

"4+1 뷰"를 적용한 ERP 개념 모델 개발

허분애*, 정기원**, 이남용**

Developing a Conceptual ERP Model by using "4+1 View"

Boon-Ae Heo, Ki-Won Chong, Nam-Yong Lee

Abstract

Nowadays, many commercial ERP products, such as Oracle, SAP, and Baan, etc, are designed based on large-scaled companies. It is difficult for small and medium-size companies with weakness in budgets and resources(e.g., human, organization, technique, and so on) to use them as it was. So, new ERP system need to be provided for small and medium-size companies.

In this paper, we model and provide a conceptual ERP model for small and medium-size companies by using "4+1 view" architecture model of Unified Modeling Language(UML). The conceptual ERP model consists of five subsystems: Manufacturing, Sales, HumanResource and Payroll, Accounting, and Trading. Especially, we describe the conceptual ERP model focusing on "Manufacturing" subsystem by using several diagrams of UML.

By using the conceptual ERP model, the ERP system's developers of small and medium-size companies can obtain many benefits: improving the efficiency of software developing process and helping user requirements gathering and description of ERP system's nonfunctional aspect as well as functional aspect.

Key word : ERP(Enterprise Resource Planning), "4+1 view", UML

* 숭실대학교 대학원 석사과정(소프트웨어공학 전공)

** 숭실대학교 정보과학대학 컴퓨터학부 교수

1. 서론

국제화, 무한 경쟁 체제의 돌입 등 급변하는 경영 환경 속에서 기업이 살아 남기 위해 새로운 조직과 비즈니스 시스템을 구축하는 것은 중요하다. 게다가 기업의 규모가 커질수록 기존의 시스템은 변화하는 경영 환경에 유연하게 대처하지 못하며, 비즈니스와 비즈니스 시스템 사이의 부조화가 발생한다 [1]. 최근 들어 국제화된 경영 환경에 적응하고 생존 경쟁에서 이기기 위해 기업의 핵심 비즈니스 시스템으로 전사적 자원 관리(Enterprise Resource Planning, ERP) 시스템을 도입하려는 기업들이 늘어나고 있다 [1].

1.1 ERP의 정의

ERP란 무엇인가? 지금까지 ERP에 대한 많은 정의가 있었다. 우선, 미국 생산관리 재고협회에서는 ERP란 기존의 MRP II (Materials Requirement Planning II) 시스템과는 다르고, 그래픽 사용자 인터페이스, 관계형 데이터베이스, 개방형 시스템, CASE 도구, 클라이언트/서버 아키텍처등의 최신 정보 기술을 채용하고 있으며, 고객주문의 수주로부터 제조, 출하, 그리고 회계 처리 등에 필요한 전사적 차원의 효율적인 자원 관리와 계획을 위한 회계기반 정보시스템이라고 정의하고 있다. 가트너 그룹의 정의에 따르면, ERP란 기존의 MRP II 시스템을 능가하고 정보 기술면에서 우수한 차세대 생산관리 시스템을 지칭하는 것으로 기술하고 있다. 다시 말해서, ERP란 기업의 생산, 물

류, 회계, 인사 등의 전반적인 기업활동을 지원하여 전사적인 비즈니스 프로세스와 기능을 통합하려는 목적의 통합 패키지 소프트웨어 솔루션이다.

1.2 ERP의 역사

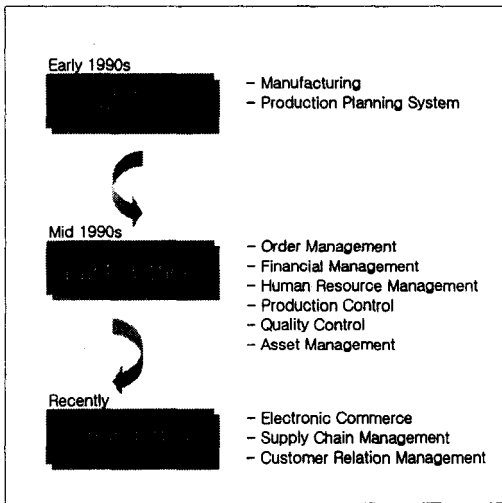
무한 경쟁체제의 치열한 생존경쟁에서 살아 남기 위해서 기업은 정보 기술 분야에서 경쟁력을 갖추어야 한다. 이에 따라, 1970년대 기업의 원활한 자재관리 및 구매 등 생산관리의 효율성 향상을 위해서 자재 소요량 계획(Material Requirements Planning, MRP) 개념이 제안되고 1980년대 MRP II로 발전되었다. ERP 시스템은 MRP II의 후계자로서 1990년대 초반에 등장하였다 [12].

ERP 시스템의 등장에서 알 수 있듯이, ERP 시스템은 생산관리 및 생산계획에 기원을 두고 있다. 그러나, 1990년대 중반에 들어서면서 ERP 시스템의 범위가 제조관리, 품질관리, 주문관리, 자산관리, 그리고 인적자원관리 등의 "백-오피스"(Back-Office) 기능으로 확대되었다 [12]. 최근에는 전자상거래(EC), 공급망관리(SCM), 그리고 고객관계관리(CRM) 등의 "프론트-오피스"(Front-Office) 기능으로 적용 범위가 더욱 확대되었다 [12]. 다시 말해서, ERP 시스템은 기업 활동 범위 전체에 걸쳐 다양한 기능을 지원하며 <그림 1>, 현재 비즈니스 시스템의 중추적인(Backbone) 역할을 수행한다 [21].

1.3 새로운 ERP 시스템을 필요성

1990년 이후로 많은 기업들이 비즈니스

프로세스를 향상시키고 기존의 비즈니스 시스템을 대체하기 위해서 ERP 시스템을 구축했다 [12]. 그러나 모든 기업들이 ERP 시스템의 도입 목적을 달성한 것은 아니다. 이유는 ERP 시스템의 크기가 매우 크며, 구축 과정이 매우 복잡하고 많은 단계의 작업들로 구성되어 있어서 구축이 어렵기 때문이다. 예를 들어, 지난 5년 동안 ERP 시스템을 적용을 시도했던 대기업 중 절반 이상의 기업들이 상당한 시간 지연과 예산 초과로 어려움을 겪었다 [2].



<그림 1> ERP 기능의 범위

많은 대기업들은 ERP 구축을 위해 Oracle [22], SAP [4], 그리고 Baan [24] 등과 같은 외국의 상용 ERP 패키지 제품을 사용했다. 이러한 상용 제품들은 대기업의 요구사항과 외국의 표준 업무 프로세스를 근거하여 구현되었으며, 다양한 기능을 지원하기 위해서 많은 모듈들로 구성되어 제품의

크기가 크고 구조가 복잡하다. 이로 인해 상용 제품을 중소기업의 ERP 구축에 그대로 적용하는데 많은 위험이 따른다.

