Advanced Materials Research Vols. 44-46 (2008) pp 373-380
online at http://www.scientific.net

Research on PLM system interoperability with a feature-object-based approach

Y.-S. Ma[a]
Department of Mechanical Engineering, University of Alberta, AB T6G 2G8, Canada.
Tel: +1(780)492.4443 Fax: +1(780)492.2200
[a]yongsheng.ma@ualberta.ca

Keywords: Interoperability; Product lifecycle management; Collaborative engineering

Abstract. Interoperability can be described as the interpretability of data types and related information by different computer systems. Standards for data format and communication protocols have been developed and adopted by different industries. So far, interoperability has only been investigated at the data level. This paper addresses a research approach proposing a system design and a set of generic methods so that to embed engineering knowledge and to achieve interoperability at the feature level in an open collaborative engineering environment. Ideally, the proposed new approach would allow knowledge rules to be embedded into the constraints of features supported by the complex associations of a multi-application engineering repository. Potentially, the system proposed offers user-defined feature types that support flexibility in feature-based information definition, sharing and mapping.

Introduction

Currently, collaborative design and manufacturing systems are developed and used to fulfill product and process information requirements in a dynamic and networked environment. The great challenge as well as the motivation for the author is to make different systems to work together effectively and efficiently based on a coherent and consistent platform. The major obstacles are incompatible data structures and the lack of data associations across different solutions; interoperability has been a bottleneck. Achieving the required interoperability among engineering software tools is highly demanded. This publication addresses the interoperability issue at a semantic feature level within the context of networked and knowledge driven collaborative design and manufacturing.

Traditionally, interoperability refers to the ability of different software packages to exchange system data. This is a critical requirement for systems that use data from other software; e.g., Computer-Aided Manufacturing (CAM) software uses data from Computer-Aided Design (CAD). Interoperability among engineering software tools is in high demand due to globalization and value chain integration [1]. International standards, such as IGES and ISO STEP [2] standards have been used for the exchange of geometric entities. To date, interoperability has only been investigated at the data level. For example, STEP has obvious limitations on semantics and high-level entity types such as user-defined features and constraints. If a parametric part is created in Pro-E with features, such as counter-bore holes, chamfers, or boss cylinders, after being exported into a STEP file and imported into UG NX, the model becomes a fixed solid block.

Essentially, the interoperability can be described as the interpretability of data types and the related information defined accordingly. The interoperability achieved with ISO STEP standard is at solid model level; its implementations are very limited for semantic content sharing and consistency check because non-geometric information is striped off during translation. High level information, like features defined in product or process models, is lost. Such translation approach also creates a tremendous number of inconsistent copies of data files in collaboration processes like the 'snow-ball' effect observed in day-to-day email attachments. Such discrepancies are great hurdles for collaborative integration among engineering partners. With the penetration of virtual-enterprise business practice in the current engineering collaborations, globally-distributed designers and engineers at different stages of product lifecycles use different semantics and engineering patterns

[3]; yet they all work on a common product with different derived 'views' of their relevant working scopes [4-6]. Hence, a new level of interoperability has to be investigated to create a solution for pervasive collaboration based on advanced engineering informatics and Web technologies.

This work proposes a system framework and a set of generic methods to embed engineering knowledge and achieve interoperability at the feature level in an open collaborative engineering environment. The expected improvement would allow knowledge rules to be embedded into the constraints of features together with the complex associations of different aspects of a product or project. An open approach is to be taken, which offers self-defining feature types that support flexibility in feature-based information definition, sharing and mapping. The research methodology will be based on object-oriented software engineering methods and web-service-oriented architecture design [7]. The long-term goal is to reduce system incompatibility in order to support an effective and efficient collaborative engineering for multiple players, applications, and stages of product lifecycles. Eventually, as a result of this research, industry productivity and competitiveness can be enhanced on a significant scale. This paper is intended to introduce the overall system design approach. Detailed research works are not covered.

