A REVIEW ON FEATURE-BASED PRODUCT REPOSITORY FOR COLLABORATIVE ENGINEERING

Y. –S. Ma[a], S. -H. Tang[b], M. M. Uddin[a]
[a] Dept. of Mechanical Engineering, University of Alberta, Canada
[b] Dept. of Mechanical Engineering, Guangdong University of Technology, China

Abstract
Product and process information integration and sharing throughout different stages of a product lifecycle is crucial to achieve collaborative engineering. A unified repository system that supports various CAx application tools is required. The ideal repository system must be able to store and maintain comprehensive engineering information. Advanced feature models implemented in a database system provide a potential solution. This paper reviews the merits and demerits of these technologies and emphasizes on the adoption of feature-oriented approach in database design. The repository system is expected to maintain the consistency and validity of information while allowing exchange of information among different systems. Since the interoperability among different CAx application systems is a bottleneck, this paper also discusses a generic feature representation schema intended to realize this interoperability. The purpose of this review is to identify technological gaps and find out promising research directions for future studies.

Keywords: Collaborative engineering; Feature technology; Interoperability; Repository

1. Introduction

Product and process development determines the long-term viability of companies and even the related economies [65]. Generally, it consists of several phases ranging from the analysis of customer requirements to the adoption of products and processes by manufacturing, namely conceptual design [43], technical design, detail design [41] and manufacturing process development [66, 67]. During the past decades, computer-based tools have been introduced in industry to perform technical tasks such as drafting, design [63, 70, 77], process planning [15] and control [61], and quality assurance. These tasks are usually done independently in a serial fashion, known as “throwing it over the wall” approach. This approach possesses the shortcomings of higher costs and longer development cycles due to unavoidable iterative backtrackings and modifications. The competition in the global market has forced companies to innovate products with the highest quality, at the lowest price [13, 21, 23]; and more importantly, shortening time-to-market will extend the effective product selling period, increase market share and achieve price priority [49, 51]. Therefore the concept of collaborative engineering was introduced.

Collaborative engineering is the application of team-collaboration practice to an organization’s development endeavors. It is built upon a computerized platform for cross-functional, highly effective and well-supported engineering decision making [38, 71, 72, 77]. Although collaborative engineering can offer substantial benefits, it is not yet clear how it can best be implemented. This approach requires an effective and efficient information repository system [7, 14, 33, 45, 57] which can integrate all the data of product lifecycle to achieve interoperability among different CAx systems. This repository management system required has to maintain the consistency and validity of product, process and management data in a concise manner. However, existing systems suffer from difficulties to fulfill these requirements.

This review begins with a section of “Feature technology” for modeling semantic product and process information, and then followed by section “Database technology”, where the trends and requirements of multi-application information sharing through database and database management systems is discussed. After that, a section of “Feature based repository” covers a newly proposed informatics approach to design an integrated repository for different CAx applications, including feature modeling, system design, and relations with geometrical modeling kernel. Next, a section of “Research issues and challenges” analyzes the limitations of the existing repository designs and highlights the new research directions in future. The “Conclusions” section summarizes this paper.

2. Feature technology

This section covers the recent works done on feature modeling, constraint solving and information sharing.

2.1 Feature definition

Features are originated in the reasoning processes and applied in various design and manufacturing activities. They were used to associate functional information with shape information. In 1980s, a feature was defined as a physical constituent of a part that is characterized as a generic shape having engineering significance and predictable properties [58]. In the early stages of feature technology development, features were usually predefined as parametric templates, which have precise geometries for feature-based modeling systems to initiate a feature object or for feature recognizers using pattern matching to recognize an application feature. For example, a feature was represented as a fixed attributed adjacency graph. In such definitions, only feature syntax, i.e. the topological and geometric relationships between different geometric entities of a feature are precisely predefined and these relationships are usually fixed [58]. Based on such fixed templates, some standards for exchanging product information among different CAx applications have been developed, including IGES, VDAFS, SET and STEP [27]. These standards focus on lower-level geometrical data. STEP has been extended to cover feature information, but it is limited to form features [29]. Those feature templates have two limitations. Firstly, they lack the flexibility to be extended. Secondly, the lack of specifications of feature semantics may result in information consistency breakdowns [52]. Bidarra, *et al*. [4] explained in detail the semantic problem suffered by most feature modeling systems, e.g. ill-defined feature semantics and lack of semantic maintenance; and proposed a semantic feature modeling approach based on cellular topology.

