

UML에 기초한 어플리케이션 자동 생성 Case Tool의 분석 (Analysis of Application Automatic Creation Case Tool Base on UML)

한 현 관* 이 명 진**
(Hyoun-Goun Han) (Myung-Jin Lee)

요 약

UML(Unified Modeling Language)은 소프트웨어 시스템의 명세화, 시각화, 생성, 그리고 문서화를 목적으로 하는 언어이다. 또한 UML은 소프트웨어 생성을 위한 청사진을 만들어줌으로써 소프트웨어 설계의 복잡한 과정을 단순화시킨다. 본 연구에서는 Rational Rose, BizWiz와 같은 자동화된 응용프로그램 생성 도구들을 실제의 사이버 쇼핑몰에 적용하여 UML 관점에서 비교 및 분석한다. 그리고 위의 결과를 바탕으로 고객관리 업무에 UML을 적용하여 체계적인 시스템 분석, 설계 및 구현방안을 제시한다.

ABSTRACT

Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It simplifies the complex process of software design, making a blueprint for construction. In this paper, we apply automated application generation tools such as Rational Rose and BizWiz to a realistic cyber shopping mall and then compare and analyze them in the view of UML. We also propose some methods for the systematic system analysis, design, and implementation by applying UML to a customer management system based on above results.

1. 서론

최근 인터넷을 이용하는 전자상거래 시장이 폭발적인 성장을 거듭하고 있으며, 정부는 2000년대 정보화 비전인 'Cyber Korea21'에서 국가 CALS/EC 체계 구축을 통하여 작고 효율적인 전자정부를 구현하려 하고 있다. 이를 위해서 정부 및 공공기관이

보유하고 있는 자료의 관리 및 공유, 전자결재, 공문서의 전달을 원활하게 하는 시스템의 개발이 필요하게 되었다. 그러나 역할 할당의 효율성 부족 그리고 다수의 단순 개발인력의 투입으로 생산성 저하 및 프로젝트 비용이 증가하였을 뿐만 아니라 최근에 개발되고 있는 소프트웨어들은 대용량이고 복잡해지는 경향을 가지고 있다. 이러한 문제점을 해결하기 위해서 객체지향 개발 방법론이 소개되어 많이 이용되고 있으며[4, 6, 8], 개발 방법론의 연구를 체계적으

* 정희원 : 대구산업정보대학 컴퓨터정보계열 교수

** 정희원 : 아시아전통과학대학교 인터넷비즈니스학과 교수

논문접수 : 2002. 12. 24.

심사완료 : 2003. 1. 10.

로 지원하기 위한 CASE 환경에 관한 연구도 진행되고 있다[1, 2].

본 논문은 객체지향 소프트웨어 모델링 기법인 UML(Unified Modeling Language)을 이용하여 효율적으로 데이터베이스를 구축하는 방법과 고품질의 소프트웨어를 경제적으로 생산하고 운영하기 위한 분석, 설계 그리고 구현 등의 과정에 걸쳐 적용되는 이론, 절차 및 기법들에 관한 연구이다. 본 연구에서는 전자상거래의 빠른 발전 추세에 맞추어 사용자의 요구사항을 정확하게 반영하는 체계적인 시스템 분석, 설계, 구현방안을 제시하고 이를 실제의 사이버 쇼핑몰에 적용함으로써 발생하는 문제점을 UML을 통해 분석, 설계 및 구현하였다. 그리고 소프트웨어 공학 측면에서 소프트웨어 개발 전과정을 지원하는 자동화 도구인 Rational Rose 2000[10], BizWiz[3], JBuilder 7[7] 등의 여러 도구들을 통해 위의 문제점을 수정 보완할 수 있는 방법을 제시한다.

본 논문의 구성은 다음과 같다: 2장에서는 UML의 개념과 특성을 소개하고, 3장에서는 시스템 분석 및 설계에 관해 그리고 4장에서는 자동 생성 시스템에 관해 서술한다. 5장에서는 본 연구의 구현결과와 각 CASE 도구들을 평가하고, 결론 및 향후 연구방향에 대해 논의한다.

2. UML

2.1 UML의 개요

UML은 객체지향 분석과 설계를 위한 모델링 언

어이며, 어휘와 규칙을 사용하여 시스템을 개념적이고 물리적으로 표현하여 의사 소통을 돕는 것을 목적으로 한다. 한편 시스템을 이해하기 위하여 UML은 하나 이상의 모델들을 서로 연결하여 사용하는데, [그림 1]은 UML을 구성하는 모델들 사이의 상호관계를 보여주고 있다[5, 9].

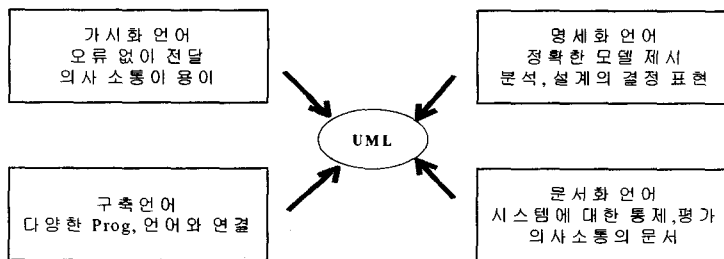
모델은 현실을 단순화하고 가시화하여 시스템의 청사진을 제공하는 것이다. 또한 시스템을 총체적으로 계획 및 표현함으로써 시스템에 중요하게 영향을 미치는 요소를 파악하며 불필요한 요소를 생략할 수 있다. 모델을 만드는 작업은 품질이 좋은 소프트웨어를 개발하기 위한 모든 활동의 중심이며, 모델 구축을 통해 개발 대상 시스템에 대한 이해를 증진시킬 수 있다. 모델에 대한 이해는 아래와 같이 모델링 알고리즘 관점과 객체지향 관점이 있다:

알고리즘 관점: 소프트웨어 시스템의 주요 구성요소인 프로시저와 함수를 제어 관점에서 분할하여 시스템을 모형화한다. 이 관점의 단점은 요구사항의 변화에 적응력이 떨어지고 대규모 시스템에서는 유지보수를 포함한 관리에 어려움이 있다.

