

CBD기반의 웹 서비스를 이용한 운동선수 관리 시스템 설계 및 구현

김경식, 이재동
단국대학교 컴퓨터 과학 및 통계학과
e-mail:natural@dku.edu

A Design and Implementation of Athlete Management System using CBD Based Web Services

Kyung-Sik Kim, Jae-Dong Lee
Division of Information and Computer Science, Dankook

요 약

웹 서비스는 벤더들에 따라 다르게 사용되던 시스템들을 표준 기술인 XML과 SOAP를 이용하여 통합할 수 있게 해주었다. 이러한 웹 서비스 시스템에서는 다양한 시스템들과 통합을 제공하기 위한 많은 처리 로직이 필요하기 때문에 시스템이 복잡해진다는 문제점을 가지고 있다. 본 논문에서는 이러한 문제점을 해결하기 위해서 CBD 방법론을 사용하며, 웹 서비스를 이용한 플랫폼에 독립적인 시스템을 설계 및 구현 한다. 제안한 방법이 효율적임을 보이기 위해 운동선수 관리 시스템을 모델로 설계 및 구현한다. 제안한 방법을 이용하여 구현한 시스템은 플랫폼 및 프로그램 언어에 독립적으로 구축이 가능하게 되었고, 시스템의 중요 부분을 컴포넌트로 제작하여 관리 및 유지보수의 효율성을 가져왔다.

1. 서론

웹 서비스는 다양한 IT 환경 내에서 통합 및 상호연계라는 필요성 때문에 대두되었고, 개방되고 표준화된 인터페이스를 기반으로 하여 하드웨어, 플랫폼 및 프로그램 언어에 중립적인 환경을 만들어 준다. 이러한 환경을 구축하기 위해서 XML[8], SOAP[5], WSDL[7], UDDI[6]과 같은 표준기술들을 결합하여 사용한다.

웹 서비스는 다양하게 존재하는 시스템과의 통합을 위하여 많은 처리로직이 필요하다. 따라서 시스템이 복잡해지며, 관리 및 유지 보수가 힘들어 진다. 이러한 문제점을 해결하기 위하여 각 처리 로직을 컴포넌트로 개발하는 것이 필요하다. 본 논문에서는 CBD 기반의 웹 서비스를 이용한 시스템을 설계 및 구현한다. 제안한 기법의 효율성을 실험하기 위하여 운동선수 관리 시스템 PIMSA(Platform Independence Management System for Athlete)을 모델로 하여 설계 및 구현한다. PIMSA의 중요한 부분들을 플랫폼, 시스템으로 나누어 컴포넌트로 구현하고, 이를 이용하여 관리용 웹 사이트 및 웹 서비스 시스템을 구축한다. PIMSA 서비스를 이용하는 클라이언트들을 각기 다른 플랫폼에서 다른 언어를 사용하

여 프로그램을 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 웹 서비스와 CBD에 대하여 알아본다. 3장에서는 CBD 방법론 중의 하나인 UML을 이용하여 설계 및 구현 과정을 살펴본다. 4장에서는 실험 환경을 구축한 후 운동선수 관리 시스템을 이용하는 클라이언트 프로그램들을 구현한다. 마지막으로 5장에서는 결론을 기술한다.

2. 관련 연구

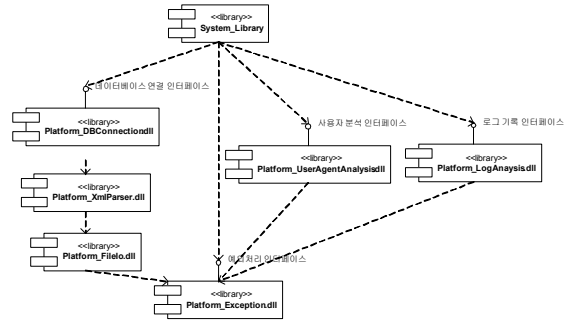
2.1 Web Services

웹 서비스는 외부에 공개 가능하고 웹이나 로컬 네트워크상의 어떤 곳에도 위치할 수 있고, 호출될 수 있는 자기 기술적인 모듈화된 프로그램이다[3]. 웹 서비스의 공급자와 사용자는 XML를 사용하여 운영체제나 언어 환경에 또는 컴포넌트 모델에 관해서는 신경 쓰지 않아도 되며, XML 웹 서비스의 사용하고 있는 운영체제나 언어 환경 또는 컴포넌트 모델에 신경 쓸 필요가 없다. 또한 웹 서비스는 XML, HTTP, SMTP와 같이 흔히 볼 수 있는 개방 인터넷 표준을 기반으로 한다.