Currently, most of the CAx systems are parametric and feature-based. The advantage of using features in product development comes from the abstraction of feature information that retains not only the associative geometric information [58] but also much of the useful non-geometric information that reflects engineering semantics within different applications [18]. Therefore, higher-level feature information must be represented and associated such that engineering meaning is fully maintained and shared among different users. This idea evolves into the concept of associative features where a feature is a set of related entities of characteristics across components, assemblies and stages of applications; and relations among feature elements are modeled, implemented, evaluated and updated comprehensively throughout a feature's lifecycle [44].

2.2 Modeling and solving constraint

One of the main tools on which feature-based modeling relies is constraint modeling [Hoffmann 2002]. Currently, almost all feature-based modeling systems use constraints to specify relationships among geometrical entities, topological entities and features. A *constraint* was traditionally defined as a specification of a relation that should be hold and associated to one or more entities or constraint variables. Constraint may have various types, such as geometric, algebraic, dimensional, semantic constraint. Geometric constraints specify the geometric relations between feature elements. Algebraic constraints define the relations among feature parameters using equations. Dimensional constraints specify distances between two feature entities. Semantic constraints specify topologic properties of feature elements. Bettig and Shah gave a classification of geometric constraints for the shape definition in CAD [3]. A constraint solving mechanism aims at determining whether it is possible to satisfy a set of constraints, and if it is possible, then assign the values to constrained variables. In CAD domain, Kumar *et al*. [36] used sequential approach to impose geometrical constraints to positioning rigid bodies during feature creation as well as assembly planning. Li *et al.* [42] used a DOF (degree of freedom)-based constructive approach to solve geometric constraints, which can also handle well-constrained, over-constrained and under-constrained situation on the basis of dependency analysis. Lee *et al.* [37] proposed a graph constructive approach which can handle ruler-and-compass non-constructible configurations, and under-constrained problems.

Ma *et al.* suggested that there are two methods to deal with constraint modeling issue, i.e. procedural or declarative ones [44]. The major difference of these two methods is that, in a procedural method, specifications of feature properties and procedures for manipulations of a feature instance are combined together in codes and runtime procedures while in declarative constraint modeling method feature properties and their manipulations are decoupled. Object-oriented (OO) feature representations are typically declarative. A merit of declarative feature definition is that more modularity is achieved by the separation of feature definitions and feature instantiations. Further more, in OO approach, the class definition with properties and methods' protocols are separate from the implementation routine functions; hence, a unified system independent feature implementation is possible [9]. Usually, constraints defined are limited within one application feature model; constraint integration of different applications has not been solved.

2.3 Feature information sharing

Traditionally, during data exchange, higher-level feature information loss occurs and only pure geometric data can be converted among different applications, although feature extraction and identification tools can partially recognize some feature information [17, 20, 25]. For example, feature relationships (constraints) cannot be recovered from the geometric data model. Some researchers [22, 30, 40] proposed to use design information as the input to derive downstream application models by feature conversion. However, their works support only one-way link which create change propagation and consistency problems. A multi-view feature modeling approach [6] was suggested that can support multi-way feature referencing by feature links, and then an "associative feature" definition was developed in [44] for establishing built-in links among related geometric entities of application-specific and multi-facet features with self-validation methods. A unified feature modeling method was recently proposed by Chen *et al.* [9], where a generic semantic feature model for different CAx applications covering three-level relations among geometric and non-geometric entities was introduced.

3. Database technology

3.1 Current trend and requirements for data repository

The need to share information between different engineering applications has long been recognized for collaborative engineering because tasks are to be carried out by multi-disciplined engineers who may be distributed in terms of both time and space; furthermore, different engineering partners need to use different applications. This practice means a product model generated from an application system has to be shared directly by other ones. Currently, interoperability among CAx applications becomes the bottleneck [48]. The development of a repository system which can manage all the information is an immediate need. The requirements for such a repository system are summarized as follows:

- Using international standards, since such standards allow participation by a wide variety of parties, including suppliers and customers.
- Consistent and unified engineering databases supporting different applications.
- A DBMS should be used to manage the storage and retrieval of the large amount of information for a reasonable sized cluster of companies. Such a DBMS can also readily allow for the integration with other functions within a company or across companies.
- Interactive 3-D graphical representations of parts and products for user manipulation.
- Mechanisms to check consistency and validity of information while sharing among different applications.