Feature Level Interoperability

In the past, the interoperability has always been a 'problem' because data formats and their embedded data structures (schemas) become means of protection for commercial interests. In recent years, the concept of 'open' data format or source codes is gaining acceptance due to customers' demands for collaboration in the global arena. It would be expected that the data structure schema for different features can be made publically available. In addition, given the consolidation of geometric modelers, common geometry-oriented functions, such as creating a solid in CAD systems, will become commodity expectation. Therefore the research in interoperability at the level of semantic knowledge becomes feasible and imperative [8, 9]. The research proposed covers the informatics modeling for semantic information sharing, mapping, manipulation, conversion and knowledge-based reasoning and automation [10]. Similar to the geometric data exchange processes, where STEP-based interoperability depends upon B-rep and CSG geometry modeling standardization, to solve the semantic interoperability problem, a scheme like 'the valuing system' of trading currencies is needed. Currently, a common and flexible standardized scheme for higher level semantic information association and sharing is not yet available [8, 11-14].

The author proposed generic engineering features for the potentially transferable 'trading currency' [14]. This is a totally new paradigm. Up to date, product models stored in the form of databases are either purely geometric or meta-linked to CAD files [15]. The first product model type lacks feature semantics and hence is too rigid for collaborative engineering. The second type is dependent upon native CAD systems for detailed entity access which are sensitive to the proprietary data scheme and the neutrally accessible files are too large in information grain size that makes network-based engineering collaboration difficult.

In the past 20 years, feature-based technology has been the corner stone for concurrent engineering. Theoretically, 'features' could have flexible definitions and are closely related to engineering expert cognition patterns. However, even though such 'features' can be very versatile and effective in engineering intelligence modeling [6, 16], so far feature-based engineering is confined to separate computer solution packages. There is no interoperability among different features of different packages; and very often features have to be 'recognized' [17]. This situation becomes unacceptable as the development of collaborative engineering has increased dramatically [18]; multiple parties and applications need to work together ever more closely along a dynamic value chain and across the stages of the whole lifecycle of a product.

The author and his coworkers had carried out initial work 'extending' the interoperability prospect up to feature level [19, 20]. It is assumed that there exists a centralized repository where all the product information is managed and stored in the form of a single or a cluster of database(s).

Semantically, all the entities, regardless geometric or non-geometric ones, are uniquely defined and represented in as well defined instances of their corresponding entity types [9]. This repository is accessible via a common interface toolkit by all different collaborating users. Generally, a database design has layers of information according to the accessibility required, or grain size. The central repository is supposed to provide services to fine-grain entities. In other words, there is no limit to the detailed data accessibility to end users. An important corner stone for this research approach is that a generic feature scheme has to be implemented based on some scalable informatics approaches, such as object-oriented programming coupled with object-relational databases [20]. Clearly, feature level interoperability can significantly enhance the richness and flexibility of knowledge representation and constraint management within a well-defined product or project modeling space.

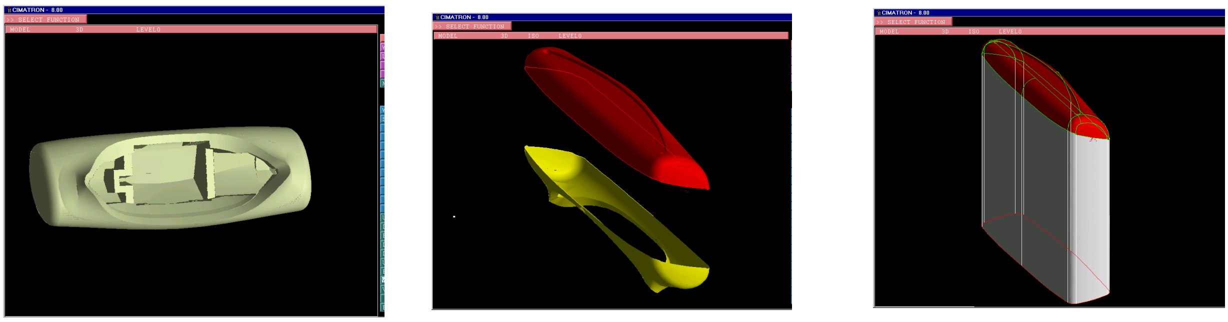
Overall Solution Framework

The proposed initial research is to develop the knowledge and working methods to enhance the semantic-level engineering collaboration on a Web-based platform that keep consistent product and process models, so that such models could support different stages of a product lifecycle (including and beyond design and manufacturing). The long term vision is to build a new generation digital engineering collaboration platform. In this direction, the first objective is to explore the most fundamental and very challenging mechanisms of interoperability at the semantic level. The proposed research covers the following six integrated key aspects: (1) Standardization of generic feature types and methods; (2) Feature mapping scheme design and mechanisms; (3) Dynamic repository; (4) Constraint management; (5) Interfacing mechanisms with application packages; (6) Communication language and the supporting algorithms. Detailed discussion about these aspects, such as the current state of art, their pertinent issues and the proposed methods are presented below.

