

# SOAP 브리지를 이용한 분산객체시스템의 연동

박 성 은<sup>†</sup> · 김 신 우<sup>†</sup> · 이 용 규<sup>††</sup>

## 요 약

기존에는 분산객체시스템 표준으로 CORBA, DCOM, EJB 등이 널리 사용되었으나, 최근에 웹 애플리케이션을 위한 표준 분산기술로 XML 기반의 SOAP이 제안되었다. 그러나 이러한 분산 기술들은 각각의 표준 환경에서만 운용되는 한계가 있다. 따라서 본 논문에서는 기존 분산객체시스템들과 SOAP 기반 시스템의 상호 운용성을 높일 수 있는 SOAP 브리지를 설계 및 구현한다. 이는 기존 분산객체시스템의 표준인 CORBA와 DCOM이 새로운 SOAP 기반 시스템과 연동될 수 있도록 그 형식들을 매핑하는 역할을 한다. 또한 이에 대한 성능 실험을 통해 평균 트랜잭션 처리시간에 대한 오버헤드가 크지 않음을 보인다. 따라서 SOAP 기반의 분산 객체들을 새로이 구축하는 대신에 기존 분산 환경의 객체들이 재활용될 수 있다. 그리고 SOAP 브리지의 실제 적용 사례로 대규모 분산 시스템인 XML 기반 전자입찰 시스템을 설계하고 구현한다. 이 시스템은 XML 표준을 기반으로 하기 때문에 기존 전자상거래 시스템이 갖는 표준화와 확장성의 문제를 해결할 수 있다.

## A SOAP Bridge for Interworking of Distributed Object Systems

Sung Eun Park<sup>†</sup> · Shin Woo Kim<sup>†</sup> · Yong Kyu Lee<sup>††</sup>

### ABSTRACT

Distributed applications based on standards such as CORBA, DCOM, and EJB have been widely used in distributed environments. Recently, another XML-based standard, SOAP, has been recommended for web applications. However, these standards have a shortcoming in inter-operation because they can be run on their own environments. In this paper, we design and implement a SOAP bridge for inter-operation of SOAP with other distributed object standards. Its function is to map formats between SOAP and CORBA/DCOM at run time. Through performance experiments, we show that the computing overhead is small compared to the whole transaction execution time. Therefore, distributed objects of other platforms can be reused instead of developing new objects based on SOAP. Also, using the SOAP bridge, we design and implement an XML-based bidding system on the web. Because it is based on the web standards, it has advantages such as extensibility and interoperability compared to previous proprietary electronic commerce systems.

**키워드 :** 습(SOAP), 습 브리지(SOAP Bridge), 분산객체시스템(Distributed Object System), CORBA, DCOM, XML, 전자상거래, 전자입찰

### 1. 서 론

CORBA[2, 13], DCOM, EJB와 같은 기존 분산객체시스템 표준들은 자원의 공유, 객체의 재사용성 등과 같은 장점이 있음에도 불구하고 복잡한 구현 환경 설정·고가의 소프트웨어 구입 등의 문제점이 있다. 따라서 최근에는 구현이 쉽고, 표준화되어 있어 이러한 문제를 해결할 수 있는 분산 기술로 SOAP(Simple Object Access Protocol)[7, 11, 14]이 많은 관심을 받고 있다. SOAP은 XML과 HTTP 기반으로 플랫폼에 독립적이고, 웹을 통해 데이터를 전송한다. 또한 이는 최근에 국제전자상거래 표준인 ebXML(electronic busi-

ness XML)[15]에서 통신 프로토콜로 채택되기도 하였다. 그러나 이러한 분산기술들은 각각의 표준 환경에서만 운용되는 한계가 있다. 따라서 이러한 문제를 해결하기 위해 이기종간 분산 객체들의 상호 운용성을 높이는 메타데이터를 정의하는 연구[4, 8]가 있었고, 최근에는 XML을 이용하여 웹을 통해 CORBA 서버의 서비스를 이용할 수 있게 해주는 응용 웹 서버를 구현하는 등의 연구[17, 18]가 있었다. 그러나 이들은 이기종간 시스템의 연동 가능성을 제안하고, 그에 따른 성능 평가가 부족하여 그 효용성을 입증하지 못했다.

따라서 본 논문에서는 이러한 문제를 해결하고 분산객체시스템을 용이하게 구축하기 위해서 SOAP 브리지를 설계 및 구현한다. SOAP 브리지는 기존 분산객체시스템들과 새로운 SOAP 기반 시스템간의 상호 운용성을 높일 수 있게

※ 본 연구는 동국대학교 논문 게재 연구비 지원으로 이루어졌음.

† 준 회원 : 동국대학교 대학원 컴퓨터공학과

†† 종신회원 : 동국대학교 컴퓨터멀티미디어공학과 교수

논문접수 : 2002년 3월 20일, 심사완료 : 2002년 11월 30일

하는데, 기존 시스템이 새로운 SOAP 기반의 시스템과 연동될 수 있다는 장점이 있다. 그리고 성능 실험을 통해 평균 트랜잭션 처리시간에 대한 오버헤드가 크지 않음을 보인다. 또한 본 논문에서는 이러한 SOAP 브리지의 실제 적용 사례로 대규모 분산 시스템인 XML 기반 전자입찰 시스템을 설계하고 구현한다. 이를 위해 XML 스키마, DOM, XSL 등과 같은 XML 관련 기술을 적용하며, SOAP과 SOAP 브리지를 사용하여 분산객체시스템을 구축한다. XML과 SOAP 기반으로 구현된 이 시스템은 기존 전자상거래 시스템이 갖는 표준화와 확장성의 문제를 해결할 수 있다.

