제20회 한국정보처리학회 추계학술발표대회 논문집 제10권 제2호 (2003. 11)

EJB 서버 시스템의 웹서비스 지원 모듈에 관한 연구

김성훈*, 김중배*
*한국전자통신연구원 모바일응용서버연구팀
e-mail:saint@etri.re.kr

A Study on Web Service Module for EJB Server System

Sung-Hoon Kim*, Joong-Bae Kim*
*Mobile Application Server Research Team, ETRI

요 약

본 논문에서는 EJB(Enterprise JavaBean) 기반의 웹응용서버 시스템에서 EJB2.1 표준 규격의 웹서비스를 지원하기 위한 모듈의 설계 사항에 관한 내용이다. 본 논문에서는 EJB2.1 표준 규격의 웹서비스에 대한 요구사항과 JAX-RPC기반의 웹서비스 시스템을 구현하고자 할 때 고려해야할 시스템 측면의 요구사항을 살펴보고, 기존 웹응용서버들에서 웹서비스를 지원하는 방식들에 대해 소개한다. 또한 시스템 요구사항들을 반영한 구현 시스템의 구체적인 설계사항들에 대해 설명한다.

1. 서론

SUN사의 EJB 표준은 분산 객체 컴포넌트 기반의 개발 방법론에 관한 표준을 기술하고 있는 산업계 표준이다. 컴포넌트 기반의 개발 방법론이 자리를 잡아감에 따라 EJB 기반의 응용 개발이 더욱 확산되어 가고 있으며, EJB 기반의 웹응용서버 시장도 날로 커져가고 있다. 이제는 웹 기반의 응용은 웹응용서버를 떠나서는 생각할 수 없는 실정이 되었다[1].

웹서비스(Web Service)는 e-비즈니스 표준을 준수하고 인터넷을 통해 제공되는 비즈니스 로직을 갖는 소프트웨어 컴포넌트로 정의된다[1]. 웹서비스는 웹응용을 제공하는 벤더나 SI 업체에게는 하나의 응용으로 인식되지만 이를 사용하는 사용자에게는 모두 포장되어 있어서 기술이나 컴퓨팅이 아닌 서비스로 인식되기 때문에 서비스라는 용어로 사용되고 있다. 또한 웹서비스는 ASP의 기술 진화의 연장선상에 있다고 볼 수 있지만 기술이나 컴퓨팅이 아닌 서비스로서 이용된다는 점에서 기존의 응용이나 소프트웨어 컴포넌트 개념과 다른 새로운 패러다임이라고 할 수 있다[2].

상기의 EJB 표준 규격은 현재 2.1버전의 표준화가 진행되고 있으며, 올해 안에 최종 버전이 나올 예정이다. EJB 2.1 표준 규격에서는 기존 2.0 규격에 웹서비스의 기능과 타이머 서비스가 추가되고, 메시지 기반 빈과 EJBQL의 기능이 개선될 예정이다[1].

본 논문에서는 기존에 기 개발된 EJB2.0 기반의 웹응용서버 시스템에서 웹서비스를 지원하기 위해 필요한 요구사항들과 이를 제공하기 위한 시스템의 설계 사항들에 대해 설명하고자 한다.

본 논문의 제2절에서는 EJB2.1 표준 규격에서 요구하는 웹서비스의 요구사항들과 JAX-RPC기반의 웹서비스를 시스템을 개발하고자 할 때 필요한 시스템상의 요구사항들을 설명하고, 3절에서는 기존 시스템들의 구현 방식을 소개하고, 4절에서는 요구사항들의 구현방안을 설명하며, 5절과 6절에서는 구현된 시스템의 구체적인 설계사항들을 설명하고 7절에서 결론을 맺는다.

2. 요구사항

EJB2.1 표준의 규격의 웹서비스에 대한 요구사항은 크게 다음과 같이 세 가지로 요약할 수 있다.

