▩ 연구논문

웹기반 Generic BOM 관리 시스템의 설계 및 구현

Modeling and Implementation for Web-based Generic BOM Management System

장 길 상[1]
Jang, Gil Sang
김 재 균[2]
Kim, Jae Gyun
이 종 훈[3]
Lee, Chong Hun

Abstract

BOM(Bill of Materials) is a listing or description of raw materials, parts, and assemblies that define a product. In manufacturing companies that produce various products with short life cycles, it is very important to manage the BOM of products with various options and versions efficiently. BOM information should be required to share among the departments of engineering, production control, material purchasing, cost management, and shop floor within a manufacturing company. In order to share BOM information efficiently, information system based on WWW(World Wide Web) should be developed. Thus, this paper models and implements Web-based Generic BOM Management System (WGBMS) for management of a number of variant products which are consisted of standard parts under ATO(Assemble To Order) production environment. Also, object-oriented methodology was used to model a generic BOM and phases of navigational design and interface design were introduced to implement the prototype of WGBMS. And Object-Relational Database Management System (ORDBMS) was used to construct the WGBMS database.

1. 서론

BOM은 부품(Parts), 조립품(Assemblies), 원자재(Raw Materials)로 구성되는 제품구조(Product Structure Tree)를 정의하고 기업의 제품 정보를 표현하는 가장 중요한 기본자료로써 제품사양, 도면, 라우팅(Routing), 인도기간, 재고수준, 생산비용 등 다양한 정보를 제공한다[13]. 또한 BOM은 제품을 구성하는 각 구성품들에 대한 구매, 발주 및 생산지시의 시점을 결정하는 기준생산계획(Master Production Scheduling) 및 자재소요계획(Material Requirement Planning)의 중요한 입력자료가 된다[14]. 따라서 BOM을 효율적으로 관리하는 것은 제조업의 생산성 향상에 매우 중요하다.

이러한 BOM 체계 중에서 현재 널리 사용되고 있는 것은 전통적인 BOM, Modular BOM, Generic BOM이다. 이 세가지 BOM의 장단점을 설명하면 다음과 같다. 전통적인

1) 동국대학교 정보산업학과 교수
2) 울산대학교 산업공학과 교수
3) 울산대학교 산업공학과 석사과정

BOM(Conventional BOM)은 각 제품에 대하여 각각의 BOM을 독립적으로 관리하는 개념으로, BOM 구성이 쉬운 것이 장점이다. 그러나, 오늘날과 같이 제품의 수명주기가 짧으면서 다양한 선택사양을 갖는 제품들을 생산하는 제조환경에서 전통적인 BOM 체계는 빈번한 BOM 구성으로 BOM 수의 증가 및 공용 부품들의 중복적인 유지, 그리고 설계변경에 따른 BOM 변경의 복잡함과 변경사항에 대한 대응의 비효율성 등의 단점을 가지고 있다. 이러한 전통적 BOM의 문제점을 해결하기 위해서 제시된 대안이 Modular BOM이다[10].

Modular BOM은 공용부품과 옵션부품으로 나누어 각각의 정보를 유지하여 전통적인 BOM의 문제점들은 극복했으나, 제품 전체 구조의 파악과 상위 옵션을 모듈(Module)에 반영하기가 어려운 문제점이 있다. 이러한 문제점을 해결하기 위해 제시된 개념이 Generic BOM이다[김정기 등, 1997].

Generic BOM(Generic Bill of Material)은 전반적인 제품 구조와 옵션 관리를 효율적으로 할 수 있고, 기존의 BOM 데이터를 재사용할 수 있으며[4], BOM 구조의 투명성을 높이고, 공용부품의 중복을 제거하여 준다[15]. 따라서, 이러한 Generic BOM이 제품의 수명주기가 짧으면서 다양한 선택사양을 갖는 제품들을 신속하게 생산해야 하는 오늘날의 제조환경에 적합한 BOM 관리 체계라고 사료된다[9].

위에서 설명된 이러한 BOM체계들의 효율적인 관리를 위한 BOM 관리 시스템의 설계 및 구현에 관한 많은 연구가 있었다. Nandakumar[1985]는 BOM 프로세스를 관계형 데이터베이스 관리 시스템(Relational Database Management System)을 이용하여 설계하고 구현하였다. 그러나 이 논문에서는 BOM 프로세서를 정보검색이라는 단순한 기능에 한정하여 제시하였기 때문에 현실적으로 활용하기에는 한계가 있었다. Chung 등[1992]은 객체지향기법을 이용한 BOM의 데이터 모형을 제시하고, C++언어를 이용하여 간단한 기능을 갖는 객체지향 BOM 시스템의 프로토타입(Prototype)을 보여 주었다. 그러나, 이 논문의 객체지향 모델링에서는 객체의 프로세스 측면을 고려하지 못하였고, C++ 언어를 이용하여 객체지향 BOM 데이터베이스를 구축하기에는 현실적으로 매우 어려웠다. 고석완 등[1997]과 오태훈 등[1997]은 객체지향기법을 이용한 전통적인 BOM 관리 시스템을 설계하였다. 김정기 등[1997]은 웹 기반의 Generic BOM 관리 시스템을 구현하는 방법을 제시하였으나, 객체지향기법을 이용한 Generic BOM의 개념적인 데이터 모형만을 제시하였다. 이동국 등[1999]은 객체지향 방법론인 OMT를 사용하여 전통적인 클라이언트/서버(Client/Server) 환경에서의 Generic BOM 관리 시스템의 설계 및 구축에 관한 연구를 수행하였다.

본 논문은 이동국 등[1999]의 클라이언트/서버 기반의 Generic BOM 관리 시스템의 설계 및 구축에 관한 연구를 BOM의 효과적인 정보 공유를 위하여 웹(Web) 기반의 시스템으로 확장한 것이다. 즉, BOM 정보를 활용 하는 많은 부서들, 예를 들면, 설계부, 제품개발부, 생산관리부, 자재관리부, 원가관리부 등 관련 부서들의 지역적인 제약을 극복하고, 고객들에게 제품 관련 정보를 제공하며, 전자상거래(Electronic Commerce)와 연계하여 제품의 견적 및 주문 시스템과 통합하기 위하여 웹 기반의 Generic BOM 관리 시스템(WGBMS : Web-based Generic Bill of Material Management System)의 개발이 요구된다. 이러한 웹 기반의 정보 시스템은 인터넷 기반의 제품 판매 전략에 적합하며, 가상공간에서 고객이 원하는 제품을 직접 구성하여 고객이 원하는 제품을 사전에 예측할 수 있게 해준다.

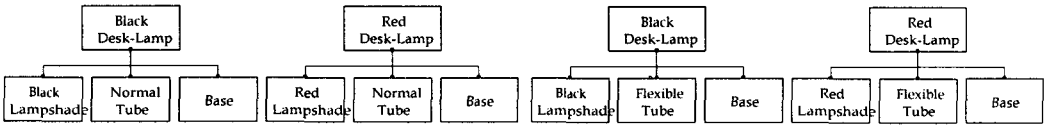
따라서, 본 논문에서는 표준 부품으로 구성된 ATO(Assemble To Order) 환경에서 수명주기가 짧고 다양한 선택사양을 갖는 제품들을 생산하는 제조환경에 가장 적합한 Generic BOM을 연구대상으로 하고, 맞춤형 컴퓨터를 생산하는 제조업체를 대상으로 웹 기반의 Generic BOM 관리 시스템을 설계하고 프로토타입 시스템을 구현하고자 한다.

2. Generic BOM

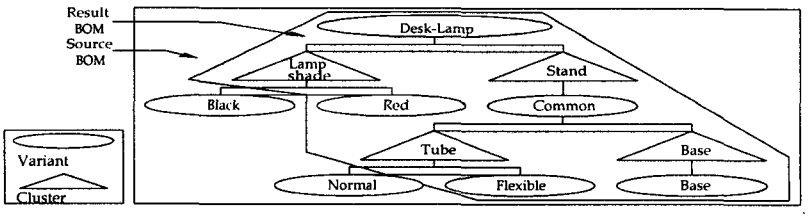
2.1 Generic BOM의 구조

동일한 구조를 다소 갖는 최종 제품(End Item)들의 집합을 Generic 제품이라고 한다. 이러한 Generic 제품의 구조를 효과적으로 표현해 주는 것이 Generic BOM이다[19]. Generic BOM은 새로운 유사제품이 추가가 될 때 제품의 옵션인 Cluster 관리를 통하여 효과적으로 반영할 수 있다. 예를 들어 새로운 옵션이 생겼다면 Cluster와 그것의 Variant들을 생성하여 BOM 정보를 관리하고, 옵션의 한 사양이 기존의 제품에 없다면 그 사양(Variant)을 추가하여 관리한다. 따라서 Generic BOM은 BOM 구조와 옵션 관리를 통해 부품정보의 재사용을 가능하게 한다. 이렇게 Generic BOM에서 전체 제품 구조 및 부품 정보를 가지고 있는 BOM을 Source BOM이라 한다. 그리고 Source BOM의 정보를 기반으로 원하는 제품의 BOM을 구성한 것이 Result BOM이다. 이러한 Result BOM을 통해 제품의 이력정보를 전통적인 BOM보다 효율적으로 관리하는 것이 Generic BOM의 특징이기 때문에, 다양한 변형 및 짧은 수명주기를 갖는 제품에 적합한 BOM이다[9].

