified according to the module classifications.
(4) The functional modules are mapped to structural modules using the function-structure interaction matrix.
(5) The hierarchical building blocks or design prototypes (Gero 1990) are used to represent the product (family) architecture from both the functional and the structural perspectives (Zha and Du 2001).
(6) The structured genetic algorithm above (sGA) is used to configure and optimize product family architecture to achieve one or multiple main objectives (see Section 4.2). Other design objectives are transformed into constraints for modules or their attributes. In addition, cost and profit models are also built as system constraints.
(7) The product family architecture is rebuilt to form a hierarchical architecture by using the optimized modules from both the functional and structural perspectives.
(8) The product family module space forms a product platform. The product family portfolio is derived from the product family module space.
(9) Standard interfaces are developed to facilitate addition, removal and substitution of modules.
(10) The product family can be generated by module configuration/reconfiguration.
(11) Product variants are evaluated and selected to satisfy the customer requirements.

Therefore, the steps for creating a module-based product family can be outlined as follows: 1) decompose products into their representative functions; 2) develop modules with one-to-one (or many-to-one) correspondence with functions; 3) group common functional modules into a common product platform; and 4) standardize interfaces to facilitate addition, removal and substitution of modules. The module-based product family design process is to develop a re-configurable product platform that can be easily modified and upgraded through the addition, substitution, and exclusion of modules to realize module-based product family.

The fundamental issues underlying the product family design include product information modeling, product family architecture, product platform and variety, modularity and commonality, product family generation, and product assessment and customization. Following the philosophy of the above stages, a modularized approach is proposed for product family design, in which a re-configurable product platform that can be easily modified and upgraded through the addition, substitution, and exclusion of modules is developed. An effective product family platform can allow a variety of derivative products to be created more rapidly and easily, with each product providing the features and functions desired by a particular market segment (Simpson et al. 2001).

Different from the traditional modular design approach, the modular family design process is roughly divided into two main stages: 1) product (family) planning, and 2) family design. It ranges from capturing the voice of customers and market trends for generating product design specifications, formulating a product platform, to customizing products for customers' satisfaction. The product planning stage embeds the voice of customers into the design objective and generates product design specifications. The product family design realizes sufficient product variety: a family of products to satisfy a range of customer demands. The knowledge-supported modular product family design process will be discussed in the next section.

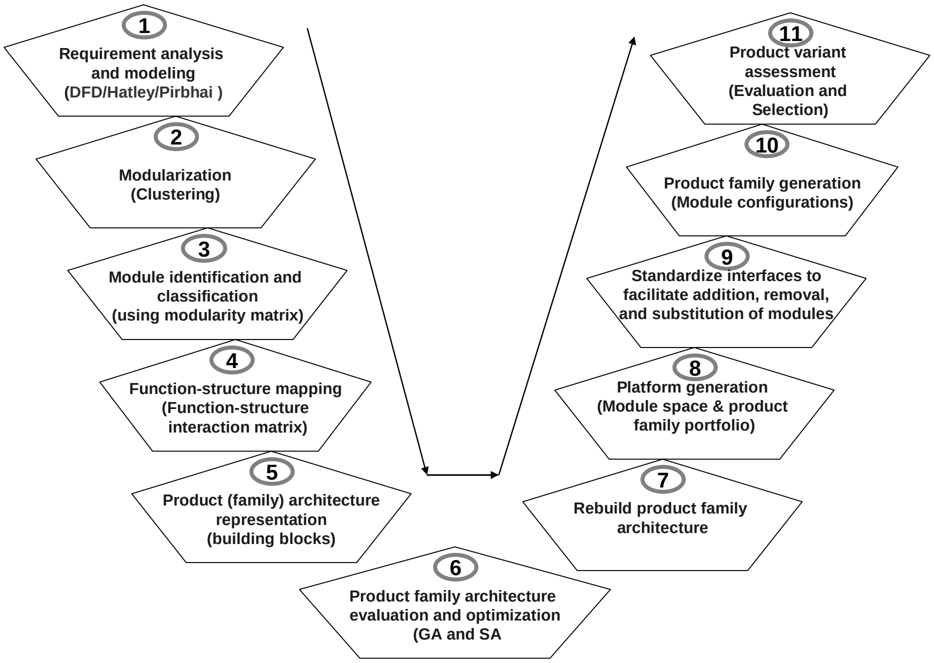

1
Requirement analysis and modeling (DFD/Hatley/Pirbhai)
2
Modularization (Clustering)
3
Module identification and classification (using modularity matrix)
4
Function-structure mapping (Function-structure interaction matrix)
5
Product (family) architecture representation (building blocks)
6
Product family architecture evaluation and optimization (GA and SA
7
Rebuild product family architecture
8
Platform generation (Module space & product family portfolio)
9
Standardize interfaces to facilitate addition, removal, and substitution of modules
10
Product family generation (Module configurations)
11
Product variant assessment (Evaluation and Selection)

Figure 7: Module-based product family design for mass customization

6. Knowledge Support Framework for Modular Product Family Design

Design process is knowledge-intensive as there is a large amount of knowledge that designers call upon and use to match the ever-increasing complexity of design problems. Given that even the most routine of design tasks is dependent upon vast amount of expert design knowledge, there is a need for some sort of knowledge support. Design knowledge refers to the collection of knowledge needed to support the design activities and decision-making in the design process. Successfully capturing design knowledge, effectively representing it and easily accessing it are crucial to increase the design "science" contents compared to the "art" nature for the product family design process. The main characteristics of product family design are modularity, commonality/reusability, and standardization. Designing product families requires knowledge defining their characteristics. Details are discussed below.

6.1 Knowledge Support Scheme and Key Issues

Once the concepts of product platform, product family architecture, modular design, etc. are established, a representation or modeling scheme is needed to model product families. Existing representation/modeling schemes for product families vary in the literature, including two types of representational models: product family architecture and product family evolution. These models are related to the formulation of the platform for product family generation and play crucial roles in the down stream stages such as product family evaluation, assessment, and product customization. The fundamental issues underlying the product family design process include product information modeling, product family architecture, product platform, product family derivation and generation, and product family evolution and assessment.

Based on the modular family design approach discussed above, we develop a knowledge intensive support framework for modular platform-based product family design, as illustrated in Figure 8. Design knowledge is classified into two categories: product family information and knowledge, and family design process knowledge. These two categories of knowledge are utilized to support two main stages, product (family) planning and family design, in the whole process of modular product family design. How knowledge is modeled and supports the modular product family design process will be discussed below. The knowledge support for product family planning stage assists the designer to capture the voice of customers and market trends, and embed them into the design objective for generating product design specifications (PDS) and customizing products for customers' satisfaction. The knowledge support for product family design stage assists designers to realize sufficient product variety — a family of products to satisfy a range of customer demands.

