

유무선 통합 웹 서비스를 위한 서버 시스템의 설계 및 구현

배성환*, 이종언**, 차시호***, 김규호****

Design and Implementation of a Server System for Integrating Wired and Wireless Web Services

Sung-Hwan Bae *, Jong-Eon Lee **, Si-Ho Cha ***, Kyu-Ho Kim ****

요약

인터넷의 등장으로 많은 사람들의 다양한 정보를 함께 공유할 수 있도록 되었으며, 사용자들의 요구가 점점 증가되어 어디서든지 인터넷에 접속하여 필요한 정보를 수집하길 원하고 있다. 현재 웹 서비스는 XML에 기반을 둔 플랫폼과 구현언어에 독립적인 컴포넌트 기반의 분산 컴퓨팅 기술로 인터넷 상 분산 애플리케이션 통합 방법의 대표주자로 자리를 잡아가고 있다. 또한, 웹 사이트에서 유지, 관리해야 할 콘텐츠의 양은 방대해지고 있고, 콘텐츠의 갱신 주기도 점점 짧아지고 있으며, 웹 사이트의 콘텐츠를 보다 효율적이고 편리하게 생성, 유지, 관리하기 위한 시스템 개발이 요구된다. 이에 본 논문은 XML로 작성된 컴포넌트와 JWSDP, 무선인터넷 Server Side Script, WML editor를 활용하여 유·무선을 통합한 웹 서비스 기능을 제공하는 유무선 통합 서버 시스템을 설계하고 구현한다.

Abstract

Since internet and XML has come out, Web Services are regarded as a new paradigm of next generation's IT are becoming a leading technology to integrate the distributed applications on internet using XML based Platforms and distributed computing based on component technologies independent on implementing languages. Nowadays internet is generally used and the use of wireless internet is rapidly rising. The quantities of contents on the web-sites to manage and maintain are getting larger according to these trends and the update cycles are getting shorter. It is necessary to solve these problems that developing the systems which can effectively generate, maintain and manage the contents on the web-sites. In the paper, a system which can integrates wired and wireless services is designed and implemented using the technologies like XML based components, JWSDP, Wireless Server Side Script, WML editor and etc..

▶ Keyword : Web Services, XML, SOAP, WSDL, UDDI

* (주)아이넷씨엔씨 연구소 주임연구원

** 광운대학교 대학원 박사과정

*** (주)웨어플러스 네트워크사업본부 기술개발팀 팀장

**** 서울보건대학 인터넷정보과 교수, 한국컴퓨터정보학회 이사

I. 서론

지금까지 통신 서비스는 유선과 무선으로 분리되어 각각 고유한 영역에서 서비스가 제공되어 왔다. 그러나 최근에는 인터넷과 멀티미디어의 수요가 증가하였고, 고객이 언제, 어디서나, 원하는 서비스를 끊임없이 제공 받기를 원하고 있다. 이러한 환경의 변화로 인해 유·무선 통합망의 필요성이 대두되고 있고, 유선망의 광대역성과 무선망의 이동성이 결합된 유·무선 통합 서비스들이 등장하고 있다. 무선망의 낮은 전송속도, 잦은 접속 단절, 높은 비트 에러율의 문제와 이동 단말기의 작은 화면 크기, 낮은 처리 능력, 단순한 사용자 인터페이스, 다양한 하드웨어 및 소프트웨어로 인하여 기존 유선컨텐츠의 재사용이 어려워졌다.

현재 웹 서비스는 XML 기반의 플랫폼과 구현언어에 독립적인 컴포넌트 기반의 분산 컴퓨팅 기술로 인터넷 상 분산 애플리케이션 통합 방법의 대표주자로 자리를 잡아가고 있다. 따라서, 웹 사이트에서 유지, 관리해야 할 컨텐츠의 양은 방대해지고 있고, 컨텐츠의 갱신 주기도 점점 짧아지고 있으며, 웹 사이트의 컨텐츠를 보다 효율적이고 편리하게 생성, 유지, 관리하기 위한 시스템 개발이 요구된다.

XML은 응용프로그램간의 정보 교환을 제공하는 수단으로 이제 자리를 굳혀가고 있다. 플랫폼과 응용프로그램 구현 언어에 독립으로 사용될 수 있다는 큰 장점 때문에 운영 시스템, 네트워킹 프로토콜, 프로그래밍 언어, 데이터베이스, 클라이언트 응용프로그램 및 서버 응용프로그램에서 XML은 기본 기술로 자리 매김하고 있다. 생성된 모든 웹 서비스 프로토콜과 기술의 데이터 표현 계층에 XML을 이용함으로써 핵심 계층에서 이들 기술을 상호 운영할 수 있다. 데이터 전송처럼, XML은 모든 네트워킹, 운영 시스템, 혹은 프로토콜을 소유한 플랫폼 바인딩을 배제한다[1,2].

