MVC Architecture 기반의 EC System용 상품전시 컴포넌트의 설계 및 구현

서 순 모[0] 양 해 술
호서대학교 벤처전문대학원 컴퓨터응용기술전공
bante97@hanmail.net, hsyang@office.hoseo.ac.kr

Design & Implementation of displaying component of product for EC System based on MVC Architecture

Seo Soon-Mo[0] Yang Hae-Sool
Graduate School of Venture, Hoseo University

요 약

전자상거래와 관련한 각종 신기술들이 쏟아져 나오고 있다. 이미 많은 EC솔루션들이 시장에 나와서 나름대로의 기술력을 인정받고 또한 수정 보완의 개선점을 찾아가고 있는 상황이다. 본 논문에서는 전자상거래 시스템의 구성 아키텍처를 MVC모델을 기반으로 하여 상품전시용 컴포넌트를 구현함으로써 기존에 제기되어진 시스템 개발 및 유지보수과정의 문제점 즉, 디자인 및 프로그램 담당자간의 원활한 의사소통 등의 문제점을 개선하고 생산성을 강화하는 등의 발전 가능성을 제고해 보고자 한다. 이에 따라 MVC모델에 관하여 연구하고, 기존의 전자상거래 시스템 개발 패턴에 대해 알아본다. 본 논문에서 MVC 모델을 통한 전자상거래 시스템용 상품 전시 컴포넌트를 구현함으로써 기존의 시스템 개발 패턴에 비한 개선점이 무엇인지 도출해 보고자 한다..

1. 서 론

전자상거래와 관련하여 수 많은 기술들이 개발되어지고 있다. 이에 따라 대학에서도 전자상거래 관련 학과가 개설되어지고 있는 등 국가적으로도 전자상거래에 대한 관심이 고조 집중되고 있다. 그러나 이러한 전자상거래의 전반적인 발전에도 불구하고 기존에 이루어지는 전자상거래 시스템의 개발 프로세스는 과거의 단일 패키지 소프트웨어와 같은 아키텍처 구성을 채택함으로써 문제점이 발생되어지고 있다. 물론 컴포넌트를 기반으로 하는 전자상거래시스템의 연구개발도 이루어지고 있다. 그러나 규모가 미약하다. 대부분의 전자상거래 시스템 개발은 3-tier기반으로 개발되고 있지만, 미들웨어 부분에서 프리젠테이션(인터페이스) 부분에 비즈니스 로직과 프리젠테이션 코드를 혼합한 개발 방식을 택함으로써 전자상거래 시스템을 개발하는 속도를 단축하고, 클라이언트의 요청과 자원의 상태에 따른 동적 내용을 손쉽게 개발할 수 있을지라도 향후 시스템의 유지 보수 작업 과정에서 많은 애로사항을 유발하는 단점을 겪게 된다. 일례로 전자상거래 시스템을 개발하는 팀에서 디자인 즉, 프리젠테이션 측면을 담당하는 사람과 비즈니스 로직을 담당하는 사람 그리고 데이터 베이스 관리자 등의 여러 담당자가 있을 때 이들간의 의사소통에서 문제점이 발생할 수 있으며 실제로 그러한 문제로 하여금 소규모의 전자상거래 개발 업체들이 어려움을 겪고 있다. 본 논문에서는 이러한 문제점을 개선하기 위한 방안의 하나로 MVC(Model-View-Controller)모델을 적용한 전자상거래 시스템을 고려하여 MVC모델 기반의 EC 시스템용 컴포넌트를 구현해 보고 그 장점을 검증해 보고자 한다. 그리하여 MVC모델을 통해서 본격적인 컴포넌트 기반의 시스템 개발로 넘어가기 위한 과도기 적 시스템 개발 모델의 한 수단으로서 이해하며, 궁극적으로 컴포넌트를 활용한 시스템의 개발이 시스템의 유지 보수 및 생산성과 재이용성을 강화시켜준다는 기존의 개념을 구현 및 실험해 보고자 한다.

2. 관련연구

2.1 MVC 아키텍처(Architecture)

MVC(Model-View-Controller) 아키텍처는 1970년대 후반부터 잘 알려져 있으며, 전통적인 OOP에 기반하여 출발하였다. 객체지향언어의 근간이라 할 수 있는 Smalltalk에 관한 MVC아키텍처의 적용 방법이 널리 알려져 있다. 근래 들어 Sun-microsystems사에서 내놓은 자바(Java)언어를 이용한 Server-Side Java인 JSP(Java Server Page)는 이러한 MVC아키텍처를 지원하는 가장 대표적인 개발 도구로 주목받고 있으며, Java의 Swing은 MVC 아키텍처를 적용한 가장 유연하고 강력한 최초의 자바 애플리케이션의 UI지원 컴포넌트이다. MVC아키텍처는 모델(Model), 뷰(View), 컨트롤러(Controller)의 세가지로 이루어진다[1][2].