Generic BOM의 구조적 특징은 크게 Source BOM과 Result BOM으로 구성되어 진다. Source BOM 내에서 완제품과 조립품은 Cluster에 의해 구성되어지고, Cluster는 오직 하나의 상위 부품을 가지며 조립품 또는 원자재와 같은 Variant에 의해 구성되어진다. Cluster는 옵션에 해당하고 각 Variant는 옵션의 사양에 해당한다. 그러므로 Result BOM은 Cluster 내의 Variant 중 하나만을 선택할 수 있으며 선택된 Variant들을 가지고 제품 생산을 위한 BOM이 구성되어 진다[9].



<그림 1> Desk-Lamp를 위한 전통적인 BOM



<그림 2> Desk-Lamp를 위한 Generic BOM

Generic BOM의 구조적인 특징은 고객의 요구에 따라 변경이 많고 제품의 옵션에 대한 선택사양이 많은 제품을 예로 들어 설명하고자 한다. <그림-1>은 여러 종류의 탁상램프(Desk-Lamp)를 전통적인 BOM의 형태로 나타내는 것이고, 이를 Generic BOM을 통해 표현한 것이 <그림-2>이다. 여기서 완제품인 탁상램프는 Cluster에 해당하는 Lamp-shade와 Stand로 구성되고, Cluster중 Lamp-shade는 Variant인 Back color와 Red color로 구성되어진다. 실제

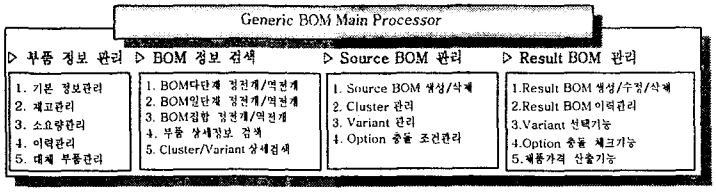
Result BOM을 구성할 때는 Cluster중 하나의 Variant만이 선택되어진다. 이러한 방법에 의해 선택된 Variant들의 모임이 바로 Result BOM을 의미한다.

2.2 Generic BOM 프로세스의 기능

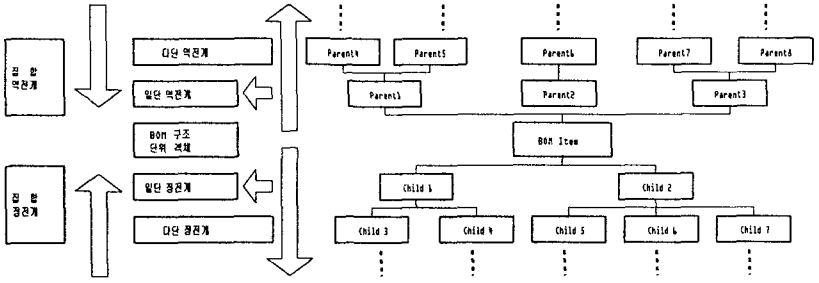
본 논문은 Generic BOM 프로세스의 주요 기능을 <그림-3>과 같이 부품 정보관리, BOM 정보검색, Source BOM관리, Result BOM관리의 4가지로 구분하였다[9].

부품 정보관리는 기본 정보관리, 재고관리, 소요량관리, 이력관리 등의 역할을 한다. 또한, 대체부품을 관리하여 필요시 효율적으로 대처할 수 있게 한다.

BOM정보 검색기능은 Source BOM과 Result BOM의 정전개 및 역전개 검색으로 나누어 진다.



<그림 3> Generic BOM 프로세스의 주요 기능



<그림 4> BOM 전개 구분

정전개는 하위 부품을 모두 검색하는 다단계 정전개와 집합 정전개, 그리고 한 단계(Level) 아래의 부품만을 검색하는 일단계 정전개가 있다. 역전개는 상위 부품을 검색하는 기능으로 그 전개 방법은 정전개와 동일하다[1,6,7,8]. <그림-4>는 BOM 정보의 전개 방법을 도시한 것으로 제품의 한 부품(BOM Item)을 중심으로 그 하위 부품을 "Child" 로 나타내고 그 상위 부품을 "Parent" 로 나타낸 것이다[5]. 또한 BOM정보 검색기능에는 Generic BOM의 특성인 옵션의 정보를 가지고 있는 Cluster정보 검색과 옵션의 한 사양인 Variant정보 검색 기능이 있다.

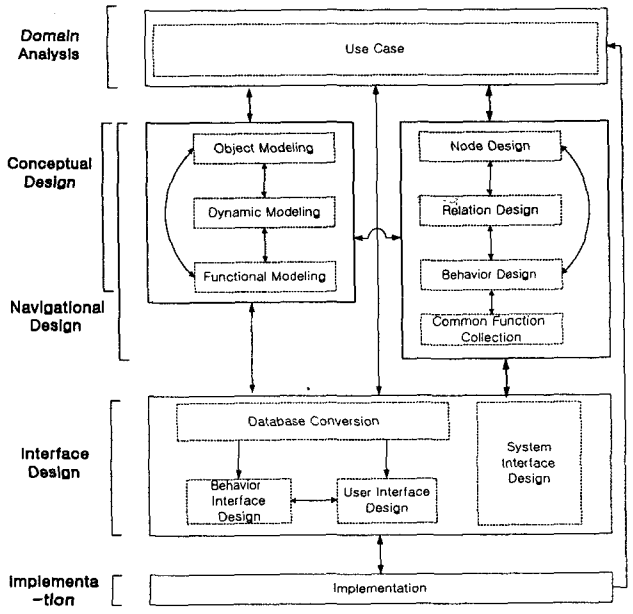
Source BOM관리 기능에서는 Source BOM을 생성하고 삭제하는 기능과 Cluster 및 Variant 정보를 관리(Source BOM 수정)하는 기능이 있다. 즉, Cluster정보 관리기능에서는 새로운 제품이 추가될 때 기존의 제품에 없는 새로운 옵션을 생성하는 기능(Cluster 생성)과 제품의 변경에 따라 옵션의 수정 및 삭제하는(Cluster 수정/삭제) 기능이 있다. Variant정보 관리 기능에서는 옵션에 새로운 사양을 추가/수정/삭제하는(Variant생성/수정/삭제) 기능이 있다. 그리고 각 옵션별 사양의 충돌을 막기 위해 충돌 부품의 정보를 추가/수정/삭제하는 기능이 있다.

Result BOM 관리 기능에서는 Result BOM의 생성, 수정, 삭제하는 기능과 수정된 Result BOM의 이력관리 및 Result BOM의 생성을 위한 사양을 선택(Variant 선택)하는 기능, 사양 선택시 충돌 부품을 체크해 주는 기능이 있다. 마지막으로 Result BOM으로 생산되는 제품의 공장도 가격과 소비자 가격을 계산하는 가격 산출 기능이 있다.

지금까지는 본 논문에서 다루고자 하는Generic BOM의 구조 및 프로세스의 특성을 기술하였고, 다음 장에서는 이러한 Generic BOM을 모델링하고 설계하기 위한 웹 기반의 객체지향 정보시스템 설계방법을 기술하고자 한다.

3. 웹 기반의 객체지향 정보시스템 설계방법

본 논문에서는 웹 기반 정보 시스템을 구축하기 위한 설계방법을 크게 5단계로 나누어 설명한다. 설계 단계는 도메인 분석(Domain Analysis) 단계, 개념 설계(Conceptual Design), 네비게이션 설계(Navigational Design), 인터페이스 설계(Interface Design) 그리고 구현(Implementation) 단계로 구성된다. 본 논문에서 사용한 설계방법은 객체지향 정보시스템의 분석 및 설계 방법론을 웹과 인터페이스 측면에서 확장한 것이다. <그림-5>는 본 논문의 웹 기반 정보시스템의 설계절차를 나타내는 흐름도이다.



