

## 핫 스왑핑을 지원하는 웹 서비스 프레임워크 설계

최종영<sup>○</sup> 오수열

목포대학교 컴퓨터공학

{jmchoi<sup>○</sup>, syoh}@mokpo.ac.kr

### Design of Webservice Framework Supporting Hot Swapping

Jong-Myung Choi<sup>○</sup>, Soo-Yul Oh

Computer Eng., Mokpo National Univ.

#### 요 약

웹 서비스는 차세대 기술임에도 불구하고, 비즈니스 측면의 중요한 요소인 보안, 핫 스왑핑, 과금 등을 지원하지 않는 문제점을 가지고 있다. 이러한 문제를 해결하기 위해서 본 논문에서는 웹 서비스를 보다 쉽게 배포하고, 사용할 수 있으면서 비즈니스에서 중요한 특성을 지원할 수 있는 프레임워크를 소개한다. 이 프레임워크는 웹 서비스 컨테이너위에서 메타 데이터를 이용해서 자동으로 보안, 핫 스왑핑, 과금 등의 기능을 제공한다. 따라서 이 프레임워크를 사용하는 경우에 보다 쉽게 웹 서비스를 개발 및 사용할 수 있기 때문에 시스템 개발에서 많은 비용과 노력을 줄일 수 있을 것이다.

#### 1. 서 론

웹 서비스[1]는 단순성, 상호 운영성, 표준화된 기술 사용, 빠른 발전 등의 장점[2]을 갖고 있는 기술이다. 현재 웹 서비스를 위한 SOAP[7], WSDL[8], UDDI[9] 등의 기반 기술들이 이미 표준으로 제정되었고, 이러한 기술들 이외에 웹 서비스의 보안[14], 트랜잭션 처리 [13], 비즈니스 프로세스[11,12] 등은 많은 단체를 통해서 활발하게 연구 혹은 표준화되고 있다. 웹 서비스 기술들이 모두 표준화되고, 널리 사용되는 경우에 비즈니스 응용프로그램 개발에서 상당히 많은 비용과 노력을 줄일 수 있을 것이다.

현재 웹 서비스를 제공하기 위한 웹 서비스 컨테이너 들은 기본적으로 2가지 문제점을 가지고 있다. 첫째는 웹 서비스 컨테이너가 WSDL을 이용한 단순한 원격 객체의 메소드를 호출하는 정도의 기능을 제공하고 있고, 비즈니스를 위해 보다 중요한 보안, 신뢰성, 성능 등의 기능은 제공되지 않거나 혹은 제한적으로 제공한다는 것이다. 둘째는 웹 서비스 컨테이너를 사용하기 어렵다는 점이다. 앞으로 웹 서비스가 활발하게 사용되면, 응용프로그램 개발의 주체는 개발자 중심에서 사용자 중심 형태로 변경되게 될 것이다[6]. 따라서 웹 서비스 컨테이너는 보다 쉽게 사용자가 원하는 비즈니스 특성에 따라 속성을 부여하면서 웹 서비스를 사용할 수 있는 형태로 변경되어야 한다.

본 논문에서는 웹 서비스 컨테이너에서 웹 서비스를 쉽고, 효과적으로 관리할 수 있는 프레임워크에 대해서 소개한다. 이 프레임워크는 웹 서비스의 라이프 사이클 관리, 보안, 접근 제한, 핫 스왑핑 기능[5], 프로세스 오케스트레이션 등의 기능을 제공한다. 또한 웹 서비스가 자원을 어느 정도 사용하고 있지를 알아볼 수 있는 모니터링 기능, 로깅 기능, 과금 기능 등을 제공할 것이다.

논문은 2장에서 관련 연구를 소개하고, 3장에서 프레임워크에 대한 내용을 소개한다. 4장에서는 결론 및 향후 연구 과제를 밝힌다.

