

# OOIDEF0 기반의 정보시스템 분석 및 설계 기법: 조선 PDM 시스템 적용사례

김재균<sup>1</sup> · 장길상<sup>2\*</sup>

<sup>1</sup>울산대학교 산업경영정보공학부 / <sup>2</sup>울산대학교 경영학부

## An Information System Analysis and Design Methodology Based on Object-Oriented IDEF0: A Case Study for the PDM System of Ship Production

Jae-gyun Kim<sup>1</sup> · Gil-sang Jang<sup>2</sup>

<sup>1</sup>School of Industrial Engineering, University of Ulsan, Ulsan, 680-749

<sup>2</sup>Dept. of MIS, College of Business Administration, University of Ulsan, Ulsan, 680-749

Recently, object-oriented techniques have been used more and more for developments of an information system. But, established object-oriented methodologies are hard to express a business process of various abstract degrees in the analysis level and independent components of the system. They have difficulties in developing a large-scale information system of manufacturing industry such as PDM and CIM. This paper proposes an information system development methodology that imports the object-oriented IDEF0 (OOIDEF0) function model in analysis level. This methodology is made up of requirements gathering, system analysis, system design, and implementation. In requirements gathering level, organization diagram and interview technique are used for input data of OOIDEF0 function model. The OOIDEF0 function model, the interface model and function descriptions are made out in analysis level. Information objects and implementation objects are designed on the basis of the OOIDEF0 function model in design level. The database is built and programming is accomplished in implementation level. In order to prove the consistency and efficiency of the proposed methodology, the PDM system for ship production is modeled and prototyped.

**Keywords:** OOIDEF0, 객체지향, 정보시스템, 방법론, PDM, 조선산업, BOM

### 1. 서론

정보시스템 개발 방법론은 정보시스템을 개발하기 위한 작업 방법이나, 절차, 산출물, 기법 등을 논리적으로 정리해 놓은 체계를 말한다. 정보시스템 개발 방법론은 기존의 소프트웨어 공학과 기업에서의 실무적 차원을 결합하여 만들어진 소프트

웨어 관련 학문의 총 집결체라고 할 수 있다. 개발자들은 방법론을 이해하고 참조하면서 시스템의 계획, 분석, 설계, 구현, 운영이라고 하는 소프트웨어 개발 생명 주기(SDLC : Software Development Life Cycle)를 따라 정보시스템 개발을 수행하게 된다(김유석, 1998). 따라서 개발 방법론은 시스템 개발의 이론적 기반이라고 할 수 있다. 이러한 방법론은 하드웨어의 기술 발

\*연락처 : 장길상 교수, 680-749 울산시 남구 무거 2동 산 29번지 울산대학교 경영학부, Fax : 052-247-7619,

e-mail : gsjang@mail.ulsan.ac.kr

2002년 9월 접수, 1회 수정 후 2002년 11월 게재 확정.

전 속도가 소프트웨어의 기술 발전 속도보다 빨라지고, 새로운 소프트웨어를 요구하는 시장의 수요가 많아짐에 따라 기존 소프트웨어의 유지 보수가 힘들어지는 문제, 즉 인건비의 상승효과와 우수 소프트웨어의 부족 현상이라고 하는 이른바 소프트웨어 위기를 해결하기 위하여 등장하였다. 이로 인해 대규모 비즈니스 시스템(Business System)을 구축하기 위한 프로젝트의 체계적인 수행, 프로젝트의 효과적인 관리, 프로젝트 수행에서 얻어진 지식 축적, 시스템 개발의 생산성 향상 등과 같은 성과를 기대할 수 있게 되었다.

이러한 정보시스템 개발 방법론은 일반적으로 다음의 3가지 범주로 구분할 수 있다(Kim, 1995). 먼저 구조적 접근 방법론으로서, 이것은 단순하고 기능적 추상화 능력과 업무 및 정보의 흐름을 표현하는 능력이 우수하다는 장점이 있으나, 기능, 정보, 동적모델 간의 일관성이 부족하고 시스템 개발단계에서 산출되는 모델간의 불일치로 많은 부가적인 노력이 소모될 수 있다. 많이 사용되는 구조적 접근 방법론으로는 DFD(Data Flow Diagram) 기반의 구조적 기법, ERD(Entity Relationship Diagram) 기반의 정보공학, IDEF 방법론 등이 있다(Adam and Gangopadhy, 1993; Young and Vesterager, 1991; Malhortra and Jayaraman, 1992). 다음으로 객체지향 방법론으로서, 이것은 행위와 속성을 통합된 객체로 캡슐화하여 시스템의 기능과 정보 측면을 통합함으로써 의미가 풍부하며 소프트웨어 생산성이 높다는 장점이 있다. 그러나 많은 연구들에서 다양한 추상화 수준으로 업무를 표현하는 능력과 시스템의 독립된 구성요소로서 업무를 표현하는 능력이 부족하고, 적당한 객체를 정의하는 것과 도출된 모델의 직관적인 검증이 난이하며, 기능객체와 클래스 메소드간의 관계가 명확하지 않다는 약점을 지적하고 있다(황성룡, 1999; Hoydalsvik and Sindre, 1993; Opdahl and Sindre, 1994; Tagg and Liew, 1993). 대표적인 객체지향 방법론으로는 Rumbaugh의 OMT(Object Modeling Technique), Shlaer와 Mellor의 OOA(Object-Oriented Analysis), Jacobson의 OOSE(Object-Oriented Software Engineering), 그리고 UML(Unified Modeling Language) 등이 있다(Kim, 1995; Shlaer and Mellor, 1991; Sutcliffe, 1991; Quatrani, 1998). 마지막으로 혼합 접근 방법론은 구조적 방법의 장점과 객체지향 방법의 장점을 적용하는 대상에 따라서 적절하게 통합하는 방법론으로서, 구조적 방법의 모델간 비일관성과 객체지향 방법의 기능적 추상화 및 객체 정의 문제를 해소하는 수단이 될 수 있다(황성룡, 1999; Kim, 1995).

특히, 관리해야 할 업무가 대규모이고, 복잡하다면 적절한 모형화 방법을 이용하여 보다 쉽게 이해할 수 있도록 구조화하여 단순하게 표현할 필요가 있다. 이러한 모형화 방법에는 여러 가지가 있으나, 대표적으로 IDEF0 모형화 방법을 들 수 있다. IDEF0 모델은 활용 용이성과 업무흐름 및 업무와 정보의 연관관계에 대한 표현능력의 우수성으로 인하여 CALS(Commerce At Light Speed) 표준으로 자리잡고 있으며, 제조업체의 통합시스템 구축시 시스템 분석용으로 널리 사용되고 있는

기능모델이다. 특히, 객체지향 패러다임 하에서 구조적 IDEF0 모델을 객체지향 패러다임으로 확장시킨 OOIDEF0(Object Oriented IDEF0) 모델은 설계단계에서 객체지향 모델을 사용하고자 할 때, 분석단계에서 사용하기에 적합한 기능모델이다(황성룡, 1999; Balin, 1989; Alabiso, 1988; Seidewitz and Stark, 1987).

본 연구에서는 이러한 OOIDEF0 기능모델을 기반으로 기능 모델, 정보모델, 그리고 사용자 인터페이스 모델을 효과적으로 통합하는 정보시스템 개발 방법론을 제시하고자 한다. 본 방법론에서는 OOIDEF0 기능모델의 기능객체와 ICOM 객체로부터 정보객체의 구조와 속성을 정의하고, 기능기술서의 기능 처리 과정을 통해 메소드를 도출하며, 정의된 정보객체와 인터페이스 모델을 기반으로 구현객체를 정의한다. 이러한 일련의 과정을 통하여 기능객체와 정보객체, 그리고 구현객체를 효과적으로 통합함으로써 분석, 설계 그리고 구현 단계간의 일관성을 유지시킬 수 있다. 또한 분석단계에서 사용자 인터페이스 설계를 수행함으로써 사용자의 요구사항을 신속하고 용이하게 수립할 수 있으며, 각 개발단계의 모델간의 통합을 용이하게 한다. 본 연구에서 제시된 방법론의 일관성과 효율성을 검증하기 위하여 조선산업의 PDM 시스템을 대상으로 모형화 과정을 설명하고 프로토타입 시스템을 구현하였다.

## 2. OOIDEF0 기반의 정보시스템 개발 방법론

본 연구에서 제시하는 방법론은 기존의 객체지향 방법론의 분석 측면을 보완하기 위한 수단으로 OOIDEF0 모형을 사용하는 구조적 방법과 객체지향 방법의 장점을 취한 혼합 방법론이다. 본 방법론의 구조(architecture)는 <그림 1>에 표현되어 있으며, 명확하고 단순한 표현을 위하여 단계간의 피드백(feedback)은 표현하지 않았다. 본 방법론은 크게 요구사항 수집단계, 분석단계, 설계단계, 구현단계의 4단계로 구분된다.

### 2.1 요구사항 수집 (Requirement Gathering)

OOIDEF0 모형화의 입력정보를 위한 자료수집의 단계로 모형화 대상의 조직도와 인터뷰 기법을 이용한다. 조직도는 기능객체와 ICOM 객체에 대한 권한과 책임을 부여하는데 사용한다. 요구사항 수집단계에서의 산출물들은 OOIDEF0 기능 모형화의 입력자료가 되며, 조직도와 인터뷰 기법뿐만 아니라 기존에 작성되어 사용되고 있는 정보흐름도(DFD: data flow diagram), 개체관계도(ERD : entity relation diagram) 등의 기법들도 활용된다.