이러한 상용 제품의 높은 구축비용과 복잡한 제품구조는 자본과 자원 즉, 인적자원, 조직, 기술 수준이 취약한 중소기업에 그대로 적용하는 것은 부적합하다 [25]. 예를 들어, ERP 시스템 도입 시 고려해야 할 중소기업의 특징에 대해서 T.J Elliott는 "중소 규모의 기업들은 작고 심플한 정보 기술 부서를 가지고 있으며, ERP 시스템과 같은 대형의 소프트웨어 구축 프로젝트를 수행한 경험이 적다." 라고 말한다 [23]. 또한 외국 기업과 상이한 기업 문화와 어음법, 무역법, 그리고 관세법 등의 독특한 국내의 경제 정책은 상용 제품의 기능 변경이나 추가를 필요로 하지만, 대부분의 상용 제품들은 커스터마이징(Customize)을 제외한 만족할만한 해결책을 제시하지 못하고 있다. 따라서, 중소기업에 ERP 시스템을 구축할 때 대기업과 다른 접근 방식이 필요하다. 그리고 중소기업은 효율적이고 경제적으로 ERP 시스템을 구축하길 원한다. 결론적으로 제안하는 ERP 개념 모델은 이러한 중소기업의 특성과 요구사항을 만족시켜야 한다.

본 논문에서는 UML(Unified Modeling Language)의 "4+1 뷰" 아키텍처 모델을 적용한 ERP 개념 모델을 제안한다. 이 ERP 개념 모델은 생산관리, 영업관리, 인사/급여관리, 무역관리, 그리고 회계관리 등 다섯 가지 서브시스템으로 구성되어 있으며, 특히 "백-오피스" 기능의 핵심인 "생산관리" 서브시스템을 중심으로 모델링을 수행했다.

2. 관련 연구

1960년대가 되자 소프트웨어 시스템의 크기가 커지고 복잡해짐에 따라 소프트웨어 위기론이 등장했다. 이를 해결하기 위한 방법으로 발전된 소프트웨어 공학 기술이 객체 기반 기술(Object-Oriented Technology)이다 [10]. 최근 들어, 기업의 비즈니스 시스템을 구축할 때 새로운 정보 기술의 적용이 활발해지면서 객체 지향 기술의 적용이 보편화되고 있으며, 그 중 객체 지향 모델링(Object-Oriented Modeling) 기법이 소프트웨어 개발과정의 분석 단계에서 많이 사용되고 있다.

객체 지향 모델링은 시스템을 분석할 때 실세계나 비즈니스를 객체로 인식하며 [3], 비즈니스 프로세스 모델링에 사용되는 최선의 방법이라고 할 수 있다 [11]. 대표적인 객체 지향 모델링 기법에는 Rumbaugh가 제안한 OMT(Object Modeling Technique) 방법이 있으며, 이것은 시스템을 객체 모델, 기능 모델 및 동적 모델 등을 사용하여 모델링한다.

모델이란 시스템을 기술하기 위해 사용하는 추상화 방법으로 시스템을 가시화, 명세화, 문서화, 그리고 구축하는 추상 개념들을 분할한 것이다 [8]. ERP 시스템처럼 복잡한 시스템의 모델링 작업은 여러 모델들을 사용해서 다양한 관점을 표현하기 때문에 많은 노력과 시간을 필요로 한다. 이론적으로, 전체 시스템은 단일의 추상화 수준, 즉 하나의 다이어그램(Diagram)을 사용해서 표현할 수 있다. 전체 시스템을 하나의 다이어그램으로 표현하면 한 눈에 전체 시스템

의 구조를 파악하는 것이 가능하기 때문에 시스템 이해에 많은 도움이 되며, 개발자와 사용자 사이의 의사소통이 원활해진다. 그러나 현실적으로 전체 시스템을 하나의 다이어그램으로 표현하는 것은 어렵다. 특히, ERP 시스템처럼 크고 복잡한 비즈니스 시스템 경우는 더욱 불가능하다 [20].

객체 지향 모델링 기법의 뷰(View) 개념을 사용하면 ERP 시스템을 기능적, 비 기능적 그리고 조직적인 측면 등 다양한 관점에서 기술할 수 있다. 뷰란 시스템 모델에 대한 조직과 구조의 투영으로, 각각의 뷰는 개발하는 시스템을 바라보는 이해 당사자들의 관점을 나타낸다. UML에서는 이러한 뷰를 기술하기 위해서 사용사례도(Use case Diagram), 클래스도(Class Diagram), 순차도(Sequence Diagram), 배치도(Deployment Diagram), 활동도(Activity Diagram) 등 시스템의 특정 측면을 강조하는 다이어그램을 다양하게 사용한다 [20].

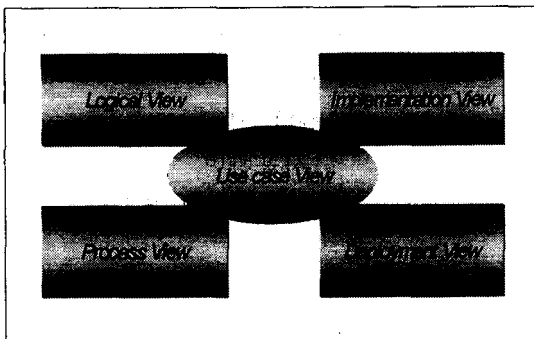
UML은 객체 지향 모델링을 위한 산업 표준으로, 객체 개념에 기초하여 소프트웨어, 시스템, 그리고 실세계의 모델을 표현하기 위해서 만들어진 모델링 언어이다 [13]. UML의 목적은 비즈니스 시스템에서 분산 처리 웹 어플리케이션 및 실시간 시스템까지 모든 타입의 시스템을 모델링하는 것으로 적용 범위가 매우 넓다 [11]. OMG(Object Management Group)의 표준에 의하면 UML은 표기법(Notation)과 메타 모델(Meta-Model)로 구성되어 있다 [16]. 표기법은 분석 및 설계에서 사용되는 도형식 구문이고, 이 표기법을 정의하는데 사용되는 것이 메타 모델이다. 그리고 UML은 표현력이 풍부

한 언어로 소프트웨어 시스템을 기술하는데 있어서 풍부한 표기법을 제공하며, 사용자에게 매우 친근한 표기법의 형태를 가지고 있다 [16].

3. 연구 프레임웍

여러 종류의 도면을 기반으로 건물을 짓는 것처럼 다양한 뷰를 사용하여 ERP 시스템을 기술한다. 각각의 뷰는 시스템을 바라보는 이해 당사자들의 관점을 나타내며, ERP 시스템의 중요한 특징을 좀 더 명확하게 기술하는데 사용된다. 다시 말해서, ERP 시스템에 대한 사용자 요구사항의 다양한 측면들을 나타낸다.

시스템 아키텍처는 전체 시스템의 구조를 나타내며, ERP 시스템을 바라보는 다양한 관점을 다루는데 사용되는 가장 중요한 산출물이다. UML과 RUP(Rational Unified Process)에 의하면, 시스템 아키텍처를 표현하는 여러 가지 관점들은 "4+1 뷰" 아키텍처 모델에 기반을 두고 있다 <그림 2>.



<그림 2> "4+1 뷰" 아키텍처 모델

UML의 "4+1 뷰" 아키텍처 모델은 사용자 사례 뷰(Use case View), 논리 뷰(Logical View), 프로세스 뷰(Process View), 구현뷰(Implementation View) 그리고 배치뷰(Deployment View) 등의 다섯 가지 뷰로 구성된다 [7, 20].

우선, ERP 시스템의 사용자 사례 뷰는 최종 사용자(User), 분석가, 테스트 담당자 등 이해 당사자의 관점에서 시스템의 행동을 설명하는 사용자 사례와 이들 사이의 관계를 표시하여 시스템 아키텍처를 구체화하는 요인들을 기술한다. 또한 사용자 사례는 ERP 시스템이 시스템의 외부 행위자(Actor)들에게 제공해야 하는 기능(Functionality)을 의미한다. UML에서 이 뷰의 정적인 관점은 사용자 사례도로, 동적인 관점은 활동도, 상태도(State Diagram)를 사용하여 표현한다.

