제33회 한국정보처리학회 춘계학술발표대회 논문집 제17권 제1호 (2010. 4)

웹 응용의 웹 서비스 전환에 관한 연구

최서익, 이주활, 변정용
동국대학교 컴퓨터멀티미디어학부
e-mail : coolington@naver.com, hwali83@naver.com, byunjy@dongguk.ac.kr

A Web Services conversion of Web Applications

Suhik Choi, Joohwal Lee, Jeongyong Byun
Dept of Computer and Mutimedia, Dongguk University

요 약

기업들의 웹서비스(Web Services)수요가 증가함에 따라 그에 따른 기존 시스템에 대한 변경 또는 재설계가 늘어나고 있다. 본 논문에서는 시스템의 재설계에 따른 문제점과 MVC(Model View Control) 구현이 안 된 시스템에 대한 MVC 분리구현 문제, 기존의 시스템과의 호환성문제, 즉 웹 응용과 웹서비스의 차이에서 발생하는 문제들에 대한 문제제기와 이에 대한 해결책을 제시하고자 한다.

1. 서론

일반적으로 기업들이 정보 시스템을 구축하는 이유는 비용절감과 효율성 증대를 위한 것이다. 그러기 위해서는 복잡한 정보시스템 구조를 단순화 시키고 시스템과 응용을 통합하여야 하는데 그러기 위해선 비용은 물론이고 기존 시스템을 어떻게 할 것인가에 부딪히게 된다.

이번 연구에선 현재 IT(Information-Technology)업계에서 최고의 화두가 되고 있는 '서비스 지향 아키텍쳐'(SOA : Service Oriented Architecture)[1]를 가장 잘 구현한 기술인 웹서비스를 기존 JSP응용시스템에 적용하여 이를 통해 상호운영성과 코드 재사용성을 높여 시스템의 효율성을 높이는데 있다[4]. 이를 위하여 기존의 시스템에는 어떠한 변화를 주어야 하는지, 그리고 그에 따르는 문제점은 무엇인지에 대하여 연구하였다.

2. 관련연구

경주전략산업[2]연구의 일환으로 본 연구의 주제인 공급사슬관리(SCM) 시스템의 JSP웹 응용이 이미 구현되어 있다. 하지만 신기술을 도입 거나 업데이트를 하기 위해선 프로그램 전체에 대하여 수정을 하거나 내부 구성을 바꿔야 하는 일이 발생할 수 있다.[3]

이 같은 문제점을 고치기 위해서 기존 프로그램을 자바 빈(JAVA Beans) 기반으로 재구성하여야 한다. 자바 빈 기반으로 구현됨으로써 새로운 신기술 도입이나 서비스를 추가 혹은 수정 할 때 가장 큰 문제인 시간과 비용을 줄일 수 있게 된다. 이후 자바 빈 부분들에 대해 웹서비스를 적용하게 되면 객체 지향이 아닌 메시지지향으로 느슨한 결합을 추구하는 웹서비스 기반이 되어 서로 다른 언어로 구현된 프로그램이나 상이한 환경에서 이뤄진 시스템이라도 효율적으로 구성될 수 있다.

3. 설계

JSP기반 웹 응용으로 구성된 공급사슬관리 시스템을 자료 접근 부분과 논리 부분을 분리하여 자바 빈으로 생성한다. 만약 웹 서비스를 적용하려는 페이지가 MVC 프로그래밍이 철저히 이루어져 있다면 모델 부분과 제어 부분에 자바 빈을 적용한다. 그렇다면 MVC 프로그래밍이 명확하지 않을 경우 어떻게 해야 할까?

먼저 기존의 프로그램에 웹서비스를 적용하기 위한 요구사항을 살펴보자.

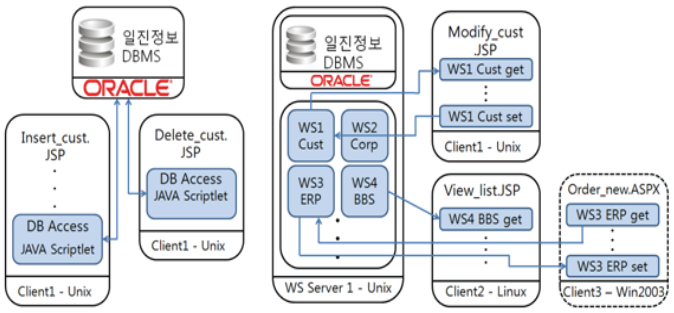
첫째, 기존 웹 응용으로 만들어진 공급사슬관리 시스템을 기반으로 재사용성과 상호운영성 보장을 위해 자바 빈을 적용하여 JSP페이지와 내부로직을 서로 분리한다.

둘째, 서비스별 분리된 자바 빈에는 서비스 요청에 맞게 각각의 서비스를 도출해야 한다. 대부분의 자바 빈에는 DBMS와 연동되어 요청된 서비스에 따라 정보 값이 신속히 바뀌거나 전달되어야 한다.

