Web Service-oriented Electronic Catalogs for Product Customization

Yongsheng Ma, Jianxin Jiao and Yimin Deng

Concurrent Engineering 2008; 16; 263
DOI: 10.1177/1063293X08100026

CONCURRENT ENGINEERING: Research and Applications

Web Service-oriented Electronic Catalogs for Product Customization

Yongsheng Ma,[1,*] Jianxin (Roger) Jiao[2] and Yimin Deng[3]

[1]*Department of Mechanical Engineering, University of Alberta, Canada*
[2]*Woodruff School of Mechanical Engineering, Georgia Institute of Technology, USA*
[3]*Faculty of Engineering, Ningbo University, China*

Abstract: This article is about online product customization, which potentially connects multiple customers and suppliers throughout an entire supply chain. This article proposes a web service-oriented approach, such that users can source competitive offers and generate feature-based CAD models from different vendors over the Internet. The key technique is to create feature markup descriptions for the requested library elements. Group technology is applied to classify and control the geometry configurations of component models. The approach enhances traditional un-parameterized and platform-dependent electronic catalogs by avoiding hard coding of catalogs within specific CAD systems, and thereby facilitates the updating of databases.

Key Words: mass customization, product catalog, web service, product development.

1. Introduction

Standard component libraries are very useful for reusing design knowledge and engineering data, and reducing product development time and cost. Recently, suppliers provided electronic catalogs with certain CAD models in the form of files, and deliver on disks. Via internet web portals, there are electronic catalogs performing as the front-end of the company's interactive interface to online customers [1]. These electronic catalogs are supposed to be permanently up-to-date, and enable integration of contents (i.e., access to product information) and direct communication between suppliers and customers [2]. An electronic catalog involves characteristics of both the technology and related business practices. Electronic catalogs are still in their infancy. A common problem of current practice is the unstructured product information and overloaded sites with no navigation support [2]. Moreover, the traditional approach treats every product variant as a separate one; so it does not work when customers are granted a high degree of freedom for specifying products. In addition, by this approach, the cost is postponed to the supplier side *per se* [3]. For a more advanced product lifecycle management strategy in a collaborative fashion, Chung and Lee [4] suggest storing all the data and CAD models in a federated relational database system that is interfaced with users via web sites. However, the CAD models downloaded are un-parameterized or 'hard-coded' allowing very limited editing functions [5]. Due to the exponential number of configurations, the number of CAD files is enormous. It is also difficult to support different suppliers due to numerous variations related to component definitions. Updating and maintaining individual CAD models turn to be a huge task requiring continuous investment. Some of them lend to be a 'CAD system bundling approach.' Major drawbacks of such a practice are observed as: (1) suppliers must rely on the CAD vendors or developers to update their catalogs, and thus have no control; (2) CAD vendors implement only the products of major suppliers but not the small or niche ones, and furthermore, they can implement very limited standard products from different catalogs; (3) since it is the CAD vendors who implement the catalogs into their CAD systems, business information regarding to design selection, ordering forms, prices, delivery terms, etc, is unavoidably striped off; (4) each CAD vendor develops its own standard component or assembly module libraries within its legacy CAD platform. This limits the implementation of those designers who do not use the same CAD systems; and (5) if end users do not upgrade their CAD system versions, they may have to work with the outdated libraries. Online mass customization involves understanding accurately customer needs and subsequently creating a complete description of a product variant that meets those needs [6,7]. Such a configure-to-order process assumes that a valid design is assembled from instances of a fixed set of well-defined component

*Author to whom correspondence should be addressed.
E-mail: yongsheng.ma@ualberta.ca
Figures 1 and 4 appear in color online: http://cer.sagepub.com