<그림 5> 웹 기반 정보시스템 설계 절차

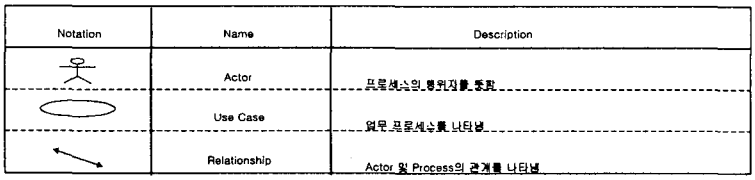
3.1 도메인 분석 (Domain Analysis)

도메인 분석 단계는 대상 업무를 분석하기 위하여 Jacobson의 OOSE(Object-Oriented Software Engineering) 방법론 중의 Use Case 모형을 이용한다.

Use Case 다이어그램의 요구분석 기법은 다른 방법론에 비해 매우 혁신적이다. 즉, 대부분의 객체지향 방법론은 객체를 매우 찾기 쉽고 인식하기 쉬운 것으로 가정하고 있다. 그러나,

개체-관계형(Entity-Relationship) 다이어그램 작성시 개체(Entity)를 식별하기 어려운 것처럼, 실제 분석에서 객체를 찾는 일이 그렇게 쉽지 만은 않다. 따라서, 이러한 객체(Object)를 사용자의 요구사항으로부터 자연스럽게 유도하고 사용자와 자연어가 아닌 정형화된 다이어그램 형식으로 의사소통을 위하여 Use Case 다이어그램이 사용된다[2]. 또한 Use Case 다이어그램은 비즈니스 프로세스 리엔지니어링(Business Process Reengineering)에서도 사용된다. 기존의 프로세스 맵(Process Map)보다 쉽게 작성할 수 있고, 이해하기 쉽다는 장점을 갖고 있으며, 재편성된 모델은 곧바로 정보시스템 구축의 분석단계에서 사용된다.

Use Case는 시스템 외부 Actor와 시스템과의 상호작용을 위하여 시스템이 제공하는 일련의 트랜젝션 또는 기능이다. 따라서 구조적 방법론의 Context 다이어그램과 비슷한 위치에 있다고 볼 수 있다. 그러나, Use Case 다이어그램은 시스템의 요구사항을 정의하기가 보다 용이하다. 각 Actor와 관련된 Use Case를 정의하면서 각 Use Case를 이루는 이벤트(Event)를 정의하게 된다. 이러한 이벤트들을 조합하여 여러 시나리오가 작성되고, 각 시나리오는 하나의 Use Case를 완성하는 여러 경로를 의미한다. 즉, 에러 상황이나 기타 상황에 따른 경로를 포함한다. <그림-6>은 Use Case 다이어그램의 기본 표기법을 나타낸 것이다. Use Case, 이벤트, 시나리오에 기술된 문서의 내용을 통하여 객체를 추출하게 된다. 객체는 보통 명사형이나 동사의 명사형 등으로 표현되어 있다.



<그림 6> Use Case 다이어그램의 기본 표기법

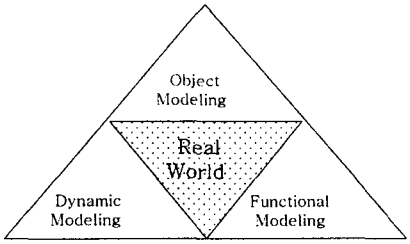
하지만 이러한 명사형에는 Actor가 존재하기도 하고, 한 객체의 특성을 나타내는 경우도 있으므로 이를 추출해 내야 한다. 여러 Use Case들로부터 객체들을 찾으면, 이들은 클래스 다이어그램(Class Diagram)이나 Sequence 다이어그램에서 사용되어 진다[2].

3.2 개념 설계 (Conceptual Design)

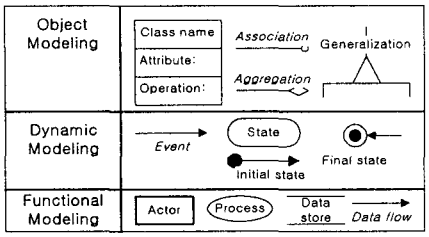
개념 설계 단계는 실세계의 정보 개체를 데이터 구조와 행위가 결합된 형태로 설계하기 위하여 Rumbaugh의 OMT 방법론을 사용하여 세가지 모형을 설계하는 단계이다. Rumbaugh의 OMT는 설계할 시스템을 <그림-7>과 같이 세 가지의 관점에서 모델링하는 방법론이다[5,15,21]. <그림-8>은 OMT에서 사용되는 기본적인 모델링 기호를 나타낸 것이다. 각 모델링은 <그림-9>와 같은 수행 절차에 의해 수행되고, 그 모델링 간에는 상호 보완하는 관계가 형성되어 진다. 각 모델링에 관한 설명은 다음과 같다[9].

객체 모델링(Object Modeling)은 시스템에서 요구되는 객체를 찾아내어 객체들의 특성과 객체들 간의 관계, 속성, 오퍼레이션을 보여준다[5,13]. 시스템의 정적인 구조를 나타내며 시스템을 객체 관점에서 실세계에 정확하게 표현할 수 있다. 그리고 동적 모델링과 기능 모델링은 OMT의 중심 모형인 객체 모델링을 기반으로 설계되어 진다.

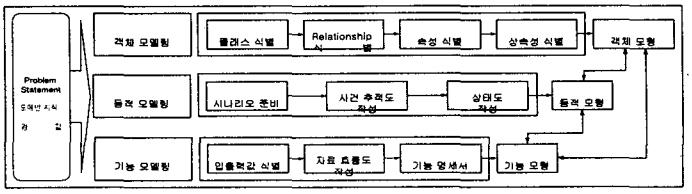
동적 모델링(Dynamic Modeling)은 객체 모델링에서 규명된 객체들의 행위와 객체들의 상태를 포함하는 라이프 사이클을 보여준다[5].



<그림 7> OMT 기본구조



<그림 8> OMT의 기본적인 모델링 기호



<그림 9> OMT의 상세 절차

특정시점에서 각 객체와 그 반응에 대한 변경사항을 보여주는 것으로, 반응시간을 보여주는 사건추적도(Event Trace Diagram)와 변경상태를 보여주는 상태도(State Diagram)는 시스템 내에서 Actor들에 의해 발생하는 사건(Event)과 업무의 단계를 한눈에 보여주므로 누락업무를 쉽게 파악하여 객체 모델링에서 설계된 객체들의 타당성을 검증하고 보완할 수 있다.

기능 모델링(Functional Modeling)은 객체 모델링에서 규명된 객체와 동적 모델링에서 밝혀진 각 객체의 상태가 변화할 때 수행되는 동작들을 기술하는데 사용된다[5]. 객체 모델링과 동적 모델링의 결과를 통해 입력 값에서 어떻게 출력 값이 발생하는지를 자료 흐름으로 보여주고, 각 프로세스에서 생성된 데이터가 어디에 저장되는 지를 자료흐름도(Data Flow Diagram)을 통해서 나타낸다. 또한, 각 프로세스는 객체 모델링의 오퍼레이션과 동적 모델링의 사건이 어떻게 수행되는지를 나타낸다.

즉, 객체 모델링을 중심으로 동적 모델링은 사건에 따른 상태변화를 통해 객체 모형을 검증하고 기능 모델링은 데이터의 흐름과 그 수행동작을 통해 객체 모형의 데이터와 함수들을 확인할 수 있다. 세 가지 모델링은 서로 보완하는 단계를 거쳐 보다 완전한 시스템을 설계할 수 있다.