셋째, 사용자 인터페이스와 페이지 간 매개변수 등은 새로 생성된 자바 빈과 페이지에서도 전과 같은 동일한 형태를 가져야 한다. 만약 매개변수의 형태 변경 시 해당하는 값을 수행해야 하는 페이지에 대해서도 전반적인 수정이 필요하게 됨으로 가급적 기존의 매개변수를 사용한다.

넷째, 구성된 내부로직에 웹서비스를 적용한다. SOAP(Simple Object Access Protocol)메시지를 통해 서비스를 요청하여 요청하는 서비스들에 대한 검증을 실시한다.

이렇게 하여 모든 분리 구현된 데이터베이스 접근 부분을 데이터베이스 관리서버에 웹서비스로 등록 및 배치(Deploy)하고 이것은 SOAP을 이용한 전송방식으로 JSP로 전달되고 전달된 결과물은 조건에 맞게 출력된다.



(그림 1) 웹 응용의 DB 접근 부분과 웹서비스를 이용한 DB 접근부분 분리구현의 예

4. 실 험

4.1 개발 환경 및 개발 툴

이번 연구에선 웹서비스 기반으로 구현된 시스템의 상호 메시지 전송의 성사유무와 실행여부를 알아보기 위해 유닉스 서버에 오라클 데이터베이스를 구축하였고 고객으로는 개인 PC를 이용하였다.

	운영체제	개발언어	개발도구
Server	Solaris 5.9	JAVA ,JSP, Javascript, SQL	Oracle Data Base 10g
Client	Windows XP	JAVA ,JSP, Javascript, CSS	NetBeans6.8 Apache Tomcat 6.0

4.2.1 웹서비스 구현

고객 정보를 불러오는 웹 응용페이지를 기준으로 설명한다. 이 페이지에서는 고객의 아이디와 비밀번호를 입력받아 이를 데이터베이스와 비교 후 사용자 정보가 가지고 있는 기업번호와 고객번호를 기준으로 사용자가 사용할 수 있는 메뉴와 기능에 대해 차별 분리하고 이를 바탕으로 주 화면을 구성하게 된다.

웹서비스에서는 고객정보 및 데이터베이스 비교 부분을 제2의 서버에 구현하여 내부 연산과 화면표현이 각각의 서버로 분리되도록 하였다.

4.2.2 자바 빈 적용

기존의 페이지에서 SQL문과 출력로직을 위해 이전페이지로 부터 정보 전달부분인 request.getParameter()로부터 전달되어지는 값을 객체에 입력한 뒤 이를 자바 빈에 저장후, 매개변수가 되도록 해준다. 또한 실행될 SQL문에 대해 테이블의 내용을 저장할 자바 빈을 만들어 주고, 이전 파일의 내부에서 수행되어지는 모든 논리부분은 객체로 옮겨와 환경에 맞게 재구성 한 뒤 마지막으로 이 객체의 반환 값으로 자바 빈에 저장된 결과들이 List형 배열로 반환되게 설정하였다.

4.2.3 웹 응용 수정

웹 응용페이지는 내부구성은 완전히 사라지고 Class를 호출하여 반환된 결과 값들에 대해 웹에 표현하는 기능만을 구사하게 된다. 또한 이를 바탕으로 결과마다 다른 페이지를 호출하여야 되며 그에 맞는 호출 부분을 갖게 된다. 즉 View와 Control로 구성된다. 이때 주의할 점은 이전버전과의 호환성과 기본구조 유지를 위해 웹 페이지는 해당 페이지가 입력 받는 값과 다음페이지로 연결할 때 전달하여야 할 값은 수정 이전과 동일하게 유지하여 호환성과 유연성을 유지해준다.

```
public class Checkin {
    private Connection con = null;
    private Statement st = null;
    private ResultSet rs = null;

    public ArrayList<StorageBean> getInfo(String ID,String PW)
    throws SQLException{
        ArrayList<StorageBean>list = new ArrayList();
        dbconn dbc = new dbconn();
        StorageBean sb = new StorageBean();

                        .
                        .
                        .

        list.add(sb);

        return list;
    }

}
```

[소스 1] Checkin 클래스는 사용자 ID와 비밀번호를 전달받아 DB와 비교 후 관련 자료를 List로 반환한다.

4.2.4 웹서비스 호출

웹서비스 구현과 동시에 자바 빈들에 대하여 중복된 부분들에 대하여 통합을 시킨다. 데이터베이스의 같은 테이블을 사용하는 페이지들은 Class들의 통합이 있을 수 있고 자바 빈을 같이 사용할 수 있다. 이러한 부분들에 대한 통합은 웹서비스로의 전환이 끝난 후에도 가능하다.