## 2. 관련 연구

기존 분산객체시스템 표준들은 독자적인 인프라로 구성되어 상호 운용성이 떨어지는 문제가 있다. 따라서 이러한 문제를 해결하기 위해 이기종 시스템 기반에서 분산객체들의 상호 운용성을 높이기 위한 연구가 있었다. 또한, 객체지향 소프트웨어의 재사용성을 높이기 위해 소프트웨어가 소개되기도 하였고[1, 3, 6, 10, 12], CORBA와 COM 연동을 위한 브리지 설계나 이들 간의 통신을 위해 메타데이터를 정의하는 등의 연구가 있었다[4, 8].

최근에는 메타 데이터로서의 특징이 강한 XML을 이용하여 웹을 통해 CORBA 서버의 서비스를 이용할 수 있게 해주는 응용 웹 서버를 구현하는 등의 연구가 있었다[17, 18]. 그러나 여기에서는 서버 구현 기술에 따른 성능 실험만을 측정하여서 이기종 시스템 연동에 따른 효율성은 알 수가 없다.

따라서 본 논문에서는 이와 관련된 연구로 CORBA 뿐만 아니라 DCOM 기반의 기존 분산객체시스템들이 XML 기반의 표준 분산기술인 SOAP을 이용한 시스템과 연동될 수 있도록 SOAP 브리지를 설계하며, 이에 대한 성능 실험을 통해 이기종 시스템 연동에 따른 효율성을 연구하고자 한다.

## 3. 분산객체시스템 연동과 SOAP 브리지 설계

본 절에서는 SOAP 브리지를 이용한 분산객체시스템 연동과 SOAP 브리지에 대해서 살펴본다.

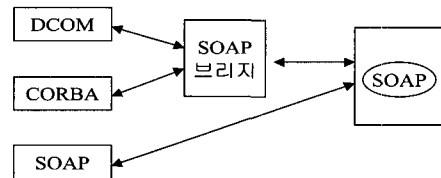
### 3.1 SOAP 브리지를 이용한 분산객체시스템 연동

(그림 1)은 서로 다른 기반의 분산객체시스템들을 나타낸 그림이다. 즉, (그림 1)(a)은 기존 분산객체시스템을 의미하는데, 이와 같은 환경에서는 서로 다른 시스템의 객체에 접근하려면 각각의 표준 방식을 따라야 한다. 그러나 (그림 1)(b)처럼 XML 기반인 SOAP을 이용하면 구축이 쉽고, 분산객체의 접근이 용이한 장점이 있다. 그러나, 현재 CORBA나 DCOM으로 구성된 분산객체시스템을 SOAP 기반으로

전부 대체하는 것은 무리가 있다.



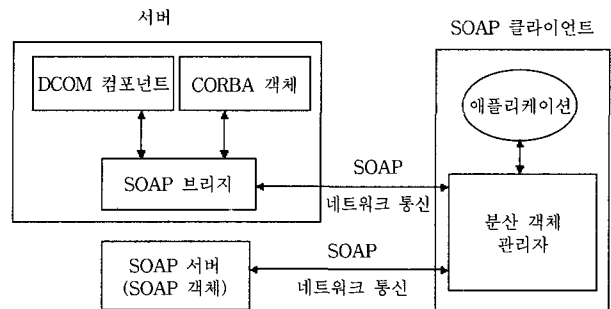
(a) DCOM/CORBA 시스템 (b) SOAP 시스템



(c) SOAP 브리지를 이용한 분산객체시스템

(그림 1) 분산객체시스템의 종류

따라서 본 논문에서는 (그림 1)(c)에서 보는 바와 같이 기존 분산객체시스템들과의 연동을 위해 미들웨어 역할을 하는 SOAP 브리지를 설계한다. 이러한 SOAP 브리지의 기능은 (그림 2)와 같다.



(그림 2) 분산객체시스템 연동

즉, SOAP 클라이언트가 요청을 보내면, 관리자가 서버의 종류를 판단하여, SOAP 브리지를 통해 CORBA에 대한 요청은 CORBA 서버에게, DCOM에 대한 요청은 DCOM 서버에게 전송한다. 이렇게 전송된 요청에 대한 결과값은 다시 SOAP 브리지를 통해 SOAP 클라이언트에게 전달된다.