객체지향 관점: 소프트웨어 시스템의 기본 요소를 객체와 클래스로 파악하여 문제 영역과 해결 영역을 모형화한다. 달리 말하면 객체의 연결관계에 의해 파생되는 부가적인 속성들을 분리하는 것을 의미한다.

2.2 UML 구성요소

UML의 구성요소를 크게 사물(thing), 관계(relationship), 그리고 다이어그램(diagram)으로 나눌

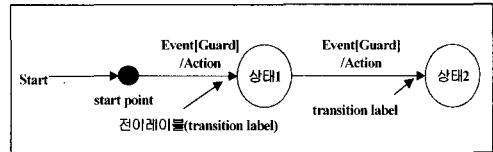


[그림 1] UML 구성 모델들 사이의 상호관계
[Fig. 1] Interaction of UML Component Models

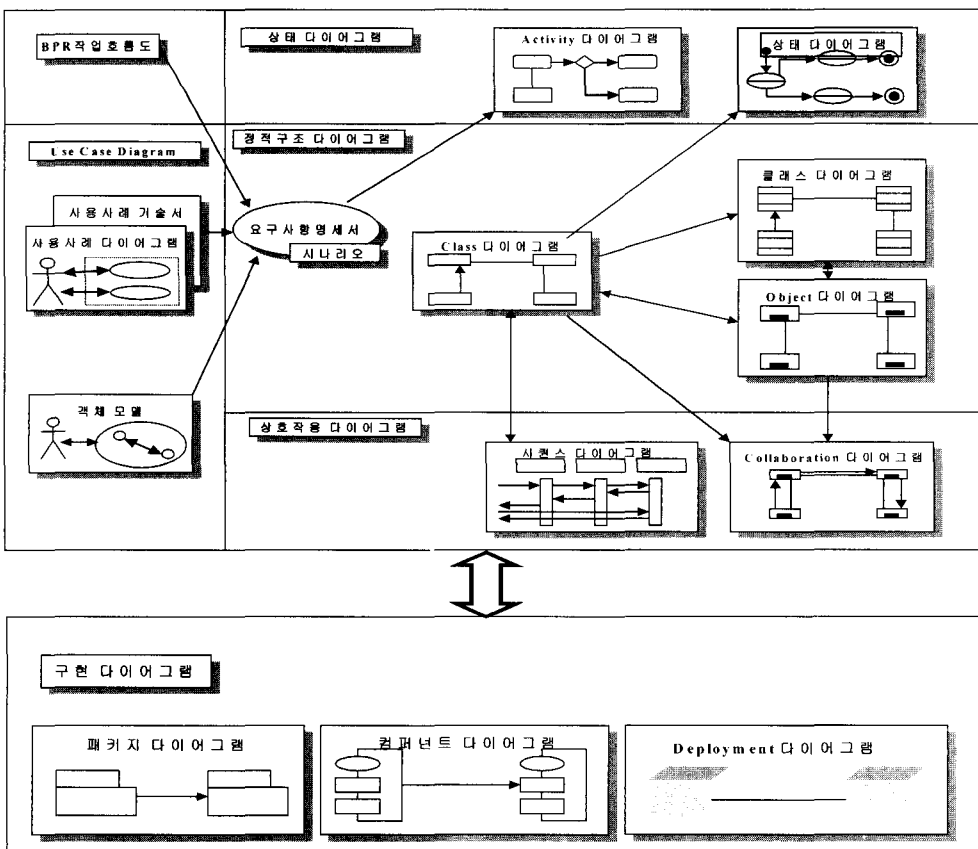
수 있는데[그림 2], 아래에서 이들에 대해 간단히 살펴본다. 사물은 추상적 개념으로 모형 구성의 기본 요소이며 대표적인 사물에는 구조 사물이 있다. 구조 사물은 UML 모형의 명사형으로써 모형의 정적인 부분이며, 개념적 표현 혹은 물리적인 요소들을 표현하며 클래스와 인터페이스가 여기에 해당된다. 관계는 구성요소 사이의 의미있는 연결을 의미하며, 관계에는 의존 관계, 연관 관계, 일반화 관계, 실체화 관계 등이 있다. 마지막으로 다이어그램은 구성요소들을 그래픽으로 표현한 것이다.

특히 UML 시스템의 구성요소에서 [그림 3]과 같은 상태 다이어그램은 시스템의 행동을 기술한 것이며, 이것은 state, transition, event, active로 구성된다. 이 다이어그램을 작성할 때는 객체 행동에 중점을

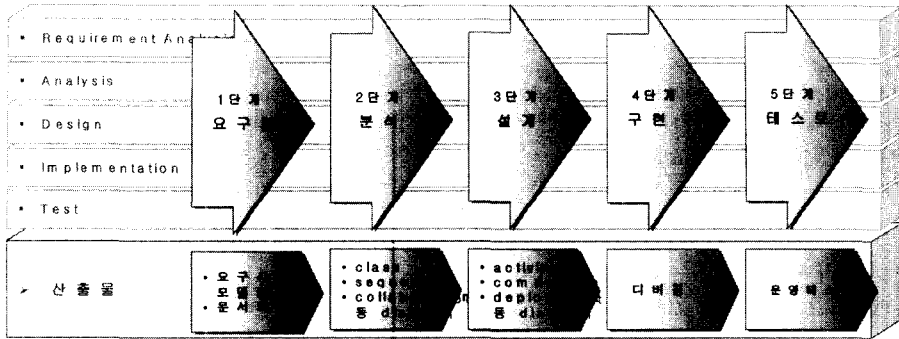
두어야 한다.



