

UML 기반의 웹 어플리케이션 자동 생성 시스템 (Web Application Generation System Based on UML)

정진영* 송정길**
(Jin-Young Jung) (Jung-Gil Song)

요 약

웹 기술이 발전함에 따라 웹 어플리케이션은 기업의 서비스들을 통합하기 위한 기반으로 자리를 잡아가고 있다. 그러나 대부분의 웹 어플리케이션의 개발은 단순히 개발자의 능력에만 의존하여 개발되고 있는 실정이다. 웹 어플리케이션의 개발시 웹의 특성을 고려한 소프트웨어 공학적 접근 및 모델링의 필요성이 대두되고 있다. 본 논문에서는 UML과 UML 확장을 이용하여 웹 어플리케이션을 모델링한 후 페이지를 추출하여 자동으로 웹 어플리케이션을 생성한다. 웹 페이지를 추출하여 자동으로 생성함으로써 대규모의 웹 어플리케이션 개발시 유지 및 보수가 용이하도록 하고자 한다.

ABSTRACT

As web technology is developed, web application becomes the basis which combines business services. However, most of the web applications have depended upon the capability of the web developer. When web application is developing, the property of the web should be considered in the ways of approach and modeling. In this paper, after modeling web application using UML and the extend of UML, web pages are extracted automatically. Thus, web application developing is easy to be maintained and to be repaired on a large scale.

1. 서 론

초기의 웹 어플리케이션 개발 시 시스템에 대한 분석과 설계 단계는 그리 중요하지 않았다. 이는 웹 구조가 간단하여 오히려 시간과 자원을 낭비하는 결과를 초래하기 때문이다. 그러나 최근 몇 년 사이에 인터넷과 컴퓨터 환경이 급격히 발전하면서 웹 어플리케이션의 응용 분야가 확대됨에 따라 웹 어플리케이션의 개발시 시스템에 대한 분석과 설계 및 표준의 웹 모델링을 이용하여 유지 및 보수에 대한 비중이 매우 높아지고 있는 현실이다[1].

이러한 위기의식의 원인은 네트워크와 인터넷 기술이 불과 몇 년 사이에 눈부신 발전을 거듭해 온 것에 비하여 웹 개발과 관련된 기반 기술은 별로 개선된 것이 없기 때문이다. 이로 인하여 오늘날에도 몇 년 전과 비교했을 때 여전히 다수의 개발인력이

이러한 위기의식의 원인은 네트워크와 인터넷 기술이 불과 몇 년 사이에 눈부신 발전을 거듭해 온 것에 비하여 웹 개발과 관련된 기반 기술은 별로 개선된 것이 없기 때문이다. 이로 인하여 오늘날에도 몇 년 전과 비교했을 때 여전히 다수의 개발인력이

* 정희원 : 대전보건대학

** 정희원 : 한남대학교

논문접수 : 2002. 4. 25

심사완료 : 2002. 5. 13

투입되어 유사한 방법으로 웹 어플리케이션을 개발하며 막대한 자금을 투입해 유지보수 작업을 수행하고 있다. 이에 대한 해법으로 전 세계적으로 웹 어플리케이션 개발시 소프트웨어 공학적 접근과 모델링에 대한 관심이 크게 고조되고 있으며, 또한 웹 어플리케이션 개발시 객체 지향 기술(Object-Oriented Technology)을 접목하려는 노력이 이루어지고 있다. 객체 지향 기술은 추상화, 캡슐화 및 상속성 등의 개념을 기반으로 확장성과 재사용성을 높이는 핵심 기술로서 현재의 소프트웨어 개발뿐만 아니라 웹 어플리케이션 개발 관련기술에도 많은 연구대상이 되고 있다[2].

본 논문에서는 이러한 시점에서 웹 어플리케이션 개발 시 객체지향기술의 표준화인 UML과 UML 확장을 기반으로 웹 어플리케이션을 모델링하고 이를 실제 웹 기술과 접목하여 자동으로 웹 어플리케이션을 생성하여 현재 웹 어플리케이션 개발 시 대두되고 있는 웹 위기의 문제점들을 해결하고자 한다.

2. 관련 연구

최근 어플리케이션의 개발환경은 기존의 업무를 웹 환경에서 사용할수 있도록 확장되고 있는 추세이다. 이러한 어플리케이션은 웹 환경만이 가지고 있는 특수한 환경으로 인하여 기존의 개발방법만으로는 웹 어플리케이션에 적용하기에 부족하다. 이에 따라 여러 가지 개발 방법들이 현재 나와 있다.

RMM(Relationship Management Method)은 웹 어플리케이션의 설계, 구축 및 관리를 위한 방법론으로 동적 DB를 유지하는데 소요되는 비용을 감소하고자 하는데 목적이 있다. RMM은 웹 어플리케이션의 분석보다는 DB를 지원하는 동적 웹사이트를 생성하고 유지하기 위한 넓은 범위의 접근법으로 단지 시스템 개발에 어울리는 엔티티와 관계에 기반하는 비즈니스 로직중심으로 현재 웹 어플리케이션에 적용되는 스크립트, 애플릿, ActiveX와 같은 기술들을 포함하는 웹 어플리케이션을 관리하기에 그 한계가 있다[3].