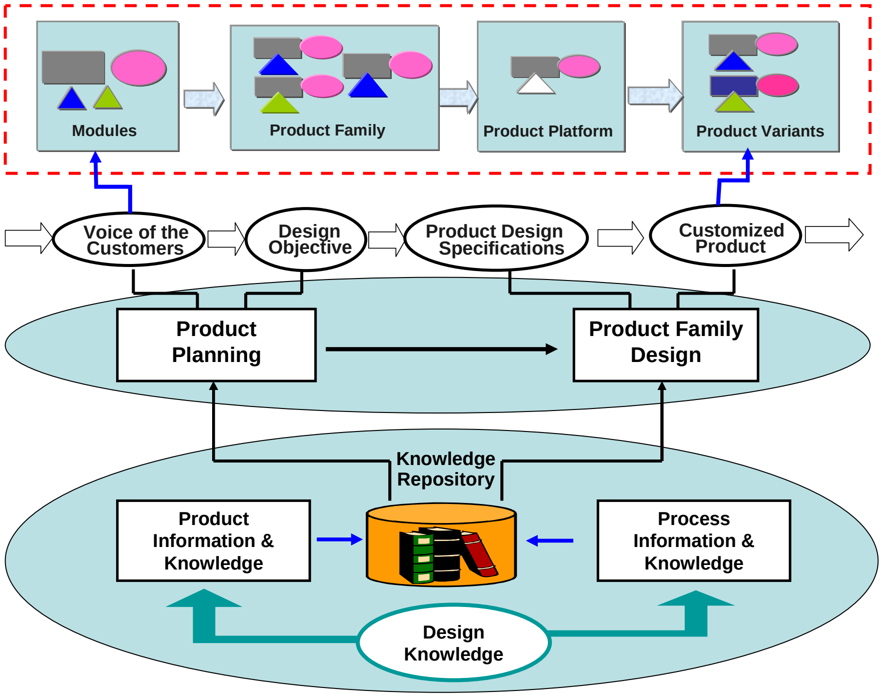


Figure 8: Knowledge support framework for module-based product family design

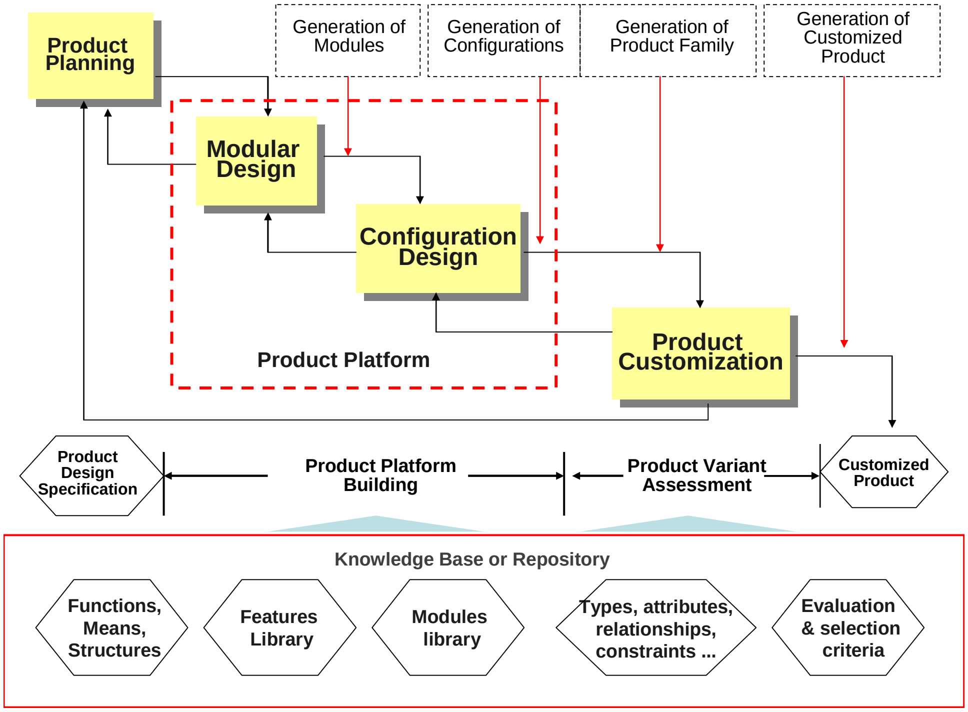

Product Planning
Generation of Modules
Generation of Configurations
Generation of Product Family
Generation of Customized Product
Modular Design
Configuration Design
Product Customization
Product Platform
Product Design Specification
Product Platform Building
Product Variant Assessment
Customized Product
Knowledge Base or Repository
Functions, Means, Structures
Features Library
Modules library
Types, attributes, relationships, constraints ...
Evaluation & selection criteria

Figure 9: Knowledge support scheme for the modular product family design process

With the understanding of the fundamental issues of product family design, a more detailed knowledge support scheme shown in Figure 9 is adopted in customer requirements modeling, product architecture modeling, product platform establishment, product family generation, and product assessment. The modular product family design process is roughly divided into three main stages: product platform building, product family derivation/generation, and product variant assessment, and is implemented through product planning for design specifications (e.g. function requirements and design constraints) generation, modular design, and configuration design and product assessment. Therefore, the key research issues for the knowledge support scheme can be summarized as follows:

(1) Product family design information and knowledge modeling: design knowledge capturing, classification, representation, and organization and management;
(2) Product architecture modeling: representing product variety, component modularization and standardization, product management, etc;
(3) Product platform establishment: exploring methods for feature-based module design and configuration design;
(4) Product family generation: generating product variants or family members; and
(5) Product assessment: evaluating product variants.

Each of the above issues has many detailed sub-issues to be addressed. The challenging ones are the product/family architecture representation and product platform establishment, which are related to the modeling of product architecture, product platform generation, and the process from the product architecture modeling to the product platform generation. The product family architecture should represent the conceptual structure and logical organization of product families from viewpoints of both customers and designers (engineering related). A well-developed product family architecture can provide a generic architecture to capture and utilize commonality, within which each new product expands so as to anchor future designs to a common product line structure.

6.2 Product Family Design Knowledge Modeling and Support

Based on the above described knowledge support scheme, the implementation of knowledge-supported module-based product family design can be achieved through two steps: 1) knowledge modeling; 2) the knowledge support process, which are discussed in this section.

6.2.1 Issues of Product Family Design Knowledge Modeling

The complexity and diversity of engineering knowledge results in high demands for knowledge modeling in engineering: the many different aspects and their relationships have to be described in a complete, consistent, coherent, and concise way. Even if we assume that the corresponding advanced knowledge processing capabilities exist, adequate modeling of engineering knowledge provides a challenge. The object-oriented representation, STEP (Standard for the Exchange of Product Model Data, officially ISO 10303) and UML/SysML (Unified Modeling Language) provide to some extent expressiveness and formal rigor as platforms for knowledge modeling in product family design. CommonKADS (http://www.commonkads.uva.nl/) as a dedicated knowledge oriented approach can be seen as a powerful framework for knowledge modeling in general, but in its current concrete form it is not expressive and differentiated enough to fulfill the high knowledge modeling demands in engineering (Sivard 2000).

Product design knowledge is a collection of data/information and knowledge needed to support the design activities and decision-making in the product/family design process. It may include all information defined and created during the design process and all knowledge used to create that information. The former is often defined as product knowledge, which includes all product or artifact-related information needed throughout the whole design process such as product specifications, concepts, structure and geometry. The latter is referred to as the process knowledge, which can be described in two aspects: design activities/tasks and design rationale. Design knowledge modeling is to capture, represent, organize and manage design knowledge in the design process. Further, the knowledge modeling process for product/family design is to elicit design knowledge in product family design and establish a comprehensive knowledge base or repository that can be retrieved and reused when necessary. The aspects related to product family design knowledge modeling are shown in Figure 10, which include design knowledge capture, classification, representation, organization and management.

The approach to modeling information and knowledge for product family architecture and platform in terms of the semantics used in platform product development is illustrated in Figure 11. The product structure and components of the generic information platform are represented in the physical domain of axiomatic design and configuration rules and mappings are represented as constraints and mappings between the functional, physical, and process domains (Sivard 2000). This model is adapted to the STEP-based product-modeling standard thereby create a standardized information platform covering the reasoning of development. It contains modeling constructs for representing alternatives, configuration rules and many other aspects of product platforms. The purpose to adapt the conceptual model to a standard is twofold: 1) the standard provides functionality and detailed information models; 2) a standard format supports the exchange of information between applications and users. With help of the product platform, customers' requirements are satisfied either by standard models or customer models configured from standard or custom modules and/or components in knowledge-based configuration systems.