#### 2. 관련 연구

웹 서비스에서는 비즈니스 프로세스를 동적으로 결합하기 위한 방법이 주로 연구되었고, 웹 서비스를 위한 프레임워크에 대한 연구는 상대적으로 적게 이루어졌다. 웹 서비스를 위한 프레임워크에 관한 대표적인 연구로는 DySCo[3]와 WSIF[4]가 있다. DySCo는 동적으로 웹 서비스를 결합하기 위한 프레임워크이다. WSIF는 웹 서비스 컨테이너에 무관하게 웹 서비스를 쉽게 호출할 수 있도록 지원하기 위한 프레임워크이다. WSIF를 사용하는 경우에 웹 서비스 컨테이너에 무관하게 웹 서비스를 사용하는 클라이언트 프로그램을 작성할 수 있다는 장점을 가지고 있다.

본 논문에서 소개하는 프레임워크는 웹 서비스 클라이언트를 쉽게 작성하는 점에서 WSIF와 유사하지만, WSIF에서는 지원하지 않는 핫 스왑핑, 보안, 접근 제어 등의 기능을 지원하기 때문에 많은 면에서 다르다.

#### 3. 본 론

##### 3.1 프레임워크 구성

웹 서비스는 많은 장점을 가지고 있지만 실제 비즈니스에서 널리 사용되기 위해서는 해결되어야 할 다음과 같은 문제점들을 가지고 있다.

- 서로 다른 웹 서비스 시그내처 - 웹 서비스의 시그내처가 표준화되어 있지 않기 때문에 다른 웹 서비스로 변경하기가 어렵다.

- 접근 제한 - 웹 서비스는 모든 사용자에게 오픈할 수 없기 때문에 웹 서비스에 적절한 접근 제한을 두어야 한다.
- 보안 - 웹 서비스는 비즈니스에서 매우 중요한 부분이기 때문에 보안에 신경을 써야 한다.
- 과금 - 웹 서비스 제공자는 무료로 서비스를 제공할 수 없기 때문에 적절한 보상을 받을 수 있는 과금 정책이 필요하다.
- 서비스 품질 보증 - 웹 서비스는 서비스 품질을 기술하고, 보증할 수 있는 방법이 있어야 한다.

웹 서비스 제공자가 위와 같은 문제점을 해결하기 위해서 웹 서비스에 여러 가지 기능을 추가하는 것은 많은 비용과 노력을 필요로 하기 때문에 구현하기 어렵다. 따라서 웹 서비스와 웹 서비스 관리 기능을 분리하기 위해서 본 논문에서는 관리 기능을 프레임워크를 통해서 제공하는 방법을 사용한다. 논문에서 소개하는 웹 서비스 프레임워크는 그림 1과 같은 구조를 갖는다.

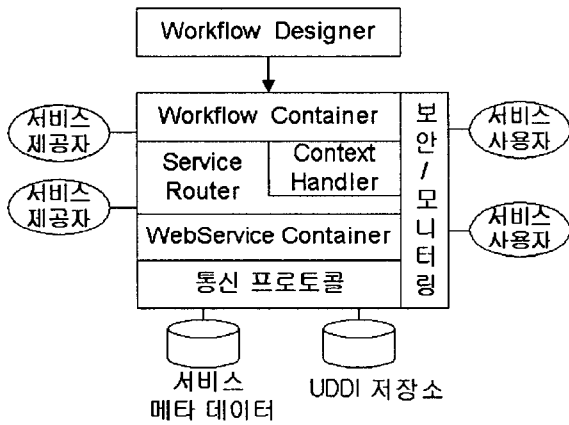


그림 1. 프레임워크 구성

프레임워크의 가장 하위에는 HTTP 혹은 MOM(Message Oriented Middleware) 형태의 통신 프로토콜이 존재한다. 통신 프로토콜은 네트워크를 통해서 SOAP 메시지를 전달하는 역할을 수행한다.

