

무선 인터넷플랫폼 환경에서의 XML웹서비스구현

권두위·김종율·도경훈

동서대학교 컴퓨터정보공학부

The Implementation of XML Web Service for Wireless Internet Platform Environment

Doo Wy Kwon, Jong Ryul Kim, Kyeong-Hoon Do

Division of Computer & Information Engineering, Dongseo University

E-mail : kdoowy@nate.com, {xmaskjr khdo}@gdsu.dongseo.ac.kr

요 약

XML웹서비스는 인터넷 상에서 표준화된 오픈 네트워크를 통해 기업내 및 기업 상호간의 모든 컴퓨터 시스템을 결합하려는 새로운 분산 컴퓨팅 패러다임으로서, 이를 통해 기업의 수익증대와 비용절감의 효과를 얻을 수 있을 것으로 기대된다.

특히 모바일 매쉬업을 위해서는 XML웹서비스의 처리가 필수적인데, 본 논문에서 개발한 WIPI 기반의 XML 표준 기술과 모바일 웹서비스 표준 기술을 통해 이러한 응용 개발이 가능하며 이러한 XML웹서비스를 실현하는데 표준 기술에는 XML과 HTTP가 있다. 본 논문에서는 WIPI 환경에서 사용할 수 있는 간단한 XML 파서를 구현하고 HTTP 통신이 가능하도록 하여 XML 웹서비스의 서비스 개발에 적용함으로써 모바일 환경에서의 XML 웹서비스 인터페이스를 구현하였다. 또한 이에 대한 실제 응용으로서 WIPI용 Java언어를 기반으로 학사지원시스템에 적용하였다.

키워드

WIPI, XML웹서비스, RPC, SOAP

I. 서 론

XML 웹서비스란 소프트웨어 기술의 발전과 함께 진화해 온 성숙된 분산 기술로서 분산 컴퓨팅 환경에서 오랫동안 컴퓨팅 모델이었던 RPC (Remote Procedure Call)를 플랫폼을 독립적으로 사용하려는 시도에서 시작되었다. 이러한 XML 웹서비스는 인터넷 기술을 이용한 표준화된 오픈 네트워크를 통해 기업내 및 기업간 모든 컴퓨터 시스템을 결합시키는 새로운 컴퓨팅 패러다임으로서, 이를 통해 기업의 수익증대와 비용절감의 효과를 얻을 수 있는 활동이라고 정의할 수 있다.

본 연구에서는 이러한 XML 웹 서비스를 실현하고자 인터넷 표준 기술인 XML과 HTTP를 사용하였다. XML 기술은 표현 가능성이 한정되어 있는 텍스트 기반의 프로토콜 기술들을 대체할 것으로 예상되며 주고받을 데이터를 XML로 나타내고 이에 대한 상대방의 서비스 인터페이스도 XML로 표현하기 위해 사용되는데 주고받을 데이터 및 서비스 호출 메시지를 위해 XML 형식의 SOAP을 사용하였다. 본 논문에서는 이러한 XML 웹 서비스를 실현하는데 주로 사용되는 인터넷 표준 기술이 XML과 HTTP라는 것에 착안하여 그에 대한 구현 및 실제 학사 관리시스템에 적용해 보는 것을 목표로 한다. 또한, 본 논문에서는 설계 및 구

현한 모바일 웹서비스를 위한 XML라이브러리 중 WIPI API를 중심으로 기술한다. II장에서는 XML웹서비스에 대한 기술과 함께 RPC의 방법과 XML웹서비스의 장점에 대해 기술하고, III장에서는 SOAP의 기능에 대한 기술과 SOAP의 통신에 대해 기술하고, IV장에서는 WIPI API의 구현 과정과 학사지원시스템에 적용하여 개발한 WIPI API의 활용을 보여준다. 결론에서는 WIPI API의 개발 과정에서 제기된 문제점과 개선 방안에 대하여 기술한다.