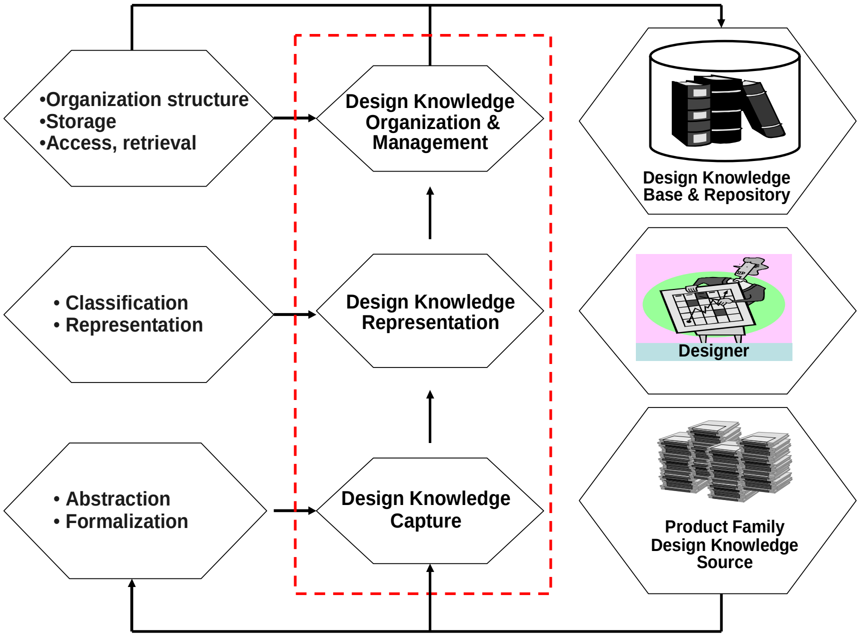

•Organization structure
•Storage
•Access, retrieval
Design Knowledge Organization & Management
Design Knowledge Base & Repository
• Classification
• Representation
Design Knowledge Representation
Designer
• Abstraction
• Formalization
Design Knowledge Capture
Product Family Design Knowledge Source

Figure 10: Knowledge modeling in product family design

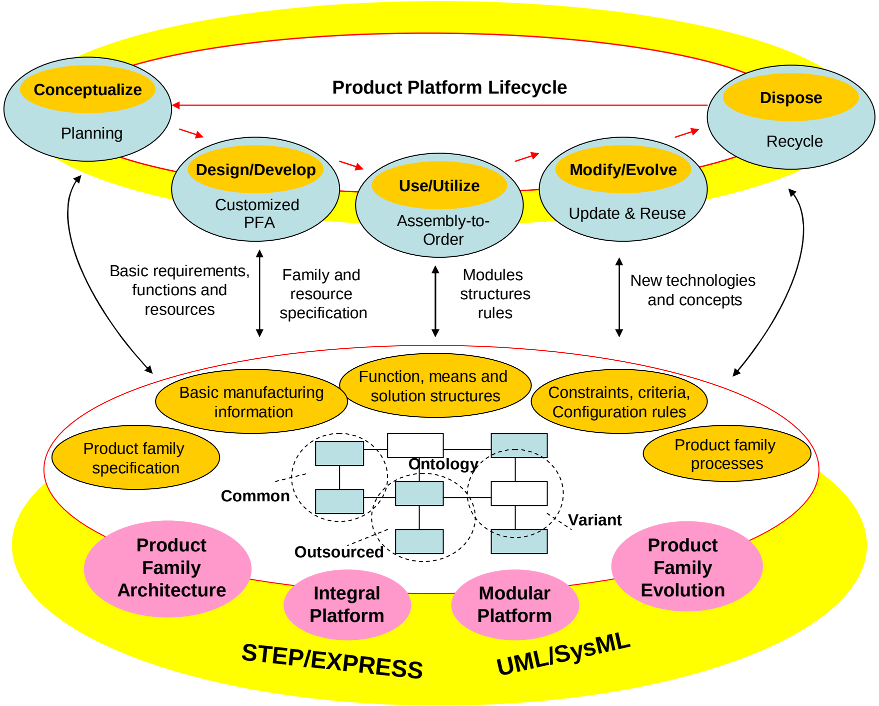
6.2.2 Knowledge Modeling and Representation for Product Family Design

Product family design starts from a set of customer/functional requirements of single product. The requirements are implemented by a set of modules described in terms of design variables of the product principle. These design variables of a module propagate to the functional requirements on the lower level elements of the module, so on and so forth until to all the modules and element are specified. With respect to the modular product family design process, three groups of knowledge are required (Du et al. 2001, 2002, Sivard 2000): 1) How to deploy the functions of the products (module) to lower level modules; 2) How to select the solutions among the standard modules or the custom modules; and 3) After being selected, all solutions are configured to be an end product (product variant). The performance of each of them is then estimated to help both the designer and the customer to make decision.

The product family design knowledge should be abstracted and classified into different categories, e.g., off-line and on-line, product data/information and design process knowledge, through analysis of product-family design process. Different categories of product family design knowledge are represented in different ways from multiple views of product/family design process. Since design knowledge includes all product data/information needed throughout the whole family design process, a new product data/information model must be employed, which may include customer / task requirements, design specifications, functions–behaviors, structures, assemblies, performance constraints/metrics, etc.

```
Product Definition:
                  Customer/Task Requirements;
                  Specifications;
                  Artifacts;
                  Features;
                  Functions-Behaviors-Structures;
                  Performance Objectives and Constraints;
                  Relationships;
Product Variety:
                  Assembly Structure;
                  Module Details;
                  Variety Parameters;
```

As shown in Figure 9, the product repository may be extensively composed of functions, means, structures, features library, modules library, types, attributes, relationships, rules, constraints, evaluation/selection criteria, etc. In practice, an effective way to create a product data/information representation model is to integrate the database representation model and the design process model. Such a data/information model still needs to be divided into two parts: one for modules and the other for module assemblies. The module representations follow the object-based formalism, while the module assemblies are based on the graph theory and its incident matrix representation as well as sGA. Following the requirements of designing product families with a high degree of commonality as well as designing several products (variants) around reusable components, the two main elements of the architecture are: 1) generic product specifications and 2) reusable solution libraries. Product family architectures and component architectures are treated in a similar way, enabling a hierarchical structure of structures. Thus, families of components may be selected from the solution library and integrated into the framework.



Product Platform Lifecycle
Conceptualize
Planning
Dispose
Recycle
Design/Develop
Customized PFA
Use/Utilize
Assembly-to-Order
Modify/Evolve
Update & Reuse
Basic requirements, functions and resources
Family and resource specification
Modules structures rules
New technologies and concepts
Basic manufacturing information
Function, means and solution structures
Constraints, criteria, Configuration rules
Product family specification
Product family processes
Ontology
Common
Variant
Outsourced
Product Family Architecture
Integral Platform
Modular Platform
Product Family Evolution
STEP/EXPRESS
UML/SysML

Figure 11: Generic platform information model (Ref: Sivard 2000)

Figure 12 illustrates the construction process (Steps 1-4) of product platform and the reuse for domain-specific applications (Step 5). Therefore, a multi-level hybrid representation schema (meta level, conceptual level, instance level, physical level, geometric level) is adopted to represent the product design process knowledge in different design stages at different levels, based on a combination of elements of semantic relationships with the object-oriented data model. To effectively manage and utilize design knowledge, a generalized design knowledge matrix is proposed, in which all tasks in the design process are listed in column while all information and design knowledge is categorized in rows. The contents of design knowledge for each task are recorded in the corresponding cells of the matrix with appropriate representations.

