

공학석사 학위논문

WIPI 기반의 XML을 이용한
모바일 야구 중계 시스템의
설계 및 구현



이 논문을 공학석사 학위논문으로 제출함.

2005년 8월

부경대학교 산업대학원

전산정보학과

이 경 미

이경미의 공학석사 학위논문을 인준함

2005년 6월 17일

주 심 이학박사 이 경 현



위 원 공학박사 여 정 모



위 원 이학박사 윤 성 대



<차례>

그림 차례	ii
표 차례	iii
Abstract	iv
I. 서론	1
II. 관련연구	4
2.1 국내 모바일 플랫폼	4
2.2 WIPI (Wireless Internet Platform for Interoperability)··	6
2.3 XML(eXtensible Markup Language) 파서	8
III. 제안 시스템 설계 및 구현	9
3.1 제안 시스템 모델	9
3.2 XML 데이터 교환 모델	11
3.3 중계 Server 설계	16
3.4 모바일 클라이언트 설계	20
3.5 기록 웹 서버 설계	25
3.6 구현 결과	28
IV. 결론 및 향후 연구 과제	32
참고문헌	33

<그림 차례>

(그림 1) WIPI의 개념적 구조도	6
(그림 2) 야구 중계 시스템의 모델	9
(그림 3) 시스템의 구성도	10
(그림 4) 야구 중계 서비스를 위한 데이터 교환 모델 구조...	12
(그림 5) 야구 중계 정보에 대한 XML Schema	14
(그림 6) 경기 일정에 대한 XML 문서	15
(그림 7) 중계 서버 클래스 다이어그램	17
(그림 8) XML 문서의 생성	19
(그림 9) 모바일 클라이언트 클래스 다이어그램	20
(그림 10) XML 문서의 질의 모듈	22
(그림 11) 사용자 인터페이스 모듈 클래스 다이어그램...	23
(그림 12) 기록 웹 서버의 JSP 상태 천이도	26
(그림 13) 중계 서버 실행 화면	28
(그림 14) 실시간 기록 웹 서버 실행 화면	30
(그림 15) 모바일 클라이언트의 실행 화면	31

<표 차례>

(표 1) 국내의 모바일 플랫폼 사용 현황	4
(표 2) 중계 서버 개발 및 사용 환경	16
(표 3) 모바일 클라이언트의 개발 및 사용 환경	20
(표 4) 사용자 인터페이스를 구성하는 주요 클래스	24
(표 5) 기록 웹 서버의 개발 및 사용 환경	25

A Design and Implementation of Baseball Mobile Broadcasting System Using XML Based on WIPI

Kyung-Mi Lee

*Dept. of Computer and Information Graduate School of
Pukyong National University*

Abstract

The existing various mobile Internet platforms are united with the standardized WIPI(Wireless Internet Platform for Interoperability) which guarantees each other's platform data compatibility and provides the portability of platform. Above all things, it is necessary to increase quantitative and qualitative supply of content for using mobile internet technology efficiently. Consequently the resent mobile service is changing into a system circumstance which offers various contents-services to customer.

In this paper, we have designed and implemented a baseball broadcasting system based on WIPI using XML(eXtensible Markup Language) for information management and data exchange efficiently. In the proposed system, data transmission is based on XML, the standardized data-format. So any platform which supports TCP/IP and HTTP can access data without any conversion. We achieved the communication

standardization between clients and servers applying XML to the data service exchange.

I. 서론

현재의 인터넷 기술은 일반 사용자의 욕구를 충분히 반영하고 남을 만큼 발전되어 왔다. 특히 유선인터넷의 초고속화와 무선인터넷의 등장은 인간의 정보활동 능력을 극대화하고 있으며 무선인터넷을 사용하는 이용자들은 유선인터넷을 통하여 할 수 있는 일들을 무선인터넷을 통해서도 할 수 있기를 바라며, 사용자들의 이런 욕구를 충족시켜 주려면 무선인터넷 환경에 맞는 콘텐츠가 개발되어야 한다. 그러나 무선인터넷을 사용하는 이용자들은 늘어나는 반면 이들의 욕구를 충족시켜 줄 수 있는 차별화 되어 있는 양질의 서비스를 제공하기 위한 콘텐츠가 절대적으로 부족한 실정이다[10].

현재 이동통신사의 평균 콘텐츠 중 90%이상이 게임이고 나머지 10%정도만 서비스 콘텐츠에 불과하다. 사용자들의 다양한 욕구를 충족시키고 또 효과적이고 실용적인 콘텐츠 서비스 개발이 시급한 상태이다[9].

또한 무선인터넷은 수십 여 종의 다양한 단말기 상에서의 제한된 콘텐츠 접근과 속도 한계, 그리고 각 이동통신사 간의 데이터 호환성 문제를 내포하고 있다. 이러한 상황적 무선인터넷 확산이라는 문제를 해결하기 위해 이동통신 3사를 비롯한 단말기 제조업체와 CP(Content Provider)들이 공통적으로 제시한 부분이 무선인터넷 표준화이며 무선인터넷 서비스 활성화 이후 다양한 서비스를 통해 다각적인 변화를 추진하고 있다. 물론 인터넷 표준화 바람은 국내뿐만 아니라 IETF(Internet Engineering Task Force) 등과 같은 국제적인 부분에서 거세게 불고 있다. 이에 따라 이동통신사의 다양한 플랫폼으로 인해 발생하는 여러 가지 문제점을 극

복하기 위해 단말기와 콘텐츠 개발에 독립적이고 개방된 국내 모바일 플랫폼 표준 WIPI(Wireless Internet Platform for Interoperability)가 이동통신 3사를 비롯하여 전파연구소, 전자통신연구원, 한국통신기술협회 및 개발자 등 여러 전문가 집단의 노력 끝에 탄생하였다.

CDMA(Code Division Multiple Access)의 도입은 아날로그 사회에서 디지털 사회로의 대변혁을 의미하며 WIPI의 도입 또한 콘텐츠와 시스템 간 호환을 위한 디지털적인 변화의 시도라고 볼 수 있다. WIPI는 각종 무선인터넷 응용 콘텐츠를 하나의 표준 플랫폼을 통하여 수용할 수 있게 하는 미들웨어(middleware)로써, 콘텐츠를 개발하는 CP들의 소프트웨어 개발비용을 최소화 하도록 하며 C언어와 Java 언어를 위한 규격을 모두 지원함으로써 다양한 콘텐츠 개발자들을 수용할 수 있도록 하였다[15, 18].

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

또한 확장성 표기언어인 XML(eXtensible Markup Language)[4]이 웹상의 콘텐츠 표현의 표준으로 결정됨에 따라 무선 데이터서비스에서 XML을 이용하면 데이터의 공용성과 프로그램의 유연성을 향상시키는데 필요한 비용과 노력을 절감할 수 있으며, HTTP를 기반으로 하여 분산 환경에서 시스템 간의 정보를 효율적으로 교환할 수 있다.

본 논문에서는 국내 표준 모바일 플랫폼 규격인 WIPI 기반의 XML을 이용하여 실시간으로 야구 중계 서비스를 위한 시스템을 설계 및 구현한다. 제안한 시스템에서 데이터 전송은 표준화된 데이터 포맷인 XML을 기반으로 설계하여 TCP/IP 및 HTTP를 지원하는 모든 플랫폼에서 데이

터 접근이 가능하며, 별도의 변환과정 없이 데이터를 사용할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에 필요한 관련 연구를 살펴보고, 3장에서는 제안 시스템의 데이터 전송을 위한 XML 데이터 교환 모델에 대해 설명하고, 이를 바탕으로 WIPI 기반의 XML을 이용한 야구 중계 시스템의 설계 및 구현에 대해 기술하고, 마지막으로 4장에서 결론 및 향후 연구방향을 제시한다.

II. 관련연구

본 장에서는 국내 모바일 플랫폼 현황 및 WIPI 개발 배경, 그리고 모바일 클라이언트의 기반이 되는 WIPI와 본 논문에서 이용하게 될 클라이언트용 XML Parser에 대해 살펴본다.