첫 번째는 EJB빈에서 제공하는 모든 웹서비스는 SUN사의 XML기반 RPC 표준 규격인 JAX-RPC를 기반으로 해야 한다. 두 번째는 EJB빈이 웹서비스의 클라이언트로써 동작할 수 있어야 한다. 이를 위해서 EJB빈의 DTD상의 환경 정보에 <service-ref> 태그를 통해 EJB빈이 사용하는 웹서비스 레퍼런스 객체(JAX-RPC Service객체)를 등록하고 사용할 수 있어야한다. 세 번째는 무상태 세션빈(Stateless Session빈) 객체가 웹서비스의 엔드포인트(endpoint)로 동작할 수 있어야 한다. 이를 위해서 JAX-RPC 표준의 스텁(Stub)과 타이(Tie)를 생성할 수 있어야 한다. Stub과 Tie 객체는 웹서비스를 RMI 매커니즘과 동일한 방식으로 제공하기 위한 JAX-RPC의 표준 규격에 준하도록 작성해야 한다.

따라서 상기의 세가지 요구사항을 종합하면 EJB 2.1 표준 규격의 웹서비스는 JAX-RPC기반으로 서버와 클라이언트로 동작할 수 있어야 한다는 것이다. EJB2.1 표준 규격에서 웹서비스의 엔드포인트로 무상태 세션 빈만을 지원하도록 한 것은 웹서비스 자체가 상태를 가지는 호출이 아니기 때문이다[2].

다음은 상기의 요구사항에 대해 무상태 세션 빈이 웹서비스 엔드포인트(end-point)의 역할을 수행할 수 있도록 EJB 서버 시스템이 가져야 할 기능에 대한 요구사항들이다.

첫 번째는 웹을 통해 전달된 클라이언트의 웹서비스 요청에 대해 URI를 분석하여 엔드포인트 역할을 수행하는 EJB빈이 호출되었는지를 분석하는 메커니즘을 제공해야 한다.

두 번째는 클라이언트가 요청한 SOAP 문서를 해석하여 데이터를 분석하고 결과로 반환된 값을 재포장하는 기능을 수행할 수 있어야 한다.

세 번째는 웹서비스로 EJB빈을 호출하였을 때 EJB 서버의 내부의 응용 로직을 수행 할 수 있도록 해야 한다. 즉, 단순히 EJB빈의 레퍼런스만을 호출하는 것이 아니라 사용자 인증, 트랜잭션 처리 등의 서버 내부의 로직을 수행하도록 해야 한다.

네 번째는 개발자가 작성한 웹서비스 인터페이스와 빈에 대해 스텁(Stub), 타이(Tie), WSDL, Service객체 등의 Artifacts들을 JAX-RPC 표준을 준수하여 생성할 수 있는 툴을 제공해야 한다는 것이다.[3].

다섯 번째는 빈이 웹서비스 클라이어언트로 동작할 때 사용하는 Service객체를 JDBC와 환경변수처럼 로컬 네이밍에 바인딩하여 서비스할 수 있어야 한다.

3. 기존 시스템의 웹서비스 지원 방식

본 절에서는 시스템의 구현 관점에서 기존에 제품화되어 상용화된 웹응용서버들의 웹서비스 지원 방식에 대해 설명한다. 기존 웹응용서버들의 구현방식은 다음과 같이 크게 세 가지 형태로 분류할 수 있다.

첫 번째는 아파치에서 제공하는 SOAP엔진인 AXIS의 EJBProvider를 이용하여 구현하는 방식이다. 이러한 구현 방식은 상기 요구사항들 중에서 JAX-RPC 기반의 웹서비스를 제공해야 한다는 점을 만족하지 못한다는 단점이 있다.

두 번째는 AXIS의 JAX-RPC Provider를 이용하는 방법이다. 이러한 방법은 EJB2.1 표준에서 언급하고 있는 JAX-RPC기반 웹서비스와는 달리 타이(Tie) 객체의 역할을 RPCProvider 객체가 수행함으로써 타이(Tie)가 필요 없는 방식이다.