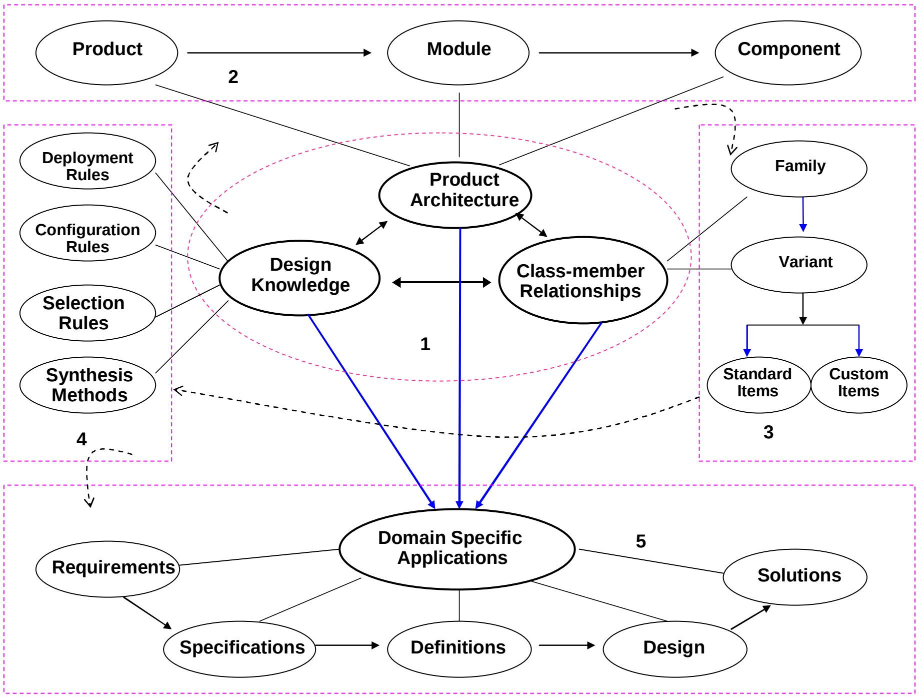


Figure 12: Modular product platform construction process, representation and reuse

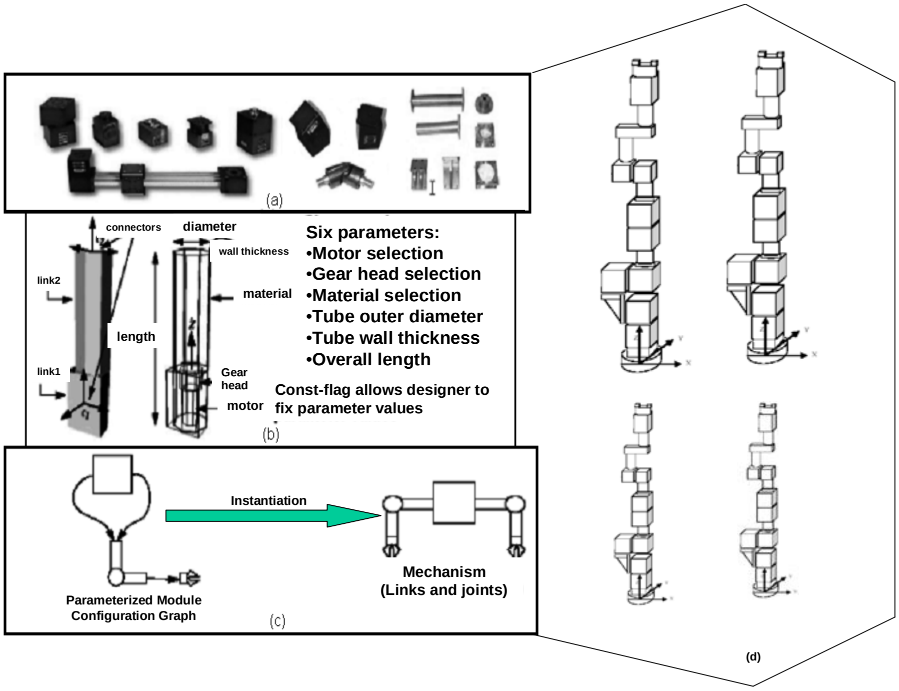


Figure 13: (a) Robot modules and library, (b) module attribute parameterization, (c) assembly / configuration, and (d) robot families

More specifically, the object-oriented knowledge representation is based on a mixed representation method and object-oriented programming (OOP) techniques, and allows designers to look at the design problem as a collection of objects or sub-problems linked together by rules. Thus it provides the designers with an expressive power to represent complex problems or information in an effective manner. If a designer can break the design problem into the form of well-defined, clearly manipulative chunks with their own self-containing information, which is interrelated through a series of rules and constraints, then the problem can be easily solved. The basic structure of this representation is described as a module. The class of an object and its instances are described by the module structure. An object-oriented module is composed of four types of slots, which are the attribute slot, relation slot, method slot and rule slot as follows:

(1) The attribute slots are used for describing the static attributes (variables) of design object.

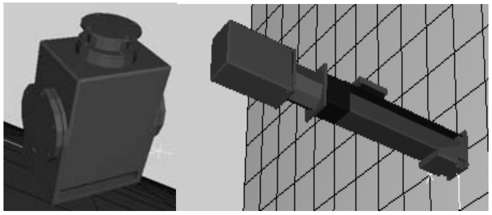
(2) The relation slot is used for describing the static relations among objects. With the help of the relation slot and according to the relation of classification, the design object can be described as a hierarchical structure. Its classes and subclasses can share the knowledge in super class. The messages that control the design process can be sent among all instances of objects. In addition, if needed, other kind of relation slots can be defined, such as the resolution, position and assembly, etc. These slots create the foundation for describing a graph in design. The hierarchical structure of object oriented knowledge representation is formed.

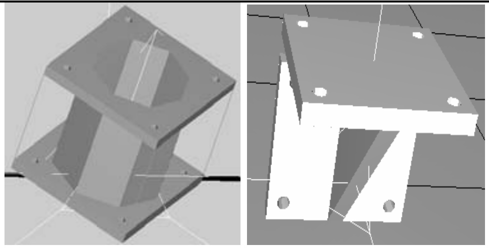
(3) The method slot is used for storing the methods of design, sending messages and performing procedural control and numerical calculation.

(4) The rule slot is used for storing sets of production rules. The production rules can be classified according to the differences among objects being treated and stored respectively in rule slots in the form of slot value.

Thus, the integrated knowledge representation scheme realizes the advantages of both object-oriented representation and rule-based representation. For illustration, an object-oriented representation instance for a robot family and its parameterized module information (e.g. link and joint modules) are described as in Table 1. For the robot manipulation system itself, the modular design is introduced, i.e., it is a collection of interchangeable modules (e.g., link and joint units) that can be assembled into many different types and configurations, as shown in Figure 13. The robot variety (family) is generated through combining the standardized modules at different levels within the generic robot architecture, i.e., a robot platform to satisfy task/ customer requirements.

