Collaborative Engineering Design Based on an Intelligent STEP Database

Soonhung Han, Young Choi, Sangbong Yoo and Namkyu Park

Concurrent Engineering 2002 10: 239
DOI: 10.1177/106329302761689151

CONCURRENT ENGINEERING: Research and Applications

Collaborative Engineering Design Based on an Intelligent STEP Database

Soonhung Han,[1,*] Young Choi,[2] Sangbong Yoo[3] and Namkyu Park[4]

[1]*Department of Mechanical Engineering, KAIST (Korea Advanced Institute of Science and Technology), Daejon, Korea*

[2]*Department of Mechanical Engineering, Chung-Ang University, Seoul, Korea*

[3]*Department of Computer Science, Inha University, Incheon, Korea*

[4]*KITECH (Korea Institute of Industrial Technology), Korea*

Abstract: A pilot system has been developed to promote the collaborative engineering design on the Internet. Product models are stored in a shared database which is formatted according to the international standard STEP. The schema of the database is based on AP203 of STEP. Metadata and ontology concepts are used to improve the search capability of the product database. The system has been applied to a machining center example to verify its feasibility.

Key Words: collaborative design, content search, ontology, RDF, STEP.

1. Introduction

The STandard for the Exchange of Product model data (STEP) product model can be used to generate a database schema in a specific domain. STEP enables sharing enterprise information, exchanging information with business partners, and managing information of the product lifecycle in a virtual enterprise. An automotive company can manage and integrate parts information with collaborating parts suppliers. Utilizing the International Standards Organization (ISO) information standard STEP makes it possible for guest engineers to perform a collaborative design through the Internet. This paper describes a method to utilize a meta-data concept, a solid modeler, and a STEP database that enables the collaborative design of products on the Web.

Recent issues such as electronic commerce, digital mockup, and ERP (enterprise resource planning) are to improve engineering processes, and to reduce time and cost by substituting digital models for paper documents, physical models, and parts lists. The information standards allow sharing digital product models among participating domain experts. Product information created during the development process can be utilized for production, assembly and disassembly, sales, maintenance manual, and user education. Figure 1 shows diverse automation systems which participate throughout the product development process of an automotive manufacturer and shows the need for sharing the product model through a STEP database. The surface model generated by a style designer is transferred to engineering department where material and sizes of parts are determined. During this process iterative evaluations of the functional sub-systems are performed by using computational simulations. Engineering changes are made based on the performance estimations and simulations. The mature design model is transferred to manufacturing department where process planners, part programmers, and NC robots utilize the digital design model. Rapid

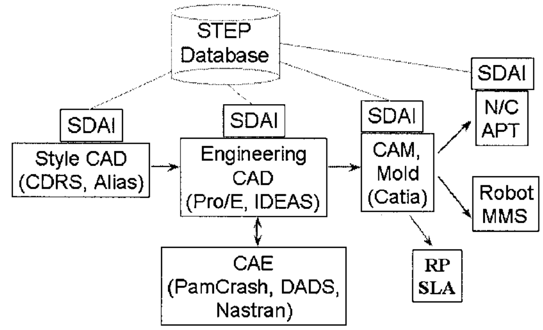


Figure 1. Flow of product information.

*Author to whom correspondence should be addressed.
E-mail: shhan@kaist.ac.kr

1063-293X/02/03 0239–11 $10.00/0 DOI: 10.1106/106329302030370

prototyping is another new tool which utilizes the digital design for fast turnaround.

As the Internet expands over the globe, it is inevitable to move some of the enterprise activities into virtual spaces where part manufacturers and maintenance engineers from remote sites collaborate to improve the product quality [22]. In order to provide the lifecycle supports for products, virtual enterprises should properly manage two types of data, i.e., the business data and the product data. Modern enterprises have experience in dealing with the business data, but it is not the case for the product data. Product data such as CAD and CAM data is usually managed only by the design and production engineers. However, for virtual enterprises, the product data need to be used in the earlier and the later stages of product lifecycle (e.g., Web catalog and service manual) [1].

Compared to the business data, the product data have complex semantics and cannot easily be exchanged among different application programs [2,3]. Even though some neutral formats of product data have been developed by standard organizations, translating them among various application programs still needs the comprehensive understanding of the complex semantics. Recently, it is widely recognized that capturing knowledge is the next step to overcome the current difficulties on sharing product data [4]. In this paper, we apply two types of knowledge (*metadata* and *ontology*) to search product data in virtual environments. Metadata means the data about the data and it serves as the *map* for locating a product data. Ontology is the working model of entities and interactions in some particular domain of knowledge or practices. Using metadata and ontology, a Web-based search system can be developed which facilitates the seamless sharing of product data among various application in virtual enterprises [21].

A STEP database has been constructed which contain product models of a machining center. To map a data structure expressed by the EXPRESS language to the relational database schema, we need special treatments. The proposed collaborative modeling system which modifies the STEP product model has been implemented with the Object Web [5] technology. It consists of server objects for modeling functions and arbitration between designer operations, and client objects for user interface. Interaction between client and server is accomplished by the CORBA middleware.

2. Searching for Product Data Using Knowledge

2.1. Metadata as a Map for Product Data

In a virtual enterprise, it is not unusual that design engineers are physically distributed, thus it is not trivial to locate specific design data. Metadata helps users in locating the necessary information about the product data. Entities included in the metadata are the product names, designer names, key features of the product, locations, and data formats. The metadata serves as a map for the product data.