### 2.2 분석(Analysis)

대부분의 객체지향 방법론은 분석단계에서 Use Case 모델이

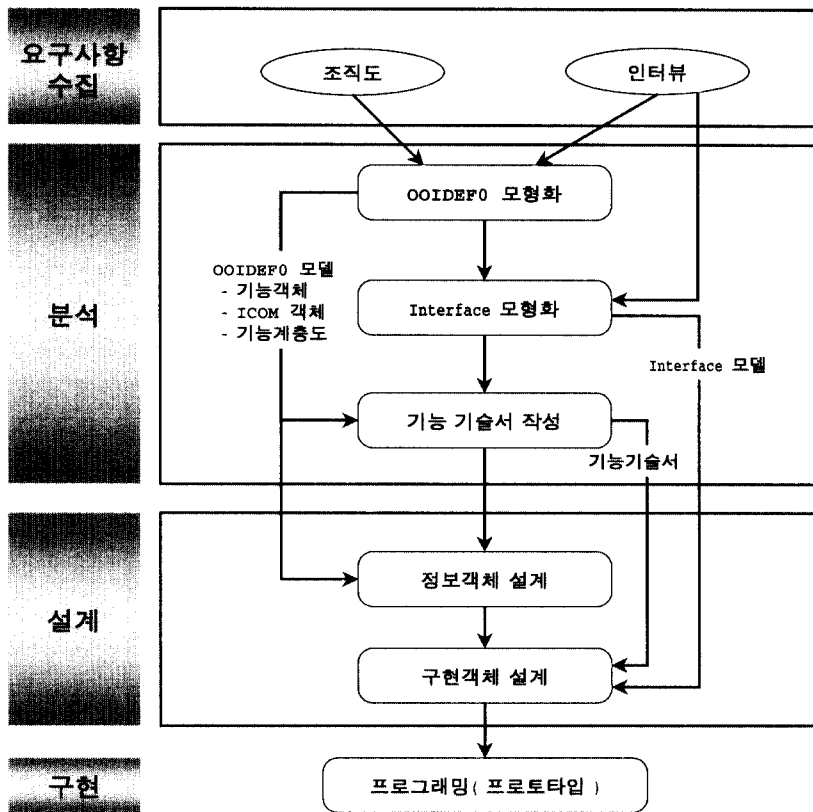


그림 1. OOIDEF0 기반 객체지향 정보시스템 개발 아키텍처.

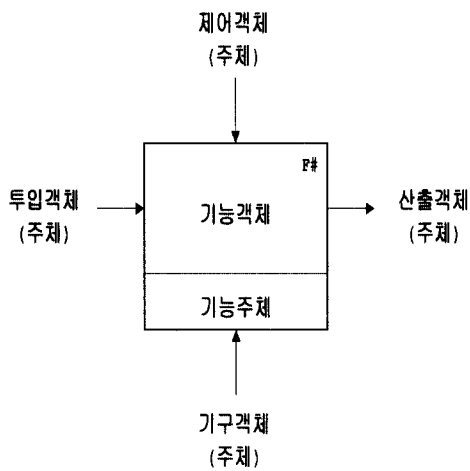


그림 2. OOIDEF0 기능모델의 기본 구조.

나 서술적 문장을 사용해 왔지만, 본 연구에서는 구조적 IDEF0 모델을 객체지향 패러다임으로 확장시킨 OOIDEF0 기능모델과 인터페이스 모델, 그리고 기능기술서를 사용한다.

OOIDEF0 기능모델은 기능객체와 ICOM 객체로 구성된다. 기능객체는 업무의 수행을 의미하며, 기능상자라 불리는 사각형으로 표현한다. 기능상자에서 오른쪽 모서리의 F# 표시는 기능 분할시 기능객체의 계층적 수준(level)을 나타내는 기능번호를 의미한다. 그리고 ICOM 객체는 투입(input), 제어(control),

산출(output), 기구(mechanism) 객체로 영어의 두음을 따서 ICOM이라 명명하며 화살표로 표현한다. 예를 들면 어떤 회사의 판매부의 업무를 OOIDEF0 기능모델 사용한다고 했을 때, 제품판매 활동은 기능객체가 되고, 고객주문 및 재고 정보는 투입객체(I), 업무지침서는 제어객체(C), 직원 및 데이터베이스는 기구객체(M), 그리고, 제품발송 정보는 산출객체(O)가 된다. OOIDEF0 기능모델의 기본적인 구조는 <그림 2>와 같이 표현된다. OOIDEF0 기능모델은 업무 담당자(business people) 또는 시스템 사용자(system user)와의 인터뷰를 통하여 도출된다. 업무 담당자는 기능의 수행순서와 기능간의 ICOM 흐름의 관점에서 기업의 실제 업무를 표현한다. 여기에서 ICOM은 기존의 구조적 IDEF0 모델에서와 같이 데이터 또는 데이터 그룹이 아니라 도면, 부품, 문서와 같은 객체이다.

시스템 설계자는 업무 담당자와의 인터뷰 과정에서 대상 영역의 업무를 모형화한다. 이러한 과정에서 기능은 기능객체가 되고, ICOM은 ICOM 객체가 된다. OOIDEF0 기능 모형화는 조직도를 이용하여 최상위 기능객체로부터 시작하여 하향식 기능분할(top-down function decomposition) 방법으로 보다 세부적인 하위수준의 기능으로 분할한다. 분할 과정은 업무의 요구된 상세수준까지 계속된다. 이러한 방법으로 모형화가 완료되면, 기능객체를 계층적으로 표현한 기능계층도(function hierarchy diagram)를 작성한다. 이는 전체적인 기능객체간의 관계를 계층적으로 표현한 것이다. OOIDEF0 기능모델은 사용자

와의 용이한 의사소통으로 빠르고 정확한 모델의 생성과 생성된 모델의 효과적인 검증을 지원하므로 기능모델과 실질적인 업무 사이의 차이(gap)를 줄이는 역할을 한다. 일반적인 객체지향 방법론이 객체모델을 도출하는데 어려움이 있는 반면, OOIDEF0 기능 모형화는 업무와 연관된 적당한 ICOM 객체를 정의하는 방법이 체계적이어서 정보객체 모형화를 용이하게 한다. 또한 조직모델과의 통합을 위하여 기능주체와 객체주체를 OOIDEF0 기능모델 내에 표현한다. 기능주체와 객체주체는 조직간의 업무관계를 표현하는데 유용하며, 기능객체에 대한 권한과 책임을 명시하는데 사용된다(황성룡, 1999).

인터페이스 모형화는 인터뷰의 내용을 기반으로 구현화면을 설계하는 과정이다. 설계된 화면으로부터 사용자의 요구사항을 구체화하며 인터페이스의 구현에 있어서 추가적인 사항을 도출해낸다. 인터페이스 모델에서 각 아이콘은 고유의 기능을 설명할 수 있어야 하며 화면에 대한 기술서 등을 이용하여 화면의 구성을 명확히 하는 것이 필요하다. 인터페이스 모델의 각 아이콘의 기능은 기능기술서를 통하여 정의되고, 정의되지 못한 부분에 대해서는 구현객체 설계과정에서 재정의된다. 본 방법론에서는 기존의 구현단계에서 이루어지던 인터페이스 모형화를 분석단계에서 수행함으로써 구현객체 설계를 보다 용이하게 지원하고 사용자의 요구사항을 보다 신속하고 용이하게 수렴할 수 있도록 한다.

기능기술서는 OOIDEF0 기능모델을 기반으로 작성된다. 기능기술서는 OOIDEF0 기능객체의 기능처리 과정(process)을 순서적으로 기술하며 기본구조는 <그림 3>과 같다. 기능기술서의 구성요소는 OOIDEF0 기능모델에서 정의된 기능객체를 기반으로 투입객체, 투입객체 속성, 제어객체, 제어객체 속성, 산출객체, 산출객체 속성 등을 명시하고 해당 기능의 처리 과정을 서술형으로 기술한다. 또한 기능객체가 포함되는 인터페이스 번호를 명시한다. 기능처리 과정은 객체 단위로 구분한다. 즉, 객체가 투입되어 산출되는 과정에서 발생하는 모든 데이터의 전이를 하나의 기능처리 과정으로 한다. 기능기술서는 정보객체와 구현객체의 메소드를 정의하는 기반이 되며, 기능처리 과정이 복잡하게 되거나 모호할 경우에는 UML의 시퀀스 다이어그램(sequence diagram)이나 콜라보레이션 다이어그램(collaboration diagram) 등을 이용하여 기능처리 과정을 상세화하여 메소드의 정의를 용이하게 한다.

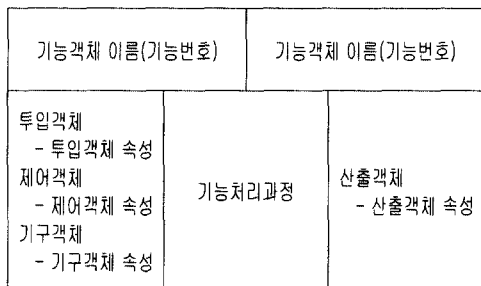


그림 3. 기능기술서의 기본 구조.

2.3 설계 (Design)

시스템 설계 단계에서는 OOIDEF0 기능모델과 기능기술서, 그리고 인터페이스 모델을 이용하여 정보객체와 구현객체를 설계하는 과정이다. 정보 및 구현 객체 설계의 도식적 표현은 UML의 Class Diagram을 이용한다. <그림 4>는 정보 및 구현 객체 모델의 기본구조를 표현한 것으로 각각의 상속(Generalization), 연관(Association), 복합(Aggregation) 관계는 하나의 기능과 로컬 정보모델을 표현하고 있다. 여기에서의 구조와 관계는 ICOM 객체 간의 일반적인 구조와 관계를 의미하는 것이 아니고, 단지 구조와 관계의 표기법을 설명하기 위한 것이다.

정보객체 설계는 OOIDEF0 기능모델에 정의되어 있는 ICOM 객체의 구조와 관계를 정의하여 OOIDEF0 기능모델의 기능내부에 로컬 정보객체를 설계한 다음, 기능계층도에 따라 상향식으로 글로벌 정보객체의 설계가 이루어지는 상향식 총괄방식(bottom up aggregation)이 사용된다. 상향식 총괄과정에서 중복객체가 제거되고 부가적인 구조와 관계가 정의된다. 이 과정에서 정보의 구조가 나타나게 되며, 객체 내의 속성이 정의된다. <그림 5>는 OOIDEF0 기능모델로부터 정보객체를 모형화하는 과정을 보여준다. 정보객체 내의 메소드 도출은 정보구조와 객체 내의 속성이 정의된 상태에서 기능기술서를 이용하여 기능처리 주체를 정의하고 작업처리 과정을 메소드로 전환하는 방법을 사용한다. 이 과정에서 부가적인 객체가 정의된다. 메소드는 일반적으로 객체 내에서 속성의 값을 가공하여 객체 내의 또 다른 속성으로 취급되는 경우와 객체를 생성, 삭제, 변경하는 경우, 서로 다른 두 객체간의 메시지 교환 등으로 구분할 수 있다. 기능기술서의 기능처리 과정은 객체 간의 메시지 교환 및 데이터의 변환을 기술하고 있으므로 하나의 기능처리 과정은 하나 혹은 그 이상의 메소드로 취급될 수 있다.