2.1 국내 모바일 플랫폼

국내의 모바일 플랫폼 사용 현황은 표 1과 같다.

<표 1> 국내의 모바일 플랫폼 사용 현황

플랫폼	개발언어	추진사	개발사	수행방식
SK-VM	Java	SKT	XCE	인터프리터(VM)
JavaStation	Java	LGT	Aromasoft, VeloxSoft	인터프리터(VM)
GVM	C/C++	SKT	Sinjisoft	인터프리터(VM)
BREW	C/C++	KTF	Qualcomm	바이너리
WIPI	Java/C	SKT/KTF/LGT	Aromasoft, Geaesoft, Geotel, InoAce, Veloxsoft, XCE	바이너리

표 1에서 알 수 있듯이 우리나라의 이동통신 3사도 자사의 환경에 맞춘 모바일 플랫폼을 적용하고 있다. 각 회사의 모바일 플랫폼이 다르면 자사의 플랫폼에 맞춰 개발한 콘텐츠는 서비스할 수 있지만 타사의 플랫폼에 맞춘 콘텐츠는 서비스할 수 없게 된다. 즉, 이러한 플랫폼의 난립은 서비스를 제공하는 이동 통신사뿐만 아니라 사용자에게도 손실을 준다.

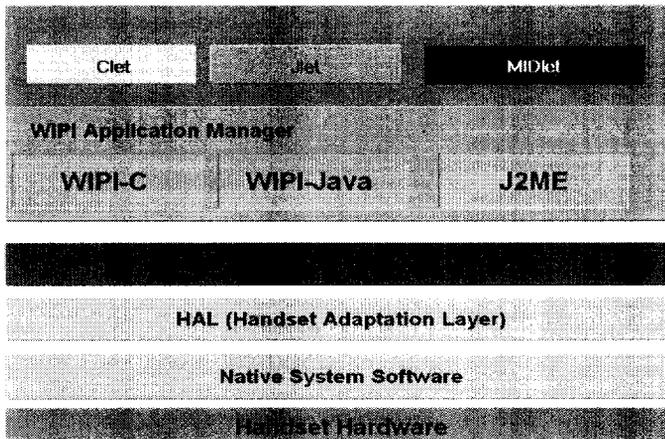
모바일 서비스가 가능하기 위해서는 각기 다른 세 분야에 있는 구성원들이 협력하여야 한다. 첫 번째는 하드웨어 디바이스를 제작하는 단말기 회사이다. 두 번째는 사용자가 이용할 콘텐츠를 개발하는 콘텐츠 제공자이며, 세 번째는 콘텐츠를 서비스해 주는 이동 통신사이다. 즉, 이 세 구성원의 협력으로 모바일 서비스는 이루어지는 것이다. 그러나 과거처럼 이동 통신사들의 플랫폼이 다르면 단말기 제조 회사에서는 각각의 회사에 맞게 단말기를 생산하여야 하는 번거로움이 있을 수 있다. 콘텐츠 제공자 역시 하나의 콘텐츠를 각각의 플랫폼에 맞게 다시 제작해야 하는 번거로움이 생기게 되고, 또한 동일한 시간에 많은 양의 콘텐츠를 개발할 수 없게 된다. 이러한 비효율적인 이유로 인해 이동 통신사에서도 다양한 서비스를 제공할 수 없게 된다. 이와 같은 문제점을 해결하기 위해 무선 인터넷 플랫폼의 표준화가 진행되었고, 그 결과 WIPI라는 무선 인터넷 표준 플랫폼이 개발되었다. 이는 단말기 제조회사 및 콘텐츠 제공자와 이동통신사 입장에서 단일 노동으로 많은 이익을 창출할 수 있어 유익해질 것이며, 서비스를 이용하는 사용자 입장에서 다양한 서비스를 이용할 수 있는 장점을 갖게 된다[17].

2.2 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 프로그램의 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 등에 대해 단말기 하드웨어나 단말기가 사용하는 OS(Operating System)에 관계없이 실행과 이식(porting)이 용이하도록 하였다[11].

WIPI 플랫폼은 개념적으로 그림 1과 같은 구조를 갖는다.



(그림 1) WIPI의 개념적 구조도

그림 1의 하단에 있는 단말기 기본 소프트웨어는 통신 기본 기능과 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 에뮬레이터가 바로 되는 것이다.

WIPI 응용 프로그램을 실행시키는 실행 엔진은 다운로드 받은 binary WIPI 응용 어플리케이션을 실행시키기 위해 링크와 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.

Clet과 Jlet은 모바일 표준 플랫폼 규격에 따라 작성된 C언어 응용프로그램과 Java 언어 응용프로그램이며 MIDlet은 MIDP 규격에 따라 작성된 Java 언어 응용프로그램이다[16].

2.3 XML 파서

일반적으로 XML 문서는 element, attribute, entity, XML Schema 등 4개의 주요 구성 요소를 가진다. element는 태그를 표현하는 것이고, attribute는 element에 대한 정보를 더하기 위해 사용된다. entity는 XML문서에서 참조할 수 있는 실질적 데이터이다. XML Schema는 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 형식인 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 메서드를 통해 파싱된 XML 구성 요소를 이벤트로 전달한다. 일반적으로 SAX가 DOM보다 CPU, 메모리 등을 적게 소모하고, 수행속도가 빠르다. 그러나 SAX는 XML 데이터에 순차적인 접근만 가능하며, 문서의 수정, 추가가 어렵다. 반면 DOM은 데이터의 접근이 쉬우며, XML 문서를 조작하기가 용이하다[1, 14].

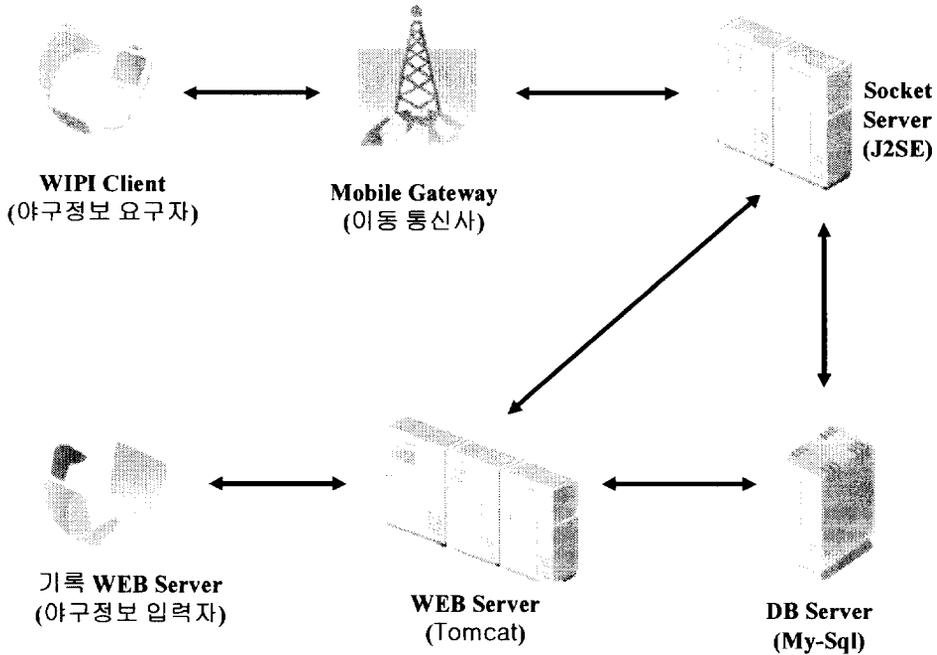
본 논문에서 구현한 시스템에서는 야구 중계 서비스를 위해 사용하는 XML 문서에 대해 접근과 조작이 필요하므로, XML 파서는 XML 데이터 처리를 위해 DOM 파서 방식을 사용하였다.