[그림 3] 상태 다이어그램
[Fig. 3] State Diagram



[그림 2] UML 시스템의 구성도
[Fig. 2] Organization Diagram of UML System



[그림 4] UML을 이용한 소프트웨어 개발단계들
 [Fig. 4] Software Development Steps that Use UML

3. 시스템 개요 및 분석

3.2 시스템 분석

3.1 시스템 개요

본 연구에서는 사이버 쇼핑몰을 중심으로 고객관리 업무에 UML을 적용하여 분석, 설계 및 시스템을 구현한다. 생산성 향상과 업무변화에 따른 유지보수 등에 신속하게 대처하도록 시스템을 구현하였으며 시스템의 개요는 다음과 같다:

- 풍부한 기능을 가진 시스템 구축
 - 상품관리, 주문관리, 고객관리, 매출관리 등
- 기존 업무를 효율적으로 관리
 - 구매/발주관리, 재고 관리, 배송관리, 고객관리 등
- 고객 정보의 대한 체계적인 관리
- 더욱 풍부한 콘텐츠 제공

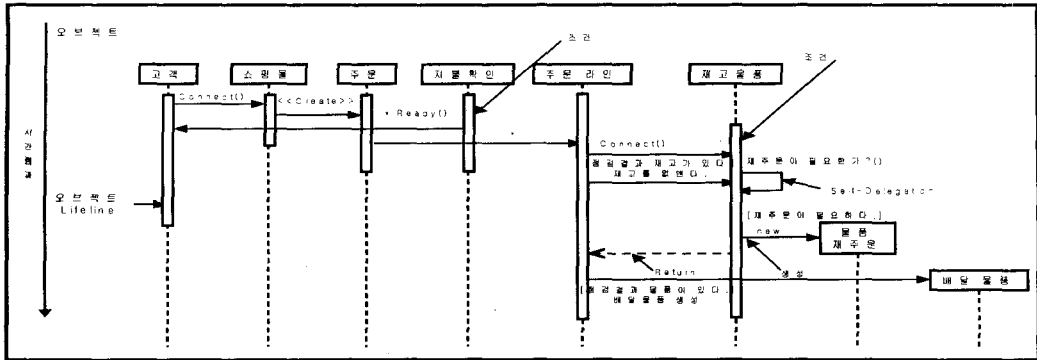
먼저 요구명세를 하고, 분석하고, 설계하고, 구현하고, 마지막으로 테스트하는 소프트웨어 개발주기의 전과정에 대해 단계 정의 및 단계별 업무들을 정의한다. 시스템의 정적, 동적, 기능적인 부분을 모델링할 수 있는 방법으로 클래스 다이어그램, 순차 다이어그램, 협동 다이어그램 등을 제공하며, 클래스들의 상위 개념을 그룹화하는 패키지 다이어그램을 제공한다. 한편 필요에 따라서 이전 단계에서 앞 단계로 피드백이 일어날 수도 있다. [그림 4]는 UML을 이용한 소프트웨어 개발단계들을 보여주고 있다.

본 연구에서 쇼핑몰을 구현하기 위해 시스템을 분석한 내용과 이에 사용된 산출물들을 <표 1>에 요약하였다.

<표 1> 시스템 분석의 내용과 산출물

<Table 1> Content and Product of System Analysis

단계	내용	산출물	
분석	소프트웨어 개발 공정으로 사용자의 요구사항을 각 모듈로 구분하여 모델링	정적 구조 다이어그램	클래스 다이어그램
			객체 다이어그램
		상태 다이어그램	활동 다이어그램
			상태 다이어그램
		상호작용 다이어그램	순차 다이어그램
			협동 다이어그램
		배치 다이어그램	

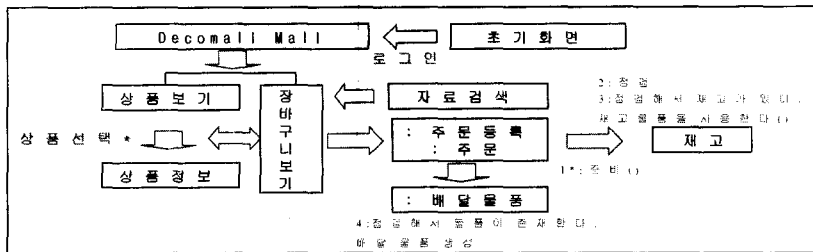


[그림 6] 순차 다이어그램

[Fig. 6] Sequence Diagram

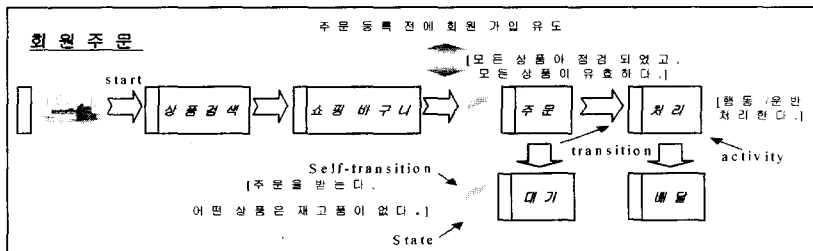
[그림 7]과 같은 협동 다이어그램의 경우에 객체들 사이의 행위를 나타내는 것은 순차 다이어그램과 동일하며, 이것은 메시지를 주고받는 객체의 구조적인 구성을 강조하며 정적인 구조에 더 중점을 두고 있다.

[그림 8]의 상태 다이어그램은 객체가 가질 수 있는 모든 상태들과 특정한 이벤트를 받았을 때 그 결과로 어떤 상태로 변화하는지를 나타낸다.



