

웹기반 궁도경기지원시스템 개발

곽창욱, 변정용
동국대학교 과학기술대학 컴퓨터공학부
e-mail:kwakcw@naver.com, byunjy@dongguk.ac.kr

A Web Based Archery Game Support System

Kwak Chang-Uk, Byun Jeongyong
School of Computer Engineering Dongguk University

요 약

전국의 궁도인구는 만 여명에 이른다. 매년 1~2주를 주기로 개최되는 이 대회는 참가접수부터 운영까지 오로지 오프라인으로만 진행되어 경기외적인 비용과 시간이 상당히 낭비된다. 이에 본 연구에서는 이용자의 접근성이 향상되고, 플랫폼에 독립적인 웹기반의 경기지원시스템을 구현하고자 한다. 시스템에 접근하는 이용자는 접수와 결과를 확인하는 대회 전/후에 집중되므로 이 때의 효율적인 DB Access 기술이 필요하다. 따라서 본 논문에서는 DBCP 기술을 이용하여 DB Connection부분에서 보다 효율적인 시스템으로 개발하고자 한다.

1. 서론

현재 대한궁도협회에서 개최하는 궁도대회는 한 해 약 60회로 1~2주를 주기로 전국 각지에서 돌아가면서 열리고 있다. 또한 이 협회에 정식으로 등록된 선수만 약 만 여명에 이르고 등록된 팀의 수는 670여개에 이른다. 개최되는 대회는 우편과 팩스로만 접수를 받고 있으며, 각 대회의 기록과 각 지역 정(亭)선수는 전산화가 이루어지지 않아 대회정보들을 실시간으로 확인할 수 있는 방법이 없다. 참가신청을 하고 대회에 참가하였을 때에도 대기시간이 길어 전국각지에서 장소와 시간에 구애받지 않고 대회 정보를 알고자하는 요구사항이 있다. 또한 등록된 선수들의 관리와 온라인으로 대회 신청/수정/취소 등을 수행하고 정기적으로 열리는 대회의 효율적인 관리를 위해 경기지원관리 시스템이 필요하다.

또한 대회 관리 시스템의 특성상 대체로 대회 전/후, 접수/결과조회 기능을 이용할 때 많은 사용자가 집중적으로 몰릴 것으로 예상된다. 본 시스템의 처음 테스트는 JDBC 기술을 사용하여 20명 이내의 사용자를 생성하여 진행하였다. 하지만 향후 몇 백명 이상을 대상으로 본 시스템을 운영하였을 때 접속할때마다 매번 데이터베이스에 Connection을 생성하는 방법은 그 작업에 상당한 리소스를 소비하게 되고 응답속도에도 영향을 미칠것으로 예상된다. 그리하여 본 시스템 처음 개발시 사용했던 JDBC 기술보다 Connection 리소스부분에서 효율적이고, 더 속도가 향상된 기술이 필요했으며 DBCP 기술을 이용하게 되었다.

HTML5 이전의 frame과 frameset을 이용한 기술에서는 여러곳에서 사용되는 것들을 구현할시 중복해서 사용해야 했다. 특히 상단, 하단, 메뉴 등의 기능은 페이지마다

반복되게 되는데 frame으로 동적으로 구현하기 어려웠다. 메뉴 같은 경우에는 frame 밖으로는 나타나지 않는 경우도 있었다. 또한 frame, frameset을 이용한 페이지 구성은 디자인 측면에서도 시대에 도태되었다. 그리하여 HTML5 과 CSS3 기술을 적용하여 기존 frame으로 구분해서 구현했던 것을 각 tag로 구분해서 CSS를 통한 수정의 편리성, 스타일의 관리를 용이하게 하였다.

본 논문에서는 경기지원시스템 구현뿐만 아니라 현재 보편적으로 쓰이고 있는 2가지의 데이터베이스 Connection 기술을 비교하여 어떠한 방법이 더 효율적이고 데이터베이스 부하를 줄이는지 실험을 진행하였다.

위의 요구사항을 바탕으로 시스템을 개발하게 되었으며, 본 시스템을 이용함으로써 참가접수를 위해 소비되는 시간을 절약하고 정보의 전산화를 통해 대회운영의 전반적인 업무들을 효율적으로 처리할 수 있게 되었다.

