Integration of heterogeneous CAD databases using STEP and the Internet

Yongjae Shin a,1, Soon-Hung Han a,*, Doo-Hwan Bae b,2

a Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, 373-1 Kusong-Dong Yusong-Ku, Taejon 305-701, South Korea
b Department of Computer Science, Korea Advanced Institute of Science and Technology, 373-1 Kusong-Dong Yusong-Ku, Taejon 305-701, South Korea

Abstract

In a large-scale design process, designers cooperate in a complex situation where a variety of software tools run on different hardware platforms. This paper presents a data enhancement approach to integrate heterogeneous Computer-Aided Design (CAD) databases through the Internet. The data enhancement means topological changes in a geometric model and additional information in design semantics. The geometric data is enhanced using a non-manifold modeler to produce data sets valuable in downstream applications such as a Finite Element Method (FEM) solver or a detail design system. As a practical example, a shipbuilding product model database has been implemented based on the Standard for the Exchange of Product (STEP) model methodology and shipbuilding features. The system has been implemented on a network environment that consists of a Web browser, Common Object Request Broker Architecture (CORBA) objects, a relational database management system, a data enhancement module, and various computer-aided applications. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: STEP; Shipbuilding feature; Non-manifold model; CORBA

1. Introduction

An engineering database should store a consistent set of product data for a Computer-Integrated Manufacturing (CIM) system. The Standard for the Exchange of Product (STEP), which provides a representational scheme for product information, is emerging while related research [6] is being conducted. The objective of STEP is not only to provide a neutral file exchange format, but also to provide a basis for both implementing and sharing product databases. Recently, Product Data Management (PDM) systems were developed for the management of engineering databases. Although a few commercial PDM systems are on the market, they are currently in their infant stage. One of the major problems to be solved before the widespread use of PDM systems is to allow the exchange of design data among various application systems, each of which uses a different data representation scheme. For example, the geometric data for CAD systems are represented in various schemes such as 2D graphic...
entities, 3D wireframe models, surface models, or solid models. These schemes have topological differences. Furthermore, even in a single representational scheme, the geometric entities are incompletely constructed along with the levels of design detail. An initial design has far less informational content than a detailed design. As a result, one-to-one translations of data items may not work as intended. Therefore, this problem cannot be solved by simple data translations. Data enhancement that includes data addition and modification is required.

This paper describes a data-sharing approach through data enhancements on a network environment. This approach consists of product modeling according to STEP methodology, feature recognition from 2D drawings, and topological changes in the geometric data using a non-manifold modeler and Common Object Request Broker Architecture (CORBA) objects in the network. We apply our approach to the exchange of ship design data, where the 2D drawing data is enhanced to the non-manifold model by feature recognition rules of ship design, and it is then stored as the property of the product model. The architecture of this approach can be applied to a PDM system since its architecture has a geometric modeling module interfaced to an engineering database.

The data exchange system is implemented on a network using both the Web and CORBA. Through the Web, various applications can share data without additional effort by the user. The communications between applications that are not supported by the Web require complex network programming. On the other hand, CORBA is used to help simplify this task. CORBA is a distributed object-computing infrastructure that has been standardized by the Object Management Group (OMG) [14]. It defines the Application Programming Interfaces (API) that enable client–server object interaction within an implementation of the Object Request Broker (ORB). The client module of the developed system is implemented as a JAVA applet running on a Web browser, while the server is located in the data enhancement module.

The domain of our application is Hyundai Heavy Industries (HHI) of Korea. Fig. 1 shows the flow of design information among different Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) systems, which are used for ship structural design at this company. Different CAD systems are used by different departments and the design results are usually delivered as paper drawings. The 2D initial structural design data from the AutoCAD system was remodeled as a 3D Finite Element (FE) model within a Finite Element Method (FEM) preprocessor, while a 3D hull model was reconstructed by the AutoDef system for production design. This situation required a data enhancement strategy that involves topological changes in geometric models in addition to one-to-one translation.