[그림 7] 협동 다이어그램

[Fig. 7] Collaboration Diagram

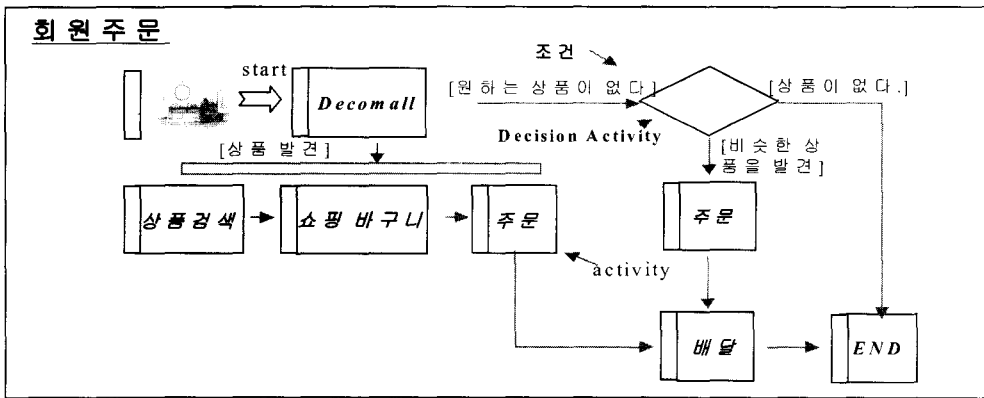


[그림 8] 상태 다이어그램

[Fig. 8] State Diagram

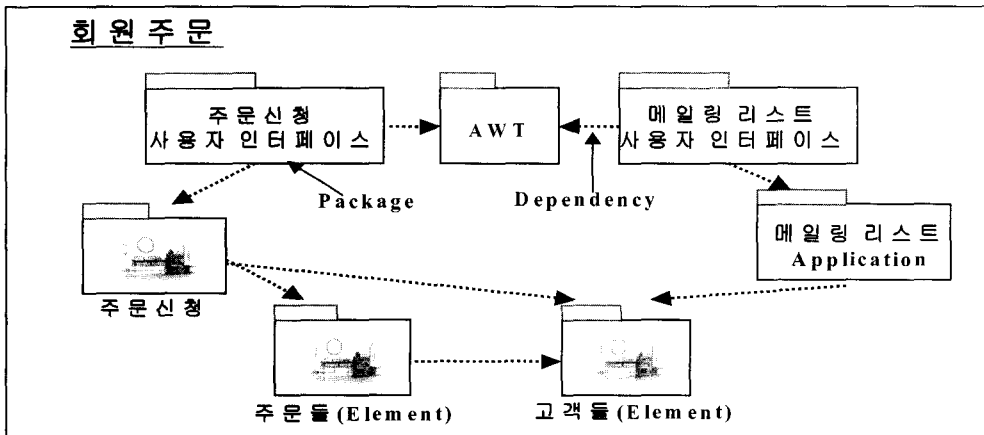
[그림 9]의 활동 다이어그램은 시스템의 동적인 부분을 모델링하는 다이어그램으로 활동들 사이의 제어 흐름을 보여 준다. 이 다이어그램은 상호작용을 표현하는 또 다른 방법이며, 이를 통해 액션이 어떻게 이루어지며 무엇을 수행하고 어디에서 발생하는지를 알 수 있다. 이 다이어그램을 이용하면 프로세서를 순차적인 혹은 병행적인 단계로 수행시킬 수 있다.

하드웨어 시스템들은 각각 고유한 특성을 가지는데, 이런 환경에서 시스템 구축은 시스템마다의 하드웨어, 소프트웨어, 소프트웨어 요소들의 관계를 나타낸 배치 다이어그램을 필요로 한다. 또한 거대한 소프트웨어를 기능적으로 분해하여 클래스들의 상위 개념을 그룹화하는 방법을 패키지라 하며 이는 클래스가 아니라 시스템에서의 모델링 원소이다. 아래 [그림 10]은 회원주문을 위한 패키지 다이어그램을 보여준다.



[그림 9] 활동 다이어그램

[Fig. 9] Activity Diagram



[그림 10] 패키지 다이어그램

[Fig. 10] Package Diagram

4. 자동 생성 시스템의 설계

UML 다이어그램을 이용하여 어플리케이션을 모델링하고 이를 자동 변환하기 위하여 BizWiz와 Rational Rose 2000을 이용하여 설계하고 구현한다.

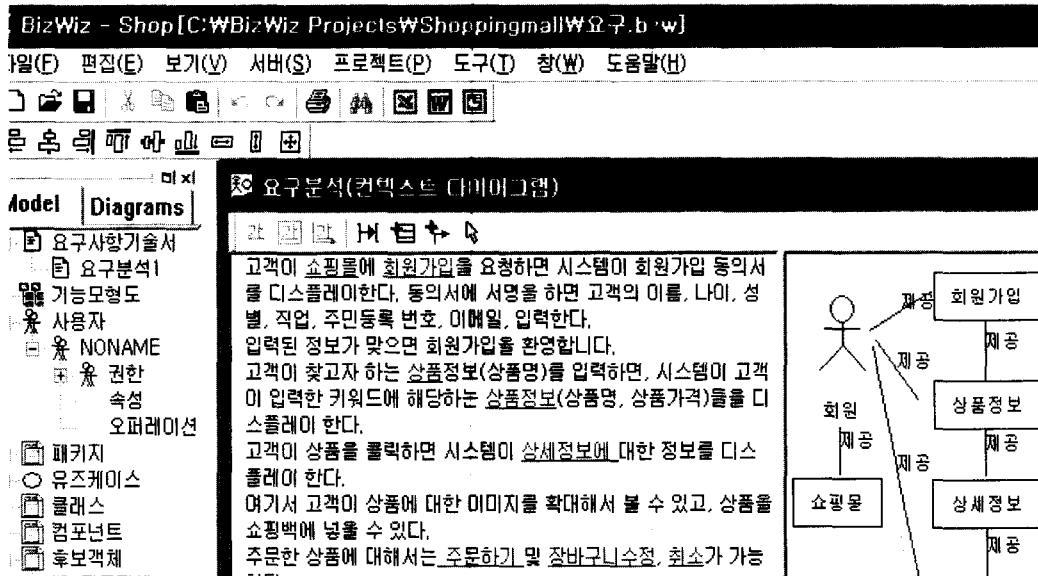
4.1 BizWiz

4.1.1 요구분석

[그림 11]과 같은 요구분석 명세서의 작성은 현 시스템의 문제점을 파악하고 시스템 개발을 위한 범위를 설정하여 해결 방안을 모색하는 것이다. 실제 개발되어질 업무 내용을 상세하게 기술 혹은 항목들을 요약 기술함으로써 사용자의 요구사항 즉, 웹브라우저를 통해 회원가입, 상품검색, 주문하기, 상품등록 등을 할 수 있도록 하였고, 데이터베이스에 구축된 자료를 검색 및 관리할 수 있도록 하였다.