마지막으로 SUN사의 웹서비스 툴킷인 JWSDP를 그대로 이용하는 방식이다. JWSDP는 썬의 XML관련 모든 표준을 제공하고 있지만 EJB2.1의 웹서비스 방식을 직접적으로 제공하고 있지는 않다. 따라서 개발자는 JWSDP를 수정하여 EJB2.1에 맞게 재개발하여야 한다.

4. 런타임 모듈 구현방안

본 절에서는 2절에서 설명한 요구사항들에 대한 EJB서버 시스템의 구현 방안에 대해 설명한다.

요구사항 첫 번째 항목인 URI 분석에 대해서는 웹컨테이너의 필터 또는 인터셉터 로직을 통해 클라이언트의 모든 요청 URI를 분석하여 EJB 서버에 배포된 빈 웹서비스의 URI 패턴과 비교하는 방법과 특정 서블릿을 작성한 후 서블릿의 Mapping 매커니즘을 이용하여 클라이언트의 모든 요청이 해당 서블릿으로 전달하게 한 후 URI 패턴을 분석하는 방식이 있다. 상기의 방식들 중에서 첫 번째 방식은 EJB 웹서비슬 배포할 때 웹컨테이너에 웹 응용을 배포할 필요가 없으며 웹 컴포넌트 웹서비스와 무관하게 EJB 웹서비스를 수행할 수 있는 장점과 웹컨테이너와 EJB서버가 하나의 프로세스에서 디자인된 경우에는 성능이 좋은 장점이 있다. 반면 두 번째 방식은 AXIS나 JWSDP를 이용하는 방식에서 사용되는 것으로 서버측 로직을 크게 손댈 필요가 없는 장점이 있는 반면 성능과 이중 배포의 단점이 있다. 따

라서 본 논문에서는 웹컨테이너의 인터셉터 로직을 사용하도록 한다.

두 번째 요구사항인 SOAP 메시지 분석은 SOAP 메시지를 직접 파싱하고 처리하는 모듈을 구현하여 제공한다.

세 번째 요구사항인 서버측 로직을 수행하는 것은 JNDI를 이용하여 빈을 룩업한 후 메소드를 호출하는 경우와 타이(Tie)의 타켓으로 웹서비스 인터페이스의 구현 객체를 호출하도록 구성하고 구현 객체에서 서버측 로직을 수행하도록 하는 방식이 있다. 상기 방식 중에서 JNDI를 호출하는 방식은 성능이 떨어지는 단점이 있으므로 구현객체를 생성하여 이를 통해 서버측 로직을 수행하도록 설계한다.

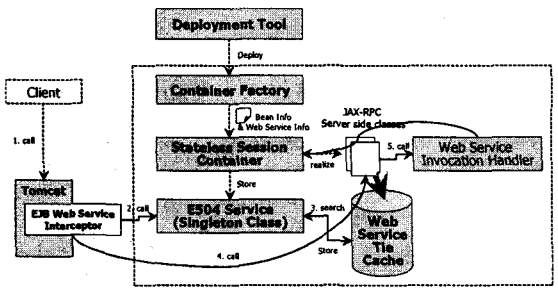
네 번째 요구사항인 Artifacts 생성은 SUN사의 JAX-RPC 컴파일러를 사용하여 제공하도록 한다.

다섯 번째 요구사항은 기존에 EJB서버가 제공하던 방식과 큰 차이 없이 DTD분석 단계에서 기존 기능에 추가하여 제공하도록 한다.

5. 런타임 모듈 구현 모델

1) 전체시스템

<그림1>은 상기 시스템 요구사항에 대한 구현 방안을 기반으로 작성된 서버 측 시스템의 구조도이다.



