

컴포넌트 기반 웹서비스를 위한 어플리케이션 프레임워크

이용환*, 민덕기
건국대학교 컴퓨터공학부

An Application Framework for Component Based WebService

Yonghwan Lee*, Dugki Min

Division of Computer Science and Engineering, Konkuk University

E-mail : yhlee@cse.konkuk.ac.kr, dkmin@konkuk.ac.kr

요약

기업의 Business Process가 복잡해지고 B2B가 증가로 인하여 기업 내 혹은 기업간 Business Process를 통합하고 자동화하여 효율적으로 업무를 처리할 때 SOA 기반의 비즈니스 프로세스의 통합 틀의 필요하다. 본 논문에서는 재 사용성과 변경 용이성을 위해 개발한 다양한 업무 EJB 컴포넌트 형태들을 서비스 기반 아키텍처 상으로 전환하기 위한 서비스 기반 애플리케이션 개발 프레임워크를 제시한다. 제시한 서비스 기반 프레임워크는 기존 CBD 기반 업무 컴포넌트를 웹 서비스로 전환하기 위한 자동화 틀뿐만 아니라 웹서비스와 EJB까지 포함한 형태로 복합 애플리케이션을 개발할 수 있는 Orchestration Server까지 포함하고 있다.

1. 서론

최근의 대규모 비즈니스 소프트웨어들은 기능적인 측면이나 품질적인 측면에서 요구사항이 날로 복잡하여 시스템 개발에 많은 인력과 비용 및 시간이 소모되고 있으며, 개발이 완료된 후에 에러가 발생할 확률도 시스템의 복잡도에 비례하여 증가하고 있다. 이러한 문제를 해결하기 위한 많은 SI 기업들은 CBD 기반 애플리케이션 프레임워크 형태로 프로젝트를 진행하고 있다[1].

기업의 Business Process가 복잡해지고 B2B가 증가로 인하여 기업 내 혹은 기업간 Business Process를 통합하고 자동화하여 효율적으로 업무를 처리하는 SOA 기반의 비즈니스 프로세스의

통합 틀이 필요하다[2, 3].

SOA기반의 비즈니스 프로세스 통합을 위한 대 상들에는 웹서비스만 존재하는 것이 아니라 EJB 형태의 컴포넌트, 레가시 시스템까지 포함하고 있다[4, 5]. 따라서 많은 기업들은 재 사용성과 변경 용이성을 위해 개발한 다양한 업무 EJB 컴포넌트 형태들을 서비스 기반 아키텍처 상으로 전환하기 위하여 서비스 기반 애플리케이션 개발 프레임워크가 필요하다

서비스 기반 비즈니스 프로세스 통합에는 기존 EAI계열의 워크플로우 엔진들이 하는 것처럼 기존 레가시를 기반으로 하면서 웹 서비스를 수용하기 위해 별도의 자신만의 명세를 통해 웹 서비스

를 수용하는 것이 아니라 웹 서비스 기반 프로세스를 정의하기 위한 사실상 국제 표준인 BPEL4WS을 기반으로 웹 서비스뿐만 아니라 EJB 들까지 포함해서 실행할 수 있는 복합적인 Orchestration Server가 필요하다[6].

본 논문에서는 재 사용성과 변경 용이성을 위해 개발한 다양한 업무 EJB 컴포넌트 형태들을 서비스 기반 아키텍처 상으로 전환하기 위한 서비스 기반 애플리케이션 개발 프레임워크 아키텍처를 제시한다. 제시한 서비스 기반 프레임워크는 기존 CBD 기반 업무 컴포넌트를 웹 서비스로 전환하기 위한 자동화 툴뿐만 아니라 웹서비스와 EJB까지 포함한 형태로 복합 어플리케이션을 개발할 수 있는 Orchestration Server까지 포함하고 있다.

논문 구성은 다음과 같다. 2장에서는 비즈니스 프로세스 관련 표준과 프로세스 처리 엔진에 대한 관련 연구를 소개한다. 3장에서는 CBD 기반 어플리케이션 프레임워크 아키텍처를 제시하고 4장에서는 CBD기반 어플리케이션 프레임워크 상에 존재하는 컴포넌트를 서비스 기반 아키텍처로 전환하기 위한 아키텍처를 제시한다. 5장에서는 결론을 기술한다.

