

STEP표준과 Web을 이용한 RPD환경 구축

강석호*, 김민수*, 김영호**

Development of a STEP-compliant Web RPD Environment

Kang, S.-H.*, Kim, M.* and Kim, Y.**

ABSTRACT

In this paper, we present a Web-enabled product data sharing system for the support of RPD (Rapid Product Development) process by incorporating STEP (STandard for the Exchange of Product model data) with Web technology such as VRML (Virtual Reality Markup Language), SGML (Structured Generalized Markup Language) and Java. Extreme competition makes product life cycle short by incessantly deprecating current products with a brand-new one, and thus urges enterprises to devise a new product faster than ever. In this environment, an RPD process with effective product data sharing systems is essential to outstrip competitors by speeding up the development process. However, the diversity of product data schema and heterogeneous systems make it difficult to exchange the product data. We chose STEP as a neutral product data schema and Web as an independent exchange environment to overcome these problems. While implementing our system, we focused on the support of STEP AP 203 UoF (Units of Functionality) views to efficiently employ STEP data models that are maximally normalized, and therefore very cumbersome to handle. Our functionality-oriented UoF view approach can increase users' appreciation since it facilitates the modular usage of STEP data models. This can also enhance the accuracy of product data. We demonstrate that our view approach is applicable to the configuration control of mechanical assemblies.

Key words : RPD process, STEP, Web, SGML, VRML, UoF, View, Product Data

1. 서론

최근 지능형생산시스템(IMS: Intelligent Manufacturing System), 동시공학(CE: Concurrent Engineering), 가상제조(VM: Virtual Manufacturing), CALS (Commerce At Light Speed), 제품정보관리시스템(PDM: Product Data Management System), 기업자원계획시스템(ERP: Enterprise Resource Planning System)과 같은 개념 및 시스템들이 중요시되고 있는 것은 급변하는 경영 환경에서 제품개발, 생산 및 영업의 제반 기업활동을 통합적으로 관리하여 고품질, 저원가의 제품을 빠르게 개발하고자 하는 노력의 하나로 해석할 수 있다. 이것은 결국 폐속제품개발(RPD) 프

로세스의 구축을 요구하는 것이라 할 수 있다.

폐속제품개발 프로세스는 제품의 설계, 제작, 유지 보수 및 재활용에 이르는 전체 수명주기 동안 발생할 수 있는 제반 문제들을 각 단계의 전문가들로 구성되는 설계 팀이 제품개발의 초기 단계에서부터 체계적으로 고려함으로써 병렬적인 제품개발을 유도하여 개발기간을 단축하고 품질과 생산성을 향상시키는 동시공학적 접근법이다. 이러한 접근법의 성패는 구성원들간 원활한 의사소통과 제품정보의 효과적인 공유에 달려 있다. 그러나 기업 내에는 이미 다양한 컴퓨터 플랫폼과, 특정 자료형태에 의존적인 여러 응용 시스템들이 사용되고 있으므로 이들로부터 효과적인 정보공유 시스템을 구축해 내기가 매우 어렵다. 기업 활동영역의 확대로 나타난 다국적 기업에서처럼 부서들간의 지리적인 유리 또한 이질적인 시스템과 함께 정보공유에 많은 어려움을 주는 요인으로 작용

*서울대학교 산업공학과

**중신회원, 서울대학교 산업공학과

하고 있다. 따라서 정보공유가 효과적으로 이루어지기 위해서는 위와 같은 이질적이고 분산된 환경을 극복할 수 있도록, 교환되는 데이터의 스키마가 중립적이어야 하며 공유 환경 역시 운영 시스템에 독립적이어야 한다. 제품정보의 공유를 위한 다양한 시스템이 개발되었지만 위 두 가지 문제를 포괄적으로 해결하고 있지는 못하다.

본 연구는 이와 같은 이질성과 중속성의 두 문제를 종합적으로 접근하여, 폐쇄제품개발 환경에서 효과적인 제품정보의 공유가 이루어 질 수 있도록 하는데 그 목표가 있다. 이를 위해, 본 연구에서는 중립적인 데이터 스키마로써 제품정보의 교환 표준인 STEP을 사용하였으며, 독립적인 교환 환경으로써 현재 급격히 그 저변이 확대되고 있는 Web과 SGML, VRML 및 Java 등의 Web 기술을 활용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구가 따르고 있는 단위기능(UoF: Units of Functionality) 별 뷰 접근법에 대하여 설명하고, 3장에서는 본 연구의 배경이 되는 STEP, SGML과 웹 기술의 활용에 대한 기존 연구를 정리한다. 4장에서는 개발된 제품정보 공유시스템의 전체 구조와 단위기능별 뷰 응용 프로그램의 실행 구조를 제시하고, 대표적인 단위기능별 뷰 응용 프로그램으로 bill_of_material 단위기능을 살펴본다. 마지막으로 5장에서 결론과 추후 연구과제를 서술하였다.

2. 단위기능별 뷰 접근법

STEP에 대한 연구가 국내외에서 활발히 진행되고 있지만, 아직까지 STEP이 제공하는 방대한 데이터 모델을 충분히 활용하고 있지는 못하다. 그 원인의 하나로 STEP에서 정의된 제품정보 스키마가 응용 분야에 바로 적용하기에는 지나치게 정규화되어 있다는 점을 들 수 있다. STEP의 데이터 모델은 다양한 응용 분야에서 중복 없이 사용될 수 있도록 고도로 정규화되어 있다. 따라서 제품정보를 얻기 위해서 분할된 여러 개의 객체를 검색하고 조합해야 하는 상황이 일반적으로 발생한다. 이를 AP 203의 Authorization 단위기능을 통해 살펴보기로 한다. Authorization은 특정 날짜에 제품 데이터의 상태에 대해서 일정한 역할을 맡고 있는 사람들이 허용한 인증정보를 기술한다. 이를 위해 Approval과 Person_organization이라는 응용객체를 사용하는데, 전자는 인증한 날짜와 이유 및 제품 데이터의 상태를 기술하며, 후자는 인증을 허용한 특정 역할의 사람과 조직을 나타낸다.

```

.....
#7201=PERSON('8311455','Master','John',$.$.$.);
#7206=PERSONAL_ADDRESS($.$.$.$.,'SNU',$.,Korea,$.$.$.,$(#7201),I.
E.Dept.);
#7207=ORGANIZATION('DSSP dep. ');
#7210=ORGANIZATIONAL_ADDRESS($.$.$.$.,'SNU',$.,Korea,$.$.$.,$(#
207),'SNU Eng. ');
#7211=PERSON_AND_ORGANIZATION(#7201,#7207);
#7600=PERSON_AND_ORGANIZATION_ROLE('design owner');
#7601=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#72
11,#7600,#7300);
#7700=SECURITY_CLASSIFICATION_LEVEL('confidential');
#7702=SECURITY_CLASSIFICATION('formation_security', 'p_d_f for v1'.
#7700);
#7706=COORDINATED_UNIVERSAL_TIME_OFFSET(5, 0, BEHIND);
#7800=APPROVAL_STATUS('approved');
#7802=APPROVAL(#7800, 'approval for release');
#7804=CC_DESIGN_APPROVAL(#7802,#7300);
#7812=APPROVAL_ROLE('approver');
#7813=APPROVAL_PERSON_ORGANIZATION(#7211,#7802,#7812);
#7815=LOCAL_TIME(08,50,0.#7706);
#7816=CALENDAR_DATE(1999,15,12);
#7817=DATE_AND_TIME(#7816,#7815);
#7818=DATE_TIME_ROLE('sign_off_date');
#7819=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#7817,#7818,#78
13);
#7820=APPROVAL_DATE_TIME(#7817,#7802);
.....