The metadata is the major part of the knowledge base, but there are no standardized contents of the metadata because it is domain specific. In this paper, we assume that the product data is represented in STEP (e.g., Application Protocol 203: configuration-controlled design) [6]. We have classified the metadata into the following six categories:

(a) **Design**: Information about the STEP physical files. It contains the header information of the files and the information provided by the users when they register the files.
(b) **Registry**: Information about the registrar, including ID, name, email address, and registration date.
(c) **Part**: Information about the parts included in the STEP files. Because STEP can define assembly models, several parts can be included in a STEP file.
(d) **Document**: Information about accompanying documents such as ID, name, description, and document type.
(e) **Person**: Information about the related personnel such as ID, name, employer, and role.
(f) **Approval**: The approval information for parts such as status, approver, role, and date.

The complete list of the metadata is summarized in Table 1, where the category names are used for the *category search*. The metadata defined in Table 1 are modeled according to resource description framework (RDF) and extracted from STEP files. The RDF model of the metadata is not discussed in this paper due to the space limitation [23,24].

2.2. Ontology to Supplement the Semantics of Product Data

The metadata can be represented as ontology. We use ontology as the meanings and relationships of vocabulary to improve the search capability of the product database. Because products are developed by designers who have different backgrounds, different terminologies can be used for the same concept, or same terminology can be used for different concepts. For example, *resource* represents slightly different concept in workflow systems compared with process planning systems [7]. In workflow systems, *resource* means the information that is used to make decisions. In process planning systems, *resource* means a person or a machine that performs the given task. In order to precisely understand the vocabulary used in the product data, we should consider the context in which it is used.

Other than the keywords defined in standardized product models, engineers can select one of several

Table 1. Categories and elements of the metadata

Category	VI. Name of Elements	Descriptions	Category	VII. Name of Elements	Descriptions
Design	fileName	Name of File	Part	Part ID	Part ID
	fileDesc	File Description		Part Name	Part Name
	preprocessor	Preprocessor for the file		Part Desc	Part Description
	schemaName	Name of Schema		Level	Configuration Level
	url	URL		Quantity	Quantity
	fileSize	Size of File		Associated Doc	Associated Document
Registry	registrarID	ID of Registrar		Associated Person	Associated Person
	registrarName	Name of Registrar		Contained By	Part that contain this part
	registrarEmail	Email Address of Registrar		Contains	Part that this part contains
	registryDate	Date of Registry		Approval Info	Approval Information
Document	docID	Document ID	Approval	Approval Status	Approval Status
	docName	Document Name		Approved By	Who Approved
	docDesc	Document Description		Approval Type	Approval Type
	docType	Document Type		Approval Date	Approval Date
Person	personID	ID of Person			
	personName	Name of Person			
	employer	Employer			
	personRole	Role of Person			

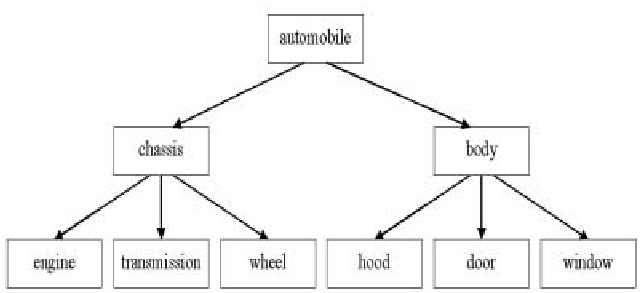


Figure 2. An example of *part-of* relationship.

synonyms to represent the same meaning. For example, [car, auto, automobile, motorcar] is an example of synonym list. Usually product designs have short descriptions about the product, which include the name, usage, specification, and key features of the product. Because the description is written in natural language, any word in a synonym list can be used. In a contents (or associative) search, we can significantly improve the results by using the synonym list.

The other common relationship among vocabulary is the *part-of* relationship. When an entity *A* is included in another entity *B*, entities *A* and *B* have the *part-of* relationship. Figure 2 shows an example of this relationship. An automobile consists of chassis and body. The chassis consists of engine, transmission, and wheel, whereas the body consists of hood, door, and window. The *part-of* relationships can be used to improve the search results. For example, when a user requests the product data of transmissions, the search engine can return the product data for chassis and its parent automobile too.

Still another application of ontology is searching the product data created in different languages. Engineers from different countries use different languages. For example, the country ''Korea'' is written as ''한국'' in Korean and ''韓國'' in Chinese or Japanese Kanji. Because proper nouns such as person name or geographical name may be written in native languages, the translation relationship can help engineers to search or understand the product data from foreign countries. Recent information systems such as Windows NT and STEP employ the Unicode to encode the multibyte character codes.

We can summarize the four requirements for the ontology as follows:

(a) **Context**: The application domain in which the word is used. The application domain can be matched with the application protocols defined in STEP. Some examples of the application protocols are associative drafting (AP202), configuration-controlled design (AP203), automotive (AP214), ship structure (AP218), and plant spatial configuration (AP227).
(b) **Synonym**: Synonymous relationship among word entities. When the synonymous relationship is defined, the context should be considered.
(c) ***Part-of* relationship**: The relationship among parts in the product structure. A part (or product, or subassembly) may contain many elements. At the same time, many parts may contain the same element. Therefore it is a directed many-to-many relationship.
(d) **Translation relationship**: It can be considered as the synonymous relationship among different languages. A word entity has the *name* of the language as its attribute.

A data model for the ontology is shown in Figure 3. The entity *word* has two many-to-many relationships, i.e., *part-of* and *synonym*. It has four attributes, i.e., *word-id*, *language*, *string*, and *context*. The data model can be implemented using either a relational database system or an object-oriented database system.

2.3. Web-based Search System for Product Data

The metadata is generated when users register data or documents. Most of the metadata is extracted automatically from STEP files. Because we designed the metadata schema based on STEP application protocols, it is not difficult to extract metadata from the given STEP files. The metadata generator interacts with the STEP file processor to extract the metadata and with the database interface for storage. A data search can be performed as a general search, category search, or extensible query language (XQL) pattern. For a general search, a keyword(s) is used to find all the metadata items that contain the keyword. The category search is based on the metadata schema as defined in Table 1. An example of the user interface for metadata search is shown in Figure 4.