2. 관련 연구

비즈니스 프로세스 관련 표준 기술로는 먼저 BPEL4WS가 존재한다[7]. BPEL4WS는 BEA Systems, Microsoft, IBM에서 제정한 웹서비스 환경의 비즈니스 프로세스를 정의한 언어로서 IBM의 WSFL과 Microsoft사의 XLANG을 통합하여 대체된 웹서비스 통합 언어이다. 두 번째로 BPMN은 비즈니스 프로세스 다이어그램(BPD)에서 비즈니스 프로세스를 시각적으로 모델링하기 위한 그래픽 표기법(Graphical Notation)으로 BPMI(Business Process Management Initiative)표준화기관에서 개발한 표준 언어이다[8]. 세 번째로 BPML은 비즈니스 프로세스를 모델링하기 위한 메타언어로서 2002년 BPMI 표준화 기관에 의

해서 제안되었고, 여러 수준의 액티비티와 트랜잭션, 데이터 관리, 동시성, 예외 처리, 작동 체계를 포함하고 있다. 또한 이질적인 시스템과 모델링 툴 간에 프로세스 정의를 교환하고 유지할 수 있도록 XML Schema로 문법을 제공한다[9, 10].

비즈니스 프로세스 엔진 관련해서 먼저 BPEL4SJ는 BPEL4WS를 생성하고 실행하기 위한 플랫폼이며 IBM(사)에서 개발한 Orchestration 서버이며, 동적 협업 웹서비스(Dynamic Composite Web Service) 제공, BPEL4WS 문서의 유효성을 검사하는 툴 등을 제공한다[11]. 두 번째로 Collaxa BPEL4WS Server는 BEA(사)의 WebLogic과 JBOSS와 같은 Web Application Server와 통합되어 구동 되는 Collaxa(사)에서 개발한 BPEL4WS Orchestration 서버이다. J2EE 기반의 아키텍처를 제공하며 클러스터링 제공, BPEL4WS 실행 시 트랜잭션 처리 보장, 시각적인 감시(audit) 추적으로 비즈니스 프로세스 실행 모니터 등의 기능을 가지고 있다[12]. 세 번째로 BizTalk Server는 SOA 기반의 엔터프라이즈 어플리케이션 및 비즈니스 프로세스 통합 플랫폼 환경에서 Microsoft(사)가 개발한 BPEL4WS를 인식하는 BPEL4WS 엔진으로 분산된 기초 위에 유연한 컴포넌트의 "Loose coupling" 방식을 이용한다[13]. 프로세스의 방해 없이 어떠한 프로세스 Activity나 컴포넌트의 추가, 제거, 재구성 등이 가능하다. 기본적으로 Long-running, 확장성 있는 비동기 트랜잭션을 지원 등의 기능을 가지고 있다.

3. CBD 기반 어플리케이션 프레임워크

그림 1은 본 논문에서 제시한 프레임워크의 추상적인 아키텍처이다. 제시한 프레임워크는 기술적인 공통 컴포넌트인 기술 공통 컴포넌트, 프리젠테이션 단과 비즈니스 로직 단을 위한 프레임워크, 개발과 유지보수를 위한 어드민 툴킷으로 구성된다.

프리젠테이션 프레임워크는 클라이언트 브라우

저를 통해 요청을 받아 세션관리, 보안, 데이터포맷 변환을 처리하고 비즈니스 로직을 가지고 있는 EJB 모듈에 요청을 전송한다. 비즈니스 로직 프레임워크는 브라우저로부터 들어오는 클라이언트 요청을 처리하기 위한 비즈니스 로직을 처리하며 관계형 데이터베이스나 레가시 시스템과 연동해 CRUD(Create, Read, Update, Delete)을 수행한다. 어드민 툴킷은 개발, 유지보수, 관리를 용이하게 하기 위한 툴킷을 제공한다.

