

# WIPI 기반의 XML를 이용한 야구 중계 시스템 설계 및 구현

이경미\*, 윤성대\*\*  
\*부경대학교 전산정보학과  
\*\*부경대학교 전자계산학과  
\*rorod@naver.com

## A Design and Implementation of a Baseball Broadcast System Using XML based on WIPI

KyungMi Lee\*, SungDae Youn\*\*

\*Dept. of Computer and Information, Pukyong National University

\*\*Dept. of Computer Science, Pukyong National University

### 요 약

기존의 다양한 무선인터넷 플랫폼이 WIPI(Wireless Internet Platform for Interoperability)라는 무선인터넷 플랫폼으로 표준화되면서 플랫폼 간 콘텐츠 호환성을 보장하고, 플랫폼의 이식의 용이성을 제공하여 다양하고 풍부한 콘텐츠들이 개발되고 있다. 본 논문에서는 다양한 어플리케이션간에 데이터를 교환하고 공유하기 편리한 XML 데이터를 생성하고, 실시간으로 야구 중계 서비스를 제공하는 시스템을 설계 및 구현하였다. 이러한 데이터서비스 교환에 XML을 적용함으로써 클라이언트와 서버 사이의 통신을 표준화하였다.

### 1. 서론

CDMA(Code Division Multiple Access)의 도입은 아날로그 사회에서 디지털 사회로의 대 변혁을 의미하며 WIPI의 도입 또한 콘텐츠와 시스템 간 호환을 위한 디지털적인 변화의 시도라고 볼 수 있다. WIPI(Wireless Internet Platform for Interoperability)는 각종 무선인터넷 응용 콘텐츠를 하나의 표준 플랫폼을 통하여 수용할 수 있게 하는 미들웨어로써, 콘텐츠를 개발하는 CP(Contents Provider)들의 소프트웨어 개발비용을 최소화 하도록 하며 C언어와 Java 언어를 위한 규격을 모두 지원함으로써 다양한 콘텐츠 개발자들을 수용할 수 있도록 하였다[1].

현재, WIPI 콘텐츠 시장을 확대하여 개발 업체들의 선의의 경쟁을 유도하는 다양한 기술개발 및 마케팅 지원 등을 적극적으로 펼쳐 위피 이용자들이 보다 많은 콘텐츠와 양질의 서비스를 받을 수 있도록 연구가 활발히 진행중이다.

또한 확장성 표기언어인 XML[2]이 웹 상의 콘텐츠 표현의 표준으로 결정됨에 따라 무선 데이터서비스에서 XML을 이용하면 데이터의 공용성과 프로그램의 유연성을 향상시키는데 필요한 비용과 노력을 절감할 수 있다.

본 논문에서는 국내 표준 모바일 플랫폼 규격인 WIPI 기반의 XML을 이용하여 실시간으로 야구 중계 서비스를 위한 시스템을 설계 및 구현한다. 모바일 환경에서 XML 데

이터 서비스는 HTTP를 기반으로 하여 분산 환경에서 시스템 간의 정보를 효율적으로 교환할 수 있으며 CPU부하를 덜어주고 무선 장치의 메모리, 네트워크 대역폭의 부하를 현저하게 줄일 수 있는 시스템 설계 및 구현하였다.

본 논문의 구성은 다음과 같다. 제2장은 관련 연구, 제3장은 시스템의 구성, 제4장은 시스템 구현 및 검증, 마지막으로 결론 및 향후과제로 구성된다.