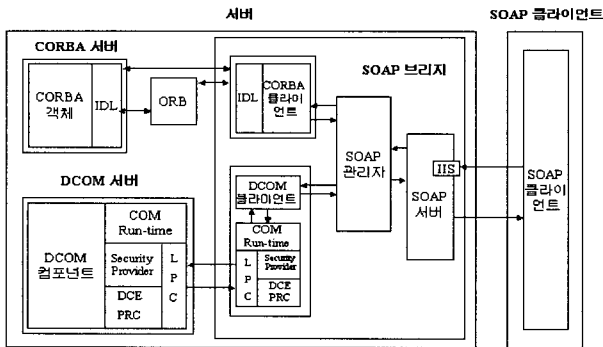
### 3.2 SOAP 브리지

(그림 3)은 CORBA나 DCOM 기반의 기존 분산객체시스템과의 연동을 위해 SOAP 브리지를 설계한 그림이다.

먼저, 그림의 왼쪽은 CORBA나 DCOM 기반의 서버이다. 그림의 오른쪽은 XML 기반의 SOAP 클라이언트이며, SOAP 브리지를 통해 CORBA나 DCOM 서버의 객체에 접근할 수 있다. 그리고 서버 안에 있는 SOAP 브리지는 이기종의 두 시스템이 서로 연동할 수 있게 한다.

그 연산 과정은 다음과 같다. 먼저 사용자로부터 입력받은 SOAP 형태의 메시지를 SOAP 브리지에게 전송한다. 그러면 브리지 안의 SOAP 서버가 이 메시지를 받아서 SOAP

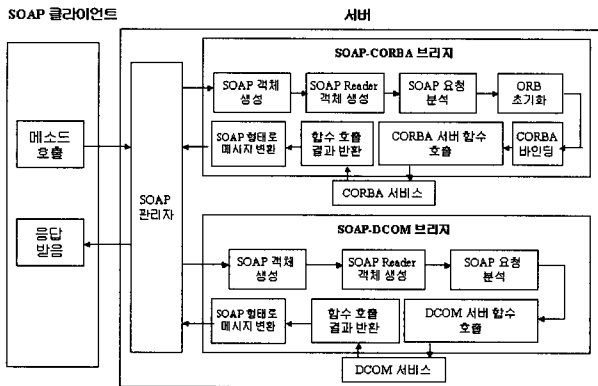
관리자를 호출하는데, 이는 SOAP 메시지 안에 있는 분산 객체의 종류를 분석하여 각각의 종류에 맞는 클라이언트 객체를 생성한다. 즉, 분석한 결과가 CORBA에 관한 것이면 CORBA 클라이언트를, DCOM에 관한 것이면 DCOM 클라이언트를 호출하여 그 매개변수 값을 전송한다. 그 값은 각 서버에게 넘겨지고, 연산 처리 후의 결과값은 각 클라이언트를 통해 SOAP 서버에게 다시 전달되는데, 이 때 그 값은 SOAP 메시지 형태로 재 변경된 후에 요청한 클라이언트에게 전송된다.



(그림 3) SOAP 브리지

3.3 SOAP 브리지 모듈

SOAP 브리지를 구성하는 모듈들은 (그림 4)와 같으며 시스템 별로 각 모듈은 다음과 같은 처리 과정을 통한다.



(그림 4) SOAP 브리지 모듈

먼저, CORBA 기반 시스템에서는 SOAP 메시지를 관리하기 위한 SOAP 객체를 생성한다. 그런 후에 SOAP 메시지를 받아들이기 위한 SOAP Reader, SOAP 응답을 해석하기 위한 SOAP Response 객체를 생성하며, 입력받은 SOAP 메시지를 해석한 후 CORBA 통신을 위해서 ORB를 초기화하고 바인딩(Binding)한다. 이후에 연결된 CORBA 서버의 원격객체 함수를 호출하고, 원하는 결과를 돌려 받으면, SOAP Response 객체가 이를 SOAP 메시지의 형태로 변환한 후 SOAP 클라이언트에게 전송한다.

DCOM 기반 시스템에서의 연산 처리 방식도 이와 비슷하지만, 원격객체를 사용하기 위해 ORB를 초기화하고 바인딩하는 과정 대신 서버 레지스트리에 등록되어 있는 컴포넌트를 클라이언트가 사용한다는 차이가 있다.

이러한 SOAP 브리지를 구현하기 위해 개발 툴로는 델파이 6.0을 사용하였고, CORBA 부분을 구현하기 위해서 Visi Broker 3.3을 ORB로 사용하였으며, DCOM을 구현하기 위해서 윈도우 COM/DCOM 런타임 라이브러리에 적합한 컴포넌트를 사용하였다.

4. 성능 실험

본 절에서는 성능 실험[5,9]을 통해 SOAP 브리지를 이용한 분산객체시스템 연동을 평가한다.

4.1 실험 환경

성능 실험을 위해 펜티엄 4-1.5GHz를 서버로 사용하였고, 펜티엄 3-1.5GHz를 클라이언트로 사용하였다. 또한 모두 윈도우즈 2000을 운영체제로 사용하였으며, 100Mbps 랜 카드로 연결하였다.

4.2 성능 실험 모델

본 절에서는 네 가지 기반의 시스템 모델을 설정하여 성능 실험을 하였다. 즉, 기존 분산객체시스템을 의미하는 CORBA-CORBA와 DCOM-DCOM 모델, 새로운 SOAP-SOAP 모델, 기존 분산객체시스템과의 연동을 위한 SOAP 브리지 모델이다. 그리고 각 시스템에 사용된 트랜잭션은 모든 성능 모델에 동일하게 적용되었으며, 각 모델의 실험 결과는 실험을 100번 반복한 후 그 평균값을 나타낸 것이다. 즉, 이 트랜잭션은 매개변수의 크기만큼 값을 하나씩 증가시켜 그 합을 구하는 동일한 원격 객체를 호출하는 것이며, <표 1>은 SOAP 브리지 모델에서의 검사점을 설명한다.