3.2 Multi-application information sharing through database

Inter-application information sharing can be done in two ways. The simplest way is sharing information in physical file format via proprietary data translators or Standard Data Access Interface (SDAI) [8, 28, 40, 78]. However, file-based approach suffers from disadvantages such as redundancy of the data over various files, multiple updates of the same data and potential conflicts, waste of storage space and non-user friendly access language. It does not provide an integrated view of product information and cannot implement an information model that is generic for different application programs.

The other way can be characterized as sharing information through access interfaces with database support. Database approach eliminates the limitations of file-based approach [60]. Modern database systems support multiple views of a common data resource to suit the needs of different functional areas and applications and to maximize reliability and availability by provision of efficient backup and recovery systems. In addition, databases can manage large amount of information and are very powerful for operations. The DBMS can also ensure the security and transparency for the users of engineering data. Therefore databases are appropriate tools for information sharing among different CAx systems. Ideally, an integrated product database can store data that covers all aspects of the entire product lifecycle [12]. Multiple applications can access the product data, and may take advantage of database features such as query processing. Ou-Yang and Chang [53] explained the design and development of a distributed, open and intelligent product data management system by incorporating agent technology. However, the research is still in a preliminary stage.

3.3 Database management systems

A database management system (DBMS) controls shared accesses to a database and provides mechanisms that ensure the security and integrity of the stored data with greater flexibility than using physical file format. Traditional relational database management system (RDBMS) organizes data into tables. The rows and columns of a table represent records and attributes. The relational model has been adopted in products like ORACLE, DB2, INGRES, INFORMIX, SYBASE, etc. In contrast, object-oriented database management system (OODBMS), employs a data model that supports object-oriented encapsulation, inheritance and polymorphism; this category includes O2, ObjectStore, Objectivity, ITASCA, GemStone, SERVIO, etc. However, OODBMS lacks the sufficient efficiency and scalability in large applications. Most commonly used databases are a kind of hybrid of RDBMS and OODBMS, and called object-relational database management systems (ORDBMS) [Date 2000]. This breed of databases benefits from both the relational and the object models in scalability and support for rich data types. ORDBMSs employ data models that incorporate OO features. All database information is stored in tables, but some of the tabular entries may have richer data structure, termed abstract data types (ADTs). Large RDBMS vendors such as ORACLE, IBM, Informix now are too evolving toward this direction. Their products are ORACLE 9i, DB2, Universal Server respectively.

4. Feature-based repository

Currently, most of the CAx applications are parametric and feature-based. Therefore, feature information (feature semantics) should be maintained [4, 6]. A collaborative engineering environment can be viewed as an integrated, heterogeneous database system. However, in reality, features, even those form-features defined in STEP [29] had not been supported. Cunha *et al.* [14] and Lee *et al.* [37] proposed to use object-oriented way to define a feature-based product database but did not implement a system for real product design. No detailed database schema and supporting mechanisms were given.

Theoretically, a product master model can be used to integrate CAD systems with downstream applications via different feature views throughout the stages of a product lifecycle [45, 39]. Wang, *et al.* [71] and Ma *et al.* [45] put forward a collaborative feature-based design system to integrate different CAx systems with database support. The definition, classification and relationship of features are described. To alleviate communication load between client and server, Koparanova *et al.* developed an EMQ (engineering mediator query) system that can support collaborative design by sharing information [35]. Great effort has been taken to validate model geometry [39]. More research has to be done to investigate how features, constraints and geometrical entities are to be managed. Furthermore, the mechanism for updating feature models in the database server side has not been developed. The drawback of the proposed structures was the lack of geometrical engine integration to support model validation.

In order to integrate different CAx applications, a generic feature model must be defined such that application-specific feature model can be defined and integrated with the entire product model generated. Xue *et al.* [74, 75] proposed a distributed and feature-oriented database modeling approach by adopting VRML as the standard data format and AutoCAD as the geometry modeling kernel among multiple users. However, their distributed databases cannot eliminate duplicated data, e.g. geometric data shared by different applications is duplicated in different databases.

4.1 Database Schema

Database schema architecture design is essential for any information system development. For example, schemas can be classified as external, conceptual and internal schemas. An internal schema specifies the physical storage model containing information such as file location, structures, indexing and access methods. External schemas are created from specific application data structure. A database may contain several external schemas. Each external schema is a subset of the overall conceptual schema. The conceptual schema is the single consistent logical model of the complete database. Such a three-level schema provides DBMS extensibility because in order to integrate an application, you merely need to add an external schema. In addition, the three-level schema makes database natural to support multiple views while at the same time maintaining the data consistency.