OOHDM은 웹 어플리케이션을 개발하기 위한 방법으로 개념 설계, 네비게이션 설계, 추상인터페이스 설계, 구현의 4개의 프로세스를 정의하여 복잡한 정

보들을 간결하게 기술할 수 있도록 해주는 객체지향 프레임워크로, 디자인 단계에서의 모듈성과 재사용성을 제공하지만, 분석 프로세스에서의 동적인 측면보다는 정적인 측면을 강조하여, 디자인과 구현사이에 계획적인 매핑을 지원하지 못한다는 단점을 지니고 있다.

Web Composition은 웹에서 나타나는 요소들을 상태와 행위를 가진 컴포넌트 객체로 표현한다. 이러한 컴포넌트는 링크, Perl Script와 같은 요소들의 모델링이 가능하다. WebComposition[7]은 파일 기반의 요소들을 객체 기반의 컴포넌트로 모델링하여 웹 컴포넌트의 재사용과 관리를 가능하게 하지만, 실시간 환경에서 웹 어플리케이션의 실제 행위를 모델링하지 못한다는 단점을 가지고 있다.

3. 시스템 분석 및 모델링

3.1 시스템 분석

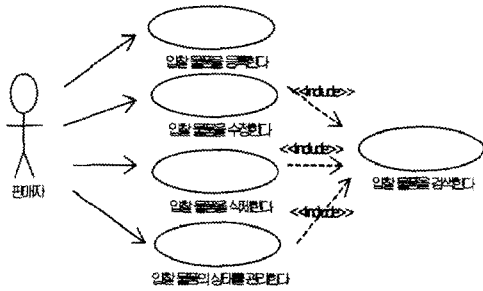
본 논문에서 UML을 이용하여 자동으로 생성될 웹 어플리케이션인 입찰 구매 시스템에 대한 시나리오인 구매자와 판매자 입장에서 분석하여 구성하였다.

(1) 구매자 입장

입찰 공고의 수정, 삭제, 재입찰을 통하여 입찰 관리가 이루어지며, 응찰 내용을 조건에 따라 검색하여 현재 응찰 현황을 보여주거나 낙찰을 결정하도록 한다. 낙찰을 결정하면 자동적으로 입찰 공고란이 낙찰 결정화면으로 바뀌고 판매자에게 통보한다.

(2) 판매자입장

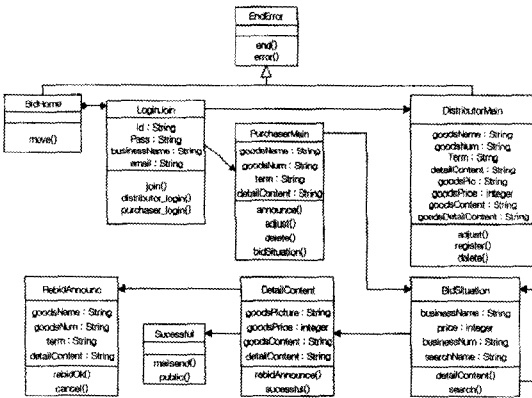
판매자는 입찰 상품을 관리하기 위해서 판매자의 신원을 확인함으로써 시작된다. 입찰 상품에 대하여 등록, 수정, 삭제, 조회를 할 수 있으며, 구매자로부터 상품에 대한 낙찰이 결정됨으로써 시스템이 종료 된다.



[그림 1] 유즈 케이스 다이어그램
[Fig. 1] Use Case Diagram

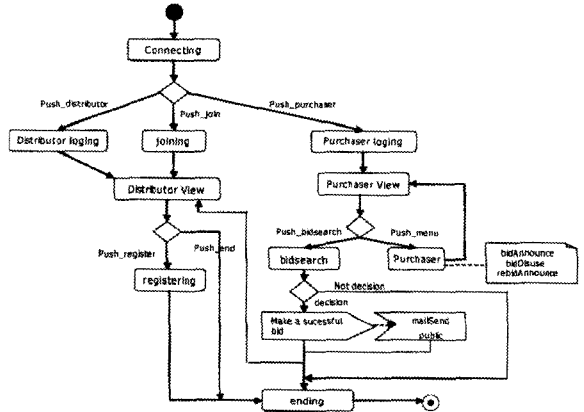
3.2 시스템 모델링

요구사항 분석단계를 기반으로 각 단계에서 필요한 클래스를 추출하게 되는데, 구매자와 판매자의 입장에서 객체 클래스를 [그림 2]와 같이 구성하였다.