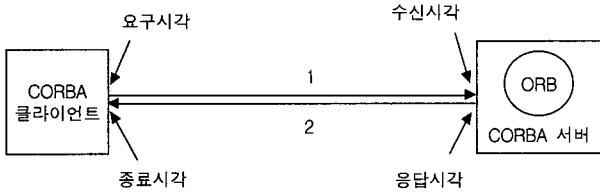
<표 1> SOAP 브리지 성능 모델의 검사점

검 사 점	의 미
요구시각	클라이언트가 서버에게 요청을 보낸 시각
서버 수신시각	서버가 요청을 받은 시각
브리지 시작시각	브리지의 수행이 시작되는 시각
브리지 종료시각	브리지의 수행이 끝나는 시각
객체 수신시각	객체가 요청을 받은 시각
객체 응답시각	객체가 응답을 보내는 시각
서버 응답시각	서버가 응답을 보내는 시각
종료시각	클라이언트가 응답을 받은 시각

4.2.1 CORBA 기반 시스템의 성능 실험 모델

CORBA 기반 시스템의 성능을 실험하기 위해서 (그림 5)

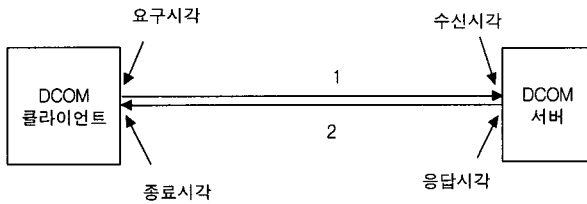
와 같은 모델을 사용하였다. 그림을 보면 CORBA 클라이언트와 ORB를 포함한 CORBA 서버가 있고, 시각을 단계별로 검사하여 연산처리에 걸리는 전체시간 뿐만 아니라 각 단계에서의 처리시간을 파악할 수 있게 한다. 아라비아 숫자로 표기한 부분은 각 구간을 의미한다.



(그림 5) CORBA 성능 모델

4.2.2 DCOM 기반 시스템의 성능 실험 모델

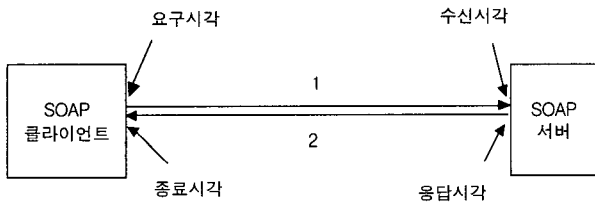
DCOM 기반 시스템에서 성능을 실험하기 위한 모델은 원격 객체에 접근하는 방식에 차이가 있을 뿐 앞에서 살펴본 CORBA 기반 모델과 유사하다.



(그림 6) DCOM 성능 모델

4.2.3 SOAP 기반 시스템의 성능 실험 모델

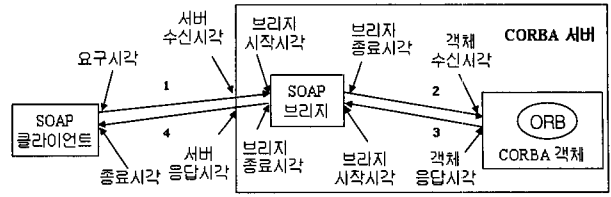
(그림 7)은 SOAP 기반 시스템에서 성능 실험을 하기 위한 모델이다. 이 시스템에서는 웹을 통해 서버에 존재하는 WSDL(Web Service Description Language)을 이용하여 원격에 있는 객체에 접근할 수 있다.



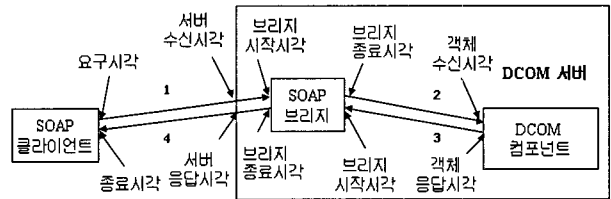
(그림 7) SOAP 성능 모델

4.2.4 SOAP 브리지 기반 시스템의 성능 실험 모델

(그림 8)과 (그림 9)는 본 논문에서 제안한 SOAP 브리지를 이용하여 분산객체시스템을 구성한 그림이다. 그림에서 보는바와 같이 SOAP 브리지를 기준으로 서버측은 CORBA나 DCOM 기반 시스템과 동일하게 구성되어 기존 분산객체시스템을 나타내고, 클라이언트 쪽은 SOAP 기반 시스템과 동일하게 구성되어 새로운 시스템을 의미한다.



