

레거시 컴포넌트의 자동화 된 웹 서비스 명세 생성 기법에 관한 연구

심성호*, 정화영**, 송영재*

*경희대학교 컴퓨터공학과

**경희대학교 컴퓨터공학과

e-mail:shim5800@cvs2.khu.ac.kr

A study on Automated Web Service Specification Generation Method of Legacy Component

Sung-Ho Sim*, Hwa-Young Jeong**, Young-Jae Song*

*Dept of Computer Engineering, Kyung-Hee University

**Dept of Computer Engineering, Kyung-Hee University

요 약

컴포넌트기반 서비스는 빠른 시간 안에 원하는 소프트웨어 제품을 적은 비용과 최소한의 시간에 생산할 수 있어 효율적인 시스템 개발 및 소프트웨어 재사용에 많은 이점을 가질 수 있다. 그러나 레거시 시스템들을 보면 대부분이 폐쇄적 네트워크를 사용하고 있으며 다른 시스템 정보를 이용 할 수가 없다. 그러나 레거시 시스템 자원을 재사용 할 수 있는 실용적인 방법론의 부재로 기존의 자원들을 재사용하지 못하고 유사 문제 영역의 시스템을 구축하기 위하여 시스템을 완전교체(Replacement)하고 있다.[1] 이런 문제점의 대안으로 떠오르고 있는 것이 웹 서비스이다.

본 논문에서는 웹 서비스의 개념과 웹 서비스에 적용하기 위한 레거시 컴포넌트의 자동화된 명세 생성을 통해 WSDL(Web Services Description Language)의 문제점을 보완하고 기존의 레거시 컴포넌트를 재사용 하므로 확장성과 자원의 재사용성을 극대화 할 수 있는 방법을 제시 하였다.

1. 서론

인터넷이 급속도로 성장한 현대사회는 정보화 사회로 전이되고 있다. 대량 생산의 산업 사회에서 정보와 지식의 가치 더 커진 정보화 사회가 이루어지도록 여러 방면에 변화가 일어나고 있다. 인터넷의 확산과 더불어 웹을 이용한 응용 소프트웨어의 사용이 증가하고 빠른 시간 안에 원하는 제품을 적은 비용과 최소한의 노력으로 생산해 내는 것에 많은 비중을 두고 있다. 기존의 시스템을 보면 대부분 폐쇄적 네트워크를 사용하고 있으며, 시스템 자체의 유연성이 없는 매우 고정적인 아키텍처를 채택하고 있어서 웹을 통해 각종 데이터를 조회한다든가, 다른 시스템의 데이터를 가져와 합성하거나 분석하는 등의 작업을 할 수가 없었다. 그러나 레거시 시스템 자원을 재사용 할 수 있는 실용적인 방법론의 부재로 기존의 자원들을 재사용 하지 못하고 유사 문제 영역의 시스템을 구축하기 위하여 시스템을 완전교체하고 있다. 가트너 그룹에 따르면, 전 세계 기업들의 중요 업무 시스템 가운데 네트워크에 연결된 시스템은 20%에 불과하며, 나머지는 서버-클라이언트 개념의 폐쇄된 시스템을 사용하는 것으로 나타났다. 최근

기업 내에서도 다수의 이질적인 시스템을 도입하면서 기업의 시스템이 산재해 있으면 이러한 분산 시스템을 통합하고자 많은 노력을 기울이고 있다. 이렇게 산재해 있는 개별적인 애플리케이션을 효율적으로 통합하는 대안으로 떠오르고 있는 것이 웹 서비스이다.

