Collaborative Product Development and Customization: A Platform-Based Strategy and Implementation

Xuan F Zha

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 research effort on the collaborative product development and customization through a platform-based strategy for collaborative design of product families for mass customization. The implementation of this strategy takes product platform as the core and base, view engine and rule-based control as the data access mechanism, and internet-enabled web-based integration and collaboration bus as an enabler to allow participants involved in product lifecycle to access to both the internal and external enterprise resources, applications, and services. In the paper, a generic interactive process model for collaborative platform design and development for product families and mass customization is proposed. Based on this model, a module-based integrated & distributed collaborative product family design and mass customization scheme 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 high-level information & knowledge modeling and the development of knowledge intensive collaborative support framework and system are addressed. A prototype system for collaborative robot family design and customization is developed. 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, collaborative product customization.

1. Introduction

The current marketplace of product 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 a mass production mode to mass customization to provide the necessary product variety. The issue of moving from mass production to mass customization, or even limited customization, makes many companies struggle to reorganize their product architectures. Products are the key fortune of a manufacturing enterprise. The product innovation drives the increase of profit margin. In recent years, researchers have done a lot of work concerned with product modeling. Although information modeling is increasingly mature, models from the early geometry model, feature-based model to current integrated model, are lack of capability to model dynamic information that implicates product knowledge. They are also lack of capability to model product families to support 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). 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; Tseng and Piller 2003; Zha and Lu 2002a, b). 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 rapidly changing marketplaces and applications. Cost efficiencies, 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, designing family of products, 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 becomes increasingly multi-disciplinary, knowledge intensive, distributed and collaborative (Sriram 2002). 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 process, it is imperative to provide knowledge intensive collaborative support in the product/family design process and share design knowledge among distributed designers. However, 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 issue of how to develop common platforms and how to integrate these distributed and heterogeneous factors under a web based unified platform.

The objective of this paper is to develop a product platform approach and strategy to support collaborative product development and customization. The focus is on the development of a knowledge intensive collaborative design methodology and system for the module-based product platform and family design and customization process. Recognizing the rationale of modular product platform planning and generation for implementing product family for mass customization, the paper discusses the fundamental issues underlying collaborative product platform development and management. A strategic collaborative framework is presented for the planning and generation of product platforms. Also presented are the issues on the knowledge intensive modeling and support for product platform development, and case study in the design of a common micro robotic platform.

2. Overview of the proposed approach: strategy and technology

Mass customization has been identified as a competitive strategy by an increasing number of companies. The challenges of mass customization are the increasingly varied demands of customers and extensively requirements of low cost and short lead-time. More and more enterprise has recognized that to share and reuse the product knowledge within product value chain is essential to win the competition. Therefore, how to model, manage and utilize the product information across the whole product lifecycle effectively and efficiently becomes an emerging issue for companies. PLM is one of the popular ideas to meet such situation, which can be used as a unified approach that provides web-based access to in-house people and partners across the supply chain (Larry 2003).

In this research, the proposed platform-based strategy and approach develop a variety of similar products by enabling the design and production of several related products from a common base for different market segments. Thus, they can direct the
development of information management system for mass customization, which emphasizes on the organization pattern of product data in order to support rapid product derivation. In the proposed strategy and approach, a product platform is built as the core and base. An interactive process is developed for collaborative platform-based design and development: design requirements and models, platform design, variants design, platform evaluation, re-negotiation, and iteration. View engine and rule-based control access strategy are introduced as a powerful data access and security mechanism. An integration & collaboration bus is constructed to allow participants involved in product development across extended enterprise to get easy, web-based access using an unified interface (web browser) to the internal and external enterprise resources, applications, and services. An integrated knowledge intensive framework is developed to support new product derivation and collaborative work across the extended enterprise, under which the organic integration of all the product assets for product lifecycle management (PLM), including components, process, tools, people as well as their relationships, across the whole product lifecycle from conception to recycling to facilitate data access/reuse. In this connection, the following key technologies are used in the developed approach and framework, which include:

1. Product platform modeling technology,
2. Web-enabled integration & collaboration technology,
3. XML-based product data exchange technology, and
4. Data security control technology.