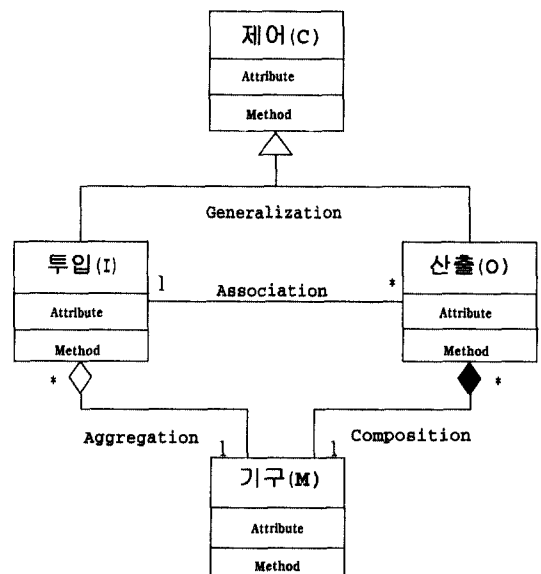


그림 4. 정보/구현객체 모델의 기본구조.

그러나 기능처리 과정이 복잡한 경우에는 다수의 객체간에 메시지 교환이 이루어지므로 투입객체와 산출객체간의 기능 처리만으로는 메소드를 도출해낼 수 없다. 이 경우에는 기능 처리 과정을 상세하게 표현하는 것이 필요한데, 본 방법론에서는 UML의 시퀀스 다이어그램(sequence diagram)을 사용하여 기능처리과정을 상세화한다.

구현객체의 설계는 정보객체와 인터페이스 모델을 기반으로 구현객체의 속성을 정의하고 해당 인터페이스 모델내의 주요기능을 기능기술서로부터 추출하여 구현객체 메소드를 도출한다. 구현객체를 정의하는 목적은 분석, 설계, 구현단계간의 일관성을 유지하고 보다 효율적인 프로그래밍을 지원하는데 있다. 이러한 구현객체는 구현단계의 프로그래밍에 직접 적용되므로 정보객체의 운용과 인터페이스 모델의 세부적인 기능을 표현하는 속성과 메소드로 구성된다. 특히 인터페이스 모델이 포함하는 각 아이콘의 기능들은 기능기술서의 기능처리 과정에 표현되지 않는 경우도 있으므로 정보객체 설계에서 누락되었거나 추가되어야 하는 객체 혹은 메소드의 경우는 구현객체 설계에 포함한다. 정보객체와 구현객체의 설계 과정을 요약하면 <그림6>과 같다.

2.4 구현 (Implementation)

구현단계에서는 시스템 설계의 산출된 모델을 통하여 시스템 개발자가 선택된 구현환경 내에서 물리적인 정보시스템을 개발하는 단계이다. <그림 7>과 같이 정보객체와 구현객체 설계의 결과로 각 객체에 대응되는 테이블이 생성되며, 객체의 속성과 메소드를 정의하는 수준의 자바 코드를 생성한다. 개발자는 이러한 결과물을 기반으로 구현객체와 정보객체의 메소드를 중심으로 구체적인 프로그래밍을 수행하게 된다.

UML의 클래스 다이어그램을 이용하여 소스 코드 및 데이터베이스 스크립트를 도출해내는 방법은 많은 연구가 있어 왔다 (Lee, 1997; Booch, 1998). 본 연구에서는 Rational사의 UML 모델

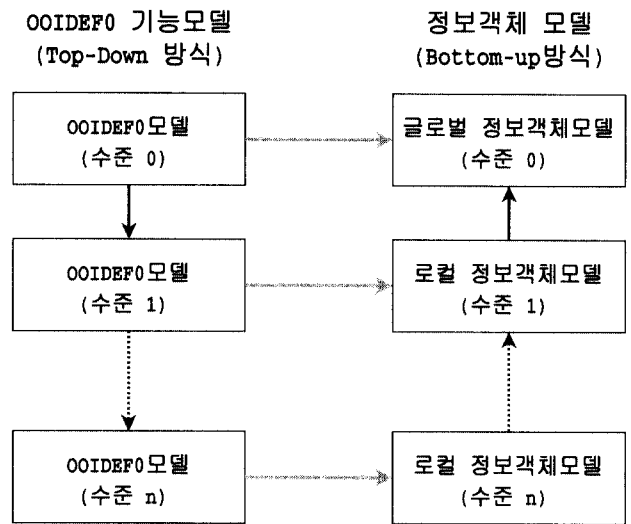


그림 5. 정보객체 모형화 과정.

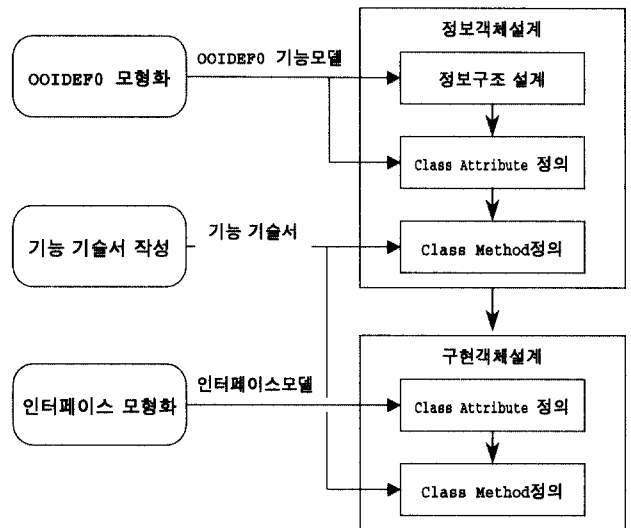


그림 6. 정보/구현객체 설계 과정.

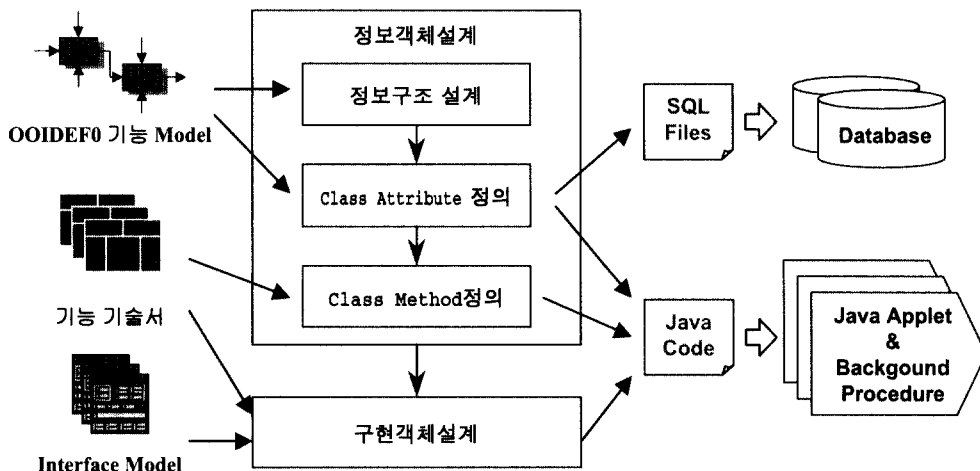


그림 7. 본 방법론의 구현 과정.

링 도구인 Rose를 이용하여 구현에 필요한 코드와 데이터베이스 스크립트를 생성하였다.

2.5 다 방법론과의 비교

본 연구에서 제시하는 방법론과 기존에 제시된 개발 방법론 중 본 연구와 관련 있는 대표되는 비교대상 방법론을 선정하여 평가항목별로 그들간의 특징을 비교한다. 본 연구에서는 모델간의 일관성, 기업의 실제 업무를 표현하는 능력인 기능적 추상성, 재사용성 · 확장성 · 유지보수성을 의미하는 소프트웨어 생산성, 의미 풍부성 (semantic richness), 객체 속성 정의 용이성, 메소드 정의 용이성, 사용자 인지성 (readability)과 활용 용이성, 기능객체 정의 여부, 그리고 구현객체 설계 포함 여부를 평가항목으로 사용한다. 기존 방법론과의 비교는 <표 1>에 정리되어 있으며, 이는 구조적방법과 객체지향방법의 특징, 그리고 각 비교대상 방법론이 보유하고 있는 단점을 기반으로 하고 있다. 객체지향 접근방법과 구조적 접근방법을 포함한

일반적인 정보시스템 개발 방법론은 기능모델, 정보모델, 동적모델만을 모형화하는 것이 보편적이고, 일부 방법론을 제외하고는 사용자 인터페이스 모델을 방법론에 포함하지 않는 경향이 있다. 사용자 인터페이스 모델을 포함한 방법론의 경우에도 정보 모형화를 수행한 후에 사용자 인터페이스 모형화를 수행하는 것이 보통이다. 김현곤(1996) 등은 사용자 인터페이스 모델을 기반으로 객체모델을 도출하는 방법론을 제시하면서, 이 방법론이 전문가에게나 경험이 부족한 시스템 분석가에게도 적용하기 쉬운 방법론이라고 정의하고 있다.

