WSDL과 CORBA IDL

*황의철, **정선태
*광주여자대학교 멀티미디어학과,
**숭실대학교 정보통신전자공학부
전화 : 062-950-3727/핸드폰 : 011-9622-0342

WSDL and CORBA IDL

Eui-Chul Hwang, Sun-Tae Chung
Dept. of Multimedia, Kwangju Womem's Univ.
Dept. of of Electronics Eng., Soongsil Univ.
E-mail : euhwang@mail.kwu.ac.kr

Abstract

CORBA has been a popular middleware, but recently Web Services has been emerging as a promising web middleware since it uses Internet established standards such as URL, HTTP, XML, and etc.

CORBA uses IDL for descrbing CORBA object interface and Web Services uses WSDL for its description. Thus, in order to port or integrate CORBA objects into Web Services, one needs to understand the relationship between CORBA IDL and WSDL.

In this paper, we analyze IDL and WSDL, and compare between two. It turns out that there exists a direct mapping between two, and that understanding operational environments and logic of two technologies is much more important in converting or integrating CORBA objects into Web services successfully.

I. 서론

미들웨어(middleware)는 운영체계와 응용 사이에 위치하여 클라이언트/서버 분산 응용이 필요로 하는 통신, 서비스(객체 또는 프로시저) 발견, 보안 등의 기능을 제공하는 소프트웨어 플랫폼을 말한다. 미들웨어를 이용하는 경우, 분산 응용 개발자는 분산 응용에 필요한 통신, 서비스 발견, 네트워크 보안 기능 등 신뢰성 있는 구현이 쉽지 않은 기능들을 미들웨어가 제공하므로, 분산 응용 논리 개발에 주력하기만 하면 되어서 개발이 보다 용이하고 빠르게 된다[1].

미들웨어에는 RPC, DCOM, RMI, CORBA, MQ Series, MSMQ 등 여러가지가 있는 데, 모든 언어와 운영체제를 지원하는 CORBA가 현재 미들웨어에서 많이 선호되어 사용되고 있다[2].

웹서비스는 웹에서 사실상 표준이 된 URL, XML, 및 HTTP 등에 기반하고, 프로시저 형태뿐만 아니라 메시지(문서) 형태의 통신방식 모두 지원하는 등의 여러가지 장점으로, 최근 들어 웹의 폭발적인 성장에 따라 유망한 웹기반 미들웨어로 각광을 받고 있다. 따라서, 현재 CORBA 등의 종래 미들웨어로 구축된 분산 응용을 웹서비스로 포팅 또는 통합하는 것이 적극적으로 추진되고 있는 추세이다[3].

이에 따라, 논문에서는 미들웨어 중 가장 중요한 코바와 웹서비스간의 변환과 연동을 위해, 먼저 코바의 인터페이스 기술언어인 IDL(Interface Description Language) 과 웹서비스 인터페이스 기술 언어인 WSDL(Web Services Description Language)에 대해 살펴보고 비교 분석하며, 코바 객체를 웹서비스에로 전환 또는 통합할 경우에 고려해야 할 점 등을 살펴본다.

II. 코바와 웹서비스

2.1 코바(CORBA)

코바는 OMG(Object Management Group)가 1991부터 규정한 미들웨어 표준안으로 현재 버전은 3.0.2이다[4]. 코바에서 코바 객체의 인터페이스를 기술하기위해,

언어 중립적인 IDL을 사용한다. 코바의 핵심은 ORB(Object Request Broker)로, 이 ORB가 클라이언트가 서버에 있는 원격 객체의 위치에 독립적으로 그 객체의 메소드를 호출할 수 있도록 하는 객체 버스의 역할을 한다. ORB는 보통 라이브러리 형태로 구현되어 지원된다.

코바 응용의 실제 구현시에는, IDL 로 기술된 인터페이스는 IDL 컴파일러를 통해 해당 언어(C++, JAVA 등)의 실체(객체, 변수, 함수 등)로 매핑되고, 매핑된 헤더 파일과 인터페이스 구현 소스는 ORB 라이브러리 및 응용과 같이 컴파일되어 클라이언트 또는 프로그램을 만들게 된다.

코바는 전송 프로토콜로 GIOP(General Inter-ORB Protocol) 프로토콜을 규정하고 있으며, 실제 TCP/IP 프로토콜 스택에서의 GIOP 구현인 IIOP(Internet Inter-ORB Protocol)를 지원한다. 다른 프로토콜에의 매핑은 지원하지 않는다. 코바는 IDL 데이터 타입의 전송 인코딩방식으로 바이너리 인코딩인 CDR(Common Data Representation)을 규정하고 있다. GIOP 에서는 데이터의 타입 정보는 보내지 않는다. 이는 데이터 수신 처리하는 클라이언트 또는 서버가 데이터 타입을 안다고 가정하고 있기 때문이다.