Standardization of Generic Feature Types and Methods. To achieve an interoperable environment, a neutral data structure that covers all types of entities to be processed is essential [11]. This part of the research should cover both geometric and non-geometric information with emphasis on constraints. At the geometry level, STEP standard has been well developed and to be used, but it will have to be redeveloped by following fully object-oriented approach. At the feature level, as the preliminary effort, a generic feature representation scheme has to be developed. Generally, a feature is a set of entities associated with references and constraints representing certain results of engineering reasoning processes, or engineering patterns, or engineering intent. A draft version of the scheme has been described in EXPRESS-G [20] which can be used as the base class (or type abstraction) of feature definition, and then child classes can be derived. The concept of 'generic' features and the corresponding repository service methods have been discussed. Upon this 'generic feature' class, a unified, associative and hierarchical feature definition scheme [10] has been explored. However, the effort is far from being conclusive yet. One part of research is to continue the investigation of the hierarchical feature scheme. Multiple application feature conversions are to be systematically analyzed although some efforts were reported for the conversions between CAD and CAE features or derivations of cooling channels [19]. This research aspect is expected to result in a 'blue print' of class diagram for versatile feature types that are commonly encountered. In addition, this part should also investigate the mechanisms for defining 'open-end' types of features. The measure of success will be based on the flexibility and capability of definimng different associated feature types based on a unified feature definition scheme.

Feature Mapping Scheme Design and Mechanisms. An interoperable, knowledge-driven and smart platform is akin to the advanced stock trading platforms in the financial sector. Their scalability and complexity are comparable. The form of a dynamic 'banking system' in the engineering domain is a collaboration environment that must support dynamic and open-end partnerships among market players. The technical issue is the effective interfaces with various applications both vertically and horizontally. While vertical interfacing will be discussed in sub-section of 3.5, this sub-section addresses the multiple systems working together with different inter- or intra-features horizontally [10].

The essential challenge here is that via dynamic mapping of feature types, or semantic patterns, the resulted information and data from different systems can be automatically stored into a neutral and consistent data structure such that all the semantic features, defined in the form of sets with well-defined constraints and entity properties, become automatically interpretable with some necessary authorization and algorithms via web-services. It should be noted that web-service technology is suggested due to its great potential in embedded information processing capability in an active manner on the internet. Then original research is required to develop effective mapping method so that efficient conversion of features between the master product model and individual application view models, and the coherent consistency evaluation for all 'co-existing' facets of the product space can be achieved. Fig. 1(a) gives an example rubber part that is used for the support apparatus of a household electric iron design by Philips®.



(a) The rubber part model in association to the master product model and the metal frame of the iron rest