논리뷰는 시스템의 기능적인 요구사항을 지원하며, 문제 영역과 그에 대한 해결책을 구성하는 클래스, 인터페이스, 그리고 이들 사이의 협력(Collaboration) 등을 포함한다. UML에서 이 뷰의 정적인 관점은 클래스도, 객체도(Object Diagram)로, 동적인 관점은 상태도, 활동도, 그리고 순차도를 사용하여 표현한다.

프로세스 뷰는 시스템 생산성, 성능, 신축성(Scalability) 등을 기술하며, 시스템의 동시성 및 동기화 메카니즘을 형성하는 스레드(Thread)와 프로세스들로 구성된다. UML에서 이 뷰는 논리뷰를 기술하는 다이어그램들을 사용하여 표현한다.

구현뷰는 시스템 릴리즈의 형상 관리(Configuration Management)를 나타내며, 물리적인 시스템을 어셈블하고 릴리즈하는데 사

용되는 컴포넌트와 파일 등을 포함한다. UML에서 이 뷰의 정적인 관점은 컴포넌트도(Component Diagram)로, 동적인 관점은 활동도, 교류도(Interaction Diagram)를 사용하여 표현한다.

마지막으로, 배치뷰는 물리적인 시스템을 구성하는 부분들에 대한 인도, 분산, 설치 등을 기술하며, 실행되는 시스템 하드웨어 형태(Topology)를 구성하는 노드들을 포함한다. UML에서 이 뷰의 정적인 관점은 배치도로, 동적인 관점은 활동도, 교류도를 사용하여 표현한다.

지금까지 "4+1 뷰" 아키텍처 모델의 다섯 가지 뷰에 대해서 간략하게 설명했다. 한가지 중요한 것은 시스템 아키텍처를 표현하는 각각의 뷰들이 완전히 독립적으로 수행되지 않음을 이해하는 것이다 [20]. 즉, 하나의 뷰에 속하는 요소들이 다른 뷰의 요소들과 연결될 수 있다. 예를 들어, 순차 도는 사용사례를 설명하고 있는 시나리오를 기반으로 하여 작성된다.

다음 장에서는 "4+1 뷰"를 적용하여 ERP 시스템을 개념 수준에서 UML의 다양한 다이어그램을 이용하여 모델링 한 결과를 제시한다. 제시한 ERP 개념 모델은 산학연 과제와 연구 결과로 본 논문에서는 ERP 시스템 중 "생산관리" 서브시스템에 초점을 맞춰 기술하기 때문에, 연구의 모든 결과와 구현을 포함하지 않는다.

4. ERP 개념 모델

한국 경제는 수출 의존적인 산업구조를 가지고 있기 때문에 제품을 생산하고, 생산

된 제품을 외국에 수출하는 활동들이 중요한 부분을 차지한다. 현재, 중소 규모의 제조업체들이 많이 치열한 생존 경쟁 속에서 열심히 활동하고 있다.

세계 경제의 개방화, 세계화 등의 경영 환경의 급격한 변화로 중소기업들은 국내 기업뿐만 아니라 외국의 많은 기업들과의 경쟁이 불가결하게 되었다. 게다가 이러한 환경 변화에 능동적으로 대처할 수 있는 유연한 비즈니스 프로세스와 시스템을 구축하고 있지 못한 것이 현실이다. 따라서 중소기업에 있어서 기존의 비즈니스 프로세스와 시스템을 대체하고, 급변하는 경영 환경에 적응하기 위해 ERP 시스템 도입의 필요성이 증대되고 있다.

일반적으로 ERP 시스템의 기능은 다섯 가지로 분류된다 [5]. 첫째, 혼합 형태의 제조(Hybrid Manufacturing) 기능을 지원하여 다양한 사용자 요구사항을 제어할 수 있다. 둘째, 자재 소요량 계획 및 다중 생산계획을 위한 시뮬레이션 기능을 제공한다. 이를 통해 자재 소요량 계획 및 다중 생산계획에 필요한 처리 시간이 단축된다. 셋째, ERP 시스템은 내부 및 외부 시스템, 기타 비즈니스 시스템, 그리고 데이터들과 고수준의 통합 기능을 제공한다. 넷째, ERP 시스템은 다른 운영 체제를 사용하며, 지리적으로 떨어진 위치의 공장에서 동시에 운용될 수 있다. 따라서 기업은 국내와 국외를 연결하는 글로벌한 비즈니스 조직의 구성이 가능하다. 마지막으로, 다양한 언어를 사용한 시스템 운영을 지원하여, 사용자는 추가적인 노력 없이 자신의 모국어로 시스템을 사용할 수 있다.

본 논문에서 제안하는 중소기업을 위한 ERP 개념 모델은 ERP 시스템을 생산관리, 영업관리, 인사/급여관리, 회계관리, 그리고 무역관리 등의 다섯 가지 서브시스템으로 구분한다. 이 모델은 위에서 언급한 다섯 가지 ERP 기본 기능과 중소기업의 특성을 기반으로 모델링 한 결과로 각각의 서브시스템의 기능은 다음과 같다.

첫째, 생산관리 서브시스템은 생산활동과 관련된 모든 정보를 다룬다. 즉, 품목관리, BOM관리(Bill Of Material), 재고관리, 구매관리, 생산계획관리, 제조관리, 외주가공관리, 설비관리, 그리고 품질관리 등에서 필요한 정보를 관리한다. 여기서 품목이란 부품, 반제품 그리고 완제품 등의 의미로 사용된다. 서브시스템의 기능에 대한 자세한 내용은 4.1절부터 4.5절에서 설명한다.

둘째, 영업관리 서브시스템은 고객주문 및 행동에 관계된 실적 및 분석 데이터 그리고 평가 결과 등 영업활동과 관련된 모든 정보를 다룬다. 즉, 제안관리, 주문관리, 계약관리, 선적관리, 반품관리, 그리고 고객관리 등에서 필요한 정보를 관리한다. 이 서브시스템의 업무 프로세스는 다음과 같다.

(1) 우선, 고객의 등록여부를 확인한다 (2) 자사의 제품 리스트를 제안하고, 고객으로부터 주문을 받은 후 주문에 따라 세부적으로 고객과 계약을 체결한다 (3) 계약 사항에 따라 제품을 선적한다 (4) 마지막으로 반품이 발생하면 반품 처리 프로세스를 수행한다. 추가적으로 첫 번째 단계에서 고객의 등록 확인이 안된 경우에는, 새로운 고객으로 간주하여 고객 정보를 시스템에 등록한다.

셋째, 인사/급여관리 서브시스템은 인적 자원의 효율적인 관리와 급여 및 정산의 일괄적인 관리를 지원한다. 이 서브시스템은 사원에 대한 기본정보 및 자격정보를 관리하는 인사정보관리, 채용 지원자에 관한 정보와 심사기준 정보를 관리하는 채용관리, 급여 및 상여를 계산, 관리하는 급여/상여관리, 회사주 정보와 동호회와 관련된 정보를 관리하는 복리후생관리, 그리고 퇴직 급여 및 연말정산을 관리하는 정산관리 등의 기능으로 구성된다.