2.2 Web Services

웹서비스는 웹에서 사실상 표준으로 받아드려진, URL, HTTP 및 XML 등에 기반하고 있으며, 대형 소프트웨어 개발 회사인 IBM, 마이크로소프트, SUN, HP 등이 적극적으로 지원하고 있기 때문에 현재 인터넷상에서 분산 응용 구축에 매우 유망한 웹 미들웨어로 각광을 받고 있다[3]. 웹서비스 구조에서는 클라이언트는 메시지 형식으로 만든 XML 문서를 사용하여 서버(웹서비스)에 요청을 보내고, XML 메시지 형식으로 응답을 받는다. 웹서비스 표준은 메시지 형식을 규정하고, 전달되는 메시지에 대한 인터페이스를 기술하고 웹서비스를 게시하고 발견하는 메카니즘을 정의한다. 웹서비스를 구성하는 요소는 WSDL, SOAP(Simple Object Access Protocol), UDDI(Universal Description, discovery, and Integration) 이다. WSDL은 웹서비스 인터페이스를 규정한다. SOAP은 웹서비스 사용을 위한 메시지 전송 프로토콜이다. UDDI는 웹서비스를 게시하고, 찾는 절차를 규정한다. 현재, WSDL, SOAP 등은 W3C(World Wide Web Consortium)에서 표준 사양화 작업을 진행하고 있으며, UDDI 는 OASIS UDDI member section에서 표준화 작업을 진행하고 있다.

데이터만 전송되는 CORBA, DCOM, RMI 등의 경우에, 해당 데이터 타입을 이해하지 못하는 수신자는 수신된 데이터를 이해할 수없어 이를 더 이상 처리하지 못하게 된다. 하지만, 데이터 구조를 스스로 기술할 수 있는 XML을 사용하여 표현되는 웹서비스의 메시지는 데이터뿐만 아니라, 데이터 표현까지 포함되므로 XML 파싱이 가능한 어느 누구도 메시지 내용을 이해할 수 있기 때문에, 인터넷상의 여러 응용 사이의 데이터 교환에 매우 유리하다. 또한, 특정 포트가 규정되어 있지 않아 파이어월 통과가 쉽지 않은 코바 객체 서비스에 비해, 웹서비스는 보통 SOAP 메시지를 HTTP 로 전송하는 데 HTTP는 파이어월을 통과하므로 서비스 사용이 용이하다.

2.3 코바와 웹서비스 비교

코바는 객체 지향 미들웨어로 메소드 호출을 통해 서비스를 제공하므로 RPC 형식만 지원하나, 웹서비스는 SOAP 메시지 전달을 통해 클라이언트와 서버가 대화하므로 RPC 형식 이나 문서 전달 형식 모두 지원이 가능하다. 웹서비스는 객체의 개념 지원은 미약하며, 데이터 전달시에 데이터의 타입 정보까지 전달한다. 웹서비스에서는 코바와 다르게 서비스의 위치를 객체 참조가 아닌 URI을 이용한다. 또한, 웹서비스는 코바와는 달리 데이터 표현에 XML 텍스트 인코딩을 사용하므로, 메시지의 크기가 커지고, 메시지의 XML 마샬링 및 파싱 처리가 필요하므로 코바보다는 처리 속도가 느리다[5]. 반면, 웹서비스는 인터넷 표준인 URL, HTTP 및 XML 등을 사용하므로, 전용 전송 프로토콜(IIOP)을 사용하는 코바에 비해 현재 인터넷 상에서 전개가 유리한 편이다. 코바와 웹서비스에 대한 보다 상세한 비교는 [6]를 참조하라.

III. CORBA IDL 과 WSDL

3.1 CORBA IDL(Interface Description Language)

CORBA IDL은 구현과 독립적이며, 언어 중립적인 객체 인터페이스 사양이다. IDL 은 객체 인터페이스를 규정하기 위해 필요한 데이터 타입, 속성, 메소드, 예외 등에 대한 문법과 구문에 대해 정의한 것이다. CORBA IDL 규정에 대한 보다 자세한 내용은 [4]을 참조하기로 하고, 여기서는 CORBA IDL 규정의 주요 내용만 간단히 살펴 본다.