2.2 CBD(Component Based Development)

CBD는 컴포넌트의 개발은 물론, 최적의 컴포넌트를 선택하고 이를 조립하여 시스템을 생성하는 새로운 소프트웨어 개발 방법론으로 다음과 같은 장점을 가지고 있다.

CBD는 이미 개발되어 있는 컴포넌트들을 조립함으로써 애플리케이션을 만들 수 있으므로 개발 시간이 단축되고, 소프트웨어 유지보수에 대한 관리가 쉬워진다. 또한 모든 테스트를 끝낸 컴포넌트만 판매되기 때문에 품질이 보증 될 뿐만 아니라 각 기능들이 컴포넌트화되어 개발되기 때문에 해당 컴포넌트들만 따로 변경, 업그레이드할 수 있다.



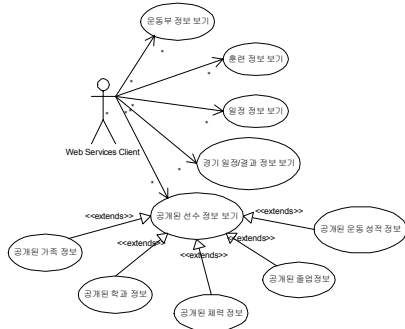
(그림 3) PIMSA 플랫폼 컴포넌트 다이어그램

3. 설계 및 구현

3.1 UML 이용한 PIMSA 설계

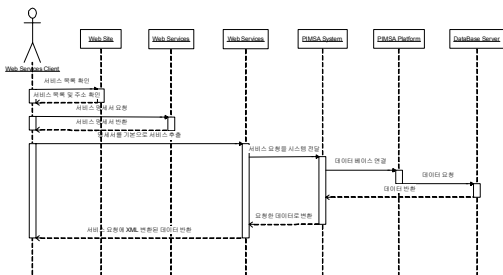
PIMSA의 분석 및 설계 방법으로 UML 채택했으며, 본 논문에서는 유즈케이스, 시퀀스 다이어그램, 컴포넌트 다이어그램, 배치다이어그램을 통해서 설명한다.

웹 서비스 클라이언트는 공개된 서비스에 접속하여 제공되는 서비스를 이용할 수 있다. 그림 1은 웹 서비스 클라이언트에 대한 유즈 케이스 다이어그램이다.



(그림 1) 웹 서비스 클라이언트 유즈 케이스 다이어그램

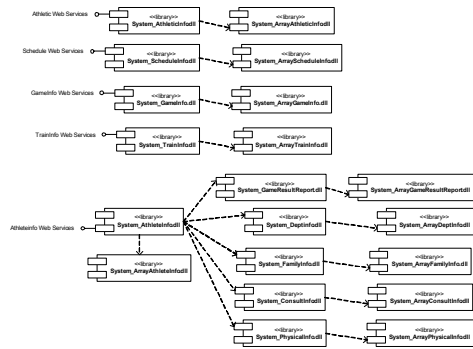
그림 2은 웹 클라이언트가 웹 서비스에서 서비스를 제공받기 위한 절차를 시퀀스 다이어그램을 통해서 보여주고 있다.



(그림 2) 웹 서비스 클라이언트 시퀀스 다이어그램

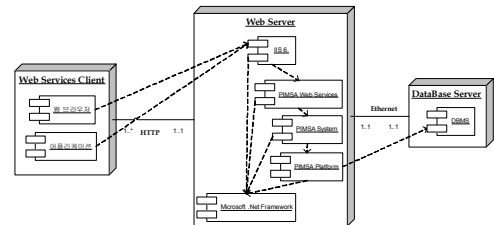
PIMSA 플랫폼은 시스템 구축에 필수적인 요소를 중심으로 하여 컴포넌트화 했다. 그림 3은 PIMSA 플랫폼 컴포넌트 다이어그램이다.