HTTP 프로토콜과 XML 기술을 기반으로 한 웹 서비스 기술은 SOAP(Simple Object Access Protocol)과 WSDL(Web Service Description Language), UDDI(Universal Discovery Description Integration) 표준이 있으며, 지금 이 시간에도 여러 관련 기술들이 개발 중에 있고, 다양한 웹 서비스 지원 툴들이 발표되고 있다. 웹 서비스는 애플리케이션 통합과 동일한 개념이라고 생각

할 수 있고 시스템을 구축하기 위한 기술이 아니라 형태 내의 기능들을 연결하고 통합하는 기술이다. 웹 서비스는 기존의 폐쇄 환경의 통합 기술로도 사용될 수 있다. DCOM으로 개발된 환경과 J2EE 환경, 그리고 CORBA로 구축된 환경을 하나로 통합시킬 수 있다. 이것이 가능한 이유는 개방형 프로토콜인 HTTP를 이용해서 XML 메시지를 전달하여 통신하기 때문이다[1][2][6].

본 논문은 XML로 작성된 컴포넌트와 JWSDP, 무선인터넷 Server Side Script, WML editor를 활용하여 유·무선을 통합한 웹 서비스를 제공하는 유·무선 통합 서버 시스템을 설계하고 구현한다.

II. 관련 연구

1. Web Services

웹 서비스는 기존의 분산된 서비스들을 HTTP와 XML을 이용하여 유연하게 묶을 수 있는 기술로 다음 세대의 웹 환경을 제시해 주고 있다. 웹 서비스 기술은 통신 프로토콜로 HTTP 프로토콜을 사용하고 데이터 기술 방법으로는 현재 범용 데이터 표현 언어의 표준으로 자리 잡고 있는 XML을 사용한다. 따라서 웹 서비스는 HTTP와 XML의 장점을 그대로 유지하여 시스템 독립성과 개발 언어 독립성을 보장하고 기존의 HTTP를 이용한 웹 환경을 사용하여 서비스를 제공한다.

웹 서비스는 현재 세 가지 표준으로 구성되어 있다. XML 통신 프로토콜인 SOAP, 웹 서비스 기술 언어인 WSDL, 그리고 웹 서비스 정보 공유 저장소 표준인 UDDI가 그것이다. 물론, 이 세 가지 기술 모두 XML과 HTTP를 기초로 하는 응용 기술들이다[3].

웹 서비스의 특징을 보면, 웹 서비스 시스템과 클라이언트는 지속적인 연결을 가지지 않으며, 상태 정보를 유지하지 않고, 사용하는 프로토콜은 XML 기반의 SOAP이며, 기존의 다른 프로토콜로 구현된 분산 컴포넌트 환경을 통합시킬 수 있으며, 인터넷상에서의 분산 컴퓨팅이 가능하다[6]. 또한, 웹 서비스가 해야 할 역할은 각 역할별 행위에 대한 규약으로 웹 서비스를 구성하는 세 가지 역할인 웹 서비스 제공자와 웹 서비스 중개자, 웹 서비스 소비자는 공개

(publish), 검색(find), 바인드(bind)라는 세 가지 행위 (Operation)를 하게 된다.

이들 행위들은 세 가지 역할의 환경과는 무관하게 플랫폼 독립적으로 통신하는 것을 원칙으로 하고 있기 때문에 통신 규약과 같은 약속된 방법으로 진행되어야 한다. 세 가지 역할이 플랫폼 독립적으로 통신할 수 있도록 하기 위해 역할별 행위에 대한 규약이 있다.

■ 공개 및 검색(UDDI)

UDDI는 웹 서비스의 공개와 검색을 위한 XML 레지스트리(XML 저장소)의 구현과 사용 방법을 표준화한 것이다. UDDI 스펙에는 웹 서비스 중개자가 운영해야 할 XML 레지스트리 기능과, 웹 서비스 제공자가 어떻게 자사의 웹 서비스를 공개해야 되는지, 그리고 웹 서비스 소비자가 어떻게 검색할 수 있는지에 대해서 언급하고 있다. UDDI 레지스트리는 각종 정보들을 생성, 저장, 검색할 수 있는 XML 기반의 자료 저장 장치를 말한다.

UDDI 레지스트리의 클라이언트가 UDDI 레지스트리에 접근해서 정보를 저장하고, 찾기 위해서는 SOAP 메시지를 이용한다. 또한 SOAP 메시지의 전송용 프로토콜 인터넷 전송 프로토콜인 HTTP 프로토콜을 사용함으로써 클라이언트의 플랫폼과 구현 언어에 독립적으로 UDDI 레지스트리를 사용할 수 있다. UDDI 레지스트리의 클라이언트는 UDDI 레지스트리 프로바이더를 통해서 SOAP 메시지를 생성하고, 전송한다[2][6].

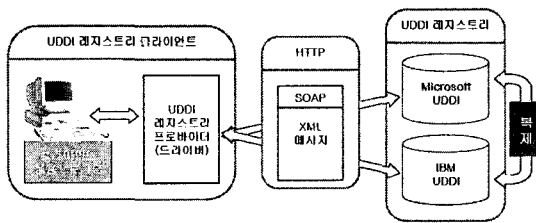


그림 1. UDDI 레지스트리 Fig. 1 UDDI registry

■ 웹 서비스 기술 언어(WSDL)

WSDL은 표준화된 방식으로 웹 서비스의 인터페이스를 기술하는 XML 기술이다. WSDL은 웹 서비스 시스템이 제공하는 원격 프로시저를 이용하기 위해 매개변수, 프로시저명, 그리고 반환형을 기술하는 방법을 표준화한 것이다. 웹 서비스 클라이언트는 WSDL 문서를 이용해서 웹 서비스 시스템에 바인딩 하는 방법을 알게 된다.

