

# 진보된 분산객체 기반의 주문 처리 시스템

송민철\*, 박종구\*, 안현식\*\*

\*성균관대학교 정보통신공학부, \*\*국민대학교 전자정보통신공학부

## Order Management System Based on Advanced Distributed Objects

Min-Chul Song\*, Jong-Koo Park\*, Hyun-Sik Ahn\*\*

\*School of Info. & Comm. Eng., SungKyunKwan University

\*\*School of Elec. Eng., Kookmin University

### 요 약

최근 많은 개발자들이 웹 서비스를 기업 내 어플리케이션 통합에서부터 기업간 데이터 전송에 이르기까지 모든 종류의 문제를 해결할 수 있는 솔루션으로 전망한다. 본 논문에서는 이러한 웹 서비스를 구현할 수 있는 기반을 제공하는 SOAP 프로토콜을 기반으로 하여 좀 더 효율적인 메시지 처리 방법을 통해 관련 전자상거래 시스템에 응용하기 위한 방안을 모색한다. 아울러 분산된 이기종 시스템간에 객체들 사이의 메시지 전송을 위하여 XML 메시지를 생성하고, SOAP Tool Kit 즉, AXIS 를 통해 좀 더 효율적인 정보 공유 방안을 제시한다. 그리고 관련 사례 연구로 분산된 상호 독립적인 어플리케이션들의 객체들 간의 원격 메시지 호출 과정을 테스트하는 주문 처리 시스템을 설계하고 구현한다.

**KeyWords** : XML, Web Services, Distributed objects

### 1. 서론

기업들은 효과적인 비즈니스 모델을 구축하기 위하여 다양한 방법을 모색하고 있다. 특히 1990년대 인터넷의 급격한 보급과 정보시스템(IT)의 발전은 전세계적으로 많은 기업들로 하여금 새로운 비즈니스 모델 채택을 강요하도록 하였다. 최근에는 전자상거래, 가상조직, e-Business 등의 개념들이 출현하면서 분산 시스템에 대한 관심과 요구가 크게 증가하고 있는 추세이다. 기업 프로세스의 확장과 분산화는 단일 기업이 갖는 지식, 능력의 한계를 극복하고 다른 조직의 핵심 경쟁력을 활용할 수 있다는 것을 의미한다[1].

분산환경의 조직모형을 지원하기 위한 정보 시스템의 구조는 업무구조, 어플리케이션 구조, 그리고 맨 아래의 기술 구조 등 세 개의 계층으로 구분할 수 있고, 어플리케이션 구조 계층의 통합에는 보통 메시지 기반의 서비스를 이용한 정보 교환 방법이 사용되고 있으나, 독립적인 어플리케이션들 사이의 상호 운용성을 지원하기 위해서는 표준화된 메시지와 전송 프로

토콜을 정의해야 하는 문제가 존재한다[2]. 이러한 문제를 위한 해결 방안으로 SOAP 과 XML 이라는 정보 기술을 들 수 있다.

인터넷을 경유하여 메시지를 전송하는 기술로는 가장 일반적인 HTTP 를 비롯하여 DCOM 과 CORBA, JAVA, RMI 같은 분산 객체 기술 등이 있다. 하지만 모두 결정적인 결점을 갖고 있어 불완전하다[3]. 이에 따라 MS 와 IBM, 로터스 등은 기존의 인프라를 그대로 활용하면서 보안상의 문제를 야기하지 않고 원격으로 메시지 전송 및 호출을 원활하게 지원하는 새로운 프로토콜 기술로 SOAP 을 공동으로 개발했다. SOAP 을 이용하면 앞서 설명한 기존의 분산 객체 기술을 결점인 방화벽, 언어 제약, 플랫폼 제약 등의 문제를 해결할 수 있다.

이러한 점들을 고려하여 본 논문에서는 분산된 이기종 시스템간에 객체들 사이의 메시지 전송을 위하여 XML 메시지를 생성하고, SOAP Tool Kit 즉, AXIS 를 통해 좀더 효율적인 정보 공유 방안을 제시하고자 한다. 그리고 관련 사례 연구로 분산된 상호 독립적인

어플리케이션들의 객체들 간의 원격 메시지 호출 과정을 테스트하는 주문 처리 시스템을 설계하고 구현한다.