1063-293X/08/04 0263–8 $10.00/0 DOI: 10.1177/1063293X08100026

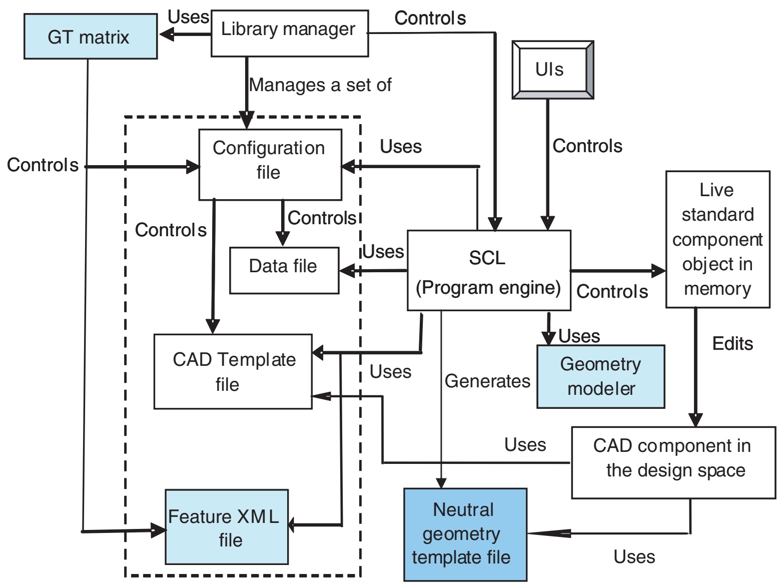

Figure 1. QuickMould SCL components.

types (e.g., component catalogs), in which components are interconnected in a predefined manner to meet a set of customer requirements subject to certain design constraints. Hundreds or thousands of configurable components and modules may be involved, which often causes errors during configuration. Such errors can create major slips in schedules and lead to costly iterations in phases further downstream.

Web service enables a standardized mechanism of integrating web-based applications using open standards over an Internet protocol backbone. Web services do not depend on a particular GUI package [8]. Web services share business logics, data and work flow processes through a programmable interface across networks [9]. The developers can associate the web service with any client GUI such as a web page or an application program to offer specific functionality to the users [10]. This mechanism enables interoperability among different applications. Therefore, web services have been well recognized as the basis for electronic commerce as well as engineering [11,12]. Companies can invoke the services of other companies to accomplish a business transaction [13]. To date, few web service applications are available for CAD collaboration [14]. This article investigates the application of the web service technology to the development of reference architecture for electronic catalogs in order to provide flexible feature-based CAD models in online product customization systems.

2. Application Case: QuickMould

This research is an extension of the implemented standard component library (SCL) and mold base library (MBL), which are two functional modules [10,15] of *QuickMould*. In QuickMould, the CAD model of each parametric element of SCL consists of three related files, i.e., a predefined CAD template, a configuration file, and a data file, as shown in Figure 1 (excluding shaded blocks). A predefined CAD template for each element defines the geometry of the standard component in a parametric manner. The configuration file defines the properties of different combinations of alterations, where key parameters can be edited through user interfaces. This file also contains component types with their identification code, name, size, and enables design alterations and specification of parameter ranges, parameter constraints, global parameters, as well as additional attributes such as available material, default delivery time, prices, etc. The data file contains the parameter values for each standard component to update the key expressions. The format of this data file is corresponding to the configuration file. Object-based methods are employed to retrieve data from these files. If the user wants to change from one type of standard components to another, then the current part will be closed first and the newly selected component template will be loaded to replace the previous one. The information representation models for standard components and assembly modules are largely inherited from each other. However, the QuickMould SCL implementation was standalone within a single CAD environment and not for collaborative CAD applications across the network. The limit is that an interactively predefined template is required for every type of standard components. Such templates are specific to individual CAD packages and thus rigid for collaborative modifications. This requirement is a hurdle for collaborative engineering. To overcome this hindrance, this research adopted a generative approach, to be elaborated below.

3. GT-based CAD Model Generation

The general gist of electronic catalogs is to represent all standard components or assembly modules with an object structure, or *class* in object technology, and all the instances of a component class can be managed by the corresponding service methods. As discussed in the previous sections, the system design and implementation mechanisms are aimed to enable web service structure and interoperability. A generative approach is proposed here using feature-based constructive solid geometry (CSG) method, which creates component geometry with a combination of feature solid primitives, such as cuboids, cylinders, spheres, cones, etc. The set of operations in CSG method include *union* (encompassing all points in space that are contained in the primitives A OR B), *intersection* (encompassing all points contained in A AND B), and *difference* (encompassing all point in shape contained in AB), etc. Almost all component geometries are generally a combination of simple primitives, so is an ejector pin, which has a basic primitive of a cylinder (Figure 2). In a component, all feature primitives have topological relations to one another. Such topological relations are modeled in the form of a graph. Geometry deviations and alterations can be assigned to the model accordingly. Other relations, such as dimensions and tolerances are also modeled in modular manner in which they can be configured accordingly for every supplier.

