

XML 기반에서 SOAP을 이용한 메시지 교환 방법 개선에 관한 연구

김주영, 최아람*, 장근, 이병수
인천대학교 컴퓨터공학과
e-mail : openarm@incheon.ac.kr

A Study on Improving Method for Message Switching using SOAP based XML

Joo-Young Kim, Ah-Rham Choi*, Keun Chang, Byoung-Soo Lee
Dept. of Computer Engineering, University of Incheon

요 약

다양한 어플리케이션 간의 분산 객체를 사용하고자 할 때 지금까지는 DCOM/CORBA bridge를 사용하여 문제를 해결해 왔다. 그러나 이러한 분산 객체 프로토콜을 인트라넷이 아닌 인터넷에서 사용할 때는 임의로 할당받은 프로토콜을 사용하여 상호접근하게 되므로 보안상의 문제를 갖게 된다. 따라서 이러한 문제를 해결하기 위하여 방화벽을 설치한다. 하지만 여러 조건을 바탕으로 들어오는 트래픽을 블로킹함으로써 효과적인 분산 객체 프로토콜을 사용하는데 또한 문제를 갖게 된다. 본 논문에서는 기존의 인터넷 인프라스트럭처에 그대로 사용할 수 있고, 이기종 환경에 관계없이 메시지를 교환할 수 있는 SOAP(Simple Object Access Protocol)에 대한 연구와 그 적용 사례로 인터넷 서점을 구현하여 제시하였다.

1. 서론

현재 분산된 객체에 대하여 데이터를 전송하는 기술은 특정 포트 사용을 요구한다. 많은 기업의 방화벽들은 보안 문제로 포트를 열어 두는 것을 우려해 HTTP가 사용하는 디폴트 포트 80을 제외한 다른 포트에 대한 사용 금지를 권장하고 있다.

분산 컴퓨팅을 구현하기 위해서는 객체들이 물리적으로 다른 위치에 있는 객체들을 호출할 수 있어야 하는데 이러한 방식 중 가장 전통적이고 범용적인 방식이 RPC(Remote Procedure Call)이다. 이러한 RPC를 이용한 DCOM(Distributed Component Object Model), CORBA(Common Object Request Architecture), RMI(Remote Method Invocation)는 각각 지원하는 플랫폼이 다르기 때문에 이기종 구성요소 간의 호출이 어렵고, 또한 구축하는데 필요한 인력과 시간 면에서 큰 비용을 요구한다. 이러한 분산 객체 기술을 인터넷에 그대로 적용할 때에는 임의로 할당받은 프로토콜을

사용하여 상호접근하게 되므로 보안상 취약점을 가지게 된다. 이러한 문제 해결을 위해 대부분의 조직은 방화벽을 설치한다. 방화벽은 내부 및 외부 상호 접속된 네트워크에 대한 트래픽을 감시, 기록함으로써 네트워크 보안을 증가시킨다. 하지만 허용치 않은 모든 포트 번호를 블로킹함으로써 효과적으로 분산 객체 프로토콜을 사용하는데 어려움을 준다. 따라서 본 논문에서는 기존의 인터넷 인프라스트럭처를 사용하여 응용 프로그램이 방화벽에 의해 뜻하지 않게 블로킹되는 일없이 다른 응용 프로그램과 직접 통신할 수 있도록 하고, HTTP와 XML을 기반으로 개발된 새로운 웹 기반 메시징 포맷으로 서로 다른 컴퓨터 시스템을 보유한 기업이 인터넷을 통해 자유롭게 메시지 교환할 수 있는 SOAP에 대한 연구를 하였으며, 그 적용 사례로써 인터넷 서점을 구현하였다.

본 논문의 구성은 다음과 같이 하였다. 2 장에서는 HTTP를 통한 RPC를 설명하였고, 3 장에서는 SOAP에서의 XML과 메시지 구조를, 4 장에서는 인터넷 서점의 구현 모델을 설명하였으며, 마지막 5 장에서 결론 및 향후 연구 과제를 제시하였다.

* 본 논문은 한국과학재단 지정 인천대학교 멀티미디어 연구센터 지원에 의한 것임

2. HTTP를 통한 RPC

클라이언트와 서버의 상호 통신은 미리 정해진 프로토콜 HTTP(Hypertext Transfer Protocol)를 통해 이루어진다. HTTP는 실시간으로 클라이언트와 서버 사이에서 통신을 하고, 이 프로토콜을 통해 전달할 수 있는 자료가 제한되어 있다는 단순함과 안정성때문에 어디서나 사용할 수 있다는 특징을 갖는다.

