

# JMS를 이용한 XML문서교환 RPC 모델 구현방법에 관한 연구

정현\*, 박해우\*\*, 강병욱\*\*\*  
\*영남대학교 컴퓨터공학과  
\*\*영남대학교 컴퓨터공학과  
\*\*\*영남대학교 컴퓨터공학과  
e-mail:law1207@yumail.ac.kr

## A Study of RPC model with XML document exchange using JMS

Heon Jeong\*, Hae-Woo Park\*\*, Byung-Ug Kang\*\*  
\*Dept of Computer Engineering, Yeung-nam University  
\*\*Dept of Computer Engineering, Yeung-nam University  
\*\*\*Dept of Computer Engineering, Yeung-nam University

### 요 약

인터넷상에서 데이터를 전달하기 위해 XML 기술을 활용하여 분산 환경을 어떻게 더욱 발전된 모습으로 지원해 나갈 수 있을 것인지를 두고 다양한 시도가 이루어지고 있다. 분산 환경에서 XML 기술을 활용할 수 있는 분야는 데이터를 표현하여 모든 곳에서 통합화, 표준화하여 모든 사용자들이 접근 가능하도록 하는 기반구조이다. 최근 전자상거래 상(B2B)의 직접적인 거래를 위해서 변환과 문서 전달 기능을 수행하는 메시징 시스템이 필요하다. 이러한 메시징 시스템을 XML과 결합하여 사용한다면 신뢰할 수 있고 확장 가능한 구조를 만들 수 있을 것이다. 따라서 본 논문에서는 JMS를 사용하여 B2B 상에서 메시지 교환을 자유롭게 할 수 있는 RPC 메시징 모델을 설계 한다. 본 메시징 모델을 전자상거래 시스템에 적용하면 B2B 메시지 전송이 용이하게 될 것이다.

### 1. 서론

웹은 현재 우리 생활에 큰 영역으로 자리 잡으면서 전자상거래 영역으로 활동분야를 넓혀가고 있다. 현재 많은 프로그램이 HTTP 전송규약을 기반으로 응용프로그램이 구축되고 있으며 웹 브라우저를 통해서 임의의 플랫폼에서도 응용 프로그램에 접근하여 정보공유와 경제활동을 할 수 있다[1].

최근 널리 사용되고 있는 분산기술들은 상호운영성의 제약과 방화벽에 의한 메시지 차단 문제점을 가지며, 원격 프로시저를 호출하지 않고 웹과의 연동시에 원격지 응용 프로그램과의 상호작용이 되지 않아 서로 통신을 할 수 있는 프로토콜 표준이 필요하다.

현재 XML이라는 표준화 된 데이터 포맷을 이용하여 인터넷상에 데이터를 전달하기 위해 XML 프로토콜이 제한되어지며, 이러한 XML 프로토콜로는 XML-RPC, SOAP, WDDX 등이 있다[12]. 이는 응용 프로그램간 데이터 교환을 지원하고 단순히 데이

터 교환뿐 아니라 그에 적합한 처리를 할 수 있는 응용프로그램까지 전달할 수 있다.

현재 많이 사용되고 있는 전자상거래 시스템에서도 XML을 이용하고 있으며, 이 전자상거래에서 사용되어지는 문서의 대부분은 XML을 이용하여 작성하고 보내어진다.

본 논문에서는 Java Messaging Service(JMS)를 사용하여 B2B상에서 메시지를 교환을 자유롭게 할 수 있는 RPC 모델을 설계 한다.

본 논문의 2장에서는 관련 연구 및 기술에 대해 소개 하고, 3장에서는 JMS를 이용한 XML 포맷의 문서교환을 위한 RPC 모델 설계에 대하여, 4장에서는 결론 및 향후 연구방향에 대해 소개한다.

### 2. 관련 연구 및 기술

#### 2.1. 메시징 서비스

메시징이란 서로 다른 애플리케이션이나 혹은 컴

포넌트 사이에 일어나는 통신의 한 방법으로 이러한 메시징 시스템은 기본적으로 Peer-To-Peer의 형태이며 한쪽 끝에서는 메시지를 보내고 다른 한쪽은 이 메시지를 수신하는 구조를 가진다.

이러한 메시징 서비스를 제공하는 제품을 보통 MOM(Message Oriented Middleware Products)라고 하며 이러한 제품들은 서로 분산/분리되어 있는 컴포넌트들을 하나로 통합하여 유연하고 신뢰성 있는 기업 환경을 구축하는데 필수적인 시스템이다 [7][10].