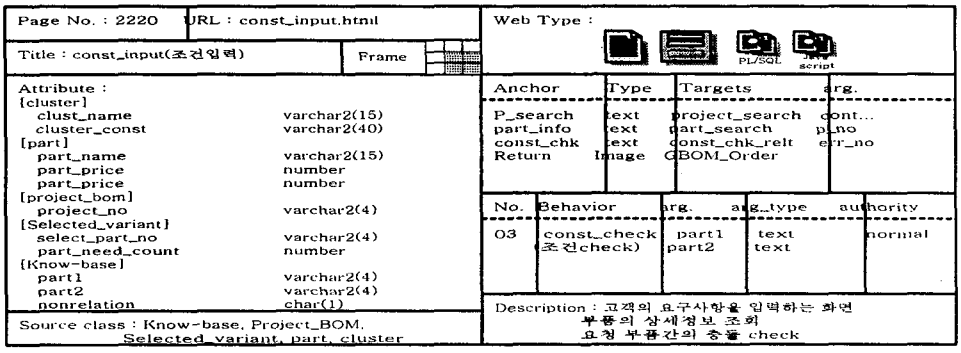
그리고, OMT에서는 소프트웨어 시스템을 세 가지 측면, 즉, 객체, 동적, 기능 관점에서 각기 보완적인 다른 면을 기술함으로써 실세계의 현상에 가까운 모형을 제시하고, 이러한 세 가지 관점을 체계적으로 통합하여 유연성과 적응력을 가진 우수한 품질의 소프트웨어를 만들 수 있는 적합한 방법이라고 주장하였다[5]. 즉, OMT에서는 객체지향 모델링의 장점을 살려 데이터 모델링 뿐만 아니라 각 프로세스의 관점 및 사건과 그 상태의 변화까지 고려한 모델링이 가능하기 때문에 실제 시스템을 실세계에 근접하게 모형화 할 수 있다.

3.3 네비게이션 설계 (Navigational Design)

네비게이션 설계 단계는 크게 도메인 분석 단계의 Use Case로부터 전반적인 업무의 흐름을 파악하고, 개념 설계 단계인 OMT에서 데이터의 기본 구조 및 연관 관계를 기반으로 설계되어 진다. 설계 과정은 노드 설계, 관계 설계, 행위자 설계, 공통 모듈 수집 과정을 거쳐서 설계되어 진다.

노드 설계(Node Design) 과정은 각 화면(View)에 필요한 속성, 사용 객체 또는 테이블(Table), URL, 제목(Title), 웹 타입(Web Type) 등을 정의한다. 또한 웹 환경의 특징 중 하나인 프레임의 위치를 표현함으로써 보다 효과적인 설계를 기대할 수 있다. 노드 설계는 객체 모델링을 기반으로 필요한 속성(Attribute)들을 나타낸다. 각 객체 또는 테이블은 속성이 정의된 데이터베이스를 뜻하고 URL은 구현시 사용되어지는 웹페이지를 주소를 나타내며, 제목은 이 노드의 제목에 해당한다. 웹 타입은 각 노드가 어떤 방법으로 구현될 지를 나타내며 이 타입에 따라 구현 언어 및 툴이 결정되어 진다.

관계 설계(Relation Design) 과정은 노드 설계를 기반으로 노드 간의 관계를 정의하는 단계이다. 노드의 관계를 위하여 앵커(Anchor)와 링크(Link)의 개념을 통해 정의되어 진다. 앵커는 다른 노드로 연결하기 위한 매개체 역할을 하고 앵커에 의해 다른 노드와 관계를 맺는 것을 링크라고 한다. 이 단계는 개념적 설계의 동적 모델링과 기능 모델링을 기반으로 작성되어 진다. 관계 설계에서 주목할 점은 노드에서 노드로 이어질 때 넘겨주는 인자(Argument)들을 정의함으로써 각 노드간의 필요한 값들을 전달할 수 있게 했다.



<그림 10> 네비게이션 설계

행위자 설계(Behavior Design) 과정은 각 노드에서 수행하는 기능을 정의한 것으로 개념적 모델의 기능 모델링을 기반으로 작성되어 진다. <그림-10>은 고객이 원하는 사양을 입력하는 노드를 정의하고, 조건입력 노드에서 다른 노드와 관계를 정의하고 그 관계를 위한 앵커와 링크가 정의되어 있다. 그리고, 제품을 구성하는 부품들 간의 충돌여부를 체크하는 행위자와 그 인자를 정의한다.

공통 모듈 수집(Common Function Collection) 과정은 OMT 방법론에 의해 설계된 모형을 기반으로 각 노드에서 공용으로 사용되는 행위자를 공용 프로세스로 정의하고, 추출해 냄으로써 중복된 프로세스의 설계 및 코딩을 막을 수 있다. 네비케이셔널 스키마(Navigational Schema)는 노드 설계, 관계 설계, 행위자 설계를 통하여 작성된 네비게이셔널 스키마 다이어그램이다. 이 다이어그램은 각 노드의 관계와 전체적인 업무의 흐름을 파악할 수 있게 한다.

3.4 인터페이스 설계 (Interface Design)

인터페이스 설계 단계는 데이터베이스 변환(Database Conversion), 행위자 인터페이스 설계(Behavior Interface Design), 유저 인터페이스 설계(User Interface Design), 그리고 시스템 인터페이스 설계(System Interface Design) 과정으로 이루어져 있다.

데이터베이스 변환과정은 객체 모형을 객체-관계형 데이터베이스로 설계하는 과정이다. 변

환 방법은 객체 모형에서의 Generalization을 객체-관계형 데이터베이스의 "Super-Type/Sub-Type"으로 표현하고, Association을 "Reference"로 표현하며, Aggregation을 "Nested Table"로 구축한다. 그리고 클래스 내의 오퍼레이션은 객체-관계형 데이터베이스의 함수로 변환한다[15]. <표-1>은 앞에서 생성된 객체 모형을 객체-관계형 데이터베이스로 변환하기 위한 방법이다[3].

객체지향 방법론	객체-관계형 데이터베이스
Generalization	Super-Type/Sub-Type
Association	Reference
Aggregation	Nested-Table
Operation	Function

<표 1> 객체 모형에서 객체-관계형 데이터베이스로 변환 방식

행위자 인터페이스 설계과정에서는 앞서 말한 개념적 설계의 기능적 모델링과 네비게이션 설계의 행위자 설계를 기반으로 실제 행위자를 구현할 수 있는 가상코드(Pseudocode)를 작성한다. 실제 코드는 아니지만 행위자가 실제 알고리즘을 설계하는 과정이다.

유저 인터페이스 설계과정은 유저가 업무를 위하여 사용하는 화면을 디자인하는 것이다. 각 화면은 네비게이션 설계의 노드를 중심으로 설계되어 지고 관계형 설계와 행위자 설계를 함께 표현하게 된다.

시스템 인터페이스 설계과정은 시스템의 계층(Tier)을 결정하고 각 계층별 모듈의 위치를 설계하여 최종 시스템 구현시 그 모듈이 사용되어지는 위치를 정의함으로써 보다 안정적인 시스템을 구현하는데 사용되어 진다.

3.5 구현 (Implementation)

시스템 인터페이스 설계 과정에서 설계된 모형을 기반으로 데이터베이스 서버 및 웹 서버를 구축하고, 이러한 물리적인 환경을 기반으로 데이터베이스를 구축하며, 또한 유저 인터페이스에 따라 화면을 구성하고, 행위자 인터페이스 설계의 가상코드를 실제 언어로 구현하는 단계이다.

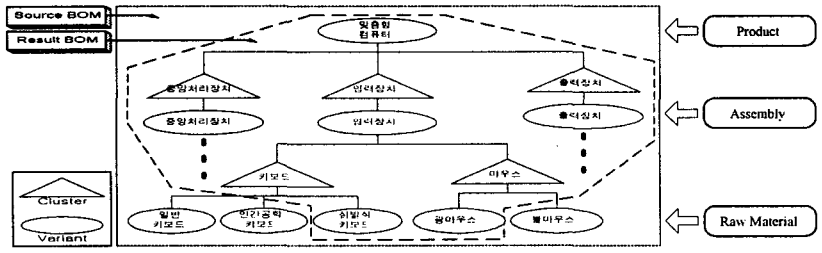
4. 웹 기반 Generic BOM 관리 시스템의 프로토타입 구현

웹 기반 Generic BOM 설계과정을 고객의 요구에 따라 변경이 많고 제품의 옵션에 대한 선택이 많은 제품인 맞춤형 컴퓨터를 생산하는 제조업체를 예로 들어 설명하고자 한다. <그림-11>은 맞춤형 컴퓨터에 대한 Generic BOM의 일부분을 나타낸다. 여기서 완제품인 맞춤형 컴퓨터는 Cluster에 해당하는 중앙처리장치, 입력장치, 출력장치로 구성되고, Cluster중 입력장치는 입력장치 Variant를 가지며 이것은 키보드와 마우스라는 Cluster로 다시 구성되어 있다. 마우스 Cluster는 볼마우스와 광마우스라는 Variant로 구성되며 실제 Result BOM을 구성할 때는 그 중 하나만이 선택되어진다. 이러한 방법에 의해 선택된 Variant들의 모임이 바로 Result BOM을 의미한다.