2.1.1 MVC-모델(Model)

모델은 컴포넌트가 다루는 자료의 구조를 추상화한다. 즉, 컴포넌트의 모든 정보를 유지하는 기능을 가진다. 정의(definitions)를 살펴보면 다음과 같다.

- 데이터와 코어(Core) 애플리케이션 기능을 포함
- User Interface(UI)에 독립적
- 키워드 : data, storage, processing

2.1.2 MVC-뷰(View)

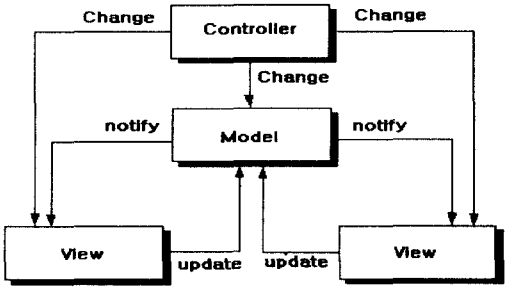
화면 출력을 담당하는 컴포넌트이다. 즉, 시각적인 표현들을 결정함으로써 색깔을 표현한다던가 폰트의 형태를 변경해 주는 등의 기능을 한다. 정의는 다음과 같다.

- 사용자에게 모델의 데이터를 표현
- 애플리케이션의 "룩(Look)"제공
- 키워드 : output

2.1.3 MVC-컨트롤러(Controller)

뷰와 모델의 제어(Control)를 담당하는 컴포넌트이다. 요구되고 입력되는 이벤트를 통해서 어떤 액션(Action)이 이루어 져야 하는지를 결정한다.

- 뷰에서 UI이벤트와 전달되는 내용을 핸들링
- 특별한 뷰(view)에 대한 전문화된 작업
- 애플리케이션의 "필(Feel)"제공
- 키워드 : input

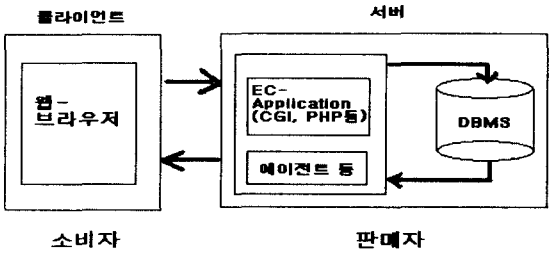


(그림 1) MVC 컴포넌트 모델간의 관계

2.2 기존의 EC시스템 개발 패턴

2.2.1 프리젠테이션과 비즈니스 로직계층의 통합 개발

이미 개발되어진 많은 전자상거래 솔루션뿐만 아니라 e-Business를 위한 많은 애플리케이션은 전통적인 개발 방법을 택해왔다. 이러한 것은 개발의 생산성 및 능동적인 애플리케이션 및 페이지를 생산할 수 있는 장점이 있지만, 유지 보수와 같은 고객의 다양한 요구 및 환경의 변화에 따른 변경작업이 어려워지는 문제점이 나타난다. 다른말로 페이지(Page)중심의 설계와 개발이라도 한다.

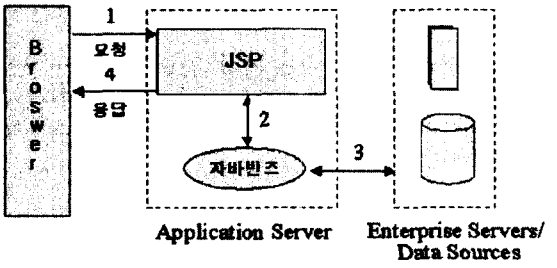


(그림 2) 전통적인 전자상거래 개발 모델

소비자의 요구가 고도화됨으로 인해서 프리젠테이션 및 비즈니스 로직이 매우 복잡하게 설계되는 경향을 보이는데, 이러한 패턴의 설계 및 개발은 팀(team)내 서로 다른 영역의 전문가사이의 의사 소통을 어렵게 하고 결과적으로 생산성을 저하시키는 문제점을 불러온다.