```

Fig. 1. Example STEP file describing Authorization.

이 두 응용객체는 응용해석모델(AIM: Application Interpreted Model)에서 여러 개의 Entity로 정규화되어 정의되어 있는데, 이를 실제 STEP 물리파일 상에서 살펴보면 다음의 Fig. 1과 같다.

#7804번의 CC_DESIGN_APPROVAL Entity로부터 현재의 내용이 #7300번 제품에 대한 #7802번 인증을 설명하고 있음을 알 수 있다. #7802번 APPROVAL은 제품 공개(for release)를 위한 내용으로써 #7800번을 통해 공개가 승인된 상태를 알 수 있다. 또한 #7813번을 통해 인증이 #7812번의 APPROVAL_ROLE을 지닌 #7211번의 사람과 조직에 의해서 이루어졌음을 알 수 있다. 인증이 얻어진 날짜와 시간 정보는 #7819번과 #7820번을 통해 얻을 수 있다. 이처럼 Authorization 단위기능을 위해서 약 20여 개의 객체가 정의되어 사용되고 있다. 따라서 전체적인 의미를 파악하기 위해서는 각각의 세분화된 Entity를 참조 관계를 이용하여 하나하나 찾아서 조합해야만 한다.

이처럼 정규화는 중복을 제거하여 정보의 일관성과 일치성을 유지한다는 장점이 있지만, 응용 프로그램의 설계 및 유자를 어렵게 만드는 측면이 있다. 사용자에게도 세밀히 정규화된 스키마를 이해하여 사용하는 것이 매우 복잡하고 까다로운 작업이 된다. Spooner와 Hardwick은 지나치게 정규화된 모델로 인해 많은 소프트웨어들이 필요 이상으로 복잡해지고,

이해하기 어려워져 오히려 효과적인 제품정보의 교환을 어렵게 만든다고 주장한다. 두 사람은 마치 데이터베이스의 뷰를 제공하는 것처럼 응용 프로그램이 필요한 정보만을 조합하여 구성된다면, 지나친 정규화로 야기되는 정보공유의 비효율성을 제거할 수 있다고 생각하였으며, 이를 위해서 STEP의 응용해석모델과 응용참조모델(ARM: Application Reference Model) 사이의 대응관계를 강조하는 뷰 접근법을 제안하였다¹¹⁾. 본 연구는 이러한 뷰 접근법을 단위기능별로 적용하여 ARM의 응용객체를 정보단위로써 활용하였다. 이것은 방대한 STEP의 데이터 스키마를 단위기능별로 모듈화하여 사용할 수 있게 하며, 사용자들이 복잡한 하부의 데이터 모델보다는 상위의 응용객체를 이용할 수 있게 함으로써 시스템에 대한 이해를 높이고, 제품정보의 변경 및 조작을 쉽게 한다는 장점이 있다.

본 연구에서 중심으로 다루고 있는 STEP Part 203번의 경우 모두 14개의 단위기능이 ARM에 정의

Table 1. AP 203의 비형상 단위기능과 응용객체

AP 203 단위기능	단위기능을 구성하는 응용객체
Authorization	Approval, Person_organization
Bill_of_material	Alternate_part, Substitute_part Component_assembly_position Engineering_assembly Engineering_make_from Engineering_next_higher_assembly Engineering_promissory_usage
Design_activity_control	Change_order, Change_request Start_order, Start_request Work_order, Work_request
Design_information	Additional_design_information Design_specification Material_specification Process_specification Specification, Usage_constraint Surface_finish_specification
Effectivity	Planned_date_effectivity Planned_effectivity Planned_lot_effectivity Planned_sequence_effectivity
End_item_identification	Product_configuration, Product_model
Part_identification	Design_discipline_product_definition Part, Part_version
Source_control	Supplier, Supplier_part_version

되어 있는데, 8개의 단위기능이 제품의 구성관리를 위해 필요한 비형상 정보에 관한 것으로 제품의 사양 및 프로세스 정보를 기술한다. 다음 Table 1에 본 연구를 통해 구현된 비형상 정보의 단위기능이 응용객체와 함께 정리되어 있다.

3. 연구 배경 및 관련 연구

본 연구의 배경이 되는 기존 연구를 크게 1) 제품정보의 교환을 위한 STEP 활용, 2) 문서정보의 교환을 위한 SGML 활용, 3) 웹과 객체지향 데이터베이스의 활용으로 구분하여 살펴보기로 한다.

3.1 제품정보의 교환을 위한 STEP 활용

STEP은 특정 시스템에 종속됨이 없이 제품수명주기 동안 발생하는 제품정보를 기술하기 위한 국제 표준이다¹²⁾. STEP은 제품수명주기 전체를 대상으로 정의되었지만 기존의 형상 교환 표준을 능가하는 효율성 또한 가지고 있다¹³⁾.

STEP을 이용하여 제품정보를 교환 및 공유하려는 많은 시도들이 있었다. 초기의 연구는 주로 STEP의 형상 정보를 중심으로 진행됐는데, VRML을 이용해 3차원 부품형상을 웹 상에서 가시화한 World Design View 시스템을 대표적인 초기 연구라 할 수 있겠다¹⁴⁾. 또 다른 주목할만한 연구로는 STEPShare 시스템을 들 수 있다¹⁵⁾. STEPShare 시스템은 분산객체의 공유 표준인 CORBA를 사용함으로써 서버측의 형상 정보를 사용자가 객체단위로 접근할 수 있게 하였다. 이 외에도 STEP의 3차원 정보에 대한 의견을 웹을 통해 교환하고자 하는 가상회의시스템에 대한 연구도 있었다¹⁶⁾.