Ⅲ. 제안 시스템 설계 및 구현

본 장에서는 데이터 교환 모델로서 XML의 역할과 XML 문서 생성 과정, 구조 및 설계에 대해서 설명한다. 그리고 이를 바탕으로 XML 기반의 야구 중계 시스템의 설계 과정 및 구현에 대해 기술한다.

3.1 제안 시스템 모델

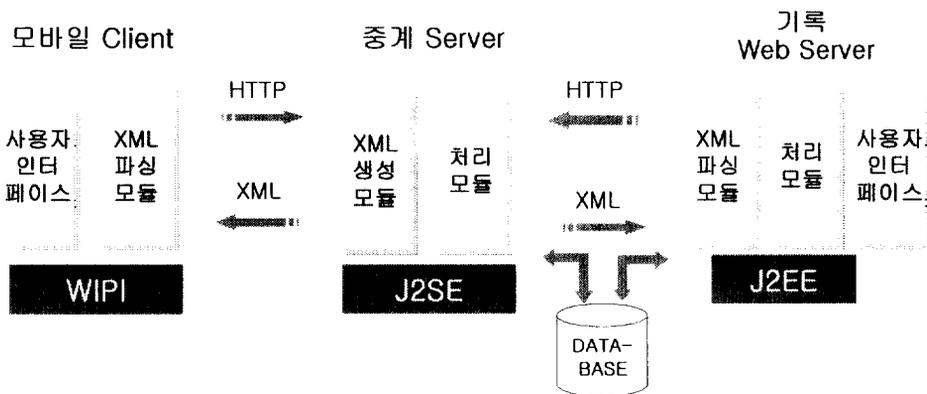
본 논문에서 구현한 야구 중계 시스템의 모델은 그림 2와 같다.



(그림 2) 야구 중계 시스템의 모델

그림 2에서 기록 웹 서버는 야구 중계 정보를 입력과 관리를 위해서 JSP(Java Server Page) 페이지에 기록을 하는 역할을 한다. 이렇게 입력된 데이터들은 Tomcat 웹 서버를 통해서 DB Server(MySql)에 입력이 되고 소켓 서버(Socket Server)와의 통신을 통해서 request, response 값들이 전송이 된다. 소켓 서버는 기록 웹 서버와 WIPI 클라이언트간의 중계 서버 역할을 한다. 실질적인 realtime 중계를 담당하게 되고 Mobile Gateway를 통해서 통신회사마다 무선인터넷을 제공하고 고객들의 여러 가지 정보를 관리하게 된다. 이 소켓 서버에 접속된 WIPI 클라이언트는 모바일 야구 정보를 실시간으로 제공받기 위한 이용자들의 핸드폰으로 원하는 정보를 서비스 받게 된다.

이에 따라 시스템 모델을 바탕으로 본 논문에서 구현한 야구 중계 시스템의 구성도는 그림 3과 같이 크게 중계 서버와 모바일 클라이언트, 기록 웹 서버로 나누어진다.



(그림 3) 시스템의 구성도

그림 3의 기록 웹 서버는 현재 진행되고 있는 경기 내용들을 중계 서버로 전송하거나 데이터베이스에 실시간으로 기록하게 된다. 그리고 모

바일 클라이언트의 사용자 인터페이스를 통해 야구 중계 서비스를 그림으로 제공받을 것인지 문자로 제공 받을 것인지에 대한 응답을 받게 되면 정보는 중계 서버의 처리모듈로 보내져 데이터베이스에 접근한다. 데이터베이스의 데이터 및 현재 경기의 진행 상태에 대한 결과는 XML 생성 모듈로 전달되고 XML 생성 모듈에서 정보를 XML 문서로 변환해 모바일 클라이언트로 다시 전달한다. 모바일 클라이언트의 XML Parsing 모듈에서 XML 문서를 파싱한 뒤 결과를 사용자 인터페이스를 통해 사용자에게 보여준다.

기록 웹 서버와 모바일 클라이언트에서 중계 서버로 정보를 보낼 때는 HTTP 커넥션을 사용하고, 중계 서버에서 모바일 클라이언트와 기록 웹 서버로 보내는 정보는 복잡하기 때문에 XML은 이런 형태의 데이터 교환에 매우 적합하다. 모바일 클라이언트는 WIPI 환경에서 Jlet[2]으로 구현하고 중계 서버는 J2SE[5]환경에서 자바언어의 소켓[6]을 이용하여 구현하며 기록 웹 서버는 J2EE[7]환경에서 JSP(Java Server Page), Servlet으로 구현한다.

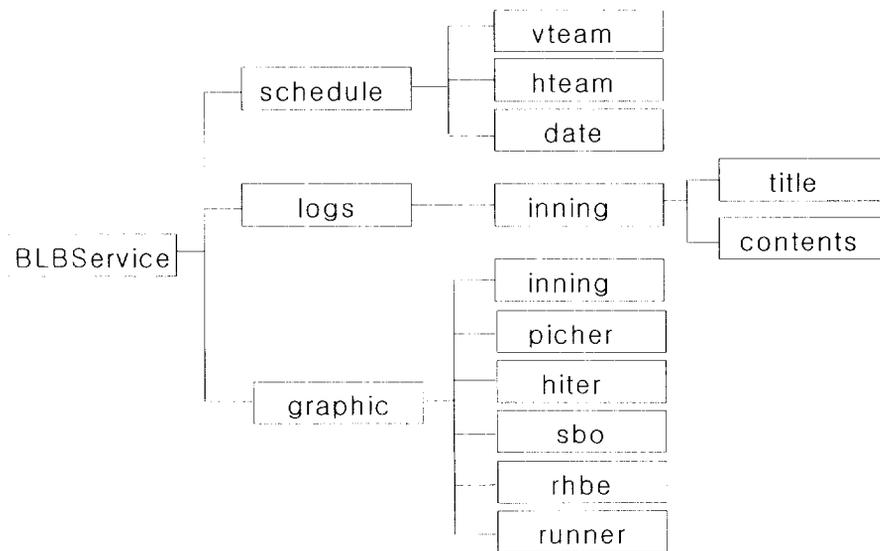
3.2 XML 데이터 교환 모델

데이터 교환 모델로서 XML 역할을 보면, 최근 XML 문서는 여러 분야에서 현재 업종별 비즈니스 표준을 목표로 XML기반 표준화가 진행되고 있으며, 주로 데이터 형이나 비즈니스 규칙들을 XML로써 정의하고 있다. 대표적으로 전자상거래 표준안인 ebXML, 신문업계의 NewsML, 의료업계의 MML 등의 각 비즈니스 업종영역에서뿐만 아니라, 수학적 표현을 위한 MathML, 휴대폰을 위한 WML, 음성 표현을 위한 VoiceXML 등의 다양한 업종 및 분야에서 XML을 기반으로 한 표준들

이 만들어지고 있다.

문서의 교환은 문서의 구조와 관계되는 것으로 XML 문서 정의형을 구성하는 DTD 또는 XML Schema에 의해 표현될 수 있으며 XML 형태의 구조화된 데이터 교환의 시도는 표준화 기구를 통해 이미 검증을 거친 상태이다. DTD는 문서의 형태를 정의할 수 있으나 지원되는 데이터 형식이 한정적이며, 임의로 사용자가 정의한 데이터 형식을 지원하기 어렵다. 따라서 데이터 구조를 DTD를 이용하여 표현할 경우 무리가 따른다. 그러나 DTD에 비해 XML Schema는 다양한 데이터 형, 관계(Relationship), 복합(Composition) 속성, 상속(Inheritance) 등을 지원하므로 유연하고 확장 가능하여 복합 구조의 데이터를 표현하기에 적합하다.

본 논문에서는 야구 중계 서비스에 대한 데이터를 XML 문서로 나타낸다. XML 문서의 정의는 DTD가 아닌 W3C[5]에서 표준으로 규격한 XML Schema[8]를 이용하여 필요한 내용을 기술하고 데이터를 XML 문서로 생성한다.