(b) Split faces for core and cavity inserts for rubber mold design

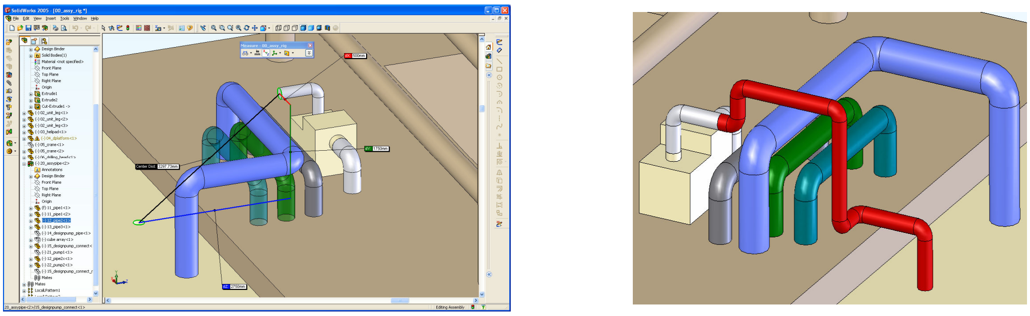
(c) Electrode model fro EDM machining

Fig. 1: Derived features from the master product model

Clearly, the part is part of the overall product design assembly. Its geometric features are matched to the original industrial design model and the metal frame that is to be interfaced with. Since the supporting apparatus is an optional sub-module of the whole product package but not the core module (the iron), so the design of it can be subcontracted locally to a contract design company (Company A) in the market country. Clearly, its design depends upon the master iron design model from the principal company. To manufacture this part, another local rubber molding company (Company B) is engaged and its mold has to be designed. Based on the input from Company A, the mold designers of Company B have to scale the model according to the thermal shrinkage factors of the rubber material used, and then split the model into patches of faces for the separation of core and cavity mold insert faces as shown in Fig. 1(b). Further down to the collaboration chain, a mold maker (Company C) is appointed to make the mold where an electrode has to be designed to produce the required inserts, then the designed faces of the mold model from Company B have to be mirrored and reconstructed as the electrode model as shown in Fig. 1(c). It can be appreciated that associative feature trees are to be tracked globally across the network for the consistency by an efficient mechanism at different sites, among a federation of databases. Collaboration scenarios could be tested among a few typical testing applications and with some commonly used systems.

Dynamic Repository. A recent work by the author and his co-worker [20] introduced a fine-grain and feature-oriented product database design to support Web-enabled collaborative engineering services. A four-layer information integration infrastructure was proposed and a solid modeler was incorporated to provide lower-level geometrical modeling services. However, a limitation was that the creation of entity-relation tables in the database was carried out manually, so dynamic feature type initialization in the database has not been realized. Ideally, an 'open' collaboration environment should allow end users to define their feature types on the fly automatically. To do so, dynamic manipulation of database schema and the automatic configuration of entity relations for fine grain feature entities and constraints must be achieved by interpreting an instructional message. Then, repository data structures should be self-created by some generic agents. With the development of active database technology, the enabling functions have been made available by vendors [13, 18]. This paper suggests systematic design and development for a dynamic, flexible, and generic feature-oriented repository system for product and process domains. More specifically, more effective and efficient database operation mechanisms should be investigated and prototyped to support the creation, modification, and editing of flexibly-defined feature types with complex entities, including geometric and non-geometric ones, as well as constraints. Active database processing capability to achieve automated 'checking-in' and 'checking-out' functions according to the user's portfolio and feature schema should be demonstrated.

Constraint management. Constraints must be explicitly defined and managed in a feature scheme to specify relationships among features, geometric or topological entities, and to provide invariant characteristics in the model. Constraints have various types, e.g. geometric constraints, tolerance constraints [10, 21]. In the generic feature definition, constraints are regarded as properties attached to a set of associated entities, e.g. geometric and non-geometric entities, or features. Although different types of constraints may have different attributes, they fall into a few common types, which can be generalized [10]. However, the evolving collaborative engineering scenarios create the dynamic solution requirement. Fig.2 shows a case of piping system design during an oil rig design process. The conceptual design is usually based on the past experience and many equipment items on board have to be supplied by specialized companies after bidding for the project.



(a) A smart design environment with specified constraints (b) A resulted piping path after optimization

Fig. 2: A case of pipeline design optimization in oil rig design

The rules to be embed in the design could be '*the minimum crossing space between pipelines has to be more than x meters*', '*the use of lower space is prioritized for the ease of maintenance*', '*reserving maximum space for operational use*', etc. In implementation, to support these rules, constraints between the pipelines and the surrounding environment have to be created and evaluated. The issue is that the surrounding environment has to be evolved from a very vague estimation, to abstracted design versions, detailed design versions, the design state with previously-used equipment in phases, and eventually to the design results with confirmed equipment installed. Fig. 2(a) shows an intermediate stage of the design. Fig. 2(b) shows one of the optimized pipeline paths. Many optional alternatives exist when a sub-system is to be added or replaced in the design and the optimization procedure has to be run iteratively. The whole design process has to be verified and reconstructed frequently.