2. 기존연구

2.1 기존 기술

2.1.1 DB Connection 부분 DBCP 기반 개발

기존 JDBC Connection 방법으로는, 다수의 사용자가 데이터베이스에 접근하고 이용하는 과정에서, 모든 사용자가 시스템에 접근할 때마다 데이터베이스 Connection을 생성하고 시스템 이용, Connection을 close하는 과정이 이루어진다. JDBC 드라이버는 시스템에 하나만 존재하면 되지만 이것을 연결하는 Connection 객체가 적게는 하나에서 많게는 수없이 생성, 소멸된다. 이 과정은 데이터베이스 관리입장에서 보았을 때 시스템에 접근할 때마다 반

복적으로 이루어지는 작업들로 데이터베이스 Connection 생성에 상당한 리소스가 사용된다.

데이터베이스 연결 작업은 서버의 자원을 이용하는 작업으로, DBCP는 계속적으로 발생한다면 시스템에 상당히 부하를 주는 요소이다. 이를 보완하기 위해 미리 여러 개의 데이터베이스 Connection을 만들어 확보해 두고 클라이언트의 요청이 있는 경우, Connection을 서비스해 주는 기법이다.[1] 일정 수의 Connection을 미리 생성하여 풀에 두고, 클라이언트의 요청이 있을 경우에 이미 확보한 Connection 객체를 서비스한다. 클라이언트의 데이터베이스 작업이 끝난 후 Connection 객체를 다시 풀로 반환한다. 이 기술을 이용하면 동시 사용자 수가 많고 데이터베이스 작업이 많은 시스템에 효율적이다.[2]

2.1.2 HTML5 / CSS3 / J-Query 활용 개발

웹표준 이전 방식에서는 구조와 표현을 위해 HTML만 사용했다. 그러다 보니 표현을 하기 위해 과도한 HTML 태그가 생겨나고 구조 또한 복잡해져 개발자와 유지보수 시 많은 시간과 인력이 낭비되는 단점이 있었다.

하지만 웹표준을 적용하면 상호 호환성 문제를 해결해 줄 수 있다. 표준 웹 기술을 사용하게 되면, 플랫폼에 따라 달리 보이거나 혹은 의도된 바와는 전혀 다른 모습의 사이트가 보이게끔 하는 것을 방지할 수 있다.

CSS를 이용하면 하나의 스타일을 가지고 모든 페이지에 같은 속성을 지정해 줄 수 있고, 또한 CSS 파일을 분리해서 통합관리하기 때문에 개발자의 작업 시간이 단축된다.[3]

J-Query는 JavaScript Library인데, 사용법이 간단하고, 문법이 직관적이며, 성능이 뛰어나다. 기타 라이브러리에 비하여 코딩이 짧다.[3] 그러나 J-Query는 간단하고 쉬운 반면 배경색을 부드럽게 바꾸거나 연결된 효과를 표현하는 것과 같은 고급 효과를 나타내기에는 부족하다.[4]

2.2 적용한 기술

① 시스템의 DB연동과 인터넷에 연결된 모든 일반 사용자에게 접근이 열린 오픈 사용자 인터페이스 환경을 제공하기 위해서 웹 기반의 시스템이 필요하다. 이를 위해 JSP를 기반으로 시스템이 작성되었다. JSP는 플랫폼에 독립적인 기술 방식으로 이용자의 시스템 플랫폼이 윈도우든 유닉스 시스템이든 어느 한 플랫폼에서 개발한 시스템을 다른 플랫폼에서 운영하는 것이 가능하게 되었다.

② 기존 JDBC 기술을 이용할 시 데이터베이스 Connection에 많은 리소스를 소비하는 것을 보완하기 위해 DBCP 기술을 시스템에 적용하였다.

3. 요구분석 및 설계

3.1 시스템 role 요구사항

시스템의 role은 3가지로 나뉘는데 다음과 같다.

(1) 시스템 이용자 요구분석

① 이용자는 협회 등록된 선수, 사투(해당 사정의 장), 관리자로 나뉜다.