4.1 도메인 분석

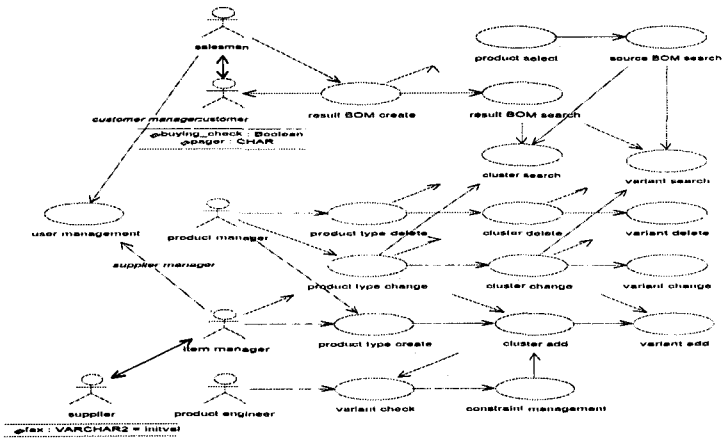
본 절에서는 제품 기본 정보와 구조적인 정보를 수집하고, Use Case 모형을 통해 WGBMS의 업무를 효과적으로 분석하는 과정을 서술하고자 한다.

본 연구에서 분석 대상이 되는 맞춤형 컴퓨터의 제품 특성은 제품 변형의 변화 주기가 거의 주 단위로 이루어지고 부품의 사양변경이 많고 새로운 부품이 많이 도입된다. 또한 부품 및 제품의 가격정보는 거의 하루 단위로 변화하게 된다. 이러한 변화에서도 고객의 요구사항에 맞는 제품을 선택하고 그 제품의 상세정보, 가격정보, 생산정보 등을 고객에게 신속하고 효과적으로 제공해야 한다. 고객의 요구에 따라 제품을 구성하는 가장 기본적인 업무를 살펴보면 다음과 같다. 고객이 제품 구입 의사를 밝히고 고객의 요구에 따라 구성하는 옵션 간의 충돌을 배제하면서 제품을 구성한다. 물론, 구성하는 부품들 간의 충돌은 배제되어야 한다.



<그림 11> 맞춤형 컴퓨터 제조업체를 위한 Generic BOM의 사례

또한, 구성된 제품을 수정하여 제품의 가격과 생산일정 정보를 알아보고 제품의 구입여부를 체크하게 된다.



<그림 12> WGBMS의 Use Case 설계

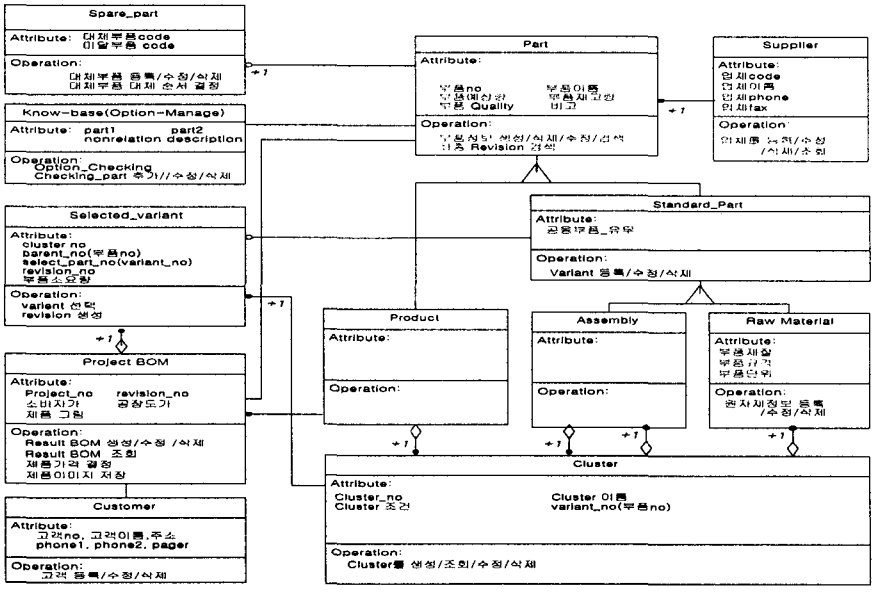
<그림-12>는 WGBMS의 기본업무를 Use Case 모형을 통하여 설계한 것이다. WGBMS의

Use Case 모형은 크게 3부분으로 나누어 진다. 고객 및 사원을 관리하는 유저 관리 부분과 사원들에 의해 Generic BOM 정보를 수정 및 삭제할 수 있는 업무 부분과 고객에 의해 원하는 제품을 선택하는 과정에 해당하는 부분으로 나누어 진다. 각 업무는 그 업무 또는 작업의 주체가 되는 Actor에 의해 수행되고 그 업무는 Use Case로 표현되어진다. 그리고 각 업무 간의 연관관계는 Relationship에 의해 업무의 흐름을 파악할 수 있다.

4.2 개념 설계

3.2절에서 서술한 객체 모델링(Object Modeling)은 시스템의 데이터 구조를 나타내며 시스템을 객체 관점에서 실세계를 정확하게 표현할 수 있게 하여 준다. <그림-13>은 OMT의 표현기법을 사용하여 객체 모형 중 Source BOM을 모델링한 것이다[9].

구조적인 측면에서 Generic BOM은 Source BOM과 Result BOM으로 나누어진다. <그림-13>을 살펴보면, Generic BOM에서는 완제품, 조립품, 원자재 간의 관계를 Cluster를 통하여 맺는다. 여기서 Cluster의 구성요소가 되는 조립품과 원자재는 Variant의 정보를 포함하게 된다. 그리고 Generalization을 이용하여 완제품, 조립품, 원자재의 각 특성에 맞게 정보를 중복없이 관리한다. Result BOM은 Source BOM으로부터 선택되어진 부품 및 관련 파리미터(Parameter) 정보를 관리하기 위한 "Select_ variant 클래스" 와 부품선택의 결과인 제품 관련 정보를 관리할 수 있는 "Project 클래스" 로 구성되어 있다. 또한Source BOM으로부터 Result BOM을 선택할 때 부품간의 충돌을 막기 위한 옵션 관리 클래스가 있다. 이 클래스는 옵션간의 충돌정보를 관리함으로써 Result BOM을 산출할 때 부품 간의 충돌을 막을 수 있다.

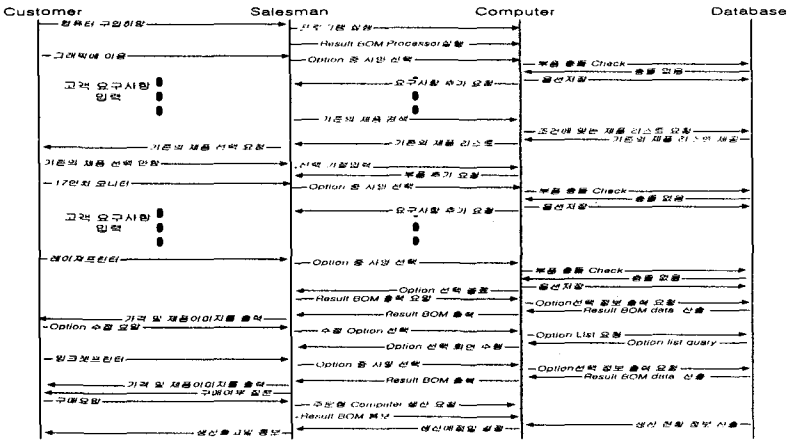


<그림 13> Generic BOM을 위한 객체 모형

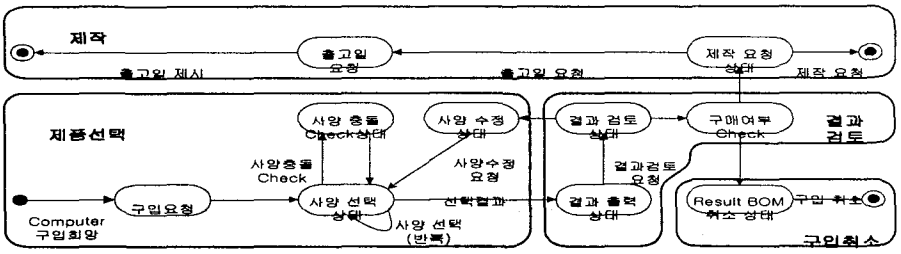
프로세스 측면에서 객체 모델링은 개체-관계 데이터 설계와는 달리 각 객체 내에 프로세스를 정의하여 객체 데이터와 객체 프로세스를 통합한 모형을 제시할 수 있다. 2.2절에서 서술한 프로세스들은 그와 관계된 클래스 내에 통합되어 각 클래스의 하단에 정의되어 진다. 주요

프로세스는 Source BOM 관리를 위한 부품 정보의 생성, 수정, 삭제하는 기능과 옵션정보를 관리하는 Cluster의 생성, 수정, 삭제하는 기능을 들 수 있다. Result BOM 관련 프로세스로는 Result BOM을 생성, 수정, 삭제하는 기능과 옵션을 선택할 때 부품 간의 충돌을 막는 옵션 체크 기능을 주요 프로세스라고 할 수 있다.