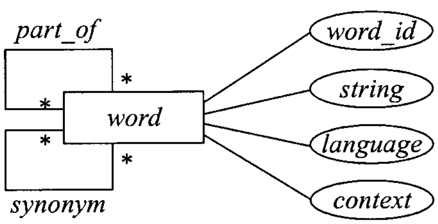


Figure 3. Data model of ontology.

Product data can be stored in either the standard neutral formats (e.g., STEP or IGES) or proprietary data formats. The standard neutral formats have been developed by standard organizations such as ISO or ANSI. The proprietary file formats, used in commercial systems (e.g., Catia, ProE, UG, ACIS, Parasolid, or AutoCAD^{TM}), are not suitable for data exchange among different systems because their data structures are not available in public. Utilizing the open architecture (e.g., Standard Data Access Interface: SDAI) and EXPRESS data models of STEP, we can implement the content search system. Users can search the product data with the characteristics such as name, material, and functions of the products.

However, words can be used with different concepts in different application domains. Traditional query may not perform a satisfactory search, or wrong products may be selected for given queries. In order to improve the accuracy of a search, we need to consider the context of given keywords. We can also leverage the content search by incorporating product structures. By considering the

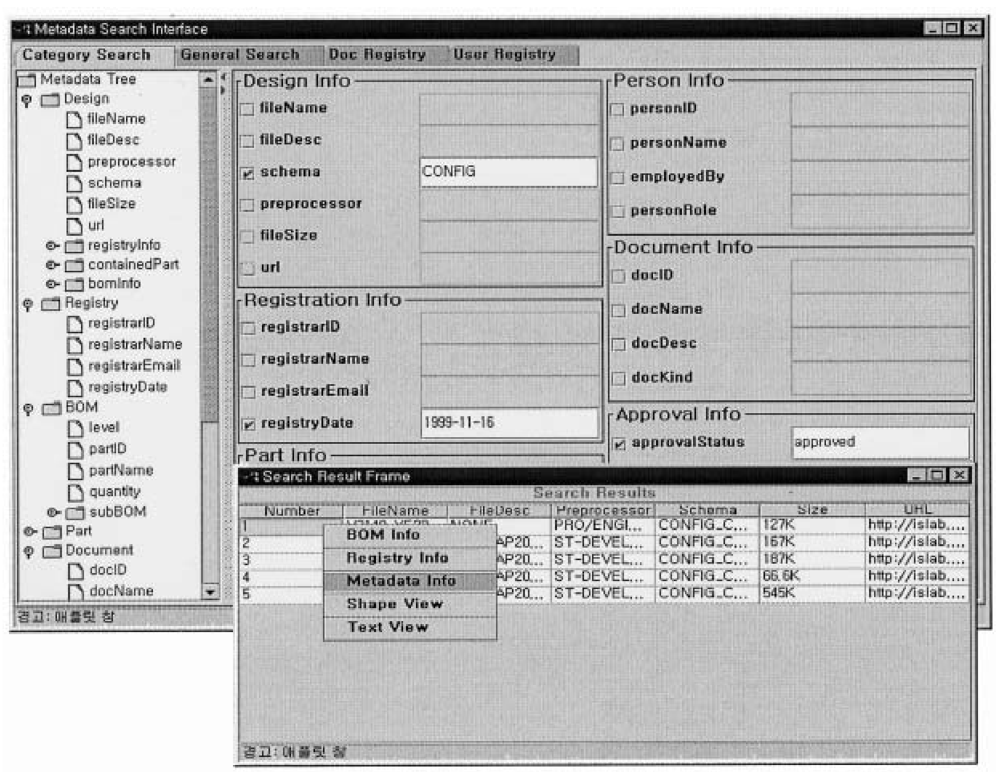


Figure 4. User interface for metadata search.

ontology of given keywords, we can utilize the existing design knowledge stored in the design database. The content search algorithm presented below is extended with ontology. Before a keyword search is made, the words that are related with the keyword are collected form the *part_of* relationship in the ontology base while considering the synonyms for each of the collected words in the same context. The procedure can be summarized as the following algorithm.

Algorithm 1 Content search with ontology extension
Input: A request (a keyword with context)
Output: List of product data: parts or products
Procedure:

1. Initialize an empty list *key_list* and *onto_list*.
2. Enter the given keyword into *key_list*.
3. Using the *part_of* relation in the ontology base, add all the words that include the given keyword to key_list.
4. For each word in *key_list*, enter the word and its synonyms into *onto_list*. Among the synonyms searched, select only those that have the same context.
5. For each word in the *onto_list* from Step 4, perform the traditional content search and return the results.

For example, suppose that a keyword and context pair is given as (chassis, AP214), where AP214 is the STEP application protocol for automotive mechanical design processes. The *key_list* in Step 2 of Algorithm 1 is now [(chassis, AP214)]. Assuming that we have the *part_of* relationship of Figure 3, the *key_list* becomes [(chassis, AP214), (automobile, AP214)] in Step 3. With synonym lists such as [car, auto, automobile, motorcar] and [chassis, frame], the *onto_list* becomes [(chassis, AP214), (frame, AP214), (automobile, AP214), (car, AP214), (auto, AP214), (motorcar, AP214)]. This list is sent to the traditional search engine, and a content search for each word and context pair is performed.

