

# Java와 ATL 웹 서비스를 이용한 애플리케이션 개발

김병규<sup>0</sup>, 박우전  
한남대학교

{s97083<sup>0</sup>, wjpark}@ce.hannam.ac.kr

## Development of Applications using Java OR ATL Web Services

Byung-Kyu Kim<sup>0</sup> Woo Jun, PARK  
Dept. of Computer Engineering Hannam University

### 요 약

웹 서비스는 기존의 컴포넌트를 대체할 새로운 기술이지만 그 유용성에 비해 아직 사용예가 많지 않고 개발할 때의 장단점이나 문제점도 많이 알려지지 않았다. 본 논문에서는 웹 서비스를 이용한 웹 3D뷰어를 개발하여 웹 서비스 개발방법과 이점을 알아내고 앞으로의 개발 방향등을 논할 것이다.

### 1. 서 론

지금까지 소프트웨어 재사용을 위하여 여러 가지 컴포넌트 기술(COM, CORBA 등)들이 나왔지만 몇 가지 문제점이 있었다. 첫째로 각각의 기술들이 서로 호환이 되지 않으며 둘째로 컴포넌트 기반의 프로그래밍 방법은 구현(버전관리, 유지보수 등)이 어렵다. 셋째로 컴포넌트의 배포가 직접 카피해 오거나 제공해 주는 서버의 주소를 미리 알고 있어야 하고 사용자가 필요한 컴포넌트를 검색하여 바로 가져올 수 없는 등의 문제점이 있다.

웹 서비스는 이런 컴포넌트 기술의 문제들을 해결해주는 웹을 이용한 서비스 기술이라 정의할 수 있다.

웹 서비스는 인터넷 문서교환 표준인 XML을 사용하여 자료를 교환하며 80포트를 사용하기 때문에 방화벽을 넘어서 서비스가 가능하다. 또한 UDDI를 통한 컴포넌트의 검색이 가능하여 필요한 컴포넌트를 언제든지 검색하여 사용할 수 있다. 여기에 기존의 COM과 EJB, CORBA와 같은 컴포넌트 기술들을 웹 서비스와 접목시킬 수 있어 컴포넌트 사이의 호환성 문제를 해결하였다.

이러한 장점 때문에 개발자들이 많이 사용하는 Java 언어를 개발한 선 마이크로시스템즈나 마이크로소프트 같은 언어 개발 회사들이 자사들의 언어인 Java, C++, C#, VISUAL BASIC언어들로 웹 서비스를 개발할 수 있도록 웹 서비스 개발 툴들을 내놓고 있다. 본 논문에서는 이러한 웹 서비스를 이용한 WEB3D애플리케이션을 작성하여 웹 서비스를 이용하는 방법과 장점, 단점을 논한다. 본 논문의 2절에서는 웹 서비스를 지원하는 대표적인 언어인 Java와 ATL을 사용하여 Java 웹 서비스와 ATL웹 서비스를 구현함으로써 개발 환경과 개발 방법 등을 비교, 분석하고 3절에서는 클라이언트로 제3의 언어인 C#을 이용하여 웹 서비스를 이용한 그래픽 뷰어를 만들어 보았

다. 그리고 4절 결론에서는 향후 웹 서비스의 개발과 효율적인 이용에 관해 논하고자 한다.

### 2. 웹 서비스의 구현 기법

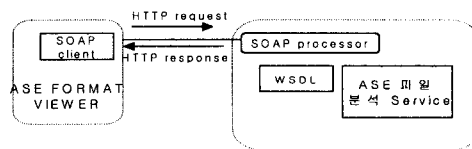


그림 1

그림 1은 구현된 웹 서비스 상에 개발할 애플리케이션의 개념도이다.

