Submitted to Proceedings of DETC'2004
2004 ASME Design Engineering Technical Conference
September 28-October 2, 2004, Salt Lake City, Utah, USA

DETC2004/CIE-93

COLLABORATIVE PRODUCT DEVELOPMENT AND CUSTOMIZATION: A PLATFORM-BASED STRATEGY AND IMPLEMENTATION

Xuan F Zha
Manufacturing System Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899
Email: zha@cme.nist.gov

Ram D Sriram
Manufacturing System Integration Division
National Institute of Standards and Technology
Gaithersburg, MD 20899
Email: sriram@cme.nist.gov

ABSTRACT

Mass customization and global economic collaboration drive the product development and management beyond internal enterprise to cover the whole product value chain. This paper presents a platform-based strategy and approach for collaborative product development and customization. The implementation of this strategy takes 1) the product platform as the core, 2) the view/search engine and rule-based control as the data access and navigation mechanism, and 3) the internet-enabled web-based integration and collaboration bus as an enabler to allow participants involved in the product lifecycle to access into both internal and external enterprise resources, applications, and services. In the paper, a generic collaborative platform design and development process model is presented for product family design and mass customization. Based on this model, a module-based integrated & distributed collaborative framework for product family design and mass customization is developed with knowledge intensive support for customer or task requirements' modeling, product architecture modeling, product platform establishment, product family generation, and product variant assessment for customization. The issues related to the high-level information & knowledge modeling and the development of knowledge-intensive collaborative support framework are addressed. Finally, a case study for collaborative design of families of modular robotic systems is given.

Keywords: Product lifecycle management (PLM), product platform, product family, design for mass customization, collaborative product development, and collaborative product customization.

1. INTRODUCTION

The current product marketplace can be characterized by the need for product variety, faster time to market, and reduction in cost. To survive in competition, companies are shifting from mass production to mass customization to provide the necessary product variety. The challenges of mass customization are the increasingly varied demands of customers and extensive requirements of low cost and short lead-time. The issues of moving from mass production to mass customization, or even limited customization, make many companies struggling to reorganize their product architectures. In recent years, much work has been done in product modeling. Although information modeling is increasingly mature, models from the early geometric model, feature-based model to the current integrated model lack capability to model dynamic information and knowledge. They also lack capability to model product families in support of efficient product varieties and derivations.

A product platform is a set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently developed and produced (Meyer and Lehnerd 1997). An effective product platform can allow a variety of derivative products, i.e., a family of products to be created more rapidly and easily, with each product family member providing the features and functions desired by a particular market segment. 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.

Platform-based product family design has been recognized as an efficient and effective means to realize sufficient product varieties to satisfy a range of task requirements and customer demands in support for mass customization (Tseng and Jiao 1996, 1998; Zha and Lu 2002a-b; Zha and Sriram 2004, Zha and Sriram et al. 2004). Therefore, converting a system from one market segment (or task) to another can be very fast and flexible so as to keep up with the rapid change of marketplaces and applications. Cost effectiveness, technological leverage, and market power can be achieved when companies redirect their thinking and resources from a single product to families of products built upon robust product platforms. Research done in the areas of mass customization, design for product variety, design for product family, etc. has outlined different characteristics for such product lines and stresses the need for developing a common platform. On the other hand, product design process has become increasingly multi-disciplinary, knowledge intensive, distributed and collaborative (Sriram 2002, Szykman et al. 2001). Knowledge intensive support becomes critical and has been recognized as a key solution towards future competitive advantages in product design and development process.

To improve the product/family design and customization process, it is imperative to provide knowledge-intensive collaborative support in the product/family design process and share design knowledge among distributed designers. Several quantitative frameworks have been proposed for the two phases described earlier in platform product development (Gonzale-Zugasti 2000, Simpson 1998, Simpson 2001, 2003, Sivard 2000, Zha and Lu 2002, Tseng and Jiao 1998). They provide valuable managerial guidelines in implementing the platform product development approach. However, there are very few systematic qualitative or intelligent methodologies for adopting the platform product development practice, despite the progress made in several research projects (Simpson et al. 2003, Zha and Lu 2002a,b, Zha and Sriram 2004, Zha and Sriram et al. 2004). Designing product platforms collaboratively as a very important topic of mass customization is not covered much in the literature yet. The current status of research does not sufficiently address the issues of how to develop common platforms collaboratively and how to integrate these distributed and heterogeneous factors under a web based unified platform.

This work aims to develop a collaborative product platform approach and strategy for product development and customization. The paper discusses the fundamental issues underlying collaborative product platform development and mass customization management. The focus is on the development of a strategic collaborative framework for platform-based product family design and mass customization, including the rationale of modular product platform planning and generation for implementing product family and mass customization.

The organization of this paper is as follows. Section 2 is an overview of the proposed approach, including the strategy and technology for collaborative product development and customization; Section 3 discusses collaborative platform product design and customization process; Section 4 presents a knowledge intensive framework for collaborative product platform design and customization; Section 5 discusses information and knowledge modeling and support process for collaborative product platform design and customization; Section 6 develops a prototype system and provides a case study to illustrate the proposed approach; Section 7 summarizes the paper and discusses future work.

2. OVERVIEW OF THE PROPOSED APPROACH: STRATEGY AND TECHNOLOGY

More and more enterprises have recognized that to share and reuse the product knowledge within a product value chain is essential for remaining competitive in the global marketplace. Therefore, how to model, manage and utilize the product information across the whole product lifecycle effectively and efficiently is a critical issue for companies. Product lifecycle management (PLM) is one of the popular ideas to address such a situation; PLM can be used as a unified approach that provides web-based access to in-house people and partners across the supply chain (http://www.prdm.net/).