II. XML웹서비스

1. XML웹서비스의 기능

현재까지 웹은 사람들을 위한 서비스를 제공하였다. 링크로 문서를 사람들에게 제공하고, 웹 브라우저를 통해 수동으로 수행되는 트랜잭션같은 작업을 제공하였다. 그러나 기존의 웹환경은 데이터베이스 서버에 있는 정보들을 수작업으로 업데이트를 시켜야 한다는 번거로움을 필요로 한다. 이러한 기존 컴퓨팅의 문제점이 몇 가지 존재한다. 첫째로 서로 다른 언어로 개발된 서비스는 상호 호환성이 없다. 둘째로 분산 컴포넌트간의 프로토콜이 상이하고 데이터 교환의 표준이 없어서 상호 호환성을 확보할 수 없다. 세 번째로는 상호간 플랫폼이 다르다면 동종의 응용 서비스라고 하더라도

다시 해당 플랫폼에 맞게 재개발해야 한다. 또한 얻고자 하는 서비스의 존재여부도 검색할 방법이 없다. 넷째로 특정언어, 플랫폼, 컴포넌트 모델에 의존적이어서 동일한 서비스를 여러 번 개발하는 중복투자의 가능성이 아주 높아진다. 마지막으로 사용자의 행동에만 의존하는 수동적인 형태로 서비스되고 기업 대 기업 간 서비스 연동을 위한 표준이 존재하지 않다. 따라서 XML웹서비스는 이러한 문제점을 해결하기 위해서 MS, IBM, SUN등 세계 IT를 이끌어 가는 대기업에서 주도하여 만들어낸 것이다. [1]

2. RCP의 방법

XML웹서비스를 설명하는데 중요한 개념 중 하나가 RPC이다. RPC는 웹서비스의 근간을 이루는 매우 중요한 기술로서 분산 컴포넌트 기술의 핵심으로 볼 수 있다. 이러한 RPC는 어떤 프로그램이 네트워크에서 물리적으로 다른 컴퓨터에 위치한 프로그램에게 서비스를 요청할 때 사용하는 프로토콜로서 최근에는 개발자가 개발한 구현을 바탕으로 컴파일에서 자동으로 호출하는 쪽과 호출 받는 쪽의 프로토콜을 맞추는 과정을 수행하도록 되어 있는 것이 일반적이다. 이때 자동으로 수행하는 과정에서 스텝과 스케레톤의 두 파일이 생성된다. 여기서 스텝이란 클라이언트 프로그램에서 서버프로그램으로 필요한 정보를 주고받기 위한 중간매체이고, 클라이언트 런타임에 의해 호출된다. 반면에 스케레톤은 서버 프로그램에서 클라이언트와 정보를 주고받기 위한 매개체로서 서버 런타임에 의해 호출된다.

3. XML웹서비스의 장점

XML웹서비스에서 XML과 인터넷 표준 프로토콜을 이용함으로써 어느 한 플랫폼이나 언어에 국한된 것이 아니고 많은 업체들이 표준화 과정에 참여하여 합의해서 만들어낸 표준으로서 오픈 스탠더드로 인식될 수 있으며 XML웹서비스는 오픈 스탠더드로 개발된 첫 번째 분산 컴퓨팅 모델이라 할 수 있다. 오픈 스탠더드로서의 XML웹서비스는 다음과 같다.

- 서비스 이용도(Availibility) : 이미 존재하는 인터넷 관련 기술을 이용
- 서비스 이용 용이성(Transparency) : HTTP가 되는 곳에서는 쉽게 이용가능
- 서비스 추상화(Encapsulation) : 내부가 어떻게 구현되었는가에 대해 사용자가 알 필요가 없음
- 플랫폼 독립성(Platform Independent) : 특정 기술에 얽매이지 않음
- 표준기반(Standard Based) : HTTP, XML기반 기술 이용
- 상호 운용성(Interoperability) : 표준 기술사용으로 상호 운용성 확보
- 지원 용이성(Support) : 표준 기술사용으로 다양한 기술적 지원 가능
- 표준화된 서비스 : 어려운 기술적인 내용은 감