본 방법론은 설계단계에서 구현을 고려한 구현객체 설계를 추가하였다. 구현객체 설계는 인터페이스 모델의 기능을 정의하고 정보객체를 운용하는 역할을 함으로써 정보시스템의 분석, 설계, 구현단계의 일관성을 유지시킬 수 있고, 사용자의 다양한 요구 지원 및 효율적인 시스템을 구축을 가능하게 하여 시스템의 유연성과 확장성, 보안성을 향상시킬 수 있다. 또한, 본 방법론은 클래스 메소드를 보다 용이하게 도출해내는 과정을 제시하고 있으며, 이것은 기존의 객체지향 또는 혼합형 방

표 1. 기존 정보시스템의 개발 방법론과의 비교

평가항목	IDEF (I-CAM)	OOA (Shlaer/Mellor)	OMT (Rumbaugh)	OOSE (Jacobson)	UML	OOIDEF0기반 통합모형화 방법론(황성룡)	본 연구의 방법론
접근방법	구조적방법	객체지향방법	객체지향방법	객체지향방법	객체지향방법	혼합방법	혼합방법
사용모델	- IDEF0 - IDEF1 - IDEF2	- 확장 개체 관계도 - 상태전의도 - 데이터흐름도 - 객체관계모델	- 객체모델 - 상태전의도 - 데이터흐름도	- Use Case 모델 - 인터페이스 모델 - 객체모델 - 상호연관도 - 상태전의도, etc	- Use Case 모델 - 클래스 모델 - 객체모델 - 상호작용 모델 - 구현모델, etc	- OOIDEF0기능 모델 - OOIDEF0정보 모델 - 객체사양서 - 노드 모델 - 인터페이스 모델, etc	- OOIDEF0 기능모델 - 인터페이스 모델 - 기능기술서 - 정보객체모델 - 기능객체모델, etc
소프트웨어 생산성	낮음	보통	높음	높음	높음	높음	높음
기능적 추상성	높음	낮음	낮음	비교적 낮음	비교적 낮음	높음	높음
모델간의 일관성	낮음	보통	비교적 낮음	비교적 높음	비교적 높음	비교적 높음	비교적 높음
의미 풍부성	낮음	비교적 높음	높음	높음	높음	비교적 높음	비교적 높음
객체속성 정의 용이성		어려움	어려움	보통	보통	비교적 쉬움	비교적 쉬움
메소드 정의 용이성		어려움	어려움	보통	보통	비교적 어려움	보통
인지성/용이성	높음	비교적 낮음	비교적 낮음	보통	보통	비교적 낮음	비교적 낮음
동적모델	고려함	고려함	고려함	고려함	고려함	고려하지 않음	고려하지 않음
기능객체		고려하지 않음	고려하지 않음	고려함	고려함	고려함	고려함
구현객체 설계	지원하지 않음	지원하지 않음	지원하지 않음	지원함	지원함	지원하지 않음	지원함
사용자 인터페이스 모델	지원하지 않음	지원하지 않음	지원하지 않음	지원함	지원함	지원함	지원함

법론과 비교할 때 시스템 설계 기간을 단축시킬 수 있고, 보다 체계적인 설계를 가능하게 하는 효과를 가져온다.

### 3. 조선 PDM 시스템 적용사례

#### 3.1 조선 생산과 조선 PDM 시스템

조선산업은 다양한 설계 및 제조 능력과 장기간의 건조기간(lead time)을 요구하는 수주 생산형 프로젝트 사업으로, 선종/선형이 다양하여 정보와 업무의 표준화가 어려운 노동집약적 산업이다. 조선산업의 또 다른 특징은 선체와 의장 등의 다양한 설계부문과 생산, 자재 등의 생산부문 그리고 많은 하청업체와 공급업체들이 연관된 매우 복잡하고 반복적인 공정을 포함하고 있으며, 설계가 완벽하지 않은 상태에서 자재조달, 계획 및 생산의 동시다발적인 수행으로 빈번한 설계변경이 발생한다는 것이다. 위와 같은 조선산업의 특징과 복잡하고 방대한 양의 데이터로 인하여 효율적인 정보시스템 개발이 어려운 실정이다(황성룡, 1999).

조선 생산은 수주단계로부터 제품설계, 자재조달, 생산, 시운전 및 인도의 제단계를 거치며, 생산공정은 크게 선각작업(hull construction), 의장작업(outfitting), 그리고 도장작업(painting)으로 구분된다. 선각작업은 선체의 외형과 골격을 만드는 작업으로 가공, 조립, 탑재공정이 이에 속한다. 의장작업은 추진, 운항, 거주환경 등 중요기관과 이를 연결하는 기능을 만드는 작업으로, Block의 조립과정에서 행해지는 선행의장, Dock에서

행해지는 의장작업, 그리고 안벽에서 행해지는 의장작업으로 구분된다. 도장작업은 부식을 방지하기 위한 작업으로 의장작업과 유사하게 선행도장, Dock 도장, 그리고 안벽에서 행해지는 도장작업으로 구분된다(박명환 외, 1995; 황성룡, 1999).

PDM 시스템은 제품개발 전체 프로세스 측면에서의 제품데이터를 기존의 직렬적, 반복적인 프로세스에서 동시적이고 병렬적이며, 협동적인 프로세스로 정립되고 적용되어야 한다는 동시공학(concurrent engineering) 개념의 구체적인 실현을 위한 접근 도구로 나오게 되었고, 제품 및 기술(engineering/technical)에 관련된 문서(document)와 이들 문서로 표현된 정보를 전 제품의 라이프 사이클에 걸쳐 관리하고자 하는 시스템이다(최열현, 1995; 이충화, 1996).

조선산업에서의 PDM 시스템은 수주산업의 특성에 맞게 설계정보를 생산정보로 변환하여 설계로부터 생산에 이르는 정보의 흐름을 단일하게 하는데 그 목적이 있다. 특히, CAD (computer-aided design) 정보를 BOM 정보, 품목 정보 등으로 변환할 수 있어야 하며, 산업 전반에 걸친 제품 및 기술에 대한 정보와 그 생명주기(life cycle)를 관리하고, 변경관리를 수행할 수 있어야 한다.

#### 3.2 조선 PDM 시스템 모형화 및 프로토타입

본 절에서는 제시된 방법론을 조선 PDM 시스템을 통해 모형화하는 과정을 보여준다. 연구 대상 모델인 조선 PDM 시스템의 구조를 간략히 표현하면 <그림 8>과 같다. 이는 조선 생산 시스템에 관련된 제반 활동들을 체계적으로 통합하는 조선

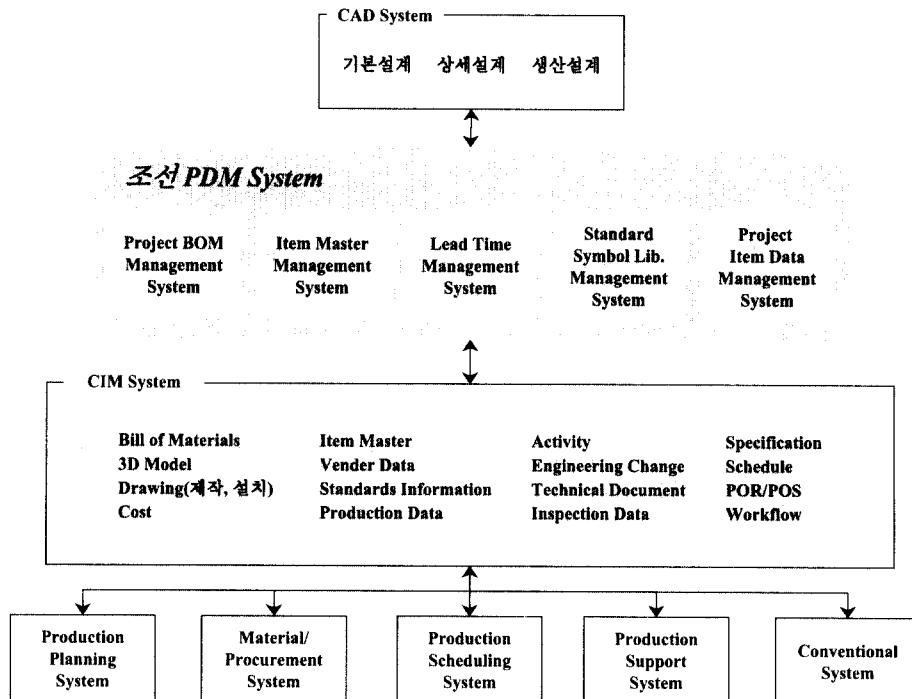


그림 8. 조선 PDM 시스템의 구조도.

CIM 시스템의 부분 시스템으로서 형상, BOM, 품목, 공정, 일정, 예산, 도면, 자재 등의 정보를 포함하는 제품모델을 통하여 CAD, 제품정보관리, 생산계획, 자재/조달, 일정계획, 생산지원 시스템과 일반 시스템인 견적, 구매, 원가, 인사급여 시스템의 업무를 효율적으로 지원하기 위한 시스템의 구조를 표현한다.

조선 PDM 시스템의 중요한 특징 중 한가지는 설계 데이터를 CIM 시스템에서 관리하는 데이터 포맷으로 변환하고 관리하는 데 있다. 따라서 PDM 시스템은 CAD와 CIM의 인터페이스의 역할을 하며, 생산 데이터가 생성되는 최우선의 위치에 있다.

프로토타입 시스템의 개발 환경은 PTC(Parametric Technology Corporation)사의 PDM 개발 도구인 Windchill을 사용하여 웹서버를 구축하고, Rational Rose와 Oracle Database 간의 인터페이스를 지원하였다. Rational사의 객체지향 모델링 도구인 Rose를 사용하여 설계단계에서의 산출물인 클래스 다이어그램과 기타 다이어그램을 작성하여 Java Source로 전환하였고, Database는 Oracle Database 8.0.4를 사용하였다. Java Applet 개발 도구로는 Symantec사의 Visual Cafe 3.0을 사용하였다.

3.2.1 요구사항 수집

요구사항 수집단계에서는 조직도와 인터뷰 기법을 이용하여 OOIDEF0 기능모형화를 수행하기 위한 입력자료를 수집한다. <그림 9>는 모형화 대상의 조직도를 간략히 표현한 것으로, 이는 OOIDEF0 기능모형화 단계의 입력 정보이며, 기능객체와 ICOM객체에 대한 권한과 책임을 부여하는데 사용한다.

3.2.2 분석

분석단계에서는 조직도와 인터뷰 기법을 이용하여 기능간의 관계 및 기능과 ICOM 객체간의 관계를 정의하기 위하여 OOIDEF0 기능모형화를 수행한 후, OOIDEF0 기능모델을 기반으로 기능기술서를 작성하고 인터페이스 모형화를 수행한다.

OOIDEF0 기능모델은 정보객체 설계의 기반모델이 되며, 설계자와 최종 사용자와의 원활한 의사소통을 지원하므로 실질적인 업무를 반영한 정확한 모델을 생성하게 한다. 조선

PDM 시스템을 전체 시스템의 관점에서 나타내면 <그림 10>과 같다.

PDM 시스템은 조선 CIM 시스템 내의 주요 시스템 중의 하나로서 표준 CAD 모델을 이용하여 3차원 CAD 시스템상에서 제품을 설계하는 Product Design(F1)과 설계와 생산의 연계 역할을 하면서 BOM, 품목, 예산, 공정정보 등의 제품에 관련된 제반 정보를 관리하는 역할을 한다. 또한, 수주, 생산계획, 자재조달, 일정계획, 생산 등의 업무를 지원하는 Order To Delivery(F3) 기능객체와 ICOM 객체를 통하여 연관관계를 가진다.