본 논문에서는 여러 업무 중 Source BOM의 Generic 제품군의 정보에서 최종 선택 제품의 Result BOM을 산출하는 업무를 사건추적도(Event Trace Diagram)와 상태도(State Diagram)를 이용하여 표현하고 있다.



<그림 14> Result BOM을 생성하기 위한 사건추적도

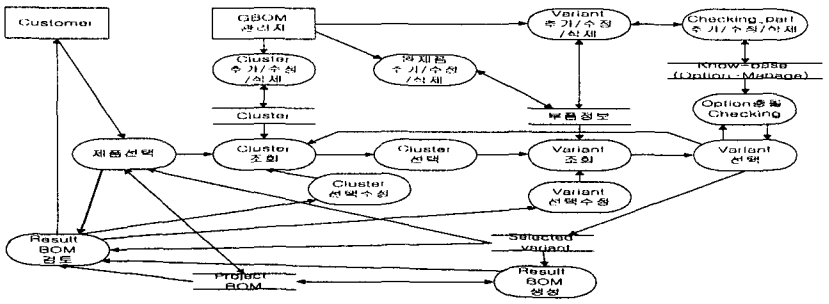


<그림 15> Result BOM을 생성하기 위한 상태도

맞춤형 컴퓨터의 기반의 업무 시나리오를 도식화한 <그림-14>의 사건추적도는 맞춤형 컴퓨터를 구입하려는 고객이 영업사원과 상담을 하고 맞춤형 컴퓨터를 위한 사양을 선택하여 결과를 산출하는 과정을 나타낸다. 컴퓨터 구입을 희망하는 고객이 방문하면, Result BOM 산출 프로그램을 수행하여 고객에게 컴퓨터 사양에 대한 질문 후 선택된 사양 정보를 저장한다. 이러한 질문을 반복하여 모든 사양의 선택을 마치면 고객이 그 결과를 검토하고 다른 사양이 있을 때에는 다시 요구에 맞게 수정한 후 맞춤형 컴퓨터에 대한 가격 및 제품 이미지를 산출하여 고객에게 제시하고 구매 여부를 체크한다. 고객이 구매를 원한다면 주문형 컴퓨터의 생산을 요청하고 생산예정일을 고객에게 통보한다. 이러한 사건 중심의 흐름에 대한 단계별 상태를 나타낸 것이 상태도이다.

상태도의 각 상태는 <그림-14>의 사건추적도의 흐름에 따라 서술되어지고 컴퓨터 구매에 관한 상태변이 과정을 보여준다. <그림-15>의 상태도는 제품선택, 결과검토, 제작 또는 구매취

소 상태로 나누어 진다.



<그림 16> WGBMS의 전반적인 자료흐름도

네 가지 상태변이 과정을 살펴보면 제품의 선택과정 후 선택 제품의 결과검토가 이루어지고 구매판단을 한다. 구매를 원한다면 제작상태가 되고 구매취소를 한다면 제작을 취소하고 Result BOM의 폐기하게 된다. 이러한 상태도는 시스템의 모델을 검증하고 고객과 엔지니어의 상호 이해를 촉진하는데 사용된다.

마지막으로 <그림-16>의 기능 모델링은 WGBMS의 전반적인 데이터 흐름과 주요 기능을 데이터의 흐름에 의해 보여주고 여기서 생성된 데이터가 어디에 저장하는 지를 자료흐름도를 통해 나타냈다. 이 모형에서는 크게 고객으로부터 주문을 입력받아 최종 제품의 Result BOM을 생성하는 기능과 BOM관리자가 Source BOM에 제품 옵션 및 사양의 추가/수정/삭제하는 기능 및 충돌 부품을 관리하는 기능이 있다. 고객의 주문을 받아 최종제품을 생성하는 기능을 보면 Actor인 고객에 의해 제품이 선택되어 지고 원하는 옵션의 사양을 선택하는 과정을 반복한 후 결정된 사양을 저장한 후 Result BOM을 생성하고 결과검토 과정을 거친다. Source BOM 관리기능은 각 옵션에 해당하는 Cluster와 그 사양인 Variant를 관리하는 것이다. 또한, 각 부품 간의 충돌을 막기 위한 옵션 충돌 체크 기능이 있다.

4.3 네비게이션 설계

네비게이션 설계는 앞에서 서술한 것과 같이 노드 설계, 관계 설계, 행위자 설계 그리고 공통 행위자를 추출하는 과정으로 진행되어 진다. 이러한 과정은 웹 설계를 보다 현실적으로 반영할 수 있다. <그림-10>은 네비게이션 설계 중 한 노드의 설계를 나타낸 것이다. 이 노드는 고객의 요구사항을 입력받고 입력된 부품 간의 충돌여부를 체크해 주는 노드이다. 그리고, 웹 시스템에 적용하기 위해 필요한 URL, 제목, 프레임, 웹 타입 등의 정보가 정의되어 있고, 이 노드에서 필요한 데이터와 다른 노드간의 관계를 나타내는 앵커(Anchor)와 타겟(Target) 그리고 그 전달인자들이 정의되어 있다. 행위자의 경우엔 행위자의 수행 권한 등이 정의된다.

WGBMS를 위한 네비게이션 설계는 Use Case에서 분류한 바와 같이 크게 3부분으로 구성되고 각각의 흐름은 Relationship을 통해 나타난다.

<그림-17>은 WGBMS의 메인 메뉴에 관련된 네비게이션 스키마 다이어그램을 나타낸 것이다.

<그림 17> 메인 메뉴를 위한 네비게이션널 스키마 다이어그램

각 노드 간의 관계는 고객 주문 업무의 흐름과 동일하게 구성되어 있고, 웹의 특성과 같이 특정 업무에 노드가 종속되지 않고 고객 추가 관련 노드와 같이 공유 노드들이 존재한다. 각 노드 내에서 실제 업무를 수행하는 행위가 또한 공통 모듈의 수집을 통해 프로세스의 중복을 막을 수 있다. 각 노드의 가장 마지막 노드는 그 업무의 수행 정보를 보여주고, 최초의 노드로 업무의 수행결과를 전달해 준다.

4.4 인터페이스 설계

데이터베이스 변환 과정에서는 4.2절에서 작성된 객체 모형을 객체-관계형 데이터베이스의 스키마로 설계하는 단계이다. 이것은 BOM의 데이터 정보를 객체로 관리하고, 프로세스를 함수로 관리하여 클라이언트의 성능을 향상시키며 옵션 정보를 용이하게 관리할 수 있게 한다. 또한 이미지 정보를 포함한 복합 데이터 및 사용자 정의 데이터를 효과적으로 관리하게 한다. 본 절에서는 3.2.4절에서 제시한 객체 모형을 객체-관계형 데이터베이스로 구축하기 위한 방법에 의해 데이터베이스를 구축하려 한다. <표-2>는 이러한 방법에 따라 객체 모형을 객체-관계형 데이터베이스의 스키마를 나타내는 데이터 정의 언어(DDL : Data Definition Language)의 한 부분이다. 본 논문에서는 SQL3 표준을 따르는 객체-관계형 데이터베이스인 ORACLE8[24]을 사용하여 Generic BOM을 위한 데이터베이스를 구축하고자 한다.

```
CREATE TYPE cluster_n AS OBJECT (
    c_cluster REF cluster_t IS cluster);
CREATE TYPE part_t AS OBJECT  (
    p_code          VARCHAR2(7),
    p_name          VARCHAR2(30),
    p_quantity      number,
    p_invent        number,
    p_supplier      REF supplier_t IS supplier,
    p_memo          VARCHAR2(50),
    v_common        char(1),
    r_quality       VARCHAR2(10),
    r_size VARCHAR2(10),
    cluster         cluster_n )
    NESTED TABLE cluster STORE AS cluster_n;
CREATE TABLE part of part_t ( p_code PRIMARY KEY
);
```