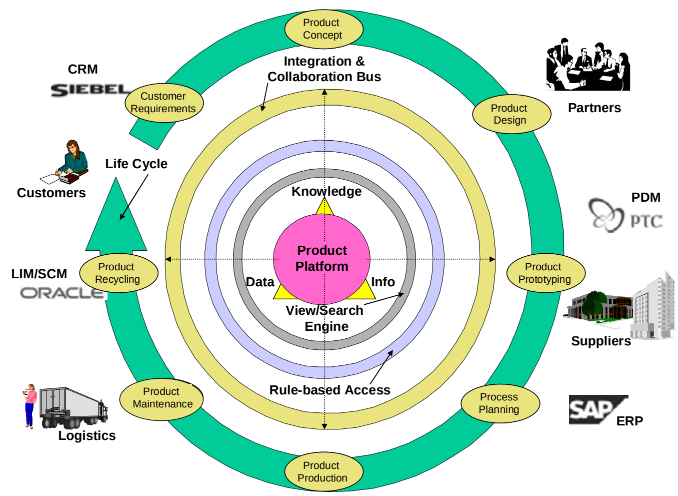

Figure 1: Platform-based strategy for product lifecycle management

In this work, a platform-based strategy is adopted to develop a variety of similar products. This is achieved by enabling the design and production of several related products from a common base for different market segments. Thus, it can direct the development of information management system for product lifecycle, which emphasizes on the organization pattern of product data in order to support rapid product derivation and mass customization. Figure 1 illustrates the platform-based strategy for product lifecycle management and collaborative development and customization. In the proposed strategy and approach, the product platform is built as the core and base of product lifecycle management and collaborative development. An interactive process model is developed for collaborative platform design and customization, including: design requirements modeling, platform design, variants design, platform evaluation, re-negotiation, and iteration. The view/search engine and the rule-based control access are introduced as a powerful data/information access, navigation and security mechanism for design/knowledge repository. The integration & collaboration bus is constructed to allow participants involved in product development across extended enterprise to get easy, web-based access using a unified interface (e.g. web browser) to the internal and external enterprise resources, applications, and services. A knowledge intensive framework is developed to facilitate data/ information access, exchange and reuse and support collaborative work, e.g., new product derivation and customization across the extended enterprise. To implement the strategy above, the following key technologies are applied, which include:

(1) Core product model and design/knowledge repository technology (Fenves 2001, Szykman et al 1999);
(2) Product platform modeling and modular product family design technology (Zha and Lu 2002a,b, Zha and Sriram 2004, Simpson et al 2001);
(3) Distributed object and web-enabled integration & collaboration technology (Zha and Du 2002, Sriram 2002, Pahng et al 1998);
(4) STEP and XML-based product data exchange technology (ISO 10303, Part 28); and
(5) Data security control technology.

Details about the platform-based strategy, approach and technology for collaborative product development and customization are discussed below.

3. COLLABORATIVE PLATFORM PRODUCT DESIGN AND CUSTOMIZATION PROCESS

In this section, we first propose a collaborative platform-based design and development model for product families and mass customization, and then present a procedure for creating an architectural product platform based on a modular design approach.

3.1 Collaborative platform product design and customization

In collaborative product development, different participants and different enterprise information systems are involved in the product lifecycle. How to integrate these distributed and heterogeneous entities and resources under a web-based unified platform is a critical issue to be addressed. In order to penetrate the organization pattern of product data, the product platform-based strategy for PLM is defined as an integrated data and information framework that supports the organic integration of all the product assets, including components, process, tools, people as well as their relationships across the whole product lifecycle. The features of the platform-based strategy for PLM and collaborative product development (Figure 1) are described as follows (http://www.prdm.net):

(1) Systematically organized product assets as a core and base. Based on the platform-based approach and strategic framework, product assets are organized in a reusable manner and organized into three tiers: data services, logical business services, and collaborative development services. All people involved in product development can easily access and reuse the correct and current information. This allows the product developers to focus and align their users, product knowledge, and business processes for product innovation.
(2) Information view/search engine and rules as a data access mechanism for design/knowledge repository. Different people have different data requirements along the product lifecycle. Meanwhile, people involved in the different phases of product development should share their information to support concurrent engineering to improve the product quality and to shorten the development time. Therefore, a mechanism, called information view/search engine, is needed to view, query, filter and synthesize or mine data. With respect to the security, data access ability should be distinguished according to the users' roles, and thus access rules should be customized to avoid invalid access and operation of the product data.
(3) Internet-enabled and web-based integration and collaboration. Based on a completely open, standards-based, web-native architecture, every engineer residing in different locations connected by Internet can share and access product content, application and services in product lifecycle. Thus, customers and suppliers can participate directly in the processes of product innovation. Component standards enable users to plug extended or alternative capabilities offered by other software providers into their environment.

With respect to the concept of product platform, an interactive process model is advocated for collaborative platform-based product family design and customization. This includes the following two tasks: design requirements and models; platform design; variants design; and platform evaluation, re-negotiation, and iteration (Gonzale-Zugasti 2000; Zha and Lu 2002a). These tasks are the challenges for collaborative product platform design and customization success. Figure 2 gives an overview of the collaborative process model applied for cell phone family design and customization. Each step in collaborative product platform design and customization process can be explored in more detail, as follows (Gonzale-Zugasti 2000):

(1) Design requirements and models. Designers and customers first collaboratively construct models (e.g. customer needs, functional requirements, and design constraints) that connect the process models, design choices to the performance indices for products in a family.
(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 and customization. Once a platform is generated, it is customized into individual product variants to meet the specific market, business and engineering needs. Therefore, a portion of the design is handed over to the individual design team 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. They 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 is necessary to renegotiate the platform choices and iterate through the design loop to arrive at an adequate family design.

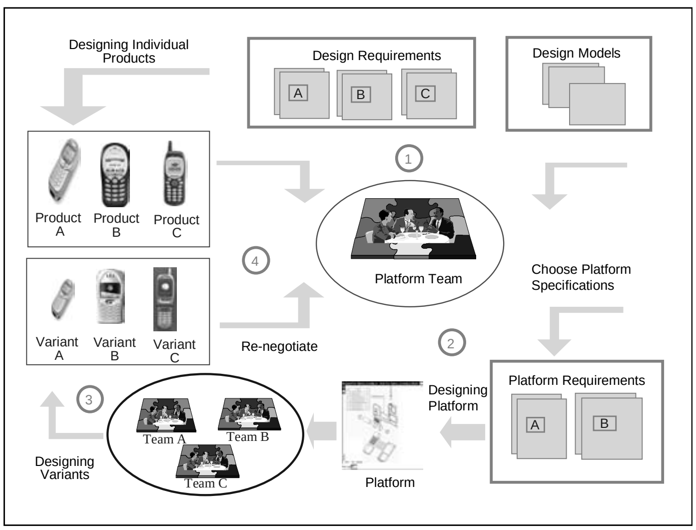

Figure 2: Collaborative platform-based product design and customization

3.2 Collaborative module-based product platform design process

Over half of all industrial products have subsystems or components or modules as a part of their basic designs. The modularity design concept has been widely used in product design for flexibility, rapid responsiveness, ease of maintenance, and rapid deployment. A modular product system is a collection of interchangeable modules (e.g. link and joint units in modular robots) that can be assembled into many different types and configurations of products (Stone et al 2000, Chen et al 1999, Chen 2001, Leger 1999, 2001). Unlike a conventional industrial product that is designed for general tasks or purposes, a modular product system has the advantage of providing feasible product configurations for specific customer or task requirements. Thus, the modular design of products provides the ability to achieve product variety through the combination and standardization of modules.