웹 서비스란 넓은 의미로 보면 웹을 통해 제공되는 서비스라고 이해할 수도 있겠지만, 이는 인터넷에 널려있는 수많은 콘텐츠 정보까지도 포괄하는 개념으로 엔지니어의 관점에서는 정확한 정의라고 할 수 없다. 최근에는 마이크로소프트나 IBM, HP등의 대형 벤더가 주장하는 웹 서비스 컴포넌트 기반 애플리케이션에 가깝다. 앞서 말한 기술적인 배경을 고려할 때, 웹 서비스는 표준 인터넷 프로토콜을 사용해 접근할 수 있는 애플리케이션이고 프로그래밍 가능한 캡슐화한 애플리케이션이고 일종의 컴포넌트 기반 애플리케이션의 특징을 가진다. 웹 서비스 표준 인터넷 프로토콜을 사용해 접근과 프로그래밍이 가능한 컴포넌트 기반의 캡슐화한 애플리케이션이라고 할 수 있다. 웹 서비스를 구성하는 기본요소 들로는 XML, SOAP, UDDI, WSDL이 있다. 이중에서 가장 활발하게 활동하며 업계 표준으로 자리를 잡고 있는 WSDL(Web Services

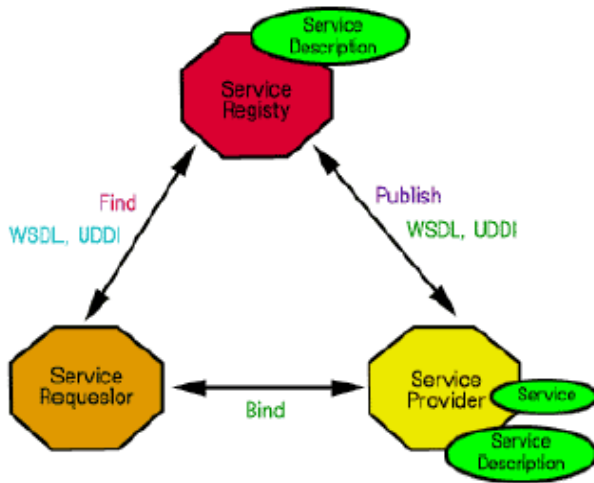
Description Language)을 이용하여 기존의 레거시 컴포넌트나 시스템 파일 데이터를 웹 서비스 할 수 있다. 그러나 데이터나 시스템 파일을 XML로 일일이 기술하는 문제점이 있어 신뢰성과 확장성이 낮다. 또 새로운 웹 애플리케이션이 추가되었을 때에는 다시 XML로 기술하여 적용을 해야 함으로 재사용성이 떨어져 진다. 본 논문에서는 웹 서비스에 적용하기 위해 레거시 컴포넌트의 자동화된 서비스 명세 생성을 통해 WSDL(Web Services Description Language)의 문제점을 해결하고 기존의 레거시 컴포넌트를 재사용 하므로 확장성과, 자원의 재사용성을 극대화 할 수 있는 방안을 제시 하였다.

2. 웹 서비스

본 장에서는 웹 서비스, 웹 서비스 구성 ,WSDL, WSDL생성과 사용, WSDL 구조, 레거시 컴포넌트에 대하여 기술 한다.

2.1 웹 서비스 구성

웹 서비스(web services)는 XML 표준을 기반으로 개발된 표준화된 XML 메시지를 통해 네트워크 상에서 접근 가능한 연산들의 집합을 기술하는 인터페이스를 지칭하는 용어로 기존의 웹 환경을 이용한 분산 컴퓨팅을 가능하게 한다.[2]



< 그림 1 > 웹 서비스 구성도

웹 서비스를 구성하는 기본 요소들은 다음과 같다.[2][3]

- SOAP : XML 프로토콜 표준으로 웹 서비스 요청 및 응답에 사용되는 메시지 형식을 정의
- UDDI : 웹 서비스에 대한 디렉토리 서비스를 지원하기 위해 개발된 레지스트리 표준으로 웹 서비스를 등록하고 검색/발견하기 위한 메커니즘을 제공
- WSDL : 웹 서비스 이용에 필요한 인터페이스와 입/출

력 메시지의 형식 등을 기술하기 위해 사용 된다.