4.2 Network-oriented integration platform

Computer network-based integration such as the concepts of intranet, extranet, and infranet has been widely applied. More importantly, internet, a worldwide system of computer networks, has become an essential broad platform to implement global design and manufacturing [1]. Computer network offers promising features. For example, the client-server architecture provides both the flexibility and the control for the sharing information among global collaborative users across the time and space zones. The potential high speed capability of the computer network can also be used to develop network-centric applications that are equivalent to stand-alone systems in terms of functionality, performance and usability [1, 26]. A number of computer network-based systems have been developed to support design collaboration [1,2].

CORBA (Common Object Request Broker Architecture) is a standard architecture for distributed object systems. It allows a distributed, heterogeneous collection of objects to communicate across the network, and has been broadly used to develop collaborative systems. A new information integration platform based on agent and internet for computer integrated manufacturing system (CIMS) was presented in [64]. Wang and Tang [72] reported a pilot application of a decision-making environment for product design in extended enterprises using collaborative design and manufacturing agents. The system was implemented in an agent-based environment for conveying design and manufacture information across traditional technology boundaries.

5. Research issues and challenges

As reviewed in this paper, although a lot of research and development works has been done on enabling technologies of collaborative engineering, technological gaps between the industry requirements and the available solutions still exist. Ideally, in order to support collaborative product development, an information repository system should:

- Use international standard, since such standard allows participation by a wide variety of parties, including suppliers and customers.
- Extend unified product information model to cover entire product lifecycle such that different applications can be integrated. Further more, it should contain uniformly higher-level feature information so that engineering semantics can be maintained.
- Provide physical repository and efficient mechanism to manage large amount of product data.
- Manage the information in a concise and effective manner such that the entire product lifecycle information needs to be generated only once. Associated data is automatically synchronized.

5.1 Incorporation of a universal geometrical modeling kernel

All feature-based modeling applications have to be built around a geometry modeling kernel. A geometry kernel can provide lower-level geometrical modeling services to support higher-level parametric feature-based modeling. Currently, there are only a few geometrical modeling kernels commercially available. ‘Parasolid’ by Siemens Inc. and ‘ACIS’ 3D by Spatial Inc. are two well-known geometrical modeling kernels that were adopted by most commercial CAD systems. Historically, ACIS had come from a more linear solid background that made it a more natural fit for CAM while Parasolid had excelled more in blending and curving, which made it better in consumer products. Both kernels are rapidly evolving. They complement existing applications by offering platforms for the modeling of curves, surfaces, and solids. They also support the integration of proprietary curve and surface subsystems. Low level geometrical entity representation and model structures have to be unified. An interface (OpenDIS) between the geometric modeling kernel and the DBMS for the implementation of CAD system that uses the STEP database as the native storage was reported in [33]. However, since STEP cannot fully cover feature information for different CAx applications, using only STEP-based product specification is not sufficient to ensure a fully-integrated comprehensive product model. Recently, the concept of feature operation was defined by Chen *et al.*, and realized using a markup language [11]. Any geometrical kernel should provide a foundation of common modeling functionality and the flexibility to be adapted and extended for particular application requirements. Core geometry definitions can be guided by the existing STEP standards.

5.2 Shared consistent repository and constraint solving for evaluation and validation

Product and process information can no longer stored in a file format [8, 78], which means duplicated data and potential conflicts. It seems active database approach supported with web services is a promising direction to incorporate constraints into product and process repository [62, 45]. However, high level semantics and ontological reasoning are limited by the absence of a unified feature scheme in engineering models [34, 50, 76]. However, advanced constraint modeling is far behind the evolvement of advanced feature definitions [5, 37]. Unification of complex constraints is new field of research with great potential impact for the proposed feature-based repository technology [9, 67]. A product change protocol was proposed to maintain the links between application feature information and the shared product master model stored in a central database [10]. However, maintaining the association between product master model and the distributed application proprietary feature semantics needs more investigation [56]. In addition, although those distributed clients in the same domain (e.g. CAD domain) can access the master model concurrently, but the consistency maintenance of the product master model with constant updates remains a problem [62].