■ 단순 객체 접근 프로토콜(SOAP)

SOAP은 원격 응용프로그램 간의 정보를 교환하는데 필요한 구조를 표준화한 것이다. SOAP의 구조는 XML 문서 형태를 가지고 있고, 전송 프로토콜로서 운반되기 위한 방법에 대해서 언급하고 있다. SOAP를 이용하면 .NET 클라이언트에서 J2EE 환경의 웹 서비스 시스템을 이용해 EJB 컴포넌트의 메소드를 호출할 수 있고, 역으로 Java 클라이언트에서 .NET 환경의 웹 서비스 시스템을 이용해서 .NET 컴포넌트의 메소드를 호출할 수 있다[3][4][6]. 언급한 세 가지 스펙간의 관계는 (그림 2)와 같다.

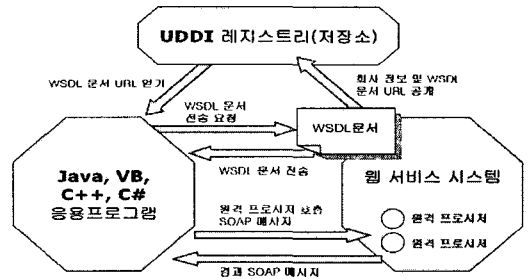


그림 2. SOAP, WSDL, UDDI 간의 관계 Fig. 2 Relationship of SOAP, WSDL, UDDI

웹 서비스 제공자 및 웹 서비스 소비자는 UDDI 스펙에 정의되어 있는 방법대로 웹 서비스 중개자가 운영하는 중앙 저장소에 정보를 공개하거나, 검색해야 한다. 물론 웹 서비스 중개자도 마찬가지로 UDDI 스펙에 정의되어 있는 방법대로 실행되는 중앙 저장소를 구현해야 한다.

웹 서비스 제공자는 WSDL이라는 웹 서비스 기술 언어로 자사가 제공하는 웹 서비스에 대한 상세 기능을 작성해서, 웹 서비스 중개자에 서비스를 공개할 때 WSDL 문서의 URL을 포함시켜 두어야 한다.

웹 소비자는 웹 서비스의 상세 기능을 기술한 WSDL 문서 경로를 확인한 뒤, WSDL 문서를 다운로드받아 호출 가능한 프로시저 이름과 매개변수에 대한 정보를 알아내야 한다. 그리고 웹 서비스 소비자의 프로그래밍 언어로서 웹 서비스의 원격 프로시저를 호출할 준비를 한다.

웹 서비스 소비자는 SOAP 스펙에 정의되어 있는 메시지 형태로 웹 서비스가 제공하는 원격 프로시저를 호출해야 한다. 그리고 웹 서비스 시스템은 결과 값을 다시 SOAP 스펙에 정의되어 있는 메시지 형태로 웹 서비스 소비자에게 보내 주어야 한다[1][3][4].

2. JWSDP (Java Web Services Developer Pack)

Sun사에서 배포하는 JWSDP(Java Web Services Developer Pack)는 자바 개발자들이 XML 응용프로그램과 웹 서비스 시스템 및 클라이언트를 쉽게 개발할 수 있도록 도와주는 여러 가지 도구 및 API를 제공하는 무료 통합 개발 툴킷(toolkit)이다.

다음 (그림 3)은 JWSDP 1.2에 포함되어 있는 전체 API 및 도구 내용을 보여주고 있다.

- XML and Web Services Security
- JavaServerFaces(JSF)
- WS-I Sample Application
- Java Architecture for XML Binding(JAXB)
- Soap with Attachments API for Java(SAAJ)
- Java API for XML Processing(JAXP)
- Java API for XML Registries(JAXR)
- Java API for XML-based RPC(JAX-RPC)
- JavaServer Pages™ Standard Tag Library (JSTL)
- Tomcat
- Ant build tool
- Registry Server

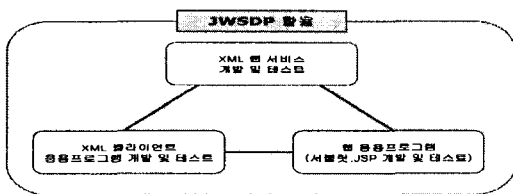


그림 3. JWSDP 활용
Fig. 3 Practical use of JWSDP

III. 유무선 통합 웹 서비스를 위한 서버 시스템 설계

1. 시스템 구조

유·무선 통합 웹 서비스를 위하여 J2EE 시스템을 사용한 웹 서비스 구조와 본 논문에서 설계하고 구현한 서버 시

스템은 (그림 4)과 같으며, 각 기능은 다음과 같다.