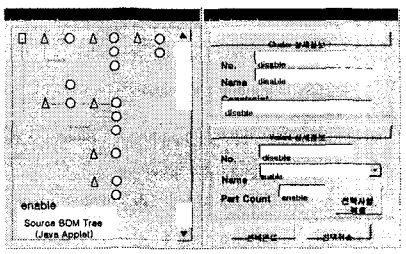
```
CREATE TYPE variant_n AS OBJECT (
    v_variant REF part_t IS part);
CREATE TYPE cluster_t AS OBJECT  (
    c_code          VARCHAR2(5),
    c_name          VARCHAR2(30),
    c_const         VARCHAR2(50),
    variant         variant_n )
    NESTED TABLE variant STORE AS variant_n;
CREATE TABLE cluster of cluster_t (
    c_code PRIMARY KEY  );
CREATE TYPE supplier_t AS OBJECT  (
    s_code          VARCHAR2(4),
    s_name          VARCHAR2(30),
    s_phone         VARCHAR2(12) )
CREATE TABLE supplier of supplier_t (s_code PRIMARY
KEY);
```

<표 2> Source BOM을 생성하기 위한 DDL의 예

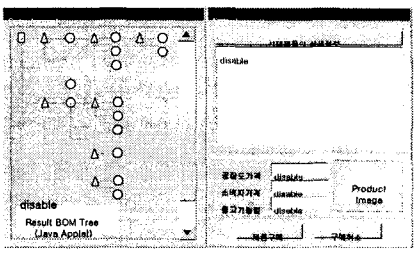
```
Main { //User check behavior
in_userid <- input
 in_passwd <- input
 while(TRUE) {
   q_passwd <- query1(userid, passwd)
   if(in_passwd == q_passwd)    break;
   else {
          print( Error : Again User ID and Password)
          in_userid <- input
          in_passwd <- input
       }
 }
 print(Current User Check)
 return True
}
```

<표 3> User Check Behavior

행위자 인터페이스 단계에서는 개념적 모형과 네비게이셔널 모형의 행위자의 수행 업무를 가상코드를 통해 나타낸다. 네비게이션 설계 단계에서 공통 모듈화되거나, 각 노드 내의 행위자의 수행 알고리즘을 표현한 것이다. <표-3>은 등록 유저 유무를 체크하는 가상코드를 나타낸 것으로 유저의 아이디와 패스워드를 입력 받고, 데이터베이스를 통해 등록된 유저인지 비교를 한 후 그 결과를 출력하는 알고리즘을 표현한 것이다.



<그림 18> WGBMS의 고객요구사양 입력 화면 설계

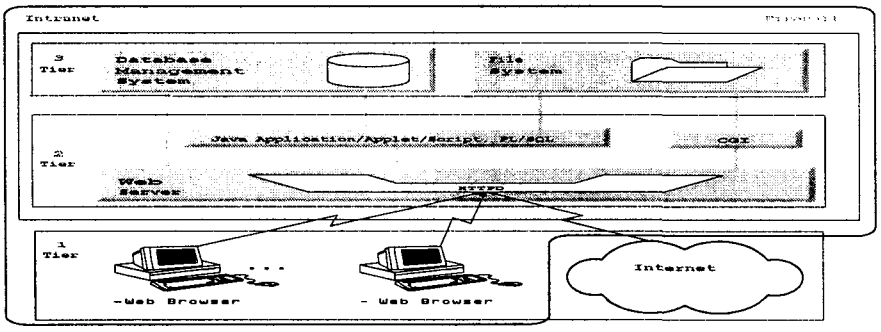


<그림 19> WGBMS의 Result BOM (최종제품) 생성 화면 설계

유저 인터페이스 설계 과정에서는 실제 유저가 사용하는 화면을 구성하는 과정이다. <그림-18>는 WGBMS 중 고객 요구 사항을 입력하는 화면으로 Source BOM을 기반으로 각 옵션인 Cluster의 사양인 Variant를 선택하여 Result BOM을 생성하는 업무를 위한 화면 설계이다.

<그림-19>은 선택된 사양을 기반으로 생성된 최종제품의 정보, 이미지, 가격, 출고가능일등을 보여주고 고객에게 구매 여부를 선택할 수 있는 화면을 설계한 것이다. 이 절에서 설계된 화면이 최종 구현 단계의 프로그램의 기본이 될 것이다.

시스템 인터페이스 설계 과정은 물리적 시스템의 구성과 소프트웨어의 구조를 나타낸 것으로 WGBMS는 인트라넷/인터넷 시스템을 구성하기 위해 3단계(Tire) 환경의 시스템이 구축되어진다. 3단계 환경은 클라이언트(Client), 어플리케이션 서버(Application Server), 데이터베이스 서버(Database Server)로 구성되어진다. 클라이언트 단계에는 웹 브라우저(Web Browser)에 의해 실제 유저가 사용하는 화면을 보여주는 단계이고, 어플리케이션 서버 단계는 웹 환경을 위한 웹서버와 웹서버를 기반으로 구동되는 자바 어플리케이션, PL/SQL 프로그램 그리고 파일 시스템의 제어를 위해 CGI 프로그램이 위치한다.



[그림 20] System Interface Design

마지막으로 데이터베이스 서버가 있는 데이터를 관리하는 데이터베이스 관리 시스템(DBMS)과 파일 시스템이 위치한다. 이렇게 설계한 시스템 인터페이스 설계인 <그림-20>을 기반으로 실제 시스템이 구축되어 진다.

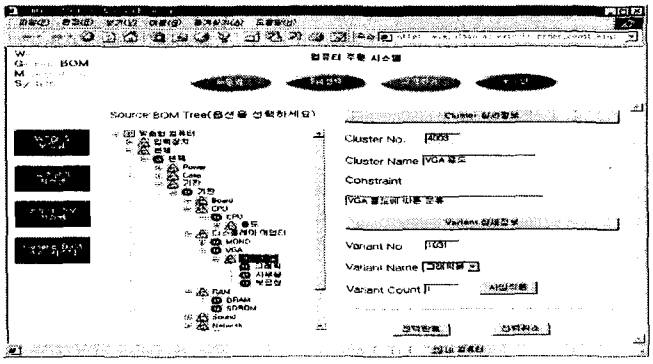
4.5 WGBMS구현

WGBMS는 객체-관계형 데이터베이스를 사용함으로써 Source BOM에 대한 옵션 및 그 사양의 추가/수정/삭제가 용이하고, 구조 및 부품정보를 쉽게 관리하여 준다. 또한 고객의 요구에 즉각 대응하여 Source BOM과 Result BOM를 효율적으로 생성할 수 있게 해준다. WGBMS의 구축 환경은 서버로는 Ultra SUN Sparc II의 Solaris 2.6에 객체-관계형 데이터베이스인 ORACLE8을 사용하고, 어플리케이션은 PL/SQL, Java Script, HTML, Java Applet으로 작성되었고 이러한 어플리케이션을 검색하는 도구로는 Netscape와 Explorer를 사용한다.

본 논문에서 구현한 맞춤형 컴퓨터의 WGBMS 기능은 2.2절에서 언급한 Generic BOM 관리 시스템의 필수적인 프로세스인 부품 정보관리, BOM 정보검색, Source BOM관리, Result BOM관리를 중심으로 구성되어 진다.

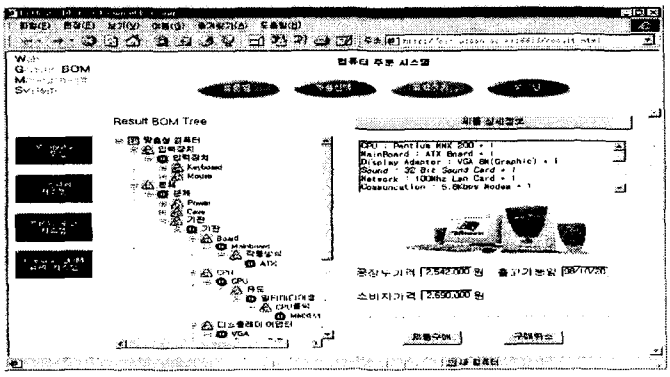
먼저 부품 정보관리는 맞춤형 컴퓨터를 구성하는 부품의 기본 정보관리 기능과 그 부품의 재고관리, 소요량관리, 이력관리, 대체부품 관리를 한다. 다음으로, BOM 정보검색에서는 부품의 상세 정보를 검색하고 일단/다단/집합 검색 기능을 통하여 컴퓨터를 구성하는 부품의 조립 수준을 알 수 있다. 또한, 각 옵션 및 사양의 상세 정보 검색이 가능하다. 세번째, Source BOM관리에서는 제품군을 생성하고 삭제하는 기능과 Source BOM을 수정하는 기능인 옵션(Cluster) 및 그 사양(Variant)의 추가/수정/삭제 기능이 있다. 또한 부품간의 충돌을 사전에 방

지하기 위해 지식기반(Knowledge-base)의 충돌정보를 관리할 수 있다. 마지막으로 Result BOM관리에서는 고객이 원하는 제품을 선택하고 검토하는 기능과 최종 선택된 제품의 가격을 산출하고 제품의 이미지 및 관련정보를 제공하는 기능이 있다.