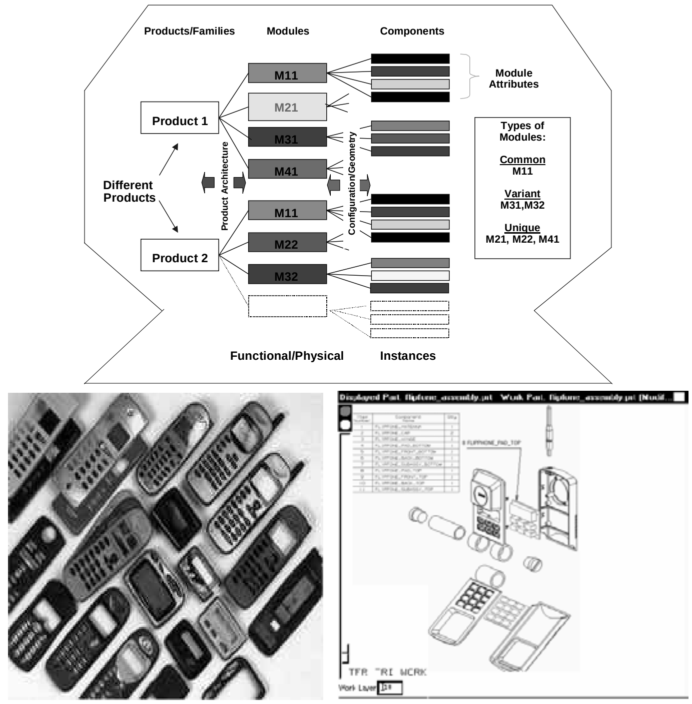

Figure 3: Product family architecture (PFA)

Product families can be decomposed into systems, modules, components and attributes (Fujita and Ishii 1997), i.e., a hierarchical structure, as shown in Figure 3. Under the hierarchical representation scheme, the product variety can be implemented at different levels within the generic product architecture, i.e., product platform. Thus, decomposing the problem into modules and defining how modules are related to one another create the model of a product/family design problem. The relationships amongst modules specify how outputs of a module are connected to inputs of other modules. The creative aspect of modular platform or family design is to develop a mechanism or configuration that is capable of performing the required customer requirements or tasks. Usually the basic functions of a modular product family are determined by the module configuration, while its performance relies mainly on the modules' sizes and characteristics. The procedures for creating module-based product families are as follows:

(1) Decompose products/tasks into their representative functions/task primitives;

(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

Following the above steps, the modular product platform or family design can be implemented, in which a re-configurable product platform can be easily modified and upgraded through the addition, substitution, and exclusion of modules.

In view of the hierarchical structure of product families (Figure 3), the architecture of product platform can be considered as the composition of five elements: product families, product modules, product variants, product components and product relations, which reflect on the four corresponding levels: capability level, function level, configuration level and instance level, as shown in Figure 4.

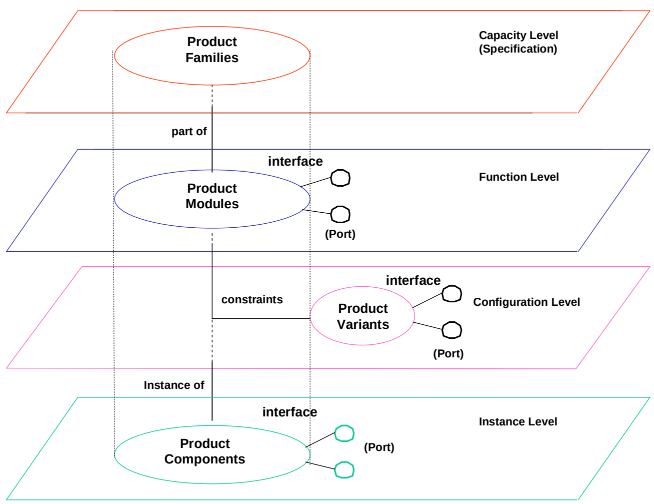

Figure 4: Product platform architecture (PPA)

4. WEB-BASED KNOWLEDGE INTENSIVE FRAMEWORK FOR COLLABORATIVE PRODUCT PLATFORM DESIGN AND CUSTOMIZATION

In this section, a modular platform-based product family design and customization using a knowledge-intensive collaborative support paradigm is presented. We first discuss a knowledge support scheme for the product platform design and customization process, and then propose a collaborative design framework for implementing the modular platform-based product family design and customization.

4.1 Knowledge intensive support for product platform design and customization

The proposed whole platform design process is roughly divided into two main stages: product platform planning and family design customization, which ranges from capturing voices of customers or task requirements and market trends for generating product design specifications, product variants to customizing products for customers' satisfaction. This complies with the general platform product development approach mentioned above.

The product platform planning stage embeds the voices of customers or task requirements into the design objective and then generates product design specifications. The product family design customization stage realizes sufficient product variety- a family of products to satisfy a range of customer demands or task requirements. Therefore, a product platform is not simply the physical modules, but the underlying knowledge on customer insights, product technologies and manufacturing process, etc. With respect to the proposed modular platform-based design approach, a knowledge intensive support framework is developed (Zha and Lu 2002a,b).

The knowledge support scheme is implemented through customer or task requirements' modeling, product family architecture modeling, product platform establishment, product family generation, and product variant assessment. Challenges in knowledge support process are summarized as follows:

(1) Design information and knowledge modeling: modular product platform design process knowledge capture, classification, representation, and organization and management, including component modularization and standardization, parameterized module and product descriptions, product configuration management, etc

(2) Product family architecture (PFA) modeling: representing product/ family architecture, variety representation and evolution;

(3) Product platform establishment: exploring methods for hierarchical modular design (configuration synthesis and optimization);

(4) Product family generation*:* generating product variants or family members;

(5) Product variant simulation and assessment: simulating and evaluating product variants.

The product family architecture (PFA) should represent the conceptual structure and logical organization of product families from viewpoints of both customers (for tasks) and designers (engineering related). It provides information support for collaborative design and customization (see Section 5.2). 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.