Table 1: Link and joint modules representation for the robot family

Module ("Joint Module") (
Number of degree of freedoms (DOFs)→ [1, 2, 3];
Motion type→ ["translation", "rotation"];
Active attributes → ["passive", "active"];
Torque ranges→ [force, torque];
Connected module types →["link", "joint", "other"];
Motion range→ [disp.(S), vel.(V), accel.(A)];
Adjustable parameter→ [initial poses];
Assembly pattern→ [no., input/output ports];
Dimension parameters→ [len.(L),wid.(W), heigh.(H)];
Dynamic parameters→ [mass, center of mass, inertial];
…
)
	

Module ("Link Module") (
Connected module types →[link, joint];
Isomorphic assembly pattern →[no., input/output ports];
Fixed dimensions→ [displacement, orientation];
Changeable parameters→ [displacement, orientation];
Dynamic parameters→ [mass, center of mass, inertial];
…
)
	

The model presented above is being incorporated and fit into the core product model (Fenves 2001, Fenves et al. 2005) and the product family evolution model (Wang et al. 2003) developed recently at US National Institute of Standards and Technology. In this connection, we define a platform artifact represented by **PlatformArtifact** class in the **Platform** package. **PlatformArtifact** is subclass of **Artifact** class in the NIST CPM package. Information about differences between the family members can be used for the development of an extensible architecture of the common core assets, and two processes may be involved: 1) platform/ family construction; and 2) platform/family evolution. **FCMArtifact** and **FEMArtifact** are subclasses of **PlatformArtifact** and **Artifact**, representing product platform (family) to be constructed and evolved. These packages can support family-based design for mass customization, including family construction model (**FCM**) and family evolution model (**FEM**). More details on the information models for platform-based product design and development will be presented in a separate paper.

6.2.3 Knowledge Support Process for Modular Product Family Design

Once the design knowledge repository is built up, the user or designer can utilize the knowledge in it to solve problems in product family design. As discussed in Sections 3, 4 and 5 above, the whole design process is roughly divided into two main stages: product platform formulation for family generation and product evaluation or assessment for mass customization. Incorporating the detailed steps in Section 5, the whole knowledge supported modular product family design process can be fulfilled.

The knowledge support process in product design evaluation for mass customization experiences the elimination of unacceptable alternatives, the evaluation of candidates, and the final decision-making under the customers' requirements and design constraints (Zha et al. 2004). With respect to the traditional approach for product evaluation (Pahl and Beitz 1996), the knowledge resources utilized in the process include differentiating features, customers' requirements, preferences and importance (weights), trade-offs (e.g., market vs investment), assemblability and manufacturablity, and utilities functions, and heuristic knowledge (e.g., production rules), etc.

In applying the above knowledge support scheme for modular product family design, the following points should be noted:

(1) System requirement modeling and analysis should be the first step in development of a modular product family.
(2) Development of a modular product family is a complex task. A systematic and structured approach is a mandatory.
(3) Functional analysis is best suited for developing a new product family, rather than modifying existing ones.
(4) Large complex products or systems have a considerable amount of constraints that limit the design of modular product families.

7. Prototype of Knowledge-Intensive Support System for Product Family Design

A prototype knowledge support system is developed to assist the designer in the product family design process. Figure 14 shows a web client /server implementation architecture of the system. As shown in Figure 14, the web-based design framework uses the design with modules, modules network, and knowledge server paradigms. The knowledge-intensive support system can thus exploit the modularity of knowledge-based systems, in that the inference engine and knowledge bases are located on server and the user interface is exported on demand to client computers via network connections. Therefore, modules under the knowledge support framework are connected together so that they can exchange services to form large collaborative integrated models. The module structure leads itself to a client (browser) / knowledge-server-oriented architecture using distributed object technology. Knowledge server systems utilize the connectivity provided by the Internet to increase the size of the user base while minimizing distribution and maintenance overheads.

The implementation of a knowledge-intensive support system uses three-tiered client/knowledge server architecture to support collaborative design interactions with a web-browser-based graphical user interface (GUI). The underlying framework and the knowledge engine are written in Java^{TM}, integrated with Java Expert System Shell, Jess/FuzzyJess (Ernest 1999, NRCC 2003). It also integrates with existing application packages such as CAD and database applications. CORBA serves as an information and service exchange infrastructure above the computer network layer and provides the capability to interact with existing CAD applications and database management systems through other Object Request Brokers (ORB). In turn, the framework provides the methods and interfaces needed for the interaction with other modules in the networked environment.

Based on the architecture of the knowledge support system, its functionality is achieved through implementing the following subsystems: web GUI, knowledge repository and advisory system for modular product family design. The knowledge repository is able to capture, store and retrieve design knowledge, including customer requirements, design objectives, design modules, design rationales, evaluation criteria and product varieties. The modular design advisory system (Design Advisor) includes a decision-making mechanism and a product module-reasoning engine. The knowledge supported product module reasoning engine is developed to reason about sets of product architectures, to translate design requirements into constraints on these sets, to compare architecture modules from different viewpoints and to enumerate all feasible modules using the "generate-and-test " or heuristic approaches. The web GUI provides users with the following abilities:

(1) examine the customers' requirements and the configuration of design problem models,
(2) generate a product platform (a set of common modules),
(3) analyze tradeoffs and varieties by modifying design parameters within modules,
(4) search for product alternatives in a product family , and
(5) select the final solutions.

