

UML 기반의 정보시스템 구축 방법론에 관한 연구 : 자동차부품 제조업체의 자재조달 시스템 적용사례

장길상* · 정동규**

* 울산대학교

** AB&IT 코리아(주)

요 약

최근 정보기술(Information Technology)이 빠르게 변화·발전 하면서 새로운 패러다임의 기술들이 속속 등장하고 있다. 그 중 최근 정보시스템 개발 도구에 대한 객체지향 기법의 적용에 관한 관심이 고조되고 있다. 따라서 객체지향 방법론의 개념을 충분히 계승하면서 실제 적용에 있어서 명확하고 효과적이며 적용하기 쉬운 개발 방법론의 필요성이 무엇보다 요구되고 있다.

이러한 취지에서 본 논문에서는 UML을 기반으로 하는 정보시스템 구축 방법론을 제시하고자 한다. 본 방법론의 특징은 UML의 관련 다이어그램들 간의 상호 연관관계를 요구사항 수집 및 분석 단계로부터 시스템 분석 및 설계, 그리고 구현 단계에 이르기까지 체계적으로 정립하였다. 이를 기반으로 정보시스템의 3대 구성요소인 DB, 비즈니스 로직을 위한 코드, GUI 폼이 일관성 있게 생성될 수 있도록 하고 있다.

본 연구에서 제시된 UML 기반의 정보시스템 개발 방법론의 유용성(Usability)을 보여주기 위하여, 자동차 부품공급업체(1차 벤더)인 'H'사와 그 협력업체들(2차 벤더) 간의 자재발주 및 납품업무 프로세스를 대상으로 본 방법론의 설계절차를 따라서 정보시스템의 분석 및 설계 작업을 수행하였다. 이를 기반으로 인터넷 기반의 EDI 시스템의 프로토타입(Prototype) 시스템을 개발하였다.

1. 서 론

최근 객체지향 분석 및 설계 기법은 UML(Unified Modeling Language)로 통합되고 있다. 하지만, UML을 기반으로 하는 객체지향 방법론이라고 해도, 이것을 활용하는 컨설팅 업체나 분석가 및 설계자 등 전문가에 따라서 개발 프로세스와 개발과정의 모델 등이 상이하다. 따라서 정보시스템 분석 및 설계를 수행하는 초심자 및 일반 분석가들에게는 실제로 시스템 개발에 적용함에 있어서 많은 혼선과 어려움을 주고 있다. 또한 기존의 많은 문헌에서 제시하고 있는 UML 기반의 객체지향 방법론들은 진행 단계(요구사항 수집, 분석, 설계, 구현 등)에 따른 산출물들의 단순한 나열만으로 시스템 구축절차를 제시하고 있어 일반 시스템 분석가 및 설계자들이 이러한 방법론들을 사용할 경우 시스템 개발 진행이 쉽지 않다. 따라서 객체지향 방법론의 개념을 충분히 계승하면서도 실제 적용에 있어서 명확하고 효과적이며 활용하기 쉬운 개발 방법론의 필요성이 무엇보다 요구되고 있다.

이러한 취지에서 본 연구에서는 UML을 기반으로 하는 정보시스템 구축 방법론을 제시하고자 한다. 본 방법론의 특징은 UML의 관련 다이어그램

들 간의 상호 연관관계를 요구사항 수집 및 분석 단계로부터 시스템 분석 및 설계, 그리고 구현 단계에 이르기까지 체계적으로 정립하였다. 이것을 기반으로 정보시스템의 3대 구성요소인 DB, 비즈니스 로직을 위한 코드, GUI 폼이 일관성 있게 생성될 수 있도록 하고 있다. 또한 본 방법론과 기존 방법론들과의 큰 차이점 중의 하나는 요구사항 수집 및 분석단계에서 관리대상이 되는 클래스 객체를 파악 및 분석하는 개념모델을 먼저 작성하고, 그 다음으로 시스템의 동적인 측면을 보여주는 순차도를 그 다음의 분석 단계에서 그려 나간다. 그 이유는 순차도가 클래스 객체 사이의 메시지 및 연산을 주고받는 것을 묘사하는 것이므로, 클래스 객체 및 그들 간의 관계를 먼저 파악하는 것이 논리적으로 타당하기 때문이다. 또한 본 방법론에서는 정보시스템의 외관을 묘사하는 메뉴구조도 및 폼 객체를 유스케이스 모델 및 설계단계에서 중요하게 고려함으로써, 보다 쉽게 정보시스템을 분석 및 설계할 수 있도록 한다.

본 연구에서 제시된 UML 기반의 정보시스템 개발 방법론의 유용성을 보여주기 위하여, 자동차 부품공급업체(1차 벤더)인 'H'사와 그 협력업체들(2차 벤더) 간의 자재발주 및 납품업무 프로세스를 대상으로 본 방법론의 설계절차를 따라서 정보시스템의 분석 및 설계 작업을 수행하였다.

II. 기존의 UML 기반 정보시스템 개발 방법론의 고찰

급변하는 비즈니스 환경에서 기업 경영 전략을 만족시키기 위해서 정보시스템은 더욱 고도화되고 복잡해지고 있다. 이러한 여건에서 서비스 공급자 혹은 개발자들이 사용자의 요구사항에 부합하는 시스템을 개발하여 적시에 공급하는 일은 점점 더 어려워지고 있다. 또한 기존 시스템의 유지 보수에 들어가는 비용이 개발비용을 초과하는 상황이며, 기존 시스템에 추가적인 기능을 도입하려는 시도는 오히려 기존 시스템을 붕괴시킬 수 있는 위험성을 가지고 있다. 이러한 한계를 극복하기 위한 가장 적절한 방법론이 UML 개발 방법론이다.

UML은 소프트웨어 시스템, 더 나아가 업무 모델링, 기타 소프트웨어가 아닌 시스템의 산출물을 규정하고, 시각화하며, 적용하고, 문서화하는 언어로써, 복잡한 대형 시스템을 모델링하는데 성공적으로 증명된 공학적 기법들을 모아 제시한 것이다. UML은 80년대 후반에서 90년대 사이에 나타난 객체지향 분석 설계 방법론의 흐름을 이어받은 객체지향 모델링 언어이다.