넷째, 무역관리 서브시스템은 다른 상용 ERP 패키지 제품에서는 볼 수 없는 독특한 서브시스템이다. 이 시스템은 기업의 수출 및 수입 업무 프로세스를 효율적이고 일괄적으로 수행할 수 있도록 지원하며, 전자 정보 교환(Electronic Data Interchange, EDI)을 위한 표준문서 형식과 국제 사업 규칙과 관련된 무역 정보 등을 포함한다. 거래처에 관한 정보를 관리하는 Buyer관리, 제안 및 진행 상황을 관리하는 오퍼(Offer) 관리, 주문내역 및 진행 상황을 관리하는 주문관리, 물건 운송과 관련된 정보를 관리하는 선적관리, 통관 비용 및 세금 관련 정보를 관리하는 통관관리, 신용장 개설 및 어음지불을 관리하는 L/C(Letter of Credit) 관리, 그리고 물건 양도 후의 불만 사항에 대한 클레임관리 등의 기능으로 구성된다.

마지막으로, 회계관리 서브시스템은 기업의 회계업무를 효율적이고 유연하게 수행하기 위해 회계작업을 전산화하고, 통합 데이터베이스를 이용해 회계정보의 정확하고 신속한 처리 기능을 가진다. 이 서브시스템은 계정, 전표 및 결산을 관리하는 일반회

계관리, 세금을 처리하는 세무회계관리, 원가회계관리 및 의사결정지원을 위한 관리회계관리, 고정자산관리, 그리고 예적금관리 및 예산/자금의 쓰임을 기획하는 예산/자금관리 등의 기능으로 구성된다.

이제부터 4.1절부터 4.5절에 걸쳐 "4+1 뷰" 아키텍처를 적용한 "생산관리" 서비스 시스템의 모델링 결과를 기술한다.

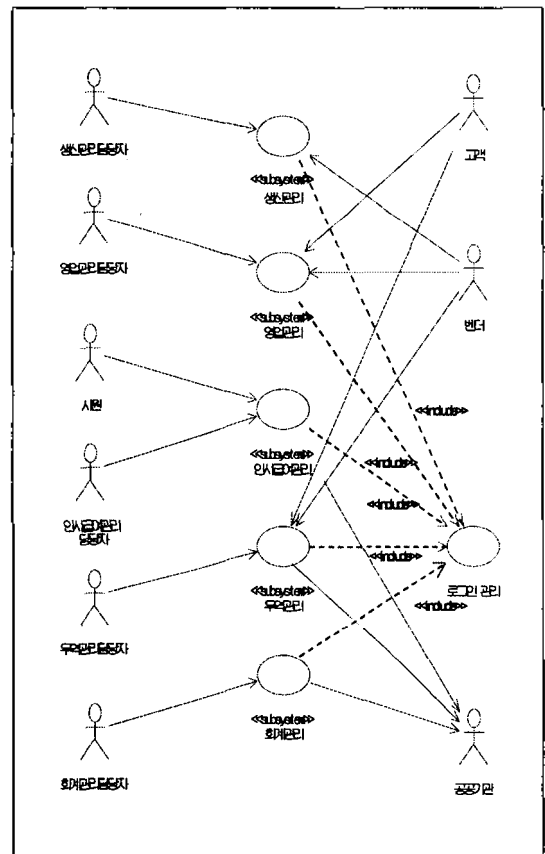
4.1 사용사례 뷰

사용사례 뷰의 정적인 측면을 나타내는 사용사례도는 ERP 시스템의 전체적인 개념(Context)을 나타내기 위해 사용자와 시스템 사이의 전형적인 상호 작용으로 ERP 시스템의 기능을 나타내는 사용사례, 행위자, 그리고 사용사례와 행위자 사이의 연관성을 표현하는 관계 등을 사용한다 [7, 20].

본 논문에서는 두 개의 사용사례도를 이용한다. 하나는 최상위 수준에서 전체 ERP 시스템을 다섯 가지 서브시스템으로 분리하여 사용사례로 표현했으며, 나머지 하나는 다섯 가지 서브시스템 중에서 "생산관리" 서브시스템의 기능을 중심으로 표현하였다. 1.2절에서 언급했듯이 ERP 시스템은 생산관리 및 생산계획 시스템에 기원을 두고 있으며, 본 논문의 목적이 중소 규모의 제조업체를 위한 ERP 개념 모델 개발이다. 따라서 "생산관리" 서브시스템을 중심으로 기술하여 ERP 개념 모델을 설명하고, 나머지 서브시스템들은 "생산관리" 서브시스템으로부터 확장될 수 있음을 가정한다.

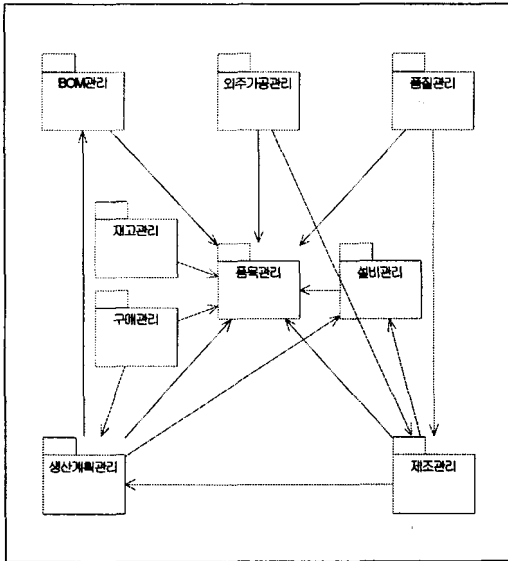
일반적으로 분해기법(Divide-and-Conquer)을 사용하여 크고 복잡한 대형의 시스템을

작은 단위의 서브시스템으로 나누면 전체 시스템의 구조를 이해하는데 도움이 되며, 관리가 용이해 진다 [19]. 따라서 제안하는 개념 모델에서는 매우 크고 복잡한 패키지인 ERP 시스템을 다섯 가지 서브시스템, 즉 생산관리, 영업관리, 인사/급여관리, 회계관리, 그리고 무역관리로 분리하였다 <그림 3>. <그림 3>의 사용사례도에서, 사용자의 로그인을 관리하는 사용사례를 제외한 나머지 사용사례들은 다섯 가지 서브시스템을 나타낸다. 그리고 각각의 사용사례와 관련된 행위자는 생산관리 담당자, 영업관리



<그림 3> 최상위 사용사례도

출하여 밀접한 관계의 클래스들을 클래스 패키지(Class Package)로 묶어 그룹화 하였다. 아홉 개의 클래스 패키지는 <그림 4>의 "생산관리" 서브시스템 사용사례도에서 나타난 각각의 사용사례와 연관성을 가진다.

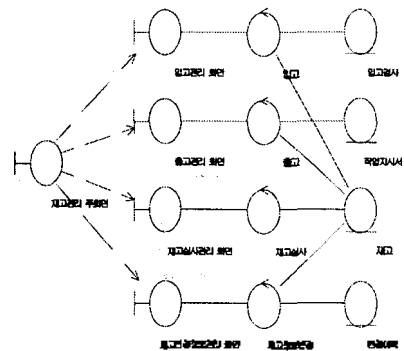


<그림 5> "생산관리" 서브시스템의 클래스도

<그림 5>에서는 클래스 패키지 사이의 관계를 나타내기 위하여 "usage" 의존관계를 사용했다. "usage" 의존관계는 두 개의 패키지를 구성하는 클래스들 사이에 적어도 하나의 "usage" 의존관계가 존재함을 의미한다. 예를 들어, "품목관리" 클래스 패키지가 변경되면, 이 클래스 패키지와 관계 있는 모든 패키지 즉, 품목관리, BOM관리, 재고관리, 구매관리, 생산계획관리, 제조관리, 외주가공관리, 설비관리, 그리고 품질관리 등이 영향을 받는다. 다시 말해서 "품목관리" 클래스 패키지와 다른 클래스 패키지들

사이에 "usage" 의존관계가 있다면, 품목 및 품목코드 등의 클래스로 구성되는 "품목관리" 클래스 패키지는 다른 클래스 패키지들이 필요로 하는 기초적인 정보를 제공한다. 그러나 두 개의 클래스 패키지를 구성하는 클래스들이 개별적으로 모두 "usage" 의존관계를 가지는 것은 아니다 [6].