추고 분산되어 있는 프로그램을 사용자가 원하는 서비스의 형태로 제공함, 표준화는 개발자나 개발 업체의 관점에서 장점이고, 서비스는 사용자의 관점에서의 장점

이와 같은 장점을 지닌 XML 웹서비스의 성공요인은 표준화와 MS, IBM, SUN등 세계 IT를 이끌어 가는 대기업에서 주도하기 때문이라고 할 수 있다. 표준화란, 예를 들어 비디오테이프, 전기 플러그, 기차 레일 등과 같이 기본적인 표준 인프리가 구축되어 있는 경우에는 그것을 바탕으로 무궁무진하게 활용이 가능 하도록 하는 규격을 말한다. 과거 Sony의 Bete 비디오 테이프가 우수한 품질에도 불구하고 마츠시다 및 JVC등과 같은 업계 주도 그룹에 의해 만들어진 업계표준 방식인 VHS에 밀려 시장에서 퇴출된 사례가 있다. 이와 달리 웹서비스는 IT분야의 최소한도의 기술표준, IT업계가 처음으로 협의하여 만들 표준기술로서 그 성공적인 확산 가능성은 높다고 볼 수 있다 [1].

III. SOAP 메시지

SOAP (Simple Object Access Protocol)은 분산 환경에서 정보를 교환하기 위한 간단하고 가벼운 표준 XML 프로토콜로 정의할 수 있으며 그 특징으로는 XML형식으로 된 문서이고 XML웹서비스에서 메시지 스택에 해당한다. 단순히 메시지 형식만을 정의하여 이기종간 데이터 교환 문제를 해결한 프로토콜이며 응용 프로그램 사이에서 공유되는 데이터 형식을 정의하는 프로토콜이다. 이러한 SOAP의 설계 목표는 그 정의에도 나타나 있듯이 간단함 및 확장성(Simplicity & Extensibility)에 있다. 즉, 분산 프로그래밍을 위한 어떠한 특별한 내용이나 제한 사항을 담고 있지 않다 [2].

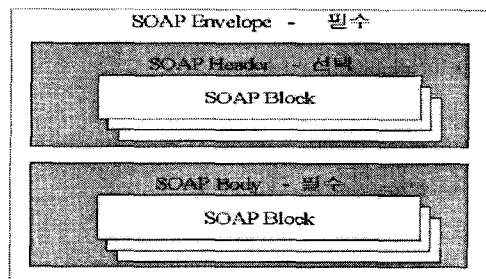


그림 1. SOAP 메시지의 구조

SOAP Header에는 메시지를 처리하는 방법, 메시지 라우팅, 인증, 트랜잭션 등을 정의한다. 그리고 SOAP Body에는 실제의 XML형식 메시지가 정의되는 부분으로 호출하고자 하는 함수명, 인자 등이 기술된다. SOAP Body내에는 SOAP Block형태의 Body Block들이 정의된다.

SOAP 메시지를 이용하여 RPC를 처리하는 방법은 웹서비스에서는 SOAP HTTP Request로 서비스를 호출하고 서비스 호출의 결과는 HTTP

SOAP 메시지로 전달받는 것을 의미하고 필요한 정보는 목적지 URL, 메서드 이름, 메서드 용법, 메서드 인자, 헤더 정보가 있다.

IV. 구현테스트

1. WIPI상에서의 XML 웹서비스의 구현

학사시스템은 서버와 클라이언트상에 통신을 HTTP 통신을 이용하여 서로 데이터를 송수신한다. 서버에서는 클라이언트에서 보내오는 SOAP 메시지를 간단한 파서를 통해 파싱하여 데이터만을 검출하여 데이터와 DB의 데이터를 비교하고, DB 데이터를 SOAP 메시지로 클라이언트에 보내게 된다. 클라이언트 또한 서버에서 송신해온 SOAP 메시지를 파싱하여 데이터를 처리한다.