[그림 1]은 문헌별로 제시하는 기존 UML 기반의 시스템 구축 절차를 단계별로 표현한 것이다. 구축절차를 제시하는 대부분의 문헌을 살펴보면, 요구사항 분석단계에서부터 적용(테스트, 유지보수)에 이르는 일련의 과정들이 개발 단계의 구분에 있어서 각 문헌들마다 다소 간의 차이는 있지만, 그 맥락은 일치한다. 그러나 UML 기반의 구축 절차는 단순히 구축 단계의 나열로써 표현하고 있기 때문에 실제 적용에 있어 다소 부족함이 있다.

참고문헌 [8]	참고문헌 [11]	참고문헌 [14]	참고문헌 [20]
요구사항 분석	요구사항 분석	요구분석	시스템 요구 정의
분석	OO 분석	분석	개념 모델 작성
설계	OO 설계	설계	유스케이스 작성
구현	OO 구현	구현	분석 모델 작성
	OO 테스트		클래스 재정의
	OO 유지보수		설계 모델
		테스트	구현

[그림 1] 기존 UML 기반의 시스템 구축 절차 비교

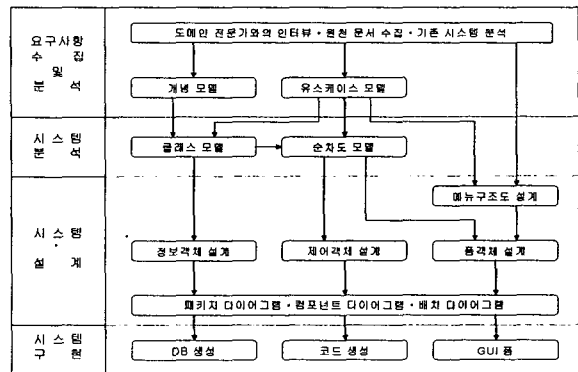
[표 1]은 문헌별로 제시하는 기존 UML 기반의 시스템 구축 절차 및 단계별 산출물을 중심으로 비교해 놓은 것이다.

[표 1] 기존 UML 기반의 시스템 구축 절차 및 단계별 산출물 비교

	참고문헌 [8]	참고문헌 [11]	참고문헌 [14]	참고문헌 [20]
요구 사항 파악	- 요구사항 기술서 - 개념도 다이어그램 - 시나리오 작성 - 유스케이스 다이어그램 - 연관 용어집 - 시퀀스 다이어그램 - 유스케이스 기술서 - 부가사항 기술서 - 종래방안 다이어그램	- 유스케이스 다이어그램 - 유스케이스 기술서 - 인벤토리 다이어그램 - 개략적인 사용자 화면	- 유스케이스 다이어그램 - 개념적 클래스 다이어그램 - 액티비티 다이어그램	- 대상 도출의 설명 - 요구사항 기술서 - 개략적인 클래스 다이어그램 - 유스케이스 다이어그램 - 유스케이스 기술서 - 시나리오
분석	- 클래스 다이어그램 - 클래스 시트 작성 - 상태 다이어그램 - 행위 다이어그램	- 클래스 다이어그램 - 시퀀스 다이어그램 - 종래방안 다이어그램 - 스프레드 다이어그램	- 클래스 다이어그램 - 종래방안 다이어그램 - 시퀀스 다이어그램 - 액티비티 다이어그램 - 상태 다이어그램	- 클래스 다이어그램 - 클래스 다이어그램 - 상태 다이어그램 - 상태 다이어그램 - 시퀀스 다이어그램 - 액티비티 다이어그램
설계	- 상세 클래스 다이어그램 - 프로세스 다이어그램 - 상세 시퀀스 다이어그램 - 상세 행위 다이어그램 - 디플로이먼트 다이어그램	- 상세 클래스 다이어그램 - 상세 시퀀스 다이어그램 - 상세 종래방안 다이어그램 - 상세 스프레드 다이어그램 - 컴포넌트 다이어그램 - 배치 다이어그램 - 단위 다이어그램 - 구현 사용자 화면 설계	- 상세 클래스 다이어그램 - 상세 클래스 다이어그램 - 상세 시퀀스 다이어그램 - 상세 액티비티 다이어그램 - 상세 상태 다이어그램 - 컴포넌트 다이어그램 - 디플로이먼트 다이어그램	- 구현 언어를 활용한 다이어그램의 정의 - 행위 다이어그램 - 컴포넌트 다이어그램 - 배치 다이어그램
구현	- 컴파일러 다이어그램 - 컴파일러 다이어그램 - 객체 타인 언어서 작성 및 생성 - 인터페이스 및 컴포넌트 구현	- 프로그래밍 (클래스, 객체지향언어)을 위한 코드작성, 객체지향언어(Java)를 위한 코드작성	- 프로그래밍 (클래스, 객체지향언어)을 위한 코드작성 - 클래스스 선언 구현 - 컴포넌트 선언 구현 - 실행 프로그래밍 구현	- 코딩 및 디버깅
테스트		- 테스트 계획 및 시행 - 테스트 결과 및 조치 보고서	- 단위 시험 - 통합 시험 - 시스템 시험 - 인수 시험	
유지 보수		- 유지보수 및 버전관리 문서		

III. UML 기반의 정보시스템 구축 방법론

본 연구에서 제시하는 방법론은 UML을 기반으로 하는 정보시스템 구축 방법론으로 그 절차는 [그림 2]와 같다.



[그림 2] UML 기반의 정보시스템 구축 절차

본 연구에서의 UML 기반의 정보시스템 구축 절차는 요구사항수집 및 분석, 시스템 분석, 시스템 설계, 시스템 적용, 테스트, 유지보수의 단계를 거쳐 수행된다.

3.1 요구사항 수집 및 분석

요구사항의 수집 및 분석 단계는 시스템 개발의

목표를 확립하는 과정이다. 이 단계에서는 도메인 전문가와의 인터뷰, 업무에 관련된 원천 문서(데이터 흐름도, 개체관계도 등)의 수집 활동, 기존에 사용되고 있던 시스템에 대한 분석 등을 통해 구축할 업무를 파악하고, 시스템의 업무흐름을 정의할 수 있기 때문에 매우 중요하다. 또한 시스템이 만족시켜야 하는 기능, 성능, 인터페이스 등과 같은 사항을 개념 클래스 모델과 유스케이스 모델(Use Case Model)의 단계를 거쳐 규명하는 과정으로 이는 전체 개발 일정 관리와도 매우 밀접한 관련이 있으며, 최종적으로 수집된 고객의 요구사항은 업무 시스템 개발에서부터 테스트 과정까지 영향을 미치게 된다. 이 단계의 산출물은 다음과 같다.