Therefore, a more systematic global constraint handling method is to be further investigated. In the proposed collaborative environment, researchers has to develop a comprehensive method that can effectively model, apply, evaluate, and edit complex, associated or cross-linked, and smart constraints, and to prove the capability for open definitions so that the end user could define their types of constraints flexibly.

Interfacing Mechanisms with Application Package. Along the vertical direction, interfacing at different levels of abstraction, e.g. knowledge-oriented rules in expert systems and procedural algorithms used in dealing with geometries, has to be supported. Some fundamental mechanisms have been developed by the author [10]. A new unified and associative feature modeling approach has been developed [14]. Multiple applications and different associative features across different stages of product life cycles can be modeled consistently under a common flexible product/process modeling scheme. Fig.3 shows an example core insert developed from a plastic molding part.

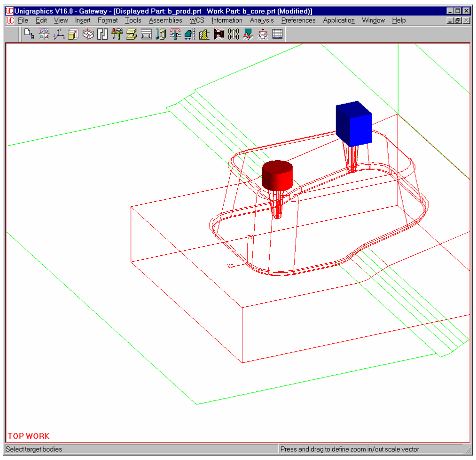


Fig. 3: A core-insert creation process in plastic mold design

As shown in the figure, clearly, the rules for machining this insert will be created with a CAM package. The high level rules to be applied include ‘*applying the right cutter for creating the drafting angle*’ and ‘*creating the minimum fillet radii without gouging*’. Eventually, tool paths will be converted into CNC machining codes. The advantage of the proposed system is that when the original plastic part has been changed, the geometry could be updated and the machining rules can be automatically reapplied to generate the new tool paths and CNC codes. If there is any issue during the re-evaluation process, notifications to the right task-owners will be served. Ideally, higher level semantic knowledge based systems can be seamlessly interfaced with detailed data-oriented procedures for tedious geometrical and non-geometrical entities via streamed and reusable entity-type-based methods supported by object polymorphism. To do so, systematic interfacing mechanisms need to be developed so that real knowledge-driven and smart close-loop engineering scenarios can be tested with some typical applications and systems. The effectiveness and efficiency of the proposed mechanisms are to be measured in stages.

Communication Language and Related Algorithms. Like XML, the commonly-used mark-up language across the Internet, a feature-oriented mark-up language (FML) should be used to support communications of feature transactions with associated operation methods or algorithms. The feature operation concept is generated and tested from the author’s past research as presented in [22]. In order to support the interpretation, processing, editing, generation of feature operation packages over the Web for effective functioning of application operations, standardization of such a feature mark-up language is required. For research purpose, a skeleton draft version of FML is to be developed to test its effective encapsulation of operations. The service methods for information integrity check, flow management, language compiling and interpretation, accessibility management, and the triggering of evaluation and validation procedures, are to be developed. As to the computer system and network design, two approaches, i.e. centralized data banking or federal association clustering, are to be investigated.

Significance of Expected Results

First, this proposed research will enhance the knowledge and existing methods from a totally new approach for the collaboration of different applications along product life cycle stages, aiming at the smart integration based on a common and consistent set of product or process models. This approach is expected to get the engineering changes propagated more effectively and efficiently in a closed loop to support design and manufacture optimization in and across different aspects of collaborative engineering. Second, engineering application systems developed by different vendors can be integrated into a collaborative value chain via ‘need-based’ information transactions such that the intellectual properties of each participating company are well protected and yet flexibly incorporated. Third, it can be envisaged that this technology will enable pervasive virtual engineering ‘transactions’ with the realization of ‘plug-and-play’ connectivity and effective streaming with highly automated and flexible simulations. Hence, collaborative engineering activities could be dynamically predicted, executed and managed. Eventually a full fledge digital collaborative engineering platform will be available not only to big multi-national companies (MNCs) but also small and medium enterprises (SMEs). Due to the flexibility and reusability, companies can significantly compress their product development or in-field construction or manufacturing time, and enhance the competitiveness and quality of products and manufacturing processes.