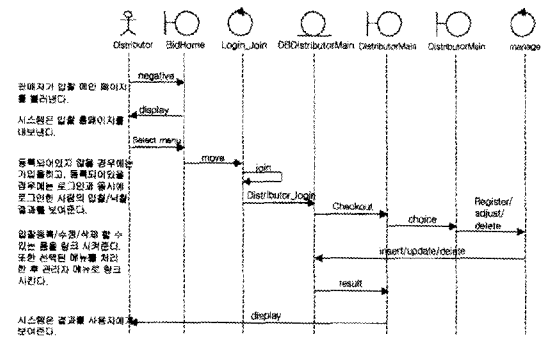
[그림 2] 도메인 클래스 구조
[Fig. 2] Domain Class Structure

액티비티 다이어그램을 이용하여, 입찰정보시스템에서 서버의 작업처리 동작을 표현한다. 액티비티 다이어그램에서는 웹 페이지 연결시 상호 작업 처리시 동작에 대하여 판매자 입장에서의 처리작업을 [그림 3]과 같이 구성하였다.



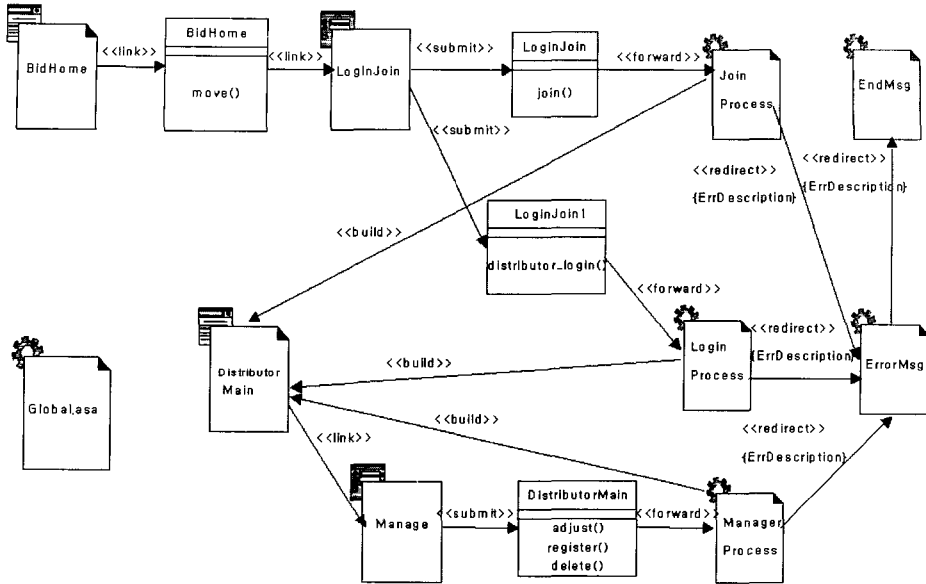
[그림 3] 액티비티 다이어그램
[Fig. 3] Activity Diagram

시간적 흐름에 의한 시스템의 데이터 처리 과정을 시퀀스 다이어그램으로 판매자가 접속한 후 처리 과정 및 처리시에 필요한 메시지 등을 시간의 흐름에 따라 시퀀스 다이어그램을 이용하여 [그림 4]와 같이 구성하였다.



[그림 4] 판매자 입장의 시퀀스 다이어그램
[Fig. 4] Sequence Diagram of Distributor Side

본 논문에서 구현하고자 하는 전체적인 웹 어플리케이션을 각각의 다이어그램을 기반으로 기존의 페이지 단위의 모델링을 사용하여 stereotype, tagged value, constraint와 같은 UML 확장 메커니즘을 이용하여 Web 고유의 아키텍처 요소들로 모델링을 하면 [그림 5]와 같다.



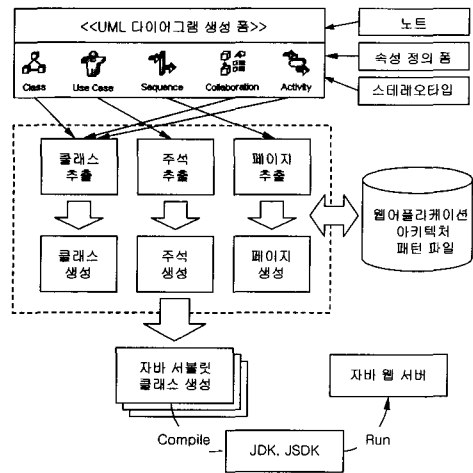
[그림 5] UML 확장을 이용한 시스템 모델링
 [Fig. 5] System Modeling Using UML Extension