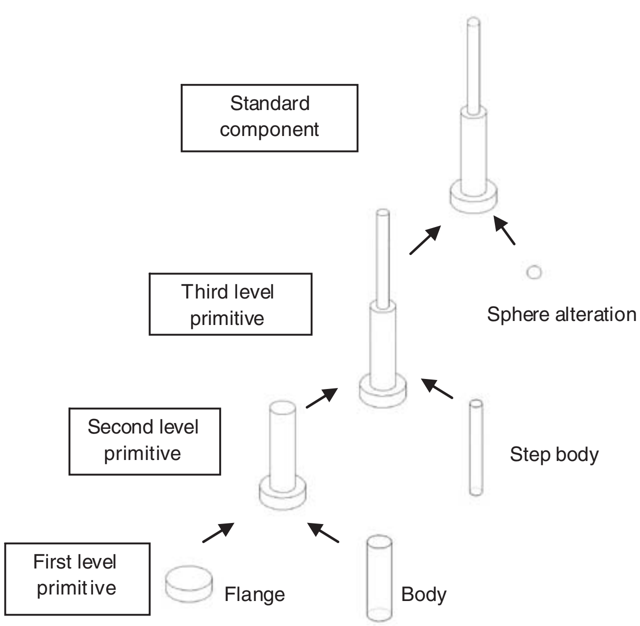

Figure 2. Constructive solid geometry of a stepped ejector pin.

To effectively achieve the reusability of feature primitives, their classification and coding become essential. Group technology (GT) principles have been applied in CAD/CAM/CAPP many applications. In this research, the GT method was adopted in a simplified manner, whereby the primitives of standard components from different supplier catalogs are modeled in a matrix with particular reference to a classification coding system. Such a matrix scheme takes advantage of the similarities in constructing the final geometrical model of mechanical components. Figure 3 illustrates the application of GT matrix in the designed system functions. One of the roles played by this GT matrix is to control the geometry construction methods. Geometry generation methods are organized in a modular manner. A generation control program recognizes all the features according to the code input by the user. Therefore, the program knows what functions to call in order to generate the standard component model. It is recommended that feature classification and coding should be standardized as much as possible, and should allow customization. For a component, its GT matrix can also classify and code other attributes, such as dimensions, tolerances, and combinations of primitives.

After user has selected necessary options, the component configuration is represented with a classification code. The corresponding attributes with selected values are retrieved from the configuration and data files. The corresponding object is then established. Next, the code is matched with the GT matrix. The object is evaluated according to the interpretation of the code and the constraints in the program. Then the geometry is generated. For each library element, a feature-based geometry description file in XML format is used as the neutral transaction media in collaboration transactions.

The novelty of this work is that, the application program creates the geometry of a component model with a CAD modeler via API functions by linking CAD modeler's 'DLL' modules and access functional routines within the standard component/assembly library system. One advantage to this is that the executables are much smaller and they can be compiled instantly. Once the

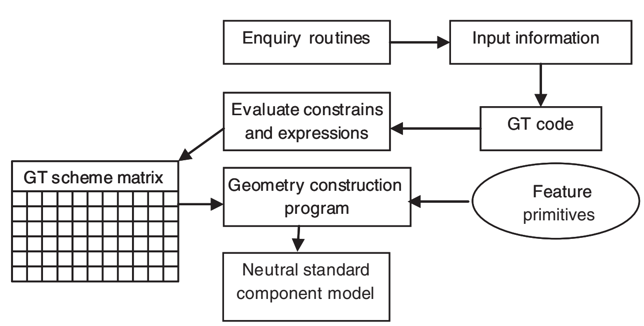

Figure 3. Implementation of GT into component feature configuration.

library application program is loaded into memory, it stays for the remainder of the CAD session. Then it runs interactively, giving the user opportunities to see the results in the design space with the standard components or assembly modules. The geometry creations of the model are composed directly by the program. By doing so, predefined part templates are eliminated. Users are now able to create standard component models without the limitation of their predefined templates.