(그림 4) 야구 중계 서비스를 위한 데이터 교환 모델 구조

그림 4의 데이터 교환모델 구조는 야구 중계 서버에서 클라이언트로 전송되는 데이터가 포함하고 있는 요소를 나타내기 위한 구조이다.

그림 4를 기반으로 엘리먼트를 선언하여 XML 문서에서 사용될 엘리먼트의 규칙과 속성을 정의하여 XML Schema를 설계한다. 그림 4를 토대로 작성된 데이터 교환 모델의 XML Schema는 그림 5와 같다.

```
<?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: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 5) 야구 중계 정보에 대한 XML Schema

위의 작성된 XML Schema는 야구 중계 서버에서 사용자에게 보낼 정보로서 그래픽과 텍스트에 대한 Schema이다.

BLBServer를 루트로 해서 schedule 엘리먼트는 경기일정에 대한 정보를 나타내고, logs 엘리먼트는 이닝별 경기내용에 대한 정보를 나타내며, graphic 엘리먼트는 경기진행 사항을 그래픽으로 보여주기 위한 정보를 나타내고 있다.

데이터 교환 모델 구조와 설계에 따라 중계 서버에서 모바일 클라이언트 측의 요청에 따른 정보들은 XML 문서로 생성되어 전송된다.

그림 6은 모바일 클라이언트 측의 사용자가 현재 진행중인 경기 내용을 요청했을 때 전송되는 XML 데이터에 대한 예를 보여주고 있다. 이것은 2005년 2월 5일 오후 1시 30분에 'LG vs SK'의 경기와, 같은 날 오후 2시에 '롯데 vs 삼성'의 경기 일정을 나타내는 XML 문서이다.