The content search system described in this section enables users or application systems to search for parts or products from existing product database without specific identifiers such as file names or drawing numbers. It promotes the utilization of existing design data. As a result, it can reduce the design lead-time and cost, and improve the design quality. An underlying basis of the content search system is utilizing the open architecture and interface methods of STEP. Because proprietary data formats used by commercial CAD/CAM systems does not have open architectures, we cannot search the contents with such a uniform interface method. On the other hand, the open structure of STEP enables us to monitor and trace the details of product data. Therefore, we can effectively manage engineering changes and their propagations based on the content search system described in this section.

3. Utilization of STEP Database on the Internet

3.1. Construction of a STEP Database

The STEP standard consists of seven interrelated classes of document parts; each class of parts is associated with a block of part numbers within the ISO 10303 standard. Three layers architecture of ANSI/SPARC was proposed as the core framework for database management systems, and has been widely used as a basis for other information systems [8]. The concept of three layers architecture which consists of internal or physical, conceptual or logical, and external layer has influenced the three layers of STEP standard which consists of implementation methods, integrated resource, and application protocols.

The database description is different from the database itself. The database description is called as the schema, and the process to generate the schema is called as the database design [9]. In STEP, product data models of a specific domain are described as application protocols (AP). An AP is documented in an application independent way using the EXPRESS language. The application activity model (AAM), application reference model (ARM), application interpreted model (AIM) of an application protocol provide the conceptual database schema [10]. STEP uses the formal information modeling language EXPRESS to specify the representation of product information. EXPRESS also provides facilities to establish an interface between schemas. To implement a STEP-based database, the EXPRESS description must be transformed to the database system data definition language (DDL) [11], which is the logical design stage of the database.

To construct a STEP-based product model database for a specific domain, the logical and physical database design processes should be followed after the conceptual schema is designed based on the application protocol. The structure of a database is often optimized on the basis of the type of queries that are known to be most prevalent. In the STEP application protocol which is used as the conceptual schema, the AIM model is well defined by the entity model. But, information is frequently accessed at the ARM level during the product design stage. It is necessary to provide both the access granularity based on the high level application program interface (API), and the data granularity using the ARM-like STEP model. To implement the product model database, the information model should be defined with the AIM elements, and the information system can be accessed based on both information models. In AP203, the AIM elements are defined considering occurrence of various information constructs.

Representation of different aspects of similar information objects is possible. For instance, to represent

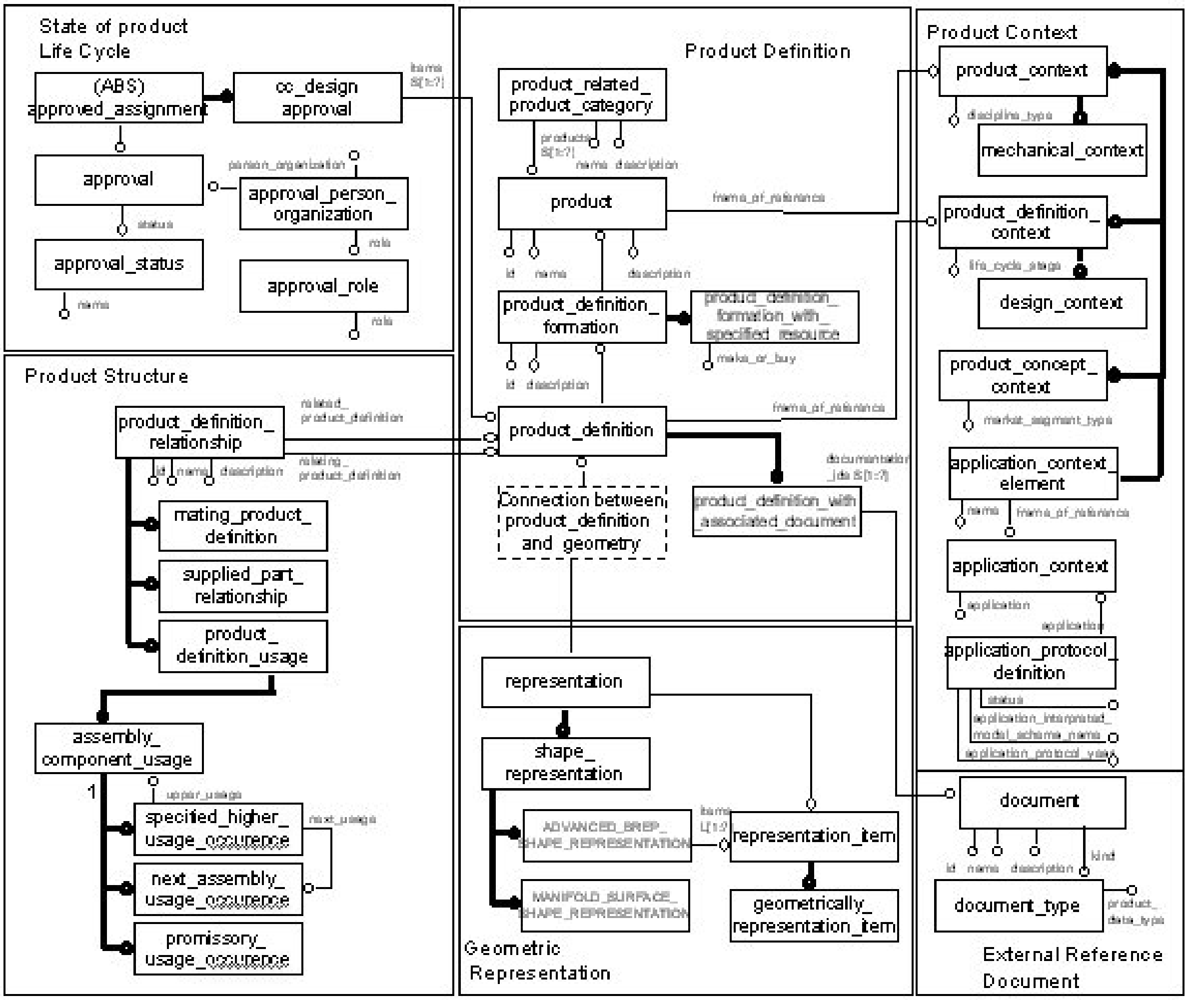


Figure 5. STEP AP203 AIM information model for the STEP database schema.