### 2. 관련연구

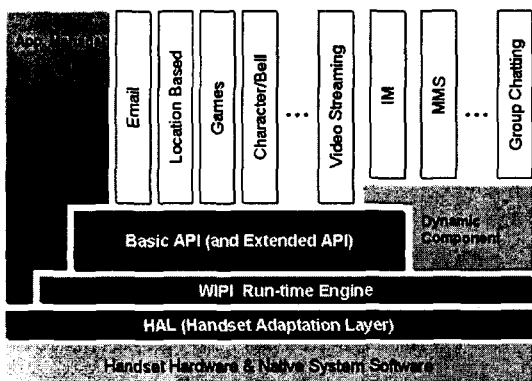
#### 2.1 WIPI

WIPI는 한국 무선 인터넷 표준화 포럼(Korea Wireless Internet Standardization Forum)에서 만들어진 모바일 표준 플랫폼 규격으로 이동통신 단말기에 탑재되어 무선 인터넷을 통해 다운로드 된 응용 프로그램 실행 환경을 제공하는데 필요한 표준 규격이다. 모바일 플랫폼 표준화 범위는 이동통신 사업자들의 요구 사항이 단말기가 최종적으로 다운로드 되는 오브젝트가 기계코드(machine code) 형태를 요구함에 따라 콘텐츠 호환을 보장하는 범위 내에서 다양한 솔루션이 개발될 수 있도록 되어 있다.

지원언어는 C, Java가 동시에 지원되는 구조이며, 플랫폼과 어플리케이션은 하드웨어에 독립적인 구현이 가능하도록 CPU, LCD, 메모리 등이 단말기 하드웨어나 단말기가

사용하는 OS(Operating System)에 관계없이 실행과 이식 (Porting)이 용이하도록 하였다[6].

WIPI 플랫폼의 전체적인 시스템 구조는 그림 1과 같다.



(그림 1) WIPI의 시스템 구조도

그림 1에서 하단에 있는 단말기 기본 소프트웨어는 CDMA 망에서는 Rex OS를 지칭하는 것으로 간단한 단말기 운영체제 기능과 통신 기능 및 각종 디바이스 드라이버가 포함된다.

HAL(Handset Adaptation Layer)은 단말기 제조사를 위한 API(Application Programming Interface)를 정의한 것으로 단말기 제조사마다 서로 다른 기기들을 지원하기 위해 HAL이라고 하는 추상화 계층을 도입한 것으로 WIPI 플랫폼에서 획기적인 것으로 받아들여지고 있다. 그리고 이 HAL이 단말기에 이식(Porting)되면 바로 WIPI 플랫폼 실행 엔진을 탑재할 수 있다. 데스크톱 윈도우즈 환경에서는 HAL을 WIN32에 맞게 이식(Porting)하면 애플레이터가 바로 되는 것이다.

모바일 표준 플랫폼은 WIPI 응용 프로그램을 실행시키는 실행 엔진으로서 다운로드 받은 binary WIPI 응용 어플리케이션을 실행시키기 위해 링크 & 로더 기능, 메모리 관리, 리소스 관리, 가비지 컬렉션 기능 등을 수행한다. Basic API는 WIPI 응용 프로그램 개발자를 위한 C 및 Java API를 말한다. 이 중에서 WIPI가 표준화의 대상으로 하고 있는 것은 HAL 계층, Basic C API, Basic Java API이고 실행 엔진은 표준화의 대상이 아니다.

## 2.2 XML(Extensible Markup Language) 파서

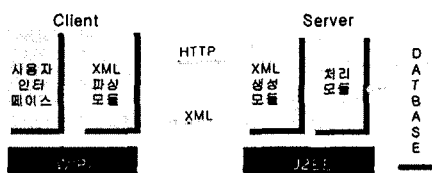
일반적으로, XML 문서는 element, attribute, entity, DTD(Document Type Definition) 등 4개의 주요 구성 요소가 있다. element는 태그 데이터를 표현하는 것이고, attribute는 element에 대한 정보를 더하기 위해 사용된다. entity는 XML문서에서 참조할 수 있는 데이터의 실질적 값이다. DTD는 개개의 XML 문서의 구조를 정의하는 XML의 선택 부분이다. XML 파싱 작업은 문서로부터 태그의 의미와 구조에 기초하여 이러한 구성 요소를 추출해

내는 것을 의미한다. XML 파서는 XML문서를 읽어서 이벤트를 발생해주는 형식의 SAX(Simple API for XML Processing) 파서와 XML 문서를 읽어서 메모리에 객체 형식으로 로딩하는 DOM(Document Object Model) 파서로 분류된다. DOM은 메모리에 트리 구조로 XML 문서 전체를 읽으며, 트리의 각 노드는 문서의 데이터를 나타낸다. SAX는 callback 메소드를 통해 파싱된 XML 구성 요소를 이벤트로 전달한다. 일반적으로 SAX가 DOM보다 CPU, 메모리 등을 적게 소모하고, 수행 속도가 빠르다. 그러나 SAX는 XML 데이터에 순차적인 접근만 가능하며, 문서의 수정, 추가가 어렵다. 반면 DOM은 데이터의 접근이 쉬우며, XML 문서를 조작하기가 용이하다[3, 7].