2. 연구 배경

2.1 분산 객체 기술

OMG 는 CORBA 표준 프로토콜인 IIOP 를 만들었다. 그리고, MS 는 프로토콜로 DCOM 을 주창하였는데, 이는 장비의 한계를 넘을 수 있는 기술이다. 강력한 세력 중의 하나이지만, IIOP 나 DCOM 보다 나중에 만들어진 Java 커뮤니티는 SUN 의 RMI 프로토콜을 사용한다[1].

이러한 프로토콜을 지원하는 응용프로그램은 네트워크를 통해 다른 컴퓨터에 있는 구성요소를 호출할 수 있다. 일반적으로 이러한 호출은 원격 프로시저가 호출되고 있는 RPC 의 형태로 있게 된다. 그리고 나서, 원격 프로시저는 호출한 응용프로그램에게 응답을 보낸다. DCOM 을 사용하는 클라이언트는 DCOM 을 지원하는 서버만 호출할 수 있다. CORBA 도 마찬가지다. 공동운동을 위해 많은 노력을 했지만, 널리 사용되진 못했다.

2.1.1 CORBA/IIOP

IIOP 는 기본적인 프로토콜을 정의하는 GIOP(General Inter-ORB Protocol)의 TCP/IP 특별판이다. 객체 모델을 확장하면, 컴퓨터 사이에서는 응용프로그램 간의 통신 프로토콜로 사용될 수 있고 네트워크를 통해서도 풍부한 프로그래밍 환경이 만들어 질 수 있다.

CORBA/IIOP 는 연결 지향적이고 상태 프로그래밍 모델을 가지고 있고, 운영체제에 독립적인 아키텍처처럼 확장성이 많지 않다[1].

2.1.2 DCOM

MS 는 자체적인 프로토콜로 DCOM 을 개발했으며 이 프로토콜은 컴퓨터간의 경계를 넘나들 수 있다. DCOM 은 가장 복잡한 분산 컴퓨팅 프로토콜 중의 하나이다.

또한 DCOM 은 CORBA/IIOP 와 개념이 유사한 커넥션 관련 과부하를 가지고 있으므로 확장성은 DCOM 의 장점에 속하지 않는다. DCOM 은 가비지 컬렉션과 같은 몇몇 다른 작업을 수행하며 매우 엄격한 보안도 구현할 수가 있지만 이러한 경우 복잡성이 증대 된다.

2.1.3 RMI

분산 컴퓨팅계에서 경쟁 관계에 있는 다른 프로토콜로 RMI 를 들 수 있다 RMI 는 서브 프로토콜 세 개 중 하나만이 비직속형이지만 성능을 고려해서 제작된 프로토콜이다. RMI 는 가비지 컬렉션을 구현하고 있

며 보안을 염두에 두고 있다. 그러나 RMI 의 단점 중의 하나는 Java 라는 단일 언어 플랫폼을 빼고서는 생각하기 힘들다는 점이다[4].

2.2 기존 분산객체 기술의 문제점

앞서 제시한 분산객체 기술의 프로토콜을 이용해서 여러 어플리케이션을 분산시키고 이를 연결해서 사용하면 강력한 엔터프라이즈 환경을 구성할 수 있다. 그런데 인터넷과 웹 환경이 일반화되면서 이들 기술에 몇몇 문제점이 나타난다. 가장 큰 문제점은 CORBA, DCOM, RMI 를 이용하여 어플리케이션 사이의 통신을 구현하기 위해서는 꽤 복잡한 환경설정과 설치를 해야 한다는 점이다. 여기에 보안이나 트랜잭션 관리를 위해서 추가적인 부분이 필요하다면 이 과정이 더욱 복잡해진다. 또 다른 문제점은 인터넷을 통한 통신이다. 분산 통신 기술이 주로 동기식이 주류를 이루고 있기 때문에 통신을 담당하는 양쪽 끝이 기본적으로 연결되어 있어야 한다. 그러나 인터넷은 연결되어 있는 것을 보장하지 않는다는 특징이 있다. 또한 방화벽 문제도 만만치 않다. 기존의 분산객체 기술은 특정 포트를 통해 통신하기 때문에 방화벽과 관련한 문제가 항상 있었다. DCOM 의 RDS(Remote Data Services)와 CORBA 의 HTTP 터널링과 같이 이러한 문제점을 해결하기 위한 대안이 없는 것은 아니지만 근본적인 해결책이라고 보기는 어렵다