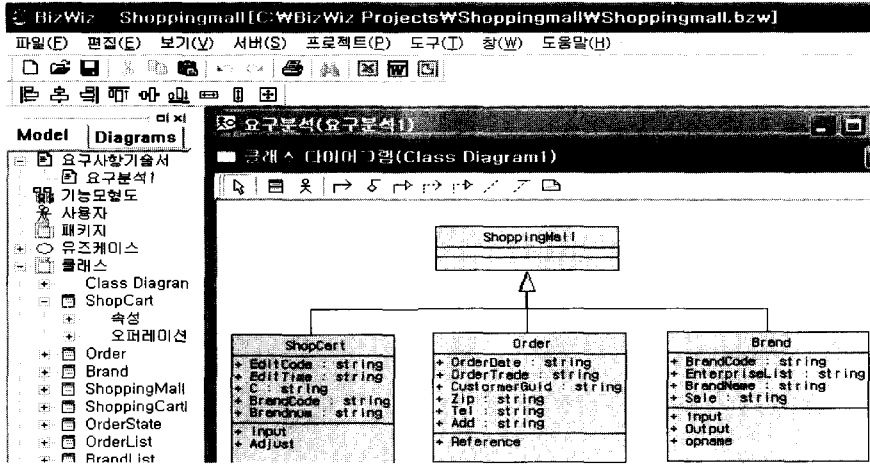
4.1.2 정적 구조 다이어그램

[그림 12]는 객체의 타입인 클래스를 표현하는 다이어그램으로 클래스의 속성과 연산 그리고 연관, 합성, 위임, 일반화, 패키지 등의 다른 클래스들과의 정적 관계에 대한 제약 등을 표현할 수 있다. 이러한 클래스 다이어그램의 목적은 다른 동적인 다이어그램에서 보여진 특징에 대해 클래스를 정의하는데 있다.



[그림 11] 쇼핑몰 시스템의 요구분석 명세서

[Fig. 11] Request Analysis Specification of Shopping Mall System

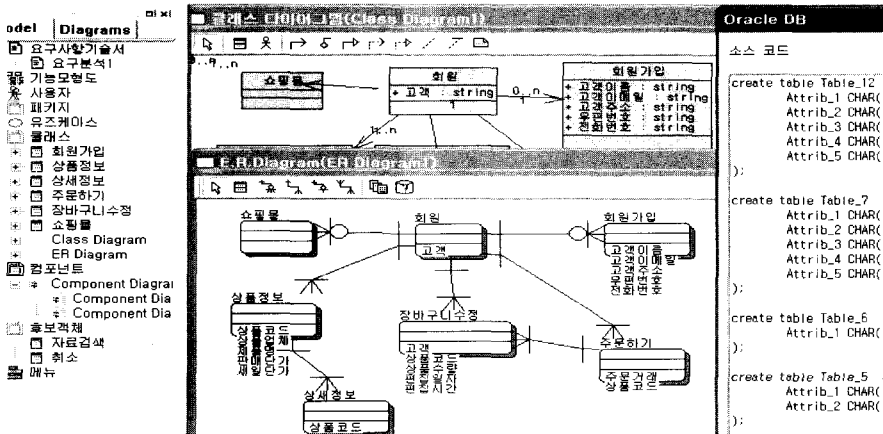


[그림 12] 클래스 다이어그램
[Fig. 12] Class Diagram

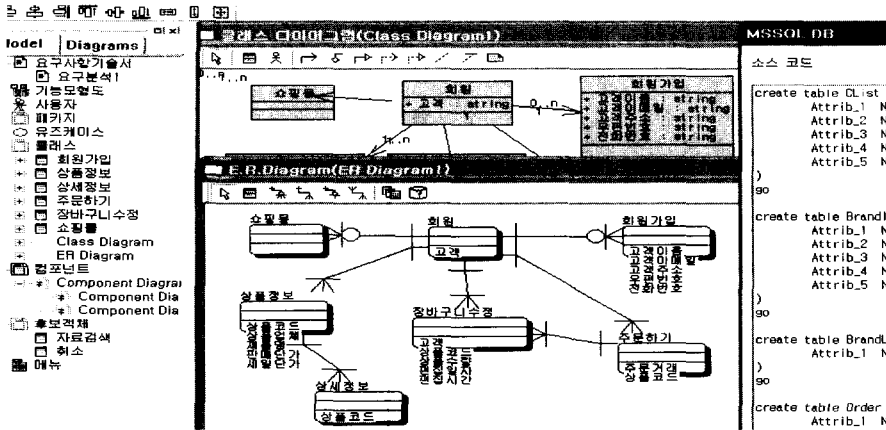
4.1.3 데이터베이스

[그림 13]은 객체, 관계 그리고 속성 등을 설정한 후에 기본적인 테이블을 생성하기 위한 DDL 코드로 Oracle을 선택한 모습이며, 여기서 ERD(Entity-Relationship Diagram)은 개체 관계성 데이터 모델을 그림으로 요약한 것이다.

[그림 14]는 기본적인 테이블을 생성하기 위한 DDL 코드로 MS SQL을 선택한 모습이다. ERD에서 관계는 클래스 다이어그램의 연관을 나타낸다.