### 2.1.1 Publish/Subscribe 메시징 시스템

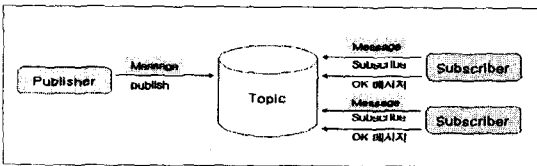
이벤트-드리븐 모델의 Publish/Subscribe 메시징 시스템에서는 Consumers와 Products가 메시지 전송 과정에 참여한다. Products는 이벤트를 Publish하고 Consumers는 관심 있는 이벤트만 Subscribe한다.

Products는 메시지를 어떤 특정한 토픽(Topic)과 결합시키며, 메시징 시스템은 그 메시지와 결합되어 있는 토픽을 이용해 메시지를 라우팅 시킨다[10].

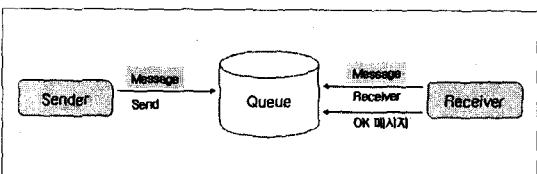
### 2.1.2 Point-to-Point 메시징 시스템

Point-to-Point 메시징 시스템에서는 어떤 메시지의 도착지를 Queue라 한다. 따라서 메시징 시스템은 Queue를 관리하게 된다. Point-to-Point 메시징 시스템에서 Sender는 특정한 Queue로 메시지를 전송하고, 메시징 시스템에서 메시지를 Queue에 대해 등록된 Receiver에게 라우팅 한다[10].

아래의 [그림 2-1]과 [그림 2-2]는 두 가지 방식의 메시징 시스템 기본적인 구조를 보여준다.



[그림 2-1] Publish/Subscribe 메시징 시스템



[그림 2-2] Point-to-Point 메시징 시스템

## 2.2. 컴포넌트 기술과 메시징 시스템

컴포넌트 소프트웨어의 주된 목적은 소프트웨어 모듈의 재사용을 증가 시키고 개발자들 사이의 협동과 협조를 증가시키기 위한 것이다.

현재 가장 널리 사용되고 있는 컴포넌트 기술들은 상호간의 호환성이 떨어진다. 이러한 컴포넌트 기술 간의 호환성문제와 웹과의 호환성 문제를 해결하기 위하여 XML을 이용하고자 하는 많은 노력이 있었다.

XML은 플랫폼 및 프로그래밍 언어에 독립적이고 확장가능하며, 그 자체만으로도 데이터와 메시지 전달을 하는데 있어 표현수단으로 사용될 수 있다.

컴포넌트 기술들을 사용하기 위해서는 원격지에 있는 메소드명을 알아야 하기에 코드의 재사용성이나 투명성을 떨어뜨려 결국 시스템 환경이 변경되면 소스코드를 변경해야한다. 이에 반해 메시징 시스템에서 메시지를 전송하는 측은 수신하는 측에 대한 어떠한 정보도 필요로 하지 않는다. 메시징 시스템은 송신자와 수신자 중간단계에 메시지를 보관하는 객체를 제공한다. 따라서 송신하는 측은 객체에 메시지를 보내고 수신하는 측은 해당 객체에서 메시지를 가져와 처리한다. 결국 송신하는 측과 수신하는 측은 서로의 정보에 관해서 전혀 알 필요가 없다.

## 2.3 XML 프로토콜

XML 프로토콜은 애플리케이션 간 교환 또는 컴포넌트 모델로서 XML을 이용하는 것으로 다음의 4가지 기능을 지원함으로써 애플리케이션 간 교환을 가능하게 한다.

- (1) 데이터 교환 시 데이터 캡슐화.
- (2) RPC와 같은 외부 함수 호출 지원.
- (3) 비구조적 데이터의 직렬화.
- (4) HTTP 프로토콜사용.

기존의 네트워크를 통해 서비스를 호출하는 방법은 그 자체가 워낙 복잡하여 방화벽 사이로 통신하기에는 어렵기 때문에 XML 기반의 원격 메소드 프로토콜이 제안되었다. 이 프로토콜은 방화벽에서 그 내용을 검사하기가 쉬운 방화벽사이로 클라이언트와 서버가 서로 통신하는 것을 가능하게 한다.

XML-RPC와 SOAP은 바로 이러한 XML 기반의 원격 메소드 호출 프로토콜이다. 이런 프로토콜은 원격 객체상의 메소드를 호출하고 그 응답을 받기 위한 메시지 포맷으로 XML을 명시한다.