특정적으로 <그림 5>의 클래스도에서는 ERP 개념 모델이 개념적인 수준에서의 모델링 결과이기 때문에 각각의 클래스 패키지를 구성하는 클래스와 이들 사이의 관계를 자세하게 표현하지 않았다. 그러나 "생산관리" 서브시스템의 논리뷰에 대한 이해를 향상시키기 위해 "재고관리" 패키지 클래스를 선택하여 클래스 패키지를 구성하는 클래스들과 이들 사이의 관계를 간략하게 기술한다<그림 6>.



<그림 6> "재고관리" 클래스 패키지의 클래스도

<그림 6>의 "재고관리" 클래스 패키지의 클래스도는 분석 단계에서 찾아낸 클래스를 경계 클래스(Boundary Class), 제어 클래스(Control Class), 그리고 개체 클래스(Entity

Class)로 분류하여 아이콘 형태로 표시했다. 그리고 의존관계와 연관관계(Association)를 사용하여 클래스 사이의 관계를 나타냈다.

그러나 <그림 6>은 속성이나 오퍼레이션 표시를 생략함으로써 클래스에 대한 자세한 설명을 제공하지 않는다. 왜냐하면 분석 단계와 같은 개념적 모델링 단계에서는 속성과 오퍼레이션 등 클래스 특성에 대한 자세한 기술은 중요하지 않기 때문이다. 클래스의 속성이나 오퍼레이션에 대한 기술은 설계 단계 및 구현 단계를 거치면서 완성된 형태를 갖추게 된다.

4.3 프로세스 뷰(Process View)

프로세스 뷰의 동적인 측면을 나타내는 순차도는 객체와 객체의 상호작용이 어떻게 이루어지는가를 기술한다. 즉, ERP 시스템을 구성하는 객체들 사이에서 메시지들의 전달 방법(Send & Receive)을 표현한다 [7, 15]. 순차도는 사용사례에 대한 설명인 시나리오를 근간으로 작성된다. 하나의 사용사례는 하나 이상의 시나리오를 가질 수 있으며, 하나의 시나리오는 하나의 순차도로 표현된다. 따라서, 하나의 사용사례는 많은 순차도로 표현될 수 있다.

"생산관리" 서브시스템의 순차도를 작성하기 위해서, 먼저 "재고관리"를 선택하여 이 사용사례와 관련된 네 개의 시나리오를 작성하였다 <표 1, 2, 3, 4>. 시나리오의 구성은 시나리오 이름(Scenario Name), 간략 설명(Brief Description), 그리고 업무 흐름(Flow of Event)으로 이루어져 있다. 각 시나리오의 이름은 재고실사관리, 입고관리, 출

고관리, 그리고 재고변경정보관리이며, 업무 흐름은 시나리오와 관련된 업무의 핵심 프로세스를 간략히 기술한다. 즉, 네 개의 시나리오는 2차 시나리오를 작성하지 않고 기본 시나리오만 기술했다.

그리고 작성된 시나리오의 업무 흐름을 바탕으로 순차도를 작성했다 <그림 7, 8, 9, 10, 11>. <그림 7>의 순차도는 "재고관리" 사용사례의 네 가지 시나리오를 전체적인 관점에서 표현한 것이다. 이 순차도에서는 "notes" 표기법을 사용하여 각각의 "notes"가 순차도와 직접 연결되도록 한다. 예를 들어, "notes"를 클릭하면 "notes" 안에 쓰여진 이름에 해당하는 순차도로 분기된다. 그리고 객체 사이의 메시지에 표시된 숫자는 시나리오의 업무 흐름 순서에 상응한다.

<그림 8>에서 11의 순차도는 재고실사관리, 입고관리, 출고관리, 그리고 재고변경정보관리 시나리오와 상응하며, 각각의 순차도는 임의적으로 번호를 붙여 기본적인 업무처리 과정만 간략히 표현했으며 선택사항, 예외 및 확장 등 예외사항 처리(Exception Handling)는 표시하지 않았다. 업무에서 발생할 수 있는 예외 사항을 모두 표시해서 순차도를 작성하기 위해서는 기본 시나리오 뿐만 아니라 2차 시나리오 등을 이용하여 사용사례를 좀더 자세하게 설명해야 한다.

<표 1> 입고관리 시나리오

시나리오 이름 : 입고관리

간략 설명 :

생산관리 담당자는 품목의 입고를 처리하고 이와 관련된 정보를 관리하기 위해서 이 시스템을 사용한다.

업무 흐름 :

- 생산관리 담당자는 입고 리스트를 생성하기 위해서 "생성" 버튼을 누른다.
- 시스템은 새로운 입고 리스트를 생성하고, 입고 검사에서 통과된 결과를 로드하여 화면에 디스플레이 한다(A).
- 생산관리 담당자는 입고 처리할 품목을 선택하고 입고 품목 리스트를 생성한다.
- 생산관리 담당자는 날짜, 수량, 재고 위치 등의 입고관련 정보를 입력한다.
- 생산관리 담당자가 "확인" 버튼을 누르면 시스템은 입고 품목 리스트를 저장하면서 "재고" 객체를 입고 정보로 갱신한다.

주의 :

- (A)단계에서, 시스템이 입고 검사 기능을 포함하고 있지 않다면, "구매 요청서"와 "품목" 객체로부터 정보를 로드하여 입고 리스트에 디스플레이 한다.

<표 2> 출고관리 시나리오**시나리오 이름 :** 출고관리**간략 설명 :**

생산관리 담당자는 품목의 출고를 처리하고 이와 관련된 정보를 관리하기 위해서 이 시스템을 사용한다.

업무 흐름

- 생산관리 담당자는 출고 리스트를 생성하기 위해 "생성" 버튼을 누른다.
- 시스템은 새로운 출고 리스트를 생성하고, 작업 지시서의 정보를 로드하여 화면에 디스플레이 한다.
- 시스템은 작업 지시서의 품목들과 관련된 재고 정보를 로드하여 화면에 디스플레이 한다.

- 생산관리 담당자는 출고 처리할 품목을 선택하고 출고 품목 리스트를 생성한다.
- 생산관리 담당자는 날짜, 수량, 재고 위치 등의 출고관련 정보를 입력한다.
- 생산관리 담당자가 "확인" 버튼을 누르면 시스템은 출고 품목 리스트를 저장하면서 "재고" 객체를 출고 정보로 갱신한다

<표 3> 재고실사관리 시나리오**시나리오 이름 :** 재고실사관리**간략 설명 :**

생산관리 담당자는 품목의 재고실사를 처리하고 이와 관련된 정보를 관리하기 위해서 이 시스템을 사용한다.