[그림 13] Oracle DB
[Fig. 13] Oracle Database



[그림 14] MS SQL DB
[Fig. 14] MS SQL Database

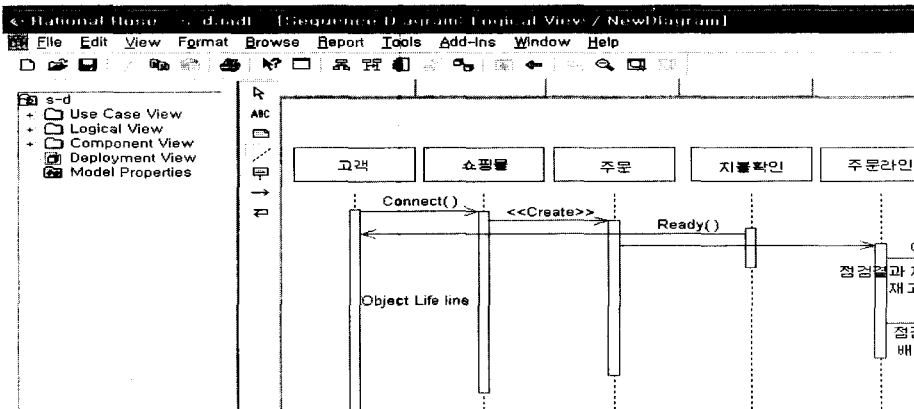
4.2 Rational Rose 2000

4.2.1 상호작용 다이어그램

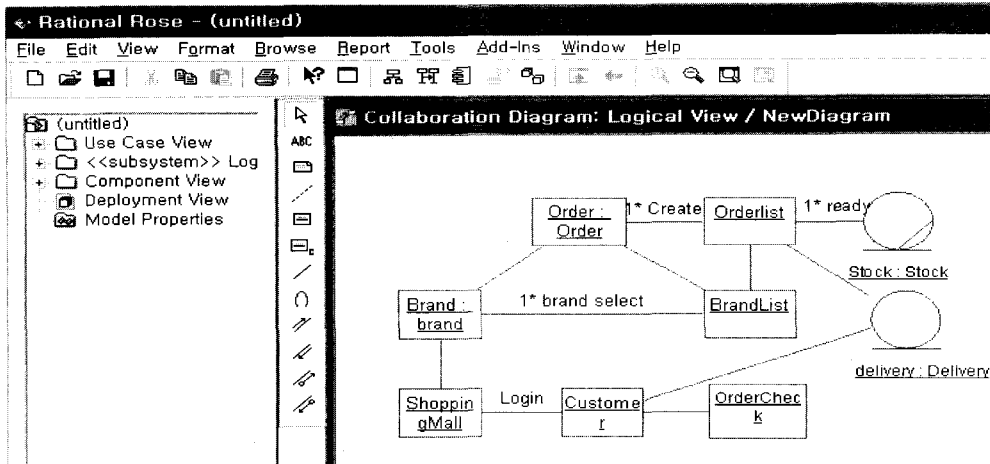
순차 다이어그램의 객체는 밑줄이 있는 객체의 이름을 포함하는 사각형으로 나타낸다. 각 객체는 객체 밑의 점선으로 표시되는 시간선이 있으며, 특히 종좌표축으로 시간개념을 도입하고 횡 좌표축으로 객체를 나열하여 그 사이의 상호작용을 표시한다. [그림 15]는 고객 객체에서 카드회사 객체까지의 시

간의 흐름을 나타내어 작성한 순차 다이어그램이다.

[그림 16]과 같은 협동 다이어그램은 시나리오를 표현하는 또 다른 방법이며, 이 다이어그램은 객체들 사이의 상호작용과 연결을 나타낸다. 사각형으로 표시되는 객체와 객체들을 연결하는 선으로 표시된다. 순차 다이어그램과 협동 다이어그램의 동적인 부분을 모델링하는 다이어그램으로 상호작용 다이어그램이라고도 하며, 특정한 객체 집합의 동적인 특성을 표현할 수도 있다.



[그림 15] 순차 다이어그램
[Fig. 15] Sequence Diagram

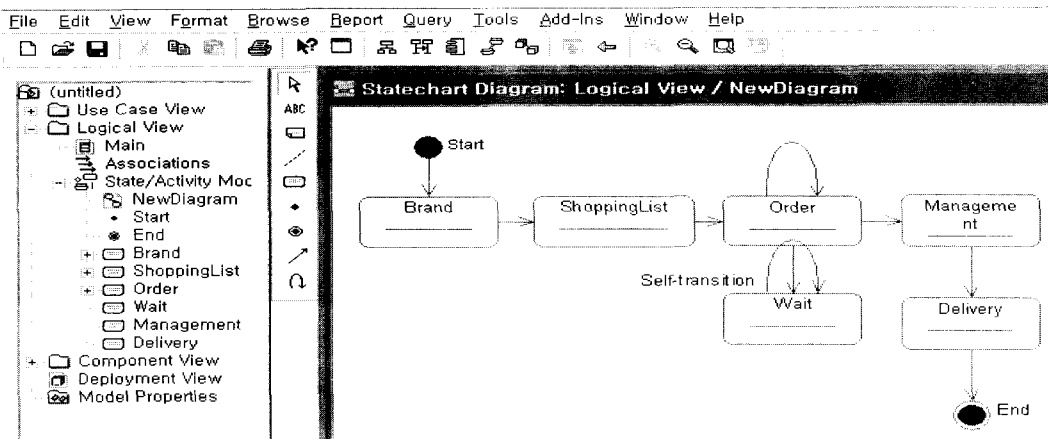


[그림 16] 협동 다이어그램
[Fig.16] Collaboration Diagram

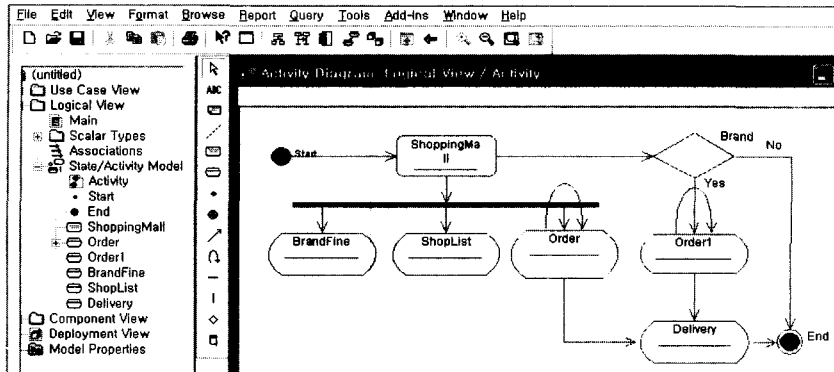
4.2.2 상태 다이어그램

상태에 대한 UML 표기는 둥근 모서리를 가진 사각형으로 표시되며 객체가 보내고 받을 수 있는 모든 메시지를 포함하며, 객체들이 보내는 두 메시지들 사이의 간격이 일반적으로 상태를 나타낸다. [그림 17]은 본 연구의 쇼핑몰을 위한 상태 다이어그램을 보여준다.