4. 자동설계시스템의 설계

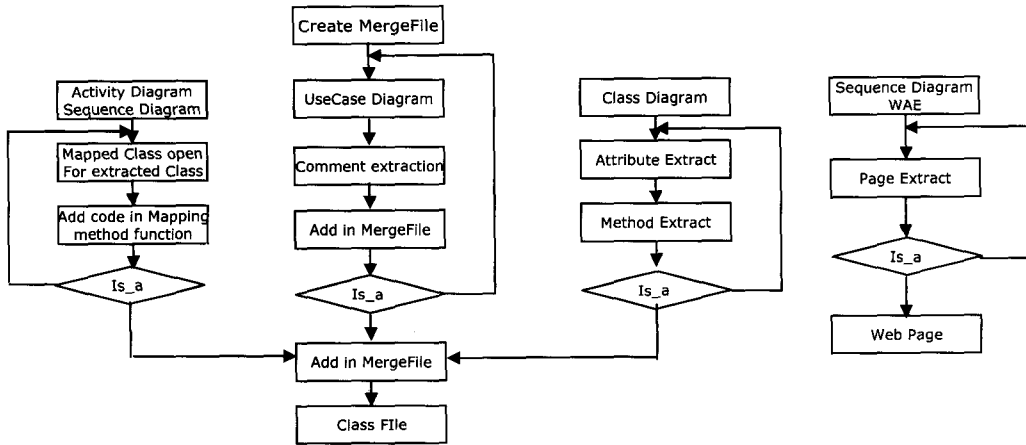
4.1 시스템 구성

UML 다이어그램을 이용하여 어플리케이션을 모델링하고, 이를 웹 어플리케이션으로 자동으로 변환하기 위한 UML-웹 어플리케이션 생성기를 [그림 6]과 같이 설계하고 구현하였다.

UML-웹 어플리케이션 생성기는 UML에서 제공하는 표준다이어그램을 모두 수용할 수 있도록 다이어그램 생성 모듈과 각각의 다이어그램으로부터 최종적으로 자동 생성될 자바 서블릿 코드에 필요한 코드들을 추출하는 모듈, 추출된 내용을 기반으로 실제 코드로 생성한 후 이를 최종적으로 자바 서블릿 코드로 변환하는 구조로 설계되었다.



[그림 6] UML-웹 어플리케이션 자동 변환기 구성도
 [Fig. 6] Configuration of UML-Web Application Generation

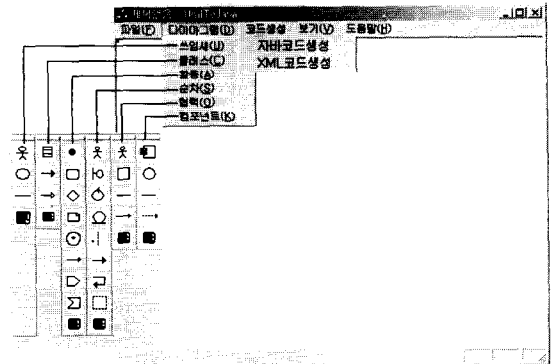


[그림 7] 자동 생성기 흐름도

[Fig. 7] Flow of Generation

[그림 7]은 웹 어플리케이션 프로그램을 생성하기 위한 자동생성기 흐름도로서 실제 클래스의 소스코드가 기록될 MergeFile을 생성한 후 클래스 다이어그램에 정의된 클래스의 속성과 메소드에 의하여 클래스를 MergeFile에 기록한다. 작성된 클래스에는 각각의 클래스들의 용도를 유즈 케이스 다이어그램에서 추출한 내용이 주석화 하여 함께 기록된다. 유즈 케이스 다이어그램과 클래스 다이어그램을 통하여 각각의 클래스는 패키지화하여 나중에 생성될 실제 웹 페이지에서 사용하게 된다. 작성된 클래스는 단순히 속성과 메소드만을 정의하기 때문에 실제 클래스가 내부적으로 어떠한 작업을 수행할 수 있도록 액티비티 다이어그램을 이용하여 발생하는 일련의 작업을 클래스내에 삽입하게 된다. 최종적으로 시퀀스 다이어그램을 통하여 웹 어플리케이션에서 실제 생성될 페이지들을 추출하여 패키지화된 클래스들을 import 하여 실제 동작이 수행되도록 웹 페이지를 생성한다.

본 시스템은 Microsoft Visual C++과 JDK 1.3, JSDK, Jakarta Tomcat을 이용하여 구현하였다. 본 시스템은 [그림 8]과 같은 메뉴로 구성되어 있으며, 기본적으로 UML 다이어그램을 작성하는 다이어그램 메뉴와 생성된 다이어그램을 실제 코드로 변환하는 변환 메뉴, 생성된 코드를 확인 및 수정할 수 있는 보기 메뉴로 구성되어 있다.



