

# 모바일 주소록 웹서비스를 위한 메시지 처리부

유가연, 이은정  
경기대학교 일반대학원 전자계산학과  
e-mail: {despoin, ejlee}@kyonggi.ac.kr

## Message Processing for Mobile AddressBook Web Services

Ga-Yeon Yoo, Eun-Jung Lee  
Dept of Computer Science, Kyonggi University

### 요 약

어플리케이션 간에 웹서비스를 이용한 통신이 증가함에 따라 SOAP 메시지를 처리하기 위한 웹 서비스 엔진이 널리 사용되고 있다. 웹서비스 클라이언트를 개발하기 위하여 WSDL로부터 바인딩 클래스를 자동 생성하는 Axis 기반의 방법이 흔히 사용되고 있다. 본 연구에서는 모바일 환경의 J2ME Axis 엔진인 MIRAE 베타 버전을 이용한 웹서비스 클라이언트 구축 경험을 소개한다. MIRAE 엔진은 StAX 파서를 사용하며 바인딩 클래스의 자동 생성을 지원하지만, 생성된 클래스가 지나치게 커져서 모바일 환경에 탑재하기에 어려움이 있었다. 본 논문에서는 StAX 파서의 드라이버를 작성하여 메시지로부터 객체를 직접 바인딩 하도록 설계한 메시지 처리부를 소개한다. 이 방법은 필요한 데이터를 수신 SOAP 메시지로부터 직접 가져오고 전송할 때도 프로그램의 데이터를 바로 SOAP 메시지로 만들어 클라이언트의 코드 크기를 줄여준다.

### 1. 서 론

최근 웹 서비스의 확산으로 XML기반의 SOAP을 이용한 데이터 송수신이 일반화됨에 따라 웹서비스 플랫폼에서는 데이터를 직렬화하여 SOAP 메시지를 생성하고 처리하기 위하여 웹서비스 엔진이 널리 사용되고 있다.

가장 대표적인 웹서비스 엔진으로는 아파치의 Axis와 Sun의 JWSDP를 들 수 있다[1,2]. 이들 시스템은 웹서비스에서 제공하는 오퍼레이션에 대한 정의를 나타내는 WSDL을 이용하여 스텝과 바인딩 클래스를 자동으로 생성하게 된다. 그 결과 프로그래머는 스텝을 이용하여 SOAP메시지를 생성하고 전송시키거나 웹서비스로부터 받아온 메시지를 처리할 수 있게 된다.

한편 모바일 웹서비스 클라이언트를 위한 웹서비

스 플랫폼으로는 J2ME 환경에서의 웹서비스 표준인 JSR172를 구현한 Axis MIRAE를 사용할 수 있다 [4]. Axis MIRAE는 XML을 스트리밍 방식으로 파싱하는 StAX 파서 API를 사용하여 SOAP을 처리하는 방식을 제공한다. 그러나 스텝 코드 생성 방식은 J2SE와 동일하다.

본 논문에서는 모바일 환경의 웹서비스 어플리케이션 사례로 주소록 웹서비스 클라이언트를 개발하였다. 주소록 표준인 vCard를 기반으로 하는 웹서비스를 구축하였으며, 이를 지원하는 모바일 클라이언트를 개발하였다. 본 연구에서는 스텝 자동 생성 방식이 가지는 생성 코드의 비효율성 때문에 모바일 환경에서는 기존의 스텝 자동 생성 방식이 그대로 사용되기 어려울 것임을 확인하였다. 개발된 클라이언트는 StAX를 이용하여 SOAP 메시지를 직접 생성

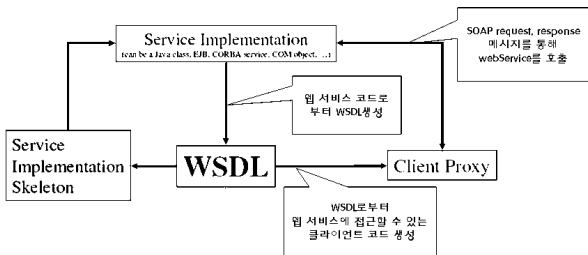
하고 처리한다.

본 논문의 구조는 다음과 같다. 1장의 서론에 이어 2장에서는 관련 연구로 웹서비스를 위한 Axis플랫폼에 대해 살펴보고, 3장에서는 본 논문에서 구현하는 주소록 웹서비스를 위한 WSDL을 소개하고 웹서비스 클라이언트 개발에 대한 방법을 살펴본다. 4장에서는 StAX를 이용한 클라이언트의 개발에 대하여 소개한다. 마지막으로 5장에서 결론에 대하여 언급한다.