```
<BLBService>
  <schedule sid="X20040205LGSK">
    <vteam>LG</vteam>
    <hteam>SK</hteam>
    <date>2005-02-05 13:30:00</date>
  </schedule>
  <schedule sid="X20040205LTSM">
    <vteam>LT</vteam>
    <hteam>SM</hteam>
    <date>2005-02-05 14:00:00</date>
  </schedule>
</BLBService>
```

(그림 6) 경기 일정에 대한 XML 문서

3.3 중계 서버 설계

중계 서버의 개발 및 사용 환경은 표 2와 같다.

<표 2> 중계 서버 개발 및 사용 환경

종류	구분	소프트웨어 및 사양	비고
중계 서버	개발환경	J2SE	Sun
	개발툴	Eclipse 2.1.2	Eclipse
	DB Server	Mysql 4.1.7	Mysql
	DB 툴	SqlGate	SqlGate

중계 서버는 XML 생성 모듈, 처리 모듈, 객체 관계형 데이터베이스로 구성되어 있다. 모바일 클라이언트로부터 사용자 요청이 전송되면 야구 중계 시스템에 요구되는 데이터를 데이터베이스에서 가져오고 현재 진행되고 있는 경기상황의 결과를 XML 생성 모듈에서 XML 문서로 변환한 뒤 모바일 클라이언트로 전송한다. 데이터베이스는 경기상황의 결과를 데이터로 저장하고 있다.

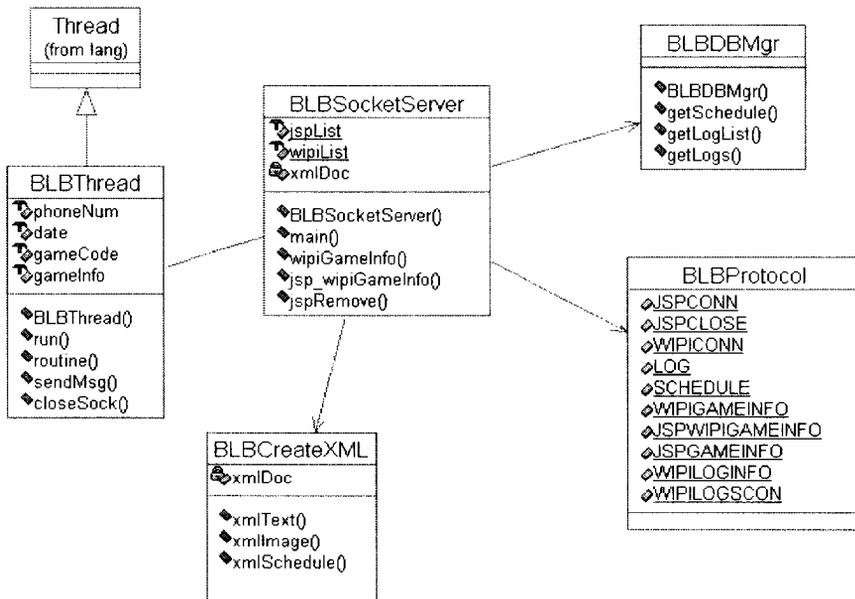
또한 중계 서버는 접속된 모든 모바일 클라이언트에게 실시간으로 전송하기 위해서 진행되는 게임마다 Hashtable을 생성하여 모바일 클라이언트가 요구하는 중계정보를 Broadcast 방식으로 데이터를 전송한다.

중계 정보는 XML 문서로 전송이 되므로 한번 전송된 데이터는 모바일 클라이언트에서 String 이상의 유용한 정보로서 가치를 가지게 된다.

그리고 기록 웹 서버에서 제공되는 애플릿(Applet)도 중계 서버를 통해서 중계 정보를 전송을 받고 있다. 이에 따라 웹 사이트와 무선 인터

넷에서 동일한 콘텐츠를 서비스할 수 있게 된다.

중계 서버의 처리 모듈은 모바일 클라이언트의 사용자 요청을 받아 처리한다. 그리고 기록 웹 서버로부터의 경기 사항에 대한 데이터 정보들을 실시간으로 계속적으로 받아 현재 진행되고 있는 야구 경기 진행 및 결과를 XML 생성 모듈로 전달한다. 그림 7은 중계 서버를 구성하는 클래스들 사이의 관계를 모델링 한 것이다.



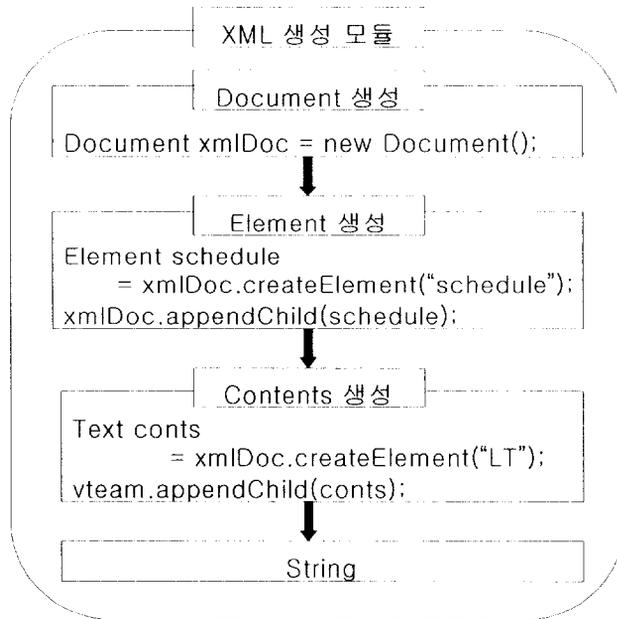
(그림 7) 중계 서버 클래스 다이어그램

BLBSocketServer 클래스는 기록 웹 서버와 모바일 클라이언트를 연결 시켜주는 소켓 서버 클래스를 생성하는 역할을 수행한다. BLBThread 클래스는 소켓 서버에 접속하는 각각의 모바일 클라이언트들을 쓰레드 클래스로 생성하는 역할을 수행한다. BLBDBMgr 클래스는

소켓 서버에 접속한 모바일 클라이언트가 데이터베이스 서버에 필요로 하는 여러 가지 데이터를 SELECT, INSERT, UPDATE 하는 역할을 수행하는 클래스이다. BLBProtocol 클래스는 기록 웹 서버와 모바일 클라이언트가 중계 서버 사이에서 데이터 request, response를 위한 Protocol을 정의 해 놓은 컨트롤 클래스이다. BLBCreateXML 클래스는 모바일 클라이언트와 기록 웹 서버에 보낼 정보를 XML 문서로 변환한다.

XML 생성 모듈에서는 클라이언트에게 보낼 정보를 XML 문서로 변환한다. 사용자에게 보낼 정보는 그래픽과 텍스트 두 가지로 나뉜다. 그래픽은 사용자가 야구 중계를 그래픽으로 보기위한 정보를 포함하고 텍스트는 야구 중계를 문자로 보기위한 정보를 포함한다.

이에 따라 새로운 도큐먼트를 생성하여 생성된 도큐먼트의 자식노드로 루트 엘리먼트를 추가하고 텍스트 필드에서 가져온 인자들을 루트 엘리먼트의 자식노드로 추가하여 트리 구조의 XML 문서를 생성한다. 그림 8과 같이 XML 생성 모듈에서는 그림 5에서의 Schema에 따라 노드를 생성하고, 각 노드들의 노드 타입을 판단하여 도큐먼트, 루트 엘리먼트, 엘리먼트, 속성, 텍스트로 구분하여 상호 관계를 정의하여 트리구조를 생성한다. 이렇게 생성된 XML 문서는 스트링 형태로 클라이언트 시스템의 XML 파싱 모듈로 전송된다.



(그림 8) XML 문서의 생성

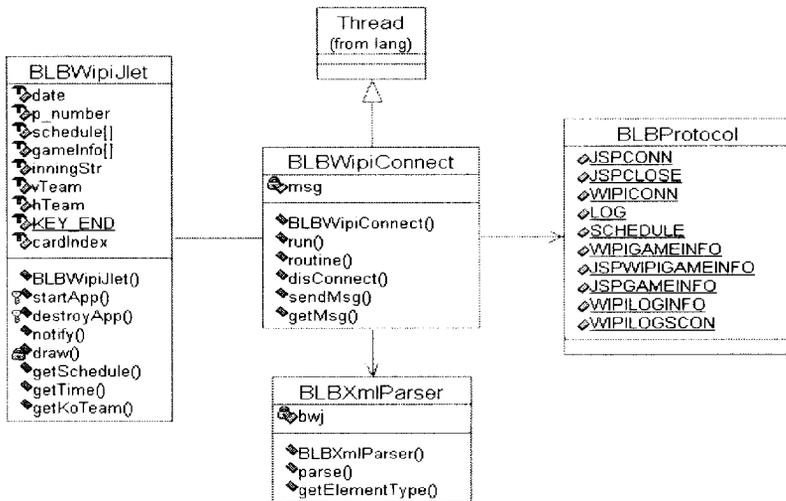
3.4 모바일 클라이언트 설계

모바일 클라이언트의 개발 및 사용 환경은 표 3과 같다.

<표 3> 모바일 클라이언트의 개발 및 사용 환경

종류	구분	소프트웨어 및 사양	비고
모바일 클라이언트	개발환경	WIPI	
	운영체제	Windows 2000 Server	Microsoft
	개발툴	Eclipse 2.1.2	Eclipse
	인터페이스	AromaWIPI-Emulator1.1.1.7	Aroma Software
	그래픽	PhotoShop 7.0	Adobe
	소리편집툴	GoldWave	
	메모리	512kb~1024kb	

모바일 클라이언트는 크게 XML 파싱 모듈과 사용자 인터페이스로 구성되어 있다. 그림 9는 본 논문에서 제안하는 모바일 클라이언트 클래스 다이어그램이다.



(그림 9) 모바일 클라이언트 클래스 다이어그램

BLBWipiJlet 프로그램은 응용 프로그램에 포함된 아이템 항목을 나열하고, 선택할 수 있도록 한다. 사용자가 해당 아이템 항목을 선택하면 HTTP 컨넥터를 통하여 해당하는 XML 문서를 읽어 들인 후 XML 문서를 시작 태그와 끝 태그를 기준으로 파싱하여 문자 서비스에 필요한 타이틀과 내용, 그리고 그래픽 서비스에 필요한 데이터들을 추출하여 사용자 인터페이스를 통해 보여 질 수 있게 설계하였다.

모바일 클라이언트 측의 주요 클래스의 내용을 살펴보면, BLBWipiJlet 클래스는 모바일 클라이언트가 최초 실행시 로딩되는 클래스이며, 생성자에는 중계 서버와 연결을 가능하게 해주는 BLBWipiConnect 객체가 생성되며, 모바일 클라이언트 인터페이스를 제어하고 관리하는 역할을 하는 클래스이다. BLBWipiConnect 클래스는 중계 서버와 모바일 클라이언트가 통신을 하기 위해서 Connection을 생성하고, 이렇게 설정된 Connection은 DataInputStream, DataOutputStream을 추출하여 통신을 원활하게 할 수 있도록 하는 클래스이다. BLBProtocol 클래스는 모바일 클라이언트와 중계 서버 사이에서 데이터 request, response를 위한 Protocol를 정의 해 놓은 컨트롤 클래스이다. BLBXmlParser 클래스는 XML 문서를 읽어들인 후 태그를 기준으로 파싱하여 내용을 추출한 뒤 그 결과값을 넘겨준다.

XML 파싱 모듈에서는 중계 서버로부터 받은 XML 문서를 파싱한다. 파싱을 통해 그래픽 서비스에 한 정보와 문자 서비스에 대한 정보를 얻게 되는데 이 정보를 각각의 Card 객체[2]에 적재되어 Display 객체[2]에 등록한다. Display는 휴대폰 단말기의 디스플레이에 나타날 수 있는 Jlet 내의 객체로써 여러 개의 생성이 가능하다.

모바일 클라이언트의 요청에 따른 정보를 추출하기 위해서, 중계 서버의 XML 생성 모듈에서 스트링 형태로 모바일 클라이언트 시스템에 전

송되는 XML 문서의 모든 노드를 질의하여, 받아들인 인자와 같은 노드의 값을 반환하거나 치환한 후 원하는 정보를 얻을 수 있다. 이와 같은 기능은 XML 질의 모듈에서 이루어지며, 노드의 질의는 루트 엘리먼트의 자식 노드 리스트를 받아들여 이 노드 리스트의 길이만큼 노드를 검색함으로써 수행된다.

그림 10은 XML 문서의 질의 모듈에 대한 내용이다.

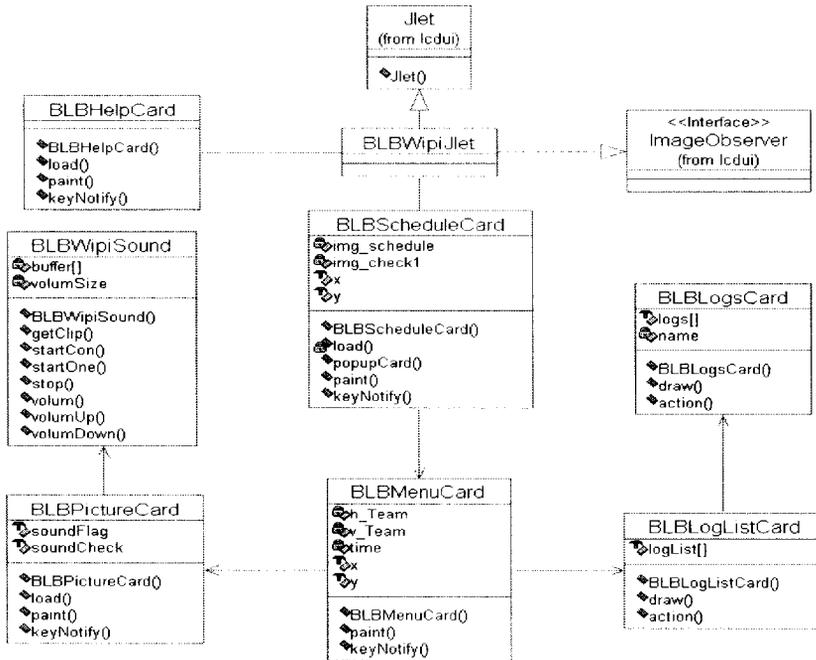
```