### 3. XML 포맷의 문서교환을 위한 RPC 모델 설계

이번 장에서는 본 논문에서 제안한 JMS를 이용한 XML 포맷의 문서교환을 위한 RPC 모델 설계방법에 대하여 논한다.

JMS를 전송수단으로 이용하는 서비스를 제공하기 위해서는 Request/Reply 형태의 방법을 이용한다. JMS API에서 제공하는 QueueRequestor 클래스를 사용하여 Request/Reply 형태의 방법을 설계하며, 아래의 [표 3-1] Request/Reply 설계의 절차를 나타낸 것이다.

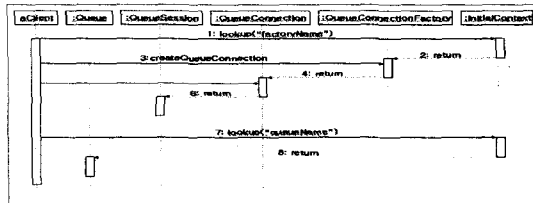
단계	내용
1	Queue에 대한 Sender 생성. 임시 Queue 생성 및 Receiver 생성.
2	임시 Queue에 전달할 헤더 필더 세팅. 메시지 전송 후 Receiver 블록킹.
3	Replier 노드에서 메시지를 받게 되는 즉시 응답 메시지를 작성 후 헤더에 명시된 Queue로 전송.
4	Request 노드는 응답 메시지 받은 후 블록킹 해제.

[표 3-1] Request/Reply 설계의 절차

#### 3.1. 노드 초기화 단계(Node Initialization)

노드를 초기화하기 위해서는 Sender나 Receiver를 생성해야 한다. 서비스를 제공해주는 객체나 큐 객체 같이 어떤 관리되는 객체를 생성하기 위해서는 일반적으로 JNDI(Java Naming and Directory Interface)를 사용한다.

아래의 [표 3-2]와 [그림 3-1]은 노드 초기화 단계이다



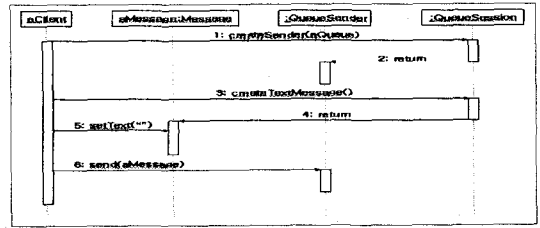
[그림 3-1] 노드 초기화 단계

단계	내용
1	메시지 제공자 연결을 위한 객체 생성.
2	도착지들에 대한 연결을 얻기 위한 팩토리 생성.
3	서비스 제공자에 대한 연결을 표현할 객체 생성.
4	연결에 대한 세션사항을 저장할 객체 생성

[표 3-2] 노드 초기화 단계

#### 3.2. 메시지 보내기 단계(Sending a Message)

아래의 [표 3-3]과 [그림 3-2]는 메시지 보내기 단계를 표현한다.



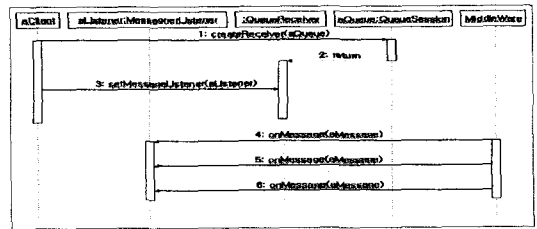
[그림 3-2] 메시지 보내기 단계

1	세션 문맥 안에서 메시지 전송 작업 정의.
2	메시지 객체 생성, 메시지 컨텐츠 세팅.
3	메시지 전송.

[표 3-3] 메시지 보내기 단계

#### 3.3 메시지 받기 단계(Receive a Message)

아래의 [표 3-4]와 [그림 3-3]은 메시지 받기 단계를 표현한다



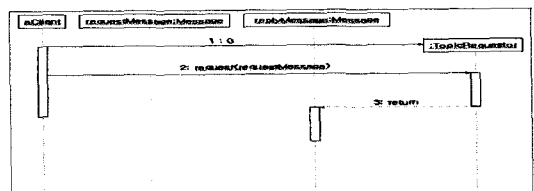
[그림 3-3] 메시지 받기 단계

1	연결에 대한 세션상세사항 저장 객체 생성
2	메시지를 받기 위한 객체 생성.
3	메시지 리스너 세팅.
4	미들웨어는 메시지를 받을 때마다 메시지 리스너 호출.