Request

```
POST /axis/services/SuDaang HTTP/1.0
Content-type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.4
Host: 127.0.0.1:5678
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: ""
Content-Length: 338

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ViewsDB soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"?>
      </soapenv:Body>
    </soapenv:Envelope>
```

그림 2. 요청 SOAP 메시지

그림 2는 클라이언트에서 SOAP 메시지를 서버로 송신하여 원하는 정보를 요청한다. 서버에서는 그림 3과 같이 응답을 하는 SOAP 메시지를 송신한다.

Response

```
POST /axis/services/SuDaang HTTP/1.0
Content-type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.4
Host: 127.0.0.1:5678
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: ""
Content-Length: 338

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ViewsDB soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"?>
      </soapenv:Body>
    </soapenv:Envelope>
```

그림 3. 응답 SOAP 메시지

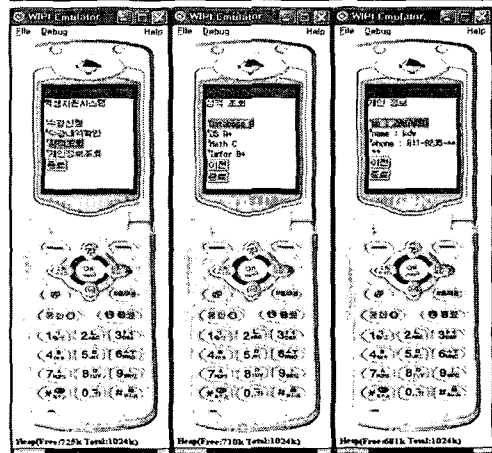
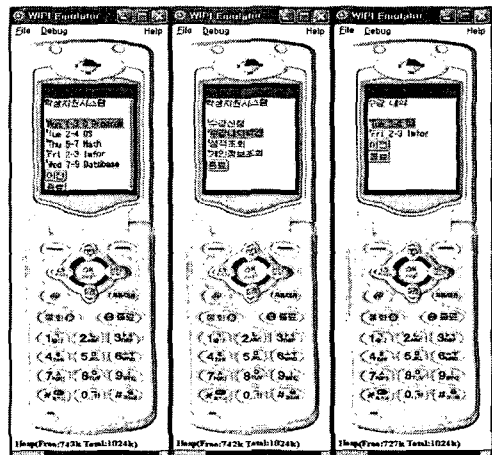
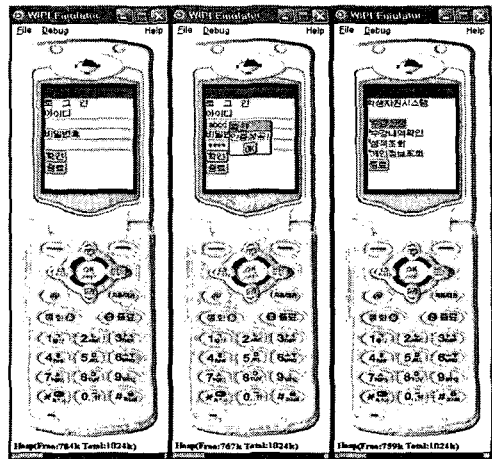


그림 4. WIPI API 구현

서버와 클라이언트간 SOAP 메시지는 String으로 송수신하게 하면된다. 서버는 JAVA의 Class를 deploy시켜 모든 상황을 처리 하게 된다. 또한, 수신한 SOAP 메시지에 대해 WIPI상에서는 XML의

파서를 제공하지 않았으므로 원활한 처리를 위해 아주 가볍고 경량화 된 XML파서를 구현하여 처리하도록 한다.

2. 학사시스템 구축 및 테스트

XML웹서비스 구축에 따른 서버는 JAVA기반의 Apache를 주 서버로 사용을 하고 웹서비스용으로 Tomcat서버와 AXIS를 설치하여 사용을 하였다. 클라이언트는 WIPI API를 통해 웹서버에 등록 되어있는 학사지원시스템의 내용 중 로그인, 수강내역조회, 수강신청, 수강신청 내역보기, 개인 정보 조회등을 구현하였다. 그림 4는 구현테스트 내용이다 [2].