#### 2.1 Java 웹 서비스의 구현

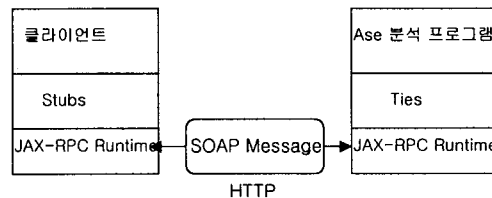


그림 2

Java 웹 서비스인 JAX-RPC 서비스 구조[1]는 그림[2]과 같다. 본 논문에서 개발하는 서비스도 다음의 구조를 따른다. Java 웹 서비스를 구현 해주기 위해선 jdk1.3 이후 버전과 Apache Soap2.0, Tomcat, Xerces XML Java Library, Java Mail을 설치하여 사용하며 지금은 선 마이크로시스템 사에서 위의 툴과 라이브러리들을 묶어서 하나의 jwsdp라는 패키지로 구성을 하였다. 이 중 soap은 웹 서비스에서 사용하는 프로토콜이고 Xerces XML Java Library는 Java에서 XML을 사용하기 위한 파서 이다. 웹 서비스는 문서교환은 XML로 전송은 soap프로토콜을 사용

한다. Java 웹 서비스는 JAXM과 JAX-RPC로 구별되는데 본 논문에서는 JAX-RPC로 구현하였으며 그림 [3]에 나타난 서비스가 ASE파일 포맷을 분석하는 ASEAnalysis라는 메소드를 서비스 해주는 웹 서비스이다.

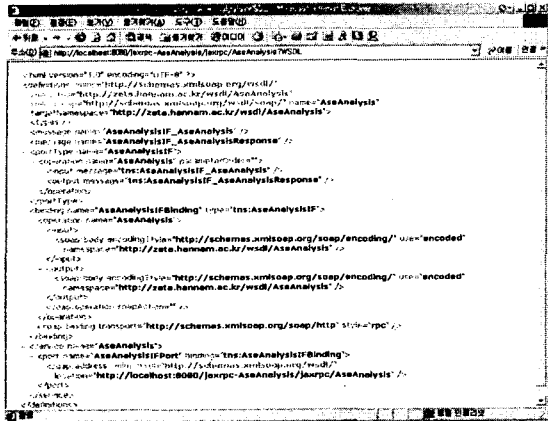


그림 3

그림 3는 완성된 ASEAnalysis 웹 서비스의 WSDL문서이다. 웹 서비스를 만들어서 배포에 성공하면 다음과 같은 WSDL 문서가 나타난다. 3절에서는 이 WSDL 문서를 참조하여 클라이언트를 작성한다. WSDL 문서는 다음의 내용을 담고 있다[3].

- 웹 서비스에서 인식하는 데이터 형식
- 메시지 스키마
- 웹 서비스가 사용하는 교환 방법 (요청/응답, 단 방향, 멀티캐스트 등)
- 웹 서비스 위치
- 오류 정보
- 머리글 정보

구현한 Java 웹 서비스 개발 방법을 알아보자. 웹 서비스 개발을 위해선 메소드의 인터페이스 파일과 구현 파일이 필요하다. 다음은 서비스해줄 메소드를 정의한 인터페이스이다.

```
public interface AseAnalysisIF extends Remote {
    public void AseAnalysis() throws RemoteException;
}
```

구현 파일에는 위 인터페이스를 구현한다. 서비스 해줄 메소드 작성이 끝난 후에는 config.xml 파일을 작성한다. config.xml 파일은 tie와 stub을 생성한다. 그리고 web.xml 파일을 작성한다. web.xml 파일은 웹 컴포넌트에 관한 배포 기술자로 웹 서비스 파일들을 하나의 컴포넌트 파일로 묶어서 서비스 해준다.