② 신규회원은 해당 정의 사투에게 승인을 받아 소속을 정하여야 한다.(비등록 회원은 시스템 사용불가)

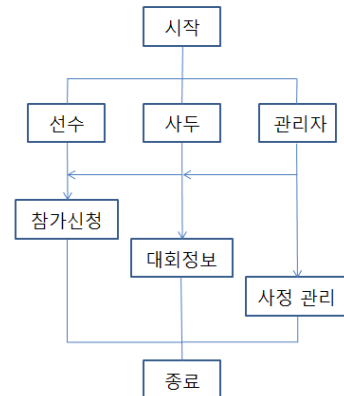
(2) 시스템 기능 요구분석

① 본 시스템은 궁도대회 운영관리 시스템으로 선수관리와 참가신청/수정/취소 등의 기능을 구현하고 전반적인 경기지원(각 종목별 경기기록/기록조회 등)을 수행하는 것이 필요하다.

3.2 시스템 설계

전체 시스템 구성은 사용자별로 크게 선수, 사투(사정 책임자), 시스템 관리자로 나눌 수 있다. 각각의 사용자들 별로 접근하는 메뉴를 달리 구현하였다.

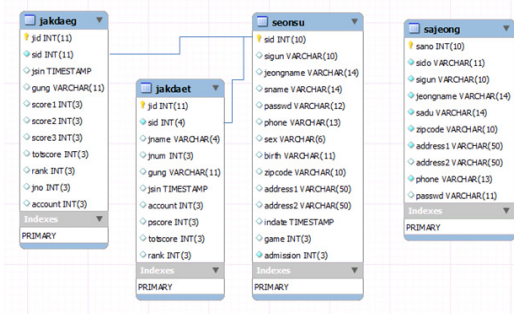
3.2.1 전체 시스템 구성도



(그림 1) 전체 시스템 구성도

3.2.2 DB 설계

DB는 그림 2과 같이 구성된다. DB Table은 단체전, 개인전, 선수, 사정 등 총 4개로 구성하였다. 선수정보가 들어있는 seonsu Table을 기준으로 sid(선수번호)를 이용하여 jakdaeg(개인전), jakdaet(단체전) Table에서 선수별 대회 정보를 공유한다. 이외에도 선수/사정 정보를 등록/수정할 때 필요한 우편번호 및 주소 Table을 관리한다.



(그림 2) DB 테이블 구성도

4. 시스템 구현 및 실험

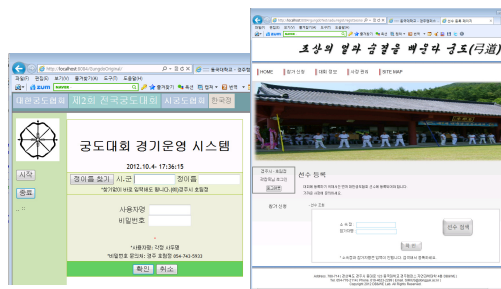
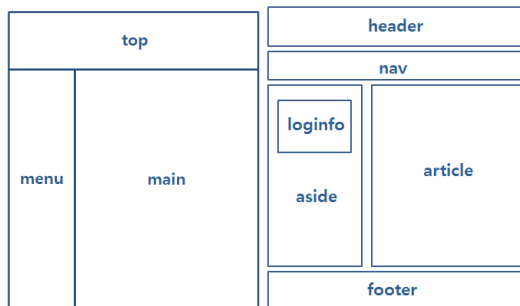
4.1 시스템 구현 환경

<표 1> 구현 환경

구분	사양
시스템	Intel(R) Core(TM) i5-2500 CPU @ 3.30GHZ
	4.00GB Random Access Memory
운영체제	Window 7 Home Premium K
개발도구	Netbeans 7.1.1 (Java)
	Apache Tomcat 7.0.22
	Mysql workbench 5.2 CE
	Apache JMeter 2.7

4.2 HTML5 tag 적용한 시스템 구현

초기 HTML의 frame과 frameset을 이용한 페이지 분할 구현방법으로는 코드의 재사용성이라는 정의에 어긋난다. 또한 해당 frame에 고정되어 유동적인 웹페이지를 구성할 수 없다. 그리하여 본 시스템에서는 HTML5에서 추가된 tag를 적용하여 구현 구조를 개편하였다. 그림 6에서 볼 수 있듯이 header, footer 등 각 tag를 사용하여 페이지를 각 부분으로 나누고 그에 맞는 소스를 include해서 사용하는 것으로 구현하였다. 이는 개발시 코드 가독력을 높여주고 중복된 코드 양도 줄어 웹 페이지가 가벼워지는 효과가 있다.