본 논문의 구현 대상이 되는 Product Data Management(F2)을 세부적으로 나타내면 <그림 11>과 같으며, 이는 하위기능으로 BOM/품목관리(F21), 도면관리(F22), 계량/부재관리(F23), 예산관리(F24), 공정정보관리(F25)를 포함한다.

<그림 12>는 BOM/품목관리(F21) 기능객체 내의 하위 기능들을 표현하고 있다. BOM/품목관리(F21)는 호선기본정보관리(F211), Block Division관리(F222), 호선BOM관리(F213), 그리고 호선품목관리(F214)의 하위 기능을 포함한다.

인터페이스 모형화는 <그림 13>에서 보여지는 것과 같이 구현화면을 기반으로 모형화된다. 인터페이스 모델은 기능기술서를 작성하는 기반이 되며, 시스템 설계단계에서 구현객체 설계의 기반이 된다. 화면 상단의 표준선 구분, 호선번호, 검색 구분, 검색내용 선택 아이콘은 리스트 박스를 나타내고 있으며, 화면 좌측의 창은 BOM의 구조를 선과 상자로서 나타낼 수 있도록 하였다. 또한 BOM 구조 창에서 노드를 선택하면 해당 품목의 정보를 우측 패널에 표현하도록 하였다. 화면 하단의 버튼은 다른 메뉴로의 이동을 위한 것이다. 데이터 조작 등의 인터페이스 모델의 주요기능은 기능기술서를 통해 정의되지만, 모델에 제시된 각 아이콘의 기능은 대부분 구현객체 설계 시에 정의된다.

기능기술서는 OOIDEF0 기능모델의 최하위 기능모델을 기반으로 각 기능의 기능처리 과정을 기술한다. <그림 14>는 호선 BOM 관리(F213) 기능하위의 호선 BOM 조회에 대한 기능기술서의 예를 보여준다.

기능기술서 상단에는 기능번호와 기능명, 그리고 기능이 속

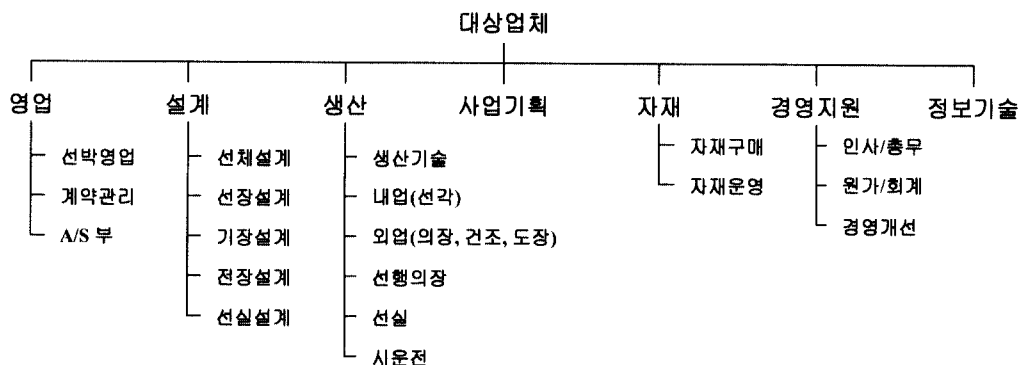


그림 9. 대상업체의 조직도.



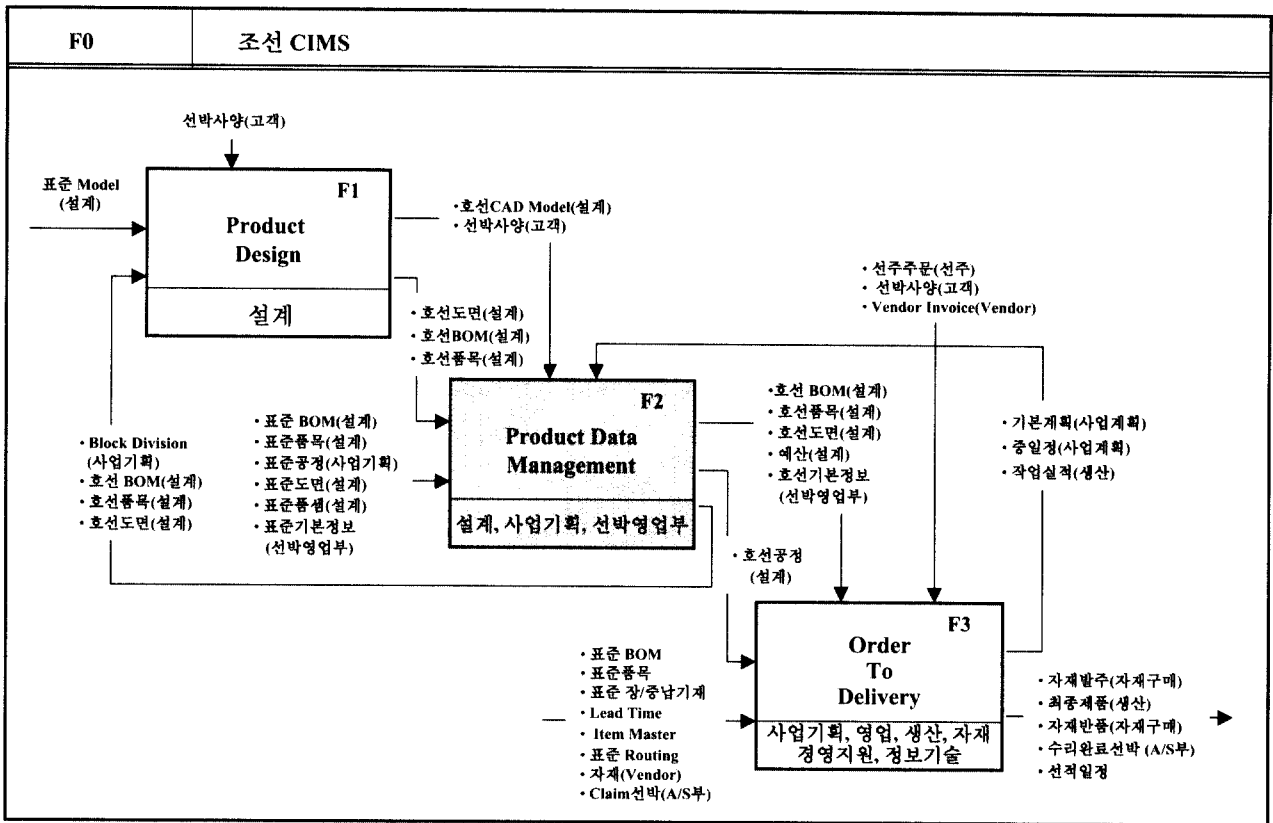


그림 10. 조선 CIM 시스템에 대한 OOIDEF0 기능모델(level 0).

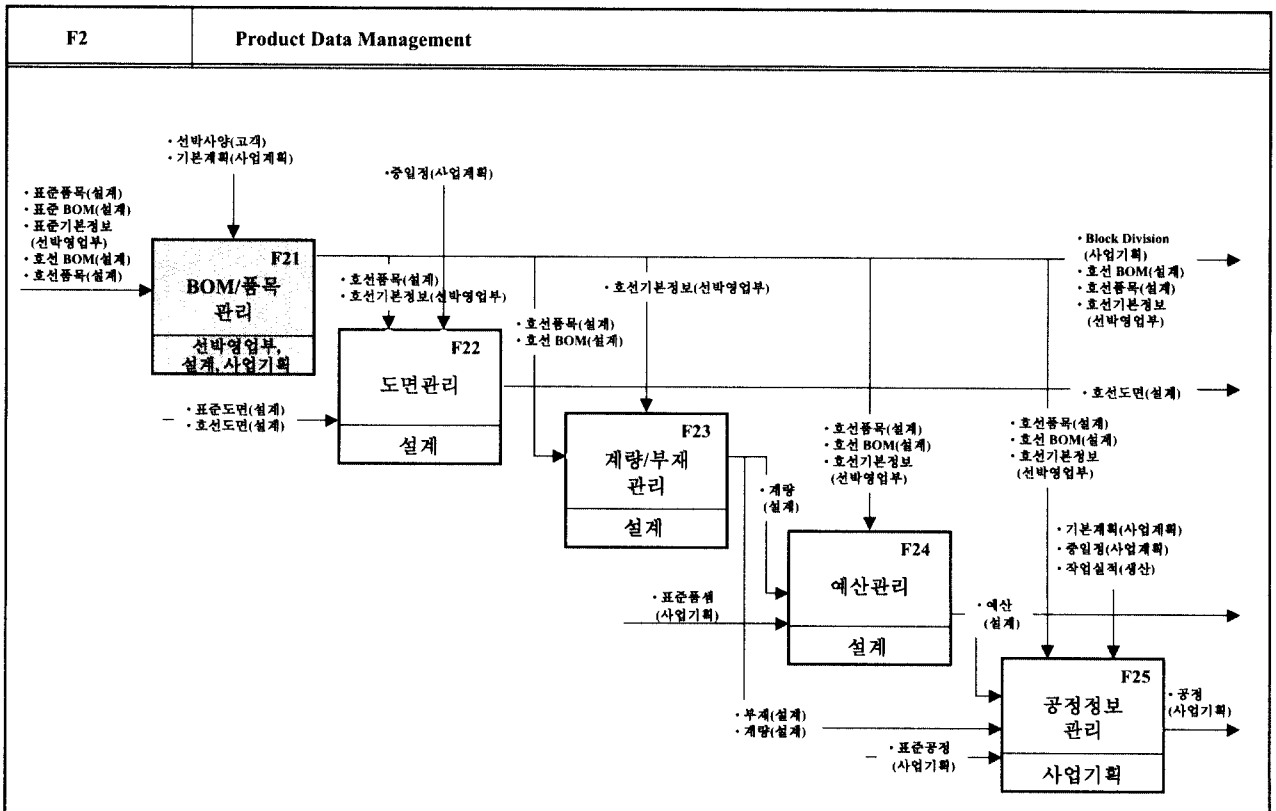


그림 11. PDM에 대한 OOIDEF0 기능모델(level 1).