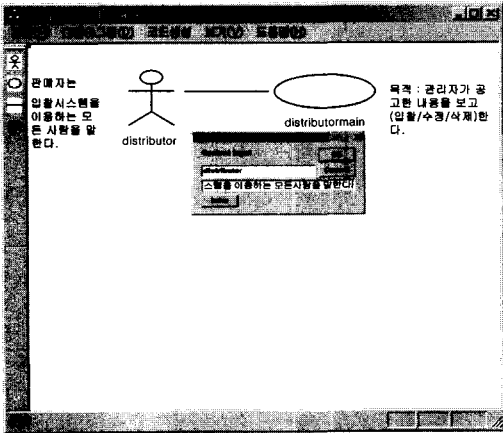
[그림 8] 메뉴 구성

[Fig. 8] Configuration of Menu

4.2 유즈 케이스의 선언

유즈 케이스 다이어그램은 시스템의 용도를 나타내는 것으로 유즈 케이스는 타원으로 표현하고 actor는 막대인간으로 표현하여 생성될 웹 어플리케이션의 사용용도를 표현하게 된다. [그림 9]는 유즈 케이스 다이어그램의 예를 나타내고 있다. 유즈 케이스 다이어그램을 이용하여 클래스의 용도 및 역할에 대한 주석을 작성하게 된다. 작성된 주석은 MergeFile에 위치하게 되며, 추후 클래스 다이어그램에서 실

제 클래스의 내용과 함께 MergeFile내에 삽입된다.



[그림 9] 유즈케이스 다이어그램
[Fig. 9] Use Case Diagram

액터와 액터에 대한 comment를 추출하여 MergeFile에 코드가 추가한 후 관계가 있는 유즈 케이스, 유즈케이스 comment 또한 추출하여 코드에 추가된다. 정의되어진 액터와 유즈 케이스가 처리완료 될 때까지 위 과정을 반복하게 되며 처리가 완료되면 프로그램에 대한 전반적인 내용을 이해하기 쉽게 해주기 위한 주석처리가 완료된다. 유즈 케이스 다이어그램으로부터 생성된 주석은 [그림 10]과 같다.

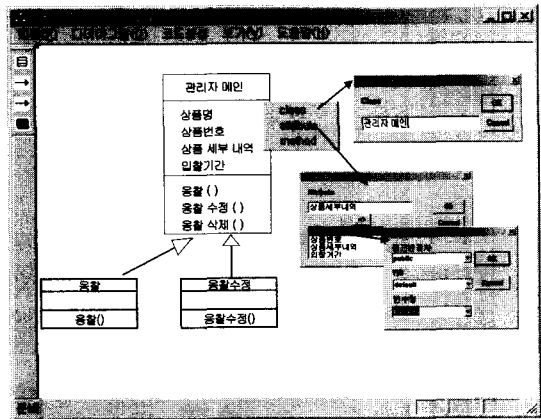
```
//distributor
//판매자는 입찰 시스템을 이용하는 모든 사람을 말한다.
//distributorMain
//목적 : 관리자가 광고한 내용을 보고 (입찰/수정/삭제)한다.
```

[그림 10] 유즈 케이스 다이어그램으로부터 생성된 MergeFile

[Fig. 10] The MergeFile From Use Case Diagram

4.3 클래스 선언

클래스 다이어그램은 시스템 혹은 도메인 내의 개체와 그 개체간의 관계를 나타내는 것으로 클래스를 표현하기 위해서 사각형을 그리고 그 안에 클래스의 이름과 attribute를 작성한 후 마지막으로 method를 입력하여 작성하게 된다. 클래스는 자바 어플리케이션의 기본으로서, 클래스 다이어그램을 이용하여 객체를 생성할 수 있는 클래스를 선언한다. 클래스에 정의된 속성들은 페이지 시작 부분에 자바 변수(필드)로 선언하여 매핑 되며, 이들 변수는 페이지 범위를 가지게 되며, 페이지 안에 선언된 모든 메소드에서 접근이 가능하며, 자바 메소드는 오퍼레이션으로 매핑된다. [그림 11]과 같이 DistributorMain 클래스는 판매자가 응찰, 수정, 삭제할 수 있도록 기본적인 구조를 제공하며, 클래스를 생성하기 위해 클래스명 그리고 각각의 필드명, 함수명과 접근 제한자, 자료형, 리턴형 등 필요한 모든 자료들을 다이어그램에서 입력받아 클래스를 생성하게 된다.



[그림 11] DistributorMain Class 다이어그램
[Fig. 11] DistributorMain Class Diagram

클래스는 attribute와 method를 추출하여 시스템에서 생성한 MergeFile에 코드를 추가하게 되며 상속 관계가 정의되어 있는가에 따라 상속된 클래스 내용을 추가시킨다. 클래스 다이어그램에서 나타난 모든 클래스가 완료될 때 까지 위 과정을 반복하며 완료

되면 각 클래스 단위로 MergeFile로 분배하여 각각의 분배된 클래스를 패키지와 한다. [그림 12]의 소스는 [그림 13]의 클래스 다이어그램으로부터 추출된 DistributorMain 클래스이다.

```
//distributor
//판매자는 입찰 시스템을 이용하는 모든 사람을 말한다.
//distributorMain
//목적 : 관리자가 공고한 내용을 보고
(입찰/수정/삭제)한다.

public class DistributorMain {

    String goodsName;
    String goodsNum;
    ...

    public void register() {}

    public void adjust() {}

    public void delete() {}

}
```

[그림 12] DistributorMain MergeFile
[Fig. 12] DistributorMain MergeFile

DistributorMain 클래스는 입찰매물의 이름을 위한 String형의 데이터타입인 goodsName과 같이 입찰매물정보를 위한 변수들을 가지게 되며, 이러한 입찰매물의 등록 작업을 위한 register(), 수정을 위한 adjust(), 폐기작업을 위한 delete() 메소드를 가진다.