본 논문에서 구현한 시스템에 사용되는 XML 데이터 처리를 위해 DOM 파서 방식을 사용하였다.

## 3. 시스템 구성

본 논문에서 구현한 WIPI 기반의 실시간 야구 중계 시스템의 전체 구성도는 그림 2와 같다.



(그림 2) 시스템의 전체 구성도

그림 2에서, 먼저 Client의 사용자인터페이스를 통해 야구중계서비스를 그림으로 제공받을 것인지 문자로 제공받을 것인지에 대한 응답을 받는다. 이 정보는 Server의 처리 모듈로 보내져 데이터베이스에 접근한다. 데이터베이스의 데이터 및 현재 경기의 진행 상태에 대한 결과는 XML 생성 모듈로 전달되고 XML 생성 모듈에서 정보를 XML 코드로 변환해 Client 즉 모바일 폰으로 다시 전달한다. Client의 XML Parsing 모듈에서 XML코드를 파싱한 뒤 결과를 사용자 인터페이스를 통해 사용자에게 보여준다.

Client에서 Server로 정보를 보낼 때는 HTTP 커넥션을 사용하고, Server에서 Client로 보내는 정보는 복잡하기 때문에 XML은 이런 형태의 데이터 교환에 매우 적합하다. Client는 WIPI 환경에서 작성한 Jlet[8]으로 구현하고 Server는 J2EE[4]환경에서 작성한 JSP, Servlet으로 구현할 것이다.

### 3.1 서버 모듈

서버 모듈은 XML 생성 모듈, 처리 모듈, 객체 관계형 데이터베이스로 구성되어 있다. Client로부터 사용자 요청이 전송되면 야구 중계 시스템에 요구되는 데이터를 데이터베이스에서 가져오고 현재 진행되고 있는 경기상황의 결과를 XML 생성 모듈에서 XML 코드로 변환한 뒤 Client모듈로

전송한다. 데이터베이스는 경기상황의 결과를 데이터하여 저장하고 있다.

### 3.1.1 처리 모듈

Client가 사용자의 응답을 받으면 Server의 처리 모듈로 전송된다. 처리 모듈은 현재 진행되고 있는 야구 경기 진행 및 결과를 XML 생성 모듈로 전달한다.

### 3.1.2 XML 생성 모듈

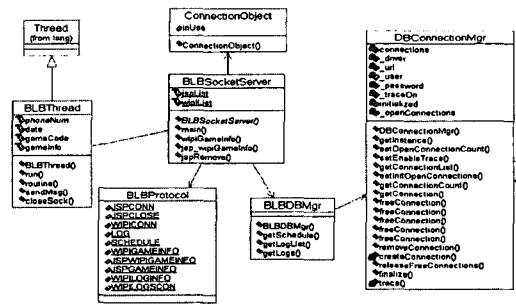
XML 생성 모듈에서는 클라이언트에 보낼 정보를 XML 코드로 변환한다. 사용자에게 보낼 정보는 그래픽과 텍스트 두 가지로 나뉜다. 그래픽은 사용자가 야구 중계를 그래픽으로 보기위한 정보를 포함하고 텍스트는 야구 중계를 문자로 보기위한 정보를 포함한다. 그림 3은 이러한 두 가지 정보를 표현하기 위한 XML의 Schema를 보여준다.

```
<?xml version="1.0"?>
<xsd:schema targetNamespace="http://www.blbssystem.org/sample"
  xmlns="http://www.blbssystem.org/sample"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="BLBService">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="schedule" maxOccurs="unbounded"/>
      <xsd:element ref="logs" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="graphic" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="schedule">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="vteam" type="xsd:string"/>
      <xsd:element name="hteam" type="xsd:string"/>
      <xsd:element name="date" type="xsd:dateTime"/>
    </xsd:sequence>
    <xsd:attribute name="sid" type="xsd:ID" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="logs">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="inning">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:group ref="ingroup"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="log_sid" type="xsd:IDREF" use="required"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="graphic">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="inning" type="xsd:string"/>
      <xsd:element name="picher" type="xsd:string"/>
      <xsd:element name="hiter" type="xsd:string"/>
      <xsd:element name="sbo" type="xsd:string"/>
      <xsd:element name="rhbe" type="xsd:string"/>
      <xsd:element name="runner" type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="g_sid" type="xsd:IDREF" use="required"/>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