- 요구사항 기술서
- 개념 모델
- 유스케이스 모델
- 유스케이스 다이어그램
- 유스케이스 기술서

일반적으로, 기존의 방법론들은 순차도 모델 과정을 먼저 수행하고, 그 다음으로 클래스를 찾아내고 클래스 다이어그램을 그린다. 이것은 객체지향 방법론을 구조적 방법론과 같이 적용한 결과이다. 본 방법론에서는 대상 도메인의 분석에 있어서 관리대상이 되는 클래스 객체를 먼저 파악 및 분석하고, 그 다음으로 시스템의 동적인 측면을 보여주는 순차도를 그려 나간다. 그 이유는 순차도가 클래스 객체간의 메시지 및 연산을 주고받는 것을 묘사하는 것이기 때문이다.

3.1.1 개념 모델

기존 업무 구조의 분석을 통해, 개략적으로 정보객체의 클래스를 모형화 하는 단계이다. 즉, 기존의 업무 프로세스에서 개념적인 클래스와 객체를 찾아내고 클래스 사이의 관계를 파악하는 과정이다.

개념 모델은 프로젝트에 관련된 멤버 사이에서 그 도메인에 관한 지식을 모순 없이 공유하며, 이후의 개발에 있어서 원활한 의사소통을 도모하는 것에 무게를 두고 작성한다. 이것에 따라서 작업의 상호 체크가 가능하며, 사양 및 용어의 오해에 의한 잦은 재작업을 방지하거나 일상적인 회의시간을 단축시켜주는 등과 같이, 프로젝트 전체의 생산성을 높일 수가 있다. 기본적으로는 도메인에 관한 용어, 시스템이 취급하는 사항들에 대한 용

어의 이해에 노력을 기울이게 되지만, 여기에서는 UML 클래스 다이어그램의 표현력을 사용하기로 한다. 이유는 문장에 의한 설명보다도 클래스 다이어그램 쪽이 보기도 쉽고, 다른 여러 가지 일들 측면에서도 설명을 들을 때에는 자연스럽게 클래스 다이어그램으로 메모를 하기 때문이다.

3.1.2 유스케이스 모델

유스케이스는 시스템과 외부의 액터와 일련의 작업 흐름을 문장으로 기술한 것이다. 유스케이스 다이어그램은 액터를 표현하는 다이어그램과 유스케이스를 기호화 한 도형을 배치하고, 이들 사이의 관계를 파악할 수 있도록 한 것이다. 유스케이스는 시스템에 대한 요구사항을 모두 포함하지는 못하지만, 보다 알기 쉽게 표현하며, 시스템이 수행해야 하는 것들을 초기 단계에서 분할함으로써 그 이후의 분석 작업을 용이하게 해 준다. 또한 이 단계에서 정보시스템의 외관을 묘사하는 메뉴구조도 및 폼객체를 이 단계에서부터 고려해야 한다. 요구사항 분석 단계에서부터 이러한 고려는 보다 쉽게 정보시스템을 분석 및 설계할 수 있는 방법이라고 판단된다.

3.2 시스템 분석

분석된 요구사항 및 모델을 기반으로 하여 클래스를 추출하고 각 클래스의 매커니즘에 따라 요구사항을 분석하고 시스템의 논리적인 모델을 재정의 하는 과정을 말한다. 이 분석 모델은 설계 단계에서 실제적인 모델이 구축되기 위한 기반을 제공한다. 이 단계의 산출물은 다음과 같다.

- 클래스 다이어그램
- 순차도
- 협력도

3.2.1 상세 클래스 모델

개념 모형을 상세화하는 과정으로 실제 업무 프로세스의 흐름, 시스템 이용에 관련된 구체적인 절차와 객체를 정의 하는 단계이다. 유스케이스 모델을 바탕으로 객체를 선별해 내고, 객체간의 의사소통 등의 구체적 요소들을 분류해내는 작업이다.

3.2.2 순차도 모델

순차도 모델단계에서는 순차도를 이용하여 시간 순으로 배열된 객체의 상호작용을 보여준다. 이 다이어그램은 시나리오에 포함된 객체와 클래스, 그리고 유스케이스가 제공하는 기능을 수행하기

위해 필요한 객체 간에 교환된 일련의 메시지로 묘사된다. 순차도는 일반적으로 개발 중인 시스템 모델에서의 유스케이스와 밀접한 연관성을 지닌다. 또한 순차도 작성을 통해 자동적으로 협력도의 도출가능하다.

3.3 시스템 설계

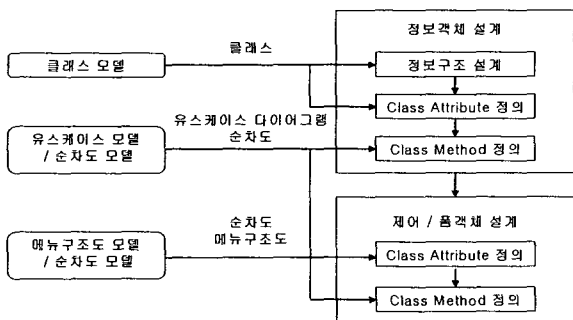
시스템 설계단계에서는 정보객체, 제어객체, 폼 객체를 설계한다. 정보 객체는 클래스 재정의 단계와 유스케이스 모델의 구조와의 관계로부터 도출된다. 이 과정에서 정보의 구조가 나타나게 되며 객체내의 속성이 정의된다. 제어/폼객체의 설계는 정보객체와 메뉴구조도 모델을 기반으로 제어/폼객체의 속성을 정의하고 해당 메뉴구조도 모델 내의 주요기능을 유스케이스 모델과 순차도 모델로부터 추출하여 제어/폼객체 메시지를 도출한다. 제어/폼객체를 정의하는 목적은 분석, 설계, 적용 단계 간의 일관성을 유지하고, 보다 효율적인 프로그래밍을 지원하는데 있다. 이러한 제어/폼객체는 구현 단계의 프로그래밍에 직접 적용되므로 정보객체의 운용과 인터페이스 모델의 세부적인 기능을 표현하는 속성과 메서드로 구성된다. 정보, 제어/폼 객체의 설계 과정을 요약하면 [그림 3]과 같다.

3.3.1 정보객체 설계

이 단계에서는 클래스 재정의 단계에서 정의되어 있는 객체의 구조와 관계를 정의하여 속성 중심의 정보객체를 설계하고, 객체의 메서드(method)를 정의한다.