(1) 데이터 타입

IDL 이 지원하는 데이터 타입은 기본 타입과 사용자 정의 타입이 있다. 기본타입에는 상수 타입, 정수 타입, 실수 타입, 문자, 스트링, 'boolean' 타입, 바이너리 데이터 전송을 위한 'octet' 타입, 그리고 어떤 타입에도 사용가능한 'any' 타입 등이 지원된다.

사용자 정의 데이터 타입에는 C 의 'typedef' 같이 데이터 타입의 이름을 재지정하는 'named type', 'enumeration', 'structure', 'union', 'array', 가변 길이 벡터를 표현하기 위한 'sequence' 등이 지원되며, 재귀적 선언이 허용된다.

(2) Interface

IDL 인터페이스는 객체의 인터페이스을 기술하는 것으로 속성과 operations, 에러 상태를 지원하기 위한 예외 규정 등을 포함한다. IDL 인터페이스는 다중 상속 및 다층 상속이 지원된다.

(3) Operations

오퍼레이션은 C/C++ 의 함수 선언과 유사하나, 파라미터가 오퍼레이션 호출에 필요한 파라미터(입력파라미터)인지, 오퍼레이션 결과로 전달되는 파라미터(출력 파라미터)인지, 또는 입출력 모두에 사용되는 파라미터인

지를 명확히 지정한다. 리턴이 필요치 않는 Oneway 오퍼레이션도 지원된다.

(4) 상속(Inheritance)
모든 IDL 인터페이스는 암시적으로 타입 Object 으로 계승되며, 다중상속 및 다층 상속을 모두 지원한다.

3.2 WSDL(Web Service Description Language)

WSDL 은 XML 로 웹 서비스를 어떻게 기술하여야 하는 것을 정의하는 사양이다. WSDL을 이용하여 클라이언트는 웹서비스를 발견하고 웹서비스가 제공하는 공개 함수를 호출할 수있다. 현재 WSDL 은 버전 1.2 가 working draft 상태로 발표되어 있다[7].

WSDL 사양은 크게 WSDL 사양은 크게 definitions, types, messages, interface(버전 1.1에서는 portType), binding, service 등의 6개 요소로 구성된다.

WSDL 문서는 정의로 구성된다. 이들은 서비스를 규정하며, 서비스는 하나 또는 그 이상의 네트워크 종점, 즉 포트로 구성된다. 각 포트는 특정 바인딩과 결합되어 있다. 바인딩은 특정 프로토콜과 엔드포인트 주소를 인터페이스에 매핑하는 것을 규정한다. 인터페이스는 하나 이상의 오퍼레이션으로 구성되어 있다. 각 오퍼레이션은 추상적인 메시지 집합으로 구성된다. 메세지는 오퍼레이션 사이에 교환되는 데이터를 기술한다. 각 메세지는 하나 또는 그 이상의 데이터를 담고 있으며, 이들은 type 으로 정의된다. WSDL은 특정 타이핑 시스템을 제한하고 있지 않으나, 보통 W3C XML Schema 사양을 기본 사양으로 사용한다. XML Schema 는 현재 W3C 의 Recommendation (2001년 5월 2일자) 이며, XML Schema에서 지원하는 데이터 타입은 simple type과 complex type 으로 나뉜다. Simple type에는 정수형, 실수형, 스트링, 바이너리 데이터 지원 외에 시간 이름 등을 지원하는 데이터 타입이 지원된다. Complex Type은 simple type으로 표현할 수 없는 복잡한 데이터 타입을 지원하는 데 사용된다.

WSDL 은 다양한 프로토콜로의 매핑을 지원한다. 이 때문에 WSDL 은 IDL에는 없는 바인딩이 정의되어 있다. WSDL 의 프로토콜 바인딩에는 SOAP, HTTP GET/POST, MIME 등이 지원된다.

Ⅳ. IDL과 WSDL 비교

OMG의 CORBA-WSDL/SOAP FTF(Finalization Task Force) 에서 현재(2003년 6월 13일 현재) IDL-WSDL 매핑 사양을 표준화 작업중에 있다[8]. 이 매핑 작업은 SOAP 1.1 과 WSDL 1.1 버전에 기반하고 있으며, WSDL 과 SOAP 에 사용된 데이터 타입은 XML Schema 에 기반한다.

SOAP 의 현재 버전은 1.2 이며 W3C의 표준화 상태는 Proposed Recommendations 이다. WSDL 현재 버전은 1.2 이며, 1.2 버전의 표준화 상태는 Working Drafts 이다. 따라서, CORBA-WSDL/SOAP FTF 의 작업은 차후 다시 수정될 전망이다. 그러나, SOAP 1.1 과 1.2 의 차이는 크지 않으며, WSDL 1.1 과 1.2 의 차이중 주목할 만한 것은 portType 이 interface 로 바뀌었다는 정도이다. 따라서, CORBA-WSDL/SOAP FTF 의 표준화 사양이 크게 변할 것으로 예상되지는 않는다.