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

3. Collaborative product family design and development process

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

3.1. Collaborative platform design and development

In collaborative product development, different participants and different enterprise information systems are involved in the product lifecycle. How to integrate and collaborate 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, PLM is defined as an integration data and information framework that supports the organic integration of all the product assets, including components, process, tools, people (e.g. customers and designers) as well as their relationships (CRM: customers relationship management), across the whole product lifecycle. Figure 1 illustrates a platform-based strategy for product lifecycle management and development. The features of this strategy are described as follows (Larry 2003):

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 leveled into three tiers: data services, logical business services, and collaborative development services. All the people involved in product development can easily access and reuse the correct and newly information. This allows the product developer to focus and align their users, product knowledge, and business processes for product innovation.

2. View engine and rules as a data access mechanism. 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 product quality and to shorten development
time. Therefore, a mechanism, called information view engine in (Larry 2003), is
needed to view, query, filter and synthesize or mine data. With respect to security,
data access ability should be distinguished according to the users’ roles, thus
access rules should be customized to avoid invalid access and operation of the
product data.

(3) Web-enabled integration and collaboration. Based on a completely open,
standards-based, web-native architecture, every engineer resides in different
locations connected by Internet can share and access product content, application
and services in product lifecycle. Thus, customers and suppliers can directly take
part in the processes of product innovation. Component standards enable users to
plug extended or alternative capabilities offered by other software providers into
their environment.

Figure 1: Platform-based strategy for product Lifecycle management (modified from
web page: http://www.prdm.net/).

Figure 2: Collaborative platform-based product family design for customization
An effective product platform can allow a variety of derivative products to be created more rapidly and easily, with each product family member providing the features and functions desired by a particular market segment. An interactive collaboration process model is proposed for designing product families: design requirements and models (e.g. task or customer requirements, function requirements, and design constraints, etc); platform design; variants design; and platform evaluation, re-negotiation, and iteration (Gonzalez-Zugasti 2000; Urcuyo et al 2003; Piller and Tseng 2003). These are the challenges of cooperation/ collaboration for mass customization success. Figure 2 gives an overview of the interactive collaboration process model applied for product family design and customization. Each of the steps in product family design process can be explored in more detail and will be discussed below.

3.2. Collaborative module-based product family design process

The modularity design concept has been introduced 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 product (Stone et al 2000; Chen et al 1999; Chen 2001; Leger 1999, 2001). Unlike a conventional industrial product that is designed for general tasks, a modular product system has the advantage of providing suitable product configurations for specific tasks. Thus, the modular design of products provides the ability to achieve product variety through the combination and standardization of modules.

Figure 3: Products/families, modules, components and attributes

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 generic product architecture (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 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;
(4) Standardize interfaces to facilitate addition, removal, and substitution of modules; and
(5) Collaborative module-based product platform.

Following the above steps, product 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. The proposed whole modular platform-based product family design process is roughly divided into two main stages, product planning and family design, which ranges from capturing voices of customers or task requirements and market trends for generating product design specifications to customizing products for customers’ satisfaction. The product planning embeds the voices of customers or task requirements into the design objective and then generates product design specifications. The product family design realizes sufficient product varieties- 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.

4. Web-based knowledge intensive collaborative framework for product family design for mass customization

In this section, a modular platform-based product family design using a knowledge intensive collaborative support paradigm is presented.

4.1. Knowledge intensive support framework for product family design

With respect to the proposed modular platform-based family design approach, a knowledge intensive support framework is developed, as illustrated in Figure 4. The developed knowledge support design framework adopts the design with modules, modules network, and knowledge support paradigms.

![Figure 4: Knowledge intensive framework for modular platform-based family design](image)
Product family design knowledge is classified into two categories: product information and knowledge, and process knowledge. These two categories of knowledge are utilized to support two main stages above, i.e., product planning and family design. The knowledge support scheme is implemented through customer or task requirements’ modeling, product architecture modeling, product platform establishment, product family generation, and product variant assessment. Challenges are summarized as follows:

(1) Design information and knowledge modeling: modular product 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 architecture 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;


The product family architecture should represent the conceptual structure and logical organization of product families from viewpoints of both customers (for tasks) 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 instantiates and extends so as to anchor future designs to a common product line structure. 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 it shows in Figure 5.