4.2 Web-based collaborative design framework

Based on DOME (Pahng et al. 1998), a web knowledge-server based distributed module modeling and evaluation framework (WebDMME) was developed in (Zha and Du 2002) to enable designers to build concurrent integrated design models using both local and distributed resources, and to collaborate by exchanging services based upon the CORBA standard communication protocol (http://www.omg.org/). WebDMME is a web-based knowledge-intensive collaborative design framework, which adopts the design with modules, modules network, and knowledge server paradigms. The knowledge intensive system based on WebDMME can thus exploit the modularity of knowledge-based systems, in that the inference engines and knowledge bases are located on server computers and the user interface is exported on demand to client computers via network connections. Therefore, modules under the WebDMME framework are connected together so that they can exchange services to form a large integrated model. The module structure of WebDMME leads itself to be a client (browser) / knowledge server oriented architecture using the distributed object technology.

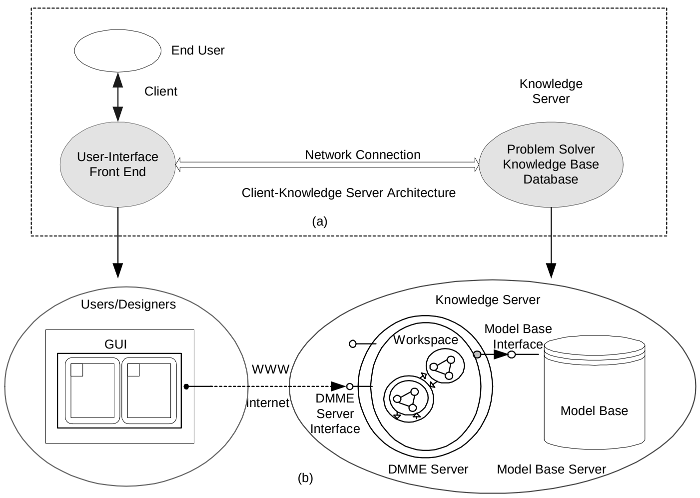

Figure 5: (a) Client-knowledge server architecture; and (b) Main components for WebDMME

Figure 5 shows the major system components of the proposed client (browser) and knowledge server architecture. Each of these components interacts with one another using CORBA so that it is not required to maintain the elements on a single machine. As a gateway for providing services, the interface of a system component invokes the necessary actions to provide requested services. To request a service, a system component has an interface pointer to the desired interface.

The WebDMME architecture allows designers or experts to publish and subscribe design modeling and decision support services on the WWW. These services will operate when information is received from other clients or knowledge servers. When module services are connected, the resultant service exchange network forms a concurrent integrated system model. Any service request in the module network can invoke a chain of service requests, if needed, to provide correct information. When a design alternative is evaluated, the local model asks for the services of subscribed models. If the subscribed models themselves need services from other models in order to provide the request services, they will again request those services from their own network to remote models. Thus, the service requests are propagated through the connected modules. The "web-top" mass customized products can be achieved by using the technologies of e-commerce and mass customization to design and set up the product platform and families on the web based on the remote-site customers and task requirements for re-configurable modular systems (Zha and Sriram 2004).

4.3 Collaborative modules network formulation under WebDMME

As discussed above, the modular modeling process decomposes a design problem into modules and defines how modules are connected to one another. The embedded model of a module produces outputs using its internal design resources as well as inputs from other modules. Figure 6 illustrates a simple distributed module network model used for the design process. The variables of the model are governed by a set of equations. As depicted in Figure 6, the interface connections between variables in different modules (A, B) could be established interactively or defined explicitly using the Model Definition Language (MDL) (Pahng et al. 1998). The embedded models defined with the variable declaration could also be created separately and linked to the model definition using keywords. Modules A and B are local to the problem. Using the remote module AB, a new design model (ABC) can be created. In this case, the problem model is made available for use as a distributed module with the outward appearance in Figure 6a. This distributed module allows users to utilize variables such as A (A_1, A_2), B (B_1, B_2), C (C_1, C_2), and AB (B_1, A_2). Figure 6b illustrates the model from the viewpoint of the ABC designer. Module C is local to the designer. Figure 6c illustrates the true integrated model created when the remote module AB and the local module C are connected. The problem model ABC is thus created, which requires additional information such as the distributed module's name and IP address.

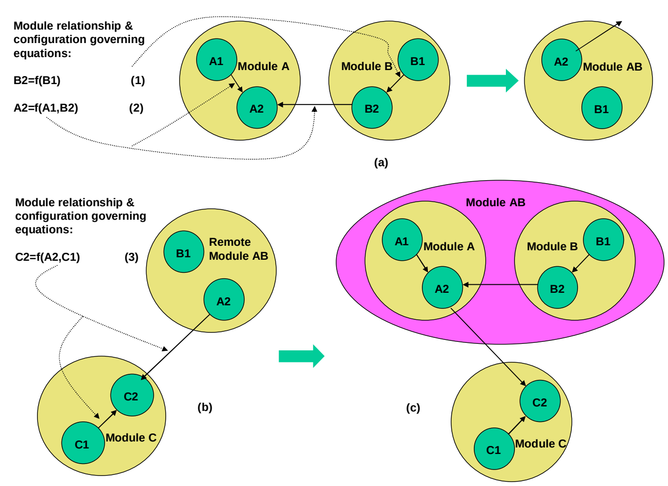

Figure 6: Distributed design model with two modules and a remote module: (a) modules A and B, (b)(c) remote module AB

It is shown that the relations between modules do not need to be changed even if the embedded model of a remote module (i.e., module AB) is changed. This flexibility enables a designer to define a model independently from the actual locations (i.e., local or remote) of embedded models. When the designer utilizes the remote module AB in conjunction with the local module C, the resulting integrated model forms a distributed computing system comprised of two autonomous computing elements. Figure 7 illustrates the configuration process for distributed modules using the system components, including the Internet and WWW resources. The embedded model of the module AB in design problem model ABC contains an object connector that manages the design information exchange with the distributed design object AB.