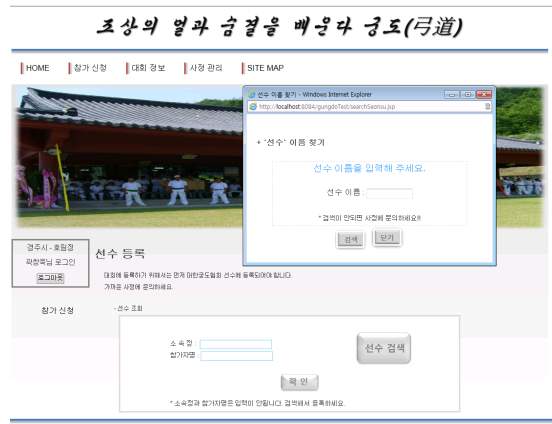


(그림 3) CSS layout 적용 전후 비교

4.3 시스템 구현

위의 기술과 요구사항을 바탕으로 운영관리 시스템을 구현하였다. 사용자는 선수, 사무원, 관리자 3가지로 나뉘며 로그인시 그 신분을 선택하도록 구현하였다.

그림 4은 참가신청하기 전에 인증된 사용자 여부를 확인하는 창이다. 협회에 등록되지 않은 선수는 시스템을 이용할 수 없도록 구현하였다. 그리하여 선수검색을 통하여 정식으로 등록된 선수만 대회신청이 가능하게 구현하였다.



(그림 4) 참가 신청

그림 5은 경기 결과 입력 화면인데, 해당선수의 점수를 선택하여 입력할 때 DB에 점수뿐만 아니라, 자동으로 랭킹 Attribute 값도 입력되도록 구현하였다. 입력 후에는 그림 6과 같이 선수별 랭킹과 점수를 확인할 수 있다.



(그림 5) 경기 결과 입력



(그림 6) 경기 결과 조회

4.4 JDBC와 DBCP 응답 속도 비교

4.4.1 DBCP 기술 적용

Apache tomcat에서의 기본 DBCP 구현은 Commons dbcp, Common Pool libraly들이 사용된다. 기능 적용을 위해 그림 7에서와 같이 Web Pages - META-INF - context.xml 환경설정이 필요하다. [5]

그림 8은 Source Package 안의 getDBConn 메서드이다. 이 메서드에서 DB Connection이 이루어져 Connection 객체를 반환하게 되는데 그림7에서 지정한 path로 DataSource를 받아와서 Connection을 생성한다.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Context antiJARLocking="true" path="/gungdotest">
3   <Resource name="jdbc/mysql" auth="Container" type="javax.sql.DataSource"
4     maxActive="1000" maxIdle="30" maxWait="10000"
5     username="byunju" password="todus1956" driverClassName="com.mysql.jdbc.Driver"
6     url="jdbc:mysql://solgeo.dongguk.ac.kr:3306/gungdotest"/>
7 </Context>
    
```

(그림 7) context.xml

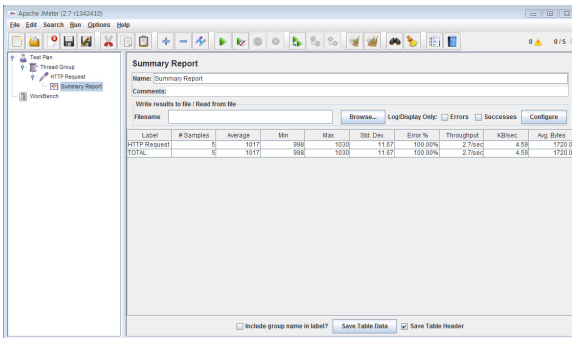
```

public Connection getDBConn() {
    try {
        Context initContext = new InitialContext();
        Context envContext = (Context) initContext.lookup("java:/comp/env");
        DataSource ds = (DataSource) envContext.lookup("jdbc/mysql");
        conn = ds.getConnection();
    } catch (SQLException ex) {
        Logger.getLogger(DBConn.class.getName()).log(Level.SEVERE, null, ex);
    } catch (NamingException ex) {
        Logger.getLogger(DBConn.class.getName()).log(Level.SEVERE, null, ex);
    }
    return conn;
}
    
