Product Database Modeling for Collaborative Product Development

Nam Chul Do*, Hyun Kim*,
Hyoung-Sun Kim*, Jae Yeol Lee*
and Joo-Haeng Lee*

Abstract

To deliver new products to market in a due time, companies often develop their products with numerous partners distributed around the world. Internet technologies can provide a cheap and efficient basis of collaborative product development among distributed partners. This paper provides a framework and its product database model that can support consistent product data during collaborative product development. This framework consists of four components for representing consistent product structure: the product configuration, assembly structure, multiple representations and engineering changes. A product database model realizing the framework is designed and implemented as a system that supports collaborative works in the areas of product design and technical publication. The system enables participating designers and technical publishers to complete their tasks with shared and consistent product data. It also manages the propagation of engineering changes among different representations for individual participants. The Web technologies introduced in this system enable participants to easily access and operate shared product data in a standardized and distributed computing environment.

Key Word: *Product Database Modeling, Collaborative Product Development, Product Data Management, World Wide Web.*

1. Introduction

In the current global market environment, companies often develop their products with numerous partners distributed around the world. For collaborative product development among partners, Internet technologies can provide a cheap and efficient basis for sharing and exchanging product data. [1]

There are currently a number of approaches to Internet-based collaborative product development. Most of these, however, concentrate on the accessibility of users and interoperability of distributed software components via the World Wide Web (Web) and loosely integrate the separated engineering data for the collaboration. Although these approaches enable participants to easily access and share product data, they provide inconsistent data that causes the participants serious additional work and wasted resources.

Here we provide a framework that can maintain the consistency of product data during collaborative product development. The framework consists of four components for maintaining the consistency of the product structure that integrates all engineering data and related changes. Based on this framework, we designed a product database model and implemented it as a Web-based product data management and technical publication system in order to verify the feasibility of this study.

This paper is organized as follows. In Section 2, different approaches to information systems for collaborative product development are reviewed. Section 3 describes a framework that enables information systems to maintain the consistency of product data during collaborative product development. In section 4, a product database model is proposed in order to support collaborative product development in the areas of product design and technical publication. Section 5 discusses prototype implementation along with an illustrated example. Section 6 concludes the paper with a few remarks on further research directions.

* Electronics and Telecommunications Research Institute

2. Related Works

We use the classification framework proposed by Huang and Mak [6] to review the current research on Web-based collaborative design. They proposed computer supported collaborative work (CSCW) and workflow management systems as the classifications of the current approaches.

Kim et al. [8] developed a Web-based collaborative design system that can be classified as a CSCW. It provides a suit of collaborative tools such as audio/video conference modules and collaborative agents. Engineering services such as computer aided engineering (CAE) and 3D model browsing are also supported in the collaborative system. Kim et al. [9] also developed 3D collaborative viewing tools for mechanical parts that can provide collaboration sessions for suppliers and buyers.

Kim et al. [8] and Saha et al. [10] proposed collaborative design systems that provide a workflow facility. The work flow controls the tasks and related resources such as documents, engineering models and software agents on the Web environment. Kim et al. integrated workflow system with the CSCW facility to specify complex design processes. Saha et al. proposed an application of the STEP standard to specify a process-oriented design history management system.

The aim of all the approaches mentioned above is to provide accessibility for distributed participants to shared product data and to increase interoperability of engineering applications via the Internet. Numerous human and software participants in a collaborative system, however, may increase the possibility of inconsistent product data. In particular, the engineering changes could cause inconsistent data if they cannot be properly propagated to other departments in a company. Therefore, our research aims to provide multiple product data representations for collaborative product development and management of engineering changes to support consistent product structure among representations for individual participants.

Our research is also related to research concerning the management of engineering changes that can be classified into three approaches: the document-oriented, process-oriented and product structure-oriented.

The document and process-oriented approaches [2,5,3,10] separate the engineering change data from product structure and store it in electronic documents or components of workflow. The separation prohibits designers from directly analyzing and reviewing product structure, which are frequent and essential operations required during the course of engineering changes, and thus forces them to gather and process various engineering change data scattered in the heterogeneous computing environment. In addition, since the design workflows are collaborative and content-dependant processes, their specification at detailed level would be a burden for designers in the process-oriented approaches [3,10].

The structure-oriented approaches [1,4] that we adopted manage engineering changes tightly integrated with product structure so that they can maintain the consistency of data with rapidly changed product structure during engineering changes. Current product data management (PDM) systems that maintain all engineering data integrated with product structure employ these approaches.

3. A Framework for Consistent Product Structure

The proposed framework targets product structure as the object of consistency maintenance, since it can integrate and manage all product data and engineering changes consistently. Figure 1 shows a framework for representing consistent product structure. The framework has four components related to the product structure: the product configuration, assembly structure, multiple representations and engineering changes. The components are tightly integrated to each other and there are some issues between them. The rest of this section describes each component and issues between them.