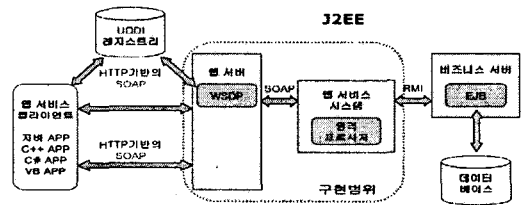


그림 4. J2EE 기반의 웹 서비스 구조
Fig. 4 Architecture of J2EE based Web Services

■ 웹 서비스 시스템

웹 서비스 시스템은 J2EE 애플리케이션 서버가 가지고 있는 서블릿 컨테이너가 될 수도 있고, J2EE 애플리케이션 서버 외부의 시스템일 수 있다. 자바에서는 웹 서비스 시스템 구현 기술로 서블릿 기술을 이용하기 때문에 서블릿 컨테이너에서 웹 서비스 시스템이 실행된다. 서블릿은 웹 서비스 소비자에서 온 원격 프로시저 호출 SOAP 메시지를 받아들여, 해당 웹 서비스 시스템의 메소드를 실행시켜 준다. 그리고 웹 서비스 시스템의 메소드는 EJB의 비즈니스 메소드를 실행시키고, 결과를 다시 서블릿으로 돌려준다. 서블릿은 다시 응답 메시지인 SOAP 메시지를 만들어 웹 서비스 소비자에게 보내준다. J2EE 1.4 스펙에서는 상태없는 세션빈(Stateless Session Bean)을 이용해서 웹 서비스 시스템을 개발할 수 있도록 하고 있다[9].

■ 웹 서버

웹 서버 역시 J2EE 애플리케이션 서버가 가지고 있는 웹 컨테이너가 될 수도 있고, J2EE 애플리케이션 서버 외부의 시스템일 수 있다. 웹 서비스 구조에서 웹 서버의 역할은 두 가지이다. 하나는 WSDL 문서를 웹 서비스 소비자에게 전송시켜 주는 역할이고, 다른 하나는 웹 서비스 소비자와 웹 서비스 시스템 간의 SOAP 메시지를 전달하는 게이트웨이 역할이다. SOAP 메시지 자체는 단순한 XML 문자이지, 전송용 프로토콜이 아니다. SOAP 메시지는 다른 전송용 프로토콜에 실려서 전달되어야 하는데, 대표적인 전송용 프로토콜은 HTTP이다. 웹 서비스 소비자는 원격 프로시저 호출 내용을 담고 있는 SOAP 메시지를 HTTP 프로토콜에 담아서 웹 서버에 접속한다. 웹 서버는 HTTP 내에 있는 SOAP 메시지를 웹 서비스 시스템에 전달한다. 그리고 웹 서비스 시스템이 만든 응답 SOAP 메시지를 다시 HTTP 프로토콜에 담아서 웹 서비스 소비자에게 전송한다.

■ UDDI 레지스트리

UDDI 레지스트리는 J2EE 기반 기술에 포함이 되지 않는다. 회사 내에서 또는 인터넷 환경에서만 웹 서비스를 이용한다면 회사 고유의 UDDI 레지스트리는 설치하여 운영할 수도 있지만, 웹 서비스 소비자가 인터넷상에 흩어져 있다면, 별도의 UDDI 레지스트리를 운영하는 서비스 중개자를 이용하는 것이 좋다. 현재 IBM 및 Microsoft사에서 상업용과 테스트용으로 공개 UDDI 레지스트리를 운영하고 있다. 만약 웹 서비스 제공자가 인터넷상의 웹 서비스 소비자에게 자사의 웹 서비스를 이용하도록 하고, 사용료를 받고자 원한다면, 이들 서비스 중개자의 UDDI 레지스트리를 이용해서 공개해야 될 것이다.

■ 웹 서비스 클라이언트

웹 서비스 클라이언트는 웹 서비스 시스템 개발 환경과는 독립적으로 개발이 가능하다. 즉 .NET으로 구현한 클라이언트는 Java로 구현한 클라이언트든 SOAP 메시지를 생성하고, 전송 및 해석할 수만 있다면 웹 서비스 클라이언트가 될 수 있다.

■ 비즈니스 서버

비즈니스 서버는 J2EE 애플리케이션 서버이다. 비즈니스 기능들은 EJB 컨테이너 상에서 실행되는 세션 빈 또는 엔티티 빈, 메시지 드리븐 빈의 메소드에 해당되며 일반적으로 EJB 서버라고 부르기도 하며, 대표적인 J2EE 애플리케이션 서버로는 BEA사의 웹로직(WebLogic) 서버이다.

본 논문에서 설계하고, 구현한 시스템의 고려 사항은 이동통신 사업자별로 지원하는 언어가 다르므로 접속하는 url 역시 이동통신 사업자의 단말기에 맞게 연결되어야 한다. 예를 들면, 011 단말기는 wml로 만들어진 url을 사용해야 하며, 017, 019 단말기는 hhtml, 016, 018 단말기는 mhtml로 만들어진 url을 사용해야 한다. 이러한 이동통신 사업자별 단말기에 서로 다른 url로 서비스를 하게 된다면 사용자들은 혼동을 초래할 수도 있다. 이럴 경우에는 모든 단말기가 접속하는 초기페이지를 만들어서 접속한 단말기의 헤더를 분석하여 단말기에서 지원하는 언어에 맞는 url로 연결시켜 해당서비스를 이용하게 하면 문제점을 해결할 수 있다.

<표 1>은 접속하는 단말기의 특정 헤더를 검사하여 자기 다른 url로 링크를 시킨다. 현재 사용 중인 웹 서버와 서버 스크립트(asp, php, jsp)를 적용하면 되며, 접속한 단말기에서 특정 헤더를 분석하여 단말기가 지원하는 언어에 맞게

연결한다.