4.1 IDL to WSDL mapping

WSDL 은 사실상 IDL+binding+ service location 으로 볼 수있다. CORBA-WSDL/SOAP FTF 사양에 의하면, CORBA IDL 에 규정되어 있는 데이터 타입, 인터페이스, 속성, 예외 등의IDL 실체는 WSDL 매핑이 모두 명확하게 규정되어 있다.

또한, CORBA IDL에 없는 WSDL 의 SOAP 바인딩도 잘 규정되어 있다. 다만, 웹서비스 endpoints 주소 지정은 구현에 따라 다르고 웹서비스/코바 브릿지 와 코바 서버의 토폴로지에 의존하므로, 이에 대한 정보는 IDL로부터 획득하기 힘들기 때문에 CORBA-WSDL/SOAP FTF 사양에서 규정하지 않고 있다.

이상에서 보면, IDL to WSDL mapping 은 잘 규정되어 있음을 알 수있다. 이러한 매핑 규칙의 자세한 사항은 [8]을 참조하고, 여기서는 간단히 중요한 몇가지 사항에 대해서만 살펴 보기로 한다.

(1) 데이터 타입
IDL에서 지원하는 데이터 타입은 WSDL에서 지원되는 데이터 타입으로 매핑된다.

(2) Object Reference
코바에서는 객체를 참조할 때, object reference를 사용한다. 반면, 웹 서비스에서는 서비스는 URI 로 참조된다. 따라서, object reference 는 URI 의 시퀀스로 매핑된다.

(3) Interfaces
IDL Interface 는 3가지 용도로 쓰인다. 첫째, 타입 정의의 명명 공간; 둘째, 오퍼레이션의 집합; 셋째, 타입. 3가지 모두의 경우 해당하는 매핑이 잘 정의되어 있다. 오퍼레이션 바인딩으로서의 인터페이스만을 보다 자세히 설명하면 다음과 같다.

이경우의 interface 는 WSDL 의 interface(portType)으로 매핑되며, 오퍼레이션들은 WSDL의 메시지 타입으로 매핑된다.

각 IDL 오퍼레이션은 'invocation', 'response', 'fault for potential system exception' 등의 3가지 메시지로 매핑된다. one way 오퍼레이션의 경우는 'response' 메세지가 없다. Invocation 메세지는 'in' 'inout' arguments 로 구성되며, response 멧세지는 'out', 'inout' arguments 와 함께, 리턴값을 포함한다. System exception 메시지는 반드시 코바 system exception을 복귀한다. Attributes 는 'get/set' accessor 오퍼레이션으로 매핑된다. Read-only attributes 는 'get' 오퍼레이션만으로 매핑된다.

(4) Interface inheritance
IDL 에서는 다중 상속을 지원한다. WSDL 에서는 다중 상속 개념이 지원되지 않으므로, 인터페이스 상속

은 부모 인터페이스에 선언된 오퍼레이션의 반복으로 매핑된다.

(5) Exceptions
IDL exceptions 은 struct 와 같이 contructed type 으로 매핑된다.

(6) SOAP bindings
2가지 형태의 SOAP 바인딩이 지원된다. SOAP RPC-style 바인딩, WS-I basic profile conformant 바인딩이다.

(7) Service Endpoints
웹서비스 endpoints 는 구현에 따라 다르며, 웹서비스/코바 브릿지 와 코바 서버의 토폴로지에 의존한다. 이 정보는 IDL로부터 획득하기 힘들어 사양의 범위밖으로 규정되어 있다.

4.2 CORBA 객체의 웹서비스 통합시 고려사항

웹서비스는 XML 기반이고 XML은 텍스트 기반이므로, 웹서비스 메시지는 바이너리 인코딩을 사용하는 코바보다 메세지 크기가 커지게 된다. 또한, XML 메시지를 마샬링하고 파싱하는 데 시간이 필요하므로, 코바 응용보다 웹서비스의 성능이 떨어지는 것은 잘 알려진 사실이다[5]. 그러나, 성능이 크게 문제되지 않은 시스템의 경우에는 이는 큰 문제가 아니다. 코바 객체를 웹서비스로 포팅하거나 통합하는데 있어서, 코바 객체 인터페이스 IDL의 직접적인 WSDL 변환은 앞 4.1 절에서 살펴본 바와 같이 별다른 어려움이 없다. 그러나, 코바 기반 응용을 웹서비스 응용으로의 전환하거나 통합하는 것은 고려하여야 할 사항들이 적지 않다. 다음은 고려하여야 할 몇가지 사항이다[9,10].