The product model is created by recognizing shipbuilding features from 2D drawings for the data exchange. Most previous feature recognition research has been concerned with the determination of manu-

![Diagram of design information flow](image-url)
facturing features from 3D solid models. Construction of 3D models from 2D drawings has been an interesting research subject for some time, but it was mainly related to reconstruction of a single machine part and required computational complexities \[9,17,18\]. Dori and Karl \[4\] proposed a scheme to achieve a conversion of engineering drawings into 3D CAD models based on semantic attributes. Pratt \[16\] explained the role of features in the product’s life-cycle, transforming the designer’s feature model into the feature model for downstream applications.

The STEP Mosaic project \[7\] is the first effort for product data sharing on the Internet, in which the STEP-based product model, CORBA, and the Web are used. The National Industrial Information Infrastructure Protocols (NIIIP) \[25\] consortium proposed a protocol that can be used for a virtual enterprise. Hardwick et al. \[8\] use STEP and CORBA to develop a prototype information infrastructure for virtual manufacturing enterprises. The goal of the Rapid Design Exploration and Optimization (RaDEO) \[3\] project is to develop the Integrated Product Data Environment (IPDE) system for information integration capabilities in order to improve the speed and quality of the engineering analysis process. An infrastructure for design retrieval has been developed based on the STEP.

2. Midship product model

Within the ship structural design office of the HHI, the data to be exchanged represented the midship drawings that constitute the basis for ship structural design. These 2D drawings are usually produced by the AutoCAD system. Fig. 2 shows a typical midship section drawing. Geometric information of the drawing is different from that of the 3D model for production design. The real model is scaled down (100:1) to simplify geometric data, and

![Fig. 2. Typical midship section drawing of a bulk carrier.](image-url)
geometric entities are omitted since precise shapes are not needed. Numerous changes are required if a real 3D model is reconstructed from 2D drawings. The task of data enhancement should make the implied information explicit, in addition to data mappings between different design models. From these drawings stored in the DXF [2] file format, LINE and ARC entities are extracted to recognize shipbuilding features. The shipbuilding features defined for the midship of a bulk carrier are deck, girder, longi, top tank, slant, web, bottom, double bottom, and side shell. These shipbuilding features are then used to construct the midship product model.

A product model should maintain consistency in order to manage heterogeneous data throughout the product’s life-cycle. It must support various data representation schemes of each CAD/CAE system, and a systematic methodology is also required. STEP provides a mechanism that can manage product data throughout the life-cycle of a product, independent from any other particular system. STEP [20] is organized as a series of numbered parts. Implementation methods such as Part 21 [12] (the physical file format), and Part 22 [11] (the method of storing and retrieving STEP data in a general database management system), are combined with a shipbuilding Application Protocol (AP) to form the basis of an actual STEP implementation. STEP uses a formal specification language EXPRESS [13] to model the product information to be represented. Use of the formal specification language enables precision and consistency in representation and facilitates implementations. Since definitions of a product are separated from their data values, a product model can be built systematically.

In this paper, the midship schema is developed using the STEP methodology to store the design data recognized from 2D drawings. This schema consists of four Building Blocks (BB), one called ship_mid is newly developed in this work, and another called hull_cross_section is modified from the European Marine STEP Association Building Block (EMSA BB) [5] to hold information related to feature recognition. In addition, two existing EMSA BBs, support_resources and generic_product_structures, are used. The EMSA BBs are the results of the shipbuilding STEP AP development efforts. A building block is an EXPRESS-based specification that is used for the definition of Units of Functionality (UoF). A UoF may include zero, one, or several BBs recursively. A BB consists of three schemas, an import schema, an export schema, and a model schema.

Among four BBs, the hull_cross_section BB, which is used for recognition of 2D drawings, has four entities: Section_line, Feature_point, Indexed_feature_points, and Hull_cross_section. The Section_line entity holds 2D line segments of the drawing, and the Feature_point entity is used to determine shipbuilding features. The Indexed_feature_points entity holds lists of Feature_points while the Hull_cross_section entity holds the final information for the shipbuilding features. Fig. 3 shows the Hull_cross_section entity of a bulk carrier, which is represented by 20 sorted Feature_points. These schemas have to be prepared before the recognition process of the drawing. Each Feature_point is determined according to recognition rules, which utilize identified segments. An identified segment is recognized by comparing attribute values of the Section_lines. The Section_line holds direction, length, and coordinate values of a line segment as its properties.