5.4 Development of an efficient feature-oriented engineering database

Although there are a few research works as reviewed in this paper, which have focused on building an engineering database to support collaborative product development, many of them focus on higher-level conceptualization [14, 19, 37, 71, 74, 75] without explicit database schema definitions. In addition, most of them did not provide efficient and effective mechanisms to support product feature information management in a consistent and concurrent manner [46, 59]. Last but not least, without higher-level information integration infrastructure, their works suffer from the problem of extensibility. To achieve a unified solution to manage diverse feature types, comprehensive modeling of a generic feature definition with built in methods of database schemas and repository access and management methods is essential. Most of the repository design lack of a higher-level generic feature definition, which can be used as a template for database schema extension in the future. Therefore, these proposed databases suffer from the scalability problem [54, 55].

5.5 Feature level interoperability

The standards of data exchange developed so far, such as IGES, SET and STEP, deal with only lower level geometric information. Interoperability of systems has been limited to geometric entity level, and semantic level information cannot be communicated [76]. To achieve the desired collaboration among various application systems [68, 79], feature level interoperability is important [48]. Feature operations can exchange or share the total information among different CAx systems [11, 32, 73]. Although many feature based CAx systems have been developed, there is no interoperability among different features of different packages [31].

6. Conclusions

A common information infrastructure that can support information sharing among a large number of CAx systems is urgently required for collaborative engineering. This paper presents an updated review on the research efforts in this direction. More specifically, it emphasizes on strategies of information sharing, the existing standards, feature technology, integration platform, database technology and feature based repository design. Technological gaps existing in the field have been identified. The authors believe that feature-based repository possess the potential to address many of the limitations. Reducing these gaps are the challenges for research to embrace the advantages from a collaborative engineering environment.

References

1. Alvares, A. J., Ferreira, J. C. E. and Lorenzo, R. M. (2008). An integrated web-based CAD/CAPP/CAM system for the remote design and manufacture of feature-based cylindrical parts, *Journal of Intelligent Manufacturing*, 19(6): 643-659.

2. Anumba, C. J., Pan, J., Issa, R. R. A. and Mutis, I. (2008). Collaborative project information management in a semantic web environment, *Engineering, Construction and Architectural Management*, 15(1): 78 - 94.
3. Bettig, B. and Shah, J. J. (2001). Derivation of a standard set of geometric constraints for parametric modeling and data exchange, *Computer-Aided Design*, 33: 17-33.
4. Bidarra, R. and Bronsvoort, W. F. (2000). Semantic feature modeling, *Computer-Aided Design*, 32: 201–225.
5. Bouikni N., Desrochers A. and Rivest L. (2006). A Product Feature Evolution Validation Model for Engineering Change Management, *Journal of Computing and Information Science in Engineering*, 6(2): 188-195.
6. Bronsvoort, W. F. and Noort. A. (2004). Multiple-view feature modeling for integral product development, *Computer-Aided Design*, 36(10): 929-946.
7. Bustos, B., Keim, D. A., Saupe, D., Schreck, T. and Vrani´C D. V. (2005). Feature-based similarity search in 3D object databases, *ACM Computing Surveys*, 37(4): 345–387.
8. Chao, P. Y., and Wang, Y. (2001). A Data Exchange Framework for Networked CAD/CAM, *Computer in Industry*, 44: 131-140.
9. Chen, G., Ma, Y. -S., Thimm, G. and Tang, S.-H. (2006). Associations in a unified feature modeling scheme, *ASME Transactions Journal of Computing & Information Science in Engineering*, 6(2): 114-126.
10. Chen, G., Ma Y. -S. and Thimm, G. (2008). Change propagation algorithm in a unified feature modeling scheme, *Computers in Industry,* 59(2-3): 110-118.
11. Chen, J. Y., Ma, Y. -S., Wang, C. L. and Au, C. K. (2005). Collaborative design environment with multiple CAD systems, *Computer-Aided Design & Applications*, 2(1-4): 367-376.
12. Chu C. -H., Chang C. -J. and Cheng H. -C. (2006). Empirical studies on inter-organizational collaborative product development, *Journal of Computing and Information Science in Engineering*, 6(2): 179-187.
13. Cowan R., Jonard N. and Zimmermann J. B. (2007). Bilateral Collaboration and the Emergence of Innovation Networks, *Management Science*, 53(7): 1051–1067.
14. da Cunha, R. R. M. and Dias, A. (2002). Feature-based database evolution approach in the design process, *Robotics and Computer Integrated Manufacturing*, 18: 275-281.
15. Dartigues C., Ghodous, P., Gruninger, M. Pallez, D. and Sriram R. (2007). CAD/CAPP integration using feature ontology, *Concurrent Engineering: Research and Applications*, 15(2): 237-49.
16. Date, C. J. and Darwen, H. (2000). *Foundation for future database systems: the third manifesto,* Addison-Wesley, 2000.
17. Dereli, T. and Filiz, H. (2002). A note on the use of STEP for interfacing design to process planning, *Computer-Aided Design*, 34: 1075-1085.
18. Dolenc, M., Klinc, R., Turk, Ž., Katranuschkov, P., Kurowski, K. (2008). Semantic grid platform in support of engineering virtual organisations, *Informatica*, 32(1): 39–49.
19. Domazet, D. S., Miao, C. Y., Calvin, C. F. Y., Kong, H. P. H. and Goh, A. 2000. An Infrastructure for Inter-Organizational Collaborative Product Development, in *Proceedings of the 33rd Hawaii International Conference on System Sciences*.
20. Fu, M. W., Ong, S. K., Lu, W. F., Lee, I. B. H. and Nee, A. Y. C. (2003). An approach to identify design and manufacturing features from a data exchanged part model, *Computer-Aided Design*, 35: 979–993.
21. Fynes, B. and Burca, S. D. (2005). The effects of design quality on quality performance, *International Journal of Production Economics,* 96: 1-14.
22. Gao, J., Zheng, D.T., and Gindy, N. (2004). Extraction of machining features for CAD/CAM integration, *International Journal of Advanced Manufacturing Technology*, 24: 573–581.
23. Gonzalez, F. J. M. and Palacios, T. M. B. (2002). The effect of new product development techniques on new product success in Spanish firms, *Industrial Marketing Management*, 31: 261- 271.
24. Hoffmann, C. M. and Arinyo, R. J. (2002). Parametric modeling, *Handbook of computer-aided geometric design,* 519-542, Amsterdam, North-Holland.