활동 다이어그램에는 활동들, 활동들 사이의 전이(transition), 결정점(decision point), 그리고 동기화 바(synchronization bar)가 나타난다. 활동은 인간이나 컴퓨터에 의해 수행이 필요한 어떤 작업을 의미한다. [그림 18]은 본 연구의 쇼핑몰을 위한 활동 다이어그램을 보여준다.



[그림 17] 상태 다이어그램
[Fig. 17] State Diagram



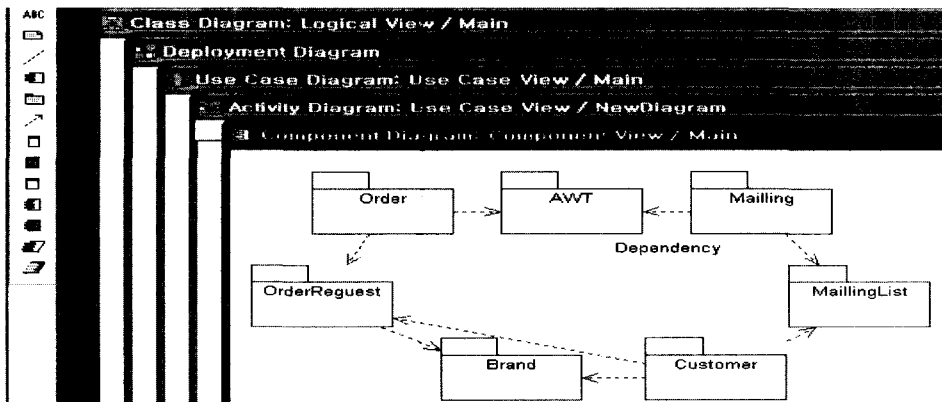
[그림 18] 활동 다이어그램
[Fig. 18] Activity Diagram

4.2.3 구현 다이어그램

아키텍처의 구현 부분은 개발 환경내에서의 실제 소프트웨어 모듈의 구성과 관련된다. 또한 이 부분은 개발의 용이성, 소프트웨어 관리, 재사용, 프로그램 언어와 개발 도구에 따른 제약조건과 관련되어 파생된 요구를 할 수 있다. 모델의 'Component View'에서 소스 코드 요소는 패키지에 포함된 소프트웨어 파일을 나타낸다. 예를 들어 C++에서는 .h와 .cpp 파일을 나타내고, Java에서는 .java 파일로 나타낸다. [그림 19]은 본 연구의 쇼핑몰을 위한 요소 다이어그램을 보여준다.

4.3 JBuilder 7

프로젝트에서 사용하는 클래스와 패키지를 UML 다이어그램 형태로 시각화하여 보여줌으로써 프로젝트의 오류를 줄일 수 있고, 모델 분석이 쉽게 이루어지며, 클래스와 클래스, 패키지와 패키지 사이의 관계를 이해하여 개발을 쉽게 할 수 있도록 도와준다.



[그림 19] 요소 다이어그램
[Fig. 19] Element Diagram

[그림 21]은 [그림 20]의 JBuilder 소스 코드를 UML 형태로 자동으로 변환시킨 모습이며, UML 탭을 클릭하면 현재 프로젝트가 가지고 있는 클래스 관계를 파악할 수 있도록 클래스 다이어그램 형태로 보여준다. 하지만 이 기능을 사용하려면 우선 컴파일 되어 있어야 한다.

5. 평가 및 결론

5.1 평가

정보 시스템이 점차 다양화, 복잡성, 분산, 웹기반 환경으로 옮겨가면서 분석, 설계 그리고 구현의 필요성이 강조되고 있다. 본 연구에서는 실제로 쇼핑몰을 운영할 때 필요한 명세서 내용과 구현 결과의 일치성을 도모함으로써 쇼핑몰 시스템이 발생시킬

```

package first1;

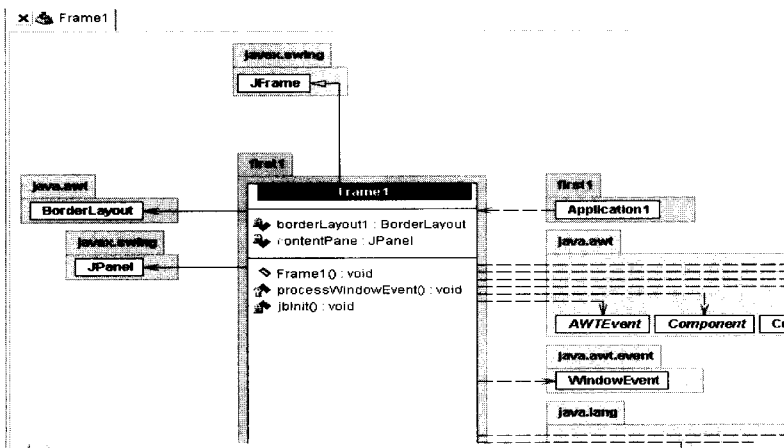
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

/**
 * <p>Title: </p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2002</p>
 * <p>Company: </p>
 * @author unascribed
 * @version 1.0
 */

public class Frame1 extends JFrame {
    private JPanel contentPane;
    private BorderLayout borderLayout1 = new BorderLayout();
}
    
```

[그림 20] JBuilder 7 소스 코드

[Fig. 20] JBuilder 7 Source Code



[그림 21] JBuilder 7 소스 코드의 시각화

[Fig. 21] Visualization anger of JBuilder 7 Source Code

수 있는 문제점을 찾아 해결 방안을 제시하였고, 자동화 도구를 이용함으로써 단위업무에서부터 통합업무까지 사용자 위주의 인터페이스로 업무 생산성을 극대화할 수 있고 시간과 비용을 절감할 수 있어 일반 기업들이 위험부담 없이 이를 도입할 수 있을 것이다.