The shipbuilding features are defined in the ship_mid BB, and written in EXPRESS language based on the type of ship, designer’s intentions, and design conventions. The defined shipbuilding features of the midship of a bulk carrier are deck, girder, longi, top tank, slant, web, bottom, double bottom, and side shell. They are comparable to machining features.

![Feature_point](image)

Fig. 3. Example of Hull_cross_section entity.
such as hole, slot, and pocket. The 2D data recognized according to the hull_cross_section BB is then enhanced to a 3D ship model. The ship_mid BB is defined to hold this enhanced data. 3D Plate entities in the ship_mid BB are generated from the recognized Hull_cross_section entity. The 3D shipbuilding features are then created based on 3D entities and feature definitions.

3. System architecture for the enhancement module

The data enhancement module, as shown in Fig. 4 has been developed using the ACIS [21] library. This supports non-manifold modeling and also processes low-level geometric operations. The enhancement module also uses the STEP toolkit ST-Developer [24] with the ROSE database. The ROSE is a part of the STEP toolkit and a STEP-compliant database to manipulate a product data defined by the EXPRESS language. This enhancement module starts from the recognition of 2D drawings. The results from this recognition procedure are then stored as shipbuilding features. The feature model is enhanced into a 3D non-manifold model. Next, both the topology data of the feature model and the geometric data of the non-manifold model are translated to proprietary data structures for downstream applications. A non-manifold model can accommodate 2D, 3D wireframe, and solid models all at the same time. Therefore, the enhancement module can support flexible and unrestricted representations of geometries such as wire edges and surface elements. The entire product information can be stored in one data structure from a 2D wireframe model in the initial design stage to a complete 3D solid model in the production stage. The data enhancement module consists of nine components. Each of these nine components is explained in the following nine sections.

3.1. Midship schema

A midship schema, written in the EXPRESS language, is compiled into C++ classes. To handle the midship model composed of EXPRESS constructs, the ST-Developer has been used. C++ classes generated by the EXPRESS compiler of this toolkit embody both member data and functions. They are supplied by the ROSE in addition to the data defined in the midship schema. To handle the C++ objects, the ROSE database holds the corresponding schema information as a compiled class is instantiated. The ROSE classes consist of the following: STEPObject classes, which are base classes of the compiled class, RoseInterface classes, which provide a high-level programming interface to the database, and Internal classes, which manage C++ objects. Together, they provide functions necessary to store, load, and search C++ objects [23] such as STEP product models.

3.2. ROSE database

The ROSE is a STEP compliant product database. This is an object-oriented database that adopts the entity definitions in the STEP, and then supports an environment for product modeling. It uses the EXPRESS language as its data definition language, and then supports the EXPRESS compiler when producing C++ classes. These C++ classes are used to develop application programs in a STEP application.

3.3. Recognition of 2D drawings

The main idea of data enhancement is to refine the geometry of the midship, since initial design data concentrates on the overall arrangement of the ship’s components. This is essentially based on pattern matching. The recognizer of the enhancement system is provided with a library of patterns corresponding to the hull’s cross-section from a bulk carrier. Each shipbuilding feature can be extracted using these patterns. Two kinds of drawings are analyzed to construct a 3D ship model. One is a typical midship section drawing that gives both a front view and a top view of the midship, and includes bulkhead information. The other is an elevation drawing, which gives the side view of the midship. Feature points are further defined so to extract lines from the 2D drawings. They are needed to recognize basic
shipbuilding features and are determined by comparing attributes in identified segments. To analyze such drawings, feature recognition rules are defined, which includes domain-specific knowledge.

3.4. ACIS geometric toolkit

For a complete geometric model required by the data enhancement, a non-manifold model is first
created by functions of a modeling toolkit, ACIS. The non-manifold geometric modeler ACIS, an object-oriented toolkit, is available to CAD system developers and is now accepted as modeling kernels in several commercial CAD systems. To create and manage a geometric model, ACIS offers C++ classes, API functions, and Direct Interface (DI) functions. The enhancement module uses ACIS functions in its two portions. One uses the kernel of ACIS to process non-manifold data. The other uses the ACIS 3DT (3D toolkit) that consists of the following: Geometry Husk, Graphic Interaction Husk, Part Management Husk, and Schema Interpreter Husk. The ACIS 3DT offers other functions for the creation of geometric data such as curves or surfaces, and functions to handle coordinate systems as well as user interaction.

