

XML-RPC를 이용한 문서교환 및 제어용

미들웨어 구조 연구

임정은⁰, 윤용익
숙명여자대학교 컴퓨터과 학과
{jungeun, yiyoon}@sookmyung.ac.kr

A Study of Middleware architecture form

document exchange and management based on XML-RPC

Jung-Eun Lim⁰ and Yong-Ik Yoon
Dept. of Computer Science, Sookmyung Women's University

요 약

표준 데이터 포맷을 이용하여 인터넷상에 데이터를 전달하기 위해 XML 기술을 활용하여 분산 환경을 어떻게 더욱 발전된 모습으로 지원해나갈 수 있을 것인지를 두고 다양한 시도가 이루어지고 있다. 그 중에서도 가장 중심적인 역할을 수행할 수 있는 분야는 정보를 표현하여 인터넷을 활용한 모든 곳에서 통합화/표준화하여 사용자들의 정보공유와 경제활동에 접근 가능하도록 하는 것이다. 분산 객체 기술의 하나인 RPC와 XML을 연동한 XML-RPC는 HTTP를 확장해 XML 메시지를 주고 받을 수 있는 프로토콜로 HTTP를 사용하고 호출요청 및 응답 인코딩에 XML을 사용하는 메커니즘을 통해 클라이언트 서버간의 상호작용을 한다. 본 논문에서는 현재 분산객체기술과 XML과의 연동에 관한 연구를 알아보고, XML-RPC를 이용한 XML문서교환 및 제어용 미들웨어 구조를 제안한다.

1. 서 론

WWW은 현재 우리 생활에 가장 큰 영역으로 자리 잡으면서 비즈니스 영역으로 활용분야를 넓혀가고 있다. 현재 많은 프로그램이 HTTP를 기반 전송 규약으로 사용하는 웹 응용 프로그램으로 구축되고 있으며, 웹 브라우저를 통해서 임의의 플랫폼에서도 응용 프로그램에 접근하여 정보공유와 경제활동을 할 수 있다. 이러한 흐름 속에서 XML[1]기술을 활용하여 응용 프로그램간 통신 수단으로 웹을 이용하도록 분산 환경을 어떻게 더욱 발전된 모습으로 지원해 나갈 수 있을 것인지를 두고 다양한 시도가 이루어지고 있다. 분산 환경에서 XML기술을 활용하여 가장 중심적인 역할을 수행 할 수 있는 분야는 데이터를 표현하여 모든 곳에서 통합화/표준화된 채널을 통하여 모든 사용자들이 접근 가능하도록 하는 기반 구조이다.

최근 널리 사용되고 있는 분산기술인 Microsoft의 COM/DCOM, OMG의 CORBA[2], JAVA Bean등은 그 기술들간 서로 호환되지 않으며, 객체기반 분산컴퓨팅 환경 하에서의 클라이언트는 ORB(Object Request Broker) 소프트웨어에 의존적으로 서버와 데이터 교환을 하기 때문에 CORBA, DCOM, Java RMI등의 ORB등은 상호운영성의 제약과 방화벽에 의한 메시지 차단이 되며, 원격 프로시저를 호출하지 않고 웹과의 연동시에 원격지 응용 프로그램과의 상호작용이 되지 않아 서로 통신을 할 수 있는 프로토콜 표준이 필요하다. 현재 XML이라는 표준 데이터 포맷을 이용하여 인터넷상에 데이터

를 전달하기 위하여 IIOP나 RPC(원격 프로시저 호출, Remote Procedure Call)를 사용하고 있다. 이러한 IIOP나 RPC는 통신 대역폭이나 계산능력 측면에서 효율적이다. XML은 Publishing과 Data Exchange라는 두 가지 주요한 Application 영역을 갖는다. XML프로토콜로 XML-RPC[3][4], SOAP[5], WDDX등이 있으며 이는 응용 프로그램간 데이터 교환을 지원하고, 단순히 데이터 교환뿐 아니라 그에 적합한 처리를 할 수 있는 응용 프로그램까지 전달 할 수 있다.

본 논문의 2장에서는 분산환경과 XML과의 연동기술에 대하여 살펴보고, 3장에서는 XML Protocol의 종류인 XML-RPC를 이용한 메시지 교환구조 및, Middleware의 Presentation기능인 XML Server를 이용한 XML문서의 Publishing에 대해서 알아보고, 4장에서는 결론 및 향후 연구과제에 대하여 기술한다.