한편 Rational Rose는 'Visual Modeler'의 기능과 순차 다이어그램 및 협동 다이어그램을 통하여 비즈니스 요구사항 분석, 시나리오 분석과 같은 동적 행동 분석을 수행한다. 또한 코드 생성 기능, Rose 도메인으로서의 접근을 제공하는 스크립트 확장 기능이 있으며 소프트웨어 개발자에게 실제로 필요한 요구들에

대해 효율적인 해결책을 제시한다. 한편 BizWiz는 사용자 컴퓨팅을 위한 객체지향 CASE 도구로써 개별 업종에 맞춘 전용 도구이며 업무 흐름을 작성하면 Asp, Java 등의 소스 코드와 데이터베이스의 스키마, 엔터티 등이 자동으로 생성되므로 시간과 비용을 절감할 수 있는 도구이다.

아래의 표2는 근래의 어플리케이션 개발 도구들에 대해서 언급 하였으며 그것들은 UML 도구들은 기본적으로 디자인 패턴을 지원하고 있으며, 소스 코드도 자동으로 생성해주는 기능을 가지고 있다. 이런 도구와 경쟁하려면 어플리케이션 개발 도구들은 코드의 시각화 및 자동화 그리고 디자인 패턴을

<표 2> UML 코드 시각화의 비교
 <Table 2> Comparison of UML Code Visualization

	JBuilder	WebGain	Visual Age WSAD
UML Code Visualization	개발자 중심의 도구로써 코드 시각화 기능을 제공	Studio 제품에는 UML 모델링 기능은 있지만 자체에 내장된 기능은 아님. 코드 시각화 기능은 없음	Rational 제품과 연계하여 UML 모델링 기능을 이용할 수 있음. 코드 시각화 기능은 없음

<표 3> CASE Tool의 기능 비교
 <Table 3> Functional Comparison of CASE TOOL

항목 / 제품명	Rose2000	Together	BizWiz
제작사	Rational	TogetherSoft	Mi Logics
개발언어 및 방법론	UML/객체지향	UML/객체지향	UML/객체지향
리포팅 서식변경 지원	없음	자체도구	XSL
작업 데이터 저장	자체포맷(TEXT)	자체포맷	XML
클래스 설계	O	O	O
Usecase 설계	O	O	O
Component 설계	O	O	O
전 Life Cycle 지원	O	O	O
자동문서화 도구지원	O	O	O
분산저장	O	O	O
Repository 관리	X	O	O
산출물 Navigation	O	X	O
Configure관리지원 (커스텀마킹)	X	X	O
Project Collaboration	X	X	O
ERDiagram	클래스 형식으로 표시	미확인	Erwin 형태 (소프트공학 방법론에 따름)
비즈니스 로직 위저드 기능	X	X	O
기본정보 리포팅 지원	*.doc, *.html	*.html	*.xml, *.xsl

적용하여 시스템을 모델링할 수 있는 기능을 가져야 한다. 아래의 비교는 UML의 관점에서 비교하였으며 다른 기능의 비교는 제외하였으며 표3은 UML 전용 어플리케이션의 비교 분석을 하였다.

5.2 결론 및 연구과제

최근 인터넷 기술과 관련 응용 프로그램이 급성장하면서 관심이 증폭되고 있으나 아직까지는 효율성 부족, 단순 개발인력의 소요로 생산성 저하 및 개발 비용증가의 문제점을 안고 있으며, 이를 해결하기 위한 많은 연구가 진행 중이다. 이러한 이유로 정보 시스템들은 인터넷 관련 기술과 관련 업무의 자동화 도구의 필요성을 느끼고 있는 실정이다. 본 연구에서는 자동 생성 시스템들을 실제의 응용 프로그램에 적용하여 UML 관점에서 분석하였고, 특히 고객관리 업무에 UML을 적용하여 체계적인 소프트웨어 분석 및 설계 방법에 대해 살펴보았다.

정보화의 도입에서부터 활용까지 생산적이며 효율적인 시스템을 구축하기 위해서는 빠르고 정확한 업무분석과 기업문화에 적합한 개발 방법이 필요하다. 또한 소프트웨어 개발팀의 구성원 각자가 프로그램을 작성하고, 서로 협동적으로 통신하여 보다 나은 소프트웨어를 개발할 수 있는 방법론 그리고 개발 노력을 단순화시키고 개발 속도를 최대화시키는 시스템과 방법론이 필요하다. 마지막으로 비용을 줄이고 시장에서의 경쟁력을 높일 수 있도록 소프트웨어 과정이 제어되고 관리되어 검증될 수 있도록 하는 방법이 개발되어야 할 것이다.

※ 참고문헌

- [1] 엄근석, 안성아, 최희석, "J2EE 어플리케이션 모델 기반의 컴포넌트 저장소 구현", 한국멀티미디어학회논문지:5(1), 2002.
- [2] 정진영, 송정길, "UML 기반의 웹 어플리케이션 자동생성 시스템", 한국컴퓨터산업교육학회논문지:3(5), 2002.
- [3] BizWiz, <http://www.milogics.com/>.
- [4] Grady Booch, "Object-Oriented Analysis and Design", 2nd Edition, Benjamin/Cummings, 1994.
- [5] Grady Booch, J. Rumbaugh, and I. Jacobson, "The Unified Modeling Language User Guide", Addison-Wesley, 2001.
- [6] Ivar. Jacobson, "Object-Oriented Software Engineering: A Use Case Driven Approach", Addison-Wesley, 1999.
- [7] JBuilder, <http://info.borland.com/devsupport/jbuilder/>.
- [8] James. Rumbaugh, et al., "Object-Oriented Modeling and Design", Prentice Hall, 1991.
- [9] OMG, "OMG Unified Modeling Language Specification Ver. 1.3", 1999.
- [10] Rational Rose, <http://www.rational.com/products/rose/index.jsp>.

한 현 관



현재 대구산업정보대학 교수
2002 ~ 현재 영남대학교 컴퓨터공
학과 박사과정
관심분야 : XML, UML, 웹서비스

이 명 진

계명대학교 컴퓨터공학부 박사
2003년 현재 아시아전통과학대학
교 인터넷비즈니스과 교수