STEP과 PDM 시스템의 연계에 대한 연구 또한 많은 관심을 받고 있는 분야이다. 이 분야의 대표적인 연구로는 STEP-PDM 스키마의 제정에 대한 연구가 있다¹⁷⁾. PDM 시스템의 기반 데이터 스키마를 STEP에 정의되어 있는 제품 데이터 스키마를 활용하여 구축함으로써 PDM 시스템간 정보교환의 문제를 해결하려는 것으로, 현재 1.1 버전까지 정의되어 있으며, 아직까지 PDM 시스템의 기능을 전체적으로 제공하는 단계에까지는 이르지 못하고 있는 상태이다.

3.2 문서정보의 교환을 위한 SGML 활용

SGML은 전자적인 형태의 문서들을 장치와 시스템에 독립적으로 표현하고 작동하는 것에 대한 국제

표준이다¹¹⁾. 문서의 논리적 구조를 DTD(Document Type Definition)에 정형화하여 정의하고, 정의된 DTD에 맞게 기술하고자 하는 내용에 태그를 달아 SGML 문서를 작성하게 된다.

STEP과 SGML 환경의 연계에 대한 대표적인 연구로는, ISO/TC184/SC4/WG3의 T14 그룹에서 진행한 SWEDCALs 프로젝트가 있다¹²⁾. 여기서는 STEP을 이용한 제품정보의 생성과 SGML을 이용한 제품 문서화 과정을 결합할 수 있도록, 제품정보의 생성 단계에서 문서화 작업에 포함될 내용들을 정보객체(Information Object)로 미리 정의한 뒤, DTD에 따라 요구되는 SGML 문서를 생성하는 접근법을 취하고 있다. STEP과 SGML을 연계하려는 국내의 연구로는 STEP으로 기술된 제품구조정보를 SGML 문서로 변환하는 방법에 대한 연구가 있다. 이 연구는 SWEDCALs에서의 접근법과 유사하게, next_assembly_usage_occurrence라는 제품의 조립구조를 기술하는 EXPRESS Entity를 SGML 문서의 Table 형태로 대응시켜 변환하였다¹³⁾. 이 연구는 STEP 정보의 공유에 대한 연구가 형상 위주에서 비형상 정보를 다루는 방향으로 확대되고 있음을 보인다라는 점에서 그 의미를 찾을 수 있다.

3.3 웹과 객체지향 데이터베이스의 활용

웹은 경제적이고 대중적인 정보전달 환경으로써, 다양한 형태의 정보를 세계 어디에서라도 쉽게 접근할 수 있게 한다. 웹은 HTML(Hyper-Text Markup Language)로 기술된 문서 정보에 바탕을 두고 있지만, CGI나 Java와 같은 다양한 기술을 통하여 웹 서버와 사용자간의 상호작용도 가능하다. 웹은 다양한 형태의 표준과 기술을 낳게 했는데, 이들 중 본 연구와 밀접한 관련이 있는 것으로는 VRML, CGI 및 Java 언어다. VRML은 웹에서 3차원 형상 정보를 표현하기 위해 설계된 텍스트 중심의 언어로서, 웹 상에서 브라우저를 통해 3차원 이미지를 볼 수 있게 한다. VRML을 제품 형상의 가시화 뿐만 아니라 부품 정보의 공개 및 개략적 조립성 평가 등에도 적용한 연구가 있다¹⁴⁾. Java 프로그래밍 언어는 사용자의 클라이언트로 전달되어 웹 브라우저 내부에서 실행되는 Applet이라는 특별한 프로그램을 제공하는데, 비교적 높은 수준의 보안을 제공하면서도 CGI 프로그램보다도 더 유연한 사용자 상호작용을 지원한다.

다양한 연구가 웹을 활용하여 이루어지고 있는데, 웹을 이용한 제품설계정보공유에 대한 스탠포드 대학의 SHARE 프로젝트¹⁵⁾, 초고속망을 활용한 동시

작업시스템의 구현 및 가시화에 대한 Visinet 프로젝트¹⁶⁾, 3차원 모델의 협력적 생성 및 편집을 위한 mWorld 프로젝트¹⁷⁾와 3차원 공간에서 다수 사용자의 협력적 가시화작업을 위한 CSpray 시스템¹⁸⁾ 등의 연구를 대표적으로 뽑을 수 있다.

본 연구에서는 STEP의 제품정보를 객체지향 데이터베이스에 저장한다. 제품정보는 일반적으로 복잡한 구조를 가지는데, 관계형 데이터베이스는 복잡한 구조의 데이터를 저장하는 데 적합하지 못하다고 알려져 있다. 이미 CAD, CAM, CSCW(Computer Supported Collaborative Work), 멀티미디어 등 복잡한 구조의 정보를 저장하는데 객체지향 데이터베이스를 사용하는 것이 일반화되고 있다¹⁹⁾. 그러나 많은 기존 연구들이 STEP 정보를 관계형 시스템에 저장하여 이용하고 있다. 이것은 STEP이 채택하고 있는 객체지향 데이터 모델과도 배치된다고 할 수 있다. 특히 STEP의 AIM처럼 정규화된 정보를 인출하기 위해서 여러 개의 Entity를 참조 연산을 통해 찾을 경우, 관계형 데이터베이스의 사용은 심각한 성능상의 문제를 가져올 수도 있다. 이러한 문제는 제품의 구조 및 조립 정보를 구성하기 위하여 상하부품 간의 참조연산을 반복적으로 수행해야 할 경우 더욱 심각해진다.

4. 시스템의 구조 및 실행 모델

본 연구를 통해 개발된 제품정보 공유시스템은 STEP의 AP 203에 정의된 데이터 스키마를 기반으로 구축되었으며, 서버에 위치하고 있는 객체지향 데이터베이스인 Objectivity/DB에 저장된 STEP 제품정보를 웹을 통해 실시간에 접근할 수 있도록 설계되었다.

4.1 전체 구조 및 운영 모습

본 연구를 통해 구축된 시스템의 전체 구조가 다음 Fig. 2에 나타나 있다.