3.3.2 제어객체 설계

제어객체는 순차도를 중심으로 작성되며, 정보객체의 운용에 관련된 속성 및 정보객체가 포함하지 않으나 적용 시 필요한 속성들을 정의한다. 또한 정보객체 설계에서 표현하지 못한 정보객체 간



[그림 3] 정보/제어/폼객체 설계 과정

의 인터페이스를 정의한다.

3.3.3 폼객체 설계

메뉴구조도 작성 단계에서 사용자에게 보여주고자 하는 화면을 찾아내고 각 화면들 사이의 관계를 파악하여 메뉴구조도를 작성한 후, 사용자 인터페이스에 관련된 객체를 정의한다.

3.3.4 객체 분류 및 배치

정보, 제어, 폼객체가 설계되고 나면, 이 객체들을 적용의 관점에서 모듈별로 분류하고 재배치하는 작업을 거치게 된다. 이러한 작업에 패키지 다이어그램, 컴포넌트 다이어그램, 배치도의 세 가지 다이어그램을 적절히 활용하면 효과적인 표현이 가능하다. 패키지 다이어그램은 클래스와 다양한 설계 산출물들을 일관성 있는 체계로써 패키지로 묶어 표현한 것이다. 컴포넌트 다이어그램은 컴포넌트 간의 유기적인 연결과 종속관계를 표현하고, 시스템의 정적 적용 관점을 표현하며, 클래스 다이어그램과 연계되어 작성된다. 배치도는 실행시의 노드들과 노드들에 포함된 컴포넌트의 위치를 표시하며, 컴포넌트 다이어그램과 연계되어 각 노드에 분산되는 컴포넌트들을 표현하게 되는데, 물리적으로 분산된 하드웨어에서 서브시스템들이 어디에서 실행되고, 서로 어떻게 교류하고 있는지를 보여준다.

3.4 시스템 구현

구현 단계는 전 단계인 설계 단계에서 나온 모델을 시스템의 실제 모습으로 변환시키는 과정으로, 이 단계에서는 정보, 제어, 그리고 폼 객체를 중심으로 서버 및 클라이언트의 프로그램들이 작성된다. 즉, 설계 단계에서 산출된 모델을 통하여 시스템 개발자가 선택된 적용환경 내에서 물리적인 정보시스템을 개발하는 단계이다.

[그림 4]와 같이 정보객체와 제어/폼 객체의 설계의 결과로 각 객체에 대응되는 테이블이 생성된다. 그리고 객체의 속성과 메서드를 정의하는 수준의 자바코드를 생성한다. 개발자는 이러한 결과물을 기반으로 폼객체와 정보객체의 메서드를 중심으로 구체적인 프로그래밍을 수행하게 된다.

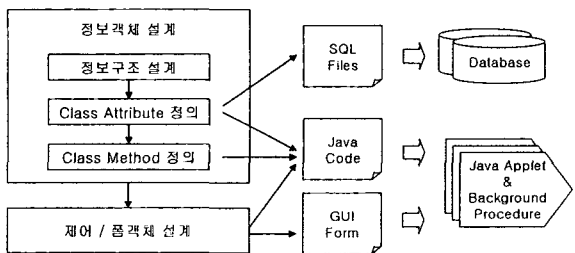
3.5 기존 방법론과의 비교

다음 [표 2]는 본 연구에서 제시하는 방법론과 기존에 제시된 개발방법론 중 본 연구와 관련 있는 대표적인 방법론들을 선정하고, 소프트웨어 생

산성, 기능적 추상성, 모델간의 일관성 등 평가항목별로 그들 간의 특징을 비교한다.

[표 2] 기존 방법론과의 비교

평가항목	OOA (Suhner-Mellor)	OMT (RumBaugb)	OOSE (Jacobson)	UML (Booch)	OO/DF 기반 통합 모델링 방법론	본 연구의 방법론
접근방법	객체지향 방법	객체지향 방법	객체지향 방법	객체지향 방법	혼합방법	객체지향방법
가용	임의도출을 기본으로 함	구조적 분석 E-R 도의 영향성 받음				
사용모델	확장 가능한 계층 데이터 흐름도 객체관계도	객체도, 상태전이도, 데이터흐름도	유스케이스도, 인터페이스도, 상호연속도	클래스도, 객체도, 구조도	객체사양서, 기능기술서, 인터페이스도	개념도, 분석도, 설계도, 정보, 프로세스도
소프트웨어 생산성	보통	높음	높음	높음	높음	높음
기능적 추상성	낮음	높음	비교적 낮음	비교적 낮음	높음	높음
모델간의 일관성	보통	비교적 높음	비교적 높음	비교적 높음	비교적 높음	높음
의미 풍부성	비교적 높음	높음	높음	높음	비교적 높음	비교적 높음
객체 속성 정의 용이성	어려움	어려움	보통	보통	비교적 쉬움	쉬움
패시드 정의 용이성	어려움	어려움	보통	보통	보통	보통
인자명 용이성	비교적 낮음	비교적 높음	보통	보통	비교적 낮음	보통
용역모델	고려함	고려함	고려함	고려함	고려하지 않음	고려하지 않음
제어/플럭스보게	지원하지 않음	지원하지 않음	지원함	지원함	지원함	지원함
사용자 인터페이스 모델	지원하지 않음	지원하지 않음	지원함	지원함	지원함	지원함



[그림 4] 구현 과정

IV. 적용 사례

4.1 요구사항 수집 및 분석

4.1.1 시스템 도입 배경

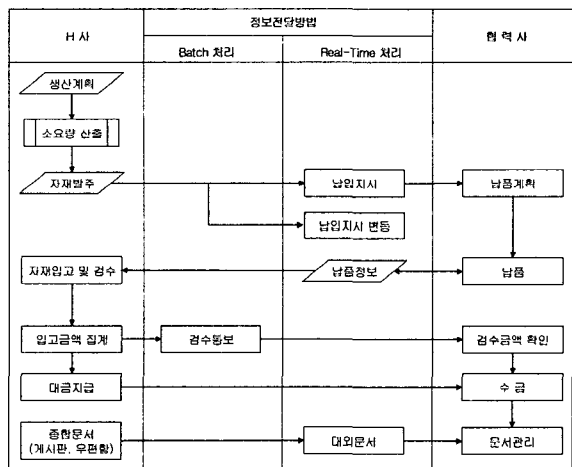
모기업과 1차 벤더 사이의 자료 흐름의 형태는 VAN, EDI, e-Procurement 등의 시스템을 이용하여 자동차 차체 투입 시 그에 따른 서열부품의 발주가 실시간으로 이루어지고 있다. 하지만 1차 벤더와 2차 벤더의 관계는 구두, 전화, 우편 및 FAX를 통한 원시적인 자재 발주형태를 취하고 있는 기업들이 많이 존재한다.