<그림 1> 시스템 구조도

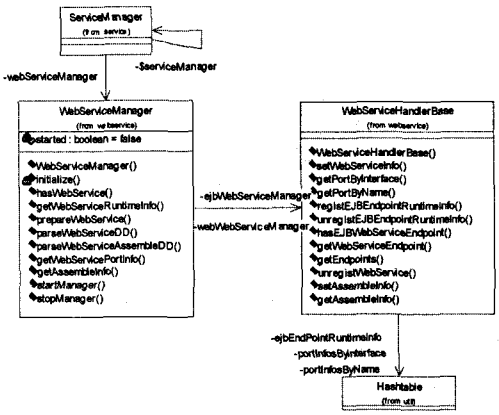
상기 시스템 구조도에서 웹컨테이너는 아파치에서 제공하는 톰캣(Tomcat)을 사용했으며, EJB 서버 시스템은 기 개발된 ETRI EJB 서버 시스템을 사용했다.

상기에서 EJB Web Service Interceptor는 요구사항 중 첫 번째의 기능을 수행하며 E504Service 모듈로부터 클라이언트 요청 URI에 대해 EJB 엔드포인트(endpoint)가 있는 지를 검색하고, 해당 EJB 엔드포인트가 검색될 경우에 해당되는 타이(Tie) 객체에 요청을 전달한다. 타이(Tie) 객체는 요구사항 두 번째의 기능인 SOAP 메시지를 해석하고 요구사항 세 번째의 기능인 서버 측 로직을 수행하기 위하여 Web Service Invocation Handler에 요청을 전달한다. Invocation Handler는 클라이언트 사용자의 인증과 트랜잭션 관련 처리 등의 서버측 로직을 수행한 후 최종적으로 무상태 세션 빈의 레퍼런스를 관리하는 컨테이너에 요청을 전달한다. 컨테이너는 빈 레퍼런스를 풀에서 불러온 후 메소드를 실행하고 결과를 타이(Tie)에 되돌린다. 타이는 반환된 결과를 SOAP 메시지로 변환한 후 클라이언트에 전달한다.

상기 <그림1>에서 Container Factory모듈은 빈의 정보를 기반으로 컨테이너를 생성하는 모듈이며, WebService Tie Cache는 생성 타이(Tie) 레퍼런스들을 관리하는 캐쉬이다.

2) 서버 측 배치 모듈

웹서비스를 위한 EJB2.1의 표준의 배포명세서는 아직 완성된 상태는 아니다. 따라서 본 시스템에서는 SUN에서 제공하는 JAX-RPC 기반의 웹서비스의 DTD를 사용하여 EJB빈의 배포명세서를 작성하였다. 본 시스템에 빈을 배포하기 위해서는 EJB빈의 명세서와 시스템에 URI등의 특화된 정보를 담고 있는 명세서가 필요하다. <그림 2>는 웹서비스 배치와 관련된 관리자의 클래스 다이어그램이다.



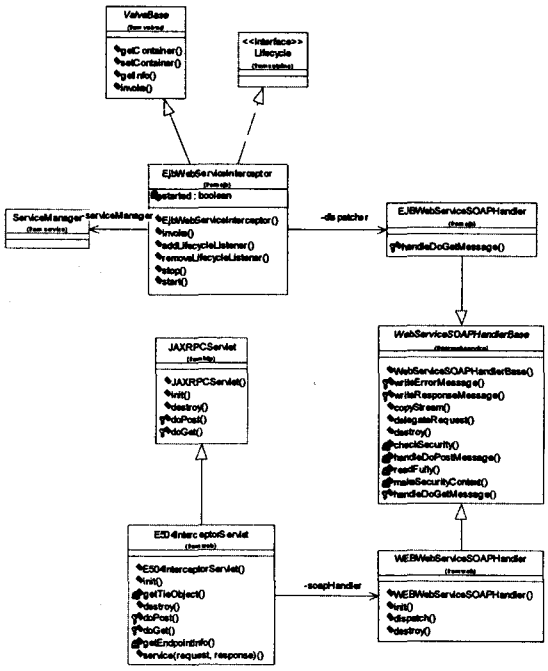
<그림2> 웹서비스 관리자 모듈

웹서비스 관리 모듈은 웹서비스 배치 시 DD의 XML 정보를 해석하고 해석된 정보를 관리하는 기

능을 수행하며, 런타임 서비스 때 클라이언트 요청에 대해 해당되는 웹서비스 엔드포인트를 런타임 모듈에 제공하는 역할을 수행한다. 웹서비스의 배치를 위한 XML 정보와 해석된 정보를 기반으로 생성된 런타임 정보는 모두 WebServiceHandlerBase에 의해 관리된다.