시스템은 웹을 중심으로 클라이언트와 서버로 나뉘며, 서버에는 STEP 정보를 저장하고 있는 객체지향 데이터베이스와 CGI 프로그램 및 비형상 정보의 SGML DTD가 위치하고 있다. 제품정보를 기술하는 STEP AP 203 파일은 구문분석기의 2단계에 걸친 처리를 통해 데이터베이스에 저장된다. 1단계에서는 각각의 AIM Entity별로 데이터베이스 객체가 생성된다. 2단계에서는 1단계에서 생성된 객체들간의 상호연관관계를 정의하는데, 모든 연관관계를 양방향

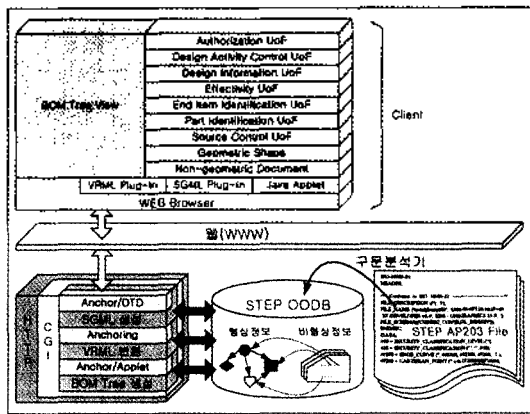


Fig. 2. System architecture and operational environment.

으로 설정한다. 본래 AIM 스키마는 객체들 간의 참조관계를 일방향성 위주로 정의하지만 실제 응용 시스템에서 이러한 일방향적 참조관계만으로는 객체의 빠른 인출이 어렵다. 따라서 성능을 향상시키고 객체의 검색이 쉽도록 양방향으로 설정한다. 이렇게 구문분석기를 통해 저장된 제품정보를 사용자의 요구에 의해서 실행되는 CGI 프로그램들이 검색할 수 있게 된다. 클라이언트는 사용자의 웹 브라우저가 되며, 각각의 단위기능 뷰 응용 프로그램들이 Applet으로 작성되어 클라이언트에서 실행된다.

4.2 형상 및 비형상 정보의 처리 구조

특정 제품의 형상 정보를 사용자가 요구하면, 이를 처리하는 CGI 프로그램이 해당 제품의 형상 정보를 조립구조에 따라 검색하여 각각의 내용을 VRML 파일로 변환하게 된다. 이 과정에서 VRML 형상 노드에는 현재의 형상에 대응되는 비형상 정보에 대한 링크 정보를 함께 삽입한다. 이것은 향후 형상 정보를 사용자가 검색하다가 특정의 부품을 선택하여 해당 부품의 형상 혹은 비형상 정보를 요구할 경우에 이를 처리하기 위한 것이다. 다음 Fig. 3은 VRML의 anchor 노드를 사용하여 웹 상의 다른 자원을 지정하고 있는 예이다.

현재의 VRML 형상 노드를 사용자가 선택할 경우

```
Anchor {
  description "To Part #07"
  parameter "target= _SGMLframe"
  url "http://www.snu.ac.kr/~g7/cgi-bin/generate_sgml.cgi?pid=07"
  children [ ..... ]
}
```

Fig. 3. Link to SGML using VRML anchor.

부품번호 07번인 제품의 비형상 SGML 문서를 생성하여 _SGMLframe이라는 프레임에 출력한다는 사실을 알 수 있다. 이렇게 VRML anchor 노드의 parameter와 url을 사용하여 비형상 SGML 문서로의 연동을 쉽게 제공할 수 있다.

비형상 문서 정보의 처리 역시 CGI를 통해 검색 결과를 SGML 문서의 형태로 변환하여 사용자에게 전달한다. SGML 문서는 STEP정보 중에서 단위기능별로 정리되어 있지 않은 기타의 비형상 정보를 종합하여 작성되는데, 여기에는 응용 프로토콜, 설계 및 제품 개념에 대한 컨텍스트 정보와 제품 카테고리 정보 등이 포함된다. 특히 대응되는 형상 정보의 링크가 SGML 문서의 생성시에 함께 삽입되도록 설계되어 있으므로, 문서가 기술하고 있는 부품의 형상을 쉽게 찾아볼 수 있다. 다음의 Fig. 4는 본 연구를 통해서 정의된 비형상 문서 정보 DTD의 일부이다. SGML 문서 내에서 다른 형상 정보로의 연동을 위해서 url을 정의하고 있음을 볼 수 있다.

실제로 위의 DTD를 기준으로 하여 작성된 SGML 문서의 일부가 Fig. 5에 나타나 있다. 문서의 링크 부분을 사용자가 선택할 경우 대응되는 SC-100이라

```
<NOTATION sgm PUBLIC "-//ISO 8879:1986/NOTATION Standard
Generalized Markup Language/EN" >
<ENTITY % TblBody "(TblHead | TblBody)">
<ENTITY % sqtbl PUBLIC "-//SoftQuad Inc/DTD CANONTBL/EN"
%sqtbl;
<ENTITY % inclusions "comment | figure | emp">
<ELEMENT CONFIGURATION -- (titleGroup, body) +(%inclusions;)
.....
<ELEMENT all_configon -- (url)>
<ELEMENT uri -- (CDATA)
<ELEMENT figure -- EMPTY>
<ATTLIST figure Imagename ENTITY #REQUIRED>
<?TAGLINK uri UR#>
```

Fig. 4. SGML DTD for textual information.

```
<!DOCTYPE CONFIGURATION PUBLIC "-//MS lab/DTD TSHOUJ/EN"
[<ENTITY amp CDATA "&">]
<CONFIGURATION>
<version_approval><title>version approval</title>
.....
</version_approval></part_version>
<part_definition><title>Part Definition</title>
<discipline_id>discipline id: <emp>design</emp></discipline_id>
<cad_file><comment>cad file</comment>
<filename>filename: <emp>no information</emp></filename>
<shape_url><comment> shape information:</comment>
<url>http://metopia.snu.ac.kr/~g7/team/cgi-
bin/markupview.cgi?cName=robot&pid=SC-100&pver=2.0</url>
</shape_url> </cad_file>
.....
<creator><comment>creator</comment>
<person> : <emp>7861123. Shade, Lim</emp></person>
<organization> : <emp>SC dep.</emp></organization></creator>
```

Fig. 5. Link from SGML document to shape information.

는 제품의 2.0 버전의 형상 정보가 인출되도록 설정되어 있음을 살펴볼 수 있다.