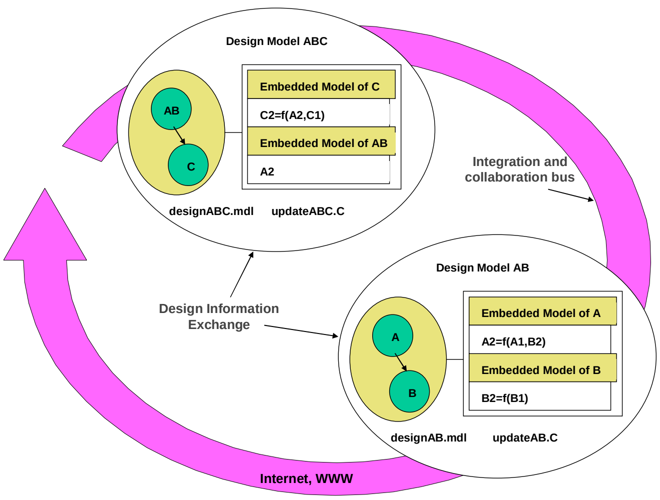

Figure 7: Module configuration under WebDMME

5. INFORMATION AND KNOWLEDGE MODELING AND SUPPORT PROCESS FOR COLLABORATIVE PRODUCT PLATFORM DESIGN AND CUSTOMIZATION

Based on the framework above, the realization of knowledge-intensive collaborative product platform design and customization can be supported through two steps: design knowledge modeling and collaborative support for the design process and design process control, which is discussed in this section.

5.1 Product platform and family design knowledge modeling

The design knowledge modeling is to elicit knowledge in the design process and to establish a comprehensive knowledge repository that can be retrieved and reused when necessary (Szykman et al. 1999, 2001). As discussed in Sections 3.2 and 4.1, the product platform and family design process starts from a set of customer/functional requirements. The requirements are implemented by a set of modules described in terms of design variables of the product. These modules' design variables propagate to the functional requirements on the lower level elements of the module, so on and so forth until all the modules and their components are specified. With respect to the product platform design and customization process, three categories of knowledge are identified (Du et al. 2001, Sivard 2000):

(1) How to deploy the functions of products (modules) to lower level modules;

(2) How to select modules among the standard ones or the custom ones; and

(3) All selected modules are configured to be an end product family. The performance of each of them has to be estimated /evaluated to help both the designer and the customer make decision.

Product family design knowledge refers to the collection of data/information and knowledge needed to represent and support the design activities and decision-making in the product family design process. The product family design knowledge should be abstracted and classified into different categories, e.g., off-line and on-line, product platform/family data/information and product family design process knowledge through analysis of product design process (Zha and Lu 2002a,b). Different categories of product design knowledge are represented in different ways from multiple views of the product family design process (Mckay et al 1996). Since product design knowledge includes all product data/information needed throughout the whole design process, a new product data and information model must be employed, which may include customer and task requirements, design specifications, functions, behaviors, structures, assemblies, performance constraints, metrics, etc. The design/knowledge 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 platform/family information representation model is to integrate the database representation model and the design process model. Such an information model still needs to be divided into two parts: one for modules and the other for modules assembly. The module representations follow the object-based formalism (Gorti et al 1998, Sriram 2002), while the module assembly is based on the graph theory and its incident matrix representation, e.g., assembly incidence matrix (AIM) (Chen 2001, Chen et al. 1999; Leger 1999). Therefore, a multi-level hybrid representation schema is adopted to represent the product platform 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, as follows:

Product Definition:
- Requirements;
- Specifications;
- Artifacts;
- Features;
- Functions-behaviors-forms;
- Performance objectives and constraints;
- Relationships;
- Design rationale;

Product Variety:
- Assembly structure;
- Module details;
- Family parameters;

Product variants;

Following the requirements of designing product families with a high degree of commonality as well as designing several products around reusable components/modules, two main elements of the architecture are: generic product specifications, and reusable solution libraries. Product architectures and component architectures (Figures 3 and 5) are treated in a similar way, enabling a hierarchical structure of structures. Thus, classes or families of components may be selected from the solution libraries and integrated into the framework. Figure 8 illustrates the construction process (Steps 1-4) of product platform and the reuse for domain-specific applications (Step 5). For illustration, an object-oriented representation instance for a modular robot family and its parameterized module information (e.g. link and joint modules) is described as follows (Zha and Sriram 2004):

```
Module ("Joint Module") {
    Number of degree of freedoms (DOFs): [1,2,3];
    Motion type: ["translation", "rotation"];
    Active attribute: ["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];
}
```

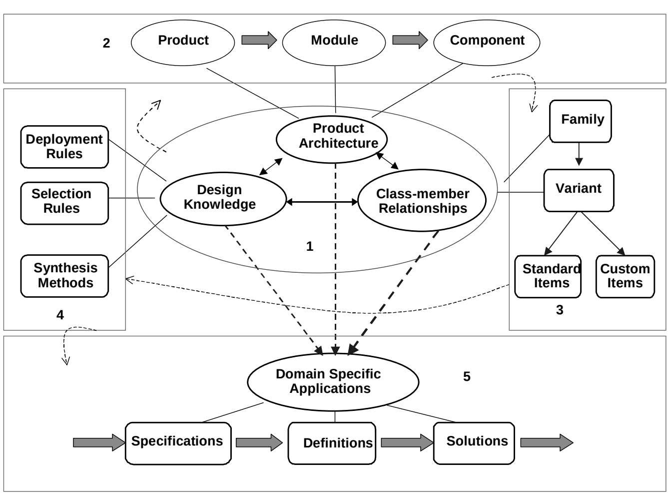

Figure 8: Modular product platform construction process, representation and reuse

5.2 Web-based modular product family design repository

Based on the WebDMME framework of collaborative design and customization in Section 4.2, the convergence of computer networking and telecommunication technologies between manufacturers and designers must be resolved for the design and development tools.

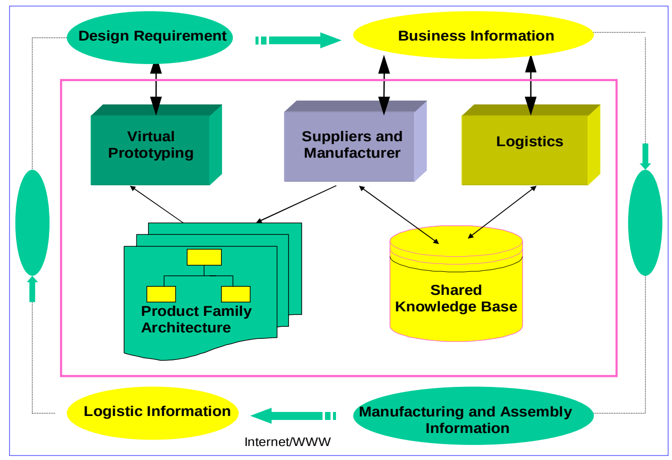

Figure 9: Information flow for web-based module design prototyping and outsourcing

Figure 9 illustrates the information flow within the web-based design process between the manufacturers and designers. The product family architecture (PFA) provides information support for collaborative design and customization. The information framework would provide designers the ability to electronically customize product families through the Internet and receive real-time responses regarding the prices and delivery dates for the desired modules along with the 3D images and performances of the customized designs. If these performances and terms are acceptable, the designer can place an order to the manufacturer, who can utilize the business information for production planning. Through this process, virtual module/product prototyping would provide the data to manufacturers for their resource planning to reduce costs while allowing interaction between designers and engineers to collaboratively develop products (Fok et al. 2001). In this aspect, the shared knowledge base in Figure 9 is established to facilitate continuous process improvement and organized as a *knowledge repository* for technical information, business performances and customer feedback.