25. Holland, P., Standring, P. M., Long, H. and Mynors, D. J. (2002). Feature extraction from STEP (ISO10303) CAD drawing files for metal-forming process selection in an integrated design system, *Journal of Materials Processing Technology*, 125-126: 446-455.
26. Huang, G. Q. and Mak, K. L. (2001). Web-integrated manufacturing: recent developments and emerging issues, *International Journal of Computer Integrated Manufacturing*, 14(1): 1-2.
27. ISO (1994). Industrial Automation Systems and Integration — Product Data Representation and Exchange — Part 1: Overview and Fundamental Principles, *ISO 10303-1*:1994 (E).
28. ISO (1995). Industrial Automation Systems and Integration — Product Data Representation and Exchange — Part 22: STEP Data Access Interface, *ISO Document TC184/SC4 WG7 N392*.
29. ISO (2000). Industrial Automation Systems and Integration — Product Data Representation and Exchange — Part 224: Mechanical product definition for process planning using machining features, *ISO Document ISO TC 184/SC4/WG3 N854*.
30. Jha, K. and Gurumoorthy, B. (2000). Automatic propagation of feature modification across domains, *Computer-Aided Design,* 32: 691-706.
31. Jiang, P., Shao X., Qiu H. and Li P. (2008). Interoperability of cross-organizational workflows based on process-view for collaborative product development, *Concurrent Engineering: Research and Applications*, 16(1): 73-87.
32. Khaled, A., Ma, Y. -S. and Miller, J. (2008). A service oriented architecture for CAX concurrent collaboration, in *Proceedings of IEEE Conference on Automation Science and Engineering*, August 23-26, Washington DC, USA.
33. Kim, J. and Han, S. (2003). Encapsulation of geometric functions for ship structural CAD using a STEP database as native storage, *Computer-Aided Design*, 35: 1161-1170.
34. Kim, K. Y., Manley, D. G. and Yang, H. J. (2006). Ontology-based assembly design and information sharing for collaborative product development, *Computer-Aided Design*, 38(12): 1233-1250.
35. Koparanova, M. G. and Risch, T. (2002). Complete CAD queried for visualization, *Proceedings of the International Database Engineering and Applications Symposium (IDEAS'02),* IEEE .
36. Kumar, A. V. and Yu, L. (2001). Sequential constraint imposition for dimension-driven solid models, *Computer-Aided Design*, 33: 475-486.
37. Lee, K. Y., Kwon, O. H., Lee, J. Y. and Kim, T. W. (2003). A hybrid approach to geometric constraint solving with graph analysis and reduction, *Advances in Engineering Software*, 34: 103-113.
38. Li M., Gao, S. and Wang, C.C.L. (2007). Real-time collaborative design with heterogeneous CAD systems based on neutral modeling commands, *ASME Transactions Journal of Computing & Information Science in Engineering*, 7(2), pp. 113 - 125.
39. Li, M., Gao, S., Fuh, J. Y. H. and Zhang, Y.F. (2008). Replicated concurrency control for collaborative feature modelling: A fine granular approach, *Computers in Industry*, 59(9): 873-881.
40. Li, W. D., Fuh, J. Y. H. and Wong Y. S. (2004). An Internet-enabled integrated system for co-design and concurrent engineering, *Computers in Industry*, 55(1): 87-103.
41. 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.
42. Li, Y. T., Hu, S. M. and Sun, J. G. (2002). A constructive approach to solving 3-D geometric constraint systems using dependency analysis, *Computer-Aided Design*, 34: 97-108.
43. Liang J. S. and Wei P. W. (2006). Conceptual design system in a Web-based virtual interactive environment for product development, *International Journal of Manufacturing Technology,* 30 (11-12): 1010–1020.
44. Ma, Y. -S. and Tong, T. (2003). Associative feature modelling for concurrent engineering integration, *Computers in Industry*, 51(1): 51 – 71.
45. Ma, Y. –S., Tang, S. –H., and Chen, G. (2007). A Fine-grain and Feature-oriented Product Database for Collaborative Engineering, in *Collaborative Product Design & Manufacturing Methodologies and Applications* (Eds. W.D. Li et al.), Springer, 109-136.