웹 서비스 컨테이너는 웹 서비스의 생성, 관리 및 제거하는 기능을 수행한다. 클라이언트의 요청이 있는 경우에 웹 서비스 엔진은 필요한 웹 서비스 인스턴스를 생성하고, 서비스를 수행하도록 한다. 이때 웹 서비스 호출을 SOAP 메시지로 변경하고, 다시 SOAP 메시지를 리턴값으로 변환한다.

서비스 라우터는 웹 서비스 사용자의 정책이나 기술적인 문제 때문에 동적으로 새로운 웹 서비스를 사용할 수 있도록 지원한다.

컨텍스트 핸들러는 웹 서비스 제공자가 사용자의 정보에 따라서 서비스 형태 혹은 품질을 변경할 수 있는 기능을 제공한다.

워크플로우 컨테이너는 정의된 비즈니스 프로세스에

따라서 작업을 수행할 수 있는 기능을 제공한다. 현재 워크플로우는 BPEL, BPML 등이 표준으로 제안되고 있지만, BPEL이 가장 유력한 표준으로 등장할 것으로 보인다. 따라서 워크플로우 컨테이너는 BPEL을 지원할 것이다.

보안 계층에서는 사용자 인증, 접근 권한 제어, 암호화 등의 작업을 수행한다. 웹 서비스는 보안을 위해서는 SSL과 같은 보안성이 있는 통신 프로토콜을 사용하고, SOAP 메시지를 암호화함으로써 보안을 강화할 수 있다. 또한 사용자 인증과 접근을 제어함으로써 인증되고 권한이 있는 사용자만 서비스를 사용할 수 있도록 한다.

모니터링 레이어는 현재 프레임워크에서 서비스되고 있는 웹 서비스에 대한 전반적인 기능들을 모니터링한다. 예를 들어, 현재 서비스되고 있는 웹 서비스가 어떤 것이고, 누구에게 서비스 되고 있는지, 현재 웹 서비스의 성능은 어떨고, 어느 정도의 자원을 사용하는지 여부를 보여준다.

로그 계층은 현재 프레임워크에서 발생하는 이벤트들에 대한 내용을 기록한다. 예를 들어, 어떤 서비스에 대한 요청이 있으며, 어떤 에러들이 발생했는지 등의 정보를 기록한다.

워크플로우 디자이너는 그래픽 형태로 워크플로우를 기술할 수 있는 도구이다. 작성된 워크플로우는 BPEL 문서로 워크플로우 컨테이너에 전달된다.

### 3.2 핫 스왑핑

비즈니스에서 새로운 파트너를 찾는 것은 매우 흔한 일이다. 정책, 기술, 가격 등 여러 가지 이유 때문에 기존의 웹 서비스 제공자로부터 새로운 웹 서비스 제공자로 변경해야 하는 일이 흔하게 발생할 수 있다. 이러한 경우 이외에도 시스템 장애나 네트워크 장애 때문에 웹 서비스 제공자를 변경해야 하는 경우도 있다. 웹 서비스는 B2B에서 매우 중요한 역할을 수행하기 때문에 제공되지 않는 서비스를 위해 무한정 기다릴 수 없기 때문에 문제가 발생하는 경우에는 새로운 웹 서비스 제공자로 빠르게 전환할 필요가 있다.

프레임워크에서 서비스 라우터는 동적으로 웹 서비스를 변경할 수 있는 핫 스왑핑을 제공한다. 핫 스왑핑은 서비스 중단, 과부하 등과 같은 서비스 제공자에 문제가 있는 경우에 실행시 동적으로 새로운 서비스 제공자로부터 서비스를 받을 수 있도록 하는 기능이다.

그림 2는 웹 서비스 S1을 사용하다가 동적으로 웹 서비스 S2 변경하는 것을 보여준다.