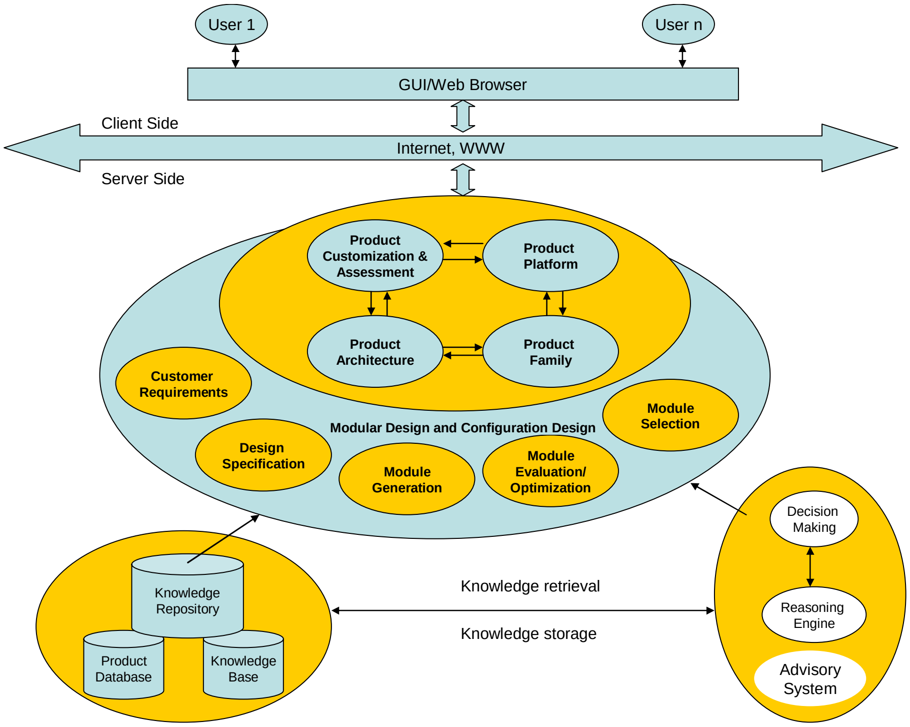
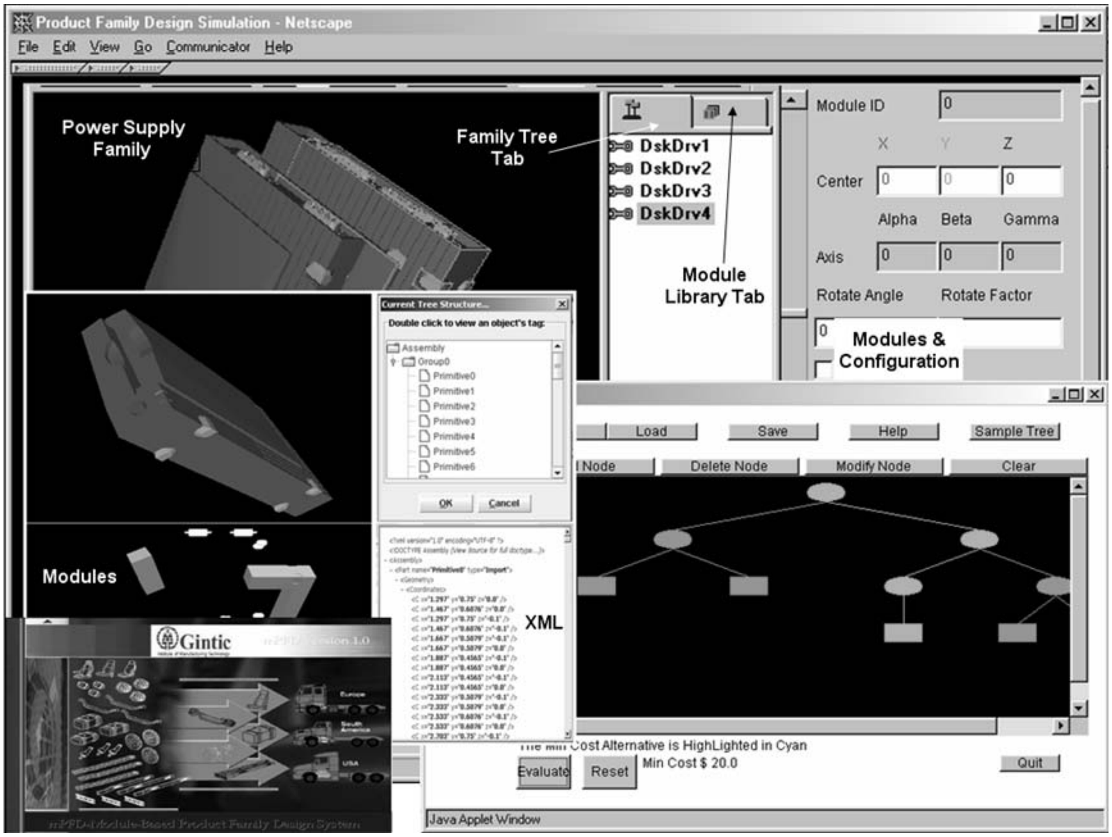


Figure 14: Web-enabled knowledge support system architecture

The web GUI is a pure client of a knowledge server, delegating all events to the associated server. For wide accessibility and interoperability, the GUI is implemented as a web-browser-based client application. The front-end side of the application is implemented as a combination of XML (eXtension Markup Language) documents, VRML (Virtual Reality Modeling Language) and Java applets. The back-end side system components include a knowledge repository, modular design server, product family generation server, product evaluation server, models and modules base server, CAD and graphics server, and a database server, and knowledge assistant and inter-server communications explanation facilities (Siegel 1996; IONA 1997). The commercial ORB implementation of Java applets (OrbixWeb^{TM}) is employed for the CORBA-based remote communication between the GUI Java applets and the back-end side system components.

The Design Advisor system, consisting of a cluster analysis module, ranking module, selection module, neural-fuzzy module, and visualization and explanation facilities, was developed in (Zha et al. 2004). The current capabilities of the prototype include capturing and browsing of the evolution of product families and of product variant configurations in product families, ranking and evaluation, and selection of product variants in a product family. The comprehensive fuzzy decision support system can visualize and explain the reasoning process and makes difference between the knowledge support system and the traditional program. With this subsystem, the designer can represent the design choices available as a fuzzy AND/OR tree. The fuzzy clustering and ranking algorithms employed in it are able to evaluate and select the (near) overall optimal design that best satisfies customer requirements. The selected design choice is highlighted in the represented tree. Figure 15 gives a screen snapshot of the prototype system used for product family design (e.g., car, power supply, robot, etc.). When fully developed, the knowledge-intensive support system for product family design can result in the following benefits:

(1) capture and manage design information and knowledge (e.g., know-how), retrieve previous knowledge;
(2) provide real-time information and knowledge services to help or assist designers in family-based product design;
(3) support communication and collaborative teamwork by sharing the most up-to-date design information and knowledge;
(4) reduce product development cycle time and lower total cost;
(5) improve customer satisfaction; and
(6) improve the competitiveness and sales of a company.



Module ID 0

Center X 0, Y 0, Z 0

Axis Alpha 0, Beta 0, Gamma 0

The min Cost Alternative is HighLighted in Cyan

Min Cost $ 20.0

Figure 15: Screen snapshot of product family design

8. Summary and Future Work

This paper presented a knowledge intensive support methodology and framework for platform-based product family design and development. An integrated modular platform based product family design scheme is proposed with knowledge support for customer requirements' modeling, product architecture modeling, product platform establishment, product family generation, and product variant assessment. The developed methodology and framework can be used for capturing, representing, and managing product family design knowledge and offer support in the design process. Finally the issues related to the implementation of the knowledge support framework are addressed. The system implementation architecture and functionality are provided to support platform-based product family design and development. When fully developed, the system can support product family design effectively and efficiently and improve customer satisfaction. Future work is required to further develop a web-based knowledge repository and design support system for module-based product family design and collaborative platform product development.

Disclaimer

Commercial equipment and software, many of which are either registered or trademarked, are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose. Part of the work was done while the first author was at Singapore Institute of Manufacturing Technology, Singapore.

References

1. Agarwal, M and Cagan, J., 1998, A Blend of Different Tastes: the Language of Coffeemakers," *Environment and Planning B: Planning and Design*, 25(2): 205-226.

2. Brown,D. C., 1998, “Defining Configuring,” http://www.cs.wpi.edu/~dcb/Config/EdamConfig.html, AI EDAM special issue on Configuration.

3. Chang T-S and Ward A. C., 1995, “Design-In-Modularity with Conceptual Robustness,” *Design Technical Conference ASME* 1995, DE-Vol. 82.

4. Chen, I.M., Yeo, S.H., Chen, G., and Yang, G.L., Kernel for Modular Robot Applications: Automatic Modeling Techniques, *Int. J. Robotics Research*, pp.225-242, 1999

5. Chen, W., Allen, J. K., Mavris D. and Mistree, F., 1996, “A Concept Exploration Method for Determining Robust Top-Level Specifications,” *Engineering Optimization*, vol. 26: pp. 137-158.

6. Chen, W., Rosen D., Allen J. and Mistree, F., 1994, “Modularity and the Independence of Functional Requirements in Designing Complex Systems,” *Concurrent Product Design*, vol. 74: pp. 31-38.

7. Collier, D. A., 1981, “The Measurement and Operating Benefits of Component Part Commonality,” *Decision Science*s, vol. 12(1): pp. 85-96.

8. Collier, D. A., 1982, “Aggregate Safety Stock Levels and Component Part Commonality,” *Management Science*, vol. 28(22): pp. 1296-1303.

9. Cho, J.R., 2000, Product Structuring for Customer, Assembly and Maintenance, Assembly Automation Lab., Industrial Engineering, Pusan National University, Korea.