46. Ma, Y. -S., Britton, G. A., Tor S. B. and Jin, L.Y. (2007). Associative assembly design features: concept, implementation and application, *International Journal of Advanced Manufacturing Technology*, 32(5-6): 434-444.
47. Ma, Y.-S., Rajeshbabu T. S. and Deng Y. -M. (2007). Web service oriented standard product library, *International Journal of Electronic Business Management*, 5(2):105-114.
48. Ma, Y. –S. (2008). Research on PLM system interoperability with a feature-object-based approach, *Advanced Materials Research*, 44-46: 373-38.
49. Mahmoud-Jouini, S. B., Midler, C. and Garel, G. (2004). Time-to-market vs. time-to-delivery Managing speed in Engineering, Procurement and Construction projects, *International Journal of Project Management*, 22: 359-367.
50. Noy, N. F., Chugh, A., Liu, W. and Musen, M. A. (2006). A Framework for Ontology Evolution in Collaborative Environments, in *ISWC 2006, LNCS 4273* (Eds.: Cruz I. et al.), Springer-Verlag, 544–558.
51. Nwagboso, C., Georgakis, P. and Dyke, D. (2004). Time compression design with decision support for intelligent transport systems deployment, *Computers in Industry,* 54: 291-306.
52. Otto, H. E. (2001). From concepts to consistent object specifications: translation of a domain-oriented feature framework into practice, *Journal of computer science & technology,* 16: 208-230.
53. Ou-Yang, C. and Chang, M. J. (2006). Developing an agent-based PDM/ERP collaboration system, *International Journal of Advanced Manufacturing Technology*, 30(3-4): 369–384.
54. Ouertani, M. Z. and Gzara, L. (2008). Tracking product specification dependencies in collaborative design for conflict management, *Computer-Aided Design*, 40(7): 828–837.
55. Ouertani, M.Z. (2008). Supporting conflict management in collaborative design: An approach to assess engineering change impacts, *Computers in Industry*, 59(9): 882–893.
56. Panchal, J. H., Fernandez, M. G., Paredis, C. J. J., Allen, J. K., and Mistree, F. (2007). An Interval-Based Constraint Satisfaction (IBCS) method for decentralized, collaborative multifunctional design, *Concurrent Engineering Research and Applications*, 15(3): 309-323.
57. Pal, P., Tigga, A. M. and Kumar, A. (2005). Feature extraction from large CAD databases using genetic algorithm, *Computer Aided Design*, 37(5): 545-558.
58. Pratt M. J., and Srinivasan V. (2005). Towards a neutral specification of geometric features, *International Journal of Computer Applications in Technology*, 23(2/3/4): 203-218.
59. Rachuri, S., Han, Y.-H., Foufou, S., Feng, S. C., Roy, U., Wang, F., Sriram, R. D. and Lyons, K. W. (2006). A Model for Capturing Product Assembly Information, *Journal of Computing and Information Science in Engineering*, 6(1): 11-21.
60. Ramakrishnan, R. and Gehrke, J. (2000). *Database management systems*, Boston: McGraw-Hill.
61. Rouibah, K. and Ould-Ali, S. (2007). Dynamic data sharing and security in a collaborative product definition management system, *Robotics and Computer-Integrated Manufacturing*, 23(2): 217-233.
62. Shehab, M., Bhattacharya, K. and Ghafoor, A. (2007). Web services discovery in secure collaboration environments, *ACM Transactions on Internet Technology*, 8(1): 5:1-22.
63. Shen, L., Li, M., Zhao, W., Zhou, Z., Li, Y., Wu, M. and Zheng, J. (2006). Web Based Cooperative Virtual Product Design Environment Shared by Designers and Customers, in *CSCWD 2005, LNCS 3865* (Eds: W. Shen et al.), Springer-Verlag, pp. 384 – 393.
64. Shen, W., Hao, Q., Wang, S. Li, Y. and Ghenniwa, H. (2007). An agent-based service-oriented integration architecture for collaborative intelligent manufacturing, *Robotics and Computer-Integrated Manufacturing*, 23(3): 315-325.
65. Shen, Y., Ong, S. K. and Nee, A. Y. C. (2008). Product information visualization and augmentation in collaborative design, *CAD Computer Aided Design*, 40(9): 963-974.
66. Sudarsan, R., Fenves, S.J., Sriram, R.D., and Wang, F. (2005). A product information modeling framework for product lifecycle management. *Computer Aided Design*, 37(13): 1399-1411.
67. Thimm, G., Lee, S. G. and Ma, Y. -S. (2006). Towards unified modelling of product life-cycles, *Computers in Industry*, 57(4): 331–341.