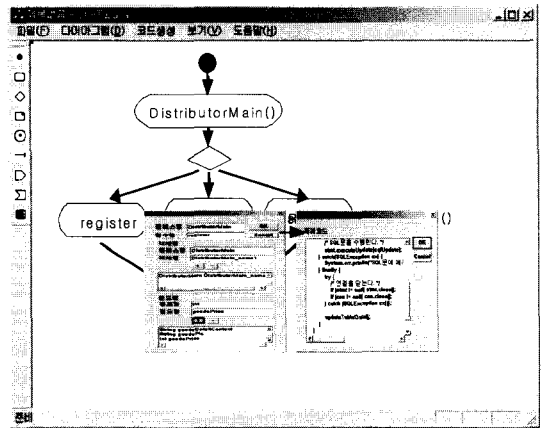
4.4 클래스 기능 확장

클래스 다이어그램은 클래스의 구조 정보만을 표현하기 때문에 액티비티 다이어그램 및 시퀀스 다이어그램을 이용하여 어떤 객체의 행동 혹은 업무 진행 과정 내에서 발생하는 진행단계와 결정위치를 추가하여 클래스에 확장된 정보를 표현하게 된다.

액티비티 다이어그램의 표시 방법은 진행단계는 모서리가 둥근 사각형으로 표현하며 결정위치는 마름모꼴로 표현하여 클래스의 기능을 확장하기 위한 용도로 사용하게 된다. 각 클래스의 연산에 대한 동

적인 행동을 결정하기 위해 유즈케이스 다이어그램을 이용한다.

[그림 13]의 액티비티 다이어그램은 DistributorMain 클래스에 대한 액티비티 다이어그램으로써 사용자의 선택을 입력값으로 받아 입력된 값에 따라 register(), adjust(), delete() 메소드로 분기하게 되며, 실제로 처리를 담당하는 메소드를 작성하기 위한 정보를 입력한다. 각 메소드들은 클라이언트측의 요구에 따라, 각 메소드를 선별하여 실행하게 된다. 이때, 필드들은 클래스 다이어그램에서 표현된 필드들을 이용하게 된다.



[그림 13] Activity Diagram
[Fig. 13] Activity Diagram

클래스에 추가된 코드들은 실제로 웹과 연동하게 될 때, 처리부분을 담당하게 된다. 소스를 생성하기 위해 활동 다이어그램으로부터 클래스와 함수를 추출하여 추출한 클래스와 맵핑된 클래스 파일을 open 하며 그 중에서 추출한 함수와 매핑된 함수 다음라인에 추가될 부분을 설정하고 또한 객체명과 필드명 필드값을 추출하여 함수에서 처리해야할 각각의 변수들을 선언과 초기화한다. 실제로 함수가 해야할 일은 추가코드에서 추출한 값으로 소스를 생성하게 된다. 액티비티 다이어그램이 완료전 까지 위 과정을 반복하게 되며 완료되어지면 아래 [그림 14]처럼 소스코드가 추출한 클래스내에 추가된다.

```

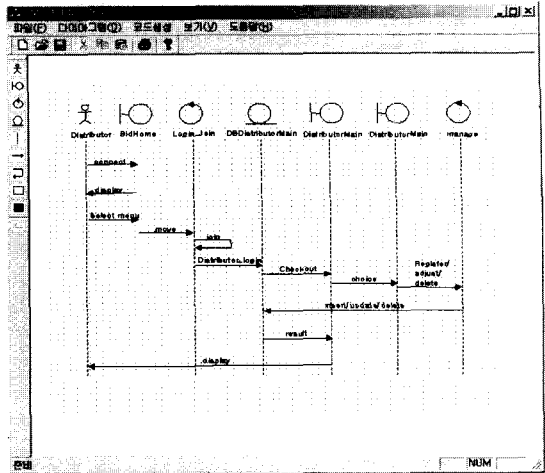
public class DistributorMain {
    ...
    public void register() {
    ...

    String sql = "INSERT INTO Distributor(goodsPrice,
    goodsPrice, goodsContent, goodsDetailContent)
    VALUES ("
    + goodsPic + ", "
    + goodsPrice + ", "
    + goodsContent + ", "
    + goodsDetailContent) ;
    doUpdateCommand(sql);
    ...
    try {
    ...
    stmt.executeUpdate(sqlUpdate);
    } catch(SQLException ex) {
    ...
    updateTableData();
    }
    }
}
    
```