10. Dasgupta, D. and McGregor, D. R., 1994, “A More Biologically Motivated Genetic Algorithm: The Model and Some Results,” *Cybernetics and Systems: An International Journal*, vol. 25: pp. 447-469.

11. Du, X.H., Jiao, J.X., Tseng, M.M, 2001, Product Platform Planning for Mass Customization, Department of Industrial Engineering & Engineering Management, The Hong Kong University of Science and Technology, Hong Kong.

12. Erens, F. and Verhulst, K., 1997, “Architectures for Product Families,” *Computers in Industry*, vol. 33 (2-3): pp. 165-178.

13. Ernest J. Friedman-Hill, The Java Expert System Shell, http:// herzberg.ca.sandia.gov /jess, *Sandia National Laboratories*, USA, 1999.

14. Fenves, S. J., 2001, “A Core Product Model for Representing Design Information,” NISTIR 6736, NIST, Gaithersburg, MD.

15. Fenves, S., Foufou, S., Bock, C., Bouillon, N., and Sriram, R. D., *CPM2: A Revised Core Product Model for Representing Design Information* , National Institute of Standards and Technology, NISTIR 7185, Gaithersburg, MD 20899, USA, 2005.

16. Finch, W.W., 1977, Predicate Logic Representations for Design Constraints on Uncertainty Supporting the Set-Based Design Paradigm, Ph.D Thesis, The University of Michigan, Ann Arbor.

17. Fujita, K., 2000, “Product Variety Optimization under Modular Architecture,” *Proceedings of Third International Symposium on Tools and Methods of Competitive Engineering* (TMCE2000), pp. 451-464.

18. Fujita, K., Sakaguchi, H. and Akagi, S., 1999, “Product Variety Deployment and its Optimization under Modular Architecture and Module Commonalization,” *Proceedings of the 1999 ASME Design Engineering Technical Conferences*, Paper No. DETC99/DFM-8923, ASME.

19. Fujita, K., Akagi. S., Yoneda, T. and Ishikawa, M., 1998, “Simultaneous Optimization of Product Family Sharing System Structure and Configuration,” *Proceedings of the 1998 ASME Design Engineering Technical Conferences*, Paper No. DETC98/DFM-5722, ASME.

20. Fujita, K. and Ishii, K., 1997, “Task Structuring Toward Computational Approaches to Product Variety Design,” *Proceedings of the 1997 ASME Design Engineering Technical Conferences*, Paper No. 97DETC/DAC-3766, ASME.

21. Gaithen, N., 1980, *Production and Operations Management: A Problem-Solving and Decision-Making Approach*, The Dryden Press, New York.

22. Gero, J.S., 1990, “Design Prototypes: A Knowledge Representation Schema for Design,” *AI Magazine* 11(4): 26-36.

23. Goldberg, D. E., 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Inc., New York.

24. Gonzale-Zugasti, J.P., 2000, Models for Platform-Based Product Family Design, PhD Thesis, MIT, Cambridge.

25. Gorti, S.R., Gupta, A., Kim, G.J., Sriram, R.D., and Wong, A., 1998, “An Object-Oriented Representation for Product and Design Process,” Computer-Aided Design, Vol.30, No.7, pp.489-501.

26. Gu, P. and Sosale, S., 1999, “Product Modularization for Life Cycle Engineering,” *Robotics and Computer-Integrated Manufacturing*, vol. 15(5): pp. 387-401.

27. Ishii, K., Juengel, C. and Eubanks, F., 1995, “Design for Product Variety: Key to Product Line Structuring,” *ASME Design Theory and Methodology Conference*, Boston, MA, DE-Vol. 83: pp. 499-506.

28. IONA, Orbix2 Programming Guide: IONA Technologies Ltd., 1997.

29. Jeffrey B. D., Gonzalez-Zugasti, J.P. and Otto, K.N. 2001, “Modular Product Architecture,” *Design Studies*, vol. 22(5): pp. 409-424.

30. Jiao, J.X., Tseng, M.M., Ma Q. and Zou, Y., 2000, “Generic Bill of Materials and Operations for High-Variety Production Management,” *Concurrent Engineering: Research and Application*, Vol. 8, No. 4, pp. 297-322.

31. Krishnan, V. and Gupta, S., 2001, “Appropriateness and Impact of Platform-based Product Development,” *Management Science*, 47(1): pp.52-68.

32. Kotler, P., 1989, “From Mass Marketing to Mass Customization,” *Planning Review*, Vol. 17(5): pp. 10-15.

33. Kusiak, A. and Huang, C.C., 1996, “Development of Modular Products,” *IEEE Trans. on Components, Packaging, and Manufacturing Technology*, Part-A, vol. 19(4): pp.523-538.

34. Lee, H. L. and Tang, C. S., 1997, “Modeling the Costs and Benefits of Delayed Product Differentiation,” *Management Science*, vol. 43(1): pp. 40-53.

35. Leger, Chris, Automated Synthesis and Optimization of Robot Configurations: An Evolutionary Approach, *PhD Thesis*, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1999.

36. Martin, M. and Ishii, K., 1996, “Design for Variety: A Methodology for Understanding the Costs of Product Proliferation,” *1996 Design Theory and Methodology Conference* (Wood, K., ed.), Irvine, CA, ASME, Paper No. 96-DETC/DTM-1610.

37. McDermott, C. M. and Stock, G. N., 1994, “The Use of Common Parts and Designs in High-Tech Industries: A Strategic Approach,” *Production and Inventory Management Journal*, Vol.35 (3): pp. 65-68.

38. McKay, A., Erens, F. and Bloor, M. S., 1996, “Relating Product Definition and Product Variety,” *Research in Engineering Design*, vol. 8 (2): pp. 63-80.

39. Meyer, M. H., 1997, “Revitalize Your Product Lines through Continuous Platform Renewal,” *Research Technology Management*, vol. 40(2): pp. 17-28.

40. Meyer, M. H. and Utterback, J. M., 1993, “The Product Family and the Dynamics of Core Capability,” *Sloan Management Review*, vol. 34 (Spring): pp. 29-47.

41. Meyer, M. H., Tertzakian, P. and Utterback, J. M., 1997, “Metrics for Managing Research and Development in the Context of the Product Family,” *Management Science*, vol. 43(1): pp.88-111.

42. Meyer, M.H. and Lehnerd, A.P., 1997, The Power of Product Platforms, New York: The Free Press.

43. NRCC (Nationa Research Council of Canada), Fuzzy Logic in Integrated Reasoning, webpage: http://www.iit.nrc.ca/IR_public/fuzzy/, 2003

44. Nutt, G.J., 1992, *Open Systems*, Prentice Hall, Englewood Cliffs, NJ.

45. Pahl, G. and Beitz, W., 1996, Engineering Design - A Systematic Approach, New York: Springer.

46. Pfaltz, J. L. and A. Rosenfeld, 1969, “Web Grammars,” *Proceedings of First International Joint Conference on Artificial Intelligence*, Washington, D.C. pp. 609-619.

47. Pine, B. J., 1993, *Mass Customization - The New Frontier in Business Competition*, Boston, MA, Harvard Business School Press.

48. Paredis, C.J.J., An Agent-Based Approach to the Design of Rapidly Deployable Fault Tolerant Manipulators, *Ph.D. Thesis*, Carnegie Mellon University, Pittsburgh, 1996.

49. Rosen, D. W., 1996, “Design of Modular Product Architectures in Discrete Design Spaces Subject to Life Cycle Issues,” *1996 ASME Design Automation Conference*, Irvine, CA. 96-DETC/DAC-1485.