납품기사 구두통보, 전화, FAX, 우편을 통한 자재 발주 등의 업무 처리는 여러 가지 문제점을 내포하고 있다. 따라서 모기업에 적시 납품을 위해서는 자재발주 업무 개선의 필요성이 요구되었고, 경영진은 협력업체와의 자재 수·발주 업무 처리를 위한 인터넷 기반의 EDI시스템 도입을 결정하게 되었다.

4.2 업무 및 요구사항 분석

대상기업의 시스템은 기존의 MIS시스템을 주축으로 구성되어 있다. 도입하고자하는 EDI 시스템은 MIS시스템으로부터 생산계획에 따른 자재소요 계획 데이터를 실시간으로 전달 받아서, 자재 수·발주 업무를 처리하도록 한다.

대상기업 MIS시스템은 MIS 데이터베이스를 중심으로 품질, 검수, 납품, 제안, 총무, 인사, 구매, 생산, 설비, 자재, 회계, 금형, 단가 등의 업무들이 있다. 또한 이들 업무를 총괄하는 고객지원팀, 관리팀, 생산팀, 생산지원팀, 재경팀, 관련부서 등으로 구성되어 있다. 모기업의 VAN 시스템으로부터 수주자료를 받는 것에서부터 업무가 시작된다. 특히, 우리가 관심을 가지고 있는 부분인 2차벤더와의 자재 수·발주 업무는 생산지원팀의 자재관리업무에서 담당한다. 이들 발주 업무의 분석을 근간으로 협력업체에서의 수주와 납품, 대상업체의 검수에 이르는 전체적인 자재 수·발주 업무를 고려한, 개선된 자재 수·발주 업무 흐름도는 [그림 5]와 같다.



[그림 5] 개선된 자재 수·발주 업무 흐름도

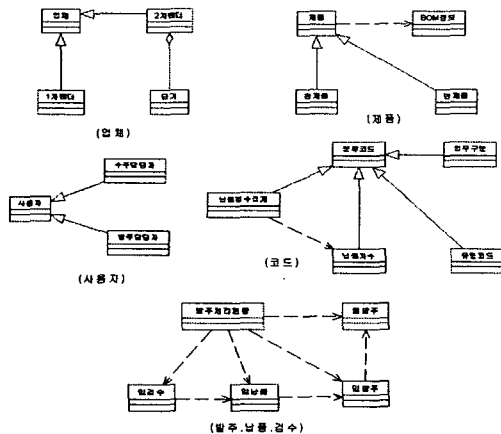
4.2.1 개념 및 유스케이스 모델

본 연구에서는 Rational사의 UML 모델링 도구인 Rose를 이용하여 정보시스템 구축의 모든 절차에 필요한 다이어그램, 구현에 필요한 코드, 그리고 데이터베이스 스크립트를 생성하였다.

본 연구에서는 수발주업무의 중심이 되는 업체, 제품, 사용자, 코드, 발주, 납품, 검수등과 같은 개념적 클래스를 도출하고, 각각의 관계를 정의하였다. 개념모델에서 요구하는 클래스 관계의 수준은 상당히 거시적이며 전체적이다. 따라서 클래스 내

부의 속성 및 메서드는 고려하지 않는 범위에서 정보의 구조 및 관계를 정의한다. 특히 발주과정에서 도출되는 발주, 납품, 검수 과정들의 관계는 분석 및 설계단계에서 자세히 도출된다. 따라서 종속적인 관계의 표현으로 나타내었다.

유스케이스 다이어그램을 바탕으로 발주관리 유스케이스에 대해 유스케이스 기술서를 작성한다.



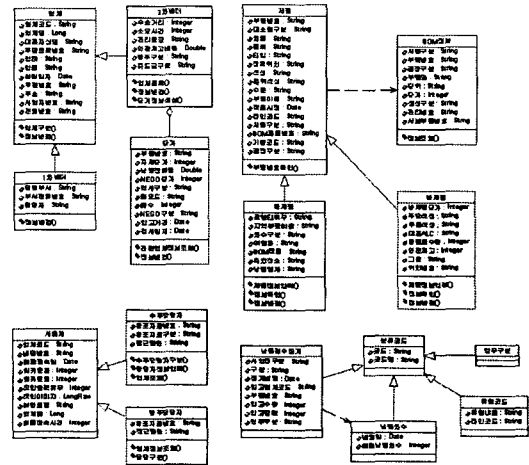
4.2 시스템 분석

4.2.1 클래스 모델

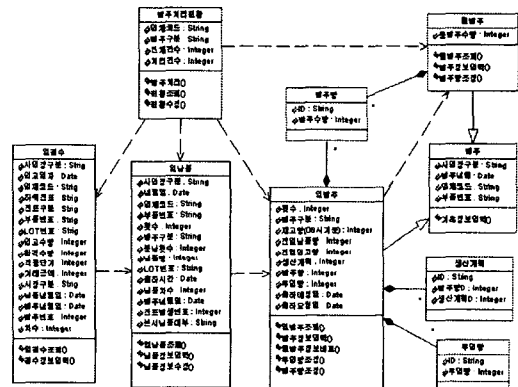
시스템 분석 단계의 산출물로서 [그림 7]의 클래스 모델은 기존 업무를 중심으로 기초정보와 업무 프로세스의 두 가지 관점으로 구분된다. 기초정보 관점의 클래스 모델은 각각 업체정보, 제품정보, 사용자 정보, 업체 내에서 사용하는 분류코드 정보로 구분된다. 업체정보의 경우 1차 벤더와 2차 벤더로 구분된다. 제품은 완제품, 반제품, 그리고 BOM 상의 부품정보로서 존재한다. 사용자는 수주 및 발주 담당자로 구분된다. 분류코드의 경우는 코드구조가 최상위 클래스로 존재한다. 그리고 하위에 납품차수, 업무구분, 유형, 검수집계 등의 클래스가 존재한다.