표 1. 이동 통신사별 초기화면 지원 PHP 코드
Table. 1 PHP code fsupporting each Telecom.'s initiation interface

```

Filename : default.php
<? // Phone Header Information, 폰정보를 알기위한
서버변수 얻기
$UserAgent = getenv("HTTP_USER_AGENT");

$phonenumber6
= getenv("HTTP_HTTP_PHONE_NUMBER");
$phonenumber8 = getenv("HTTP_PHONE_NUMBER");
$SubNo = getenv("HTTP_X_UP_SUBNO");
$phone1 = strchr($SubNo,"itouch");
$phone2 = strchr($SubNo,"ezweb");
$pos1
= strchr($HTTP_USER_AGENT,"UP.Browser/3.1");
$pos2
= strchr($HTTP_USER_AGENT,"UP.Browser/3.2");
$pos3
= strchr($HTTP_USER_AGENT,"UP.Browser/4.0");
$pos4
= strchr($HTTP_USER_AGENT,"UP.Browser/4.1");
if (substr($UserAgent,0,3) == "SKT"
|| substr($UserAgent,0,3) == "STI"){
//011,017휴대폰인 경우
$redirect = "wml/Index.php";
}elseif (substr($UserAgent,0,3) == "111"
|| substr($UserAgent,0,3) == "121"){
//LGT-AUR 브라우저폰인경우
$redirect = "wml/Index.php";
}elseif ((strlen($phone1) >=5)
&& (strlen($pos4)>=5)){ //017휴대폰 UP4.1
$redirect = "upwml/Index.php";
}elseif (strlen($phone1) >=5)
{ //017휴대폰
$redirect = "hhtml/Index.php";
}elseif ((strlen($phone2) >=5)
&& (strlen($pos4)>=5)){ //019휴대폰 UP4.1
$redirect = "upwml/Index.php";
}elseif (strlen($phone2) >=4)
{ //019휴대폰
$redirect = "hhtml/Index.php";
}elseif (strlen($phonenumber) >=1
|| strlen($phonenumber6) >=1){
//016,018휴대폰,MBPHONE인 경우
$redirect = "mhtml/Index.php";
}elseif (strlen($pos1) >=5 || strlen($pos2) >=5)
{ //UP브라우저3.1/3.2
$redirect = "hhtml/Index.php";
}elseif (strlen($pos3) >=5 || strlen($pos4) >=5)
{ //UP브라우저4.0/4.1
$redirect = "wml/Index.php";
}else { //IE인경우(유선홈페이지가 있는경우)
$redirect = "mhtml/Index.php";
}
header("Location:http://localhost/kwcs/
".$redirect);
?>
    
```

2. 유무선 게시판의 연동

무선인터넷 게시판의 설치를 이전에 유선 웹 게시판을 설치하여 웹 브라우저 상에서 게시판의 글을 입력하고 지우고, 수정 및 삭제하는 작업을 한 뒤 유선 게시판과 Database를 기준으로 무선 게시판을 연동한다. 무선인터넷 게시판은 ASP+WML, HDML, mHTML, UPWML 언어로 개발하였으며, 게시판 데이터베이스 내에 게시판 테이블

을 공유하여 사용한다. 각각 작성해하는 페이지는 게시판 메뉴페이지, 목록보기, 내용보기, 관련글보기, 글쓰기, 글등록, 검색하기, 검색결과표시, 답변글작성, 작성글수정등 각각의 페이지를 처리해야 한다.

표 2. 단말기에 따른 콘텐츠 정보 표시
Table. 2 Contents information presentation for each device

```

FileName : Mview.asp
<%
'표시할 내용이 폰에서 제공하는 사이즈를 초과할 경우
'특정사이즈로 나누어 여러 페이지에 걸쳐 표시.
dim totParano '총 나눌 문단수

function multiPageview(byval str, byval pno, byref
totParano)
    bytecount = 0
    pageunit=1
    '특정사이즈 셋팅
    decksize=400

    if len(str)>0 then
        for i=1 to Len(str)
            if Asc(Mid(str, i, 1)) < 0 then
                bytecount = bytecount + 2
            else
                bytecount = bytecount + 1
            end if
        next

        if bytecount>=(decksize * pageunit) then
            ' 문자조합(^@|)를 사용하여 페이지를 나눔.
            tmp = tmp & Mid(str, i, 1) & "^@|"
            pageunit=pageunit+1
        else
            tmp = tmp & Mid(str, i, 1)
        end if
    next

    dim arraystr
    arraystr = split(tmp,"^@|")

    if pno-1 > ubound(arraystr) then
        multiPageview = arraystr(ubound(arraystr))
    else
        multiPageview = arraystr(pno-1)
    end if
    '총 나눌문단수
    totParano = ubound(arraystr) + 1
else
    multipageview = str
    totParano = 0
end if
end function
%>
    
```

페이지 처리에 있어, 한 페이지의 내용이 길 경우 멀티 페이지 처리는, WAP 단말기가 매우 작은 메모리를 가지고 있기 때문에, WML deck의 크기에 제한이 있다. 이 크기는 서버에서 보내는 단순한 텍스트 XML 코드가 아니라 게이트웨이를 거쳐서 WAP 단말기로 가는 컴파일된 코드의 크기를 말한다. 컴파일되지 않은 파일의 크기는 매우 큰데 게이트웨이를 통과하면서 컴파일되어 사이즈가 작아지게 된다. UP 브라우저 3.2의 경우 1492byte가 최대크기이며 UP 브라우저 4.x 경우 2048byte의 제한이 있다. 따라서 데이터베이스로부터 읽어오는 글의 양이 많을 경우 내용을

여러 페이지에 걸쳐 나누어 보여 주어야 한다.