the product geometry, *shape_representation* class which is a subclass of *representation* is used, and this AIM element is the super class of *manifold_surface_shape_representation* and *geometrically_bounded_wireframe_shape_representation*. The proper subclass of *representation* should be decided according to the geometric information of the product. In this study, B-rep (boundary representation) solid geometry is represented using the *advanced_brep_shape_representation* class. Figure 5 shows the information model which consists of AIM elements of AP203. It is used to store the product model of TVL160, a machining center of Tongil Heavy Industry. This information model is a subset of AP203. STEP data instances based on this information model have been generated by the STEP module of the 3-dimensional CAD system SolidEdge. The data load manager stores the STEP physical files of TVL160 into the STEP database, and all the parts and assembly models of TVL160 are stored in the STEP database. The TVL160 is made of 23 design units which is again consists of numerous parts.

The Oracle relational database is used for the storage and retrieval of the STEP database. EXPRESS has the object-oriented description structure, and the STEP data evolving from the generation process inherits the characteristics of EXPRESS. To map various data structures provided by EXPRESS into the RDB schema, another mapping process is necessary. *Entity type* is mapped into a *relation*. *Aggregation type* is mapped into one *relation* for entity itself, and another *relation* for several related entities. To process the generalization hierarchy modeling, *super entity* is mainly used. The product structure schema is defined by a subtype of *product_definition_relationship* of Part 41 of STEP. The product model generated in this research is composed of direct parents. One table named *product_definition_relationship* is used to represent the assembly information [25].

3.2. Utilization of the STEP Database Using CORBA

An integrated database based on STEP is constructed using the Oracle database system, and the data

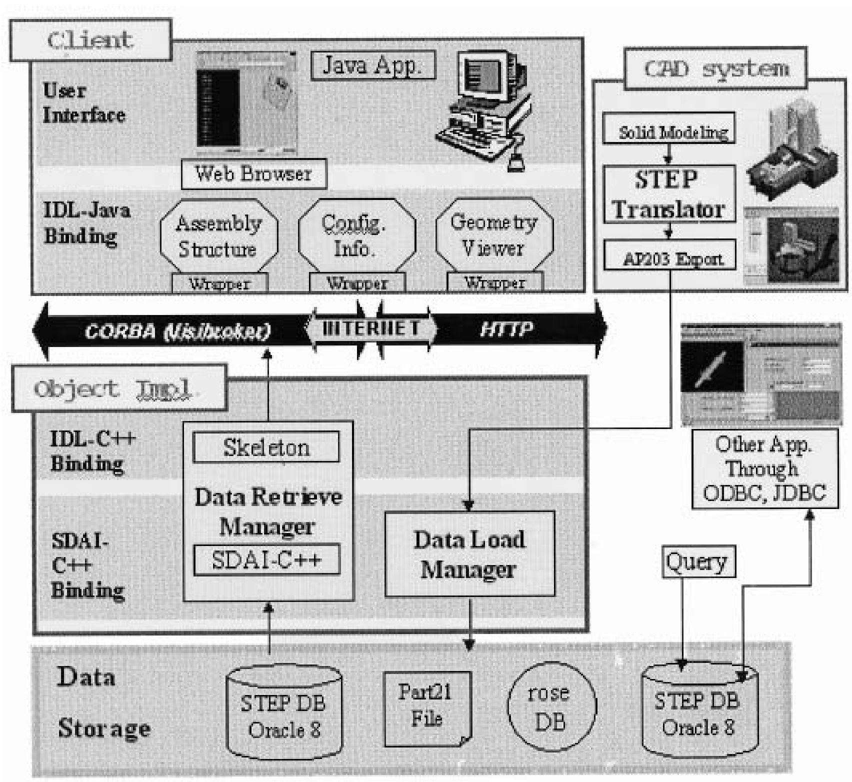


Figure 6. System overview of the STEP database.

input–output module is constructed using CORBA. Figure 6 shows the system overview of the STEP database on the Internet. The data retrieval manager, which consists of the skeleton code and the SDAI C++ binding code, is invoked when the client requests data. It retrieves the product structure information, the assembly information, and the product geometry. Then it hands them over to the CORBA client. The client applet visualizes the assembly structure using the JFC JTree GUI, and the surface geometry using JAVA. This CORBA client applet allows access to the STEP database by using a Web browser.

CORBA enables consistent access from different host platforms, different operating systems, and therefore enables the application integration. Geographically distributed modules, and heterogeneous modules within a system can be integrated through CORBA. The STEP database is constructed using Oracle-8i on the Sun Solaris 2.6 operating system, and the object implementation is built up using the skeleton code and the code for database retrieval [12,13]. The client application is programmed as a Java applet which can be executed on any hardware or software environment. Sharing of engineering information using the STEP database reduces the design inconsistency of a company, supports the lifecycle of a product, shortens the product development lead time, and raises the competitive powers of the company.

4. Collaborative Geometric Modeling System

4.1. Separation of Modeling Kernel for User Interface

Geometric models play an important role in a collaborative design session. Because the geometric model of an engineering product is large and complex, it is not usually stored in a database. Also the developer should decide how to separate the software modules of cooperative solid modeling applications in a distributed environment. The decision must be made based on the fact that which information should be shared and which should not. Another factor is to minimize the network traffic which is generated by distributed modeling operations [14,15,16].