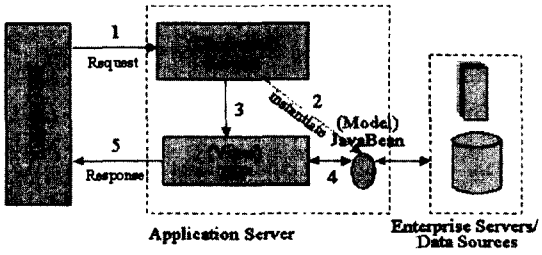
3. EC시스템용 상품전시 컴포넌트의 설계 및 구현

먼저 MVC모델을 적용한 Server-Side 애플리케이션 개발 모델중에 JAVA 진영의 JSP는 모델 1과 모델2가 있다. 단순한 애플리케이션 개발에 적합한 모델 1은 [1] (그림 3)에 나타난 바와 같이 간단한 구조를 지닌다.



(그림 3) JSP Model 1 Architecture[1]

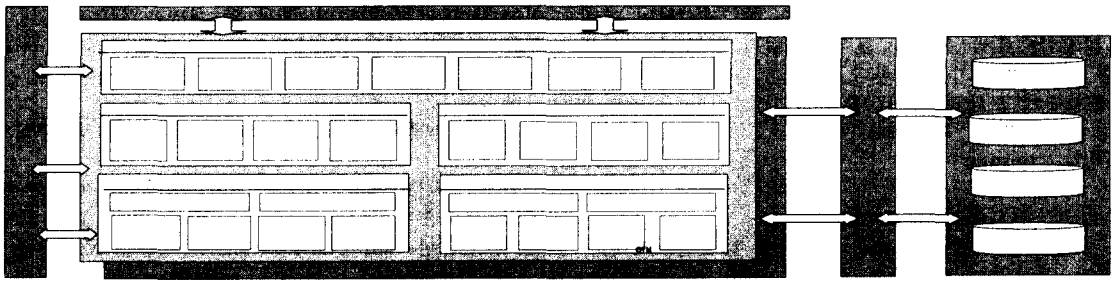
또한 좀 더 복잡한 애플리케이션 설계 및 구축에 적합한 모델 2는 요청과 응답을 분리 적용 개발하는 구조로서 (그림 4)에 표현된 바와 같다.



(그림 4) JSP Model 2 Architecture[1]

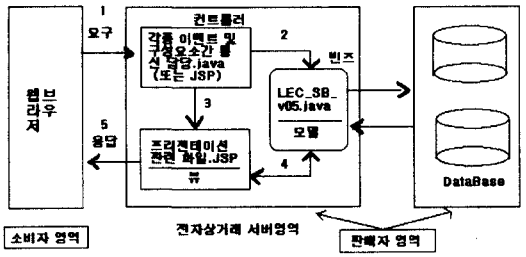
본 연구에서는 JSP Model 2의 구조를 기반으로 하여 상품전시용 컴포넌트를 구현하고 이를 적용하여 경량 EC시스템을 구현한다. 즉, 구성은 구현을 담당하는 JSP 부분과 컨트롤러 부분을 담당하는 서블릿(Servlet), 그리고 모델 부분을 담당하는 자바빈즈(JavaBeans)를 통해서 시스템을 설계하고 구현하도록 한다. 이중 상품전시용 컴포넌트는 모델부분에 위치한다. 이러한 구조를 그림으로 도식화하면 (그림 5) MVC모델에 기반한 EC 시스템이 된다.

개발되는 플랫폼은 레드햇 리눅스 계열의 와우 리눅스 6.1을 사용하며 Mysql 3.23.22 버전을 사용하고, Apache 1.3.19, Jakarta-Tomcat을 사용하며, Java 1.3.1을 사용하여 JSP와 서블릿 그리고 DBMS를 사용할 수 있는 환경을 구축하였다. 기본적으로, 구현되어지는 전자상거래용 상품전시 컴포넌트는 자바빈즈로 이루어지며 컴포넌트의 기능 구성은 상품등록, 상품관리, 상품



(그림 6) EC 시스템을 구성하는 컴포넌트군 집단[3]

분류관리, 이미지 관리, 상품 상세 보기, 구매하기 등 기본적인 내용[3]들을 선정하였다. 뿐만 아니라 이러한 컴포넌트들을 구현하고 시험하기 위해 회원관리, 쇼핑카트 등을 추가로 구현하였다.