(그림 8) SOAP-CORBA 브리지 성능 모델



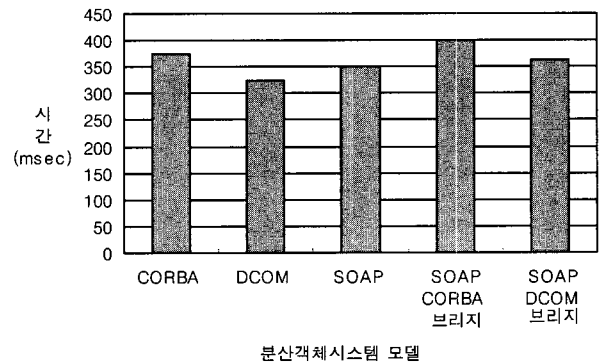
(그림 9) SOAP-DCOM 브리지 성능 모델

4.3 실험 결과 분석

본 절에서는 각 실험 모델을 기반으로 성능 실험한 내용을 분석하고, 이를 통해 SOAP 브리지 기반의 시스템으로부터 얻을 수 있는 장점과 특징들을 살펴본다.

4.3.1 트랜잭션 처리시간

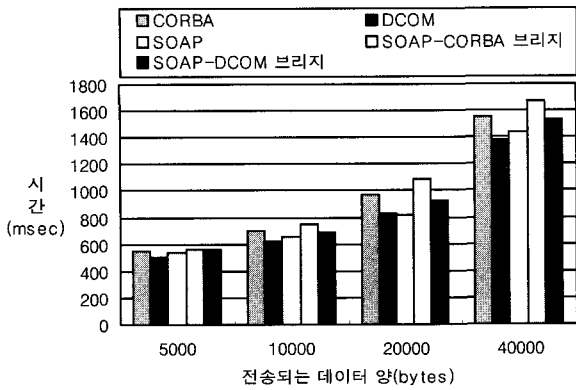
각 성능 모델들이 분산환경에서 동일한 트랜잭션을 처리하는 시간 즉, 서버 수신시각부터 서버 응답시각까지를 측정해 보았다. 그 결과는 (그림 10)과 같으며, CORBA나 DCOM 기반 시스템이 SOAP 브리지 기반 시스템과 연동할 때 발생하는 평균 오버헤드가 SOAP-CORBA는 9%, SOAP-DCOM은 2%로 성능 저하가 크지 않다.



(그림 10) 평균 트랜잭션 처리시간

4.3.2 전송되는 데이터 양에 따른 트랜잭션 처리시간

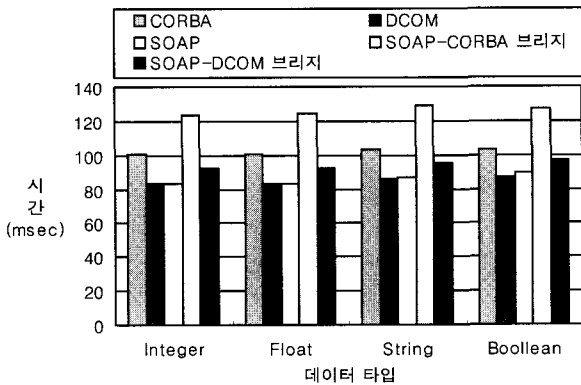
각각의 모델에서 전송되는 데이터 양에 따른 성능 실험을 측정해 보았다. 그 결과는 (그림 11)과 같으며, 측정된 결과를 보면 전송되는 데이터 양의 증가에 따른 평균 트랜잭션 처리시간 차이가 (그림 10)과 같은 비율로 나타난다. 즉, 데이터 양의 증가에 따른 평균 트랜잭션 처리시간 차이는 크지 않음을 알 수 있다.



(그림 11) 전송되는 데이터 양에 따른 평균 트랜잭션 처리시간

4.3.3 데이터 타입에 따른 처리시간

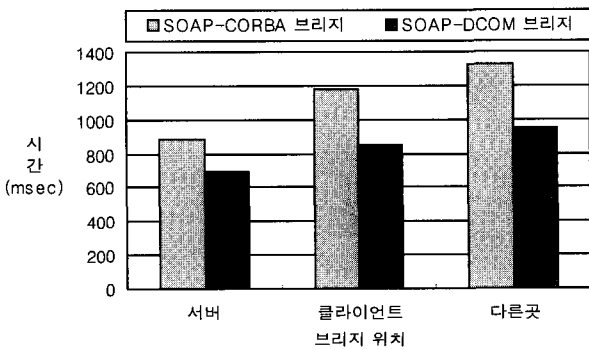
각 기반에서 데이터 타입에 따른 성능 실험 결과는 (그림 12)와 같고, 각 데이터 타입에 따른 평균 트랜잭션 처리시간은 동일한 비율로 나타나며 성능에 큰 영향을 주지 않음을 알 수 있다.



(그림 12) 데이터 타입에 따른 평균 트랜잭션 처리시간

4.3.4 SOAP 브리지 위치에 따른 트랜잭션 처리시간

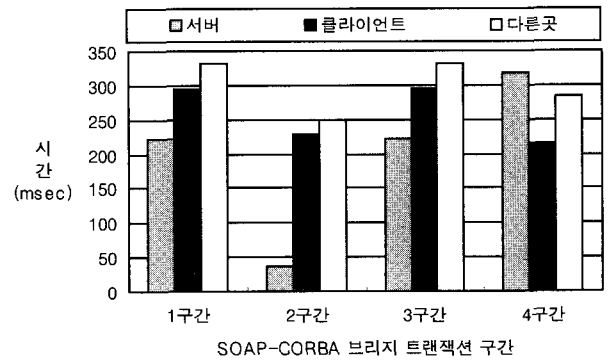
SOAP 브리지의 위치에 따른 트랜잭션 시간을 측정한 결과를 (그림 13)에서 보면, CORBA나 DCOM 모델에서 브리지가 서버에 있을 때 평균 트랜잭션 처리시간이 가장 적음을 알 수 있다.