[그림 14] DistributorMain Class 소스
 [Fig. 14] Source Code of DistributorMain Class

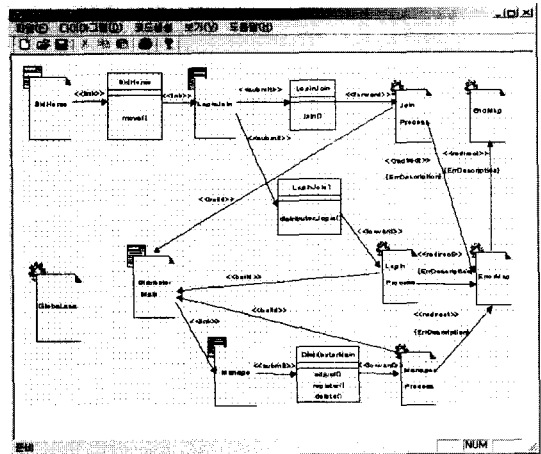
4.5 페이지 추출 및 메인 함수 생성

웹 어플리케이션은 기본적으로 페이지 단위로 프로세스가 진행하게 된다. 본 논문에서는 시퀀스 다이어그램을 이용하여 객체들간의 유기적인 관계와 객체들 간의 상호 메시지를 파악하여 웹 어플리케이션에 사용될 페이지를 추출하는 방식을 취하여 클래스 다이어그램으로부터 추출된 클래스를 추출된 페이지에서 이용하도록 구성하였다. 시퀀스 다이어그램은 어떤 시스템에 속해 있는 한 객체가 다른 객체와 어떻게 교류하는지를 시간에 걸쳐 나타낼 때 사용되는 것으로 객체는 위 부분에 위치하며 또한 시간은 위에서 아래로 흐르며 흐름을 나타내기 위하여 화살표를 사용한다. [그림 15]는 판매자 입장에서의 시퀀스 다이어그램을 표현한 것으로 시퀀스 다이어그램을 통해서 웹 어플리케이션의 실제 프로세스를 수월하게 될 페이지를 추출한다.



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

추출된 페이지는 [그림 16]과 같이 표현하게 된다. 상단에는 페이지의 이름을 기입하며, 이때 시퀀스 다이어그램의 클래스로부터 추출된 페이지를 사용하게 된다.



[그림 16] 페이지 구성
 [Fig. 16] Configuration of Page

5. 구현 결과

실제 웹과 연동되도록 하기 위해 시퀀스 다이어그램으로부터 추출된 페이지를 작성하고, DistributorMain 클래스를 이용하여 실제 자바 서블릿에서 import하여 자동으로 서블릿 코드가 생성되도록 한다. 생성된 소스코드는 [그림 17]과 같으며, 자바 서블릿의 기본 메소드인 doGet()메소드를 포함하며, DistrubatorMain 클래스를 import한 후 새로운 객체를 생성하여 Adjust() 메소드를 호출하고 있다.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import DistributorMain

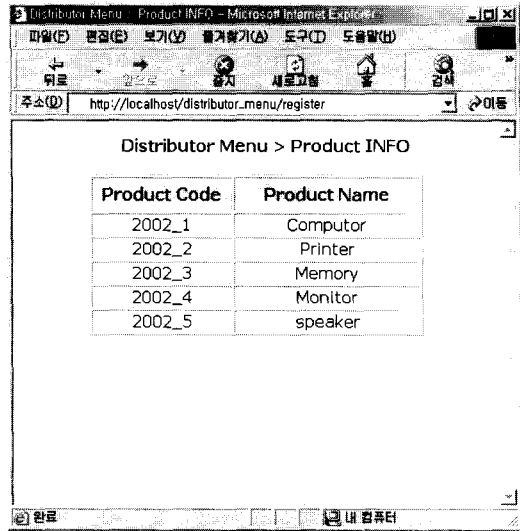
public class Admin extends HttpServlet
{
    public void doGet(HttpServletRequest req,
    HttpServletResponse res)
    throws ServletException, IOException {
        //물품등록을 위한 물품의 목록을 받아 온다
        String product_name = req.getParameter
        ("product_name");
        String product_code = req.getParameter
        ("product_code");