업무 흐름 :

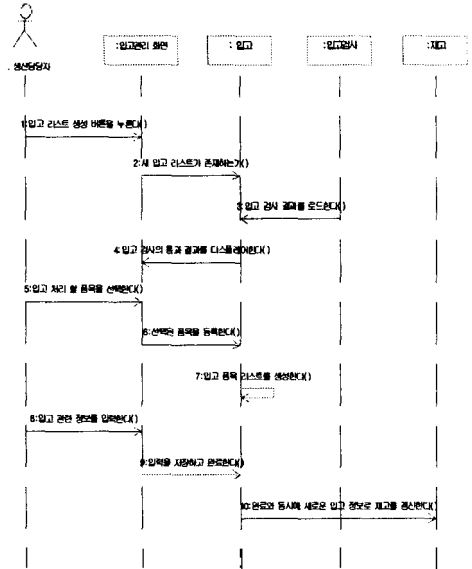
- 생산관리 담당자는 재고실사리스트를 생성하기 위해 "생성" 버튼을 누른다.
- 시스템은 재고실사 리스트를 생성하고, 재고 정보를 로드하여 디스플레이 한다.
- 생산관리 담당자는 재고실사 품목을 선택하고 재고실사 품목 리스트를 생성한다.
- 생산관리 담당자는 재고실사 결과를 입력한다.
- 생산관리 담당자가 "확인" 버튼을 누르면 시스템은 재고실사 품목 리스트를 저장하면서 "재고" 객체를 재고실사 정보로 갱신한다.

<표 4> 재고변경정보관리 시나리오**시나리오 이름 :** 재고변경정보관리**간략 설명 :**

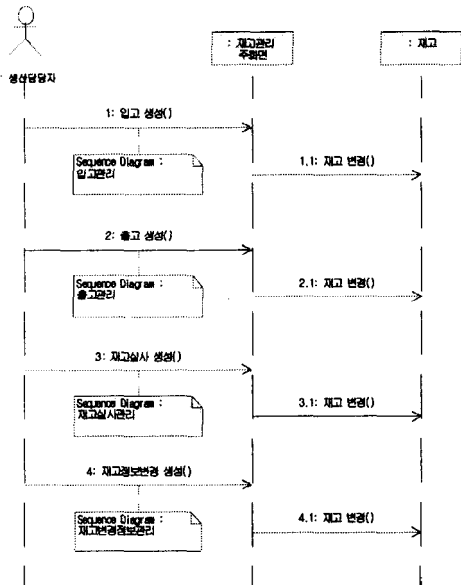
생산관리 담당자는 품목의 재고변경정보를 처리하고 이와 관련된 정보를 관리하기 위해서 이 시스템을 사용한다.

업무 흐름 :

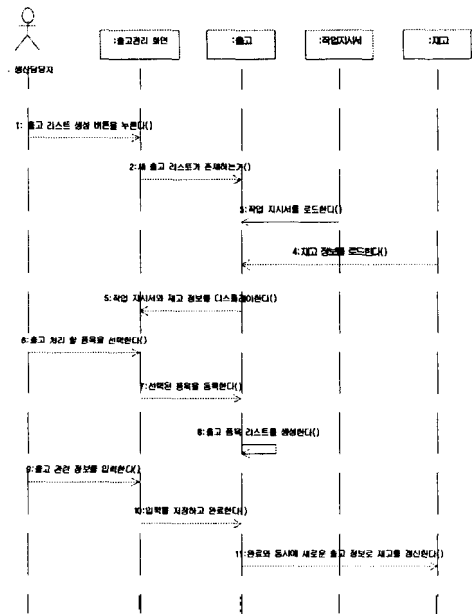
- 생산관리 담당자는 재고정보를 변경하기 위하여 "수정" 버튼을 누른다.
- 생산관리 담당자는 변경할 품목 코드를 선택한다.
- 생산관리 담당자는 선택된 품목 코드의 품목 정보를 "재고" 객체로부터 로드하여 디스플레이 한다.
- 생산관리 담당자는 새로운 정보를 입력한다.
- 생산관리 담당자가 "확인" 버튼을 누르면 시스템은 새로운 재고변경정보를 저장하면서 "재고" 객체를 재고변경정보로 갱신한다(A).
- (A)단계에서 재고변경정보의 이력 정보도 동시에 저장한다.



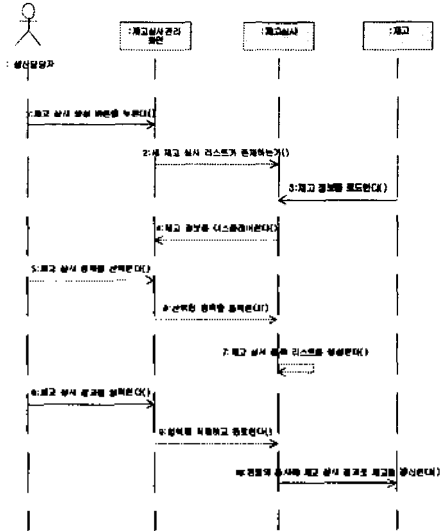
<그림 8> "입고관리" 순차도



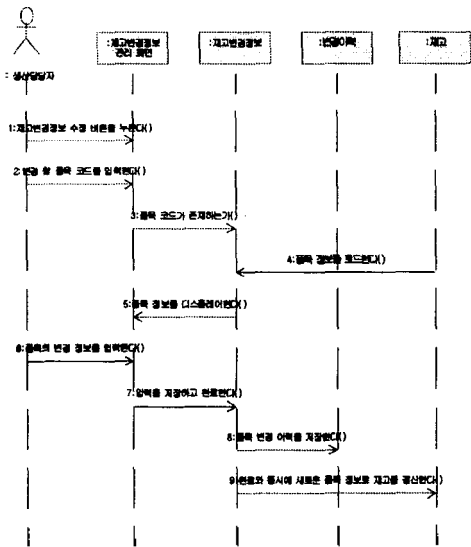
<그림 7> "재고관리" 순차도의 개요



<그림 9> "출고관리" 순차도



<그림 10> "재고실사관리" 순차도



<그림 11> "재고변경정보관리" 순차도

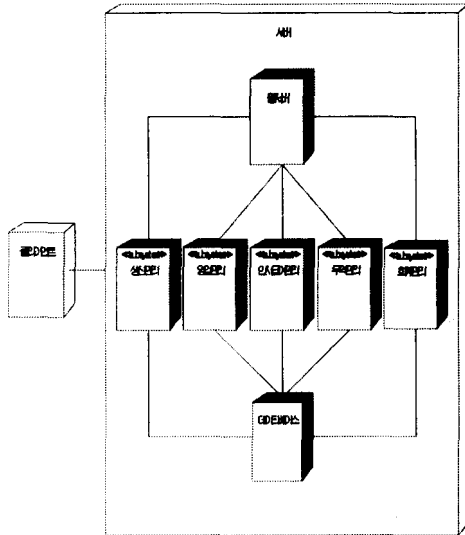
4.4 배치뷰

배치뷰의 정적인 측면을 나타내는 배치도는 ERP 시스템의 하드웨어 유닛(Unit) 구조와 각 유닛에서 수행되는 소프트웨어를 포함한 시스템 형태의 물리적 구조를 기술한다 [7, 15]. ERP 개념 모델에서는 적은 비용으로 고효율의 ERP 시스템을 구축하길 원하는 중소기업을 위하여 2계층 클라이언트/서버(2-Tiered Client/Server) 아키텍처를 ERP 시스템의 하드웨어 구조로 적용했다 <그림 12>.