2.2 WSDL(Web Services Description Language)

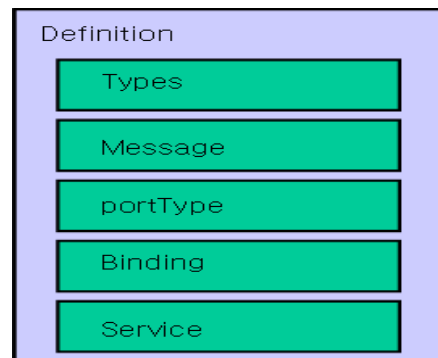
WSDL은 기업에서 UDDI에 등록된 웹 서비스를 어떻게 찾고 알아볼 수 있는지에 관한 표준 XML vocabulary 다.[4] WSDL은 MS SOAP과 IBM의 NASSL(Network Accessible Service Specification Language)에 기반을 두고 있다. 웹 서비스를 기술하는 스크립트인 WSDL은 XML 포맷으로 구성되고 HTTP를 통해 전달될 수 있으며 인터페이스를 정의 하는 IDL에 해당한다. WSDL을 이용해 웹 서비스 제공자는 사용자에게 해당 웹 서비스의 정확한 인터페이스와 사용되는 데이터 타입, 전송 프로토콜에 대한 상세 정보를 전달할 수 있다.

2.3 WSDL 생성과 사용

특정 서비스를 웹 서비스화 하기 위해 클라이언트에게 공개해야 하는 들이 있다. 기본적으로 클라이언트 구현을 위해 필요한 서비스 인터페이스를 공개해야 하고, 그 인터페이스 내부적으로 사용하는 클래스 혹은 데이터 타입 정보, 서비스에 접근하기 위한 종단점 등을 클라이언트에게 전달해야 하는데, 이러한 역할을 는 것은 WSDL이다. 여기서 눈여겨볼 만한 것은 WSDL에 서비스의 종단점이 포함된다는 점인데, CORBA 혹은 CORBA를 전송 계층에서 사용하는 J2EE에서는 IOR을 사용해서 하던 일을 웹 서비스에서는 WSDL에 포함시킨 것이다. 즉, WSDL은 CORBA의 IDL + IOR의 역할을 하고 있다.

2.4 WSDL 구조

WSDL 파일 구조는 <그림 2>처럼 파일 전체를 싸고 있는 definitions로 시작해서 내부에 types, message, portType, binding, service로 구성되어 있다. 즉 WSDL 파일은 여러 가지 정의로 구성되어 있고, 그 정의는 5가지이다.[4]



<그림 2> WSDL 파일의 구조

WSDL 파일의 각 영역별 의미[4]