[그림 8]의 업무프로세스 관점의 클래스모델은 시스템이 운영되는 시점에서 발생하는 정보의 구조로 표현된다. 이러한 정보는 발주, 납품, 검수의 세 가지 단계에 걸쳐 정보가 생성되고, 각 단계에서 처리현황정보가 존재하게 된다. 발주정보의 경우는 월발주와 일발주로 구분되고, 각각 발주량 및 생산계획, 투입량 등의 정보를 포함하는 형태로 표현된다. 클래스 모델에서 각 정보간의 선후 관계는 종속적 관계로 표현된다.

이러한 클래스 모델은 시스템 설계단계에서 정보 객체의 형태로 변환되고, 필요에 따라 기초정보, 수·발주정보, 분류체계정보로 구분되어 각각의 다이어그램으로 표현되고 구현된다.



[그림 7] 기초정보관점의 클래스 모델



[그림 8] 업무프로세스관점의 클래스 모델

4.2.2 순차도 모델

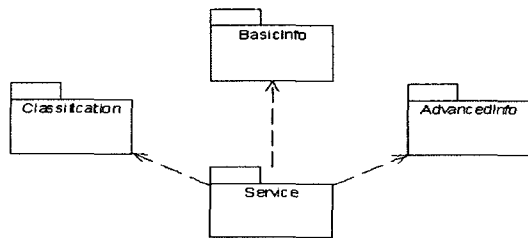
자재 수·발주 업무의 순차도를 작성한다. 1차벤더 담당자가 1차벤더용 기초자료를 검색 후 발주 정보를 입력한다. 2차벤더 담당자는 발주 정보를 받아 벤더용 기초자료를 검색하고, 납품가능여부를 확인 후 납품정보를 1차벤더에 전송한다. 그런 다음, 실제 제품이 납품, 검수 과정을 거치게 된다. 검수과정의 결과는 다시 2차벤더 담당자에게 전송된다.

4.3 시스템 설계

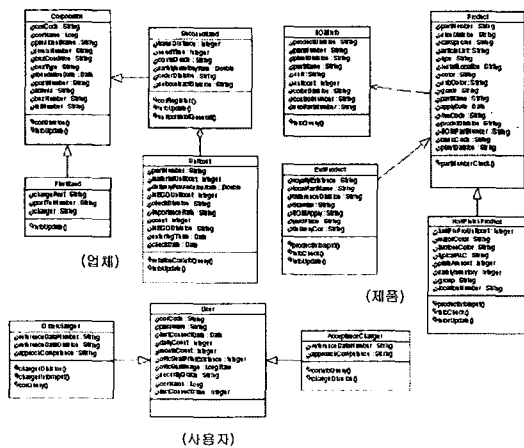
4.3.1 정보객체 설계

시스템설계 단계에서 표현되는 패키지의 형태는 다음과 같다[그림 9].

- BasicInfo 패키지는 시스템에서 사용되는 기본적인 정보의 형태 및 구조를 표현하는 것으로 정보의 대상, 주체 등이 정의된다. 구성 내용으로는 번더정보, 사용자정보, 제품정보 등이 있다.
- AdvancedInfo 패키지는 업무 진행 과정에서 발생하는 정보를 표현하며 기본적으로 수·발주 정보가 구성된다.
- Classification 패키지는 수·발주 과정 및 회사 내에서 기 정의된 코드 및 분류체계를 표현하기 위한 클래스의 집합으로 표현된다.
- Service 패키지는 각 패키지간의 정보전달 및 사용자 인터페이스로의 정보전달 과정에서 필요한 클래스들로 구성된다.



[그림 9] 정보/제어 패키지 다이어그램

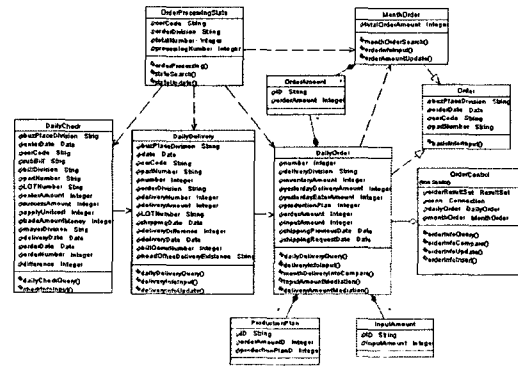


[그림 10] 기초정보 클래스 다이어그램

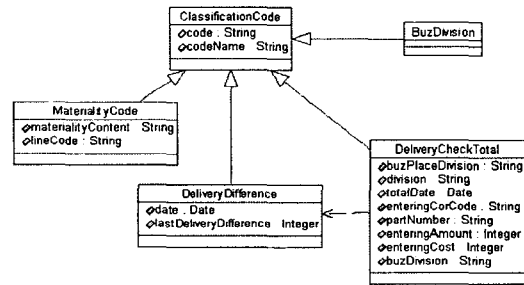
기초정보 클래스 다이어그램은 기초정보관점의 클래스 모델을 기준으로 구성된다. 각각의 클래스는 물리적인 클래스 수준으로 상세하게 재정의 되고, 재사용성 및 의미관계를 명확히

하는 관점에서 설계된다. 이 다이어그램의 주요한 클래스는 업체, 제품, 사용자로 구분되고, 각각의 클래스는 상속 및 연관 관계를 적절히 구성하는 몇 개의 클래스로 분리 되어 구성된다.

분류체계정보 클래스 다이어그램은 수·발주 및 회사 내의 기 정의된 코드 및 분류체계를 표현한다. 분류코드 클래스를 업체구분, 자재코드, 납기구분, 납기검수합계 클래스가 상속받고, 납기검수합계 클래스는 납기구분 클래스에 의존관계를 갖는다.



[그림 11] 수·발주정보 클래스 다이어그램



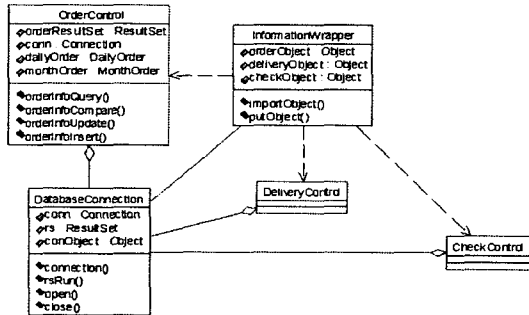
[그림 12] 분류체계정보 클래스 다이어그램

수·발주 정보 클래스 다이어그램은 수·발주정보의 기술로부터 발주, 수주, 검수 통보 과정에서 발생하는 정보들이다. 각각의 업무과정을 클래스로 표현하고, 클래스간의 관계는 실제적인 관계로 표현하지는 않고 종속관계로 표현한다. 발주객체의 경우는 일발주와 일발주의 두가지 주요 클래스로 표현되고, 각각의 클래스는 몇가지 내부클래스의 집합을 속성의 형태로 가진다.