〈표 2〉는 입력된 기타 정보가 긴 경우 특정 byte 수로 페이지를 분리하는 함수를 사용하여 특정사이즈를 정하며 여러 페이지로 나누어 사용자에게 보여줄 수 있도록 한다.

3. 시스템 정보 구조

실질적인 정보 구조 설계 전에 사전검토 작업을 완료하고, 그 다음으로 사이트 전체의 정보흐름을 적용한 정보 구조 설계를 했다. 이 과정에서 사이트의 목적과 사용자가 필요로 하는 것을 반영한 상위 메뉴, 메뉴의 우선 순위, 향후 확장 가능성이 있는 구조를 결정해야 한다. 특히 상위 메뉴의 변경 또는 추가는 많은 비용 또는 리소스가 필요하다는 사실을 반드시 공유하고, 초기에 이에 대한 대안을 모색하는 것이 바람직하다. (그림 5)는 본 논문에서 설계한 학과 시스템을 기반으로 SMS 계층 구조를 나타낸다.

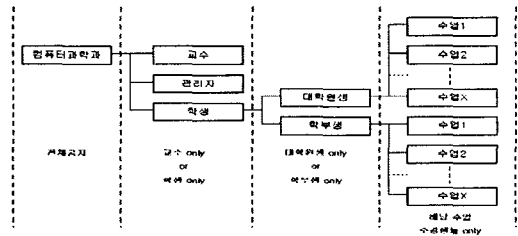


그림 5. SMS push 서비스 view 계층 구조
Fig. 5 hierarchical architecture of SMS push service view

색션의 경우 메뉴 결정의 원칙과 메뉴의 우선 순위 선정 및 변동 관계는 어떻게 해야 하는가에 대한 논의와 가이드 라인을 공유하고 향후 운영 및 관리할 수 있을 때 초기의 컨셉이 파괴되지 않도록 구성해야 한다. 특히 자칫하면 운영자 위주 또는 기술 편향적인 성향이 부각될 수 있는 위험성을 배제하는 원칙을 반드시 숙지해야 한다.

또한 SMS push 서비스 view 계층 구조를 위해 SMS는 기본적으로 양방향 무선호출 기능을 가지고 있고, IS-95-A 상의 Data Burst Message를 이용하여 단말기와 기지국사이에 송수신 하게 되며, 일반적으로 상당히 제한적인 길이의 메시지만 사용한다. 따라서 메시지의 길이의 발신은 100byte, 수신은 150byte 이내로 사용한다. SMS를 사용하는 방법도 여러 가지여서 메시지 수신인 MT(Mobile Terminated), 메시지 발신인 MO(Mobile Originated), 기지국에서 Broadcast 하는 방송형 서비스, 양방향 통신특성을 이용한 주문형 서비스 등이 있다[10].

SMS와 push 메시지를 전송하는 방식은 Gateway 서버를 경유하는 방법과 모뎀을 이용한 방법이 있는데 이러한 서비스를 하려면 제공업체에 유료로 이용하고 인터페이스와 관련 프로그램을 제공받아야 하기 때문에 본 논문에서는 생략하였다.

IV. 구현

1. 구현 구조

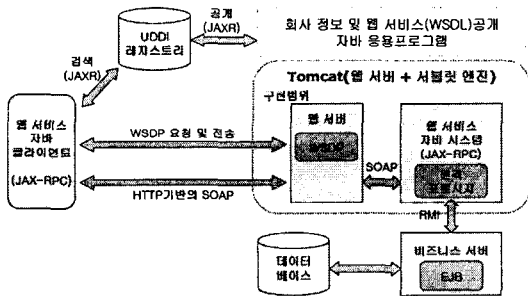


그림 6. JWSDP를 이용한 웹 서비스 개발 구조
Fig. 6 Web services development architecture using JWSDP

JWSDP는 동기적인 웹 서비스를 개발해 주는 각종 API 및 도구를 가지고 있다. (그림 6)은 JWSDP를 이용한 웹 서비스 개발 구조와 본 논문에서 구현하는 범위이다. 웹 서버는 JWSDP 1.2 버전 안에 포함되어 있는 Tomcat을 사용하고 Apache와 함께 연동하여 사용할 수도 있다. 본 논문에서 구현시에는 Apache와 Tomcat을 연동하여 구현하였다.

JWSDP 1.1 버전에서는 메시지 기반의 비동기적인 웹 서비스 시스템을 개발할 수 있는 JAXM(Java API for XML Messaging)가 포함되어 있으나, JWSDP 1.2부터는 JAXM이 포함되지 않았다. JWSDP 1.1은 JWSDP 1.2와 사용 방법에 있어서는 크게 다르지 않다. 그래서 본 논문에서는 JWSDP 1.2를 이용해서 요청/응답형의 동기적인 웹 서비스 시스템에 대해서 다루었다.