(그림 5) MVC모델에 기반한 EC 시스템 구조

3.1 EC시스템용 상품전시 컴포넌트 구현

다음에 나타나는(그림 7)은 상품의 데이터베이스 테이블이며, (그림 8)은 구현된 상품전시를 담당하는 빈즈 컴포넌트의 소스 코드이다.

```
mysql> explain goods;
+---------------+-------------+------+-----+---------+----------------+---------------------------------+
| Field         | Type        | Null | Key | Default | Extra          | Privileges                      |
+---------------+-------------+------+-----+---------+----------------+---------------------------------+
| goods_id      | int(11)     |      | PRI | NULL    | auto_increment | select,insert,update,references |
| category_id   | char(3)     |      |     |         |                | select,insert,update,references |
| goods_name    | varchar(100)|      |     |         |                | select,insert,update,references |
| goods_price   | int(11)     |      |     | 0       |                | select,insert,update,references |
| goods_basic   | text        | YES  |     | NULL    |                | select,insert,update,references |
| goods_detail  | text        | YES  |     | NULL    |                | select,insert,update,references |
| goods_image   | varchar(100)| YES  |     | NULL    |                | select,insert,update,references |
| goods_store   | int(11)     | YES  |     | NULL    |                | select,insert,update,references |
+---------------+-------------+------+-----+---------+----------------+---------------------------------+
8 rows in set (0.00 sec)
```

(그림 7) 상품 테이블 구조

상품 테이블은 상품 아이디, 카테고리 아이디, 상품이름, 가격, 기본설명, 상세 설명, 이미지, 재고량의 순서로 설정되어져 있으며 이외에도 회원, 카테고리, 구매 등의 다양한 테이블이 설치됐다. 소비자가 요청을 하면 서블릿에서 마우스 클릭 같은 이벤트를 접수하고 빈즈 및 뷰 컴포넌트에게 메시지를 넘긴다. 이러한 것은 (그림 5)의 순서에 나타난 순서에 의해 구현되었다.

```
// ************************변수 선언 부분(고객)*********************************
  String cust_id;
  String cust_pass;

  String cust_email;
  String cust_tell;
  String cust_mobile1;
  String cust_job;
  String cust_post_no;

////////////////////////////////{ BLOCK }////////////////////////////

  public LEC_SB_v0_5() {

      try {
        Class forName("org.gjt.mm.mysql.Driver");
        // IP 주소 지정
        String url = "jdbc:mysql://127.0.0.1:3306/L_EC_Top";

        // 데이터베이스 연결
        Sm_Conn = DriverManager.getConnection(url, "관리자아이디", "비밀번호");
      }
      catch(Exception e) {

      }
  }

  public void getProducts_List() {
      try {
        String jsql = "SELECT * FROM product_item";

        PreparedStatement Q_PStat = shopConn.prepareStatement(jsql);
        ResultSet St_Rs = Q_PStat.executeQuery();
        St_Rs.next();
        int number = St_Rs.getInt(1);

        int products_id[] = new int[number];
        String product_items[] = new String[number];
```

(그림 8) LEC_SB 자바빈즈 코드(상품전시용 컴포넌트)

4. 결론 및 향후계획

본 연구에서는 MVC아키텍처에 기반하는 전자상거래 시스템을 개발하기 위하여 모델 부분의 자바 빈즈 컴포넌트를 구현하였다. 제2장에서는 관련연구로서 MVC아키텍처 및 기존 EC개발 패턴에 관해 알아보았고 제3장에서는 MVC아키텍처를 적용한 EC컴포넌트의 설계와 구현을 언급하였다.

컴포넌트에 관한 관심이 매우 거세어 지고 있다. 생산성 및 재 이용성의 측면이 매우 매력적이기 때문이다. 향후 계획으로는 다양한 컴포넌트를 더 구현하여 안정적인 EC 시스템을 구현하고 실험하여 성능을 개선해 보는 것이다.

참고문헌

[1] Understanding JavaServer Pages Model 2 architecture : Exploring the MVC design pattern, http://www.javaworld.com/
[2] The Java Dependency Mechanism and the MVC Approch - http://genome-www.stanford.edu/~sac/java/mvc
[4] 서순모, 양해술,"동적 컴퓨팅 환경에서의 전자상거래 컴포넌트 도입방안",한국정보처리학회 소프트웨어공학연구회지 제3권 제4호, 2000.12
[5] 서순모,양해술, 박정호, "리눅스 전자상거래시스템의 소비자 의사정보 제공과 확인을 위한 에이전트 개발",한국정보처리학회 춘계학술대회논문집, 2000, CD-ROM