68. Tseng, K. C. and Abdalla, H. (2006). A novel approach to collaborative product design and development environment, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 220(12): 1997-2020.
69. Venkataraman, S., Shah, J. J. and Summers, J. D. (2001). An investigation of integrating design by features and feature recognition, in *Proceedings of IFIP Conference*, FEATS 2001, Valenciences, Frances.
70. Wang C. H. and Chou S. Y. (2008). Entities' representation modes and their communication effects in collaborative design for SMEs, *International Journal of Manufacturing Technology,* 37(5-6): 455-470.
71. Wang, H. F., Zhang, Y. L., Cao, J., Lee, S. K. and Kwong, W. C. (2003). Feature-based collaborative design, *Journal of Material Processing Technology*, 139: 613-618.
72. Wang J. X. and Tang M. X. (2006). A Multi-agent Framework for Collaborative Product Design, in *PRIMA 2006, LNAI 4088* (Eds.: Z. Shi and R. Sadananda), Springer-Verlag, pp. 514 – 519.
73. Wang, Y. and Nnaji, B. O. (2006). Document-driven design for distributed CAD services in service-oriented architecture, *Journal of Computing and Information Science in Engineering,* 6(2): 127-138.
74. Xue, D. and Yang, H. (2004). A concurrent engineering-oriented design database representation model, *Computer-Aided Design,* 36(10): 947-965.
75. Xue, D. and Xu, Y. (2003). Web-based distributed system and database modeling for concurrent design, *Computer-Aided Design*, 35: 433-452.
76. Yang, Q. Z. and Zhang, Y. (2006). Semantic interoperability in building design: methods and tools, *Computer-Aided Design*, 38(10): 1099–1112.
77. Zha, X. F., Sriram, R. D., Fernandez, M. G. and Mistree, F. (2008). Knowledge-intensive collaborative decision support for design processes: A hybrid decision support model and agent, *Computers in Industry,* 59(9): 905–922.
78. Zhang, Y.P. (2000). An Internet based STEP Data Exchange Framework for Virtual Enterprises, *Computers in Industry,* 41: 51-63.
79. Zheng X., Sun G. and Wang S. (2006). An approach of virtual prototyping modeling in collaborative product design, in *CSCW 2005, LNCS 3865* (Eds.: W. Shen et al.), Springer-Verlag, 493 – 503.