2.3 새로운 표준 SOAP 의 등장과 웹 서비스

SOAP 은 모든 네트워크 프로토콜, 주로 HTTP 에서 RPC 와 Messaging 을 지원하는 XML 기반의 프로토콜이다. 인터넷이 기업 컴퓨팅의 핵심 플랫폼으로 자리잡아가고 있는 상황에서 인터넷을 통한 원격 어플리케이션 활용을 원활하게 지원하기 위한 목적으로 만들어 졌다. 이러한 분산 어플리케이션을 실현하는 중요한 기술로서 RPC(Remote Procedure Call)가 있다.

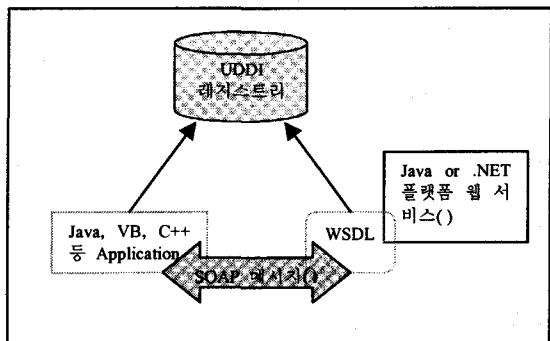


그림 1. 웹 서비스 메커니즘과 SOAP

그림 1 은 웹 서비스 메커니즘과 SOAP 의 역할을

나타내고 있다.

앞에서 살펴봤듯이 지금 까지 사용해 온 여러 RPC 기술들은 플랫폼 호환이나 보안 측면에 제약이 많다. 이러한 단점을 극복하고자 XML의 유연성과 HTTP의 대중성에 기반을 둔 SOAP이 제안 되었다[5].

다음의 3 절과 4 절은 SOAP 과 웹 서비스의 메커니즘을 사용하여 분산된 상호 독립적인 어플리케이션들의 객체들 간의 원격 메시지 호출 과정을 테스트하는 주문 처리 시스템을 설계하고 구현한다.

### 3. 주문 처리 시스템

#### 3.1 주문 처리 시스템의 필요성

현재 많은 기업들은 보다 효율적인 주문 처리 프로세스를 지향한다. 생산된 부품을 공급해주는 본사와 그 부품들을 공급 받아 판매하는 판매지점이 존재하는 형태의 기업환경에서 주문 처리 시스템이 이용될 수 있다. 아직도 많은 중·소규모 기업에서는 폼 메일을 이용하거나 팩스 또는 전화를 통해 주문을 처리하고 있다. 이것은 많은 경영자나 지점 운영자들이 문서화한 서류에 익숙해 있기 때문이다.

상품 관리에 대한 필요성은 물론 매출 증대에 따른 인터넷의 접목을 위해서도 주문 처리 시스템의 중요성은 더욱 절실해 지고 있다. 주문처리 시스템을 통해 주문 리스트에서 주문의 상세 정보, 주문한 상품의 정보 등을 확인하고 또한 거래 내역과 처리 과정을 확인 할 수 있도록 하여 기업 환경의 신뢰도를 높일 수 있다.

#### 3.2 주문 처리 시스템의 구성

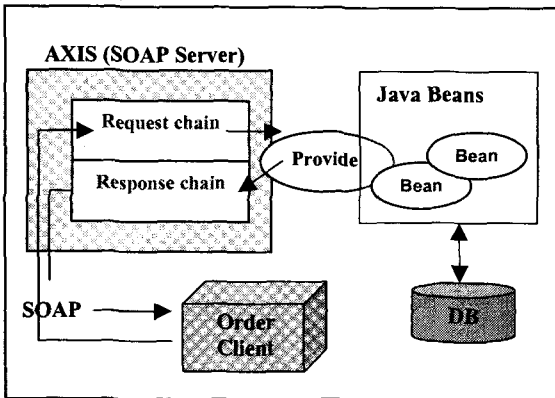


그림 2. 주문 처리 시스템의 개략적인 동작 과정

그림 2 는 본 논문에서 구현한 주문 처리 시스템의 개략적인 동작을 보인 것이다.