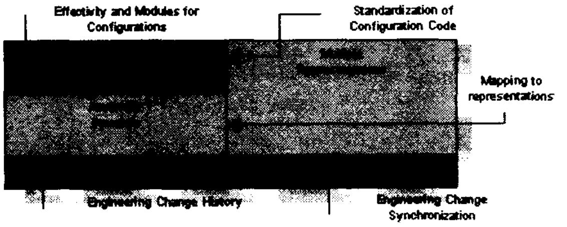

Effectivity and Modules for Configurations
Standardization of Configuration Code
Mapping to representations
Engineering Change History
Engineering Change Synchronization

Figure 1. The Framework for consistent product structure

The product configuration enables products to be assembled from instances of a fixed set of pre-defined components that interact with each other in predefined ways. This helps manufactures quickly respond to various customer needs for products without additional preparation of tests or production processes for the new specification. The components for product configuration are represented as variants that can change their product structure with a specific condition, termed the effectivity [7]. Since they have the assembly structure as their sub product structure, the issue of the effectivity and modules for configuration exists between the product configuration and assembly structure.

The assembly structure represents the physical composition of products. It describes each physical part participating in products and their constituent relationships with other parts. The changes of the assembly structure cause engineering changes and they should be propagated to the product configuration and multiple representations through the mechanism for propagating engineering changes.

To support collaborative product development, a product database model should provide different views of the product for individual participants from the base data representation. The multiple representations map the base product configuration and assembly structure to each participant's view. To provide proper content for the individual participants, the view should filter the base assembly structure. For example, a part planner selects or expands the base assembly structure to indicate service parts for product operations. Since each participant has a different view of the product structure, the standardization of configuration code should be completed to exchange consistent data on product configurations among the participants.

Engineering changes in the framework manage a history of structure changes in product configuration and assembly structure and usually initiate a wide range of information processing that synchronizes the multiple representations with the changed base product structure.

4. Product Database Model for Product Data Management and Technical Publication

In this section, we describe a formal representation of the product structure for the collaborative product development in the areas of product design and technical publication. Before a description of the data representation, we introduce parts catalogs that are generated by technical publication systems.

4.1 Parts Catalogs

Parts catalogs contain part details and an engineering diagram for every component of a product. The part details include serial number, quantity and component variant information. The engineering diagram shows each part and how those parts are assembled to form the component. Customers and service engineers use parts catalog to visually identify the parts they require during the diagnosis of problems and the maintenance of machines.

Publishing parts catalogs is normally a critical path in the product development process, since it requires additional planning and mapping of the product structure for the publication as well as completion of product design. After publication, parts catalogs should maintain consistency with the design data of the product throughout the product life cycle, even if engineering changes incur changes of the product structures and part information.

4.2 Product Database Model

Figure 2 shows a simplified entity relationship diagram for product database model. In the diagram, there are four schemas, or sets of related entities, which correspond to the components of the proposed framework. We explain the entities according to the schemas they belong to.

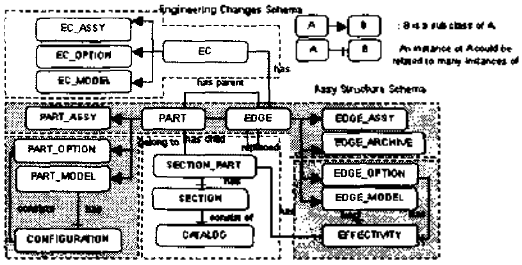


Figure 2. Product database model for product design and technical publication

4.2.1 Product Configuration

In the product database model, the 'PART_MODEL' entity represents end products that have components for possible product configurations. It is a sub class of the generic 'PART' entity that forms the product structure with the 'EDGE' entity through the 'has_child' and 'has_parent' relationships.

The 'PART_OPTION' entity that is also a sub class of the 'PART' entity represents the components of the product configuration. The 'PART_OPTION' instances belong to a 'PART_MODEL' instance and are specified with the 'EDGE_MODEL' entity that inherits the 'EDGE' entity. A 'PART_OPTION' instance could have a set of the 'PART_ASSY' instances that provide a physical assembly of product configurations. The 'PART_OPTION' entity represents variants that can change product structure and store a history of product structure changes. In order to represent changes of product structure, the effectivities are specified on the 'EDGE_OPTION' entity that links the 'PART_OPTION' to the 'PART_ASSY' entity. The 'EFFECTIVITY' entity represents the effectivity on the 'EDGE_OPTION' entity.