사용자가 특정 제품의 정보를 요구할 때, 시스템은 해당되는 제품의 형상 및 비형상 정보를 CGI를 통해 함께 처리하여 제공하는데, 형상 정보는 VRML 파일로, 제품의 개념 및 카테고리와 같은 부가 정보는 비형상 정보로서 SGML 파일로 생성된다. 생성된 파일들은 사용자의 웹 브라우저에 설치되어 있는 plug-in 프로그램을 통해 가시화 되는데, 이때 다른 형상 및 비형상 정보로의 참조 링크가 앞서 언급했던 형태로 삽입되어 제공된다. SGML 문서의 경우에는 가시화를 위해 DTD가 문서와 함께 사용자에게 전달된다.

4.3 단위기능별 뷰의 실행 구조

본 연구를 통해 구현된 단위기능은 Authorization, bill_of_material, design_activity_control, design_information, effectivity, end_item_identification, part_identification 및 source_control로 모두 8개이다. 다음의 Table 2에 구현된 단위기능의 내용과 기능이 정리되어 있다.

이들 단위기능은 각기 하나 씩의 CGI 프로그램과

Table 2. 구현된 단위기능의 내용과 기능

단위기능	내용 및 기능
Authorization	인증정보와 관련된 제품 데이터, 날짜, 인증자의 역할 등의 정보 제공
Bill_of_material	제품구조정보 제공, 형상 및 비형상과 다른 단위기능으로의 연동 기능 제공
Design_activity_control	초기 부품 설계의 요구사항 및 개정된 부품 버전의 요구사항 정보, 설계 이력(history)을 기술하는 정보 제공
Design_information	부품의 설계와 제조에 관련된 사양과 제약조건 등 완성 부품의 구체적인 특성값을 기술하는 정보 제공
Effectivity	제품 모델에서 부품들의 날짜별, 로트별 및 일련번호별 사용 및 투입 계획에 대한 정보 제공
End_item_identification	판매되는 최종 제품의 구성 및 모델에 대한 정보 및 구입과 제작 여부에 대한 정보 제공
Part_identification	부품 개념 및 특정 버전에 대한 수평 주기 단계 및 생성 날짜 정보 제공
Source_control	부품의 공급 및 하청업체에 대한 정보와 certification에 대한 정보 제공

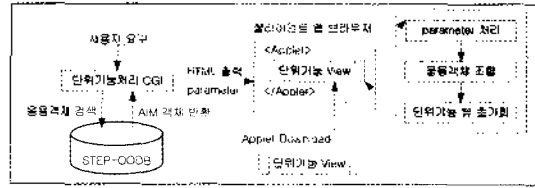


Fig. 6. Implementation of view application systems.

Java Applet을 통해 뷰로서 제공되는데 8개의 단위기능 모두 동일한 실행 구조를 가진다. 다음의 Fig. 6은 이러한 뷰 응용 시스템의 실행 구조를 나타내고 있다.

사용자의 정보검색요구에 의해서 실행된 단위기능 처리 CGI는 먼저 STEP-OOB 내에서 해당되는 단위기능에 정의된 응용객체를 검색하게 된다. 이때 각 응용객체는 실제로는 데이터베이스 내에서 AIM Entity로 분해되어 저장되어 있기 때문에, 결국 전체 AIM 객체들이 반환된다. CGI는 이들 AIM 객체들을 응용객체별로 구조화한 다음, 결과 HTML 파일을 생성할 때, 뷰 처리 Applet의 parameter 형식으로 사용자의 웹 브라우저에 출력하게 된다. 생성된 HTML 파일이 클라이언트의 웹 브라우저에 전달될 때, 여기에 포함되어 있는 단위기능 뷰 Applet도 함께 전달된다. 단위기능 뷰 Applet은 클라이언트 웹 브라우저에서 실행됨과 동시에 자신의 parameter로 전달된 AIM 객체들을 응용객체 형태로 조합한 다음, 사용자에게 뷰의 형태로 초기화하여 제공하게 된다.

다음의 Fig. 7은 본 연구에서 구현된 단위기능 간의 상호연동관계를 나타낸다. 전체 단위기능의 중심에 bill_of_material 단위기능이 위치하고 있음을 알 수 있다. 이것은 대부분의 제품정보가 직접 혹은 간접적으로 제품구조정보와 관련을 맺고 있기 때문이다. 따라서 본 시스템 내의 모든 단위기능별 뷰가 bill_of_material 단위기능을 통해서 접근이 가능하도록 구현되었다.

제품의 형상 정보와 비형상 정보가 VRML과 SGML의 anchor와 url을 통해서 상호 연동되고 있음

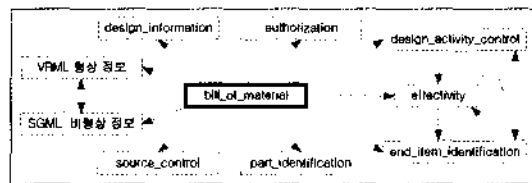


Fig. 7. Interactions among UoFs.

을 여기서도 확인할 수 있다. Effectivity 단위기능의 경우 재귀적인 참조를 하고 있는데, 이것은 현 부품을 포함하는 상위부품의 Effectivity로의 연동을 제공하고 있기 때문에 나타나는 것이다.

4.4 전체 시스템의 운용 예

제품 개발의 초기에 제품 개념과 모델의 컨텍스트 등의 정보를 설계자는 비형상 SGML 문서의 형태로 얻을 수 있게 되며, 여기에서 초기 제품 버전의 설계에 대한 요구사항 및 작업지시 정보를 Design_activity_control 단위기능의 Start_request 및 Start_order 등의 응용객체를 통해 얻을 수 있다. 설계자는 설계 과정에서 부품 및 제품의 사양 및 특성치를 Design_information, End_item_identification 및 Part_identification 등의 단위기능을 통해서 제공하게 되며, 다른 설계자가 제공하는 정보 역시 함께 참조할 수 있게 된다. 이런 과정을 통해 초기 제품의 버전이 완성되며, 완성품의 BOM 정보가 Bill_of_material 단위기능을 통해 제공된다. 제품 정보의 변경과 관

련된 정보가 Authorization 단위기능을 통해 인증되며, 완성품의 제조를 위한 유효성 관리는 Effectivity 단위기능에 의해서, 부품 공급과 관련된 정보는 Source_control 단위기능을 통해 제공된다. 초기 버전에 대한 변경 요구 및 작업지시가 다시 Design_activity_control 단위기능의 Change_request와 Change_order 응용객체를 통해 발생하면 설계자는 새 버전의 제품 설계를 비슷한 과정을 통해 시작하게 된다. 이처럼 반복적인 설계 과정을 시스템에서 제공하는 단위기능별 뷰 응용 프로그램을 통해서 추적 및 관리할 수 있으며, 웹을 통해 원거리에서도 쉽게 접근할 수 있다. 이와 같이 단위기능별 뷰 응용 시스템은 제품의 초기 설계 작업에서부터 그 진행 과정 동안의 정보를 체계적으로 공유 및 관리할 수 있게 함으로써 캐속제품개발 프로세스를 지원하게 된다.