        //물품에 대한 정보를 저장한다
        register(product_name, product_code)
    }
}
```

[그림 17] DistributorMain 클래스를 적용한 자바 서블릿 코드

[Fig. 17] JavaServlet Code Using DistributorMain Class

[그림 18]은 실제 생성된 웹 페이지를 이용하여 물품을 등록한 화면을 보여주고 있다. 본 논문에서는 UML을 이용함으로써, 객체 지향적으로 분석 설계가 이루어지는 한편 페이지를 자동으로 추출하여 코드를 생성하게 된다. 이러한 객체지향 기술을 자바 기반의 웹 서비스인 자바 서블릿에 접목하여 웹 프로그래밍을 단순화하는 하였다.



[그림 18] 상품등록 완료 페이지

[Fig. 18] Page of Product Register

6. 결론

본 논문에서는 웹 어플리케이션의 개발과정 시 소프트웨어 공학적 접근을 위하여 UML 다이어그램을 이용하여 웹 어플리케이션을 모델링하고, 페이지를 추출하여 실제 웹 서비스가 이루어지도록 자바 서블릿 코드로 자동 변환될 수 있도록 하였다. 객체 지향의 표준 기법인 UML을 이용함으로써, 요구사항의 분석과 설계단계에서의 재사용이 가능할 뿐만 아니라, 대규모 웹 어플리케이션개발에 있어서 인력낭비와 차후 유지 보수에 대한 비용절감 효과가 있다.

향후 생성된 페이지에 대한 검증 및 평가가 이루어져야 할 것이며, 향후 비즈니스 로직과 프리젠테이션 로직을 분리하여 원하는 때에 웹 어플리케이션의 룩앤필을 변경할 수 있도록 XML과 XSLT를 이용하는 방법에 대한 연구가 필요하다.

※참 고 문 헌

[1] Jim Conallen, Building Web Applications with UML, Addison-Wesley, 1999.
 [2] D. Lowe, R. Webby, 1998, Web Development Process Modeling and Project Scoping : Work in Progress, WebE '98.
 [3] Cheesman, Daniels, UML Components A Simple Process for Specifying Component-Based Software, Addison Wesley, 2000.
 [4] C. Larman, Applying UML And patterns, Prentice Hall, 2001.
 [5] Tomas Isakowitz와 2인, 1995, RMM: A Methodology for Structured Hypermedia Design, Communications of the ACM, Volume 38, No.8, pp.34-44.
 [6] Jingfeng Li와 1인, 2000, Modeling Web Application Architecture with UML, IEEE 2000.
 [7] Hans-Werner와 1인, 1997, WebComposition: An Object-Oriented Support System for the Web Engineering Life Cycle, Sixth International World Wide Web Conference.
 [8] Daniel Schwabe와 1인, 1995, "Building Hypermedia Applications as Navigational Views of Information Models", Proceedings of the 28th Annual Hawaii International Conference on System Sciences, Volume 3, pp.231-240.
 [9] Agent Technology Green Paper ver.1, <http://www.objs.com/agent/index.html>
 [10] Rational Software White paper <http://www.rational.com/products/whitepapers/100420.jsp>.
 [11] C. McClure, Software Reuse Techniques: A Guide to Adding Reuse to the Software Process, Extended Intelligence, 1996.
 [12] 송정길, 2000, Web을 이용한 실과교육 교수·학습 자료 개발, '01 추계 학술발표회, 한국인터넷정보학회 제1권 2호, pp.93-97.
 [13] 김수동, 실무자를 위한 소프트웨어 공학, 에드텍, 1999.
 [14] OMG Agent Platform Special Interest Group, <http://www.objs.com/agent/index.html>.
 [15] 변성철, 2000, UML을 이용한 인터넷 비즈니스 프로세스 모델링 및 3계층 인터넷 컴퓨팅 환경에서의 구현에 관한 연구, 광주과학기술원.

[16] 이훈기, 1998, Java 기반으로 작성된 UML 표기법 지원 CASE 도구의 설계 및 구현, '98 정보과학회 가을 학술 발표 논문집.
 [17] 정기원, 2001, 노트와 스테레오타입을 위한 UML 기반 웹 어플리케이션 설계 사례, '01 추계 학술발표회, 한국정보과학회 제28권 1호(A), pp.658-660.
 [18] 송정길, 2000, 웹상에서의 가상교육 평가시스템 설계, '01 추계 학술발표회, 한국인터넷정보학회 제1권 2호, pp.114-119.

정진영



1992년 한남대학교 전자계산학과(공학사)
 1994년 한남대학교 대학원 컴퓨터공학과(공학석사)
 1996 ~ 현재 1988년 한남대학교 대학원 컴퓨터공학과(박사과정)
 1997 ~ 현재 대전보건대학 멀티미디어소프트웨어과 전임강사
 관심분야 : 멀티미디어 문서처리(XML), 객체지향 모델링 및 방법론, 분산시스템 및 실시간 시스템 등

송정길



1966년 한남대학교 수학과(이학사)
 1982년 홍익대학교 대학원 전자계산학과(이학석사)
 1988년 중앙대학교 대학원 전자계산학과(이학박사)
 1990년~1991년 University of illinois 객원교수
 1979년~ 현재 한남대학교 정보통신·멀티미디어 공학부 교수
 관심분야 : 멀티미디어 문서처리(XML), 객체지향 모델링 및 방법론, 전자상거래, 분산시스템 및 실시간 시스템 등