HTTP가 방화벽을 통과할 수 있다는 특징을 이용하여 원격지 응용 프로그램과 상호작용할 수 있을 것이라는 가정을 하게 되었고, 그 결과 HTTP를 통한 간단하고 이식성 높은 원격 프로시저 호출 방법으로 XML RPC가 정의되었다. 현재 XML RPC에 대한 다양한 연구가 진행 중에 있으며 그 중 하나로 SOAP을 들 수 있다. SOAP의 최초 공개 릴리즈는 XML RPC를 기본으로 하고 있다.

SOAP은 포트 80을 할당받은 HTTP의 맨 위에 분산 객체 프로토콜을 올려 전달한다. HTTP는 GET, POST, PUT 등의 Request 메시드가 있다. SOAP은 XML을 사용하여 요구 및 응답 메시지 형식을 정의한 후 HTTP POST를 사용하여 전달한다. 그러나 서버에 대한 첫 번째 요청으로 M-POST를 사용할 것을 요구한다. M-POST는 HTTP Extension Framework 규격을 바탕으로 정의된 새로운 HTTP 요청 메시드로 M-POST를 사용한 호출이 실패하면, 클라이언트는 POST를 사용하여 호출을 재시도해야 한다. POST를 통해 전송할 수 없는 HTTP 헤더를 확장된 M-POST를 사용하게 함으로써 HTTP 프로토콜에 정확하게 헤더를 추가하고, SOAP 사용자에게 더 많은 융통성을 제공한다.

2-1. HTTP Request

HTTP 프로토콜 상에서 브라우저가 어떤 데이터 요구를 하기 위해 보내는 HTTP 메시지이다.

```

GET/default.htm/HTTP/1.1
Accept: text/*
User-Agent: Mozilla/4.0
Host: www.mcp.com:8080
{CR}{LF}
    
```

(그림 1) HTTP Request

2-2. HTTP Response

HTTP 프로토콜 상에서 서버가 수신한 요구에 대해 처리한 결과를 브라우저에게 응답으로 보내 주는 HTTP 메시지이다.

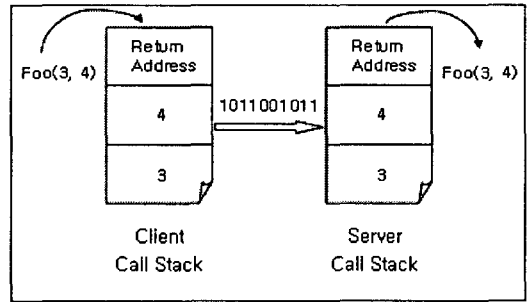
```

HTTP/1.1 200 OK
Content-Type: text/html
{CR}{LF}
<html>
<head><title>default.htm page</title></head>
<body>HTTP is good</body>
</html>
    
```

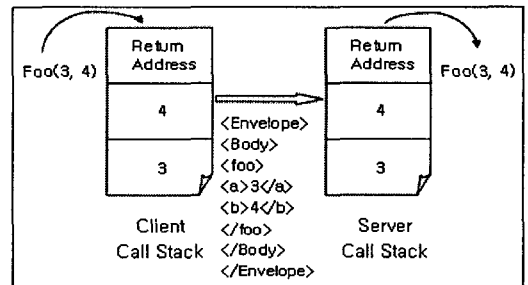
(그림 2) HTTP Response

3. SOAP에서의 XML과 메시지 구조

SOAP 메시지를 기술하기 위하여 특정 바이너리 프로토콜을 사용하지 않고, XML이라는 웹 표준을 사용한다. 다음 (그림 3)과 (그림 4)는 각각 메서드 호출을 위한 기존의 바이너리 프로토콜을 사용하는 방식과 XML을 사용하는 방식을 나타낸다.

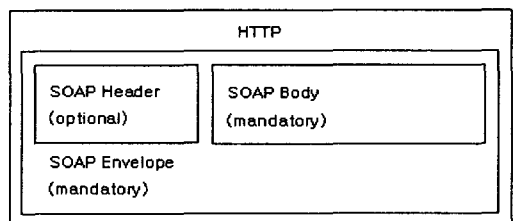


(그림 3) Remoting method call stacks



(그림 4) Remoting SOAP method calls

SOAP 클라이언트는 호출할 메서드와 파라미터를 XML로 기술하여 보내고, 받는 쪽 SOAP 서버에서는 이러한 XML을 파싱하여 필요한 내용을 추출하고, 로직을 수행한 후에 답을 전달한다. 메시지 안에 포함되는 모든 요소와 속성의 정의는 XML 규격에 따라야 한다. XML 메시지 형식은 XML 표준에 기초하여 표준화되기 때문에 서비스 호출 및 응답에 XML 문서를 사용하는 SOAP은 언어 및 운영체제 간에 통신하는데 제약을 받을 필요가 없다. SOAP 메시지는 항상 SOAP Envelope와 Body를 포함하고, SOAP Header는 선택사항이다. 이런 요소들 각각은 XML namespace로 첫 머리에 놓이게 된다. 다음 (그림 5)는 SOAP 메시지 구조를 나타낸다.