Conclusions

This proposed research approach aims to investigate an interoperable collaboration method among multiple engineering application systems across product life cycle stages. Theoretical study on the system modeling, dynamic functioning repository, and fine-grain engineering features associated with both semantic and procedural aspects of engineering intelligence is presented. Based on web-oriented information system architecture, a generic feature mark-up language (FML) that supports feature object interactions is also proposed. Ultimately, the schematic system design can be useful to enable consistent product models and effective communications in a flexible and open-ended manner. To achieve this, a unified and associative feature modeling approach is to be adopted based on the object-oriented software engineering methodology. Web-service and scalable fine grain database technologies are to be applied. This research approach has a long term vision of unified and integrated web-based collaborative engineering, but this paper focuses on some seeding research areas.

References

[1] Kubotek USA Inc: The *2006 CAD Interoperability Survey Results*, URL: www.kubotekusa.com, 2006

[2] ISO: *Industrial Automation Systems and Integration — Product Data Representation and Exchange — Part 42: Integrated Generic Resources: Geometric and Topological Representation*, *ISO 10303-42,* 1994 (E).

[3] R. Sudarsan, S.J. Fenves, R.D. Sriram, F. Wang: Computer Aided Design Vol. 37 (2005), p. 1399.

[4] L.H. Wang, W.M. Shen, H. Xie, J. Neelamkavil, A. Pardasani: Computer-Aided Design Vol. 34 (2002), p. 981.

[5] W.F. Bronsvoort, A. Noort: Computer-Aided Design Vol. 36 (2004), p. 929.

[6] J. Gao, D.T. Zheng, N. Gindy: Int. J. of Adv. Manuf. Tech. Vol. 24 (2004), p. 573.

[7] Q. Hao, W.M. Shen, Z. Zhang: Advanced Engineering Informatics Vol. 19 (2005), p. 123.

[8] Q.Z. Yang, Y. Zhang: Computer-Aided Design Vol. 38 (2006), p. 1099.

[9] M.Z. Ouertani, L. Gzara: Computer-Aided Design (2008), in press.

[10] G. Chen, Y.-S. Ma, G. Thimm, S.-H. Tang: ASME Trans. J. of Comp. & Info. Sci. in Eng. Vol. 6 (2006), p. 114.

[11] M.J. Pratt, V. Srinivasan: Int. J. of Comp. App. in Tech. Vol. 23 (2005), p. 203.

[12] K.Y. Kim, D.G. Manley, H.J. Yang: Computer-Aided Design Vol. 38 (2006), p. 1233.

[13] C. Ou-Yang, M.J. Chang: Int. J. of Adv. Manuf. Tech. Vol. 30 (2006), p. 369.

[14] Y.-S. Ma, G. Chen, G. Thimm: J. of Intell. Manuf. (2008), in press.

[15] A. Saaksvuori, A. Immonen: *Product Lifecycle Management* (Springer, Heidelberg 2005).

[16] A. Noort, G.F.M. Hoek, W.F. Bronsvoort: Computer-Aided Design Vol. 34 (2002), p. 899.

[17] W.D. Li, J.Y.H. Fuh, Y.S. Wong: Computers in Industry Vol. 55 (2004), p. 87.

[18] D. Xue, H. Yang: Computer-Aided Design Vol. 36 (2004), p. 947.

[19] Y.-S. Ma, T. Tong: Computers in Industry Vol. 51 (2003), p. 51.

[20] Y.-S. Ma, S.-H. Tang, G. Chen, in: *Collaborative Product Design & Manufacturing Methodologies and Applications*, edited by W.D. Li, et al. Chapter, 6, Springer, London, (2007), p. 109.

[21] J.H. Panchal, M.G. Fernandez, C.J.J. Paredis, J.K. Allen, F. Mistree: Concurrent Engineering Res. and App. Vol. 15 (2007), p. 309.

[22] J.Y. Chen, Y.-S. Ma, C.L. Wang, C.K. Au: Comp. Aided Des. & App. Vol. 2 (2005), p. 367.