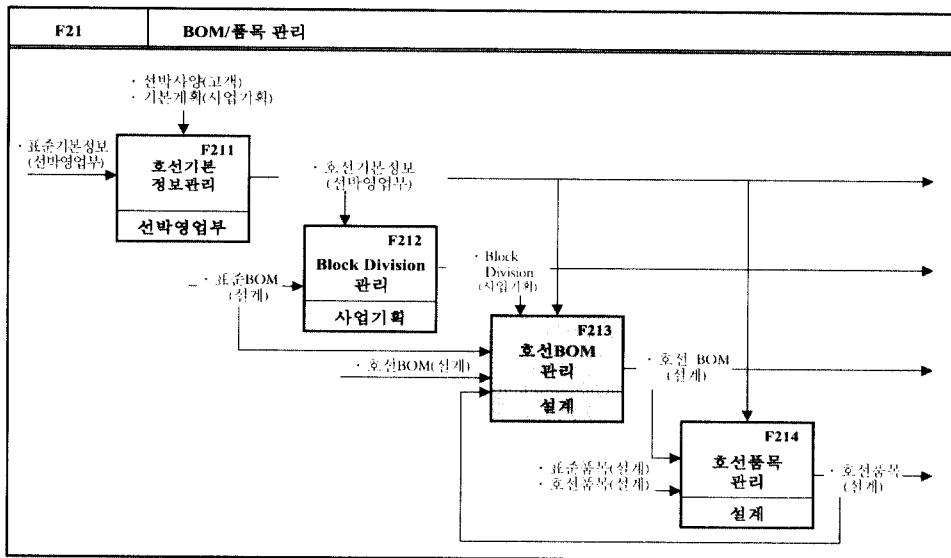


그림 12. BOM/품목관리에 대한 OOIDEFO 기능모델(level 2).

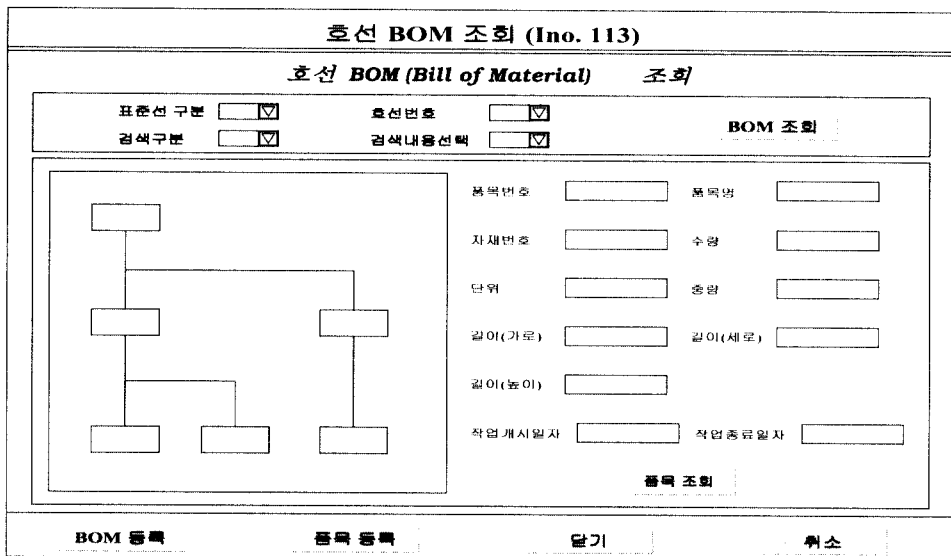


그림 13. 호선 BOM 조회 인터페이스 모델.

한 인터페이스의 번호를 명시한다. 좌측에 투입/제어 객체의 이름과 객체의 속성을 명시하고 우측에 산출객체와 산출객체 속성을 명시한다. 기능처리 과정은 객체의 속성이 어떠한 과정을 거쳐 산출되는가에 대한 내용을 비교적 상세하게 기술한다. <그림 14>에서는 호선 BOM의 구조를 화면 상에 보여주는 일련의 과정을 기능처리 과정의 기술을 통해 보여주고 있으며, 그 순서는 호선번호에 대한 품목 리스트 작성, BOM 구조에 따른 품목 리스트의 정렬, 검색 조건에 따른 BOM 구조 조회, 선택 품목에 대한 품목 상세정보 조회 등으로 구성되어 있다. 검색조건에 따른 BOM 구조 조회와 같은 기능처리 과정이 복잡한 경우 기능기술서 상에는 객체 단위의 기능처리 과정만을 기술하고, 정보객체 또는 구현객체 설계시에 시퀀스 다이어그램 등을 활용한다.

### 3.2.3 시스템 설계

요구사항 분석단계에서 모형화된 OOIDEFO 기능모델을 기반으로 정보객체와 구현객체를 설계하는 단계이다. OOIDEFO 기능모델의 최하위 기능을 중심으로 정보객체의 구조와 속성을 정의하고, 기능기술서를 기반으로 각 객체의 메소드를 정의한다.

정보객체 설계는 OOIDEFO 기능모델에 정의되어 있는 ICOM객체의 구조와 관계를 파악하여 정의된 개략적인 정보객체의 구조를 나타내고, 이러한 개략적인 구조로부터 좀더 자세한 객체간의 관계와 속성, 그리고 클래스 메소드 등을 정의함으로써 이루어진다. <그림 15>은 호선 BOM 정보객체간의 관계와 속성을 최종적으로 표현한 예를 보여준다. 화면 상단에 보이는 클래스가 BOM 구조를 가지는 클래스이며, 하위

기능 번호		F2231	기능 명		호선 BOM 조회	Interface 번호		113
INPUT		PROCESS				OUTPUT		
객체명	객체 속성	처리 내용				객체 속성	객체명	
호선BOM 노트	호선 번호	1. 노트 List 추출 - 호선 BOM 노트 정보로부터 해당 호선 전체에 대한 BOM 노트 List 를 추출한다.				품목번호	BOM Structure Display Pannel	
	품목번호	2. 노트 구조 조합 - 해당 호선에 대한 BOM 구조를 조합한다. - 조합은 품목번호와 해당 품목의 모품목번호와의 관계를 이용한다.				BOM 노트	BOM 노트	
호선 BOM 세부정보	품목번호	3. BOM 구조 추출 - 조합된 호선 BOM 구조를 선택된 검색 조건별로 추출한다 - 해당 조건은 프로그램 파라미터로 전달되며 전체, B/D BOM, 도면 BOM, 그리고 특정 품목번호로 한다.				품목명	BOM Contents Pannel	
	품목번호	4. 품목번호에 대한 품목 속성 추출 - 호선 BOM 세부정보로부터 해당 품목 번호에 대한 세부 속성을 추출한다. - 해당 품목의 세부 속성은 품목형태 두자리로 구분한다. - 품목형태는 선가의 경우 부재, 단위블록, 중조블록, 조립블록, PE 블록, 탑재블록등으로 구분한다.				수량	단위	
	품목번호	5. 자품목 수량 추출 - 호선 BOM 세부정보로부터 해당 품목의 자품목 수량을 추출한다.				해당 속성	자품목 수량	

그림 14. 호선 BOM 조회 기능기술서.

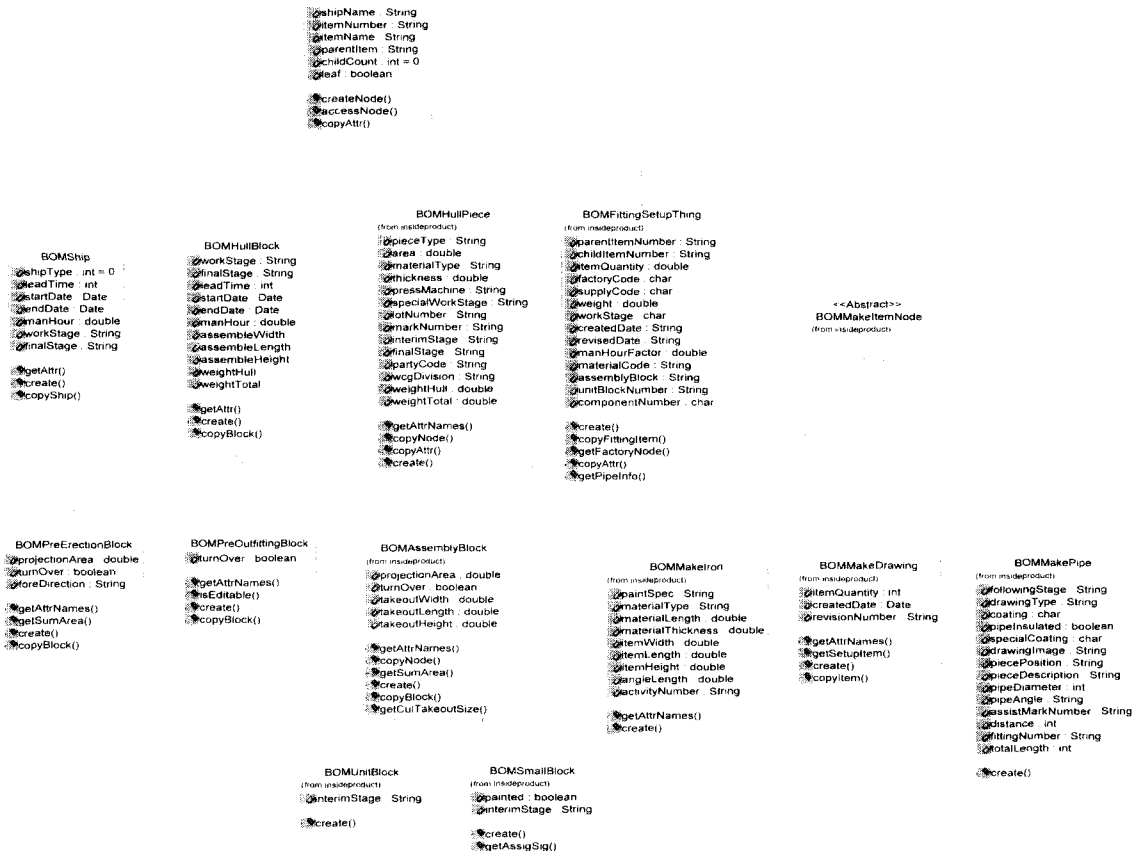


그림 15. 호선 BOM 정보객체의 구조.

로 각 노드에 대응되는 품목의 상세정보를 가지는 클래스들이 존재한다. 각 클래스가 가지는 속성은 품목의 상세정보를 나타내며 메소드는 객체의 생성에 관계되는 메소드만을 표현하고 각 속성의 조작에 관계되는 메소드는 생략하였다. 'get(속성)', 'set(속성)' 등의 속성 조작에 관계되는 메소드는 java 코드 생성시에 모두 고려된다.