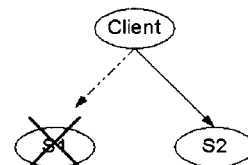


그림 2. 핫 스왑핑

S1과 S2 웹 서비스는 시그니처가 다를 수 있기 때문에 웹 서비스에 관한 메타 정보들을 관리해야 한다. 웹 서비스의 메타 정보들은 다음과 같은 XML 형태로 관리된다. 표 1은 메타 정보의 예를 보여준다. getQuote라는 웹 서비스는 여러 개의 provider로부터 서비스 받을 수 있다. 각 provider는 웹 서비스를 받기 위한 주소와 메소드 이름, 매개 변수의 개수와 타입 정보를 가지고 있다. 이때 각 웹 서비스마다 매개 변수 타입이 다른 경우에는 mapper를 통해서 자동적으로 타입 변환이 이루어진다.

```
<service>
  <name>getQuote</name>
  <provider>
    <url>203.23.3.1:8080</url>
    <method>
      <name>get_quote</name>
      <return>long</return>
      <arglist>
        <arg>int</arg>
      </arglist>
      <mapper>GeneralMapper</mapper>
    </method>
  </provider>
</service>
```

표 1. 서비스 메타 정보

4. 결론 및 향후 연구 과제

웹 서비스는 상호 운영성이 뛰어나기 때문에 업체로부터 많은 관심의 대상이 되고 있다. 본 논문에서는 웹 서비스를 쉽게 관리하고, 보안, 핫 스왑핑 등의 기능을 제공할 수 있는 웹 서비스 프레임워크에 대한 내용을 소개하였다.

웹 서비스 프레임워크는 비즈니스 프로세스를 효과적으로 결합할 수 있는 BPEL 컨테이너, 웹 서비스 모니터링을 수행할 수 있는 기능, 로깅 기능, 핫 스왑핑을 제공할 수 있는 기능을 제공한다. 또한 웹 서비스를 위한 과금 정책도 제공한다. 웹 서비스 프레임워크는 웹 서비스를 개발하는 제공자와 웹 서비스 사용자들이 효과적으로 웹 서비스를 사용할 수 있도록 도와줄 것이다.

현재 웹 서비스 프레임워크는 보안, 핫 스왑핑 등의 기능을 구현하였으며, 워크플로우 컨테이너를 결합하는 작업을 수행하고 있다. 향후에는 설계된 프레임워크의 나머지 기능들을 구현할 것이다.

참고 문헌

1. W3C, Web Services Architecture, <http://www.w3c.org/>.
2. 이경하 and 이규철, "웹 서비스의 향후 발전 방향",

- 정보처리학회 학회지, Vol. 9, No. 4, pp. 24-30, Jul., 2002.
3. Rama Akkiraju, et al., "A Framework for Facilitating Dynamic e-Business Via Web Services", in Proc. of OOPSLA, 2001, available at <http://researchweb.watson.ibm.com/CoopDS/publications.html>.
4. Matthew J. Deftler, et al., "Web Services Invocation Framework (WSIF)", in Proc. of OOPSLA, 2001.
5. Feng , "Dynamic Evolution of Network Management Software by Software Hot-Swapping", in Proc. of International Symposium on Integrated Network Management, pp. 63-76, 2001.
6. 한동수 and 고인영, "웹 서비스 환경에서의 사용자중심 동적 e-비즈니스 응용 프로그램 통합(eAI) 프레임워크", 정보과학회지, pp. 30-40, Vol. 22, No. 7, Jul., 2004.
7. Simple Object Access Protocol (SOAP) 1.2, <http://www.w3.org/TR/soap/>.
8. Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/wsdl/>.
9. Universal, Description, Discovery and Integration (UDDI), <http://www.uddi.org/>.
10. Business Process Execution Language for Web Services Version 1.1, <http://www-106.ibm.com/developerworks/library/ws-bpel/>.
11. Business Process Modeling Language, <http://www.bpmi.org/>.
12. Business Transactions Protocols, <http://www.oasis-open.org/>
13. Web Services Transaction (WS-Transaction), <http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/>.
14. Web Services Security (WS-Security), <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>.