## 2.2 ATL 웹 서비스의 구현

ATL을 이용한 웹 서비스의 구현은 비주얼 스튜디오.NET을 사용하였다. 프로젝트 생성 후 프로젝트를 보면 ATL 웹 서비스는 헤더 파일과 C++의 소스파일인 .cpp파일로 이루어져 있다. 헤더파일을 보면 ATL 웹 서비스의 메소드 인터페이스를 정의해주는 부분이 다음과 같이 있다. \_interface Istore3Service 이 안에 다음과 같이 파일 분석 서비스의 인터페이스를 정의해 주었다.

```
[id(2)] HRESULT ReturnVertex([in] ATLSOAP_BLOB files,BSTR FileName, [out, retval] ATLSOAP_BLOB* result);
```

그리고 다음과 같이 메소드를 구현한다.

```
[soap_method ]
HRESULT ReturnVertex( ATLSOAP_BLOB files,BSTR FileName, ATLSOAP_BLOB* result)
```

컴파일을 하면 ReturnVertex메소드의 구현이 되었음을 나타내는 wsdl문서가 나타난다. ATL은 비주얼 스튜디오가 많은 부분을 자동으로 처리해 주지만 반드시 틀을 써야만 하는 단점이 있다. 이제 ATL과 Java 웹 서비스를 이용하여 클라이언트를 구현한다.

## 3. 웹 서비스를 이용한 애플리케이션 구현

본 절에서는 앞 절에서 구현한 Java 웹 서비스와 ATL 웹 서비스를 이용하여 ASE파일을 분석하여 웹에서 보여주는 뷰어를 만들었다. 클라이언트로는 위 언어들과 관계가 없는 C#언어를 사용하여 Java 웹 서비스와 ATL웹 서비스가 서비스 제공에 문제가 없는지를 본다. C#에서 웹 서비스를 이용해서 애플리케이션을 만들기 위해서는 프로젝트에서 웹 참조를 선택한 후 Java와 ATL웹 서비스의 WSDL문서의 주소만을 써주고 참조추가 버튼을 눌러주면 바로 사용할 수 가 있게 된다. 하지만 Java 웹 서비스를 참조하여 메소드를 사용할 경우 처음에 실행이 되지 않는 문제가 생기는데 이유는 비주얼 스튜디오의 버그로 웹 서비스가 일반적으로 80번 포트를 사용하는 반면에 Java 웹 서비스는 톰캣이 8080 포트를 사용하는 관계로 8080포트로 웹 서비스를 만들 경우 비주얼 스튜디오에서는 8080포트를 80포트로 인식한다. 때문에 실행이 되지 않을 경우 참조된 WSDL 문서를 열어 포트를 8080 포트로 조절하면 해결된다. 참조된 웹 서비스의 메소드는 로컬에 있는 메소드 같이 사용이 가능하여 다음과 같이 객체 생성 후 메소드를 호출하였다.

```
store3Service ase1 = new store3Service();
FileStream readstream = new FileStream(temp,
    FileMode.OpenOrCreate , FileAccess.ReadWrite);
byte[] bytePic;
BinaryReader fileread = new BinaryReader(readstream);
bytePic=
```

file.read.ReadBytes(Convert.ToInt32(readstream.Length));  
 ase1.ReturnVertex1(bytePic); //웹 서비스 메소드 호출  
 다음과 같이 웹 서비스의 객체를 생성하여 웹 서비스의  
 메소드를 호출한 다음 인수로 바이트 스트림으로 파일을  
 전송하였다. 웹 서비스와 통신을 할 때 자료형은 XML스  
 키마 형식으로 변환되어 전달된다. 아래 표에 각 언어  
 간 XML스키마와 호환되는 자료형을 조사하였다.

표 1 ([1], [4])

XML스키마	Java	Visual C++	C#	Visual Basic
boolean	boolean	bool	bool	boolean
byte	char	char	char	Char
byte	byte	_int8	sbyte	SByte
decimal			decimal	
enumeration			Enum	
unsignedByte		unsigned_in t8		
short	short	short	short	short
unsignedShort		unsigned short	ushort	UInt16
int	int	int	int	integer
unsignedInt		unsigned int	uint	UInt32
long	long	_int64	long	Long
unsignedLong		unsigned _int64	ulong	UInt64
double	double	double	double	Double
float	float	float	float	Single
BSTR		BSTR	string	문자열
base64Binary	byte[]	ATLSOAP_ BLOB	byte[]	byte[]

웹 서비스를 개발할 때 웹 서비스 내부에서 어떤 자료  
 형을 써도 상관없이 웹 서비스에서 제공하는 메소  
 드의 반환 값과 인수는 반드시 위의 표에 대응되는 자료  
 형을 사용하여야 한다. 예를 들어 Visual C++의 경우 인  
 수나 반환 값으로 이진파일을 사용할 경우 ATLSOAP\_BLOB  
 를 사용하지 않고 다른 형식으로 파일을 받거나 주려고  
 하면 컴파일도 되지 않는다.