3.5. Translation to finite element model

For the structural analysis system, analysis data such as the elemental type, number of nodes, Young’s modulus, and Poisson’s ratio should be first extracted from the ship product model. Then to create finite element meshes, a new C++ class derived from the MESH_MANAGER class of the ACIS geometric toolkit is defined while the faceting API function of ACIS is used. This faceter computes a number of facets and then outputs them according to the protocol. The three provided protocols are: (1) coordinate protocol, which outputs the polygon vertices as explicit coordinates; (2) indexed protocol, which first outputs a list of coordinates of all the vertices in a face mesh, then outputs polygon vertices as indices referred to this list; and (3) global indexed protocol, which is similar to the indexed protocol, but also considers the entire body as a single mesh. The indexed protocol is used in the derived class to generate analysis data. The REFINE-MENT class is also used for the accuracy of the meshes. The class controls the maximum size of a cell, maximum number of subdivisions on a face, approximate aspect ratio of each grid cell, triangular control to specify the extent of triangulation, and triangular smoothing. Fig. 5 shows a finite element

![Finite element model for the Nastran system.](image)
model for the structural analysis solver NASTRAN, using default property values of these elements.

3.6. Translation to Virtual Reality Modeling Language (VRML)

The module for creating a VRML [1] model is similar to the module for generating a finite element model, because a VRML model consists of nodes and polygonal elements. A C++ class derived from the ACIS class redefines virtual member functions according to the VRML format. In addition, the names of shipbuilding features are now used to set the colors of the VRML model shown in Fig. 6.

3.7. Translation into a STEP AP203 file

A STEP AP203 model on the general CAD system Pro/Engineer is shown in Fig. 7. It is a wireframe model extracted from the non-manifold model component of ACIS, which is stored as properties of the midship product model. The AP203 is one of the STEP application protocols. As an international standard, it is supported by major commercial CAD vendors.

![Fig. 6. A VRML model.](image)
3.8. CORBA objects

The CORBA objects consist of a JAVA applet that runs on a Web browser with an object implementation coded in the C++ language. The applet sends a DXF file of the AutoCAD that resulted from the initial structural design. The object implementation receives the DXF file and then executes an enhancement module. When the client function that saved the DXF file is invoked within the applet, the corresponding server function of the object implementation then saves the file in the memory of the enhancement module. To implement the CORBA objects, the CORBA tool Visibroker [10] is used.

3.9. Connection to the Oracle database

The data defined by the STEP methodology can be saved in the ROSE database. But the existing design data that has been prepared by a different method cannot be saved in it. In addition, the ROSE database does not support general data management such as Structured Query Language (SQL) transactions. It would be better if the STEP data were saved in a general database management system. In this work, the Oracle database is utilized together with its adaptor module known as ST-Oracle. This tool generates SQL files in creating tables within Oracle, and produces C++ and ProC codes, which load and unload STEP physical files to and from the Oracle database. The ProC code is comprised of C codes that include SQL statements. It is compiled by the Oracle pre-compiler. These programs can be used independently with an application developed by ST-Developer.

The following codes were generated by the ST-Oracle tool.

1. $(\text{Schema}\_\text{sql}$ and $(\text{Schema}\_\text{drop}\_\text{sql}$
   SQL files for creating and deleting tables in the ORACLE system according to the EXPRESS schema.
2. $(\text{str}\_\text{load}\_\text{h}$ and $(\text{str}\_\text{load}\_\text{c}$
   C++ codes to load a STEP physical file to ORACLE.
3. $(\text{str}\_\text{sql}\_\text{h}$ and $(\text{str}\_\text{sql}\_\text{pc}$
   ProC codes including SQL statements. It is required by the C++ code for item 2.
4. **rts_(Schema)_load.h** and **rts_(Schema)_load.c**
   C++ code to unload a STEP physical file from ORACLE.

5. **rts_(Schema)_sql.h** and **rts_(Schema)_sql.pc**
   ProC codes including SQL statements. It is required by C++ code for item 4.