2. 관련연구

2.1 아파치 웹서비스 플랫폼 Axis

웹서비스를 제공하기 위한 엔진으로서 Sun사의 JWSPD[1]와 아파치에서 오픈소스로 제공되는 Axis[2]를 많이 사용하고 있다.



(그림 1) Axis 웹 서비스 툴 킷

Axis는 자바를 위한 웹서비스를 가능하게 하는 SOAP 엔진이다. Axis는 SOAP 메시지 처리 기능 이외에 웹서비스를 위한 코드 자동 생성 기능을 포함하고 있다. 그 중에서 WSDL2Java는 WSDL을 읽어 웹서비스가 가능한 Stub 및 바인딩 클래스 코드를 생성해 주는 툴이다[3].

stub클래스는 원격 서비스를 호출 코드이고 바인딩 클래스는 웹서비스를 위한 매개변수나 리턴 값의 값을 가질 수 있는 객체를 생성하는 역할을 한다.

2.2. Axis MIRAE

Axis MIRAE는 J2ME플랫폼에서 웹 서비스에 대한 SOAP기반 접근을 가능하게 해주는 JSR172(Java ME Web Services)의 구현체이다[6]. 아파치 Axis의 J2ME 버전으로 개발되었다가 새로 MIRAE라는 독립 프로젝트로 분리된 Axis MIRAE는 현재 베타 버

전 상태로 개발 진행 중이다. 본 연구에서는 이 시스템의 개발자인 이창신 씨의 지원을 통해 개발 중인 버전을 이용한 시스템 구축을 진행하였다[5,7].

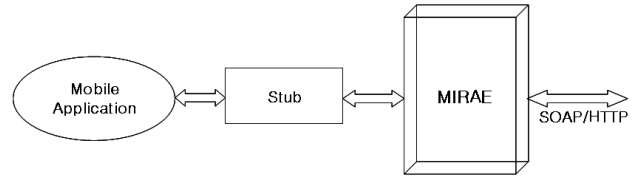


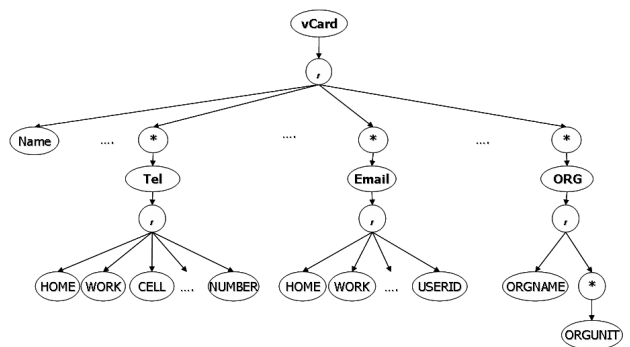
그림 2 Axis MIRAE의 구조

MIRAE는 크게 XML을 파싱하는 JAXP와 웹 서비스 클라이언트 개발 및 실행을 담당하는 JAX-RPC로 나눌 수 있다. 또한 SOAP 엔코딩 방식으로 doc/literal 만을 지원하므로 본 논문에서는 WSDL을 doc/literal방식으로 정의하였다.

3. WSDL 및 클라이언트 개발 방법

3.1 AddressBook WSDL

본 논문에서는 모바일 주소록 웹서비스를 개발하기 위하여 vCard 표준을 따르는 스키마를 정의하였다. vCard는 비즈니스 카드를 자동화한 개인 데이터 교환 수단의 표준으로 아웃룩 등 여러 프로그램에서 지원되고 있다. (그림 3)에서 vCard 타입을 간략하게 트리로 표현하였다.