From the above discussion, in the context of web-based modular product family design repository, the main purpose of PFA is to provide for an informational linkage between the designer and the manufacturer. This can be accomplished using a web-based tool to match designers' needs to the capabilities and features available in different module groups. The information required can include cost evaluation and the technical analysis of compliance and compatibility between modules within the design. With reference to the product platform architecture (PPA) in Figure 4, a standard structure of

PFA is defined by the manufacturers in consultation with the designers, taking into consideration the functional difference of various types of standard modules. Without a common PFA structure, it would be difficult for a designer to search and locate suitable module models among the collections of different manufacturers on the information highway. Figure 10 shows the flow diagram of the module search engine. In this engine, several manufacturers and designers could cooperatively work together concurrently. This would allow the active sharing of information. In this aspect, a common database structure would minimize the time and effort to accomplish the required search.

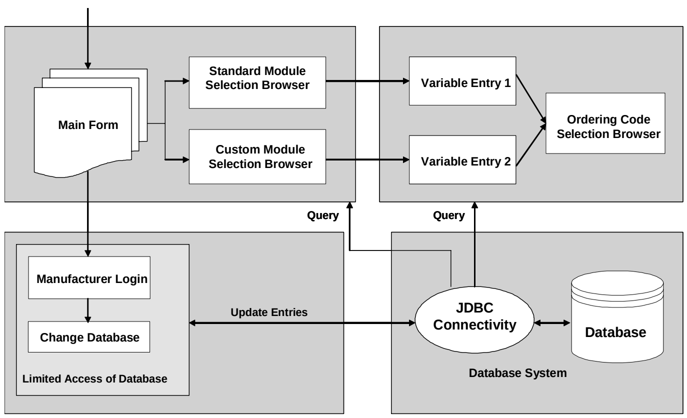

Figure 10: Flow diagram of a module search engine

5.3 Knowledge supported collaborative design process

Once the design knowledge repository is built up, the user/ designer can utilize the knowledge to solve the collaborative product family design problem. For a modular product system, the collaborative design problem model must be made available for the use of distributed modules and the implementation of collaborative module network. The relationships among modules are relatively simple and can be described by assembly/configuration graph or matrix (Leger 1999). Normally, a product family (e.g. robot family) is assembled through the use of fixed types of modules in the module inventory (repository) while maintaining re-configurability and inter-operability.

Therefore, the collaborative modular design process is concentrating mainly on the following tasks:

(1) Configuration of the distributed "plug-and-play" modules. To ensure the rapid deployment and unified control operation of products with different structures and DOFs.

(2) Product configuration optimization. To establish an optimization model for the rapidly configurable system based on an appropriate selection of module components according to the task/customer requirements, kinematics and dynamics performance objectives and constraints.

(3) Control of system component. To provide reconfiguration capability at system level and coordinate tasks among various system devices and modules.

(4) Simulation for system. The CAD-featured environment is used for the visualization, evaluation and simulation of the re-configurable system, and provides the real-time interface to the control of re-configurable products.

With respect to the product configuration and its optimization, a distributed module network can be formulated based on the above collaborative design process model, in which the variables of the model are governed by a set of equations and matrices, e.g., AIMs, transformation matrices, etc. The interface connections between variables in different modules could be established interactively or defined explicitly using IDL. The embedded models defined with the variable declaration could also be created separately and linked to the model definition. The distributed module allows users to utilize some variables and to evaluate the published modules.

The whole knowledge support process for product platform design and customization can be fulfilled through 11 steps discussed in (Zha and Lu 2002a,b). Customers/clients, designers, and manufacturers should collaborate each other in each or some of these steps. As an open environment, the developed framework can facilitate the implementation of collaborations.

5.4 Knowledge supported collaboration process control

The collaboration process control has two primary functionalities: 1) controlling the data operation authorities and 2) managing the data operation sequences. The operation authorities are controlled by access rules according to the user's role (client, designer or manufacturing engineer) (see view/search engine in Section 3.1). The collaborative design and prototyping process (e.g. module outsourcing) requires different data access interfaces while the data operation sequences are more complex. The remaining of this section outlines a solution proposed to this issue. We take the module outsourcing process as an example, and activities in the whole outsourcing order life cycle are classified into 10 categories:

- Submit an order: place and specify an order;
- Specify the product and module: define the requirements; function, and structure of the requested items;
- Make agreement: make the contract;
- Upload document: upload the shared documentations into the document repository;
- Payment: pay for the ordered items and services;
- Design: product design activities;
- Production: product processing and production;
- Delivery: shipping the products;
- Service: after-sale services provided by OEM if required; and
- Closed: close the order.

There exist certain operation sequences among activities above. For instance, product specification happens only after order specifying, and the agreement is made after the order and product are specified completely. In order to specify the process status qualitatively, there are four states for each kind of activity defined: “Will Do,” “Initiating,” “Updating,” and “Done.” The data such as the order specification, product specification, and operation sequences are dependent on the status of those activities. For example, if the status of “Submit an Order” is “Done,” the order specification would not be able to be updated. If the designer or manufacturing engineer/user wants to update the specification, he/she must set the status in advance. The system would send an alert email automatically to both the manufacturing engineers and the order-related clients for any status change.

6. PROTOTYPE SYSTEM AND CASE STUDY

In this section, we discuss a prototype system developed in this work and used as a collaborative design and customization platform. For illustration, a distributed design model is built for collaborative micro-robotic system design.

6.1 Prototype collaborative design and customization system

The solution to providing distributed collaborative design is to develop a web-based collaborative design tool. To facilitate the rapid construction of integrated models for robot customization, a prototype system is implemented under the WebDMME framework with concurrent integration of multiple cooperative modules (Zha and Lu 2002a,b). WebCDMC is a Web-based Collaborative Development and Mass Customization environment, consisting of 3D design, modeling and simulation software tools, which enable the collaborative creation and customization of product families.