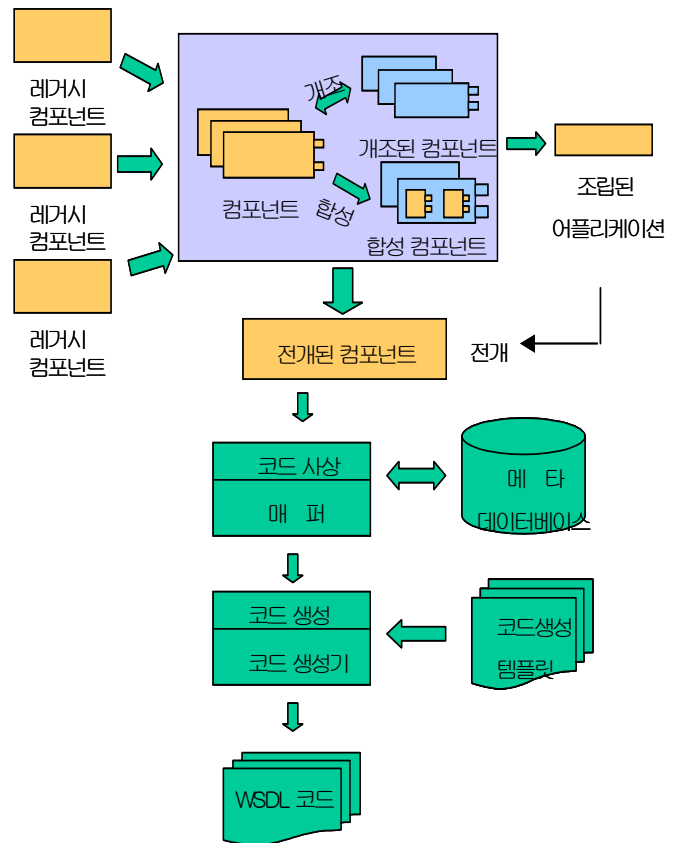
- types:클라이언트와 서버 간에 교환되는 메시지를 기술 하기 위한 데이터 타입 정의를 제공한다.
- message:전송된 데이터의 추상화된 정의를 나타낸다. 메시지는 어떤 타입 시스템 내부의 정의와 연관된 논리적인 part들로 구성된다.
- portType:추상화된 연산들의 집합. 각각의 연산들은 입력 메시지와 출력 메시지를 참조한다.
- binding:특정한 portType에 의해 정의된 연산과 메시지에 대한 구체적인 프로토콜과 데이터 포맷 스펙을 구체화한다.
- service:웹 서비스를 위한 개별 프로토콜마다 XML 웹 서비스를 사용하기 위한 종단점을 정의한다.

2.5 레거시 컴포넌트

일반적인 컴포넌트 기반의 소프트웨어 개발에는 컴포넌트를 자체적으로 개발하거나 제3자가 개발한 컴포넌트를 구입하여 사용하게 되는데 이러한 경우 많은 자원 낭비뿐만 아니라 재사용을 고려한 소프트웨어를 처음부터 개발해야 한다는 문제점이 발생한다. 이러한 문제점을 해결하기 위해서는 레거시 시스템을 컴포넌트화 시켜 자원을 재활용의 효율을 높인다.[5] 레거시 시스템을 컴포넌트화 하는 방법에는 기존 플랫폼에서 새로운 플랫폼으로 데이터를 이동하는 방법으로 주로 사용되는 이전(Legacy migration) 방법과 단일화된 기존 시스템을 래퍼와 인터페이스를 통하여 사용되는 래핑을 (Legacy wrapping) 방법, 하나의 기존 시스템의 응집도를 높이고 결합도를 낮추어 컴포넌트 화하기 위한 재구조화(Legacy restructuring) 방법이 있다.[1][5] 이렇게 생성된 레거시 컴포넌트를 웹 서비스에 적용하면 자원의 효율적 사용이 가능하다. 그러기 위해서는 WSDL 생성이 필요하다.

3. 레거시 컴포넌트를 위한 WSDL 생성

기존 레거시 컴포넌트를 이용하여 웹 서비스를 할 때 하나의 컴포넌트를 이용하면 웹 서비스의 규모가 작아지기 때문에 많은 서비스를 제공할 수 있게 컴포넌트 합성을 이용하여 제공한다. <그림 3>는 레거시 컴포넌트들을 합성해서 합성된 컴포넌트를 웹 서비스화 할 수 있게 WSDL 명세를 자동 생성하는 과정을 표현하고 있다.



<그림 3> 시스템 전체 구성도

3.1 WSDL 코드 생성

XML 코드 생성에 관여하는 모듈은 코드 사상, 코드 생성 모듈이며, 순서대로 수행되면서 코드 생성이 이루어진다. 이러한 분할은 통해 각 모듈은 서로로부터 분리되고, 독립적으로 변경 할 수 있다. 코드 사상 모듈에서는 매핑을 사용해 패턴 구조를 구성하는 메타모델들을 이용해 메타모델 데이터베이스를 구축하며 사용자의 코드 생성 요청이 있을 경우에 , 메타모델 데이터베이스의 메타모델을 코드 생성 템플릿에 매핑 시킨다 .마지막으로 코드 생성 모듈에서는 매핑이 끝난 코드생성 템플릿을 입력으로 하여 코드 생성기를 통해 XML 코드를 생성한다.