In order to specify the product configuration, the representation provides the 'CONFIGURATION' entity whose instances have a set of the 'PART_OPTION' instances. The set of instances represent a complete end product that has all the necessary functions of an end product.

4.2.2 Assembly Structures

To represent the physical assembly structure, the 'PART_ASSY' and 'EDGE_ASSY' entities form the node and edge of a directed acyclic graph data structure that represents conventional mechanical assemblies. They also inherit the 'PART' and 'EDGE' entities respectively in the representation. The inheritance mechanism of the data definition method in this representation helps us represent the complex product structure with simple specifications.

4.2.3 Multiple Representations

In this paper, we are only interested in the product representation for the parts catalogs among many product representations. The 'CATALOG' and 'SECTION' entities form a parts catalog and its contents. The 'SECTION_PART' entity links the 'PART_ASSY' to the 'SECTION' entity that reorganizes the list of parts for customers to easily identify the part they seek. The 'SECTION' entity is also a variant that can represent the different structures of the parts catalog, so the 'SECTION_PART' entity has effectivity values represented by the 'EFFECTIVITY' entity.

4.2.4 Engineering Changes

The 'EC' entity is a generic entity that manages identifiers of engineering changes and a history of product structure changes. The three sub classes of the entity, the 'EC_MODEL', 'EC_OPTION' and 'EC_ASSY' entities represent the engineering changes on each level of the product structure, *the model, option* and *assembly*, and provide additional information on different abstraction levels for each engineering change.

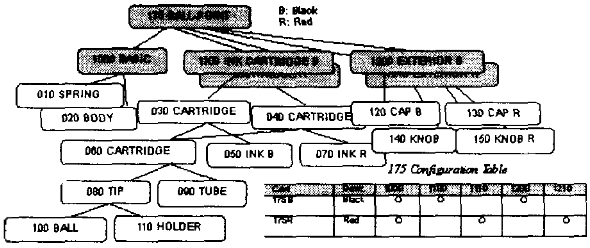
The 'EDGE_ARCHIVE' and the 'replaced' attribute of the 'EDGE' entity are used to manage and trace engineering changes. The 'EDGE_ARCHIVE' entity archives replaced or cut 'EDGE' instances between parts to maintain a history of structure changes. The 'replaced' attribute is used to trace the replaced product structure from the set of changed 'EDGE' instances stored in the 'EC' entity. The 'has history' relationship links a set of changed 'EDGE' instances. Finally, a selected 'EDGE' instance in the set of instances can link the old replaced edge through the 'replaced' attribute.

5. Implementation

A Web-based information system was developed based on the proposed product database model in order to support integrated product data for product design and technical publication. Using a relational database management system and Web programming languages, it translates the product database model to relational tables and corresponding application programs. In particular, we limit the user interfaces of the system to a HTML based environment to allow the access of any users who have standard Web browsers. This section illustrates the implementation of the product database model for product design and technical publication. To illustrate the implementation, a design and technical publishing process of a simple ball-point pen was constructed.

Figure 3 shows the product configurations and assembly structure of the example ball-point pen. The 175 ball-point pen has five options, components of the product configuration, that constitute two kinds of product configurations, black and red types (configurations 175B and 175R). The configuration table in the lower right corner of the figure shows the types of configurations made from the composition of the options.

The options are linked to the assembly structure with proper effectivity values of a model and its serial numbers. During engineering changes, the effectivity values represent the engineering changes with the serial numbers on which the engineering changes are applied. The assembly structure of the ball-point pen represents the physical composition of the product. The main engineering data such as part drawings and specifications will be maintained based on the structure.



[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1210
175B	Black	O	O		O	
175R	Red	O		O		O

Figure 3. The product structure of 175 Model

Figure 4 shows the mapping of the base product structure to the representation for the parts catalog. The parts catalog has two sections that reorganize the contents of the list of service parts. The edges in the mapping have the quantity data used for service operations and effectivity values of an option code and serial numbers for the propagation of engineering changes. The drawings in the parts catalog show engineering diagrams for each section.

Figure 5 shows an engineering change and its propagation to the parts catalog. In the figure, the colored parts and dot-lined edges are created during the engineering change. Effectivities on the edges between option '1100' and part '030' and the new part '031' have values that indicate when the engineering change is applied to the product. The engineering changes are applied from the 50th product of model 175.

The engineering change notification system notifies the technical publishers that there are changes on the base product structure. Then they review the changes and simply create new edges with proper effectivity for the new parts. In the figure, the publisher creates the edge between the '(2) CARTRIDGE BLACK' in the parts catalog and '031 CARTRIDGE BLACK' in the base product structure. This maintains the consistency between the parts catalog and base product structure.