다음 Fig. 8은 시스템의 전체적인 운용 모습을 구현된 단위기능별 뷰와 함께 모두 정리한 것이다. 전체 8개의 단위기능별 뷰와 VRML 형상 정보 및 SGML 비형상 정보가 모두 나타나 있으며, 중앙의

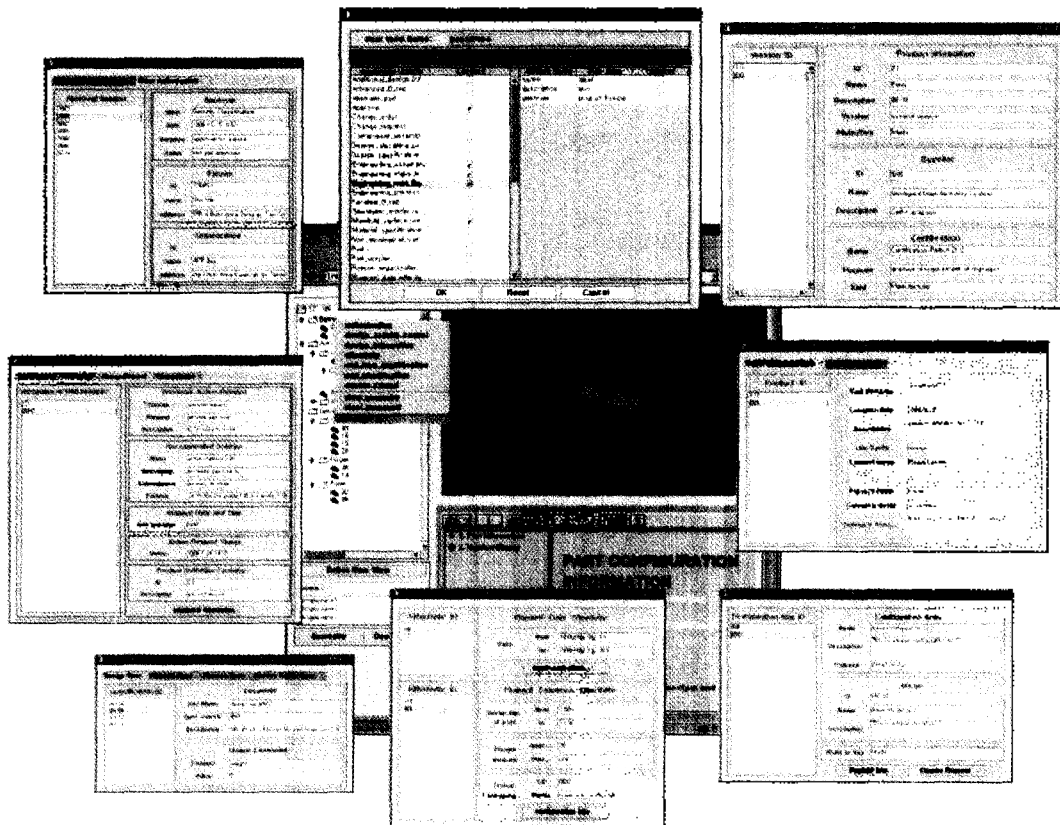


Fig. 8. User interfaces using UoF's provided by the system developed.

좌측에 구현된 bill_of_material 단위기능으로부터 나머지 단위기능으로의 연동을 선택할 수 있음을 볼 수 있다.

4.5 Bill_of_material 단위기능

bill_of_material 단위기능은 부품을 구성하기 위해 필요한 재료나 구성 요소들의 구조화된 목록을 표현한다. 여기에는 Alternate_part, Component_assembly_position, Engineering_assembly, Engineering_make_from, Engineering_next_higher_assembly, Engineering_promissory_usage, Substitute_part로 모두 6개의 응용객체가 정의되어 있다. Alternate_part 응용객체는 형태, 치수 및 기능 면에서 다른 부품과 상호 교환될 수 있는 부품을 나타내는 것으로 규격 나사와 같이 다양한 회사에서 각기 다른 부품 번호를 가지고 생산되지만 규격에 맞게 정확히 상호 대체될 수 있는 부품들을 가리키게 된다. Component_assembly_position 응용객체는 조립품의 기하학적 공간 속에서 구성요소의 위치와 방향을 지정하는데 사용된다. Engineering_assembly 응용객체는 조립품과 부품 혹은 하위부품과의 부자관계를 나타낸다. 따라서 각각의 Engineering_assembly 응용객체는 실제로 Engineering_next_higher_assembly 응용객체나 Engineering_promissory_usage 중의 하나로 나타나게 된다. Engineering_make_from 응용객체는 특정의 부품이 다른 부품을 기본으로 하여 설계되었을 경우가 두부품간의 관계를 나타내기 위해 정의되었다. Engineering_next_higher_assembly 응용객체는 앞서 언급했듯이 Engineering_assembly의 한 형태으로써 조립 계층도상에서 바로 상위 부품과의 관계를 기술한다. Engineering_promissory_usage 응용객체 역시 Engineering_assembly의 한 형태이지만 Engineering_next_higher_assembly와는 달리 조립 계층도상에서 구체적으로 그 위치를 결정짓지 않은 상태의 부품을 전체 구조 속에서 상위 부품과 관계 맺는데 사용된다. Substitute_part 응용객체는 Alternate_part 응용객체와는 달리 형태, 치수 및 기능이 정확히 들어맞지는 않지만 조립품 속에서 다른 부품의 요구 조건을 충족시킬 수 있어서 대체가 가능한 부품을 가리킨다. 이처럼 부품간에는 여러 가지 관계가 나타날 수 있는데, 실제의 STEP 물리파일 상에서 조립품은 NEXT_ASSEMBLY_USAGE_OCCURRENCE 라는 Entity를 통해 두 부품의 상하 조립관계를 나타내게 된다. 다음 Fig. 9는 이러한 제품구조를 나타내는 STEP물리파일의 일부이다.

```

.....
#9650=NEXT_ASSEMBLY_USAGE_OCCURRENCE('ASSY101','0001','SC-
21',#6331,#6320,'2.0');
#9651=NEXT_ASSEMBLY_USAGE_OCCURRENCE('ASSY102','0001','SC-
27',#6331,#6326,'2.0');
#9652=NEXT_ASSEMBLY_USAGE_OCCURRENCE('ASSY103','0001','SC-
31',#6331,#6330,'2.0');
#9653=NEXT_ASSEMBLY_USAGE_OCCURRENCE('ASSY104','0002','21-
08',#6320,#6307,'2.0');
#9654=NEXT_ASSEMBLY_USAGE_OCCURRENCE('ASSY105','0002','21-
18',#6320,#6317,'2.0');
.....
    
```

Fig. 9. AIM entity describing BOM structure.