4.2. Web-based collaborative design framework

A web knowledge-server based distributed module modeling and evaluation framework (WebDMME) was developed 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 (Zha and Du 2002). WebDMME is a web-based collaborative design framework, which adopts the design with modules, modules network, and knowledge server paradigms. The knowledge intensive system can thus exploit the modularity of knowledge-based systems, in that the inference engine and knowledge bases are located on a server computer and the user interface is exported on demand to client computers via network connections (e.g. internet, WWW). Therefore, modules under WebDMME framework are connected together so that they can exchange services to form large integrated models. The module structure of WebDMME leads itself to a client (browser) / knowledge server oriented architecture using distributed object technology. Figure 6 shows the main system components of the proposed client (browser) and knowledge server architecture. Each of these components interact with one another using a communication protocol, 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 must have an interface pointer to the desired interface.

![Figure 6: (a) Client-knowledge server architecture; and (b) Main components for WebDMME](image)

The WebDMME architecture is designed to allow designers or experts to publish and subscribe to 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” product families can be achieved by using the technologies of e-commerce and mass customization to design and set up the mass customized products on the web based on the remote-site customers and task requirements for re-configurable modular systems (Zha 2002).
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 (Phang et al. 1997). Figure 7 illustrates a simple distributed module network model used for design process. The variables of the model are governed by a set of equations. As depicted in Figure 7, the interface connections between variables in different modules (A, B) could be established interactively or defined explicitly using the Model Definition Language (MDL). 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 7a. This distributed module allows users to utilize variables such as Module A (A₁, A₂), Module B (B₁, B₂), Module C (C₁, C₂), and Module AB (B₁, A₂). Figure 7b illustrates the model from the viewpoint of the ABC designer. Module C is local to the designer. Figure 7c 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 7: Simple distributed design model with two modules and a remote module: (a) modules A and B, (b)(c) remote module AB](image)

It is shown that the relations between modules do not need to be changed even if the embedded mathematical model of a remote module (i.e., module AB) is changed. This flexibility enables a designer to define a model independently from the actual location (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 8 illustrates the configuration process for distributed modules using the system components, including the Internet and web 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.

![Diagram of module configurations under WebDMME framework]

**Figure 8: Module configurations under WebDMME framework**

5. Knowledge modeling and support process for collaborative product family design for mass customization

Based on the proposed scheme above, the implementation of knowledge intensive collaborative support for product family design can be achieved through two steps: knowledge modeling and knowledge support for collaborative design process, which is discussed in this section.

5.1. Product family design knowledge modelling

The knowledge modeling process for product design is to elicit design knowledge in the product design process and to establish a comprehensive knowledge repository that can be retrieved and reused when necessary. Product family design starts from a set of customer/functional requirements of the product. The requirements are implemented by a set of modules described in terms of design variables of the product principle (Figure 3). These design variables of a module propagate to the functional requirements on the lower level elements of the module, so on and so forth till to all the modules and element are specified. With respect to the product family design process, three groups of knowledge are required (Du et al. 2001): 1) How to deploy the functions of products (module) to lower level modules; 2) How to select the solutions among the standard ones or the custom ones; and 3) After being selected, all of the solutions have to be configured to be an end product. The performance of each of them has to be estimated to help the decision making of both the designer and the customer.

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 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
data/information and 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 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 product repository may be extensively composed of functions, means, structures, features library, modules library, types, attributes, relationships, rules, constraints, evaluation/selection criteria, etc.

Figure 9: Modular product platform construction process, representation and reuse (Ref: Du et al 2001)

Product Definition:
- Customer/Task Requirements;
- Specifications;
- Functions-Behaviors-Structures;
- Performance objectives and constraints;

Product Variety:
- Assembly Structure;
- Module Details;
- Family Parameters;

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 (Gorti et al 1998), 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, 2001). Therefore, a multi-level hybrid representation schema 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. 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: 1) generic product specifications, and 2) 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 9 illustrates the construction process (Steps 1-4) of product platform and the reuse for domain-specific applications (Step 5), e.g., robot family. For illustration, an object-oriented representation instance for robot family and its parameterized module information (e.g. link and joint modules) is described as follows (Zha 2002):

Module (“Joint module”) {
    Number of 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: [mas, center of mass, inertial];
}.