(그림 5) SOAP 메시지 구조

3-1. SOAP Envelope

SOAP은 HTTP 맨 윗부분인 Header에 분산 객체 프로토콜을 올려 전달한다. XML을 사용하여 요구 및 응답 메시지 형식을 정의하고, HTTP BODY부분에 자리를 마련하였다. 이것은 메시지를 표현하는 상위 XML 요소로서 SOAP Envelope라 한다. 이 Envelope에 원격지의 서비스를 사용하기 위한 정보를 담아 전달한다. 웹 서버는 이 Envelope를 HTTP 요청 내용으로 간주하고 SOAP 서버에게 보내고, SOAP 서버는 XML 파서로 받은 메시지 형식을 분석하여 필요 부분을 추출한다.

3-2. SOAP Header

SOAP 메시지에 포함시키거나 포함시키지 않아도 되는 선택사항이다. Header는 클라이언트 상황을 더욱 세세하게 전달하기 위하여 기술한다.

□ SOAP-ENV : mustUnderstand

이 요소는 0 혹은 1 값을 가진다. 만일 이 요소가 1 값을 가질 경우에는 이 요소를 처리하는 SOAP 구현은 반드시 전달된 대로 이해해야 하며 이해하지 못한 경우에는 오류를 반환해야 한다. mustUnderstand 요소가 없거나 0의 값을 갖는 경우에는 무시해도 된다.

□ SOAP-ENV : actor

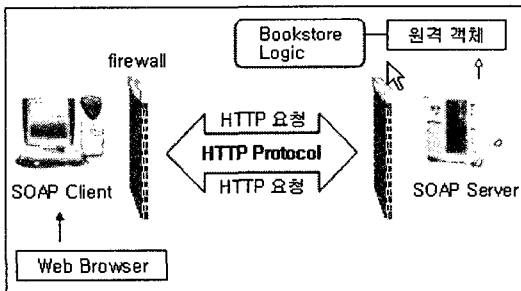
이 요소는 수신자를 지정하는 역할을 한다.

3-3. SOAP Body

SOAP Body는 메시지를 수신하는 SOAP 서버에게 전달될 실제 내용을 기술한다. Body는 호출할 메서드와 파라미터를 담거나 예러와 서비스 요청 결과를 표시하기 위한 수단으로 사용된다.

4. 인터넷 서점 구현 모델

본 논문에서는 고객이 SOAP을 통해 주문을 할 수 있는 인터넷 서점을 구현하였다. 다음 (그림 6)는 SOAP을 사용해서 인터넷 서점을 구현한 흐름을 보여 준다.



(그림 6) SOAP을 사용한 인터넷 서점 흐름도

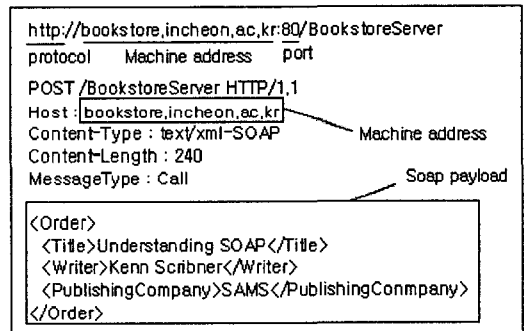
먼저 SOAP 클라이언트로 작동하는 프로그램은 온라인 주문을 하기 위해 필요한 정보, 즉 책 제목과 저서를 XML 메시지로 기술하여 SOAP Envelope에 담는다. 이것은 HTTP 프로토콜을 사용하여 SOAP 서버에

게 전달하며 SOAP 서버는 전달받은 것을 받아 XML 문서로 변환하여 필요한 정보를 추출하고 인터넷 서점 로직을 가진 객체로 전달한다. 객체는 전달된 정보를 이용하여 인터넷 서점 로직을 수행하여 주문에 성공했는지를 알려 주는 boolean 값과 배달될 때까지 걸리는 시간을 다시 SOAP 서버로 전달한다. SOAP 서버는 받은 결과를 XML 메시지로 변환하여 SOAP Envelope에 담는다. 이것 역시 HTTP 프로토콜을 사용하여 SOAP 클라이언트에게 전달한다. SOAP 클라이언트는 전달받은 응답을 해석하고, 웹브라우저 갱신을 통해 사용자에게 결과를 보여 준다.

사용자가 인터넷 서점에서 주문을 하기 위해서는 다음과 같은 메서드를 사용한다.

Boolean Order([in] Title string, [in] Writer string, [in] PublishingCompany string, [out] Delivery integer);

이 메서드를 호출할 때는 사용자가 원하는 책 제목과 저자, 출판사를 문자열로 표현하여 전달한다.