구현객체는 사용자 인터페이스 모델과 정보객체간의 관계를 기반으로 설계되며, 해당 기능기술서의 기능처리 과정으로부터 클래스 메소드를 도출하고, 사용자 인터페이스에서 보여지는 데이터를 구현객체의 속성으로 정의한다. <그림 16>는 호선 BOM 조회에 해당하는 구현객체의 예이다. 인터페이스 모델에서 정의하였던 아이콘에 대한 메소드를 구현객체에 반영하여 리스트 박스의 내용을 쿼리하는 메소드를 추가하였고, 정보객체를 생성하여 BOM 조회 버튼이 클릭되었을 때 화면에 보여지도록 하는 메소드를 정의하였다. 그 외에 BOM 조회화면의 기능을 지정하는 메소드를 정의하였다. 구현객체는 정보객체를 운용하는 메소드가 중심이 되며, 구현객체를 기반으로 프로그래밍을 수행하였을 때 추가적인 객체 또는 인터페이스를 최소화하는 방향으로 설계하여야 한다.

### 3.2.4 구현

구현 단계에서는 시스템 설계단계의 정보객체와 구현객체를 이용하여 데이터베이스를 구축하고, 자바 프로그래밍을 수행한다. <그림 17>은 정보객체의 구조로부터 도출한 SQL 스크립트와 자바 코드의 예를 보여준다. 본 연구에서는 SQL 스크립트와 자바 코드를 생성하기 위한 도구로서 Rational Rose를 사용하였다. Rational Rose는 모델로부터 소스 코드를 생성하는 기

능만을 가지고 있으나, PTC(Parametric Technology Corporation)사의 PDM 개발 도구인 Windchill의 기능을 Rose에 추가시켜 SQL 스크립트를 생성할 수 있도록 하였다.

본 연구에서 구현한 프로토타입 시스템은 조선 PDM 시스템의 서브시스템 중에서 BOM/품목 관리이다. <그림 18>은 조선 PDM 시스템의 전반적인 메뉴 구조를 설명하고 있다. <그림 19>는 제품정보 관리시스템의 메인 화면으로 BOM/품목 관리, 예산 관리, 도면 관리, 공정정보 관리, 계량/부재 관리 등의 메뉴를 포함하며, 시스템 접근 초기에 보여지는 사용자 인증화면을 통해 나타난다.

메인 화면의 각 메뉴를 클릭하면 해당 서브시스템의 화면으로 이동하게 된다. 메인 화면과 각 서브시스템의 화면은 HTML을 사용하여 작성하였다. 본 연구에서는 BOM/품목 관리시스템을 프로토타입 시스템으로 개발 하였으므로 BOM/품목 관리시스템 이하 화면을 중심으로 개발화면을 구성하였다. <그림 19>에서 BOM/품목 관리 아이콘을 선택하면 호선 BOM 관리, 호선 품목 관리, 호선 기본정보 관리, Block Division 관리 등의 메뉴가 포함된 메뉴화면이 나타나게 된다. <그림 20>은 서브시스템 메뉴화면에서 선택된 호선 BOM 관리화면을 나타낸다. 호선BOM 관리화면은 호선 BOM 조회, 호선 BOM 등록, BOM 변경관리, 호선 BOM 실적관리의 메뉴를 포함한다.

<그림 21>은 호선 BOM 관리 메뉴에서 선택된 호선 BOM 조회화면을 나타낸다. 호선 BOM 조회화면은 <그림 13>의 호선 BOM 조회 인터페이스 모델을 기반으로 구성되었다. 화면 상단에는 BOM을 조회하기 위한 조건을 입력하고, 하단 좌측에는 각 품목의 품목번호를 BOM 구조에 따라 나타낸다. 좌측의 품목번호를 선택하고 우측 하단의 품목 조회버튼을 선택하

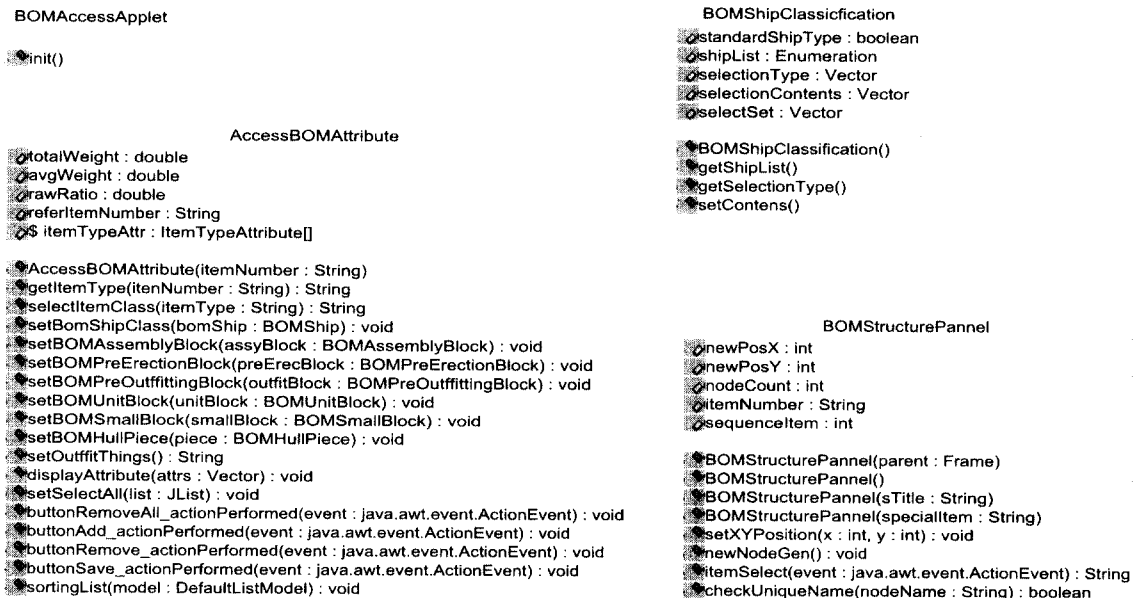


그림 16. 호선 BOM 조회 구현객체의 구조.

<pre>CREATE TABLE BOMHullBlock ( deptNumber VARCHAR2(200), greatCompantNo VARCHAR2(200), objNumber VARCHAR2(200), shipNumber VARCHAR2(200), materialCostC5 NUMBER, productCostC5 NUMBER, productMakeMHC5 NUMBER, productManhourC5 NUMBER, productSettingMHC5 NUMBER, totalCostC5 NUMBER, cuttingLengthB5 NUMBER, fittingLengthB5 NUMBER, metalWeightB5 NUMBER, pieceCountB5 NUMBER, pressMachineCountB5 NUMBER, weightB5 NUMBER, weldingLengthB5 NUMBER, paintAreaA5 NUMBER, materialCostA2A5 NUMBER, productCostA2A5 NUMBER, productMakeMHA2A5 NUMBER, productManhourA2A5 NUMBER, productSettingMHA2A5 NUMBER, totalCostA2A5 NUMBER, createStampA2 DATE, modifyStampA2 DATE, classnameA2A2 VARCHAR2(200), CONSTRAINT PK_HCDHullBlock PRIMARY KEY (idA2A2)) STORAGE ( INITIAL 20K NEXT 20K PCTINCREASE 0 )</pre>	<pre>public class BOMHullBlock extends BOMNode implements ExternalName {  private static final String PREFIX = "fontdes'gn.persist.hull.hllResource "; private static final String FLASHNAME = "HCDHullBlock.class.getName()"; private static final String VERSION_ID = "Header: 0"; private static final String GREAT_COMPANT_NO = "greatCompantNo";  private static int GREAT_COMPANT_NO_UPPER_LIMIT = 1; private static int greatCompantNo = 1;  public static final String HCDPATHT_001 = "HCDPaintObj"; private BCDPaintObj[] theHCDPaintObj;  public static final String HCDPPL_ANALYZE = "theHCDHullAnalysis"; private HCDPPLAnalysis theHCDHullAnalysis;  public static final String HCDCSI_INFO = "theHCDCostInfo"; private HCDCostInfo theHCDCostInfo; static final int serialVersionUID = 1; public static final long serialVersionUID = 7406182178603386331L; protected static final long OLD_FORMAT_VERSION_UID = 18834868288164276431L;</pre>
--	---

그림 17. SQL 스크립트와 자바 코드(호선BOM).

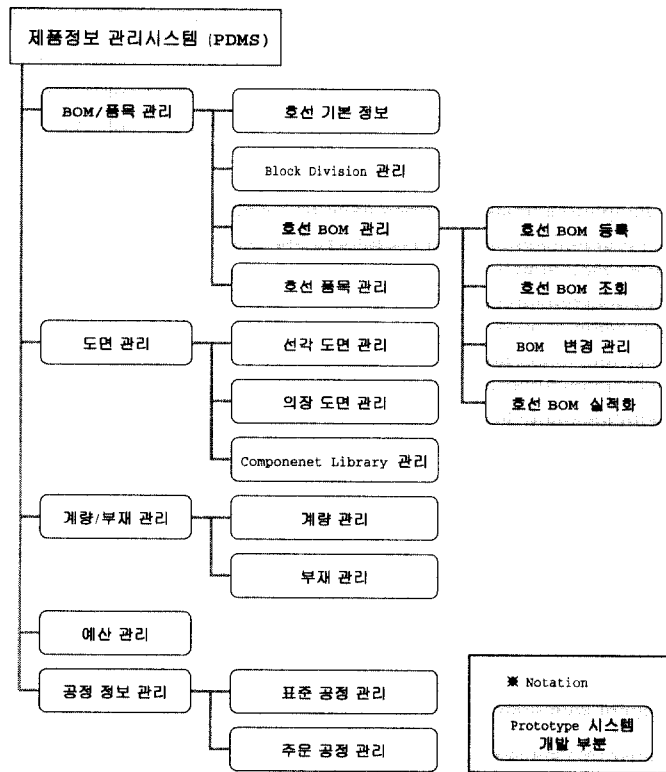


그림 18. 조선 PDM 시스템 메뉴 구조.

면 해당 품목의 상세정보가 나타난다. 화면 하단의 버튼들을 이용하여 다른 애플릿으로 이동할 수 있다.