2 분산환경과 XML 연동기술

분산기술은 웹과의 호환성을 직접 가지지 못하므로 웹을 통해 데이터 교환에 사용하려는 어플리케이션에서는 별도의 프로토콜을 지원하도록 요구받는다. 이러한 컴포넌트 기술과 웹과의 호환성 문제를 해결하기 위해서 XML을 이용하고자 하는 많은 노력이 있어왔으며, XML은 기존의 복잡했던 분산 환경을 상당 부분 단순화시키는데 큰 역할을 수행할 수 있었다.

2.1 CORBA와 XML

최근 분산 컴포넌트 기술인 OMG(Object Manageme

nt Group)의 CORBA(Common Object Request Broker Architecture)와 웹과의 연동을 위해 SOAP(Simple Object Access Protocol)을 이용한 SCOAP(Simple Corba Object Access Protocol)[6]이 제안되어 있는 상태이다. SOAP(Simple Object Access Protocol)이라는 프로토콜은 Microsoft사에 의해 제안되었으며, 플랫폼에 독립적으로 인터넷에 분산된 서비스, 오브젝트, 또는 서버에 접근할 수 있도록 HTTP 프로토콜을 지원하는 XML프로토콜이다. SOAP에서는 HTTP 혹은 SMTP 상에서 단순히 서비스 요청/응답에 대하여 표준 XML문서 구조를 정의하여 서비스 제공업자와 사용자 간에 상호 교환하도록 구성되어 있다. 서비스를 요청하는 문서에는 사용할 서비스의 정보와 입력값을 채워서 보내주면 서비스 제공업체는 이를 분석하여 해당 서비스를 수행한 후 그 결과값을 응답 문서에 넣어서 돌려주는 방식이다. 현재 SOAP의 표준화가 되어있지 않은 상태이다.[7]

2.2 WDDX

WDDX(Web Distributed Data eXchange)는 프로그래밍 언어 사이에서의 복잡한 데이터 구조를 교환하기 위하여 고안된 XML 프로토콜의 한 종류로서 XML을 이용하여 데이터 직렬화(Serialization)를 통하여 데이터를 전달한다. 직렬화된 메시지는 데이터에 대한 타입 정보와, 변수 이름, 변수 값을 포함한다. 그러나, WDDX는 데이터 직렬화에 대한 방법만 정의되어 있고 remote 메소드 호출의 기능이 없으며, XP(XML Protocol)바인딩이 HTTP로 한정되어 있고, 지원하는 데이터 타입 또한 제한되어 있다.

2.3 XML-RPC

XML-RPC(XML-Remote Procedure Call)는 XML 프로토콜의 한 종류로 XML을 이용하여 원격지 머신에 대한 함수호출을 캡슐화한 후 원격지 머신에 XML문서를 전달하고 그 결과값으로 XML문서를 돌려받는다. 전통적으로 RPC(원격 프로시저 호출, Remote Procedure Call)를 사용하는데 가장 큰 방해물은 RPC의 인코딩이었다. XML-RPC에서는 전송은 HTTP로하고 인코딩은 XML을 이용한다. 새로운 웹 표준으로 자리잡은 XML은 매우 간단한 데이터를 텍스트로 표현할 수 있도록 할 뿐만 아니라 데이터 구조의 표준을 제공한다.

```

POST /RPC2 HTTP/1.0
User-Agent: Frontier/5.1.2 (WinNT)
Host: betty.userland.com
Content-Type: text/xml
Content-length: 181

<?xml version="1.0"?>
<methodCall>

<methodName>examples.getStateName</methodName>
  <params>
    <param>
      <value><i4>41</i4></value>
    </param>
  </params>
</methodCall>
    
```

그림 1 XML-RPC Request example

그림1은 XML-RPC에서 Request하는 예를 보여준다. DOM과 SAX는 XML에 접근할 수 있는 API를 제공했고, RPC라이브러리를 쉽게 구현할 수 있도록 하였다. XML을 인코딩으로서 사용할 경우 매우 빠르고 가벼우므로, XML-RPC라이브러리에 대해서 좋은 성능을 발휘하며, XML-RPC는 원격 프로시저 호출에서 다른 원격지 서버에 안정적으로 접근할 수 있는 API 모델에 적당하다.