웹 서비스를 제공하려면 J2SE 버전은 1.3.1_07부터 사용해야 한다. 운영 체제는 가급적 Windows 2000 Server

급 이상을 사용하는 것이 좋다. 왜냐하면 웹 서비스 시스템(Web Service System)은 Servlet 엔진이 구동 중인 웹 서버에서 실행되는 것이다. 따라서 웹 서버를 설치 운영할 수 있는 운영 체제 버전이어야 한다. Windows 2000 Professional이나 Windows XP Professional은 개발 및 테스트 용도로는 사용할 수 있으나, 웹 서비스(Web Service)를 배치해서 실제 서비스 용도로는 사용할 수 없다. 왜냐하면 동시 사용자수가 10명으로 제한되기 때문이다.

본 논문의 구현 환경은 다음과 같다.

- Web Server와 Server Side Script(ASP, JSP, PHP) 사용할 Server Side Script를 사용하기 위해 운영체제와 웹 서버를 결정해야 한다. 본 논문에서 사용한 운영체제는 Windows 2000 Server이고 웹 서버는 Apache와 Tomcat을 사용하였다.

- 인터넷 웹 서버에 MimeType 추가
- Server Side Script 언어별 ContentType 선언 형식 Script 파일 내의 Header가 선언되지 않은 경우는 Contenttype Error(HTTP406 error)가 발생한다.

- 웹 서버 환경 설정 (Apache)
Apache를 설치한 후 ApacheGroup/Apache/Conf 폴더 내의 mime.types 파일에 MimeType을 추가한다.

2. 구현 사례

앞서 제시한 것처럼 학과 사이트를 대상으로 유무선 통합 웹 서비스를 위한 서버 시스템을 구성하였다. 구현된 것을 보기에 앞서 유무선 통합 서버 시스템을 설치하기 위한 프로그램을 만들어 설치하도록 하였다. 웹 서버 파일을 자동으로 설치하는 방법을 선택하였으며 서버와 데이터베이스의 연동은 따로 설정을 해야만 한다.

(그림 7)는 실제로 유무선 인터넷 게시판이 연동됨을 보여주고 있다. 011, 017, 016, 018, 019 각 통신사에 따라 해당 페이지로 이동한다. 실시간으로 목록보기와 글쓰기 및 검색까지 가능하다.

(그림 8)은 XML 방식으로 검색과정을 보여준다. 이 그림에서 검색어를 입력하면 해당되는 내용을 골라서 보여지게 된다. (그림 9)는 무선인터넷 접속화면인 메인 페이지와 게시판이다.

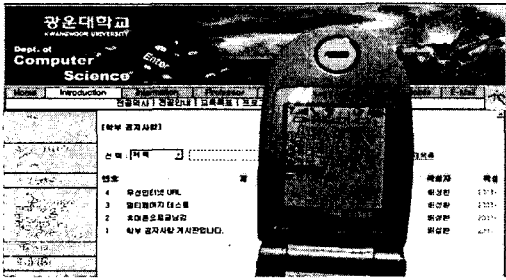


그림 7. 유무선 인터넷 게시판 연동
Fig. 7 Internet board linkage for wired & wireless

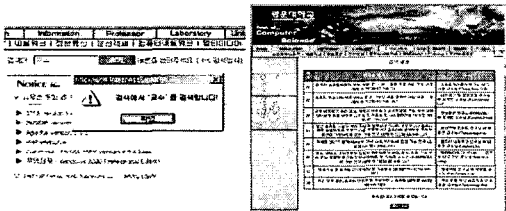


그림 8. XML 방식을 사용한 검색
Fig. 8 Search using XML technique

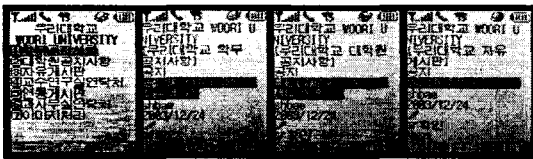


그림 9. 무선인터넷과 게시판의 연동
Fig. 9 linkage of wireless internet and wired internet board

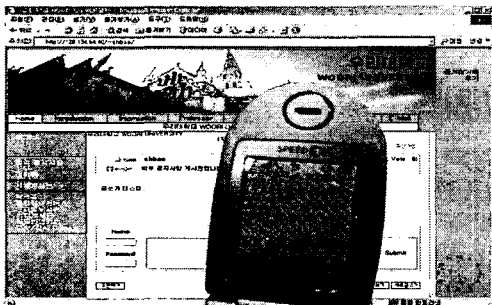


그림 10. 인터넷 게시판의 연동 화면
fig. 10 linkage interface of internet board

본 논문에서 앞서 제시한 것처럼 유무선 인터넷 연동의 중요성은 실시간으로 사용자에게 정보를 제공하기 위해서이다. (그림 10)는 무선인터넷 게시판과 연동되는 화면을 보인 것이다.