(그림 7) SOAP Request

이 요청은 HTTP 프로토콜을 사용하여 위의 (그림 7)과 같은 요청 형식을 통한다. (그림 7)을 살펴보면, SOAP은 표준인 HTTP POST Request 메서드를 사용하여 전달하는 것을 알 수 있다. 이 요청은 bookstore.incheon.ac.kr에 있는 BookstoreServer 서버로 전송되고, 메시지 컨텐트 타입은 XML 코드로 작성한 메서드 호출이 포함된 HTTP 메시지의 Body를 지정하는데 사용되도록 새로 정의한 text/xml-SOAP을 사용한다. 메시지의 길이는 240 바이트이고, MessageType은 SOAP 요청을 포함하고 있다는 것을 나타낸다. 이 요청 메시지에는 서버로 서비스 호출 요청을 전달하는데 필요한 모든 정보가 포함되어 있다.

bookstore.incheon.ac.kr 서버는 위의 요청 형식을 받아 처리한 후에, (그림 8)과 같은 응답을 한다.

```

HTTP/1.1 200 OK
Connection : close
Content-Type : text/xml-SOAP
Content-Length : 177
MessageType : CallResponse
<SerializedStream>
  <OrderResponse>
    <_return>1</_return>
    <Delivery>7</Delivery>
  </OrderResponse>
</SerializedStream>
    
```

(그림 8) SOAP Response

이 응답 메시지는 전과 마찬가지로 표준 HTTP POST 를 사용하고, 응답 코드 200 을 포함한다. Connection 은 TCP 커넥션이 닫혀야 한다는 것을 나타낸다. MessageType 는 Body 가 호출의 결과를 담고 있음을 나타내는 CallResponse 값을 갖는다. 성공적인 SOAP 응답은 Response 앞에 원래의 메서드 이름, 즉 Order 메서드 요소안에 응답 데이터를 포함한다.

5. 결론

본 논문에서는 자료에 접근하기 위하여 SOAP 을 사용해 인터넷 서점을 구현하여 제시하였다.

지금까지 RPC 모델을 보다 효율적으로 사용하기 위한 수많은 분산 객체 프로토콜이 사용되어 왔으나 현재의 인터넷 인프라스트럭처에 그대로 사용되지 못하는 문제점을 가지고 있다. 그러나 SOAP 을 사용하면 현재의 인프라스트럭처를 변경하지 않고 원격 메서드 호출이 가능하다. 기존의 RPC 방식과는 달리 메서드를 호출하는데 특정 프로토콜을 사용하지 않고 XML 표준에 기본을 두고 있어 비호환성의 문제를 해결하였다. 그러나 HTTP 를 사용하는데 있어서의 전송 부분과 XML 구문 분석, XSD 스키마 처리 부분으로 인하여 상호 운용성 문제가 발생할 수 있다. 또한 제한된 XML 언어가 너무 많아 이것을 체로 거르는 작업은 프로그래밍 언어에 대한 공식적인 설명에 대한 해설에 익숙하지 않은 사람들에게는 힘든 부분이 될 수 있다. 이러한 문제를 해결하기 위하여 일부 XML 표준 그룹들이 좀 더 나은 문서와 코드를 적극적으로 발표하고 있다.

참고문헌

[1] Extensible Markup Language(XML) 1.0,
<http://www.w3.org/TR/1998/REC-xml-19980210>
 [2] HTTP Extension Framework,
<http://www.w3.org/Protocols/HTTP/ietf-http-ext>
 [3] Jung Min Choi, Ji Sun Lee, Byoung Soo Lee, "A Study on Effective Solving Techniques for RPC Problems in Distributed Computing Environment", EALPIIT 2002, 2002
 [4] Kenn Scribner, Mark C.Stiver, "Understanding SOAP", 2000 SAMS
 [5] SCOTT SEELY, "SOAP | Cross Platform Web Service Development Using XML", 2002 Prentice Hall PTR
 [6] Simple Object Access Protocol(SOAP),

http://www.microsoft.com/korea/msdn/workshop/xml/general/SOAP_White_Paper.asp
 [7] Simple Object Access Protocol(SOAP) 1.1,
<http://www.w3.org/TR/SOAP/>
 [8] SOAP Toolkit 2.0 Documentation,
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/soap/htm/kit_intro_19bj.asp
 [9] SOAP Web Services, <http://www.soap-services.com>
 [10] Weon Joon Kang, Hong Seong Park, "Design and Implementation of the Internet-GEM Using XML-SOAP", ChuncheonMultimedia Conference, 2001
 [11] 웹 서비스 보안,
<http://www.microsoft.com/korea/msdn/vstudio/nextgen/technology/security.asp>