```
</xsd:element>
<xsd:group name="ingroup">
  <xsd:sequence>
    <xsd:element name="title" type="xsd:string"/>
    <xsd:element name="contents" type="xsd:string"/>
  </xsd:sequence>
</xsd:group>
</xsd:schema>
```

(그림 3) XML 정보에 대한 스키마

위의 스키마를 통해 작성된 XML코드를 Client로 보낸다. 그리고 그림 4는 서버 모듈에 대한 클래스들 사이의 관계를 모델링 한 것이다.



(그림 4) 서버 모듈 클래스 다이어그램 설계

### 3.2 클라이언트 모듈

Server에서 보내온 XML 코드는 Client의 XML 파싱 모듈에서 파싱한다. 그리고 사용자 인터페이스에서 결과를 보여주거나 사용자의 응답을 받는다.

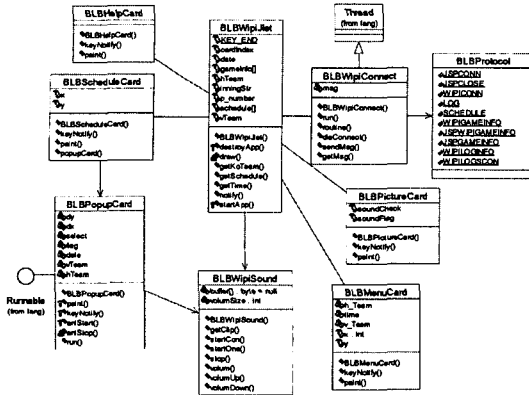
#### 3.2.1 XML 파싱 모듈

Server로부터 받은 XML 코드를 파싱한다. 파싱을 통해 그래픽 서비스에 한 정보와 문자 서비스에 대한 정보를 얻게 되는데 이 정보를 각각의 Card 객체[8]에 적재되어 Display 객체[8]에 등록한다. Display는 휴대폰 단말기의 디스플레이에 나타날 수 있는 Jlet내의 객체로써 여러 개의 생성이 가능하다.

#### 3.2.2 사용자 인터페이스

WIP에서 제공하는 High Level UI(User Interface)와 Low Level UI(User Interface)[8]를 이용하여 휴대폰 사용자를 위한 인터페이스를 설계하였으며, 그 내용을 보면 다음과 같다. 사용자가 로그인에 성공하면 메뉴에 해당하는 Card객체를 담은 Display객체를 화면에 보인다. 각 메뉴는 다른 카드와 링크되어 세부 정보를 표현한다. 그리고 사용자가 야구 중계 서비스를 그래픽 또는 문자로 제공받을 것인지를 선택하면 실시간으로 업데이트된 Display를 보여준다. 그림 5는 클라이언트 모듈에 대한 클래스들 사이의

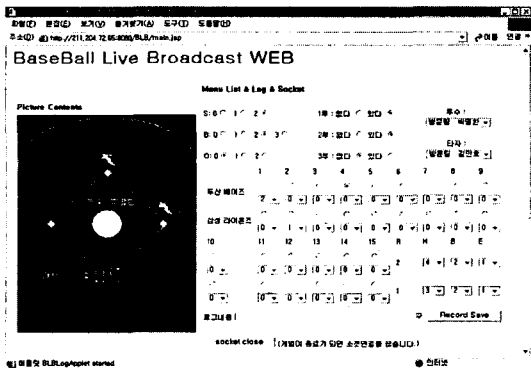
관계를 모델링 한 것이다.



(그림 5) 클라이언트 모듈 클래스 다이어그램 설계

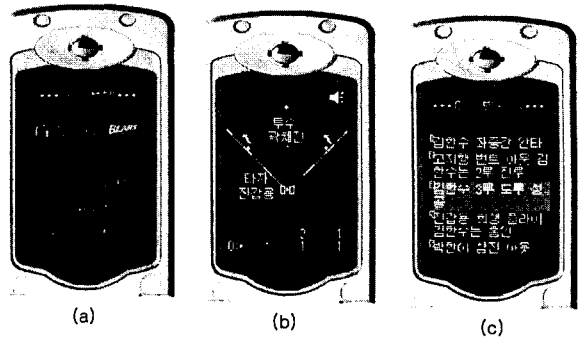
4. 시스템 구현 및 검증

WIPI 단말기용 응용 프로그램은 Jlet을 이용하고 서버용 프로그램은 JAVA[5]를 이용하여 개발하였고 그래픽은 Applet[5]을 이용하여 디자인하였다. 개발한 Jlet 응용 프로그램을 AromaWIPI-Emulator1.1.1.7[1]를 이용하여 동작검증을 수행하였으며, 그림 6은 서버 모듈에서 사용자에게 야구 중계 서비스를 위한 웹 인터페이스 화면이다.