PIMSA 시스템은 운동선수 관리시스템의 기능을 중심으로 하여 컴포넌트화 했다. 그림 5은 PIMSA 시스템 컴포넌트 다이어그램이다.



(그림 4) PIMSA 시스템 컴포넌트 다이어그램

그림 5은 PIMSA의 전체적인 시스템 구성을 보여주고 있는 배치 다이어그램이다.



(그림 5) PIMSA 배치 다이어그램

3.2 CBD기반의 구현

1) 구현 알고리즘

PIMSA 시스템의 핵심 알고리즘은 PIMSA_SERVICES과 PIMSA_SYSTEM 으로 구성된다.

PIMSA_SERVICE 알고리즘

PIMSA_SERVICES 알고리즘은 웹 서비스에서 클라이언트의 요청에 대한 처리를 담당하고 있는 알고리즘이다.

Algorithm PIMSA_SERVICES

Input : Web Services Clients Input reference to WSDL files
 Output : Web Services processing result value (e.g. DataSet, user define type)

```

STEP 1 RECEIVE(clients request)
STEP 2 SERVICES_PROCESSING(using
      PIMSA_SYSTEM Component)
1) GET_INFORMATION(configuration file)
2) WRITE_LOG(user information)
3) GET_RETURN_TYPE(request parameter )
4) IF return value type=DataSet THEN
      CALL PIMSA_SYSTEM(Dataset)
    ELSE
      CALL PIMSA_SYSTEM(array)
    END IF
STEP 3 SEND_RESULT( Web Services Clients)
    
```

PIMSA_SYSTEM 알고리즘

PIMSA_SYSTEM 알고리즘은 웹 서비스에서 요청받은 서비스에 대한 로직을 처리하는 알고리즘이다.

Algorithm PIMSA_SYSTEM

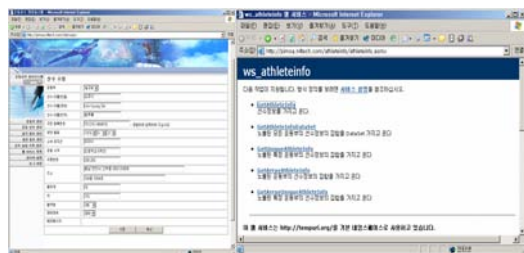
```

Input: input reference to component
specification
Output : System component processing result
value (e.g. DataSet, user define type)

STEP 1 RECEIVED(web services system request)
STEP 2 GET_INFORMATION(sqlquery and
      connection string from database
      configuration files)
STEP 3 COMPONENT_INITIALIZE( using
      database Component)
STEP 4 RECEIVED_DATA(using Database
      Component with sqlquery)
STEP 5 IF return value type=Dataset THNE
STEP 6 SEND_RESULT( DataSet)
STEP 7 ELSE
1) GET_RECORD_COUNT(using dataset)
2) DECLARE_USER_DEFINED_ARRAY
      (using record count)
3) while( record count!=end)
      INSERT_USER_DEFINED_ARRAY(Dataset)
4) SEND_RESULT(user defined array)
      END IF
    
```

2) 시스템 구현

그림 6은 관리 웹 사이트와 웹 서비스의 화면을 보여주고 있다.

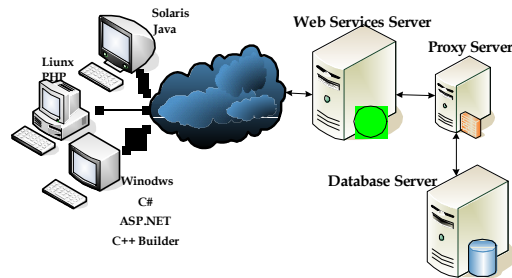


(그림6) PIMSA의 구축된 화면

4. 실험 및 결과

4.1 실험 환경

PIMSA는 윈도우 2003에서 .NET 프레임워크와 IIS 6.0을 이용하여 웹 사이트 및 웹 서비스를 제공할 수 있도록 구성하였다. 데이터베이스 서버는 MS-SQL 2000이 사용되었다. 클라이언트들은 윈도우, 리눅스, 솔라리스 환경에서 실험할 수 있도록 환경을 구축하고 클라이언트 프로그램들을 구현하였다.