4.3.2 제어객체 설계

제어(서비스) 클래스 다이어그램은 프로세스 측면의 다이어그램이다. 순차도의 결과물로서, 각각

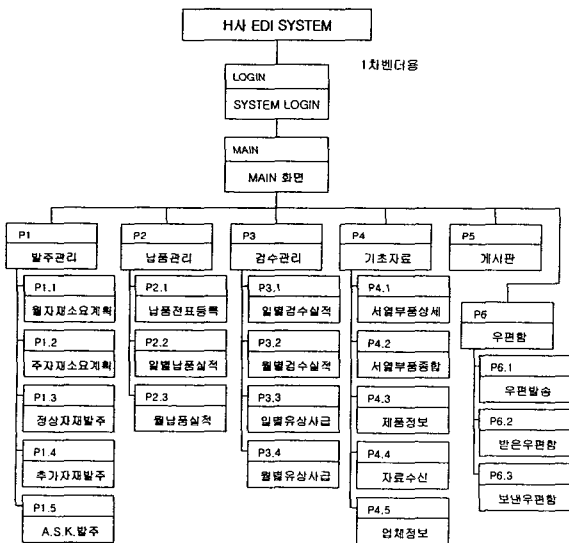
의 필요 메서드를 정의한다. 메서드의 기본적인 기능은 데이터베이스와 폼(메뉴구조)과의 연계이므로 데이터베이스 커넥션객체와 데이터 전달객체의 연결을 중심으로 하는 구조로 표현된다.



[그림 13] 제어(서비스) 클래스 다이어그램

4.3.3 메뉴구조 설계

자재 수·발주 시스템의 특성상 1차벤더사용자와 2차벤더사용자의 시스템 메뉴구조는 다르다. 1차 벤더의 메뉴구조도는 [그림 14]와 같다.



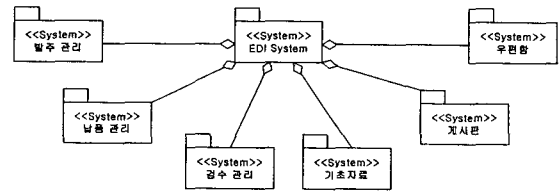
[그림 14] 1차벤더 메뉴 구조도

4.3.4 객체 분류 및 배치

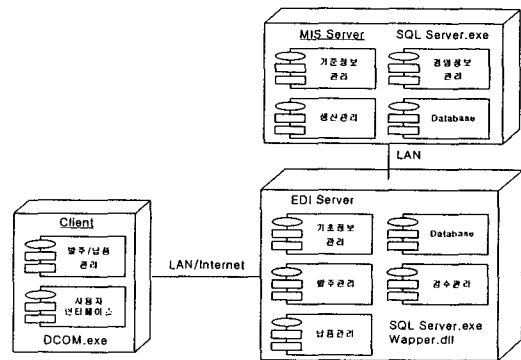
[그림 15]는 본 시스템의 패키지 다이어그램을 나타낸 것이다. EDI시스템은 발주, 납품, 검수 관리, 기초자료, 계사판, 우편함의 패키지들로 구성되고, 각 패키지를 모두 포함하는 관계로 구성된다.

[그림 16]은 본 시스템의 배치 다이어그램이다. 하드웨어 시스템들의 고유한 특성을 담고 있으며, 노드를 통해 각 시스템마다의 하드웨어, 소프트웨어

어 컴포넌트들의 관계를 나타낸다.



[그림 15] 패키지 다이어그램



[그림 16] 배치 다이어그램

V. 시스템 구현

5.1 적용 환경

일반적으로 웹 어플리케이션의 구조를 살펴보면, [그림 17]과 같은 형태를 가지는데, 마이크로소프트의 ISAPI, 네스케이프의 NSAPI가 이에 해당된다. ASP(Active Server Pages), JSP(Java Server Pages)는 이러한 형태의 기술들에 관계된 것으로, 본 시스템도 이러한 웹 어플리케이션 구조를 따르도록 한다.

본 시스템의 하드웨어 자원과 소프트웨어 자원의 구성은 H사의 EDI 시스템을 다수의 클라이언트가 인터넷을 통해 연결되는 구조를 가진다.

본 시스템의 네트워크 구성도는 'H'사 내부적으로는 MIS서버가 허브를 통해 EDI시스템과 연결되어있고, H사 클라이언트들이 허브를 통해 접속 가능하다. 또한 모든 전송 문서는 방화벽을 통해 라우터를 거쳐 인터넷으로 연결된다. 이를 통해 각각의 협력업체들이 접속하는 형태의 네트워크 구조를 나타낸다.

5.2 데이터베이스 테이블

[표 3]은 업체 마스터와 업체별 단가에 대한 데이터베이스 테이블이다.

[표 3] 데이터베이스 테이블(업체 MASTER)

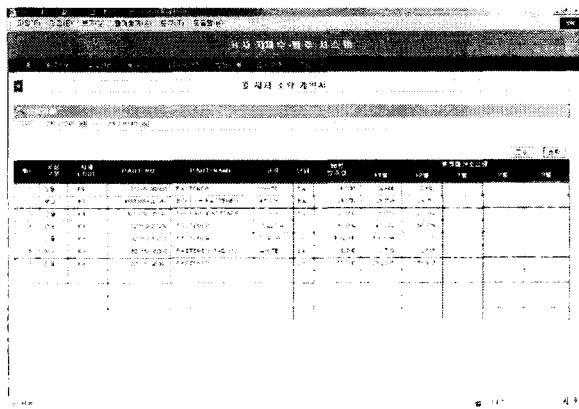
필드(한)	필드명(영)	TYPE	LEN	KEY	NULL	DEF	F/K	CHECK	REMARK	
업체코드	VENDCD	CHA	4	P1	NOT				-관리등급 A:매우중요 B:중요 C:보통 -발주구분 Y:발주 대상업체 N:발주 제외 대상업체	
업체명	VENDNM	VAR	30							
대표자성명	NAME	VAR	20							
주민등록번호	JNO	CHA	14							
업태	HOBST	VAR	20							
업종	HKIND	VAR	30							
- 중 략 -										
FAX번호	FNO	CHA	13							
당사와의 거리	KM	NUM	3		NOT	O				
평균소요시간	HOURL	NUM	3		NOT	O				
관리등급	CGRAD	CHA	1			C		A,B,C		
안전재고비율	SAFINV	NUM	5		NOT	O				

5.3 구현화면

적용 화면은 크게 대상기업의 담당자를 위한 화면과 협력업체 담당자를 위한 화면으로 나누어진다. 그리고 다시 월/주/일 단위로 나누어지며, 각각은 발주, 납품, 검수 화면으로 구분된다. 기초자료와 관련된 화면은 공통적으로 제공된다.