NodeList scheduleList = xmlDoc.getElementsByTagName("schedule");
for(int i=0 ; i<scheduleList.Length ; i++){
    String vteam
        = scheduleList.item(i).getFirstChild().getFirstChild().getNodeValue();
    String hteam
        = scheduleList.item(i).getFirstChild().getNextSibling().getFirstChild().getNodeValue();
    String date
        = scheduleList.item(i).getLastChild().getFirstChild().getNodeValue();
    String sid
        = scheduleList.item(i).getNamedItem("sid").getNodeValue();
}

```

(그림 10) XML 문서의 질의 모듈

사용자 인터페이스는 WIPI에서 제공하는 High Level UI(User Interface)와 Low Level UI(User Interface)[2]를 이용하여 휴대폰 사용자를 위한 인터페이스를 설계하였으며, 그 내용은 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 것인지 선택하면 실시간으로 업데이트된 Display를 보여준다. 그림 11은 사용자 인터페이스 모듈에 대한 클래스들 사이의 관계를 모델링 한 것이다.



(그림 11) 사용자 인터페이스 모듈 클래스 다이어그램

표 4는 각 클래스들의 기능들을 보여준다. BLBWipiJlet 클래스에서 XML 문서를 분석한 뒤 원하는 정보들을 각각 해당되는 Card 클래스들을 통해서 클라이언트 화면에 보여지게 된다.

<표 4> 사용자 인터페이스를 구성하는 주요 클래스

클래스	기능
BLBWipJlet	모바일 클라이언트가 최초 실행시 로딩되는 클래스이며, 모바일 클라이언트 인터페이스를 제어하고 관리하며 마지막으로 종료를 하기 위한 역할을 하는 클래스이다.
BLBHelpCard	Card 클래스를 상속받은 클래스이며, 모바일 클라이언트의 사용자가 쉽게 사용하기 위해서 사용 설명서를 보여주기 위한 역할을 하는 클래스이다.
BLBLogListCard	Card 클래스를 상속받은 클래스이며, 모바일 클라이언트에서 문자 서비스를 받기 위해서 이닝리스트를 중계 서버에서 전송을 받아서 보여주기 위한 역할을 하는 클래스이다.
BLBLogsCard	Card 클래스를 상속받은 클래스이며, 모바일 클라이언트에서 문자 서비스를 받기 위해서 아닝마다 자세한 내용을 중계 서버에서 전송을 받아서 보여주기 위한 역할을 하는 클래스이다.
BLBMenuCard	Card 클래스를 상속받은 클래스이며, 모바일 클라이언트에서 선택한 게임에 대한 정보와 그림으로 또는 문자로 서비스 받을 건지 여부를 선택하는 내용을 보여주기 위한 역할을 하는 클래스이다.
BLBWipiSound	WIPI API에서 제공하는 Clip, Player 클래스를 이용해서 모바일 클라이언트에서 사운드를 play하고 stop 등 여러 가지 사운드 제어하기 역할을 하는 클래스이다.
BLBScheduleCard	Card 클래스를 상속받은 클래스이며, 모바일 클라이언트에서 오늘의 게임에 대한 목록을 중계 서버에서 전송을 받아서 보여주기 위한 역할을 하는 클래스이다.
BLBPictureCard	Card 클래스를 상속받은 클래스이며, 모바일 클라이언트에서 그림으로 실시간으로 진행되는 야구게임을 중계 서버에서 Broadcast 하면 자동으로 게임에 대한 정보가 변경될 수 있는 역할을 하기 위한 클래스이다.

3.5 기록 웹 서버 설계

기록 웹 서버의 개발 및 사용 환경은 표 5와 같다.

<표 5> 기록 웹 서버의 개발 및 사용 환경

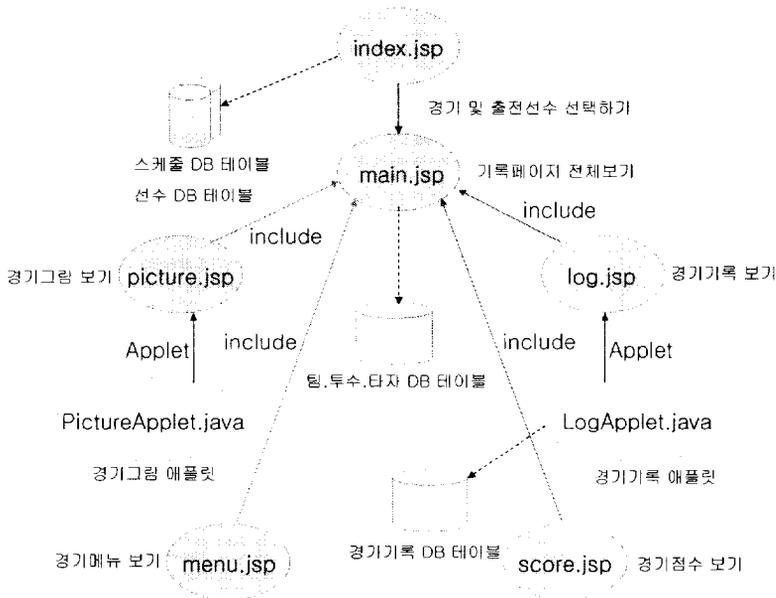
종류	구분	소프트웨어 및 사양	비고
기록 웹 서버	개발환경	J2EE	Sun
	운영체제	Windows 2000 Server	Microsoft
	서버	Tomcat 4.1	Apache Software
	개발툴	JBuilder	Borland Korea

기록 웹 서버는 처리 모듈, XML 파싱 모듈, 사용자 인터페이스, 객체 관계형 데이터베이스로 구성되어 있다. 기록 웹 서버는 야구 중계 정보를 실시간으로 입력하고 관리를 하기 위해서 웹 서버에 기록을 하는 역할을 한다. 현재 경기 진행 사항에 대한 정보가 웹 서버에 기록되면 이 정보들은 데이터베이스에 저장되고 실시간으로 모바일 클라이언트에게 전송을 하기 위해 중계 서버로 전송이 된다. 그리고 중계 서버의 XML 생성 모듈에 의해 XML 데이터로 변환한 뒤 모바일 클라이언트 뿐만 아니라 기록 웹 서버에도 중계 정보가 전송이 되어 사용자 인터페이스를 통해 웹 화면에 보여지게 된다.

기록 웹 서버의 XML 파싱 모듈은 모바일 클라이언트의 XML 파싱 모듈과 같은 맥락이므로 생략한다.

모바일 클라이언트의 사용자 인터페이스는 WIPI에서 제공하는 User Interface를 이용하여 휴대폰 사용자를 위한 인터페이스를 설계한 반면 기록 웹 서버의 사용자 인터페이스는 자바 Applet을 이용하여 설계하였다. 역시 비슷한 맥락이므로 생략한다.

기록 웹 서버의 처리 모듈은 경기장에서 기록원이 경기 일정과 경기의 모든 기록, 팀, 선수의 모든 내용들을 삽입 또는 수정하는 과정을 지원한다. 기록 웹 서버가 지원하는 각 JSP 구현 페이지와 Applet들의 상태 천이도는 그림 12와 같다.



(그림 12) 기록 웹 서버의 JSP 상태 천이도