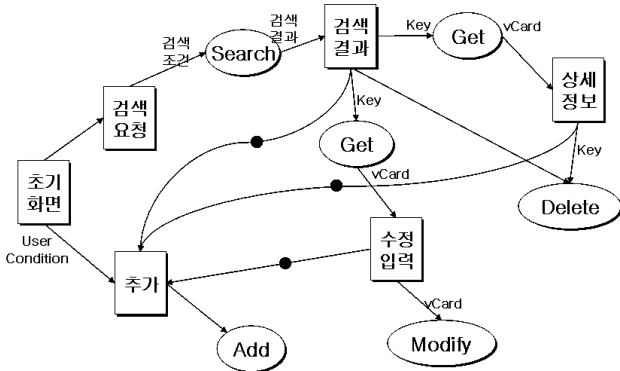
(그림 3) vCard Type

또한 주소록 웹서비스의 WSDL은 다음과 같은 연산을 정의한다. 주소록 검색을 위한 Search 연산은 검색 키워드를 매개변수로 받아 간략하게 정의된 vCard 타입의 리스트를 리턴한다. 검색의 결과로 받은 주소록의 상세정보를 보여주는 Get연산은

vCard의 id를 매개변수로 받아 id에 해당하는 vCard의 전체 정보를 리턴한다.

주소록 추가와 수정을 위한 연산은 Add와 Modify이고 각각 vCard 타입의 매개변수를 받아 처리 결과를 true/false로 리턴한다. 또한 주소록 삭제를 Delete 연산은 vCard의 id를 매개변수로 가지고 처리 결과를 true/false로 리턴한다.

아래 (그림 4)는 모바일 주소록의 연산의 흐름을 나타낸다.

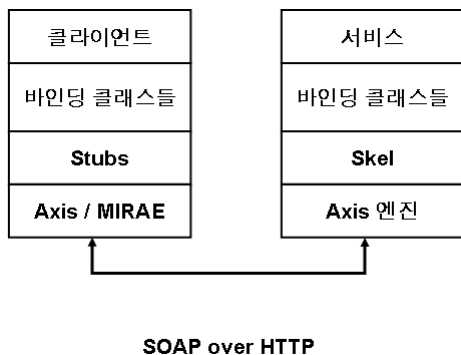


(그림 4) 모바일 주소록의 연산 흐름

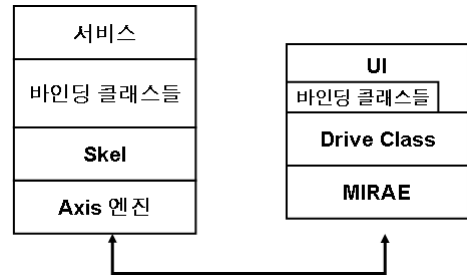
## 2.2 웹 서비스 클라이언트 개발 방법

### 2.2 웹 서비스 클라이언트 개발 방법

WSDL을 이용하여 웹서비스 클라이언트를 개발하는 방법으로는 WSDL2Java라는 툴을 이용하여 바인딩 클래스와 stub을 생성하여 stub을 통해 SOAP 메시지를 주고 받는 방법과 직접 SOAP 메시지를 생성하고 StAX파서를 통해 SOAP 메시지를 직접 파싱하는 방법이 있다.



(그림 5) Axis를 이용한 클라이언트 개발



(그림 6) StAX를 이용한 클라이언트 개발

WSDL2Java를 이용하여 바인딩 클래스를 생성할 때 모든 요소 정의에 대해 클래스가 생성된다. 주소록 WSDL의 경우 32개의 클래스가 생성됨을 확인할 수 있었다. 특히 Empty타입을 가지는 요소와 여러 요소들에서 공유하여 사용되는 요소에 대하여 모두 별도의 클래스가 생성된다. 따라서 바인딩 클래스와 stub을 자동 생성하는 방법은 제한된 자원을 가지는 모바일 환경에서는 바람직하지 않은 것으로 보인다.

## 4. StAX를 이용한 클라이언트 구현

StAX는 자바 기반 XML pull parser의 표준 API로서[8] 이벤트를 받아서 처리하는 SAX와는 달리 어플리케이션에 의해 요청되었을 때 이벤트를 반환한다. 본 논문에서는 MIRAE에서 제공하는 StAX의 구현을 이용하여 요청 SOAP코드를 생성하고 응답 SOAP코드를 처리하게 된다.

StAX를 이용한 XML문서를 읽고 쓰기 위한 작업은 대부분 XMLStreamReader와 XMLStreamWriter 인터페이스 객체를 통하여 이루어진다. 이들 객체들은 스트리밍으로 이동되는 커서를 가지고 있다. MIRAE에서는 Dispatcher 클래스에서 웹서비스와의 연결을 가지고 XMLStreamReader와 XMLStreamWriter 객체를 생성해주는 역할을 한다.