The shaded blocks of Figure 1 show the extended design of new SCL. The new design makes use original QuickMould SCL data structure representation, which is flexible enough for many companies' standard components. The new SCL design using the GT method offers some advantages in comparison with the configuration file of the previous design. Firstly, the GT method is generic in controlling the geometry creation of the standard component. Secondly, this GT method can cater more than just one type of standard component. The classification code for a component can be used for indexing, searching and organizing modular methods. However, matching supplier-specific dimensions to the generic generation parameters needs customization effort.

4. Web Service-oriented Catalog System Architecture

To avoid the drawbacks of the existing standard component/product libraries, the web service architecture has been proposed [15,16]. The main advantage of this web service approach is that the end user or web service provider can access many catalogs from different suppliers via a uniform platform. At the same time, a new web catalog service can be created by connecting the suppliers according to a common protocol. Customers could compare the specifications, prices, quality, etc. for a required component. This will provide a fair opportunity for OEMs as well as suppliers. The system is divided into three major modules: (1) End user (client) module in which user applications, interfaces, transactions, uploading and down-loading functions, are hosted, and also called request module. (2) Web service module, in which i the portal server module, is implemented with Java Web Services technology. (3) CAD application module that is integrated with a solid modeler and provides fine-grain geometric services and data processing capability. These three modules are further discussed in detail in the following sub-sections.

4.1 End User (Client) Module

This module generates the request code through the end user's selection. Basically, all the catalog elements are generated with the reference catalog numbers, each of which is the key for the customer to order according to their requirement. This module generates the catalog number as request key. The catalog selection and catalog key generation are processed through the server. The implementation can be done with simple JavaScript, HTML programming languages and ActiveX components. For the actual implementation, developers can use the Java Server Pages (JSP) and Java bean technology to connect the client selection and data retrieval from database through Java programming.

4.2 Web Server Module

For this research, the Appache Tomcat web server is selected. Tomcat is the Servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. Tomcat is developed in an open and participatory environment and released under the Apache Software License. In the proposed system, catalog providers publish their services and WSDL information in the UDDI registry under the category of standard components for a industrial sector. The end users (service requesters) discover the above service by using the UDDI registry with the client application, or manually, and uses the published WSDL to generate the client proxy. At runtime the client uses the client proxy to construct and send SOAP messages to the web service. The process sequence of the system upon reception of the end user's requests has been reported in [15]. One of the important components in this system is the database. All the catalog data have to be kept in the database for the electronic catalogs. Since enormous amount of research and development have been done in this area, this research does not focus on this area and a small database had been developed for testing purpose only.

4.3 CAD Application Module

The previous generative library system structure [15] has to be modified into client and server parts to be functional over the web. Instead of using a specific CAD system to generate standard component geometry, a neutral CAD modeler, such as ParaSolid, ACIS, or STEP-based 3D solid modeler, etc. is suggested. This is for the necessary interoperability. Most of the major CAD packages are gradually merging their neutral data formats and will enhance the translation functions due to the product data management (PDM) or product lifecycle management (PLM) requirements. For the actual implementation, catalog vendors can use any 3D modeler according to their requirements [5,10]. There is a feasible solution to achieve feature-level uniformity for CAD model generation by using a neutral feature operation scheme as reported in [17,18]. For this research, the ACIS 3D CAD modeler [19] has been chosen simply for the availability of research licenses. ACIS has all the necessary basic objects available including NURBS entities with functions [19], and user can easily to create their required CAD objects. As such, once the geometry generation instructions are received from web service in the form of XML data, by using ACIS classes and functions, a generic SCL generation 'engine' can be created, which is designed and implemented to generate ACIS 3D files. Such geometry entities are then inserted into the end user's CAD system. It can be seen that, when the end user sends the catalog request key to the web server, it process internally to find the configuration and parameter values from the database and return as a feature XML file, as shown in a shaded block at the left-bottom corner of Figure 1, to the client (end user) side. The returned XML SOAP contains this feature XML file with related data to generate the CAD object and there should be an application program to handle this XML data to CAD object conversion. Presently, the SCL engine is installed in the client side and the received XML message is processed with the functions of ACIS modeler. The CAD modeler sequence of the system has been illustrated in Ma [15].