3) 인터셉터 모듈

<그림3>은 인터셉터 모듈의 클래스 다이어그램이다. <그림3>에서 EJB 웹서비스 인터셉터는 톰캣의 Valve 구조를 이용하여 구현하였으며, 상기에서 설명한 바와 같이 클라이언트 요청 URI를 분석하여 해당 EJB 웹서비스를 찾는 역할을 수행한다.



<그림3> 인터셉터 모듈 클래스 다이어그램

SOAP핸들러 클래스 SOAP 메시지를 파싱하고 처리하는 역할을 수행하며, 최초에 해석된 SOAP 정보를 기반으로 SOAPMessage 객체를 생성한 후 컨텍스트 객체에 담아서 인터셉터에 찾은 EJB웹서비스 엔드포인트를 직접 호출하는 역할을 수행한다. 엔드포인트에서 반환된 값은 컨텍스트에 저장되어 있으며 이 값은 SOAPHandler에 의해 클라이언트에 되돌려지게 된다.

6. 핸들러 (Handler) 및 보안 처리

JAX-RPC 표준의 핸들러는 클라이언트와 서버 간에 주고받는 SOAP 메시지에 대해 메시지가 전송되기 전과 전달 받은 직 후 암호와 및 복호화, 로깅, 인증, 캐싱, 응용에 특화된 SOAP 헤더 처리 등의 사용자가 임의로 정의한 작업을 처리를 위하여 존재한다. 개발자는 핸들러를 작성하여 메시지가 전달되기 전후에 수행하는 또 하나의 비즈니스 로직으로 사용할 수 있다[3].

본 시스템의 클라이언트 측면에서는 WSDL에 명시된 클라이언트 측 Handler를 보고 서버 측 로직을 호출하기 전에 Handler를 호출하도록 구성하며 서버 측면에서는 Artifacts 컴파일러를 통해 타이(Tie)를 생성할 때 타이(Tie)에서 수행할 핸들러들을 명시한 후 타이(Tie)객체에 반영하는 방식으로 지원하고 있다.

EJB2.1의 웹서비스에서 지원해야 할 보안의 기능은 기본적으로 J2EE의 보안과 개념이 동일하다. 단 인증의 방법은 HTTP Basic Auth와 HTTP Symmetric Auth만을 제공하도록 하고 있다. 이러한 인증은 타이(Tie)에 설정된 핸들러들이 호출되기 전에 이루어져야 한다. 그 이유는 핸들러에서 비즈니스 로직을 수행할 수도 있기 때문이다. 이를 위해 본 시스템에서는 URI 매핑 후 인터셉터 단에서 사용자의 인증과 접근제어와 관련된 기능을 수행하도록 설계하였다.

7. 결론

본 논문에서는 EJB서버 시스템을 위한 EJB2.1 규격의 웹서비스 지원 모듈의 요구사항과 설계사항들에 대해 설명하였다. 본 논문의 EJB 표준 규격은 아직 미완성 상태이나 현재의 기준으로 볼 때 큰 변화는 없으리라 본다. 그러나, 추후 완성되는 2.1 표준 규격에 따라 구현된 본 시스템을 검증할 필요가 있으며 구현된 시스템의 성능 측면에서의 비교 분석이 필요하다.

참고문헌

[1] Linda G. Demichiel 외 2인, "Enterprise JavaBeans Specification, Version 2.1," Proposed Final Draft 2, Sun Microsystems, 2003

[2] David A. Chappell, Tyler Jewell "Java Web Services" O'RellLLY

[3] "Java API for XML-based JAX-RPC 1.0", SUN JSR-101