<그림 12>의 배치도에서 ERP 시스템은 두 가지 노드로 구성되어 있다. 하나는 클라이언트 부분으로 사용자 작업 수행을 관리한다. 나머지 하나는 서버 부분으로 웹서버(Web Server), 데이터베이스(Database), 그리고 다섯 가지 서브시스템으로 구성된다. 데이터베이스는 서브시스템들이 공유하는 데이터를 통합 관리하는 부분이고, 웹서버는 클라이언트와 서브시스템 사이의 연결 작업을 담당하는 부분이다. 그러나 사용자는 클라이언트에서 웹서버 통하지 않고 직접적으로 서브시스템에 접근할 수도 있다.

그리고 다섯 가지 서브시스템은 실질적인 ERP 기능을 수행하는 부분으로, 파이프-앤-필터(Pipe and Filter) 아키텍처를 적용하여 서브시스템 사이의 관계를 표현했다. 파이프-앤-필터 아키텍처 패턴에서 각 부분들은 독립적으로 기능을 수행하며, 서브시스템들은 데이터에 대해서만 상호 의존성을 가진다. 따라서 다른 서브시스템에 영향을 주지 않고 사용자의 요구사항 변경에 따른 특정 서브시스템의 기능 변경이 가능하며, 시스

템 전체의 기능 향상을 위한 새로운 서브시스템으로의 대체나 추가가 용이하다.



<그림 12> ERP 개념 모델의 배치도

또한 클라이언트와 서버는 각각 물리적인 기계를 나타낸다. 이 중에서 서버는 웹 서버, 데이터베이스, 다섯 가지 서브시스템을 하나의 물리적인 기계에 위치시켜 구성하거나, 이들 각각을 물리적으로 분리시켜서 구성할 수도 있다. 그러나 제안한 ERP 개념 모델은 구축에 필요한 하드웨어 비용을 절감하기 위해서 웹서버, 데이터베이스, 다섯 가지 서브시스템을 하나의 물리적인 기계에 위치시켜 서버를 구성하였다. 최근에 정보 기술이나 비즈니스 부분에서 인터넷 비즈니스(Internet Business)의 중요성이 부각되면서 ERP 시스템에도 최선의 웹 기술(Web Technology)의 적용이 늘어나고 있다. ERP 시스템과 웹의 통합은 사용자가 ERP

시스템의 전용 프로그램을 설치하지 않고도 넷스케이프나 인터넷 익스플러어를 사용하여 인터넷을 통해 ERP 시스템에 쉽게 접근할 수 있게 해준다. 다시 말해서, 웹 어플리케이션 서버를 사용함으로써 개발자들은 ERP 기능과 외부 시스템의 다른 데이터 소스들과의 결합 및 ERP 프로세스에 영향을 미치지 않고 새로운 비즈니스 로직(Business Logic)을 첨가하는 작업이 용이해진다[14]. 예를 들어, 다섯 가지 서브시스템은 웹 어플리케이션 서버 형태로 구성할 수 있다.

4.5 구현뷰

구현뷰의 정적인 측면을 나타내는 컴포넌트도는 소프트웨어 컴포넌트와 이들 사이의 관계로 코드의 구조를 표현한다. 컴포넌트란 개념(Concept) 및 기능을 물리적 아키텍처로 구현한 것이다 [7].

본 논문에서는 컴포넌트도를 제시하지 않았다. 왜냐하면 ERP 개념 모델이 파일 형태의 컴포넌트와 같은 소프트웨어 개체가 아닌 개념적 모델링이나 비즈니스 도메인(Domain)과 관련된 개념을 기술하는데 초점을 두었기 때문이다. 일반적으로 컴포넌트도는 개념적 수준의 시스템 모델링을 수행하는 분석단계에서는 산출되지 않고, 설계와 구현 단계를 거치면서 획득되는 산출물이다.

지금까지, "4+1 뷰" 아키텍처 모델을 적용한 ERP 개념 모델을 설명하기 위한 다이어그램들과 그에 대한 간략한 설명을 기술했다. "4+1 뷰" 아키텍처 모델의 적용은

ERP 시스템과 관련된 이해 당사자들의 관점을 나타내기 위한 방법으로, 다양한 관점에서 ERP 개념 모델을 모델링하는 것을 가능하게 한다.

5. 결론

최근에 ERP 시스템은 급변하는 경영 환경에 적응하기 위한 기업의 핵심 전략 정보 시스템으로서 도입되고 있다. 본 논문에서는 UML과 "4+1 뷰" 아키텍처 모델 등의 개체 지향 모델링 기술을 적용하여 중소기업, 특히 제조업체를 위한 ERP 개념 모델을 제안했다.

ERP 시스템 구축 시 ERP 개념 모델을 사용함으로써 중소기업의 개발자들은 소프트웨어 개발 프로세스의 개선, 사용자 요구사항 수집의 용이성, 그리고 ERP 시스템의 기능적, 비 기능적 측면을 다양한 관점에서 기술의 가능성 등의 많은 이점을 얻을 수 있다.

첫째, 객체 지향 모델링 방법은 전체 소프트웨어 개발 프로세스 걸쳐 객체에 기반을 두고 동일한 개념적인 프레임워크를 제공한다 [9]. UML로 표현된 ERP 개념 모델은 구축과정에서 시스템의 기초 정보로 사용되고, 이로 인해 산출물의 재사용 촉진 및 재작업이 줄어들고, 일부 작업이 스킵되거나 삭제됨으로써 소프트웨어 개발 프로세스의 효율성이 향상된다[13, 17]. 예를 들어, 분석 단계에서 설계된 클래스도는 설계 단계에서 정제 및 재정의되고 구현 단계에 적용된다. 이러한 Seamlessness는 구축하는 ERP 시스템에 유연성과 추적성을 제공하여 사용자

요구사항 변경의 추적 및 시스템 적용, 그리고 유지 보수 작업이 편리해 진다.

둘째, ERP 개념 모델은 유사한 기능을 가진 회사들을 지원하는 유연성을 가진 일반적인 모델로 개발되었다. 즉, 제안한 개념 모델은 ERP 시스템을 구축하는 각 회사의 특별한 요구사항을 만족시키는 공통 프레임워크이다. 예를 들어, 자원과 자본이 취약한 중소기업의 특성을 고려하여 2계층 클라이언트/서버 구조를 ERP 시스템의 하드웨어 구조로 선택하고, 웹 서버, 데이터베이스 그리고 다섯 가지 서브시스템을 하나의 물리적인 기계로 서버를 구성하였다. 또한 비즈니스 요구사항 변경에 따른 ERP 시스템의 기능 변경을 용이하게 하기 위해 파이프-앤-필터 아키텍처를 모델용 서브시스템 사이의 관계 표현에 적용하였다.

그리고 UML의 사용사례도는 전통적인 구조적 분석에서 사용되고 있는 배경도(Context Diagram)처럼 사용된다. 즉, 객체 지향 모델링 방법에서는 사용자 요구사항을 수집, 분석하는 도구로 배경도 대신에 사용 사례도를 이용한다. 따라서 구축할 ERP 시스템의 사용자 요구사항 수집 및 분석 작업을 할 때 제안한 ERP 개념 모델의 사용 사례도를 활용함으로써 시스템 분석가는 좀더 쉽게 작업을 수행할 수 있게 된다.

셋째, 제안한 ERP 개념 모델은 "4+1 뷰" 아키텍처 모델을 적용하여 ERP 시스템의 기능적 측면을 포함한 비 기능적 측면까지 이해 당사자들의 다양한 관점을 기술하고 있다. 이는 이해 당사자들이 복잡한 ERP 시스템을 이해하는데 도움을 준다.