#9650번의 NEXT_ASSEMBLY_USAGE_OCCURRENCE Entity는 #6331번의 부모 부품과 #6320번 하위 부품을 연결하고 있음을 알 수 있다. 전체 Entity를 하나하나 살펴보면 부모 부품으로는 나타나지만 하위 부품으로는 한번도 나타나지 않고 있는 제품이 최종품이 되며, 위의 경우에는 #6331번이 최종 제품이 된다. 제품구조를 나타내는 이러한 Entity들이 객체지향 데이터베이스 내부에 양방향의 연결 관계를 통해 저장되어 있기 때문에, bill_of_material 단위기능 처리 CGI는 데이터베이스를 1차로 검색하여 최종품을 찾은 다음, 나머지 하위 부품을 breadth-first 방식을 통해 검색하여 전체 제품구조를 출력하게 된다. 다음 Fig. 10은 제품구조를 기술하는 데 사용되는 EXPRESS Entity를 표현한 EXPRESS-G 다이어그램이다.

next_assembly_usage_occurrence가 product_definition_relationship으로부터 상속된 relating_product_definition과 related_product_definition을 사용하여 두 product_definition 간을 부자관계로 연결하고 있음을 알 수 있다. 대체품간의 관계는 좌측 상단의

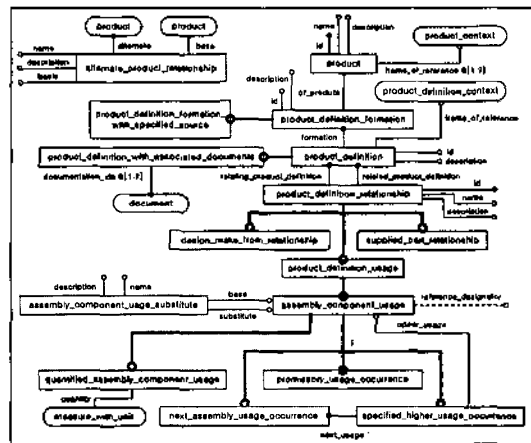


Fig. 10. EXPRESS-G diagram for product structure^[17].

alternate_product_relationship을 통해 두개의 product를 연결함으로써 기술되고 있음을 볼 수 있다.

결국, bill_of_material 단위기능을 제대로 구현하기 위해서는 product_definition Entity와 이로부터 상속되어 정의된 모든 Entity를 찾아서 이들의 상하 관계를 연결해야 할 필요가 있다. 이 과정을 보통의 관계형 데이터베이스 상에서 수행할 경우 제품의 조립 단계만큼의 테이블 조인이 발생하는 등 상당한 계산량이 드는 작업이 된다. 그러나 본 연구에서 사용하는 객체지향 데이터베이스에서는 비교적 쉬운 객체참조연산을 통해서 전체의 제품구조를 제공할 수 있다는 장점이 있다.

본 연구를 통해 구현된 bill_of_material 단위기능의 운용 모습이 다음의 Fig. 11에 나와 있다. 시스템의 메인 화면에서 좌측에 트리 형식으로 나타나는 bill_of_material 단위기능은 다른 단위기능 뷰 응용 프로그램과는 달리 다른 모든 단위기능 뷰와의 연동이 집중되어 있으며, 제품의 전체 조립 단계를 모두 보여준다. BOM의 특성상 모든 부품들을 한꺼번에 살펴볼 수 있기 때문에 본 연구에서는 bill_of_material 단위기능에 시스템의 전체 기능을 통합시켰다. 트리상의 특정 부품이나 부분품을 선택함으로써 해당 부품의 형상과 관련 문서를 계층적으로 살펴볼 수 있다.

나머지 7개의 단위기능 뷰 응용 프로그램 역시 앞서의 Table 2에 정리된 기능을 제공하기 위하여 여러 AIM Entity들을 조합하여 응용객체별로 정보를 제

공하는데, bill_of_material 단위기능과 동일한 실행 구조를 갖도록 작성되었다.

5. 결론 및 추후 연구과제

본 연구를 통해 STEP AP203 물리 파일을 객체지향 데이터베이스에 저장하여, 웹을 통해 실시간에 공유할 수 있는 시스템을 개발하였다. 제품의 형상 정보와 비형상 정보 뿐만 아니라 단위기능으로 정의되어 있는 정보를 개별적인 응용 프로그램을 통해 뷰로써 구현하였다. 기존 연구의 단편적인 접근을 극복하고 STEP 데이터 스키마를 전체적으로 활용할 수 있는 시스템의 구축을 위해 VRML, SGML 및 웹 기술을 종합적으로 활용하였다. 본 연구를 통해 구현된 단위기능은 제품의 초기 설계 단계에서부터 제품정보의 생성 및 변환 과정을 추적할 수 있게 한다. 사용자는 bill_of_material 단위기능을 통해 다른 단위기능을 접근할 수 있으며, 제품 형상과 비형상 정보를 제품구조에 따라 계층적으로 검색할 수 있다. 더욱이 사용자를 복잡한 STEP AIM 스키마로부터 분리시켜 보다 상위의 응용객체별로 이해할 수 있게 하였다. 이와 같은 단위기능 위주의 뷰 접근법은 향후 더욱 방대해질 수밖에 없는 STEP 데이터 모델의 모듈화된 사용을 가능케 한다. 본 연구 결과는 또한 STEP-PDM 스키마의 개발과 같은 표준화 노력에 대한 가능성을 검증하는 한 예가 될 수 있다.

갈수록 양질의 제품을 빨리 개발해야만 하는 상황에서 기존 제품정보의 효율적 재사용은 훌륭한 대안이 될 수 있으며, 효과적인 제품정보공유시스템의 개발은 이런 노력의 중심에 있다. 본 연구 시스템의 중립적인 데이터 스키마와 독립적인 사용 환경은 산적해 있는 제품정보공유 관련 문제들을 해결해 나갈 수 있는 대안이 될 수 있다.