3.1.1 코드 매핑

코드 매핑에서 매핑은 컴포넌트 구조를 구성하는 메타모델들을 메타모델 데이터베이스를 통해 코드 생성 템플릿에 파라미터로 연결한다. 컴포넌트에서 각 메타모델들을 추출해 메타모델 정보를 갖는 메타모델 데이터베이스를 구축한다. 코드를 생성하기 위해 매핑이 메타 모델 데이터베이스에 저장된 메타모델을 코드생성 템플릿에 파라미터

의 형태로 전달한다.

3.1.2 코드 생성

매퍼가 코드생성 템플릿에 메타모델들을 매핑 시키면, 코드 생성기가 이를 번역하여 코드를 생성한다. 컴포넌트 구조를 XML로 변환하기 위해서 XMI를 사용하여 코드생성 템플릿을 구성하였다. 코드생성 템플릿 사용의 이점은 시스템 다른 부분의 간섭 없이 생성된 코드를 수정할 수 있다. 예를 들면 새로운 코드생성 템플릿을 사용함으로써, XML 외에 C++이나 Java 코드를 생성할 수도 있다. XMI 코드생성 템플릿에서 Header, Content, Extensions 요소를 XML 코드로 생성하는 의사코드이다. Header요소에서는 전송되는 모델 데이터가 따르는 메타모델과 XMI의 버전을 기술하며 Content요소는 모든 클래스와 인터페이스 메타모델들에 대한 기본 정보와 이에 포함된 속성과 파라미터를 비롯함 함수들의 정보를 기술한다. 또한, 속성과 함수에 대한 초기 값의 데이터 형과 클래스와 인터페이스에 연관된 관계들의 정보를 기술한다. 끝으로, Extensions요소에서는 Contents요소에서 기술된 메타모델들에 대해서 좌표, 스타일 같은 가시적인 표현정보를 기술한다. 코드생성 템플릿에 매핑된 메타모델 정보를 코드 생성기가 번역하여 XML코드로 생성한 화면이며, 생성된 XML 코드는 플랫폼에 독립적이며, XMI를 지원하는 CASE 도구에서 호환성을 가지므로, 시스템 설계시 재사용 컴포넌트로 사용 될 수 있다.

4. 결론 및 향후 연구과제

본 논문에서는 웹 서비스의 개념과 레거시 컴포넌트의 자동화된 WSDL 코드를 이용하여 웹 서비스를 제안했다. 본 논문에서 제시한 레거시 컴포넌트의 자동화된 WSDL은 레거시 시스템을 완전교체하거나 패쇄적인 공간에서만 사용 되어지던 컴포넌트를 이용함으로써 자원의 재사용성과 확장성이 향상된 웹 서비스에 목적을 두고 설계하였다. 향후 연구 과제로는 레거시 컴포넌트 WSDL 코드 생성기에 대한 세부적인 설계와 구현이다.

참고문헌

- [1] Lawrence Wilkes, "Creating Components from Legacy Applications", CBDi Forum Journal, December 1998.
- [2] Header Kreger, IBM Software Group, "Web Services Conceptual Architecture (WSCA 1.0)", <http://www-4.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>, May 2001.

- [3] Martin Gudgin, et al., "SOAP Version 1.2 Part 1: Messaging Framework", <http://www.w3.org/TR/2002/WD-soap12-part1-20020626/>, Working Draft, W3C, June 2002.
- [4] "Web Services Description Language (WSDL) 1.1" W3C, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [5] J. Disbal and D. Lawiess and Bing Wu. Legacy Information Systems : Issues and Directions, IEEE Software, pp 103-111, 1999