Based on the proposed integrated distributed models, WebCDMC can offer a uniform model construction effort, e.g., kinematics, dynamics and calibration across computer simulation and real-time control of arbitrary product configurations (Dalton and Taylor 2000). WebCDMC is intended to be a uniform interface for modular products and is portable to modular product families from different vendors. Thus, it allows the user to quickly integrate the hardware components into modular systems, and to manage their operations in the reconfigurable system collaboratively. The system can provide distributed users access to module servers throughout the network over the internet/intranet and web. With this system, designers in different teams and organizations may participate and collaborate in the design process. On the other hand, the web-based system can provide end-user an optimized product configuration according to the input task requirements. The user does not need to start the design work from scratch. Rather, based on the result of optimization, the user/designer can fine-tune the suggested design or layout. The development effort and time for product can be greatly reduced.

The implementation of the prototype system uses the two-tier client/ knowledge server architecture to develop a front-end graphical user interface (GUI) using Java and integrating with VRML/XML browser and viewer to support collaborative design interactions in family design (Zha and Lu 2002a,b). The underlying framework modules, knowledge support engines and collaborative mechanisms are written in Java™/Java Bean™ They can also integrate with existing design and simulation packages such as CAD/solid modeling, kinematics/dynamics analysis and evaluation, and module/model database applications for configuration synthesis and optimization.

To enable real-time communication/talking among customers, designers and manufacturing engineers, Windows NetMeeting is incorporated into WebCDMC and used as a video design conferencing tool. The whiteboard system is implemented in NetMeeting. Using the chat function of the NetMeeting, designers/users can talk to each other and discuss with customers, other designers, and manufacturing engineers for any issues. Using the program sharing function of the NetMeeting, designers/users can also share CAD systems with manufacturing engineers to discuss or design the product together in real-time. If designers/users share an image editing software, such as the Microsoft Paint, and import the image file of the object (module, product variant, family), then they can mark it up, and discuss any issues.

6.2 Case study: collaborative micro-robotic system design for customization

In this case study, we suppose that customers, designers and manufacturing engineers from different teams, divisions, or companies at remote locations would like to participate in designing a customizable modular micro-robotic system. The micro-robotic system consists of three major parts: a modular robot manipulation system, a work platform including a microscope system, and micro components to be assembled. The overall topology of the micro-robotic system design problem and design workspace is described in (Zha and Du 2002). Designers from the robot and gripper manufacturing teams provide their models to the robotic system design team who in turn develops the technical models for the robot system. The robotic system manager collaborates with the robotic system designer by providing models and data for robotic operating conditions and requirements. The design models are used to develop cost evaluation and redesign models. The robot and gripper manufacturing teams respectively develop models for their products so that the designer and customer can obtain performance predictions for different parametric configurations and operating conditions. The microscope system and some modules need outsourcing. These individual models are constructed, published and served by each party. If a single individual creates the model to provide these services they work in an individual workspace.

On the other hand, for the robot manipulation system itself, the modular design concept 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 11. The robot variety (family) is implemented through the combination and standardization of modules at different levels within the generic product/robot architecture, i.e., platform to satisfy customer/task requirements. Figure 12 illustrates the collaborative outsourcing process and process control for robot modules, including module cataloging, specification, configuration and email communications between designers and engineers. Figure 13 shows screen snapshots of the web-based collaborative customization design and visualization sessions for micro-robotic assembly system: (a) the robot platform session; (b) the robot family design session; and (c) the microscope design visualization session.

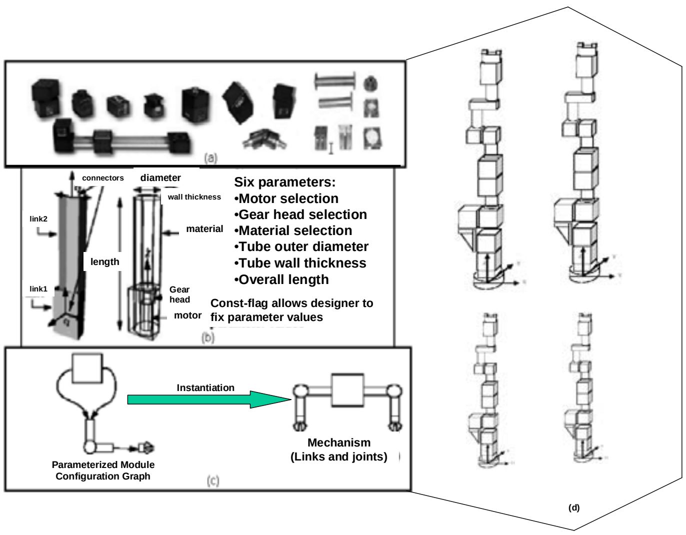

Figure 11: (a) Robot modules, (b) module attribute parameterization, (c) assembly / configuration, and (d) robot families

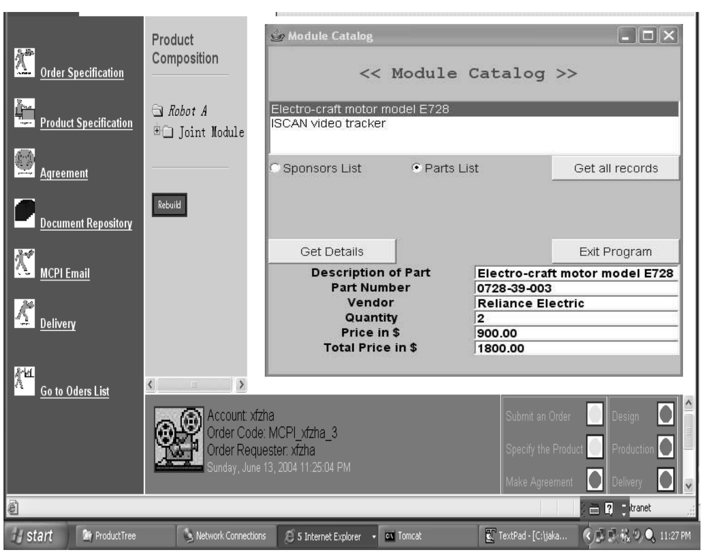

Figure 12: Screen snapshot of collaborative outsourcing of modules and configuration