본 연구와 관련하여 추가적으로 연구되어야 할 사항들은 다음과 같다. 우선 제품 형상 정보를 단순히 검색하는 것뿐만 아니라, 직접 웹을 통해 동시적인 조작이 가능하도록 해야 한다. 이러한 공동작업이 이루어지기에는 VRML의 근사적 모델이 너무 제한적이다. 따라서 웹을 통해 동시적 설계가 가능한 3차원 형상 모델이 개발될 필요가 있다. 이 형상 모델에는 웹 상의 링크와 같은 기능이 함께 포함되어 설계되어야 할 것이다. SGML을 이용한 비형상 정보의 표현 역시 최근 활발히 이용되고 있는 XML에 비해 링크 기능이 부족하다고 할 수 있다. 보다 확장된 연동기능을 위해서 XML의 사용을 고려하는 것도 바

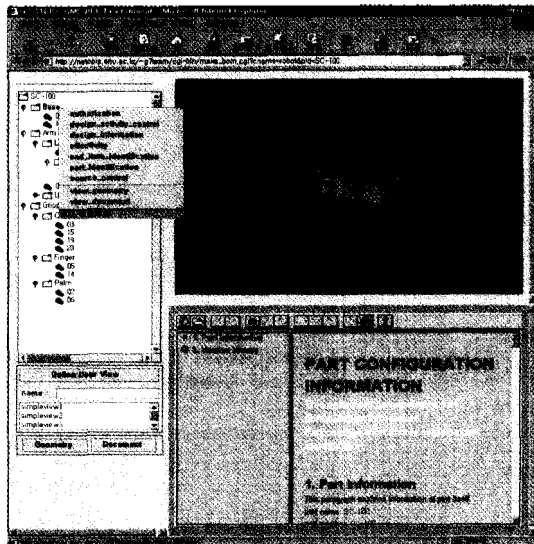


Fig. 11. Use of bill_of_material UoF.

람직하겠다. 본 연구가 주로 AP203의 단위기능을 활용하여 전개되었기 때문에 다른 응용 프로토콜 영역으로 확대하는 것 또한 의미있는 연구가 될 것이다.

후 기

본 연구는 산업자원부 국제공동연구의 결과이며, 서울대학교 공학연구소에서 수행한 위탁과제입니다.

참고문헌

1. Spooner, D.L. and Hardwick, M., "Using views for product data exchange," *IEEE Computer Graphics and Applications*, Vol. 17, No. 5, pp. 58-65, September-October, 1997.
2. Owen, J., *STEP An Introduction*. Information Geometers, Winchester UK, pp. 11-16, 1997
3. PDES Inc., Program Annual Report, http://pdesinc.ati-corp.org/annual_99/index.html, Sep. 1999.
4. 김철영, 김남국, 김영호, 강석호, "웹과 STEP을 이용한 제품설계정보 공유시스템," 한국 CAD/CAM 학회 논문집, 제1권, 제3호, pp. 203-214, 1996.
5. 최 영, 양상욱, "STEPShare: 웹 환경에서의 CORBA 기반 3D STEP 브라우저," 한국 CAD/ CAM 학회 학술발표회 논문집, pp. 17-23, 1998.
6. 김덕수, 문왕식, 이동규, 장태범, 한성배, 김현, "웹과 STEP기반 3차원 가상회의," 한국 CAD/ CAM 학회 학술발표회 논문집, pp. 11-16, 1998.
7. PDM Schema Version 1.1, <http://www.epmtech.jotne.com/pdm/startpage.html>, Jun. 1999.
8. 김숙자역. SGML의 모든 것, 성안당, 1997.
9. SWEDCALs, "Interoperability between STEP and SGML," <http://info.admin.kth.se/SGML/Bibliotek/Literatur/whitep/>, Swedish Association for CALS, 1995.
10. 김태환, 김선호, "STEP과 SGML의 통합 방법론-Part 44를 중심으로-" 한국 CAD/CAM 학회 학술발표회 논문집, pp. 331-335, 1998.
11. 최영상, 김영호, "웹을 이용한 실시간 3차원 공동작업 부품정보증개시스템 개발," 한국 CAD/CAM 학회 논문집, 제4권, 제2호, pp. 87-99, Jun. 1999.
12. Park, H., Cutkosky, M., Conru, A.B. and Lee, S.-H., "An Agent-Based Approach to Concurrent Cable Harness Design," *AI for Engineering Design, Analysis and Manufacturing*, Vol. 8, pp. 45-61, 1994.
13. Lamoite, W., Flerackers, E., Reeth, F.V., Earnshaw, R., De Matos, J.M., "Visinet: Collaborative 3D Visualization and VR over ATM Networks," *IEEE CG & A*, Vol. 17,

No. 2, pp. 66-73, Mar.-Apr. 1997.

14. Dias, J.M., Galli, R., Almeida, A.C., Belo, C.A., Rebordao, J.M., "mWorld: A Multiuser 3D Virtual Environment," *IEEE CG & A*, Vol. 17, No. 2, pp. 55-65, Mar.-Apr. 1997.
15. Pang, A. and Wittenbrink, C., "Collaborative 3D Visualization with CSpray," *IEEE CG & A*, Vol. 17, No. 2, pp. 32-41, Mar.-Apr. 1997.
16. Objectivity/DB version 4.0, *Using Objectivity/C++*, Objectivity Inc, 1996.
17. INTERNATIONAL STANDARD, ISO 10303-203, *Application Protocol: Configuration controlled design*, 1st ed. Dec. 1994.



강 석 호

1970년 서울대학교 물리학과 학사
1972년 미국 Univ. of Washington 산업공학과 공학석사
1976년 미국 Texas A & M Univ. 산업공학과 교수
1976년-현재 서울대학교 산업공학과 교수
관심분야: 생산정보시스템의 설계와 운용, 지능형 생산 시스템, 경영정보 시스템, 생산 계획 및 통제



김 민 수

1994년 서울대학교 산업공학과 학사
1996년 서울대학교 산업공학과 석사
1996년-현재 서울대학교 산업공학과 박사과정
관심분야: STEP, Collaboration, Network Security



김 영 호

1985년 서울대학교 산업공학과 학사
1987년 서울대학교 산업공학과 석사
1988년~1990년 한국과학기술연구원 연구원
1993년 North Carolina 주립대 산업공학과 공학박사
1995년-현재 서울대학교 산업공학과 교수
관심분야: PDM, 동시공학, Work flow, 인터넷 응용