[표 3-4] 메시지 받기 단계

#### 3.4 서비스 요청 단계

아래의 [표 3-5]와 [그림 3-4]는 서비스 요청을 표현 한다.



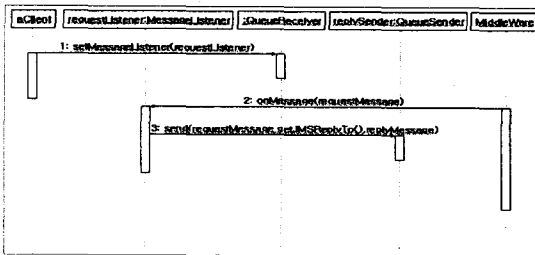
[그림 3-4] 서비스 요청하기 단계

1	QueueRequestor 객체를 생성.
2	request()를 호출하여 그 요청을 가지고 있는 메시지를 파라미터로 전달한다.
3	응답을 받게 되었을 때에는 receive()는 그 응답메시지를 반환.

[표 3-5] 서비스 요청 단계

3.5. 서비스 제공 단계

아래의 [표 3-6]과 [그림 3-5]는 서비스 제공 단계를 표현한다.



[그림 3-5] 서비스 제공 단계

1	receiver 객체생성 후 MessageListener 객체를 그 객체에 대한 메시지 리스너로 세팅.
2	null 큐에 대한 sender 객체 생성.
3	요청이 전달되면 미들웨어는 메시지 리스너 객체의 onMessage()를 호출.
4	응답메시지를 생성 후 전송.

[표 3-6] 서비스 제공 단계

4. 결론 및 향후 연구방향

XML은 분산환경에서 서로 다른 응용 프로그램들의 통신 수단으로 지원되며, 기존의 복잡한 분산환경을 단순화 시키는데 큰 역할을 수행할 수 있다.

XML-RPC는 전송은 HTTP를 이용하고 인코딩은 XML을 사용하여 분산환경에서 메소드의 매개변수와 반환타입이 XML-RPC에서 인코딩 가능한 것이라면 XML-RPC로 그 메소드를 호출하여 분산된 시스템에 있는 클라이언트가 네트워크의 또 다른 부분에 있는 서버에서 실행될 수 있는 작업을 요청할 수 있다.

본 논문에서는 JMS와 XML을 결합해서 사용하면 XML자체의 높은 이식성이라는 이점과 메시징에 대한 국제적인 인식이라는 이점이 더 추가된다. 또한 XML 기반의 전송수단을 이용하면 방화벽을 통

해서도 그대로 전송될 수 있는 HTTP를 충분히 활용하는 분산된 애플리케이션을 만들 수가 있다.

향후 연구과제로는 제안된 방법의 검증을 위해 설계방법을 토대로 시스템을 구축하여야 하며, 안전한 메시지 전달을 위하여 XML 보안을 적용하여야 할 것이다.

참고문헌

[1] 임정은, 윤용익, "XML-RPC를 이용한 문서교환 및 제어용 미들웨어 구조 연구" 한국정보과학회 논문지, 2002

[2] 이경하, 이경철, "XML 프로토콜", 한국 정보과학회지 제19권 1호, 2001

[3] 임종선, 주경수, "효율적인 e-Procurement를 위한 XML 메시징 시스템 개발", The Journal of Korean Institute of CALS/EC Vol 7, No.2, 2002

[4] 김용수, 주경수, "SOAP을 기반으로 한 XML Hub 시스템 개발", 정보처리학회 논문지 제10-D권 제2호, 2003

[5] 임종선, 주경수, "전자상거래를 위한 XML 메시징 시스템 개발", 로지스틱연구 제10권 제1호, 2002

[6] 임종선, 주경수, "TCP/IP 기반의 XML 메시징 시스템 설계 및 구현", 멀티미디어학회 논문지 제6권 제3호, 2003

[7] Kal Ahmed, Professionnal Java XML, 정보문화사, 2002

[8] Danny Ayers, Professionnal Java Data, 정보문화사, 2002

[9] Simeon Simeonov, Building Web Services with Java, 인포북, 2002

[10] 박천구, 문창수, EJB & WebLogic, 가메출판사, 2003

[11] Apache SOAP Documentation : User's Guide <http://ws.apache.org/soap/docs/index.html>

[12] XML Protocol Comparisons <http://www.w3.org/2000/03/29-XML-protocol-matrix>

[13] Java Message Service <http://java.sun.com/products/jms>