The geometric model and the data from a modeling operation must reside in the modeling server since the model is managed by the modeling kernel. User interface functions including data input and display of the shape can be located at client machines. Although most display functions are provided as client functions, the actual implementation of the display operation must be divided between the server side module and the client side module at a certain level of rendering operation. For example, the server may generate a bitmap image of the shape based on the transformation matrix provided by the client, and send the image back to the client.

In this case, the display response time from the server is constant (screen full of bitmap image) regardless of the size or complexity of the model. However, the real time response of display is almost impossible to achieve because of the time necessary for the image transmission.

If the server sends out a triangulated model with attributes to clients, clients can share views of model based on the view transformation data from the server. In this case, rendering response depends only on the capacity of the client machine. However, clients need to receive triangles and viewing attributes again whenever the model is modified. The display response time can be enhanced by controlling the sizes of the triangles according to the condition of network and client's needs such as the level of detail.

Another aspect that needs to be considered in a collaborative modeling session is that operations on the same model requested from different clients can conflict with each other. To avoid this concurrency problem, clients are allowed to edit the model only through services provided by the server. Only the modeling kernel in the server has the direct access to the model under operation. Locking operation is also possible to prohibit modifying topology and geometry data which is under modification by another client. The server sends out a message to clients to notify the access status.

4.2 Implementation

4.2.1 ARCHITECTURE OF THE IMPLEMENTATION

Figure 7 shows components of the client and server modules. The server objects are implemented with C++ language. Server objects include the session objects that manage clients and play the role of wrapper to bridge CORBA and the solid modeling kernel. Objects for modeling operation and objects of the modeling kernel are also included in the server objects. Model data and annotation data are stored in the server database. This client code consists of the user interface implemented with Java3D [17] and the Swing package, the callback object to respond to requests from the server, and the core objects for ORB (Object Request Broker) connection.

All the model data resides in the server, and modeling operations of the collaborative modeling system are executed at the server. Only a subset of the model data resides at the client side. They are the triangulated shape data and the corresponding index that relates display data. They are the minimal set of data necessary for display and picking operation at the client site. Additional model data can be retrieved by making explicit requests to the server.

4.2.2 IMPLEMENTATION TOOLS AND ENVIRONMENT

The first step in the software development with CORBA is to define the interface between client and server with IDL (interface definition language) [19,20]. Interface definition between modules is important even in the development of standalone applications. Once the interfaces between modules are clearly defined, it is easy to describe it with IDL. Clients and the server of the pilot implementation interact through the API wrapper of the solid modeling kernel and interfaces which control the user login and logout.

The IDL interface definition is compiled to generate the server skeleton and the client stub. We have used Orbix from Iona Technology as the IDL compiler to generate the C++ server skeleton and OrbixWeb [18] to

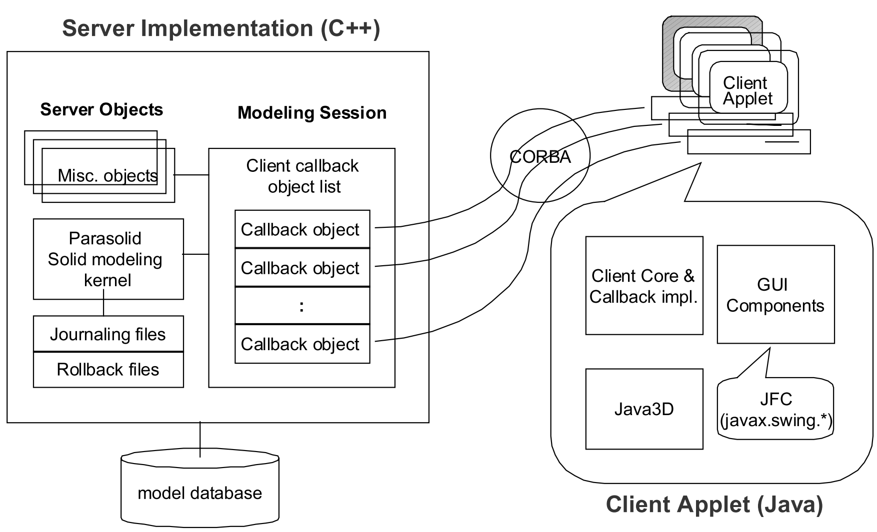


Figure 7. Modeling system for distributed collaborative design.

generate the Java client stub. The server has been implemented with the Parasolid solid modeling kernel from Unigraphics Solutions and the Orbix ORB library. The server has been compiled with the Microsoft Visual C++. The client code has been implemented in Java platform 2 with JDK and Java3D from Sum Microsystems to achieve the hardware and O/S independence. The client code may be provided from the Web server as an applet with ORB classes.

4.3. A Typical Session of Collaborative Modeling

The modeling scenario of a collaborative design within the implemented system is similar to the process to invoke general Object Web applications. A designer connects to the web server and opens a document that contains the client applets. The client software is then downloaded and executed to connect the designer to the modeling server. The request by the first client invokes the server programs and activates session objects. Once the session starts, the solid modeling kernel is initialized and modeling operations can be called by the client. If another client or designer wants to join the current session, the session manager decides whether to allow the client to join the session or not.

Figure 8 shows the process of modifying a simple block to model a machine part. Screen shots 1, 3, 4, 6 are captured from the designer α computer, and screen shots 2,5 are captured from the designer β computer. (The dialog box has a different color strip.) A circle is sketched on a block by the designer α in the screen shot 1. The designer α inputs the extrusion height in the screen shot 1. As the result of this operation, designer β's screen is automatically updated (extruded) with the modified shape. The designer β generates a hole inside the extruded cylinder in the screen shot 2. The result of this operation is again shown on the designer α screen in the screen shot 3. Following pictures show other collaborative operations and blending operations. This modeling session has been carried out on two PCs connected with LAN.