다음 예는 웹서비스의 연결과 XMLStreamWriter과 XMLStreamReader객체를 생성해 주는 코드이다.

```
dispatch = new Dispatch(uri,"",null,null);
requestWriter = dispatch.getRequestWriter();
...
requestReader = dispatch.getRequestReader();
...
```

XMLStreamWriter객체는 writeStartElement 메소드와 writeEndElement 메소드를 가지고 XML이 가지는 요소의 태그를 써준다. writeNamespace로 요소가 가지는 namespace를 지정해준다. 터미널 요소 노드의 경우에는 writeCharacters를 통하여 노드의 값을 지정할 수 있다.

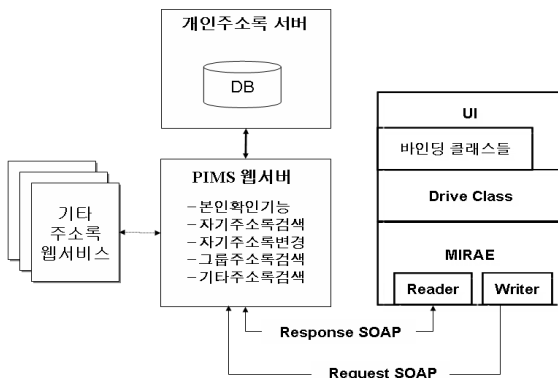
```
requestWriter.writeStartElement(elemName);
requestWriter.writeNamespace("tns",uri);
requestWriter.writeCharacters(value);
requestWriter.writeEndElement();
```

XMLStreamReader는 require를 통하여 태그들을 파악할 수 있다. 또한 getElementText를 통하여 요소가 가지는 값을 읽을 수 있다.

```
requestReader.require(XMLStreamConstants.START_ELEMENT, namespace, elemName);
requestReader.getElementText();
requestReader.require(XMLStreamConstants.END_ELEMENT, "", elemName);
requestReader.nextTag();
```

주소록 웹서비스 클라이언트에서는 웹서비스의 결과로 전달받은 주소록 데이터나 검색 결과 등을 사용자 정의 바인딩 객체에 저장한다. 이 객체는 데이터를 저장하고 전달하는 역할만 하는 가벼운 클래스이다.

아래 (그림 7)은 사용자 정의 바인딩 클래스와 StAX 드라이버를 사용하는 클라이언트 시스템의 전체 구조를 보여준다.



(그림 7) 모바일 주소록 웹서비스

## 5. 결론

본 논문에서는 모바일 웹서비스의 어플리케이션의 사례로 StAX를 이용하여 SOAP메시지를 직접 생성하고 처리하는 주소록 웹서비스 클라이언트를 개발하였다. 개발된 주소록 웹서비스 클라이언트는 모바일 클라이언트는 모바일과 같이 제한된 환경에서도 효과적이고 효율적으로 고성능의 SOAP의 처리를 제공할 수 있다.

## 참고문헌

- [1] JWSDP(Java Web Services Developer Pack), <http://java.sun.com/webservices/jwsdp/index.jsp>
- [2] apache axis, <http://ws.apache.org/axis/>
- [3] axis user-guide, <http://ws.apache.org/axis/java/user-guide.html>
- [4] MIRAE, <http://ws.apache.org/mirae/index.html>
- [5] MIRAE Source Code, <https://svn.apache.org/repos/asf/webservices/mirae>
- [6] jsr172, [http://java.sun.com/j2me/docs/j2me\\_jsr172.pdf](http://java.sun.com/j2me/docs/j2me_jsr172.pdf)
- [7] 이창신, J2ME 웹 서비스에서 JAX-RPC, [http://www.imaso.co.kr/?doc=bbs/gnuboard.php&bo\\_table=article&wr\\_id=5329](http://www.imaso.co.kr/?doc=bbs/gnuboard.php&bo_table=article&wr_id=5329), 마이크로소프트웨어, 2004. 9.
- [8] StAX, <http://stax.codehaus.org>
- [9] WSDL, [www.w3.org/TR/wsdl](http://www.w3.org/TR/wsdl)
- [10] JAXB, <http://java.sun.com/webservices/jaxb/index.jsp>
- [11] vCard, <http://jabber.org/jeps/jep-0054.html>
- [12] Mobile device optimization <http://www-128.ibm.com/developerworks/wireless/library/wi-devo/>
- [13] Which style of WSDL should I use? <http://www-128.ibm.com/developerworks/webservices/library/ws-whichwsdl/>