지금까지 웹 서비스로 애플리케이션을 만들면서 경험한  
 장점과 단점을 나열하였다.

- 장점 : 1. 다중 플랫폼과 다양한 언어의 지원  
 2. 인터넷이 되는 곳이라면 언제든지 사용가능  
 3. 클라이언트에서 일일이 컴포넌트를  
 업 데이트 하지 않아도 된다.

- 단점 : 1. 실행시간에 서비스를 참조하기 때문에 웹  
 서비스 중단 시 클라이언트 실행불가  
 2. 네트워크 트래픽 발생시 예외 발생 또는  
 실행시간 지연

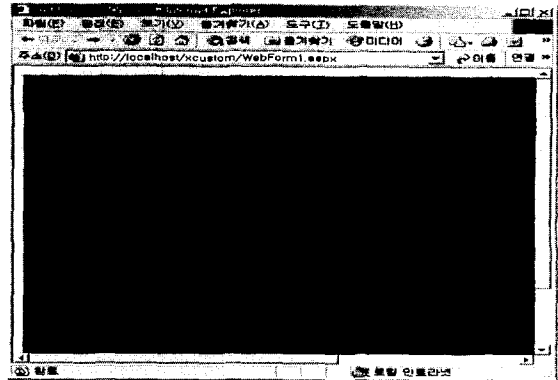


그림 4

호출한 메소드의 반환 값이 돌아오면 웹 서비스는 성공  
 적으로 실행된 것이다. 이제 웹3D 뷰어를 실행시키면 웹  
 서비스가 분석한 ASE파일을 읽어들이어 보여준다[그림 4].  
 웹 3D뷰어는 C#용 OPENGL라이브러리인 csgl을 사용하여  
 구현하였다. 뷰어는 웹 서비스가 분석한 파일의 정보를  
 읽어 그것을 익스플로러에 그림 4와 같이 보여준다.

#### 4. 결론

본 논문을 통해 웹 서비스 이용시의 장점/단점과 이용  
 할 때 생기는 문제점과 해결책을 보았다. 본 논문에서  
 사용한 웹 서비스는 웹의 장점/단점을 상당부분 그대로  
 가지고 있어 앞으로의 애플리케이션 개발에 참고할 수  
 있다. 개발한 서비스들을 서로 다른 언어들에서 사용하  
 여 언어중립 플랫폼중립 이라는 것을 보였다. 이것은 언  
 어와 개발플랫폼간의 호환성 문제를 해결하여 개발자들  
 이 특정 언어를 모르더라도 손쉽게 사용 수 있게 해준  
 다. 또한 재사용 가능한 컴포넌트이며 인터넷 환경에서  
 필요한 메소드를 호출할 수 있게 되었다는 것도 보였다.  
 또한 파일 뷰어의 경우 기존의 DirectX나 ATL을 사용한  
 웹3D와 비교해서 간단하게 웹에서 3D의 구현이 가능하기  
 때문에 다양한 분야에 적용할 수 가 있다.

#### 5. 참고 문헌

- [1] <http://java.sun.com>
- [2] David A. Chappell and Tyler Jewell, "Java  
 Web Service", O'REILLY, 2002.
- [3] Ashish Banerjee, Aravind Corera, Zach  
 Greenvoss와 "C# Web Services" WROX.
- [4] Microsoft MSDN library.
- [5] <http://www.webservices.org>.
- [6] Mason Woo and Jackie Neider, "OpenGL  
 Programming Guide", Addison Wesley 2002.