적용 화면의 예로는 Main화면과 ID/Password 입력화면을 제외한 자재 수·발주 업무와 직접적인 관련이 있는 협력업체 담당자용 화면을 중심으로 구성하였다.

[그림 17]은 모두 월 단위를 중심으로 각각 자재 발주, 납품, 검수와 관련된 업무 처리 화면으로, 부품번호와 발행년월일을 기준으로 조회가 가능하고, 출력 기능이 제공된다. 제품정보를 조회할 수 있는 기초자료와 관련된 화면들은 부품번호별, 부품종류별 조회가 가능하고, 출력 기능이 제공된다.



[그림 17] 발주관련 화면

VI. 결론 및 향후 연구방향

본 연구에서는 UML을 이용한 정보시스템 개발 방법론을 제안하였다. 그 적용 가능성을 입증하기 위하여, 자동차 부품업체인 H사의 자재 수·발주 업무처리를 지원하는 인터넷 기반의 EDI 시스템에 적용하였다. 이 적용사례를 통하여, 본 연구의 방법론을 단계적으로 적용해 보면, 정보시스템의 핵심적인 구조인 정보모델, 제어모델, 폼(사용자 인터페이스) 모델을 체계적으로 분석·설계하여 도출 하였다. 따라서 기존의 방법론들에 비해 개발 단계간의 일관성 유지가 가능하고, 사용자 요구사항의 파악이 용이하다는 것을 보여 줄 수 있었다.

기존의 대부분 객체지향 방법론들은 방법론의 진행단계(phase) 및 산출물들이 단순히 나열되어 있기 때문에, 이러한 방법론들을 사용할 경우 효율적인 시스템의 구축을 진행해 가는 것이 쉽지 않았다. 또한 기존의 방법론들은 순차도를 먼저 작성하고, 그 다음으로 클래스 다이어그램을 작성한다. 이것은 객체지향 방법론을 구조적 방법론과 같이 적용한 결과라고 볼 수 있다.

본 방법론에서는 대상 도메인의 분석에 있어서, 관리대상이 되는 클래스 객체를 먼저 파악 및 분석하였다. 그 다음으로 시스템의 동적인 측면을 보여주는 순차도를 그려 나간다. 그 이유는 순차도가 클래스 객체 사이의 메시지 및 연산을 주고 받는 것을 묘사하는 것이기 때문이다. 따라서 개념적 클래스 다이어그램을 먼저 작성하여야 한다. 또한 정보시스템의 GUI를 묘사하는 메뉴구조도 및 폼객체를 유스케이스 모델 및 설계단계에서 중요하게 고려함으로써, 보다 쉽게 정보시스템을 분

석 및 설계할 수 있도록 하였다.

참고문헌

- [1] 권순덕, XML과 EDI를 이용한 WorkFlow 시스템 설계 및 구현(자동차 주문 시스템에 적용), 울산대학교, 석사학위논문, 2000.
- [2] 김오환, 정경수, 자동차 부품업체의 EDI 활용, 경북대 경영대학원, 한국정보시스템학회지 2000년 춘계학술대회발표논문집, p21~p32, 2000.
- [3] 김창완, UML과 자바, 대림출판사, 2003.
- [4] 김철민, 중소 부품업체의 WEB EDI 솔루션에 관한 연구(자재발주 시스템 사례연구), 울산대학교 정보통신대학원, 석사학위논문, 1999.
- [5] 김형수, 인터넷을 통한 조달업무의 구현. 울산대학교 정보통신대학원, 석사학위논문, 2001.
- [6] 류형규, 이순천, 류시원, UML 기반 객체지향 클라이언트/서버 구축, 홍릉과학출판사, 2000.
- [7] 문태수, 김호진, UML을 이용한 XML/EDI 시스템 설계 및 구현, The Journal of Korean Institute of CALS/EC Vol. 7, No. 3, Dec. 2002.
- [8] 민병수, 기업간 전자상거래를 위한 XML/EDI 시스템의 개념적 설계에 관한 연구. 울산대학교 정보통신대학원, 석사학위논문, 2001.
- [9] 박현철, UML과 로즈를 활용한 객체지향 소프트웨어 개발, PTC Korea, 1999.
- [10] 신정범, OOIDEFO 기반의 객체지향 정보시스템 분석 및 설계 기법에 관한 연구, 울산대학교, 석사학위논문, 2000.
- [11] 이우용, 고영국, 박태희, 김준수, UML과 객체지향 시스템 분석설계, 그린출판사, 2002.
- [12] 황성룡, 문치용, 김재균, 장길상, 객체지향 IDEF 기반의 통합모형화 방법론 : 조선 CIMS 분야 사례연구, 경영정보학회연구, 제 9권, 제 3호, 9월, pp.47-73, 1999.
- [13] Craig Standing, Methodologies for developing Web applications. Ingormatin and Software Technology 44(2002) 151-159.
- [14] Jingfeng Li, Jian Chen, Ping Chen. Modeling, Web Application Architecture with UML, IEEE(2000).
- [15] John M. Artz, A top-down methodology for building corporate Web applications. Internet Research vol.6 1996. p.64-74.
- [16] Kassem Saleh, Documenting electronic commerce systems and software using the unigied modeling language. Information and Software Technology 00(2001) 000-000.
- [17] Kenji Takahashi, Eugene Liang, Analysis and design of Web-based information systems, Computer Networks and ISDN System 29(1997) 1167-1180.
- [18] Luciano Baresi, Franca Garzotto, Paolo Paolini, Extending UML for Modeling Web Applications, IEEE(2001).