3. XML-RPC를 이용한 문서교환 및 제어용 미들웨어 구조연구

이번 장에서는 본 논문에서 제안하는 XML-RPC를 이용한 XML문서의 교환에 대하여 기술한다. Client가 Server에 데이터를 요청하면 Server는 다른 웹서버상에 있는 필요한 XML DTD를 선별하여 Client에게 원하는 데이터 내용을 보여주기 위해 XML데이터를 사용하기 위한 분산객체기술인 XML-RPC를 이용한다.

분산객체지향시스템에서 데이터를 가져오기 위해서 HTTP프로토콜을 사용하는 XML-RPC를 이용하였는데, 실제로 XML-RPC라이브러리는 클라이언트에서 서버로 보내진 Request를 받아서 디코딩하고, 그 Request를 핸들러 메소드에 연결한다. 그리고 나서 그 요청된 자바 메소드를 호출하고, 서버는 그 메소드의 결과를 다시 XML로 인코딩한다.

본 논문에서 제시하는 4가지 시스템은 Remote Call 시스템, Gethering시스템, Scheduling시스템, XML-DB Mapping시스템(XML2DB, DB2XML Mapping시스템)이다. 그림 2는 XML-RPC[14,15]를 사용한 문서교환 및 제어용 미들웨어의 연동구조를 보여준다. [10,11,12]

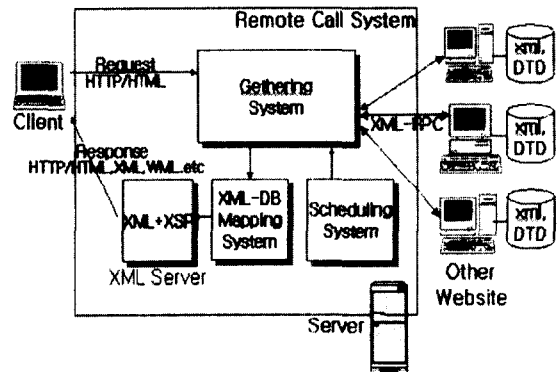


그림 2 XML-RPC를 이용한 문서교환 및 제어용 미들웨어 연동구조

Remote Call 시스템은 나머지 전체시스템을 포괄하는 시스템이다. Gethering시스템은 XML-RPC를 사용하여 외부의 데이터를 가져오는 시스템이고, Scheduling시스템은 네트워크 두절과 같은 외부시스템에서 데이터를 가져올 수 없을 때 필요한 함수명, 클래스명을 정기적으로

가져와 정보Table에 저장하기 위해서 매 주기마다 정보를 가져오도록 한다. Scheduling시스템에 의해 정기적으로 가져오는 데이터는 가공을 통해 XML-DB시스템에 저장된다. Gethering시스템에서 XML-RPC를 이용해 Gethering된 외부 시스템의 XML문서는 XML Parser를 거쳐서 DOM, SAX대신에 JDOM을 사용하여 원하는 XML문서를 추출한다.

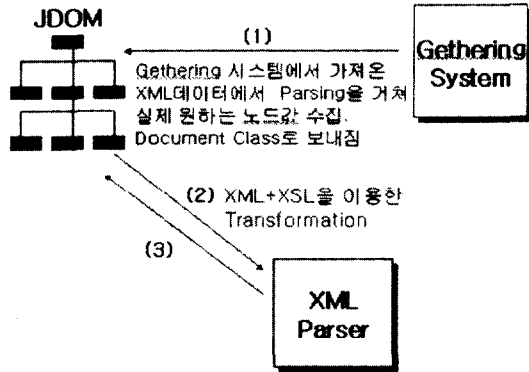


그림 3 JDOM을 사용한 XML 노드추출

그림 3은 JDOM을 사용하여 XML문서를 추출하는 과정이다. JDOM은 자바환경에서 개발자들이 보다 직관적이고 간단하게 XML문서를 표현하고자 만들어진 오픈소스 API로 DOM[9]과 SAX같은 기존의 표준 API위에 추상화를 실현한 것이다. DOM과 SAX에 비해 데이터를 효율적으로 reading, manipulation, writing하고, 쉽게 XML문서를 표현할 수 있는 방법이며, DOM에 비해 메모리 사용량이 적고, DOM이나 SAX로 구성된 소스에서 데이터를 추출할 수 있고, DOM이나 SAX형태로 결과를 줄 수가 있다. JDOM을 이용해서 XML문서의 원하는 부분이 추출되면 파서를 사용해서 XML+XSLT를 적용한 원하는 문서를 만든다. 만들어진 각각의 문서는 JDOM을 사용하여 실제로 우리가 원하는 정보를 다른 최종 XML 문서로 통합되어지며, XML엘리먼트와 DB테이블 Column매핑을 위한 XML-DB Mapping System에 저장된다.