(그림 6) 실시간 야구 중계 서비스 화면

그리고 그림 7은 WIPI 기반의 야구 중계 시스템의 클라이언트 화면이다. (a)는 사용자가 야구 중계 서비스를 받기 위한 메뉴화면이다. 사용자가 메뉴를 선택하면 서버에서는 야구 진행 상황의 결과를 XML 코드로 변경해 Client 보낸다. Client에서는 XML코드를 파싱하고 정보를 Display객체에 적재해 화면에 보인다. (b)는 그래픽으로 경기 진행을 보기위한 화면이며 (c)는 문자로 서비스를 받기위한 화면이다. (b), (c)화면은 서버에서 데이터를 전송할 때마다 실시간으로 자동 업데이트 되어 경기 진행 및 결과를 볼 수 있다.



(그림 7) 실시간 야구 중계 시스템 수행 화면

5. 결론

본 논문에서는 WIPI 기반의 서버-클라이언트 모델을 사용한 실시간 야구 중계 시스템을 설계하고 이를 구현하였다. 또한 XML을 적용함으로써 클라이언트와 서버 사이의 통신을 표준화하였으며, 사용자에게 WIPI 단말기를 이용하여 어디서나 실시간 야구 중계 서비스를 제공해 줄 수 있는 장점을 가지고 있다. 또한, 모바일 환경에서 XML 데이터 서비스는 HTTP를 기반으로 하여 분산 환경에서 시스템 간의 정보를 효율적으로 교환하고, CPU부하를 덜어주며, 무선 장치의 메모리, 네트워크 대역폭의 부하를 현저하게 줄일 수 있는 시스템 설계 및 구현을 하였다.

향후 연구로는 좀 더 다이나믹한 그래픽과 사용자 편의성을 위한 Client상의 사용자 인터페이스 전환이 필요하다.

<참고문헌>

[1] MobileJAVA, <http://www.mobilejava.co.kr>.  
 [2] Harvey M., etc., "How to Program XML", Prentice Hall, December, 2002.  
 [3] C.Kirkegaard, A.Moller, M.I.Schwartzbach, "Static analysis of XML transformations in Java", IEEE Trans on SE, Vol.30, No.03, pp.0181-0192, 2004.  
 [4] Wrox Author Team, "Beginning J2EE 1.4", WROX Press, March, 2003.  
 [5] James Hart, "Java J2SE 1.4 Core Platform Update", WROX Press, March, 2002.  
 [6] 이상윤 외, "한국 무선 인터넷 표준 플랫폼(WIPI)의 표준화 현황 및 발전 전망", 한국 정보 과학회지, 22권 1호, 2004년 1월.  
 [7] 강미연 외, "WIPI 기반의 모바일 단말을 위한 내장형 XML 파서 및 뷰어", 정보과학회 2004년 춘계학술대회, 31권 1호, 2004년 4월.  
 [8] 박수원 외, "위피 모바일 프로그래밍", 한빛미디어, 2003.