The description model for a modular product design process with two modules A and B can be illustrated as follows:

Module (“Module A”){
    Variable "A_1" ();
    Variable "A_2" ();
    Dependency "a_1";
    Dependency "b_2";
    EmbeddedModel “calculateA_2” ();
}

Module (“Module B”) {
    Variable "B_1" ();
    Variable "B_2" ();
    Dependency "b_1";
    EmbeddedModel “calculateB_2” ();
}

Design (“Product family”) {
    Module (“Module A”);
    Module (“Module B”);
}
5.2. Knowledge supported collaborative design process

Once the product design knowledge repository is built up, the user/designer can utilize the knowledge in it 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, 2001). Normally, a product/family (e.g. robot family) is assembled through the use of fixed types of modules in module inventory (repository) while maintaining re-configurability and inter-operability. Thus, the design process is concentrating mainly on the following tasks:

1. Re-configurable and “plug-and-play” kinematics and dynamics model. To ensure rapid deployment and unified control operation of products with different structures and DOFs (degree of freedoms).

2. Product configuration optimization. To establish optimization model for the rapidly configurable system based on an appropriate selection of module components according to the task/customer requirements.

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

4. Simulation for system. CAD-featured environment for the visualization, evaluation and simulation of the re-configurable system, and provide 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., A1Ms, transformation matrices, etc. The interface connections between variables in different modules could be established interactively or defined explicitly using the Model Definition Language (MDL). 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 full knowledge support process for product family design can be fulfilled through 11 steps discussed in (Zha and Lu 2002a; Zha 2002). For collaborative product family design process, customers and designers or designers and designers should collaborate each other in each or some of these steps. As an open environment, the developed framework can facilitate the implementation of collaborations.

6. Prototype system and case study

6.1 Prototype collaborative robot customisation 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 web-based collaborative environment, WebRobot, is implemented under the WebDMME framework with concurrent integration of multiple cooperative knowledge sources and software. WebRobot is a comprehensive suite of scalable and configurable software tools. It is therefore an integrated distributed collaborative development environment, consisting of 3D design, modeling and simulation software tools, which enable the collaborative creation of robot families.

Based on the proposed distributed models, WebRobot can offer uniform model construction effort (e.g. kinematics, dynamics and calibration (Chen et al. 1999)) across computer simulation and real-time control of arbitrary robot configurations (Dalton and
Taylor 2000). WebRobot is intended to be a uniform interface for all modular robots and is portable to modular product systems from different vendors. It can be used both for simulation and for on-line execution of a task, regardless of whether the robot is executing or is simulated to be executing the task as a stand-alone application, or as part of a workcell process. Thus, it allows the user to quickly integrate the hardware components into modular robots, and to manage their operations in the reconfigurable system collaboratively. Key features of WebRobot include: 1) Internet/intranet enabled and WWW-based; 2) Module and product/family builder; 3) Collaborative configuration/reconfiguration, 4) 3D graphical task collaboration and simulation, 5) Kinematics and dynamics models, 6) Trajectory and task planning, and 7) Task execution and collaboration.

The system can provide distributed users access to WebDMME module servers throughout the network over the internet/intranet and WWW. With this system, designers in different teams and organizations may participate and collaborate in the design process of robots. On the other hand, the web-based system can provide end-user an optimized robot 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 WebRobot 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; Zha 2002). The underlying framework modules, knowledge support engines and collaborative mechanisms are written in Java™. 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 robot configuration synthesis and optimization.

6.2 Case study

In this section, a distributed design model is built for collaborative micro-robotic system family design, and the developed system WebRobot is used as a collaborative design and customization platform. Suppose that customers, designers 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 components: 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 illustrated in Figure 10, in which the 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 (Zha and Du 2002). 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. 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.
Figure 10: Problem topology of micro-robotic system design

Figure 11: (a) Product modules (Chen 2001), (b) Module parameterization (Leger 2001), (c) Assembly / configuration, and (d) Robot families
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. Figure 12 shows screen snapshots of the web-based collaborative robotic system family design: (a) the robot platform session; (b) the robot family design session; and (c) the microscope design session.

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 design of product families for mass customization. The proposed collaborative modular platform-based product family design scheme integrates and collaborates distributed customer or 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, organizing, and 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 model needs to be decomposed explicitly enough to make it easier and more flexible to implement. The prototype system still needs to be further developed to test the full approach. When fully developed, the system can effectively and efficiently support collaborative product family design and improve customer satisfaction.

Disclaimer

The bulk of this work was done while the author was at Singapore Institute of Manufacturing Technology, Singapore. 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.

References