로그인을 시도하게 되면 ID와 Password를 SOAP 메시지로 서버로 전송하여 서버에서는 내용을 검색하여 인증여부를 클라이언트로 SOAP 메시지로 전송을 하여 다음 서비스를 이용할 수 있다. 인증 후 수강신청 메뉴에서 현재 개설되어 있는 강좌를 열람 후 선택을 하게 되면 자신이 원하는 과목을 서버의 DB로 저장을 하게 된다. 그리고 자신이 신청한 내용을 확인하고자 할 때는 수강내역조회를 하여 자신의 신청한 과목의 요일과 시간, 과목명을 조회 할 수 있다. 성적조회와 개인정보 조회는 서버에 있는 DB에서 자신의 성적과 간단한 개인정보를 확인 할 수 있다.

그림 5는 서버와 클라이언트의 통신 사이에 SOAP 메시지의 통신여부를 서버에서 검색 할 수 있는 AXIS라이브러리에서 제공하는 TCP Monitor를 통해 SOAP 메시지를 검색한 내용이다.

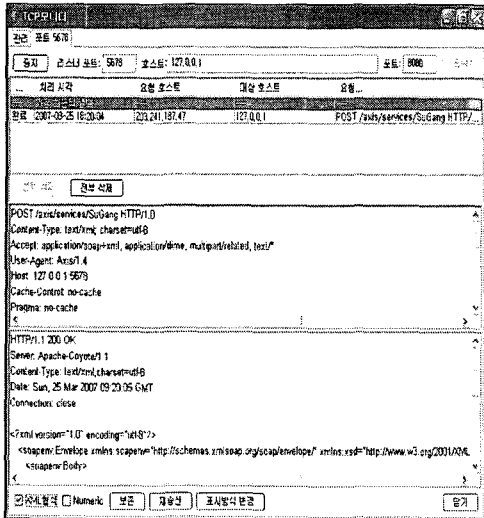


그림 5. TCP Monitor

V. 결론

본 연구에서는 서버와 클라이언트가 SOAP 메시지를 이용한 통신에서 표준화된 XML 웹서비스의 장점을 최대한 사용 하였다. 현재 학교에서 이용할 수 있는 학사지원 관리시스템의 서비스 중 일부만을 포함하고는 있지만, 무선이라는 점을 감안했을 때 많은 서비스를 제공하기보다 이동시 가장 편하게 사용할 수 있고, 간단한 내용만을 사용한다는 점을 착안하여 불필요한 요소를 줄이고 효율성이 높은 내용만으로 구현하여 그 실제 시스템에서의 활용 가능성을 확인하였다. 현재 휴대폰에서 통신을 하게 되면 데이터 전송량에 따라 그 비용과 시간에서 손실이 많이 발생 하지만, 이동성을 지니고 있어 그 효과는 상당히 기대된다. 서버의 처리 용량 및 비용문제를 해결, 보안하여 앞으로 학생들이 많이 사용할 수 있도록 지원을 할 계획이다.

참고문헌

- [1] 권수갑, "Web Service동향",중소기업청 정보화지원단 2003.6
- [2] 오지훈, "Web Services 통신 프로토콜 분석", "인프라벨리"2006.7
- [3] 모바일 표준 플랫폼 규격 2.0 WIPI 2.0, KWISFS.K-05-002
- [4] 박건태, 김승엽, "Jlet으로 배우는 위피 프로그래밍", 한빛미디어, 1999
- [5] 박수원, 안은석, 이경철, "위피 모바일 프로그래밍", 한빛미디어, 2003
- [6] W3C, Web Services Architecture, www.w3.org
- [7] "Mobilejava Developer Community" www.mobilejava.co.kr