5. System Implementation

The system implementation takes into account the major aspects discussed in the previous sections. Several software modules, including web service module development and deployment, web server deployment, user interfaces, database system, and CAD modeler application and integration, have been prototyped. As mentioned before, Java technology has been used to create the client/server system and also the web services. The programming language used is Java II standard edition JDK 1.4.2, containing all the basic Java libraries and tools. The Java Web Services Developer Pack, version 1.3, is used for web service development. It is an integrated toolkit that allows Java developers to build, test and deploy XML applications, web services, and web applications. Web Server and Servelets are developed with Tomcat 5.0. Servlet/JSP container for Java Web Services.

5.1 Web Service System Components

The system architecture can be simplified as three layers. The upper layer is an Interface layer, i.e., 'Data.jdbcIF', in which there are direct SOAP interfaces. The interface classes for SOAP calls represent the gateway through which the main functionality offered by this catalog service system is supported. The middle layer, i.e., 'Data.src', implements business logic for catalog service. The lower layer is the database of the application. The package named 'Data' providing all the low level data transaction functionality needed for this system. The components on the server and web service transactions during each web service cycle has been reported [15]. Once the web service request is received by the server, it will create the required Enterprise JavaBean (EJB), which then package the request and responding instructions into Java Archive Registry (JAR) or Web Archive Registry (WAR) format before processing. The instructions deploy a server that generates and initiates a service object accordingly; and at the same time, service interface is created such that a functional transaction server is ready with server ties. Via SOAP messages, client stubs are then created accordingly. At this stage, a service process has been established. Transactions could be in both ways; the functions include displaying information, making enquiry about components, requesting and delivering CAD models, etc.

5.2 CAD Libraries

In this research work, the challenge is to generate common standard component or assembly geometry with parametric feature primitives and their relations in an object-oriented and generic approach. Self-contained and consistent objects have to be defined for different library components. This is due to the fact that knowledge-based design procedures need to be taken into consideration; as it could help users to solve problems that require significant human expertise for their solutions. The library functions should allow users to create, edit, query, and destruct the standard components. The library elements need to be easily developed and extended since the catalogs of suppliers will be updated regularly while in-house-made components are to be incorporated too. The library elements are required to be information rich such that all the information of each standard component can be easily extracted and made available to users at different levels and for different applications.

The prototype has been developed to evaluate the proposed new method. Ejector pins in plastic mold design have been chosen as the component type for prototyping. They have the basic shape of cylinders and one of the most common types of standard components available in all the supplier's catalogs. Furthermore, the proposed solution can be extended for other pin types such as core pins and center pins due to the similarity in shapes. When creating the element geometry models of standard components, the program captures the user's selections with attributes and translates them into the form of GT codes, which are mapped to the generation program branches, indicated as 'styles.'. The implementation has showed that it is efficient using object-oriented feature generation methods to implement GT-based geometry generation. However, this prototype needs some enhancements. First, Generic UIs with consistent object definitions are needed to display multiple selections (supplier, category, major type, subtype, and alterations) in a dynamic configurable manner [10,20]. Secondly, this program lacks the capability to search for replacement or next-to-exact-fit components based on user selection through intelligent matching and evaluation. The main contribution of this work is the ability to generate part model systematically from a feature-based XML product description without any predefined template. The prototype system generates models based on the CSG method by calling all the primitive functions. In fact, these functions can be distributed across networks via SOAP messages and dynamically linked by the SCL generation engine over the client side to support the CAD model generation transparently.

6. Application of Mold Design Libraries

An intelligent mold design prototype system developed [10] emphasizes on the information flow and application integration. Its user interfaces could provide instant information such as the available product types/variants, and sizes; the picture of the product alongside with its master parametric dimensions; configurable parameters; prices, etc. If user selects a type of components from a supplier whose catalogs have been incorporated in the SCL, a GT code is then generated according to the predefined scheme/matrices. The program engine will retrieve the configuration file and the data file for the respective component, and display the available configuration options and key dimensions [10]. Then the object representing this component is established; the CAD model is then created. Examples are shown in Ma [15] and Ma et al. [16]. When updating the CAD model according to the values stored in the data file, only the input parameters for the generative routines, which are used for the generation functions for the component features, are updated. The lower level feature models are subsequently recreated automatically with a geometry construction reference mechanism. All parameters that are defined in different components can be modified via a set of consistent user interfaces [10]. Some standard components have dimension ranges and constraints that limit the value of certain key parameters. Such information is also modeled as constraints and contained in the configuration and data files. After selecting the standard component supplier, type, and size, the program engine will enrich the live standard component object in the design space (memory). Then the library will update the geometric model retrieved with respect to the data that the user has already selected. The method introduced can be extended to assembly configurations as well. As introduced by Ma [15], modular feature-based assemblies can be modeled into libraries based on different suppliers' product catalogs. Figure 4 shows the mold base assembly structure expressed in 'AND–OR' graph while mold bases generated based on QuickMould assembly features can be found in a published work [20].