50. Reddy, G. and Cagan,J., 1995, “An Improved Shape Annealing Algorithm for Truss Topology Generation,” *ASME Journal of Mechanical Design*, vol. 117: pp. 315-21.

51. Rothwell, R. and Gardiner, P., 1990, “Robustness and Product Design Families,” *Design Management: A Handbook of Issues and Methods* (Oakley, M., ed.), Basil Blackwell Inc., Cambridge, MA, pp. 279-292.

52. Rushton, G., Z., 2000, Development of Modular Vehicle Systems, Department of Industrial and Manufacturing Systems Engineering, University of Michigan, Dearborn.

53. Sanderson, S. and Uzumeri, M., 1995, “Managing Product Families: The Case of the Sony Walkman,” *Research Policy*, vol. 24: pp. 761-782.

54. Samuel, A.K., and Bellam, S., 2000, http://www.glue.umd.edu/~sbellam/

55. Sanderson, S. W., 1991, “Cost Models for Evaluating Virtual Design Strategies in Multi-cycle Product Families,” Journal of Engineering and Technology Management, vol. 8: pp. 339-358.

56. Shirley, G. V., 1990, “Models for Managing the Redesign and Manufacture of Product Sets,” *Journal of Manufacturing and Operations Management*, vol. 3 (2): pp. 85-104.

57. Siddique, Z. and Rosen, D.W., 2001, “On Discrete Design Spaces for the Configuration Design of Product Families,” *Artificial Intelligence in Engineering, Design, Automation, and Manufacturing*, vol. 15, pp. 1-18.

58. Siddique, Z., and Rosen, D.W., 1999, “Product Platform Design: A Graph Grammar Approach,” *Proceedings of DETC'99, 1999 ASME Design Engineering Technical Conferences*, Sept.12-16, 1999, Las Vegas, Nevada, DETC99/DTM-8762.

59. Siegel, J., CORBA: Fundamentals and Programming: OMG, 1996.

60. Simpson, T. W., 1998, A Concept Exploration Method for Product Family Design', Ph.D Dissertation, System Realization Laboratory, Woodruff School of Mechanical Engineering, Georgia Institute of Technology.

61. Simpson, T. W., Maier, J.R.A., and Mistree, F., 2001, “Product Platform Design: Method and Application,” *Research In Engineering Design*, vol.13, pp.2-22.

62. Sivaloganathan, S., Andrews, P.T.J., and Shahin, T.M.M., 2001, Design Function Deployment: A Tutorial Introduction, *Journal of Engineering Design*, vol.12, No.1, pp.59-74.

63. Sivard, G., 2000, A Generic Information Platform for Product Families, Doctoral Thesis, Royal Institute of Technology, Sweden.
64. Sriram, R.D., 1997, Intelligent Systems for Engineering: A Knowledge-based Approach, London: Springer-Verlag, UK.
65. Sriram, R.D., 2002, Distributed and Integrated Collaborative Engineering Design, Sarven Publishers, Glenwood, MD 21738, USA.
66. Stadzisz, P. C. and J. M. Henrioud, 1995, "Integrated Design of Product Families and Assembly Systems," *IEEE International Conference on Robotics and Automation*, Nagoya, Aichi, Japan, vol. 2 of 3: pp. 1290-1295.
67. Stone R. B., Kristin L. W. and Crawford, R. H., 2000, "A Heuristic Method for Identifying Modules for Product Architectures," *Design Studies*, vol. 21(1): pp 15-31
68. Stokes, M., 2000, Managing Engineering Knowledge: MOKA Methodology for Knowledge Based Engineering Applications, MOKA Consortium, London.
69. Suh, N. P., 1990, The Principles of Design, New York: Oxford University Press.
70. Szykman, S., Sriram, R.D., and Regli, W.C., "The Role of Knowledge in Next-generation Product Development Systems," Journal of Computing and Information Science in Engineering, Transactions of ASME, Vol.1, pp.3-11.
71. Szykman, S., Racz, J.W., Bochenek, C., and Sriram, R.D., 2000, "A Web-based System for Design Artifact Modeling," Design Studies, Vol.21, No.2, pp.145-165.
72. Tichem, M. et al., 1997, "Designer Support for Product Structuring - Development of a DFX Tool within the Design Coordination Framework," *Computers in Industry*, vol. 33(2-3): pp. 155-163.
73. Tong, C. and Sriram, D. (Eds.), 1991a, Artificial Intelligence in Engineering Design: Volume I -- Representation: Structure, Function and Constraints; Routine Design, Academic Press.
74. Tong, C. and Sriram, D. (Eds.), 1991b, Artificial Intelligence in Engineering Design: Volume III -- Knowledge Acquisition, Commercial Systems; Integrated Environments, Academic Press.
75. Tseng, M.M. and Jiao, J.X., 1996, "Design for Mass Customization," *CIRP Annals*, Vol.45, No.1, pp.153-156.
76. Tseng, M. M. and Jiao, J.X., 1998, "Product Family Modeling for Mass Customization," *Computers in Industry*, vol. 35(3-4): pp 495-498.
77. Ulrich, K. and Tung, K., 1991, "Fundamentals of Product Modularity," Proceedings of ASME Winter Annual Meeting Conference, Atlanta, GA. DE Vol. 39: pp. 73-80.
78. Ulrich, K., 1995, "The Role of Product Architecture in the Manufacturing Firm," *Research Policy*, vol. 24(3): pp. 419-440.
79. Ulrich, K. T. and Eppinger, S. D., 1995, Product Design and Development, McGraw-Hill, Inc., New York.
80. Uzumeri, M. and S. Sanderson, 1995, "A Framework for Model and Product Family Competition," *Research Policy*, vol. 24: pp. 583-607.
81. Vuuren, W.V. and Halman, J.I.M., 2001, "Platform-Driven Development of Product Families: Linking Theory with Practice," Proceedings of Conference on "The Future of Innovation Studies", Eindhoven University of Technology, The Netherlands.
82. Wang, F., Fenves, S.J., Sudarsan, R., and Sriram, R.D., Towards Modeling the Evolution of Product Families, *Proceedings of 2003 ASME DETC*, Paper No. CIE-48216.
83. Wheelwright, S. C. and Sasser, W. E., 1989, "The New Product Development Map," *Harvard Business Review*, vol. 67 (May-June), pp. 112-125.
84. Wheelwright, S. C. and Clark, K. B., 1992, "Creating Project Plans to Focus Product Development," *Harvard Business Review*, vol. 70 (March-April): pp. 70-82.
85. Yu, J.S., Gonzalez-Zugasti, J.P., and Otto, K.N., 1999, "Product Architecture Definition Based Upon Customer Demands," *Journal of Mechanical Design, Transactions of the ASME*, vol. 121(3): pp. 329-335.
86. Zha, X.F., and Du, H., 2001, "Mechanical Systems and Assemblies Modeling Using Knowledge Intensive Petri Net Formalisms," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol.15 (2), pp.145-171.
87. Zha, X.F., and Lu, W.F., 2002a, "Knowledge Support for Customer-Based Design for Mass Customization," *AID'02,* Kluwer Academic Press, pp.407-429.
88. Zha, X.F., and Lu, W.F., 2002b, "Knowledge Intensive Support for Product Family Design," *Proceedings of 2002 ASME DETC*, Paper No. DETC02-DAC 34098.
89. Zha, X.F., Sriram, R.D., Lu, W.F., and Wang F., 2004, "Evaluation and Selection in Product Design for Mass Customization," *Business and Technology in New Millennium,* Cornelius T. Leondes (ed), Kluwer Academic Publishers, USA.