JSP 구현 페이지들은 서버에서 실행되는 코드로서 기록 웹 서버에 HTML 소스를 출력하는 기능을 수행한다. 'index.jsp'는 웹 클라이언트에서 경기 및 출전선수 선택하기 메뉴를 가진 HTML 코드를 출력한다. 'main.jsp'는 기록페이지 전체보기를 가진 HTML 코드를 출력하며, 'picture.jsp', 'log.jsp', 'menu.jsp', 'score.jsp' 페이지를 include하고 있다. 'picture.jsp'는 경기의 모든 상황을 그림으로 표현하기 위한 페이지이다. 이런 그림을 표현하기 위해서 'PictureApplet.java' 애플릿을 <plugin>태

그를 사용해서 페이지에 포함하여 실행내용을 출력한다. 'log.jsp'는 경기의 모든 진행 사항을 경기기록 DB 테이블에 접근을 해서 경기 로그내용을 표현하기 위한 페이지이다. 이런 경기 로그내용을 표현하기 위해서 'LogApplet.java' 애플릿을 <plugin>태그를 사용해서 페이지에 포함하여 실행 내용을 출력한다. 'menu.jsp'는 경기 중에 일어날 수 있는 메뉴를 세팅하기 표현한 페이지이다. 경기 중에 일어날 수 있는 메뉴는 볼카운트, 아웃 카운트, 주자현황, 현재 투수, 현재 타자, 이닝별 스코어, 전체점수, 안타수, 포볼, 에러 카운트 등이 있다. 'score.jsp'는 경기 중에 일어났던 모든 기록과 현재 출전선수의 명단 및 기록을 표현하기 위한 페이지이다. 'menu.jsp'에서 기록했던 모든 기록들은 이 페이지에서 표현이 된다. 'main.jsp'에서 일어나는 모든 경기 진행 그림과 텍스트들은 중계 서버로 전송이 되며 전송된 모든 내용들은 실시간으로 중계 서버에 접속된 모바일 클라이언트에게 각각 선택된 경기스케줄에 따라 전송이 된다.

3.6 구현 결과

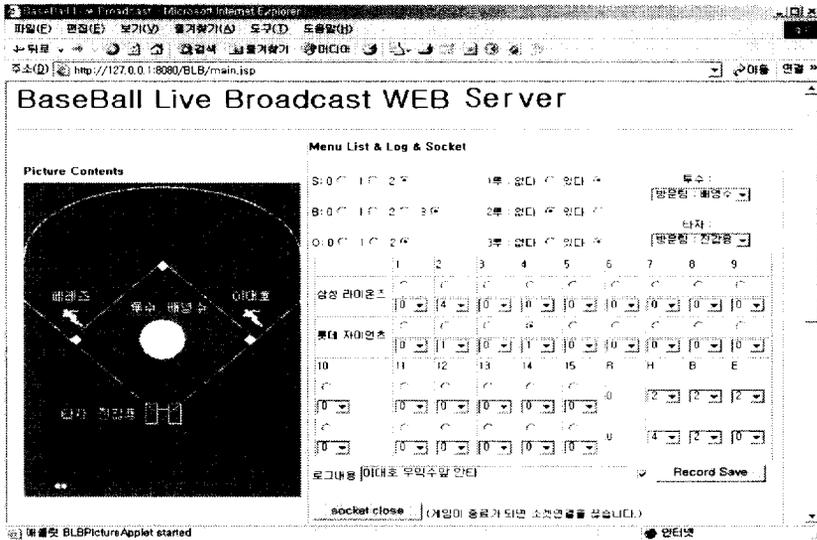
야구 중계 시스템의 서버 프로그램을 구현하기 위해서 자바로 구현한 소켓을 사용하였다. 중계 서버의 실행 결과 화면은 그림 13과 같다. 그림 13에서 중계 서버는 초기 서버 소켓(Server Socket)을 생성하고 바인드(Bind) 한다. 리스닝(Listening)상태에서 단말기와 기록 웹 서버의 접속 요청을 기다리다 접속이 이루어지면 해당 단말기의 접속 IP와 포트번호를 보여주고, 모바일 클라이언트에서 데이터 요청 시 해당하는 정보들을 XML 문서 형태로 모바일 클라이언트에 전송하게 된다.

기록 웹 서버에 새로운 정보가 기록이 되거나 변경이 되면 중계 서버에서 변경된 데이터에 대해 XML 문서로 변환 후 모바일 클라이언트와 기록 웹 서버에 전송하게 된다.

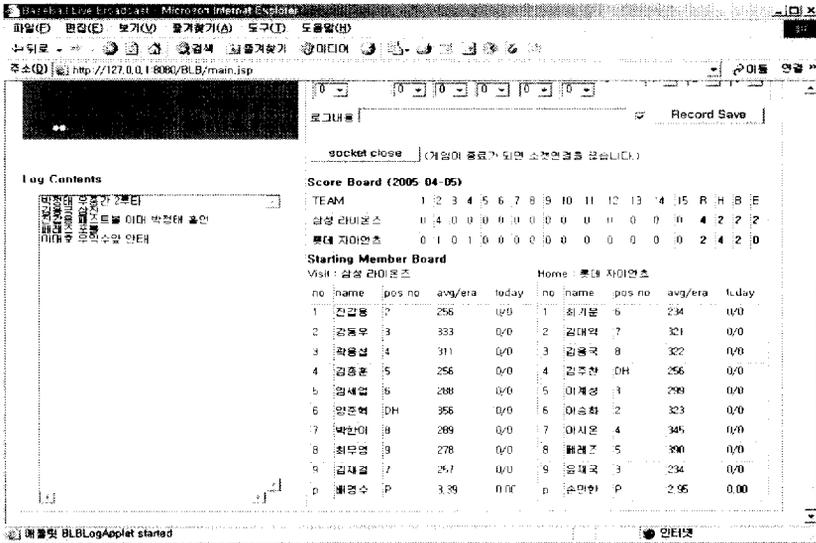


(그림 13) 중계 서버 실행 화면

그림 14는 기록 웹 서버의 실행 결과 화면이다. 기록 웹 서버에서는 현재 경기 진행 사항에 대한 새로운 정보 또는 변경되는 정보들을 계속 업데이트하게 된다. 업데이트하는 즉시 중계 서버에게 실시간으로 알려줘 중계 서버가 야구 중계 서비스를 실시간으로 중계할 수 있도록 한다. 기록 웹 서버에서도 (a)그래픽 서비스 화면에서 애플릿을 통해 현재 진행되는 경기 사항을 그래픽으로 확인할 수 있으며, (b) 문자 서비스 화면에서는 현재 진행되는 경기 사항을 문자로 확인할 수 있다.



(a) 그래픽 서비스 화면

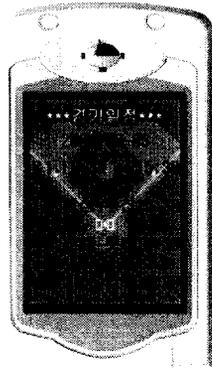


(b) 문자 서비스 화면

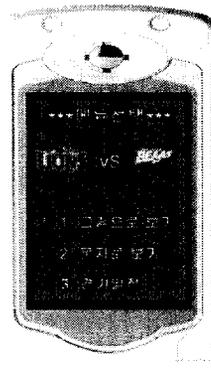
(그림 14) 실시간 기록 웹 서버 실행 화면

모바일 야구 중계 서비스를 이용하는 사용자 인터페이스는 아로마소프트사에서 제공하는 휴대폰 에뮬레이터 AromaWIPI-Emulator를 이용하여 구현하고 작업을 마친 뒤 실제 단말기에서 동작하도록 하였다. 그림 15는 WIPI 기반의 야구 중계 시스템이 모바일 클라이언트에서 실행되는 화면이다. 그림 15에서 (a)경기일정화면은 중계 서버에서 전송된 야구일정을 보여주는 화면으로 1번에서 4번 사이에 모바일 클라이언트가 원하는 게임을 선택하면 (b)선택화면으로 전환된다. (b)선택화면은 사용자가 야구 중계 서비스를 받기 위한 메뉴로서, 사용자가 메뉴를 선택하면 중계 서버에서는 야구 진행 상황의 결과를 XML 문서로 변경해 모바일 클라이언트로 보낸다. 모바일 클라이언트에서는 XML 문서를 파싱하고 정보를 Display객체에 적재해 화면에 보인다. (c)그래픽화면은 그래픽으로 경

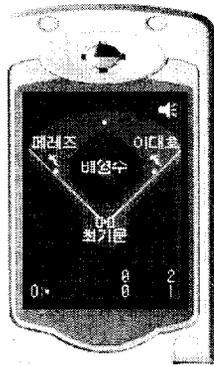
기 진행을 보기위한 화면이며, (d)텍스트화면은 문자로 서비스를 받기위한 화면이다. (c), (d)화면은 중계 서버에서 데이터를 전송할 때마다 실시간으로 자동 업데이트 되어 경기 진행 및 결과를 볼 수 있다.



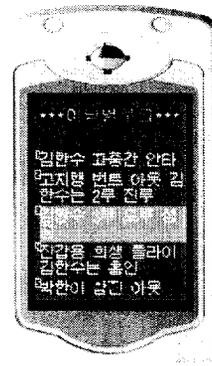
(a)경기일정화면



(b)선택화면



(c)그래픽화면



(d)텍스트화면

(그림 15) 모바일 클라이언트의 실행 화면

IV. 결론 및 향후 연구 과제

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 제공할 수 있는 장점을 가지고 있다. 또한, 모바일 환경에서 XML 데이터 서비스는 데이터의 공용성과 프로그램의 유연성을 향상시키는데 필요한 비용과 노력을 절감할 수 있으며, HTTP를 기반으로 하여 분산 환경에서 시스템 간의 정보를 효율적으로 교환할 수 있는 시스템을 설계 및 구현하였다.

현재 모바일의 발전 속도는 어느 누구도 가늠할 수 없을 정도로 빠르게 변화하고 있다. 이러한 변화 속에서 다양한 욕구를 가지고 있는 클라이언트와 많은 종류의 device의 모든 요건을 충족시키기는 결코 쉬운 일은 아니다. 그러나 본 논문에서 제안하는 시스템에서 데이터 전송은 표준화된 데이터 포맷인 XML을 기반으로 설계하여 TCP/IP 및 HTTP를 지원하는 모든 플랫폼에서 데이터 접근이 가능하며, 별도의 변환과정 없이 데이터를 사용할 수 있다.