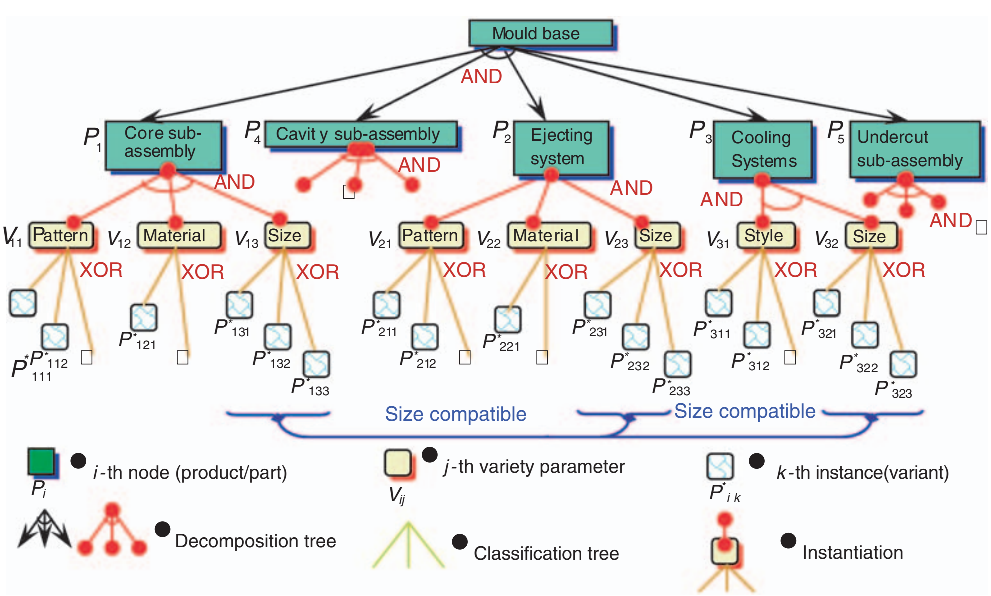

Figure 4. Mold base structures represented with AND–OR graph.

7. Conclusions

Web-based catalogs for standard component and assembly libraries need to be enhanced in terms of coherency. This article shows that a web service-oriented 3D parametric SCL can effectively host the required services from different suppliers. Standard components are generated from CSG feature primitives with modular generic methods. GT is applied to manage the range of geometry configurations of component models. The novelty of the proposed approach is the feature-based product representation and the modular generative processing method of a CAD modeler. These techniques constitute the foundation for implementing a neutral feature markup language necessary for virtual enterprise integration. The web service architecture improves the current practice by eliminating the need for rigid, cumbersome predefined CAD templates.