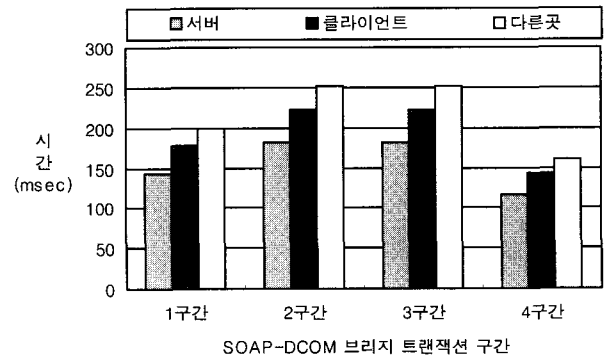
(그림 13) SOAP 브리지 위치에 따른 평균 트랜잭션 처리시간

4.3.5 SOAP 브리지 위치에 따른 구간별 트랜잭션 처리 시간

브리지 위치에 따른 연산 처리 시간을 구간별로 분석한 결과이다. 이는 (그림 8)과 (그림 9)에서 숫자로 나타낸 구간을 기준으로 검사한 트랜잭션 처리시간이다. 즉, SOAP 브리지를 이용하여 기존 시스템들과 연동할 때, SOAP-CORBA 시스템과 SOAP-DCOM 시스템의 특성을 알아보기 위한 성능 평가로써 위치에 따른 구간별 특성을 알아보는 것이다. 그 결과를 보면 CORBA 모델에서는 브리지가 서버에 있을 때 ORB를 초기화하고 바인딩한 후 서버를 요청하는 시간이 포함된 2단계가 상대적으로 적게 걸림을 알 수 있고, DCOM 모델에서는 비교적 균일한 분포를 나타낸다.



(그림 14) SOAP-CORBA 브리지 위치에 따른 구간별 평균 트랜잭션 처리시간



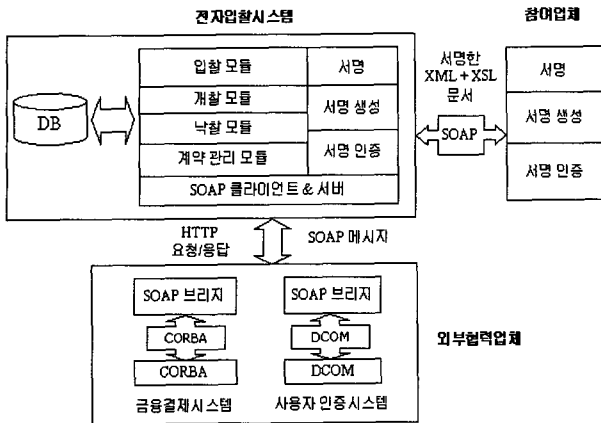
(그림 15) SOAP-DCOM 브리지 위치에 따른 구간별 평균 트랜잭션 처리시간

5. SOAP 브리지를 이용한 XML 기반 전자입찰 시스템 설계

본 절에서는 앞에서 설명한 SOAP 브리지를 실제 적용하기 위한 사례로 XML 기반 전자입찰 시스템을 설계한다.

5.1 시스템 구성도

(그림 16)을 보면 전자입찰 시스템은 입찰, 개찰, 낙찰 등과 같이 입찰에 필요한 기본적인 모듈들로 구성된다.

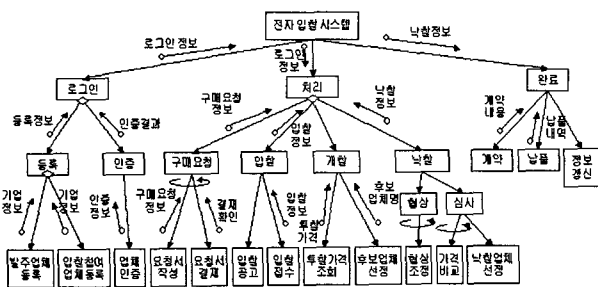


(그림 16) 시스템 구성도

또한, 이를 기반으로 참여업체들과 통신하기 위해서 SOAP을 이용하며, 기존의 SOAP을 지원하지 않는 외부 협력업체와의 연동을 위해 SOAP 브리지를 사용한다. 즉, 전자입찰시스템은 참여업체 입장에서는 서버가 되면서 외부 협력업체들과 협력하기 위해서는 클라이언트 역할을 하게된다.

5.2 프로그램 설계

전자입찰은 입찰 참여업체가 인터넷 기반의 전자입찰 시스템을 통해 입찰 업무를 수행하는 전 과정을 의미한다. 이를 처리하기 위한 주요 프로세스들은 (그림 17)에서 보는 바와 같다.