Client가 요청한 XML데이터가 생성되면, Middleware의 Presentation 기능인 XML Server를 사용하여 클라이언트가 가지고 있는 다양한 브라우저에 의존적이지 않도록 XML+XSP[13]를 사용하여 서버쪽에서 클라이언트 쪽 브라우저에 Publishing한다. XSP(eXtensible Server Pages)는 정적인 XML문서를 지양하는 대신 XSP를 이용해서 동적인 XML문서를 만들어준다. 이것은 동적인 XML문서를 기반으로 만들어진 웹 퍼블리싱을 위한 XML Server의 기술중의 하나이다.

4. 결론 및 향후 연구과제

XML은 분산환경에서 서로 다른 응용들의 통신수단으로 지원되며 기존의 복잡했던 분산 환경을 상당 부분 단

순화시키는데 큰 역할을 수행할 수 있다. XML-RPC는 전송은 HTTP를 이용하고 인코딩은 XML을 사용하여 분산환경에서 메소드의 매개변수와 반환 타입이 XML-RPC에서 지원하는 인코딩 가능한 것이라면 XML-RPC로 그 메소드를 호출할 수 있으므로 분산된 시스템에 있는 클라이언트가 네트워크의 또 다른 부분에 있는 서버에서 실행될 수 있는 작업을 요청할 수 있다. 분산기술은 웹과의 호환성을 직접 가지지 못하므로 XML Protocol중에서 XML-RPC를 사용하여 이러한 컴포넌트 기술과 웹과의 호환성 문제를 해결하기 위한 연구구조를 살펴보았다. 앞으로의 연구과제는 메시지 암호화를 통해 안전한 메시지 교환이 이루어지도록 보안쪽에 연구가 필요하며, 서버부하의 분산을 위한 Loadbalancing과 서버부하과다에 의한 서버의 다운시에 세션 정보를 유지하기 위한 다중서버의 클러스터링지원을 위해 Server-side caching이 필요하겠다.

[참고문헌]

- [1] XML(eXtensible Markup Language), W3C. "http://www.w3c.org/XML"
- [2] CORBA Specification, Ver 2.4.2, OMG "http://www.omg.org"
- [3] W3C, "XML Protocol Working Group Chapter", "http://www.w3c.org/2000/09/XML-Protocol-Chapter."
- [4] Dave Winer, UserLand Software Inc, "XML-RPC Specification", "http://www.xml-rpc.com/spec"
- [5] W3C, SOAP version 1.2 Working Draft "http://www.w3.org/TR/2001/WD-soap12-20010709/"
- [6] SOAP(Simple Corba Object AccessProtocol, OMG, "ftp://ftp.omg.org/pub/docs/orbos/00-09-03.pdf"
- [7] George M. Doss, Wordware Publishig, Inc. "CORBA Developer's Guide with XML"
- [8] JDOM "http://www.jdom.org" "http://www.newInstance.com"
- [9] W3C, DOM(Document Object Model) "http://www.w3c.org/DOM"
- [10] Hiroshi Maruyama, Kent Tamura, Naohiko Uramoto, Addison-Wesley, 2000, "XML and Java Developing Web Applications"
- [11] Frank Boumphrey and Etc, 1999, "Professional XML APPLICATIONS"
- [12] Breck McLaughlin, O'REILLY. 1999, "JAVA AND XML"
- [13] XSP(eXtensible Server Pages) "http://xml.apache.org/cocoon"
- [14] XML-RPC HOWTO "http://xmlrpc-c.sourceforge.net/xmlrpc-howto/xmlrpc-howto.html"
- [15] XML-RPC Library for Java "http://classic.helma.at/hannes/xmlrpc"
- [16] XML-RPC and SOAP projects for Python. "http://python.scripting.com/directory/13/xmlrpc"