4. **Network architecture for data sharing**

   To utilize the enhancement module for data exchange within the shipyard, a network architecture is required. In this paper, an architecture has been proposed, as shown in Fig. 8. It is based on the Internet and consists of Web browsers, Web server, CORBA object, data enhancement module, databases, and the following application systems used at each design stage.

4.1. **The scenario for data exchange**

   In order to show the exchange of CAD data throughout the network, our data-sharing system operates according to the following five steps.

   - **Step 1**: Deliver the 2D drawings which resulted from the initial midship design.
   - **Step 2**: Recognize 2D drawings and the data enhancement based on the EXPRESS schema.
   - **Step 3**: Construct the 3D non-manifold model using ACIS, and then store the product model.
   - **Step 4**: Translate the product model to the file format suitable for downstream application systems (AutoDef, Nastran, VRML).
   - **Step 5**: Download data to these downstream applications.

   Fig. 9 shows the Web page for uploading AutoCAD DXF files to the data enhancement module. This page includes the JAVA applet that is the client component of the CORBA application. Because the client is implemented in the JAVA language, it is executable throughout the Internet. Fig. 10 shows the Web page for downloading finite element data files that are enhanced from the midship product model.

4.2. **Connection of the Web browser and the enhancement module**

   The Web browser is connected to the server that handles HTML documents. The JAVA applet that
Fig. 9. Data uploading by the CORBA client of JAVA applet.

runs on a Web browser is downloaded from the Web server and is then connected with the data enhancement module independently of the Web server. This means the data enhancement module and the Web server do not have to run on the same machine. The enhancement module communicates with the JAVA applet according to the OMG Interface Definition Language (IDL) rather than the HTTP protocol. Consequently, the database can be separated from the Web server. Since a Web server may serve many browsers, the load on the Web server can be reduced.

The communication between the object implementation of the CORBA located in the enhancement module and the JAVA applet comprising the client part of the CORBA application is achieved by the following IDL specification.

```idl
module dxf {
  interface Ship_idManager {
    Ship_id open(in string name);
  };
  interface Ship_id {
    string get_result();
  };
}
```

Two interfaces, Ship_idManager and Ship_id, are defined. The Ship_idManager interface has a function, open, which uploads DXF files, while the
Ship_id interface has a function, get_result, which shows the result of uploading. This IDL file is compiled to C++ and JAVA code by the CORBA tool Visibroker. Both the C++ code for the object implementation and the JAVA code for the client are used by CORBA objects. The Ship_idManager and the Ship_id are generated into classes, while the open and the get_result are generated into their respective member functions. These classes are instantiated to the distributed CORBA objects that
bind the client and the object implementation. If the open function is invoked by the JAVA client, the open function of the C++ object implementation is then activated, while the DXF file of the client computer is saved in the object implementation.

4.3. Connection of the data enhancement module to the database

The Relational Database Management System (RDBMS) Oracle is used for storage and retrieval of product data. The Oracle supports SQL for the manipulation of the database and SQL*Plus for user interactions. However, it cannot support hierarchical data structures, and therefore restricts the length of identifier names (tables, columns, etc.) to 30 characters. But the EXPRESS language has no such limit, and many entity definitions in EXPRESS exceed this length. In addition, a number of RDBMS keywords do not match the keywords in EXPRESS. The following modifications [22] to the identifier names are needed to use Oracle:

1. If the name is a keyword of the RDBMS, change the name with the prefix, ‘‘R_’’.
2. If the name is greater than 30 characters, the last vowel or repeated character is removed until the name is less than 30 characters.
3. If the name still exceeds 30 characters, the name should be truncated.

A table is created for each of the entities defined in an EXPRESS schema, and the created tables are managed by an oid_mapping table. For example, the following entities are translated to SQL statements to create the corresponding tables.