(그림 17) 프로그램 구조도

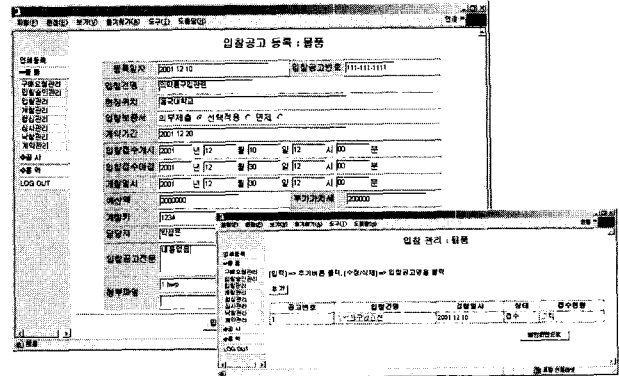
먼저, 발주업체나 입찰 참여업체는 업체등록을 한다. 발주업체가 입찰공고를 내면 입찰 참여업체들은 입찰서를 인터넷상으로 제출하고, 입찰이 마감되면 개찰을 실시한다. 이를 통해서 후보업체들을 선정하여 협상을 하는데 협상이 끝나면 최종 심사를 하고, 낙찰업체를 선정하여 통보한다. 낙찰업체와 계약관계가 성립되면 모든 입찰 업무가 마무리 된다.

6. SOAP 브리지를 이용한 XML 기반 전자입찰 시스템 구현

본 절에서는 앞에서 설계한 내용을 기반으로 구현한 화

면들을 살펴본다.

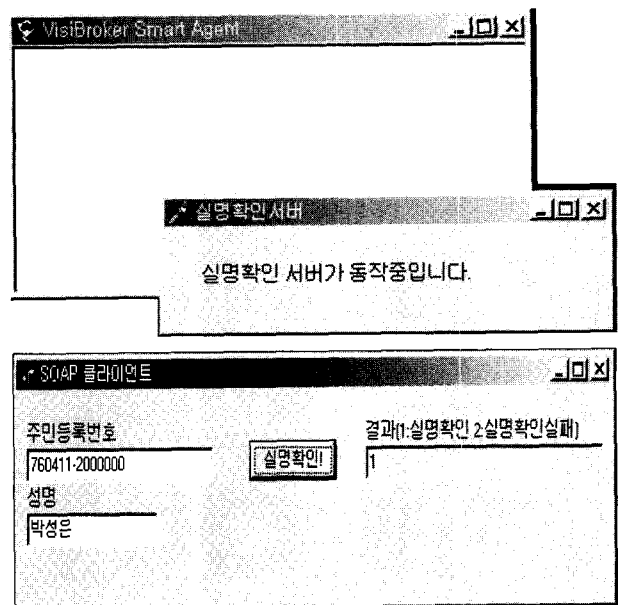
6.1 사용자 인터페이스



(그림 18) 입찰공고 화면

(그림 18)은 발주업체가 입찰공고를 하기 위해 등록할 수 있는 화면이다. 발주업체는 왼쪽 그림과 같은 사용자 인터페이스를 사용하는데 XML과 XSL을 사용한 웹 기반 인터페이스로 구현되었다. 발주업체가 등록한 내용은 데이터베이스에 저장되고, 그 리스트는 입찰 참여업체들이 확인할 수 있도록 설계되었다. 같은 방식으로 입찰에 관련된 사용자 인터페이스는 XML과 XSL을 이용한 웹 기반에서 설계되었으며, 발주업체의 입찰등록, 수정, 공고, 개찰, 협상, 심사 등의 기능과 참여업체의 입찰검색, 확인, 입찰참여, 협상, 계약 등의 기능이 구현되었다.

6.2 SOAP 브리지를 이용한 연동



(그림 19) ORB와 CORBA 서버 실행화면과 SOAP 클라이언트 화면

(그림 19)는 SOAP과 CORBA 기반의 시스템이 서로 연동되는 부분인 SOAP 브리지의 실행화면이다. 발주업체는 입찰에 참여한 사용자의 신원 확인을 위해 외부 인증기관의 서비스를 사용하게 된다. 인증 서비스가 CORBA 기반일 때, SOAP 기반 시스템과 연동하기 위해 SOAP 브리지가 필요하다. 발주업체에서는 SOAP 메시지를 사용하여 인증 서비스를 요청하게 되고, 인증 시스템에서는 브리지를 통해 응답을 받은 후 CORBA 서비스를 사용하여 사용자 인증을 한다. 이 때 필요한 것이 CORBA 서버와 ORB이며 그림의 왼쪽 화면에 나타나 있다. 그림의 오른쪽 화면은 브리지에 요청을 보낸 후 응답을 받은 결과 화면이다. 같은 방식으로 DCOM 서버와도 연동된다.

### 7. 결론 및 향후 연구

본 논문에서는 기존 분산객체시스템과 SOAP 기반 시스템의 상호 운용성을 높일 수 있는 SOAP 브리지를 구현하였고, 이를 실제 적용한 예로 XML 기반 전자입찰 시스템을 설계하고 구현하였다.