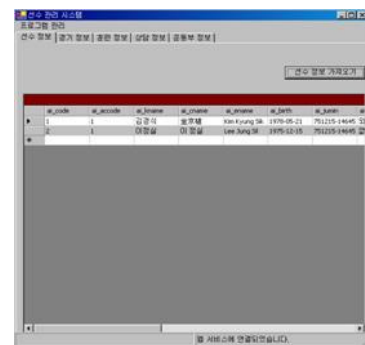
(그림 7) 실험 환경

4.2 클라이언트 프로그램

클라이언트 프로그램들은 PIMSA에서 제공하는 서비스를 이용하여 구현하였다.

1) 윈도우 환경

윈도우 2003 서버에서 .NET 프레임워크와 MSTK2[2] 툴킷을 이용하여 클라이언트 프로그램을 구현하였다.



(그림 8) 윈도우 환경에서 C#를 이용하여 구현한 클라이언트 프로그램

2) 유닉스 환경

솔라리스에서 JDK 1.4 SDK와 아파치 AXIS[1]를 이용하여 클라이언트 프로그램을 구현하였다.

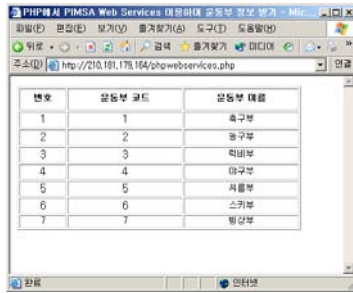


(그림 9) 솔라리스 환경에서 자바를 이용하여 구현한 클라이언트 프로그램

3) 리눅스 환경

리눅스의 Apache서버에서 PEAR::SOAP[4]를 이용하여 클라이언트 프로그램을 구현하였다.

[8] Extensible Markup Language, <http://www.w3.org/XML/>, 2003



번호	운동부 코드	운동부 이름
1	1	축구부
2	2	농구부
3	3	탁구부
4	4	배구부
5	5	서클부
6	6	스카우
7	7	빙상부

(그림 10) 리눅스 환경에서 PHP를 이용하여 구현한 클라이언트 프로그램

4.3 실험 결과

웹 서비스 클라이언트들은 플랫폼 및 프로그래밍 언어에 상관 없이 웹 서비스에서 제공하는 서비스를 받을 수 있었다. 또한, 웹 서비스는 비즈니스 로직에 대한 처리를 담당하고 제공함으로써 클라이언트에서는 시스템에 대한 전체적인 이해 없이 제공되는 인터페이스에 대한 이해만으로 프로그램 구현이 가능하였고, 이것을 이용하여 구현 시간의 단축 가져왔다. 시스템의 핵심부분을 플랫폼 및 시스템을 구분하여 컴포넌트로 구현한 후, 이것을 이용하여 시스템을 구축함으로써 개발의 기간의 단축을 가져왔고, 컴포넌트화된 부분에 따라 수정 및 업데이트할 수 있어서 관리 및 유지보수가 쉬워졌다.

5. 결론

본 논문에서는 웹 서비스 기술과 컴포넌트 개발방법인 CBD를 이용하여 운동선수 관리시스템인 PIMSA를 설계 및 구현하였으며, 각기 다른 플랫폼에서 다른 언어를 사용하여 PIMSA 서비스를 이용하는 클라이언트 프로그램을 구현하였다. 시스템의 중요 부분을 컴포넌트로 제작하여 유지보수 및 관리가 쉬워졌으며, 웹 서비스를 이용하여 플랫폼 및 프로그래밍 언어에 독립적인 시스템 구축이 가능하게 되었다.

참고 문헌

- [1] Axis Project, <http://ws.apache.org/axis/java>, 2003
- [2] MSDN, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/WebServicesAnchor.asp>, 2003
- [3] Patrick C., "Professional XML Web Services", wrox, 2002
- [4] PHP Extension and Application Repository, <http://pear.php.net>, 2003
- [5] Simple Object Access Protocol, <http://www.w3.org/2002/ws/>, 2003
- [6] Universal Description, Discovery and Integration, <http://www.uddi.org>, 2003
- [7] Web Sservices Description Language, <http://www.w3.org/2002/ws/>, 2003