**SQL > select * from oid_mapping**

<table>
<thead>
<tr>
<th>OID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x027279D968000003FFFE44E00000F8C0400000000</td>
</tr>
<tr>
<td>0x027279D968000003FFFE44E00000F8C0400000001</td>
</tr>
<tr>
<td>0x027279D968000003FFFE44E00000F8C0400000002</td>
</tr>
<tr>
<td>0x027279D968000003FFFE44E00000F8C0400000003</td>
</tr>
<tr>
<td>0x027279D968000003FFFE44E00000F8C0400000004</td>
</tr>
<tr>
<td>0x027279D968000003FFFE44E00000F8C0400000005</td>
</tr>
</tbody>
</table>

Each table has an additional field, (table_name)_id. This is an object identifier for specifying objects when they are attributes for other objects, and it is saved in the oid_mapping table. In addition, the RoseDesign, Name_Table, and Schema tables are created to address the situation where different data sets are stored by different schemas.

The following example shows the result of queries after the data is saved in the Oracle database. They illustrate how a circle entity that has a point entity as its attribute is saved. The circle with an object identifier (oid_key) of value 4 has the point of object identifier 0 as its attribute through the oid_mapping table. The point 0 is used as the center of the circle 4.

**ENTITY point;**

  * x: REAL;
  * y: REAL;

**ENTITY circle;**

  * center: point;
  * radius: REAL;

**CREATE TABLE point (**

  * point_id integer NOT NULL REFERENCES OID_Mapping(OID_KEY),
  * x float,
  * y float);**

**CREATE TABLE circle (**

  * circle_id integer NOT NULL REFERENCES OID_Mapping(OID_KEY),
  * center_id integer,
  * radius float);**

<table>
<thead>
<tr>
<th>OID_KEY</th>
<th>DESIGN_ID</th>
<th>ENTITY_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Point</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Point</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>Point</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>Line</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>Circle</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>Text</td>
</tr>
</tbody>
</table>
### 5. Conclusion

In this paper, a product model database has been implemented based on the STEP methodology, and a data enhancement module has been developed to exchange shipbuilding CAD data. The information modeling language EXPRESS is utilized to represent the product model data, while the geometry data which is represented differently for each CAD/CAE system has been further enhanced for downstream systems. Simple translation of CAD data according to the STEP standard is not enough in the cases when two sets of CAD data do not have a one-to-one relation. In this paper, a ship structural design model prepared in two dimensions has been enhanced into a 3D model based on the ACIS non-manifold modeler and feature recognitions, and then shared by the downstream design applications using the Web and CORBA objects. The tedious and repeated data input work because of data inconsistency between different CAD/CAE systems can be therefore reduced. It is estimated that modeling time for the structural analysis of a midship is reduced by 1 day from 7 days, and that for detail design shrinks by 7 days from 1 month.

CAD data is usually saved in its own database in the CAD system. Saving this data in a RDBMS such as the Oracle during the design process may lead to unsatisfactory performance, but it is needed for data backup and exchange. In addition, if a PDM system is used, an engineering database must be utilized to avoid data inconsistencies and unnecessary duplication of the product data within each CAD system. This paper presents an approach to use a DBMS including the product models according to the STEP methodology. The network architecture consisting of the Web and the CORBA is described through an example. Since the architecture supports a graphic user interface on a Web browser with distributed objects for internal communications, it can be used as a foundation for the realization of the CIM within industrial organizations.

### Acknowledgements

This work was partially supported by a contract with SOREC of Chungnam University.

### References

Yongjae Shin received a BSc in 1992 and a MSc in 1994 from the Department of Naval Architecture, Seoul National University, Korea. He is currently completing a PhD in the Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST). His research interests include the STEP, information flow in CAD, and the CIM.

Soon-Hung Han is an Associate Professor in the Department of Mechanical Engineering at the Korea Advanced Institute of Science and Technology (KAIST). He is leading the Intelligent CAD laboratory at the KAIST and the STEP Committee of Korea. His research interests include the STEP, geometric modeling kernels, and design expert systems. He has a BSc and a MSc from the Seoul National University, Korea, and a PhD from the University of Michigan.

Doo-Hwan Bae is an assistant professor in the Computer Science Department at the Korea Advanced institute of Science and Technology, Taejon, Korea. He received a BSc from Seoul National University, Korea. He received a BSc from Seoul National University, an MSc from the University of Wisconsin-Milwaukee, and a PhD from the University of Florida in computer and information sciences in 1992. His research interests include object-oriented software engineering, software process, and component technology.