<그림 21> WGBMS의 고객요구사양 입력 화면

<그림-21>는 Result BOM 관리 기능에서 사용자가 옵션의 사양을 선택하는 화면으로 Source BOM을 기반으로 Cluster의 Variant를 선택하고 저장하는 화면이다. <그림-21>의 좌측은 Source BOM을 구성하는 완제품(□), 조립품(△), 원자재(○)의 구조를 나타내고, 우측 상단은 부품의 옵션(Cluster) 정보를 제공하며, 우측 하단은 옵션 중 원하는 사양을 선택하는 부분이다.



<그림 22> WGBMS의 Result BOM(최종제품) 생성 화면

<그림-22>은 <그림-21>에서 선택된 사양을 기반으로 Result BOM을 생성하고 제품의 상세 정보를 고객에게 제시하며, 공장도 가격 및 소비자 가격과 출고 가능일 등의 기본정보를 제공하고 구입여부를 체크한다.

이렇게 구현된 WGBMS의 특징은 기존의 다른 BOM과 달리 Source BOM과 Result BOM의 관리 기능을 통하여 Generic 제품군의 정보를 효과적으로 관리할 수 있고, 옵션 및 그 사양의 추가/수정/삭제 기능을 통하여 제품의 설계 변경이 있을 때 빠르게 대응할 수 있으며, 제품의 이력정보 및 생성된 제품정보의 수정 및 삭제를 용이하게 한다. 그리고, 웹 기술을 기반으로 하여 제품 연구소에서 생성된 BOM 정보를 조립 현장에 적용이 가능하고 인터넷을 통해 보다 폭 넓은 고객층의 확보가 가능하다.

5. 결론

본 연구에서는 짧은 수명주기 및 다양한 변형을 갖는 제품을 생산하는 제조회사의 효과적인 BOM관리를 위하여 Generic BOM을 모델링하고, 이를 웹상에서 효과적으로 관리하기 위하여 웹 기반의 Generic BOM 관리 시스템(WGBMS : Web-based Generic BOM Management System)을 설계하였으며, 표준 부품들을 사용하여 다양한 선택사양을 갖는 맞춤형 컴퓨터를 생산하는 제조업체를 대상으로 프로토타입을 구현하였다.

구체적으로, WGBMS를 구현하기 위하여, 업무분석에는 객체지향 방법론인 OOSE 중 Use Case를 사용하여 업무를 파악하고, Generic BOM의 구조적인 측면과 기능적인 측면은 객체지향 방법론인 OMT를 사용하여 데이터 및 프로세스를 설계하였으며, 웹 환경에서 시스템의 효과적인 구축을 위해서는 네비게이션 설계방법을 도입 하였다. 또한, WGBMS 데이터의 효과적인 관리를 위해서 객체-관계형 데이터베이스인 ORACLE8을 사용하여 데이터베이스를 구축하였다.

향후의 연구계획은 개념적인 모델링 방법으로 OMG의 표준으로 대두되고 있는 UML(Unified Modeling Language)을 사용하여 분석 및 설계하고, 이 결과를 객체지향 데이터베이스를 사용하여 구현하는 것이다. 또한 연구범위를 설계, 문서관리, 그리고 PDM (Product Data Management) 및 ERP (Enterprise Resource Planning)의 BOM 모듈과의 통합 등 관련 분야로 확장하는 것도 흥미로울 것이다.

참고문헌

[1] 고석완, 김선호, "객체지향기법을 이용한 BOM관리 시스템 개발," 1997 추계 IE 학술대회 논문집, 인하대학교, 1997.

[2] 김유석, "객체지향 표준표기법을 알자," Microsoft Developer Journal, Vol.4, pp.10~15, 1998.

[3] 김윤수, "Oracle을 이용한 GDMO MIB의 표현," Oracle Open World 98, Oracle, 1998.

[4] 김정기, 김영호, 강석호, "Web-based BOM," 1997 춘계 IE/MS 공동학술대회 논문집, 포항공과대학교, pp.401~404, 1997.

[5] 류병우, 김정훈, 김상균, 김윤수, 최익성, 김준섭, 김윤하, 오원식, 박희원, "웹 기반의 객체 모델 관리 시스템 개발," 98한국 CAD/CAM학회 학술발표회 논문집, pp.466~471, 1998.

[6] 문희석, 김선호, "Group Technology를 이용한 설계정보관리 시스템의 개발," 한국정밀공학회지, 14권, 1호, pp.58~68, 1997.

[7] 문희석, 김선호, 신용하, "동시공학적인 도면정보관리시스템 개발," 산업공학, 9권, 1호, pp.41~52, 1996.

[8] 오태훈, 김정률, 김선호, "객체지향형 설계정보관리시스템 모델링에 관한 연구," http://www.sws.co.kr/support/tech/shk.htm, 1997.

[9] 이동국, 김재균, 장길상, "객체지향기법을 이용한 Generic BOM 관리 시스템 (GBMS)의 설계 및 구현", 산업공학, 12권,1호,pp.102-113, 1999.

[10] 이 한표, 이 춘열, 이 국철, "Family BOM 데이타베이스 구조에 대한 대안 : 목적별 BOM 연결구조의 간접 표현 방법," 1995추계 IE 학술대회 발표 논문집, 경희대학교, 1995.

[11] 장성우, "객체-관계형 데이터베이스:최근의 데이터베이스 관련기술의 개요," 경영과 컴퓨터, 1998.

[12] 정남기, 유철수, CALS시대의 생산관리, 청문각, 1997.

[13] Chung,Y., and Fischer,G.W., "A Conceptual Structure and Issues for an Object-Oriented Bill of Materials(BOM) Data Model," Production Planning & Control, Vol.3, No.3, pp.314~326, 1992.

[14] Clement, J., Coldrick, A., and Sari, J., Manufacturing Data Structuries, Oliver Wight Publications, 1992.

[15] Derr, K.W., Applying OMT (A Practical Step-by-Step Guide to Using the Object Modeling Technique), Sigs Books, New York, 1995.

[16] F.Garzotto, D.Schwabe and P.Paolini, "HDM- A Model Based Approach to Hypermedia Application Design", ACM Transaction on Information Systems, Vol.11, #1, Jan. 1993, pp.1-26

[17] Heeseok Lee, Choongseok lee, "A Scenario-Based Object-Oriented Methodology for Developing Hypermedia Information Systems, KAIST, 1995

[18] Heeseok Lee, Jongho Kim, "A View Based Methodology for Design Hypermedia Applications, KAIST, 1995

[19] Hegge, H.M.H., and Wortmann, J.C., "Generic Bill-Of-Material : A New Product Model," International Journal of Production Economics, Vol.23, pp.117~128, 1991.

[20] Hegge H.M.H., "A Generic Bill-Of-Material Processor Using Indirect Identification of Produccts," Production Planning & Control, Vol.3, No.3, pp.336~342, 1992.

[21] Kim, S.H., Cho, S.S., Kim, S.U., and Park, H.K., "Design and Implementation of Group Decision Support System using Object-Oriented Modeling Technique," Interface, Vol.10, No.1, pp. 169~187. 1997.

[22] Lee. H., and Suh,W., "A Workflow-Based Hypermedia Development Methodology," 한국 경영정보/전문가시스템 학회 '98 공동 춘계 학술대회, 동아대학교, 1998.

[23] Mather, H., Bills Of Materials, Dow Jones-Irwin, Homewood, 1987.

[24] Melnick, J., Oracle8 Server SQL Reference 8.0 Volume 2, ORACLE, 1997.

[25] Nandakumar, G., "The Design of a Bills Of Material Processor Using a Relational Database," Computers in Industry, Vol.6, pp.15~21, 1985.

[26] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W., Object-Oriented Modeling and Design, Prentice-Hall, New Jersey, 1991.

[27] Trappey, A.J.C., Peng, T.K., and Lin, H.D., "An Object-Oriented Bill-Of-Materials System for Dynamic Product Management," Journal of Intelligent Manufacturing, Vol.7, pp.365~371, 1996.