V. 결론

웹 서비스는 애플리케이션 통합과 동일한 개념이라고 생각할 수 있고 시스템을 구축하기 위한 기술이 아니라 형태 내의 기능들을 연결하고 통합하는 기술이다. 인터넷과 XML 등장 이후, 차세대 IT 패러다임으로 급부상하고 있는 웹 서비스는 XML에 기반을 둔 플랫폼과, 구현언어에 독립적인 컴포넌트 기반의 분산 컴퓨팅 기술로 인터넷 상 분산 애플리케이션 통합 방법의 대표주자로 자리를 잡아가고 있다. 웹 서비스는 기존의 폐쇄 환경의 통합 기술로도 사용될 수 있다. DCOM으로 개발된 환경과 J2EE 환경, 그리고 CORBA로 구축된 환경을 하나로 통합시킬 수 있다. 이것이 가능한 이유는 개발형 프로토콜인 HTTP를 이용해서 XML 메시지를 전달하여 통신하기 때문이다.

본 논문에서는 유무선 통합 웹 서비스를 위한 서버 시스템을 개발할 수 있도록 통합 개발 환경을 설계함에 있어서 시스템 구조와 유무선 통합 사이트 구축시 고려사항을 감안하여 동일한 URL로 유무선 사이트를 구축하는 것과 게시판의 연동, WAP 단말기는 작은 메모리를 가지고 있기 때문에 글의 양이 많을 경우 내용을 여러 페이지에 걸쳐 나누어 보여주는 멀티 페이지 처리를 하였고, 시스템 정보구조를 설계 및 구현, SMS push 서비스 View 계층 구조를 설계하였다. 본 논문에서 제안한 유무선 통합 웹 서비스 개발 환경은 JWSDP, 무선인터넷 Server Side Script, WML Editor 등으로 구성되어 있다. 웹 서비스 환경이기 때문에 검색은 XML 방식을 사용하였다. 또한 이들을 단일한 환경으로 통합한 유무선 통합 환경으로 개발하였다. 현재, 웹 서비스와 무선인터넷 기술은 계속 발전해 가고 있으며, 관련 스펙들은 꾸준히 보완되고 있고 본 논문에서 소개된 유무선 통합 웹 서비스 및 무선 인터넷 기능을 지원하기 위하여 지속적인 연구가 수반되어야겠다.

참고문헌

- [1] David A. Chappell, Tyler Jewell, "Java Web Services", O'Reilly, 2002. 3.
- [2] Andrew Lader, Lonnie Wall, "Building Web Services and .NET Applications", McGraw-Hill, 2002. 3.

- [3] Simeon Simeonov, "Building Web Services with Java (Making Sense of XML · SOAP · WSDL · UDDI)", SAMS, 2002.
- [4] Henry Bequet, "Beginning Java Web Services", Wrox Press, 2002. 9.
- [5] DEITEL, DEITEL, NIETO, LIN & SADHU, "XML How to program", Prentice Hall.
- [6] 신민철, "熱血講義 XML 웹서비스", FREELEC, 2003.
- [7] <http://www.w3c.org/2002/wsdl/>
- [8] 이종호, 이람, 최병호, "인포메이션 아키텍처 웹사이트 설계의 법칙", 한빛미디어, 2003.
- [9] 이종언, 차시호, 조국현 "J2EE 기반 망 관리 시스템의 프리젠테이션 계층 프레임워크의 설계", 한국정보과학회 춘계학술발표 논문집 29권 1호 p.460~462, 2002. 4.
- [10] Anthony Y. Lui, Shanika A. Karunasekera, and Chokchai Leangsuksun "The Enhanced Service Manager: A Service Management System for Next-Generation Networks", Bell Labs Technical Journal, July-September 2000.
- [11] Si-Ho Cha, Jae-Oh Lee, Yeong-Keun Choi, Kuk-Hyun Cho, "Implementing an XML-based Universal Network Management System in Java", Springer-Verlag's Lecture Note in Computer Science (LNCS) Vol. 2343, June 2002.
- [12] <http://www.sun.com/software/learnabout/>
- [13] <http://msdn.microsoft.com/>



이 종 언

2001년 광운대학교 전자계산학과 졸업(이학사)
 2003년 광운대학교 대학원 컴퓨터 과학과 졸업(공학석사)
 2003년 ~ 현재
 광운대학교 대학원 컴퓨터과학과 박사과정
 <관심분야> 망 관리, 차세대 네트워크, 이동 애드혹 망, 소프트웨어 컴포넌트 기술



차 시 호

1995년 순천대학교 전자계산학과 (이학사)
 1997년 광운대학교 대학원 전자계산학과(이학석사)
 2004년 광운대학교 대학원 컴퓨터 과학과(공학박사)
 1997년 ~ 2000년
 대우통신 종합연구소 선임연구원
 2004년 ~ 현재
 (주)웨어플러스 기술개발팀 팀장
 <관심분야> 네트워크 소프트웨어, 네트워크 관리, 차세대 인터넷, 소프트웨어 컴포넌트 기술



김 규 호

광운대학교 전자계산학과(이학사)
 광운대학교 대학원 전자계산학과 (이학석사)
 광운대학교 대학원 전자계산학과 (이학박사)
 서울보건대학 인터넷정보과 교수
 한국컴퓨터정보학회 이사
 <관심분야> 차세대 네트워크 관리, 사이버교육

저 자 소 개



배 성 환

2002년 서경대학교 컴퓨터과학과 졸업(이학사)
 2004년 광운대학교 대학원 컴퓨터 과학과 졸업(공학석사)
 2004년 ~ 현재
 (주)아이넷씨엔씨 연구소
 주임연구원
 <관심분야> 네트워크 관리, 웹 서비스, 소프트웨어 컴포넌트 기술