<그림 22>에서 보여주고 있는 호선 BOM 품목 등록화면은

호선 BOM 품목 하위의 품목을 등록하는 기능을 가지고 있으며, 호선 BOM 조회화면에서 BOM 품목을 선택한 다음, 화면 하단의 BOM 등록 버튼을 선택하면 나타난다. <그림 23>은

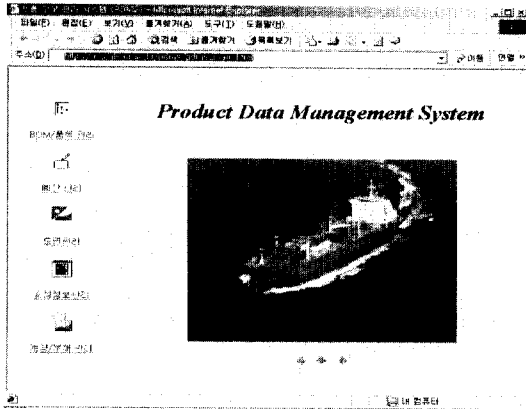


그림 19. 제품정보 관리시스템 메인 메뉴화면.

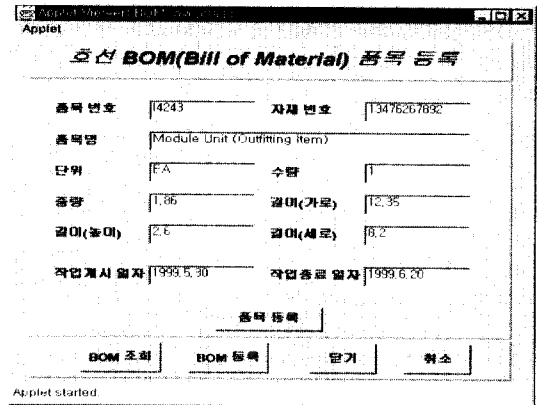


그림 22. 호선 BOM 품목등록 구현화면.

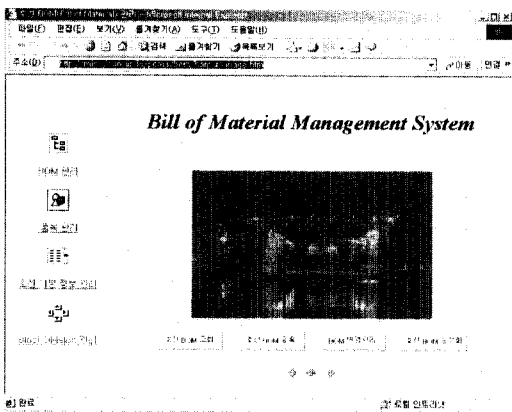


그림 20. 호선 BOM 관리 메뉴화면.

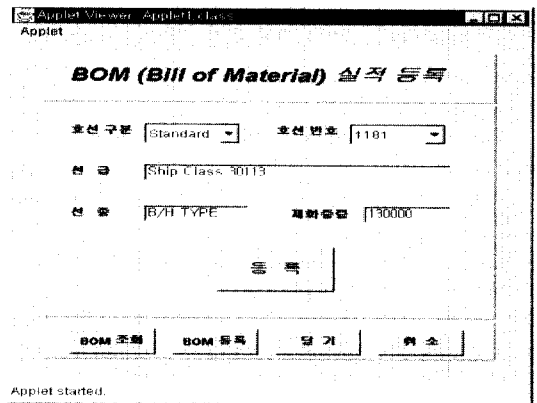


그림 23. 호선 BOM 실적등록 구현화면.

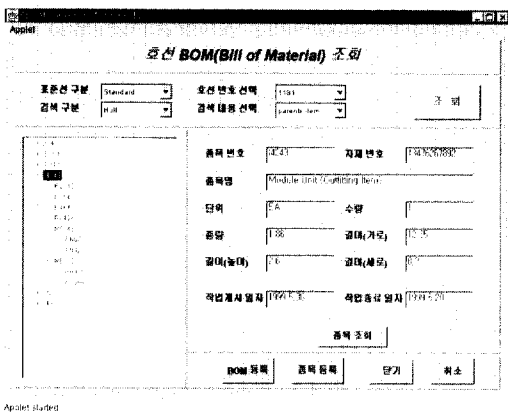


그림 21. 호선 BOM 조회 구현화면.

호선 BOM 실적화 화면을 나타낸다. 호선 실적화는 호선이 완료된 후 호선의 정보를 실적으로 처리하는 것을 의미하며, 실적 호선 Database에 입력하는 과정이다. 호선 실적화는 완료 정보가 입력된 호선의 번호를 입력하면, 해당 호선의 상세정보가 나타나게 되며 사용자는 이것을 확인하고 등록버튼을 선택한다. 등록 과정은 배치처리된다.

구현화면은 인터페이스 모델을 기초로 프로그래밍되며, 본

연구에서는 자바 애플릿으로 구현하여 웹 상에서 실행이 가능하도록 하였다. 프로그래밍은 객체지향 IDEF 기반의 정보시스템 개발 방법론을 제시하고, 응용사례로서 조선 PDM 시스템의 서브시스템인 BOM 관리시스템을 대상으로 방법론의 단계별 적용 과정을 보여주었다. 제시된 방법론은 일관된 프레임워크 하에서 기능모델, 정보모델, 사용자 인터페이스 모델을 체계적으로 통합함으로써 기존의 방법론들에 비하여 개발단계간의 일관성 유지가 용이하고, 사용자 요구사항의 수렴이 용이하다는 장점을 가진다.

#### 4. 결론

본 연구에서는 제조업체의 대용량 시스템 구축을 효율적으로 지원하는 객체지향 IDEF 기반의 정보시스템 개발 방법론을 제시하고, 응용사례로서 조선 PDM 시스템의 서브시스템인 BOM 관리시스템을 대상으로 방법론의 단계별 적용 과정을 보여주었다. 제시된 방법론은 일관된 프레임워크 하에서 기능모델, 정보모델, 사용자 인터페이스 모델을 체계적으로 통합함으로써 기존의 방법론들에 비하여 개발단계간의 일관성 유지가 용이하고, 사용자 요구사항의 수렴이 용이하다는 장점을 가진다.

본 연구에서는 다루지 않았으나, 업무 수행시간, 업무의 지연, 그리고 이벤트의 동시 유발을 통한 기능의 동적 순서와 같은 임시적이면서 시간에 따라 변하는 현상을 강조하는 동적모델 측면을 최근 객체지향 분석 및 설계에 가장 많이 사용되는

UML(Unified Modeling Language)과 연계함과 동시에 객체지향 데이터베이스를 사용한 시스템 개발에 관한 추후 연구가 필요하다.

## 참고문헌

- Adam, N. R., Gangopadhy, A.(1993), Integrating Functional and Data Modeling in a Computer Integrated Manufacturing System, *IEEE 9<sup>th</sup> International Conference on Data Engineering*, 302-309.
- Alabiso, B.(1988), Transformation of Data Flow Analysis Models to Object Oriented Design, *OOPSLA'88 Proceedings*, 335-353.
- Balin, S. C.(1989), An Object-oriented Requirements Specification Method, *Communications of the ACM*, 32, 608-623.
- Choi, Y. H.(1995), PDM System for CE Implementation, *Computer World*, 12, 162-167.
- Fowler, M. and Scott, K.(1997), UML Distilled-Appling the Standard Object Modeling Language, *Addison Wesley*.
- Hoydalsvik, G. M. and Sindre, G.(1993), On the Purpose of Object-oriented Analysis, *OOPSLA'93 Proceedings*, 240-255.
- Hwang, S. R., et al.(1999), An Integrated Modeling Methodology Based Object-Oriented IDEF : A Case Study of Ship Building, *The Journal of MIS Research*, 9(3), 47-73.
- Jacobson, I.(1992), Object Oriented Software Engineering : A Use Case Driven Approach, *Addison-Wesley*.
- Kim, H. K., Hotaka Ryosuke(1998), Introduction to Object-Oriented Information System Design. Dong IL Publishing Co. Ltd., Seoul, Korea.
- Kim, J. I.(1995), Function, Information, Dynamics, and Organization Integrated Modeling Methodology for Enterprise Systems Integration, *Dissertation, Arizona State University*.
- Kim, Y. S.(1998), A Development Methodology for Information System, *Microsoft Developer Journal*.
- Lee, C. H.(1996), A Case Study of Constructing Integrated Research and Development System Using CALS/PDM, *IE Magazine*, 3(1), 58-62.
- Malhotra, R. and Jayaraman, S.(1992), An Integrated Framework for Enterprise Modeling, *Journal of Manufacturing Systems*, 11(6), 126-441.
- Opdahl, A. L. and Sindre, G.(1994), A Taxonomy for Real-World Modeling Concept, *Information Systems*, 19(3), 229-241.
- Quatrani, T.(1998), Visual Modeling with Rational Rose and UML, *Addison Wesley*.
- Park, M. H., et al.(1995), Overview of Domestic Ship Industry and Domestic/Abroad Researches Related to Industrial Engineering, *IE interfaces*, 8(2), 5-22.
- Rumbough, J.(1991) et al, Object-oriented Modeling and Design, *Prentice Hall*.
- Scidewitz, E. and Stark, M.(1987), Toward a General Object-oriented Software Development Methodology, *Ada Letters*, 6(4), 54-67.
- Shlaer, S. and Mellor, S. J.(1991), Object Life Cycles : Modeling the World in States, *Yourden Press*.
- Sutcliffe, A. G.(1991), Object-oriented System Development : Survey of Structured Method, *Information and Software Technology*, 33(6), 433-442.
- Tagg, R. and Liew, B.(1993), Object-oriented Database Methodology-State of the Art, *Proceedings of the 11<sup>th</sup> British National Conference on Database : Advanced in Database*, 147-161.
- Young, R. E. and Vesterager, J.(1991), An Approach to CIM System Development Whereby Manufacturing People Can Design and Build Their Own CIM Systems, *International Journal of Computer Integrated Manufacturing*, 4(5), 288-299.



### 김재균

1979년 인하대학교 산업공학 학사  
 1981년 한국과학기술원 산업공학 석사  
 1992년 한국과학기술원 경영과학 박사  
 현재: 울산대학교 산업정보경영공학부 교수  
 관심분야: CIM, PDM, DB응용, WEB Application, 통신망설계 등



### 장길상

1986년 울산대학교 산업공학과 학사  
 1988년 한국과학기술원 산업공학 석사  
 1997년 한국과학기술원 경영정보공학 박사  
 현재: 울산대학교 경영학부 경영정보학 전공 교수  
 관심분야: DB 응용, ERP, e-business, DW, 생산 정보시스템, 객체지향 분석 및 설계 등