결론적으로, 다섯 가지 서브시스템으로

간략화해서 제안한 ERP 개념 모델을 적용함으로써 중소기업의 개발자들은 ERP 시스템의 구조 및 기능에 대한 명확한 이해가 가능하고, 공통 프랫폼의 적용으로 사용자 요구사항 수집 및 분석, 개발 및 유지보수 작업이 용이해진다. 따라서 소프트웨어 개발 프로세스의 효율성이 향상되고, 이로 인해 고품질의 ERP 시스템 구축이 가능해

진다. 또한 사용자는 ERP 개념 모델을 통해 개발자들과의 의사소통 능력이 향상된다.

향후에는 "생산관리" 서브시스템 및 다른 네 가지 시스템에 대한 자세한 설계 작업이 요구되며, ERP 시스템과 웹을 통합하기 위한 최신 기술에 대한 교착이 요구된다. 그리고 이를 바탕으로 제품화를 위한 구현 작업이 수행되어야 한다.

참고문헌

- [1] Christopher P. Holland and Ben Light, "Global Enterprise Resource Planning Implementation", *Proceedings of the 32nd Hawaii International Conference on System Sciences*, pp.1-10, 1999
- [2] Christopher P. Holland and Ben Light, "A Critical Success Factors Model For ERP Implementation", *IEEE Software*, pp.30-36, May/June 1999
- [3] Cory Casanave, "Business-Object Architectures and Standards", http://dbdoc.ajou.ac.kr/cetus/oo_business_objects.html
- [4] Gene Bylinsky, "The challengers move in on ERP", *Fortune*, pp. 306c-313c, Nov 1999
- [5] Gregory S. Lawson, "Enterprise Resource Planning(ERP) - Breakthrough or Buzzword?", *Oracle*, pp.291-295
- [6] Ivar Jacobson, Grady Booch, and James Rumbaugh, *The Unified Modeling Language Reference Manual*, Addison Wesley, 1999
- [7] Ivar Jacobson, Grady Booch, and James Rumbaugh, *The Unified Modeling Language User Guide*, Addison Wesley, 1999
- [8] Ivar Jacobson, Grady Booch, and James Rumbaugh, *The Unified Software Development Process*, Addison Wesley, 1999
- [9] Jean-Marc Jezequel, Alain Le Guennec, and Francois Pennaneach, "Validating Distributed Software Modeled with the Unified Modeling Language", *The Unified Modeling Language: first international workshop selected paper/UML 98 : Beyond the Notation*, Springer-Verlag, pp.365-377, 1999
- [10] Jeff Sutherland, "Business Objects in Corporate Information Systems", *ACM Computing Surveys*, Vol 27, No 2, June, 1995
- [11] Jim Arlow, Wolfgang Emmerich, and John Quinn, "Literate Modeling-Capturing Business Knowledge with the UML", *The Unified Modeling Language: first international workshop selected*

- paper/UML 98 : Beyond the Notation*, Springer-Verlag, pp.189-199, 1999
- [12] Joshua, "The Origin and Future of ERP Outsourcing", <http://www.erp-outsourcing.com/>
- [13] Martin Hitz and Gerti Kappel, "Developing with UML-Some Pitfalls and Workarounds", *The Unified Modeling Language: first international workshop selected paper/UML 98 : Beyond the Notation*, Springer-Verlag, pp.9-20, 1999
- [14] Martin Marshall, "Web application servers give green light to ERP", *Information week*, CMP Media Inc., Apr, 1999
- [15] Mohamed Mancona Kande, Shahrzade Mazaher, Ognjen Prnjat, Lionel Sacks, and Marcus Wittig, "Applying UML to Design an Inter-domain Service Management Application", *The Unified Modeling Language: first international workshop selected paper/ UML 98 : Beyond the Notation*, Springer-Verlag, pp.200-214, 1999
- [16] Pavel Hruby, "Structuring Specification of Business Systems with UML(with an Emphasis on Workflow Management Systems)", *OOPSLA 98 Business Object Workshop IV*, <http://jeffsutherland.org/oopsla98/paval.html/>
- [17] Philippe Desfray, "Automation of Design Pattern: Concepts, Tools and Practices", *The Unified Modeling Language: first international workshop selected paper/UML 98 : Beyond the Notation*, Springer-Verlag, pp.120-131, 1999
- [18] Philippe Kruchten, "Architectural Buleprints - The 4+1 View Model of Software Architecture", <http://www.rational.com/uml/resources/whitepapers/>
- [19] R. Youngs, D. Redmond-Pyle, P. Spaas, and E. Kahan, "A standard for architecture description", *IBM SYSTEMS JOURNAL*, Vol. 38, No. 1, pp.32-50, 1999
- [20] Ruth Malan and Dana Bredemeyer, *Functional Requirements and Use case*, 1999
- [21] Scott Buckhout, Edward Frey, and Joseph Nemeec Jr, "Making ERP Succeed: Turning Fear into Promise", *IEEE Engineering Management Review*, Strategy and Business, Second Quarter, pp.60-72, 1999
- [22] Thomas F. Gattiker and Dale L. Goodhue, "Understanding the Plant Level Costs and Benefits of ERP: Will the Ugly Duckling Always Turn Into a Swan?", *Proceedings of the 33nd Hawaii International Conference on System Sciences*, pp.1-10, 2000
- [23] "Mid-sized Firms Face Rough Road with ERP Adoption", *The Manufacturing Report*, Lionheart Publishing, Inc., May, 1999
- [24] . "Survey of Manufacturing : Meet the Global Factory", *IEEE Engineering Management Review Spring 1999*, <http://www.economist.com/>

저자소개

허분애(ios_h@hanmail.net)

1995년 숭실대학교 전자계산학과를 졸업했으며, 1995년부터 1998년 까지 삼성전자에서 근무하였다. 현재 숭실대학교 컴퓨터학과에서 석사과정 중이며, 소프트웨어공학이 전공이다.

주요관심분야는 Enterprise Resource Planning, Workflow, Electronic Commerce, Software Process, and Software Methodology 등이다.

정기원(chong@computing.soongsil.ac.kr)

1983년 University of Texas at Arlington에서 Computer Science Engineering 박사학위를 받았으며, 현재 숭실대학교 컴퓨터학과 교수로 재직 중이며, 한국감리인협회 회장, 한국전자거래학회 부회장 등을 역임하고 있다. 저서로서 소프트웨어 프로세스와 품질, 중소기업을 위한 소프트웨어 품질 시스템 등이 있다.

주요관심분야는 Software Process, Software Methodology, Electronic Commerce, and Real-Time Systems 등이다.

이남용

1979년 숭실대학교 컴퓨터학부를 졸업하고 국군정보사에서 정보시스템 개발과장을 지내고, 한국국방연구원에서 18년간 근무하였다. 한국국방연구원 해외장학위탁교육으로 미국 미시시피주립대학교에서 MIS박사학위를 취득하였고, 현재, 숭실대학교 컴퓨터학부 교수로 재직 중이며 한국정보통신기술사협회정책위원회 위원장, 한국전자거래학회 편집위원장, 국방부, 중소기업청, 서울시청 등의 자문위원으로 활동하고 있다.

주요관심분야는 전자상거래, 소프트웨어공학(UML, EJB, CORBA 등), 경영정보시스템(MIS, CRM, SCM, ERP, KMS 등) 이다.