References

1. Hsu, C.-C. and Chen, Y.-C. (2007). Mining of Mixed Data with Application to Catalog Marketing, *Expert Systems with Applications*, **32**(1): 12–23.
2. Stanoevska-Slabeva, K. and Schmid, B. (2000). Internet Electronic Product Catalogs: An Approach Beyond Simple Keywords and Multimedia, *Computer Networks*, **32**(6): 701–715.
3. Li, Y., Lu, Y., Liao, W. and Lin, Z. (2006). Representation and Share of Part Feature Information in Web-based Parts Library, *Expert Systems with Applications*, **31**(4): 697–704.
4. Chung, J. and Lee, K. (2002). A Framework of Collaborative Design Environment for Injection Moulding, *Computers in Industry*, **47**(3): 319–337.
5. Raghothama, S. and Shapiro, V. (2002). Topological Framework for Part Families, In: *Proceedings of ACM SM*'02, Saarbrucken, Germany, 17–21 June.
6. Jiao, J. and Helander, M. (2006). Development of an Electronic Configure-to-Order Platform for Customized Product Development, *Computers in Industry*, **57**(3): 231–244.
7. Zhang, Y. and Jiao, J. (2007). An Associative Classification-based Recommendation System for Personalization in B2C E-Commerce Applications, *Expert Systems with Applications*, **33**(2): 357–367.
8. Younas, M., Awan, I. and Duce, D. (2006). An Efficient Composition of Web Services with Active Network Support, *Expert Systems with Applications*, **31**(4): 859–869.
9. Wang, S., Shen, W. and Hao, Q. (2006). An Agent-based Web Service Workflow Model for Inter-enterprise Collaboration, *Expert Systems with Applications*, **31**(4): 859–869.
10. Ma, Y.-S., Tor, S.B. and Britton, G.A. (2003). The Development of a Standard Component Library for Plastic Injection Mould Design Using an Object Oriented Approach, *International Journal of Advanced Manufacturing Technology*, **22**: 611–618.
11. Domenico, B., Francesco, G., Stefania, M. and Maurizio, V. (2004). A Web Service Based Framework for the Semantic Mapping amongst Product Classification, *Journal of Electronic Commerce Research*, **5**: 114–127.
12. Davies, N.J., Fensel, D. and Richardson, M. (2004). The Future of Web Services, *BT Technology Journal*, **22**(1): 118–130.
13. Calladine, J. (2004). Giving Legs to the Legacy — Web Services Integration within the Enterprise, *BT Technology Journal*, **22**(1): 87–98.
14. Lee, C.K.M., Ho, G.T.S., Lau, H.C.W. and Yu, K.M. (2006). A Dynamic Information Schema for Supporting Product Lifecycle Management, *Expert Systems with Applications*, **31**(1): 30–40.
15. Ma, Y.-S. (2006). Web Service-Oriented Standard Product Library, *International Journal of Agile Manufacturing*, **9**(2): 65–72.
16. Ma, Y.-S., Rajeshbabu, T.S. and Deng, Y.M. (2007b). Web Service Oriented Standard Product Library, *International Journal of Electronic Business Management*, **5**(2): 105–104.
17. Xue, D. and Dong, Z. (1997). Coding and Clustering of Design and Manufacturing Features for Concurrent Design, *Computers in Industry*, **34**(1): 139–153.
18. Bidarra, R. and Bronsvoort, W.F. (2000). Semantic Feature Modeling, *Computer-Aided Design*, **32**: 201–225.
19. Spatial Corp. (2001). ACIS, 3D Geometric Modeling 6.3.1 Objects (ACSCD1631).
20. Ma, Y.-S., Britton, G.A., Tor, S.B. and Jin, L.-Y. (2007a). Associative Assembly Design Features: Concept, Implementation and Application, *International Journal of Advanced Manufacturing Technology*, **32**(5–6): 434–444.

Dr Yongsheng Ma

Dr Yongsheng Ma is an Associate Professor in Department of Mechanical Engineering, University of Alberta. Between 2000 to September 2007, he was with the school of Mechanical and Aerospace Engineering, Nanyang Technological University. His main research areas include product lifecycle management, feature-based product, and process modeling. Dr Ma received his BEng from Tsing Hua University, Beijing (1986), MSc and PhD degrees from the University of Manchester, UK in 1990 and 1994, respectively. He had worked for Ngee Ann Polytechnic and the Singapore Institute of Manufacturing Technology before 2000.

Dr Jianxin (Roger) Jiao

Dr Jianxin (Roger) Jiao is currently an Associate Professor of Computer-Aided Engineering and Design in the Woodruff School of Mechanical Engineering, Georgia Institute of Technology, USA. Prior to joining Georgia Tech, he was an Associate Professor of Systems and Engineering Management in the School of Mechanical and Aerospace Engineering at Nanyang Technological University, Singapore. His research interests include mass customization, design theory and methodology, reconfigurable manufacturing systems, engineering logistics, and intelligent systems.

Dr Yimin Deng

Dr Yimin Deng is a Professor of Ningbo University, Faculty of Engineering, Ningbo, Zhejiang, China. His research areas include design theory and methodology, plastic injection molding optimization, CAD/CAM, smart materials and structures.