판매지점에서 재고 확인 후 재고를 다시 채워야 할 때, 구매 주문을 위한 클라이언트 어플리케이션을 이

용해 주문 정보를 채워 넣은 후 본사에 보내면, 본사의 부품 주문 처리 웹 서비스를 통해 처리 된다. 그리고 주문 정보를 리스트화 하여 클라이언트에게 다시 리턴한다.

서버의 주문 처리 시스템은 주문 클라이언트의 SOAP 요청을 받아서 처리하고 서비스를 호출하며, SOAP 응답을 리턴한다. 메시지 경로는 메시지가 서비스 쪽으로 흐르는 요청 플로우와 응답 메시지가 돌아오는 응답 플로우로 나눌 수 있다. 그리고 주문 처리를 위한 각각의 프로세스를 **JavaBeans** 로 컴포넌트화 시켰다. 컴포넌트 기반 기술의 장점인 재사용성을 기반으로 어플리케이션 개발을 할 경우 생산성을 높일 수 있다. 현재 컴포넌트 소프트웨어 시장에서 가장 큰 위치를 점하는 컴포넌트 모델 및 운영 구조는 MS의 COM 과 SUN의 **JavaBeans**, OMG의 **CORBA**, Oracle의 **Cartridge** 등이 있다. 아래의 표 1 은 이들 4 가지 정보 기술 아키텍처를 비교한 것이다[7].

표 1. 컴포넌트 기술 비교

	COM+	Java Beans	CORBA	NCA Cartridges
시장 장악력	강함	커짐	커짐	작음
컴포넌트 운영주체	Windows	JVM	다수의 시스템 존재	Oracle 제품군
언어 독립성	높다	Java 만 허용	높다	높다
플랫폼 독립성	제한적	높다	높다	높다

#### 4. 주문 처리 시스템의 구현

본 절에서는 3 절에서 제시한 주문 처리 시스템의 구성을 바탕으로 원격 메시지 호출 과정을 테스트한 주문 처리 시스템의 구현 절차를 기술한다.

논문에서 구현한 웹 서비스 기반의 주문처리 시스템은 Java 플랫폼 기반의 SOAP Tool Kit 중 하나인 Apache AXIS 을 이용하여 구현하였다. AXIS 는 최신용 SOAP 구현이다. 현재, Java 와 함께 사용되는 표준 SOAP 프로그램으로 Apache SOAP 버전 3 이다. AXIS 는 처음부터 다시 구현했으므로, 설계상 유연성이 있고 모듈러 구조를 취하며, 이전 버전과는 확연한 차이를 보인다. 그리고 Java Sublet 엔진으로 **Jakarta Tomcat v4.0.18** 을 사용했고, AXIS 엔진이 SOAP 메시지를 처리하기 위해 XML 파서가 필요한데 **Xerces XML** 파서를 사용하였다.

아래의 그림 3 은 구매 주문을 위한 클라이언트의 주문 양식을 나타낸 것이다. 공급 받을 자체부품의 부

품번호, 부품명, 단위, 수량, 단가 등의 정보가 작성된다.

일련 번호	Part No.	Part Name	단위	수량	단가	금액
1						
2						
3						
4						

그림 3. 처리될 주문 양식

이 후 주문 양식을 통해 작성된 정보는 HTTP 를 통해 웹 서비스로 SOAP 메시지를 전송한다. 그 처리는 SOAP 서버(SOAP 엔진)에게 위임되고 SOAP 서버는 주문 처리 웹 서비스를 호출한다.

본 논문에서 구현한 주문 처리 웹 서비스 즉, Spare Part Order 웹 서비스는 판매 지점의 구매 주문을 받아서 처리하고 요청된 정보와 그 결과 정보를 데이터베이스에 저장한다. 데이터베이스는 테스트하기 쉽고, 가벼운 MySQL v3.23 을 사용하였고, 클라이언트로부터 얻은 정보를 처리하기 위하여 JDBC 를 사용하여 레코드 셋의 필드 값을 받아 시스템 변수로 전환하는 부분과 각각의 변수에 레코드 셋에서 받아온 값들을 매핑 시키는 과정을 각각 JavaBeans 로 구현하였다.