5. Conclusion

This paper describes a collaborative design system which utilizes the metadata, the STEP database and the collaborative geometric modeling on the Internet. Metadata is used to search for product models using the content search. The product model database has been constructed according to the STEP concepts. And the models from the database can be manipulated collaboratively.

We have investigated the knowledge that helps users or application systems to share product data in a distributed and virtual enterprise environment, and developed a prototype of Web-based search system. The knowledge we have considered is metadata and ontology. The metadata provided a map of product data space that can facilitate navigation of the product data. The concept of ontology is adopted to supplement the representation of product data. We have defined

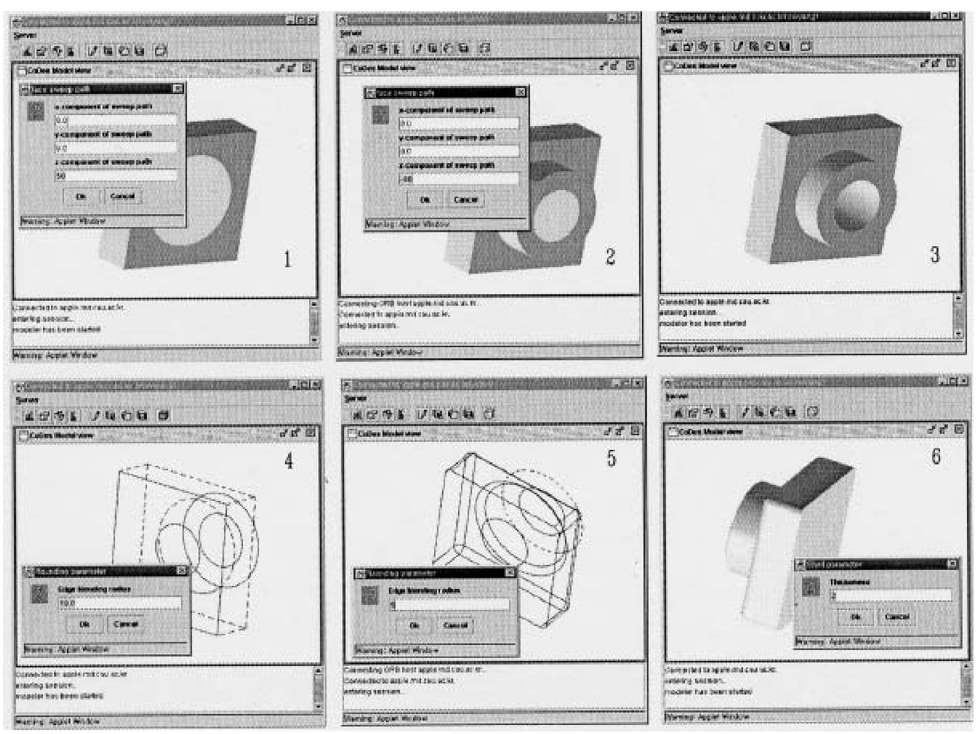


Figure 8. A process of real-time collaborative design.

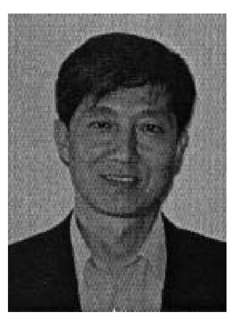
context, *synonym*, *part-of*, and *language translation* relationships, which can significantly improve search capability by extending traditional keyword-based query into content-based or product structure-based search of product databases.

A STEP database has been constructed by storing product models according to the STEP standard model. The logical schema is generated from the AIM schema of AP203. To apply the STEP database to an industry example, the product data of a machining center is constructed as the STEP database. The modules to manage and to retrieve the STEP database on the Internet have been implemented using CORBA and Java. Collaborative 3D modeling can be achieved on any Web browser though the proposed distributed collaborative modeling environment. Introduction of Java and CORBA technology into the solid modeling system enables remote 3D modeling services on the Internet, and the real time 3D collaborative modeling among heterogeneous platforms is possible. Sharing of 3D design data through the network without a high priced and complicated software may be another benefit of the proposed environment.