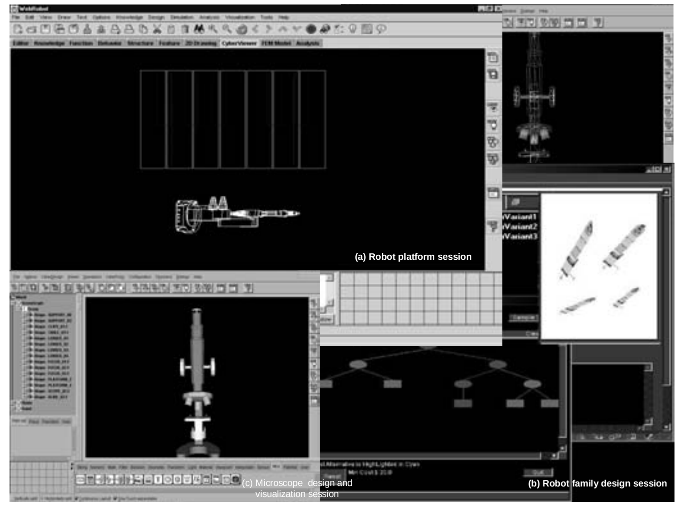

Figure 13: Screen snapshots of micro-robotic system customization design and visualization sessions

7. SUMMARY AND FUTURE WORK

This paper presented a platform-based strategy and approach to support collaborative product development and customization. The focus was on the collaborative modular design of product families for mass customization. The proposed scheme integrates distributed customer/task requirements' modeling, product architecture modeling, product platform establishment, product family generation, and product assessment. The developed collaborative framework and system can be used for capturing, representing, managing and sharing product family design knowledge and offer collaborative support for product family design for mass customization process. The approach can be used as guidelines for a product data management system to evolve into a product lifecycle management system that will benefit for product knowledge sharing, product quality and processes of product innovation, and finally shorten the time-to-market and the customers' satisfaction. The features of the approach can be described: systematically organized product assets as a core and base; view engine and rules as data access mechanism; and the web-enabled integration and collaboration bus as the pivot. Some of them have been tested and applied to practical problems. For example, a product platform has been developed and used to support design of modular robotic families, and its effectiveness is validated. However, there is still much work to be done in future. The platform model needs to be decomposed explicitly enough to make it easier and more flexible to implement and manage. The prototype system still needs to be further developed to test the full approach. Also, the model presented in the paper will be re-implemented to fully comply with the core product model (Fenves 2001) developed at the National Institute of Standards and Technology, USA. This work is underway.

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 with Nanyang Technological University and Singapore Institute of Manufacturing Technology, Singapore.

REFERENCES

Chen, I.M., Yeo, S.H., Chen, G., and Yang, G.L. (1999), Kernel for modular robot applications: automatic modeling techniques, International Journal of Robotics Research, 18(2): 225-242

Chen, I.M. (2001), Rapid response manufacturing through a rapidly re-configurable robotic workcell, Robotics and Computer Integrated Manufacturing, 17(3): 199-213

Dalton, B. and Taylor, K. (2000), Distributed robotics over the Internet, IEEE Robotics and Automation Magazine, 7(2): 22-27

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.

Fenves, S. J., (2001), A Core Product Model for Representing Design Information, NISTIR 6736, NIST, Gaithersburg, MD

Fok,S. C., Xiang, W., and Yap, F.F. (2001), Internet-enabled virtual prototyping workbench for fluid power systems, Proceedings of ICFP'2001, China

Fujita, K. and Ishii, K. (1997), Task structuring toward computational approaches to product variety design, Proceedings of ASME, DETC97/DAC-3766

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

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, 30(7): 489-501

ISO 10303, TC184/SC4, STEP Part 28: XML representation of EXPRESS schemas and data

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, 8(4): 297-322.

Leger, C. (1999), Automated Synthesis and Optimization of Robot Configurations: An Evolutionary Approach, PhD Thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania

McKay, A., Erens, F. and Bloor, M. S. (1996), Relating product definition and product variety," Research in Engineering Design, 8 (2): 63-80.

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

Pahng, F. Senin, N. and Wallace, D. (1998), Distribution modeling and evaluation for product design problems, Computer-Aided Design, 30(6): 411-423

Sriram, R.D. (2002), Distributed and Integrated Collaborative Engineering Design, Sarven Publishers, Glenwood, MD 21738, USA

Simpson, T. W., Maier, J.R.A., and Mistree, F., (2001), Product platform design: method and application, Research in Engineering Design, 13(1): 2-22

Simpson, T.W., Umapathy, K., Nanda, J., Halbe, S., and Hodge, B. (2003), Development of a framework for web-based product platform customization, Journal of Computing and Information Science in Engineering, Transactions of the ASME, 3(2): 119-129

Sivard, G. (2000), A Generic Information Platform for Product Families, Doctoral Thesis, Royal Institute of Technology, Sweden

Szykman, S. , Sriram, R.D. , Bochenek, C. , Racz, J., (1999), The NIST Design Repository Project, Advances in Soft Computing - Engineering Design and Manufacturing, Roy, R., Furuhashi, T. and Chawdhry, P. K. (eds.), Springer-Verlag, London, pp.5-19

Szykman, S., Sriram, R.D., and Regli, W. (2001), The role of knowledge in next-Generation product development system, ASME Journal of Computing and Information Science in Engineering, 1(1): 3-11

Tseng, M.M. and Jiao, J.X. (1996), Design for mass customization, CIRP Annals, 45(1): 153-156

Tseng, M. M. and Jiao, J.X. (1998), Product family modeling for mass customization, Computers in Industry, 35(3-4): 495-498

Zha, X. F. and Du, H. (2002), Web-based collaborative framework and environment for designing and building robotic systems, Proceedings of 2002 IEEE International Conference on Robotics and Automation (ICRA'02), USA

Zha, X. F. and Lu, W. F. (2002a), Knowledge intensive support for product family design, Proceedings of ASME, DETC 02/DAC-34098

Zha, X. F., and Lu, W. F. (2002b), Knowledge support for customer-based design for mass customization, Artificial Intelligence in Design' 02, Kluwer Academic Publisher

Zha, X.F., Sriram, R.D., Lu, W.F., and Wang, F.J. (2004), Evaluation and selection for product design for mass customization, vol. V, Chapter 6, Intelligent Knowledge-based Systems: Business and Technology in New Millennium, Cornelius T. Leondes (ed), Kluwer Academic Publishers, USA

Zha, X.F., and Sriram, R.D. (2004), Platform-based product design and development: knowledge support strategy and implementation, vol. I, Chapter 1, Intelligent Knowledge-based Systems: Business and Technology in New Millennium, Cornelius T. Leondes (ed), Kluwer Academic Publishers, USA