웹 응용페이지로부터 자바 빈과 Class로 구현이 완료된 부분들은 다시 다른 서버 혹은 다른 시스템으로 웹서비스를 등록하고 웹 페이지에서는 Class에 대한 호출이 아닌 해당 웹서버에서 제공하는 웹서비스를 SOAP형태로

```
<%-- start web service invocation --%><hr/>
<%
try {
    login.CheckService service = new login.CheckService();
    login.Check port = service.getCheckPort();
     // TODO initialize WS operation arguments here
    java.lang.String id = reqID;
    java.lang.String pw = reqPW;
    // TODO process result here
    java.util.List<java.lang.Object> result = port.match(id, pw);
} catch (Exception ex) {
    // TODO handle custom exceptions here
}
%>
<%-- end web service invocation --%><hr/>
```

[소스 2] JSP 페이지 내에서의 웹서비스의 호출

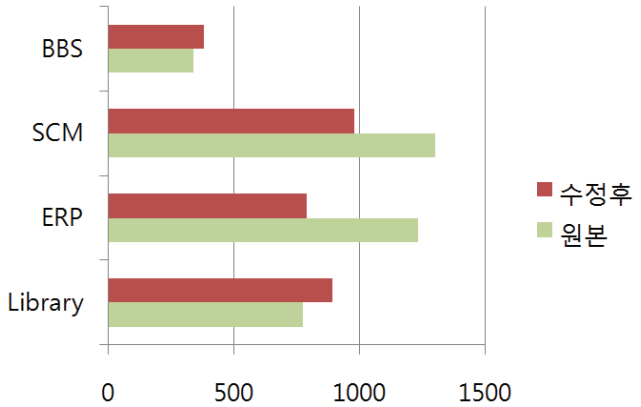
호출하도록 수정한다.

5. 결론

본 논문에서는 기존 웹 응용으로 구성된 공급사슬관리 시스템에 대해 개발의 편의성을 늘리고 효율적인 자원관리를 위하여 프로그램의 전반적인 재구성을 통해 SOA를 적용한 분산구조 서비스를 제공하며 상호운영성을 높이기 위한 목적으로 SOA를 구현하기 위한 최적의 모듈인 웹서비스를 기반으로 설계 및 구현을 통해 실험해 보았다.

이때 기존 웹 응용이 MVC로 구성되어 있었다면 웹서비스로의 전환은 크게 어려움이 없을 것이다. 하지만 이와는 거리가 먼 시스템 구성 때문에 이를 다시 MVC에 가깝게 새롭게 구성해야 되었고, 이 때문에 웹 페이지사이 매개변수들에 대한 수정은 사실상 불가능 하였다. 또한 오픈소스를 이용한 부분이나 게시판 혹은 페이지 스킨과 같은 부분에 대한 수정은 재개발에 가까웠으며 이들에 대한 수정은 실제 시스템에 대한 효율성이나 필요성에 따라 재개발 할 것인지 혹은 현재 상태를 유지 할 것인지는 개발 시작 전에 먼저 고려해야 할 사항이다.

이렇게 완성된 웹서비스는 시스템의 요구사항 변화에 능동적으로 대처할 수 있었다. 가령 새로운 기업조건을 추가하게 되면 서비스를 해주는 해당서버의 내부 구성을 수정해 주는 것으로 이를 구현할 수 있었고 한곳에서 집중적으로 처리 되던 연산에 대해서는 필요시 여러 서버로 분산시킴으로써 효율성향상에 도움이 되었다. 또한 시스템을 설계할 때 논문 이상의 효율적 시스템을 구성 한다면 어떠한 조건과 환경에서도 시스템은 능동적으로 유연하게 대처할 수 있으리라 판단된다. 뿐만 아니라 웹서비스의 상호 다른 시스템과의 연동성으로 개발언어와 운영체제가 다르더라도 하나의 시스템을 구성할 수 있는 점은 전혀 다른 시스템을 구성하더라도 기존에 사용 중인 구성요소들에 대해 재사용할 수 있어 코드 재사용성을 높였으며 기업의 입장에서는 효율성과 유연성을 동시에 확보할 수 있었다.



(그림 2) 웹서비스 전환 전과 전환 후의 전체 프로그램 라인 수 비교

사사

본 연구는 교육과학기술부와 한국 산업기술재단의 지역혁신 인력양상사업으로 수행된 결과임.

참고문헌

[1] 토마스 얼 , “SOA 서비스 지향 아키텍처”, 에이콘 2006.

[2] 일진정보, “SCM”, 2008.

[3] 한건우, 김경욱, 변정용 “자동차 부품 거래를 위한 웹서비스 기반 기업 정보관리 서비스”, 2009 한국정보과학회 학술심포지움 제3권 제2호, p.45~49 2009.12

[4] Web Services Description Available at http://www.w3.org/standards/webofservices/