References

1. Hardwick, Martin, Spooner, David, Rando, Tom and Morris, K.C. (1996). Sharing Manufacturing Information in Virtual Enterprises, *Communication of the ACM*, **39**(2): 46–54.
2. Carter, D. and Baker, B. (1992). *Concurrent Engineering: The Product Development Environment for the 1990s*, Addison-Wesley Publishing Company, Reading, Massachusetts.
3. McKay, A., Bloor, M. and de Pennington, A. (1996). A Framework for Product Data, *IEEE Transactions on Knowledge and Data Engineering*, **8**(5).
4. Liebowitz, Jay (1999). *Knowledge Management Handbook*, CRC Press.
5. Orfali, R., Harkey, D. and Edwards, J. (October 1997). CORBA, Java, and the Object Web, *Byte Magazine*, http://www.byte.com/art/9710/sec6/art3/.htm.
6. ISO TC184/SC4, STEP Part203: Configuration Controlled Design, (1994).
7. Schlenoff, C. Ivester, R. and Knutilla, A. (1998). A Robust Process Ontology for Manufacturing Systems Integration, In: *Proceedings of 2nd International Conference on Engineering Design and Automation*, Maui, Hawaii.
8. Folwer, Julian (1995). *STEP for Data Management Exchange and Sharing*, Technology Appraisals.
9. Elmasri, R.A. and Navathe, S.B. (1994). *Fundamental of Database Systems*, Addison-Wesley.
10. Krebs, T. and Luehrsen, H. (1995). STEP Database Integration Platform for Concurrent Engineering. In: *Proceedings of 2nd Int. Conf. on Concurrent Engineering*, pp. 131–142, (McLean, Virginia, August 23–25), Concurrent Tech. Co., Johnstown, PA.
11. Loffredo, Thomas (1998). *Efficient Database Implementation of EXPRESS Information Models*, RPI.
12. Inprise (1998). *Visibroker for C++ ver. 3.3 Programmer's Guide*.
13. Inprise (1998). *Visibroker for Java ver. 4.0 Programmer's Guide*.
14. Potter, C. (November 1997). Web-Enabled Engineering: Step-by-step, *Computer Graphics World*, pp. 64–69.
15. Kempfer, L. (April 1999). Tools for Web Collaboration, *Computer-Aided Engineering*.
16. Anand, V.B. (1993). *Computer Graphics and Geometric Modeling for Engineering*, John Wiley and Sons, Inc.
17. Bouvier, D.J. and Dankwardt, K.P. (February 1999). *Getting Started with the Java 3D™ API*, K Computing.
18. (1996). OrbixWeb™ *Programming Guide*, Iona Technologies.
19. Siegel, J. (1996). *CORBA Fundamentals and Programming*, John Wiley and Sons, Inc.
20. (October 1999). *CORBA Manufacturing: Manufacturing Domain Specifications*, Object Management Group Inc.
21. Han, S., Choi, Y. and Yoo, S. (March 2000). STEP Technology for Factory Applications (in Korean), Report of Korea Institute of Industrial Technology.
22. Jonathan, F. Gerhand, David Rosen, Allen, Janet K. and Mistree, Farrokh (2001). A Distributed Product Realization Environment for Design and Manufacturing, *Journal of Computing and Information Science in Engineering*, ASME, **1** (3): 235–244.
23. World Wide Web Consortium (2000). Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation 27 March 2000, available at http://www.w3.org/TR/2000/CR-rdf-schema-20000327/.
24. World Wide Web Consortium, Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, available at http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.
25. Oh, Youchon, Han, Soonhung and Suh, Hyowon (2001). Mapping Product Structures between *CAD* and PDM Systems Using UML, *CAD*, **33**(7): 521–529.

Soonhung Han



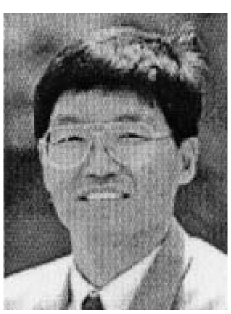
Soonhung Han is an Associate Professor in the Department of Mechanical Engineering (http://me.kaist.ac.kr) at the KAIST (Korea Advanced Institute of Science and Technology, www.kaist.ac.kr). He is leading the Intelligent CAD laboratory (http://icad.kaist.ac.kr) at the KAIST, and the STEP Center of Korea (http://kstep.or.kr). His research interests include the STEP (ISO standard for the exchange of product model data), VR (virtual reality) for engineering design, and knowledge-based design system. His domain of interests include automotive and shipbuilding industries. He has a BS and a MS from the Seoul National

University, another MS from the University of Newcastle upon Tyne of UK, and a Ph.D. from the University of Michigan of USA. He is involved in the professional societies of CAD/CAM (www.cadcam.or.kr) and CALS/EC (www.calsec.or.kr) in Korea. He is the editor-in-chief of the Web-based journal, International Journal of CAD/CAM (www.ijcc.org). He is a key member of the ERC (engineering research center) for VR (www.vrrc.org) of Korea.

Young Choi

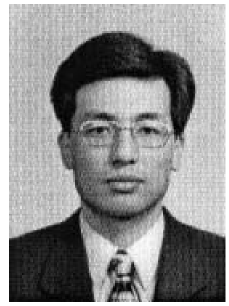
Young Choi is a Professor at the Department of Mechanical Engineering of Chung-Ang University, Seoul, Korea. His current research interests include STEP-based collaborative engineering, design analysis integration and Web technologies for engineering collaboration. He received BS in Mechanical Engineering from Seoul National University in 1979 and MS in Production Engineering from KAIST in 1981. He also received Ph.D. in Mechanical Engineering from Carnegie Mellon University in 1989 for his thesis on Non-manifold representation of solids. He has three years industry experience from 1981 until 1984 and several years of research experience in KIST, Korea and NIST, USA. He has published a number of papers on geometric modeling, STEP-based application and collaboration in design.

Sang Bong Yoo



Sang Bong Yoo received a B.S. degree in Control and Instrumentation Engineering from Seoul National University, Seoul, Korea in 1982, and M.S. degree in Electrical and Computer Engineering from the University of Arizona in 1986, and a Ph.D. degree in Electrical and Computer Engineering from Purdue University, W. Lafayette, Indiana, U.S.A. in 1990. Currently, he is a Professor of Department of Computer Science at Inha University, Incheon, Korea. Previously, he was a Technical Manager at Samsung Electronics Co., Seoul, Korea. His research interests include Engineering Databases, Distributed Systems, System Integration, and Knowledge Management.

Namkyu Park



Namkyu Park earned the B.S. ('86), the M.S. ('88), and the Ph.D. ('94) degree in industrial engineering from Seoul National University in Korea. He was in the School of Industrial Engineering at Purdue University as a Postdoctoral Researcher. He was a Faculty Member of Dong-Shin University in Korea. Since 1996, Dr. Park has been with Korea Institute of Industrial Technology (KITECH), which is one of the national research institutes in Korea. His research interests include information system, enterprise modeling and process engineering, electronic commerce, supply chain management, and intelligent manufacturing system. Recently he founded a startup company focusing integration aspects of information among organizations. Dr. Park has also had keen interest in national industrial information policy planning since he joined the KITECH. He is actively serving as the member of several committees for national technology policy planning, and Korean delegation to the international IMS program.