향후 연구로는 좀 더 동적인 그래픽과 사용자 편의성을 위한 클라이언트상의 사용자 인터페이스 전환이 필요하다. 그리고 자원의 규모가 작은 모바일 단말기에 효율적으로 내장 가능한 XML 파서를 설계 구현하는 것을 들 수 있다.

참 고 문 헌

- [1] C.Kirkegaard, A.Moller, M.I.Schwartzbach, "Static analysis of XML transformations in Java", IEEE Trans on SE, Vol.30, No.03, pp.181-192, 2004.
- [2] Artem Gamash, "A Geographic XML-based Format for the Mobile Environment", Processing of the 34th Hawaii international Conference on System Science, 2001.
- [3] Kawaguchi, K., "W3C XML Schema Made Simple", XML.com, 2000
- [4] Harvey M., etc., "How to Program XML", Prentice Hall, 2002.
- [5] Harold, Eliotte Rusty, "Java Network Programming", Oreilly, 1999.
- [6] James Hart, "Java J2SE 1.4 Core Platform Update", WROX Press, 2002.
- [7] Wrox Author Team, "Beginning J2EE 1.4", WROX Press, 2003.
- [8] XML Schema Part 0 : Primer
- [9] 송관호, 2004 Survey on the Wireless Internet Use, 한국인터넷진흥원, 2004.
- [10] 이우진 외, "소프트웨어 콘텐츠의 모바일 적합성 분석을 위한 웹 기반 모델" 정보과학회논문지, 제31권 제10호, pp.1276-1285, 2004.
- [11] 이상윤 외, "한국 무선 인터넷 표준 플랫폼(WIPI)의 표준화 현황 및 발전 전망", 한국정보과학회논문지, 제22권 제1호, pp.16-23, 2004.
- [12] 권은정 외, "XML을 기반으로 한 관계형 데이터베이스 메타데이터 리파지토리 설계 및 구현", 정보과학회논문지, 제9-D권 제1호, pp.1-10, 2002.

- [13] 이상윤 외, "모바일 환경에서 vCard 서비스를 위한 웹 에이전트의 설계 및 구현", 한국정보과학회논문지, 제9-D권 제3호, pp.477-486, 2002.
- [14] 강미연 외, "WIPI 기반의 모바일 단말을 위한 내장형 XML 파서 및 뷰어", 정보과학회 춘계학술대회, 제31권 제1호, pp.865-867, 2004.
- [15] 배석희, "모바일 플랫폼 표준화 동향 및 향후 발전방향", TTA 저널, 제 82호. 2002.
- [16] KWISF, "모바일 표준 플랫폼 WIPI V2.0.1", 한국무선인터넷 표준화 포럼, 2004.
- [17] 박수원 외, "위파 모바일 프로그래밍", 한빛미디어, 2003.
- [18] MobileJAVA, <http://www.mobilejava.co.kr>.

감사의 글

늦은 나이에 대학원을 시작하면서 설레임과 두려움, 또 열심히 하겠다라는 각오도 있었습니다. 이러한 각오 앞에서도 수없이 많은 어려움이 있었지만 끊임없는 격려와 힘을 주신 분들이 많이 계셨기에 소중한 경험을 갖고 좋은 결실을 맺게 되었습니다. 도와주신 많은 분들께 감사의 인사를 드립니다.

먼저, 지도교수님이신 윤성대 교수님께 감사드립니다. 세미나와 수업과정을 통해 교수님으로부터 참된 연구자의 마음가짐과 삶에 대한 성실한 태도를 지닐 수 있었습니다. 앞으로 더욱더 성실히 생활해 나가겠습니다. 그리고 바쁘신 와중에도 학위논문 심사를 위해 많은 조언과 관심을 가져 주신 이경현 교수님, 제 잦은 실수에도 아낌없는 도움과 가르침을 주신 여정모 교수님께 감사드립니다.

연구실에 많은 분들께 감사드립니다. 오랜 시간동안 함께 하면서 웃고 떠들고, 정보도 교환하며 생활했던 많이 일들이 저에게 좋은 추억으로 행복한 시간으로 오래 간직될 것입니다. 논문을 준비하면서 부족한 점들을 세심히 봐주시고 가르침과 도움을 주신 박사님들, 박현호 선생님, 황순환 선생님, 고석범 선생님, 윤종찬 선생님, 박상일 선생님 정말 감사드립니다. 같이 논문을 준비하면서 즐겁기도 또 힘들기도 했지만 서로에게 힘과 격려가 되어준 천소영 선생님, 유은아 선생님, 정귀녀 선생님, 서정애 선생님 감사드립니다. 혼자, 동기들이 아닌 후배님들과 함께 수업하면서 많은 도움과 애정을 주신 권은경 선생님, 김종필 선생님 그리고 다른 2년차 선생님들 모두에게 감사드립니다. 남을 항상 배려하는 홍진숙 선

생님, 한번 얘기하며 끝이 없고 너무 재미있는 이미나 선생님, 서영선 선생님, 집에 가는 길이 너무 즐거운 벗이 되어준 김민국 선생님, 따뜻한 말 한마디가 힘이 되어준 임선자 선생님, 연구실을 위해 수고가 많은 학부생들 그리고 미처 언급하지 못한 분들 모두들 감사드립니다.

대학원을 무사히 마칠 수 있도록 육아 및 가사일을 도와주신 친정 부모님, 시부모님께 감사드립니다. 부모님의 헌신적인 사랑에 항상 감사하며 최선을 다해 열심히 살아가겠습니다.

마지막으로, 대학원 생활에 있어 적극적인 지원과 격려와 사랑을 아끼지 않은 저에게 큰 힘이 되어주는 든든한 배우자 정동진씨, 많이 돌봐주지 못했지만 너무나도 예쁘고 사랑스럽게 잘 자라준 딸 서윤이에게 정말 고맙다는 말과 사랑한다는 말을 전합니다.

2005년 6월 23일

이 경 미 올림