본 논문에서 제안한 SOAP 브리지는 여러 성능 실험을 통해 다음과 같은 장점과 특징이 있음을 알 수 있다. 첫째, SOAP 브리지를 이용하면 기존 분산객체시스템을 그대로 활용하면서 SOAP 기반의 시스템과 연동할 수 있다. 둘째, SOAP 브리지를 이용하여 시스템을 연동할 때 그 오버헤드가 크지 않음을 알 수 있다. 셋째, 여러 성능 실험의 결과를 통해 SOAP 브리지는 분산 시스템에 매우 적절하며, 이러한 SOAP 브리지를 활용하여 설계하고 구현한 XML 기반 전자입찰 시스템은 XML과 SOAP을 기반으로 하기 때문에 기존 전자상거래 시스템이 갖는 표준화·확장성의 문제를 해결할 수 있다.

향후에는 EJB와 같은 다른 분산객체시스템과의 연동도 연구할 것이며, SOAP 브리지를 포함한 전자입찰 시스템의 성능 평가에 대한 연구도 할 것이다.

### 참 고 문 헌

[1] Brian W. Beach, "Connecting Software Components with Declarative Glue," Proc. of the 14th International Conference on Software Engineering, Melbourne, Australia, pp.120-137, 1992.  
 [2] George M. Doss, "CORBA Developer's Guide with XML," Wordware, 1999.  
 [3] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, "Design Patterns : Elements of Reusable Object-Oriented Software," Addison-Wesley, Massachusetts, 1995.  
 [4] Dimitri Konstantas, "Object Oriented Interoperability," Lec-

ture Notes in Computer Science, Vol.707, pp.80-102, Springer, 1993.  
 [5] Carlos ORyan, et. al., "The Design and Performance of a Pluggable Protocols Framework for Real-time Distributed Object Computing Middleware," Proc. of the Middleware 2000, New York, USA, pp.372-395, 2000.  
 [6] James M. Purtilo and Joanne M. Atlee, "Module Reuse by Interface Adaptation," Software Practice and Experience, Vol.21, No.6, pp.539-556, 1991.  
 [7] Kent Sharkey and Scott Seely, "SOAP Cross Platform Web Service Development Using XML," Prentice Hall, 2001.  
 [8] Glenn Smith, John Gough and Clemens Szyperski, "A Case for Meta-Interworking : Projecting CORBA meta-data into COM," Proc. of the Technology of Object-Oriented Language, Melbourne, Australia, pp.242-253, 1998.  
 [9] Michiharu Takemoto and Takayuki Nakamura, "Performance Evaluation of a Fault-tolerant Mechanism Based on Replicated Distributed Objects for CORBA," Proc. of the Object-Oriented Real-time Distributed Computing, Magdeburg, Germany, pp.95-102, 2001.  
 [10] Satish R. Thatte, "Automated Synthesis of interface adapters for reusable classes," Proc. of the POPL '94, Portland, Oregon, USA, pp.174-187, 1994.  
 [11] Brain E. Travis, "Microsoft XML and SOAP Programming for BizTalk Servers," Microsoft Press, 2000.  
 [12] Daniel M. Yellin and Robert E. Storm, "Protocol Specifications and Component Adaptors," ACM Transactions on Programming Languages and Systems, Vol.19, No.2, pp. 292-333, 1997.  
 [13] Sun's CORBA & Middleware Strategy, <http://www.sun.com/software/neo/>.  
 [14] SOAP 1.1 Spec., <http://www.w3.org/TR/SOAP>.  
 [15] ebXML Spec., <http://www.ebxml.org>.  
 [16] Recent Data Sheet, <http://www.roguewave.com>.  
 [17] 이호섭, 홍충선, "CORBA와 XML의 연동을 위한 응용 웹 서버 구조 성능 분석", 한국정보과학회 추계 학술발표논문집, 제28권 제2호, pp.577-579, 2001.  
 [18] 이호섭, 홍충선, "웹 응용서버와 SOAP을 이용한 CORBA와 XML의 연동구조", 멀티미디어저널, 제1권 제2호, 2001.



### 박 성 은

e-mail : pse76@dgu.ac.kr

2000년 동국대학교 컴퓨터공학과(학사)

2002년 동국대학교 컴퓨터공학과(석사)

2002년~현재 동국대학교 컴퓨터공학과

(박사과정)

관심분야 : XML 및 웹, 스토리지시스템,

데이터베이스



**김 신 우**

e-mail : purian@dgu.edu

1997년 동국대학교 컴퓨터공학과(학사)

2000년 동국대학교 컴퓨터공학과(석사)

2000년~현재 동국대학교 컴퓨터공학과  
(박사과정)

관심분야 : XML 및 웹, 스토리지시스템,  
데이터베이스



**이 용 규**

e-mail : yklee@dgu.edu

1986년 동국대학교 전자계산학과(학사)

1988년 한국과학기술원 전산학과(석사)

1996년 Syracuse University(전산학박사)

1978년~1983년 정보통신부 국가공무원

1988년~1993년 한국국방연구원 선임연구원

1996년~1997년 한국통신 선임연구원

1997년~현재 동국대학교 컴퓨터멀티미디어공학과 교수

관심분야 : XML 및 웹, 스토리지시스템, 데이터베이스