웹 서비스를 생성하기 위해서는 웹 서비스 설치용 디스크립터를 작성해야 한다. 즉, AXIS 는 AXIS 엔진의 속성을 설정하는 목적으로 WSDO 라는 XML-기반 설치용 디스크립터를 정의한다. WSDO 파일은 다양한 어플리케이션에 고유한 handle, chain 그리고 transport 를 쉽게 설치할 수 있도록 해준다[8]. 그림 4 는 구매 주문을 처리하기 위한 웹 서비스를 생성하기 위한 WSDO 파일의 내용을 나타낸다.

```

<deployment
name="SparePartOrderService"
xmlns="http://xml.apache.org/axis/wsdd/"
xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance">
<service name="SparePartOrderService" provider="java:RPC">
<parameter name="className" value="com.min.bo.SparePartSOBO"/>
<parameter name="allowedMethods" value="placeOrder"/>
</service>
</deployment>
    
```

그림 4. 주문 처리 웹 서비스의 WSDO 파일

이렇게 작성된 Spare Part Order 웹 서비스를 통해 판매 지점과 본사를 이어주는 상호 독립적인 어플리케이션들간의 원격 메시지 호출이 이루어진다. 처리 결과, 즉 주문내역은 주문 처리 시스템을 통해 확인할 수

### 5. 결론 및 향후 과제

인터넷 기반의 기업간 또는 기업 내부간의 거래는 시간과 공간을 초월한 판매 및 구매와 실시간의 정보 교환의 장점을 가진다. 이러한 장점은 새로운 시장의 개척과 기존 업무의 효율 및 비용의 절감 측면에서 많은 기업들이 발 빠르게 인터넷 기반의 전자상거래를 도입하는 이유이다. 그러나 기존의 인터넷 기반의 전자상거래 역시 해결해야 할 문제점을 가지고 있는데, 데이터 공유 즉 정보 인프라의 호환성이 가장 큰 문제점이라 할 수 있겠다.

이러한 문제 해결을 위해 본 논문에서는 시스템 간의 통신에 대한 메시지 형식의 표준인 SOAP 과 웹 서비스 등의 새로운 기술을 통하여 좀 더 향상된 분산 컴퓨팅 환경에서 기업 내 주문 처리 시스템에 응용하려 시도했다.

하지만 새로운 분산객체 기술을 이용해 주문 처리 시스템을 제시했지만 본 논문에서의 어플리케이션은 최소한의 기능만을 구현하였다. 완벽한 프로그램이 되기 위해서는 사용자 인터페이스의 확장, 특히 주문서의 통계의 의한 매출관리, 상품정보의 연동에 의한 재고관리, 회원정보와의 연동에 의한 고객 마일리지 적용 등 앞으로 적용하여야 할 많은 부분에 대하여도 염두에 두어야 할 것이다.

### 참고문헌

- [1] Zoran Zaev 외, "XML Web Service", wrox, 2002
- [2] 민대기·장태우·박찬권·박진우, "분산객체와 XML 기반의 생산계획 컴포넌트 개발에 관한 연구", IE Interfaces Vol. 15 No. 1 pp. 10-19, 2002
- [3] 정지훈, "웹 서비스", 한빛 미디어, 2002
- [4] Sun Microsystems, "RMI spec", <http://java.sun.com/products/jmi/index.html>
- [5] W3C, "Simple Object Access Protocol Specification", <http://www.w3.org/TR/SOAP>
- [6] 전제철, "DCOM 기술분석 및 전망", 한국정보처리학회 제 7 권 제 4 호 p54-58, 2000
- [7] 문태수·조규중·김호진, "컴포넌트 기반의 XML/EDI 시스템 설계 및 구현", 정보시스템학회 p98-109, 2001
- [8] S.Jeelani Basha·Romin Irani, "The Next Generation of Java Soap", wrox, 2002