(1) 코바 동작 배경 및 웹서비스 동작 배경에 한 고려
기본적으로 코바는 객체 지향 모델 기반이나, 웹서비스는 객체 모델을 별로 지원하지 않는다. 또한, 객체나 실체 참조에 위치 투명한 객체 참조(Object reference)를 사용하는 코바와는 다르게, 웹서비스는 URI를 사용한다. 코바에서 객체에 대한 참조자를 네이밍 서비스(naming service)를 통해 가져 오는 데 반해, 웹서비스에서는 UDDI를 통해 해당 웹서비스를 찾는다. 코바나 웹서비스에서 신뢰성있는 응용 구축을 위해, 예외 상황 처리가 지원된다. 그러나, 코바에서 응용에 관련된 예외는 웹서비스의 Fault 로 변경하기 쉬우나, 객체 모델에 근거한 객체 시스템의 예외는 웹서비스에서는 해당하는 의미를 발견하기가 쉽지 않다.

(2) 코바 응용논리와 웹서비스 응용 논리에 대한 고려
코바 에서는 객체는, 세션 작업시에만 생성되어 세션이 끝나면 소멸되는 객체인 세션 객체와 서비스가 진행되는 동안 계속 살아있는 서비스 객체로 구분될 수 있다. 예를 들어, 팩토리 객체, 네이밍 서비스 객체 등은 서비스 객체이다. 만일, 코바 팩토리 객체등과 같은 서비스 객체를 웹서비스로 구현하는 경우, 따라서 클라이언트가 세션 객체의 생성·소멸을 제어하도록 하는 사용 문맥을 따르도록 하면, 다른 모든 코바 세션 객체들도 웹서비스로 구현되어야 할 필요가 있다. 그러나, 이 경우 객체간에 전달되는 통신 프로토콜의 자세한 내용을 클라이언트부터 감추기 어렵기 때문에 내부의 분산 응용 서비스 논리가 노출될 수있다. 그러므로, 경우에 따라서는 코바 응용 서비스 전체를 하나의 웹서비스로 구축하여야 할 필요가 존재한다.

V. 결론

본 논문에서는 코바 객체를 웹서비스로 포팅 또는 통합 하는 경우에 고려하여 할 사항을 알아보기 위해, 코바 객체 인터페이스를 기술하는 IDL과 웹서비스 인터페이스를 기술하는 WSDL를 각각 살펴 보고 비교하여 보았다. 살펴본 결과, IDL의 WSDL 직접적인 매핑은 별다른 어려움이 없으나, 코바와 웹서비스 동작 환경, 코바 응용 논리와 웹서비스 응용 논리 등의 고려가 더 중요하고 이에 대한 면밀한 분석이 필요함을 알 수 있었다.

참고문헌

[1] R. Orfali, et. al., *The Essential Client/Server Survival Guide, 2nd ed.*, John Wiley & Sons, INC., 1996.

[2] S. vinoski, "CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments," IEEE Comm. vol.35, no.2, pp.46-55.

[3] J. Roy and A. Ramanujan, "Understanding Web services," IEEE IT Professional , vol. 3, issue 6, Nov.-Dec. 2001, pp. 69-73.

[4] Object Management Group, *Common Object Request Broker Architecture: Core Specification*, OMG Document formal/02-12-06 .

[5] R. Elfwing, et. al., "Performance of SOAP in Web Service Environmnet Compared to CORBA," APSEC'02, 2002.

[6] A. Gokhale, et. al., "Reinventing the Wheel? CORBA vs. Web Services", Practice and Experience Track, Eleventh International Conference on World Wide Web (WWW2002), Honolulu, Hawaii, May 7-11, 2000.

[7] World Wide Web Consortium Working Drafts, "Web Services Description Language Part 1, 2, 3," http://www.w3.org/TR/

[8] OMG FTF 문서, "CORBA to WSDL/SOAP Interworking Specification," Jan. 2003, http://www.omg.org/techprocess/meetings/schedule/CORBA-WSDL_SOAP_FTP.html

[9] S. Vinoski, "Web Services Interaction Models - Part 1: Current Practice," IEEE Internet Computing, vol.7, no.3, May/June, 2002, pp. 89-91.

[10] S. Vinoski, "Web Services Interaction Models - Part 2: Putting the Web into Web Servicese," IEEE Internet Computing, vol.7, no.3, July/August, 2002, pp..90-92.