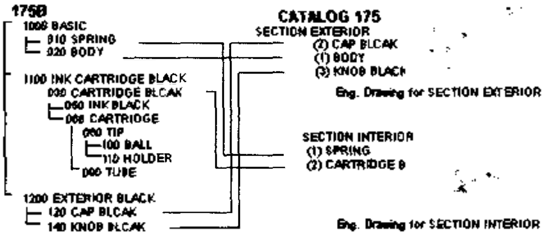


Figure 4. Mapping product Structure to parts Catalog

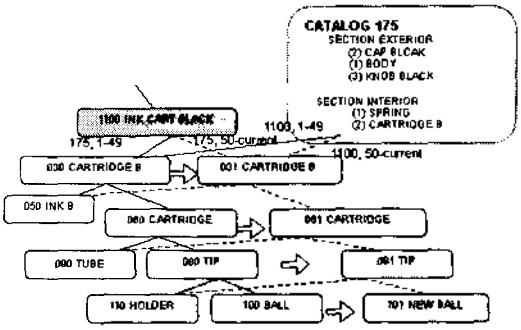


Figure 5. An engineering change and its propagation to the parts catalog

We are currently working on a technical publication system following the implementation of the functions for product data management. Figure 6 shows the windows of the implemented product data management and technical publication system: the product structure browser, engineering change notification system and parts catalog editor.

6. Conclusion

Web-based collaborative product development enables companies to develop their product with partners distributed around the world. However, current approaches to collaborative product development provide loosely integrated information systems via the Internet. This provides inconsistent product data and causes additional work and wasted resources for the participants.

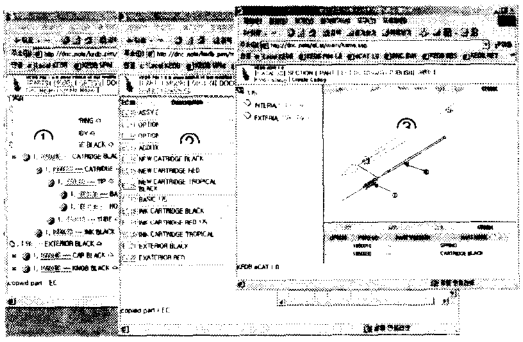


Figure 6. Windows of the implemented systems: 1) the product structure browser, 2) engineering change notification and 3) parts catalog editor

In order to avoid this inefficiency, we provide a framework that can maintain the consistency of product structure during collaborative product development. Based on the framework, we develop a product database model for product design and technical publication. The representation is implemented as a Web-based product data management and technical publication system in order to verify its feasibility. The system provides a representation for parts catalogs from the base product database model. It also maintains consistency between the base representation and the parts catalogs through the synchronization of engineering changes.

Many associated issues remain. One of these is the management of an interoperability matrix that shows whether a part can be replaced with other parts. The personalized on-line catalog system is another issue left for further research. It is strongly related to the issues of the multi-language support and configuration-based catalog publications.

References

[1] Agilesoft, Why Document Management Systems Don't Work for the ECO Process, White paper, Agilesoft, 1999.

[2] Baya, V., Information Handling Behavior of Designers During Conceptual Design: Three Experiments, Ph.D. Thesis, Dept. of Mechanical Engineering, Stanford University, Sep, 1996.

[3] Brown, D. C., Using Design History Systems for Technology Transfer, Proceedings of MIT-JSME Workshop on Cooperative Product Development, pp. 545-559, Nov, 1989.

[4] Enovia, Enovia VPM 1.0 Users Manual, Enovia, 1999.

[5] Hardwick, M., Downie, B., Kutcher, M. and Spooner, D., Concurrent Engineering with Delta Files, IEEE Computer Graphics and Applications, Vol. 15, No.1, pp. 62-68, 1995.

[6] Huang, G.Q. and Mak, K.L., Web-integrated manufacturing: recent developments and emerging issues, INT. J. Computer Integrated Manufacturing, Vol. 14, No.1, pp. 3-13, 2001.

[7] ISO 10303-203, STEP PART 203: Configuration Controlled Design, ISO, 1994.

[8] Kim, H., Lee, J. Y. and Han, S. B., Process-centric distributed collaborative design based on the Web, Proceedings of ASME Computers in Engineering Conference, DETC99/CIE-9081, Las Vegas, Nevada, 1999.

[9] Kim, Y., Choi, Y. and Yoo, S., Brokering and 3D collaborative viewing of mechanical part models on the Web, INT. J. Computer Integrated Manufacturing, Vol.14, No.1, pp.28-40, 2001.

[10] Shah, J., Jeon, D., Urban, S., Bliznakov, P. and Rogers, M., Database Infrastructure for Supporting Engineering Design Histories, Computer-Aided Design, Vol.28, No.5,pp.347-360, 1996.