```

(그림 8) getDBConn Method

4.4.2 JMeter를 이용한 JDBC와 DBCP 응답속도 비교

시스템 개발 전 Apache사의 JMeter라는 프로그램을 사용하여 동일한 조건을 설정하고 각각 JDBC와 DBCP 기술의 응답시간 비교를 하였다.



(그림 5) Apache JMeter 2.7

<표 2> JDBC 기술을 사용한 경우

구분	1*5	10*5	50*5	100*5	200*5
samples	5	50	250	500	1000
average	181	465	2416	3950	7040
min(ms)	163	158	254	207	241
max(ms)	214	918	5681	14962	34668

<표 3> DBCP 기술을 사용한 경우

구분	1*5	10*5	50*5	100*5	200*5
samples	5	50	250	500	1000
average	87	181	1348	2547	5694
min(ms)	80	75	71	55	54
max(ms)	102	684	8444	18600	31231

서로 비교한 항목은 총 4가지로 samples : 서버에 요청한 횟수, average 평균 응답시간(ms단위), min 최소 응답시간(ms단위),max 최대 응답시간(ms단위)

JDBC와 DBCP 응답속도 비교 실험은 localhost Server 상에서 실험이 진행되었으며, 동일한 소스로 비교 하였다.

5. 결론

현재 각종 시스템들은 플랫폼에 독립적인 웹기반으로 많이 구현된다. 하지만 웹기반으로 구현할 경우 대외특성 상 개최 전/후에 시스템에 많은 사용자가 몰려 이전의 JDBC 기술로는 데이터베이스 연결에 상당한 부하가 걸리게 된다. 실제로 기존 JDBC 기술로는 사용자가 접근할 때마다 Connection 객체를 만들고 이용 후 다시 종료하는 것을 반복한다. 이를 보완하기 위해 데이터베이스 Connection 객체를 미리 만들어 놓고 그것을 가져다 쓰는 방법인 DBCP 기술을 이용하여 시스템 효율성을 향상하였다. <표1>과 <표2>에서 볼 수 있듯이 비교 스트레스 실험 결과, DBCP 기술을 적용했을 때 평균응답속도가 JDBC 기술을 적용하였 때 비해 20% ~ 60% 감소하였다.

또한 기존 frame, frameset 기반으로 웹페이지를 작성할때보다 HTML5, CSS3 layout 사용을 통해 필요한 부분에 include해서 사용함으로써 중복된 코드를 줄일 수 있었다. frame, frameset으로 페이지 구성시에는 페이지단위로 구현이 힘들고, 동적 웹페이지 구성이 어려웠지만 HTML5 tag 사용으로 구현이 가능해졌다.

본 시스템은 웹에서만 이루어지는 시스템으로써 향후 모바일 버전을 개발, 현재 시스템과 연동하여 경기 감독관이 점수를 확인 후 실시간으로 DB에 해당 선수의 점수기록값을 입력하는 연구가 이루어져야하겠다.

참고 문헌

- [1] 김영찬 외 1명, 동시사용자 접근이 용이한 데이터베이스 커넥션 풀 아키텍처, 하계학술발표논문집, 한국컴퓨터정보학회, 2008
- [2] 강환수 외 1명, Perfect JSP 웹프로그래밍, 인피니티북스, 2009
- [3] 양용석, 처음부터 다시배우는 HTML5&CSS3:실전 웹표준사이트 제작까지, 로드북, 2011
- [4] 이선웅 외 1명, 웹 접근성을 고려한 jQuery 유저 인터페이스 효과 플러그인, 추계학술대회, 한국정보통신학회, 2011
- [5] <http://tomcat.apache.org/tomcat-6.0-doc/jndi-datasource-examples-howto.html>
- [6] 김영찬 외 4명, 동시접속 사용자 접근을 고려한 데이터베이스 커넥션 풀 아키텍처, 한국콘텐츠학회논문지, 한국콘텐츠학회, 2009
- [7] 이재환 외 1명, DB Connection Pool 관리를 위한 성능 진단도구의 설계 및 구현, 춘계학술대회, 한국정보처리학회, 2003