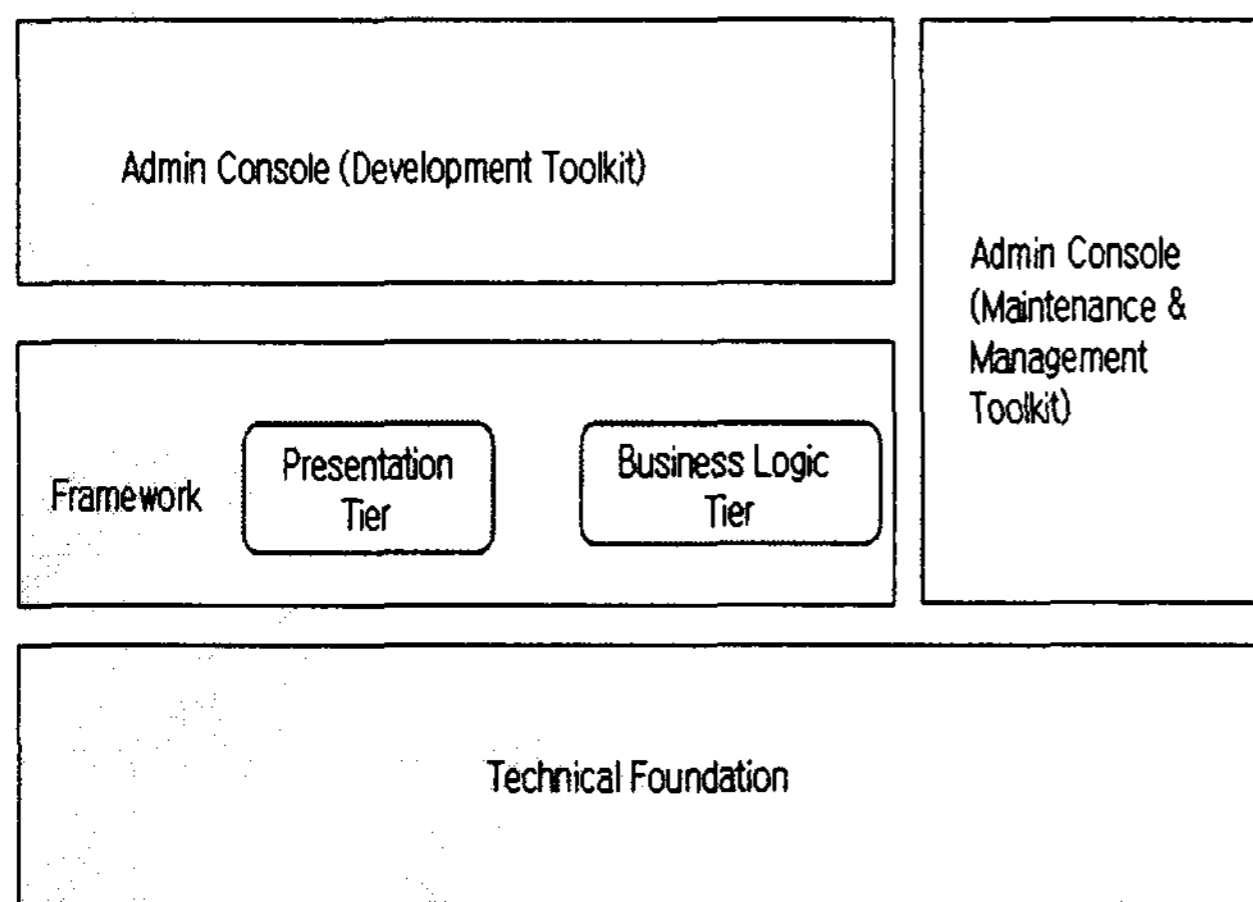


그림 1 추상적 프레임워크 아키텍처

그림 2는 어플리케이션 프레임워크가 가지는 전체 아키텍처이다. 프레임워크는 비즈니스 로직을 EJB 컴포넌트 형태로 개발하기 위한 비즈니스 로직 프레임워크와 컴포넌트를 기반으로 어플리케이션을 만들기 위한 프리젠테이션 단 프레임워크로 구성된다.

비즈니스 로직 프레임워크는 Front, System Interface, Business Interface, Data Abstract와 같은 네 개의 레이어로 구성된다. Front 레이어는 모든 EJB요청에 대한 진입점 역할을 수행하며 보안, 로그 그리고 날짜와 같은 모든 업무 공통 로직을 수행한다. System Interface 레이어는 각 서버 시스템이 외부 클라이언트에게 제공하는 서비스에 대한 인터페이스를 제공하며 Business Interface에 대한 캡슐화 역할을 수행한다. Business Interface 레이어는 기업의 비즈니스 로

직 업무를 구현하고 있으며 Data Abstract 레이어는 비즈니스 로직 처리를 위해 관계형 데이터베이스나 레가시 시스템에 CRUD를 수행한다.

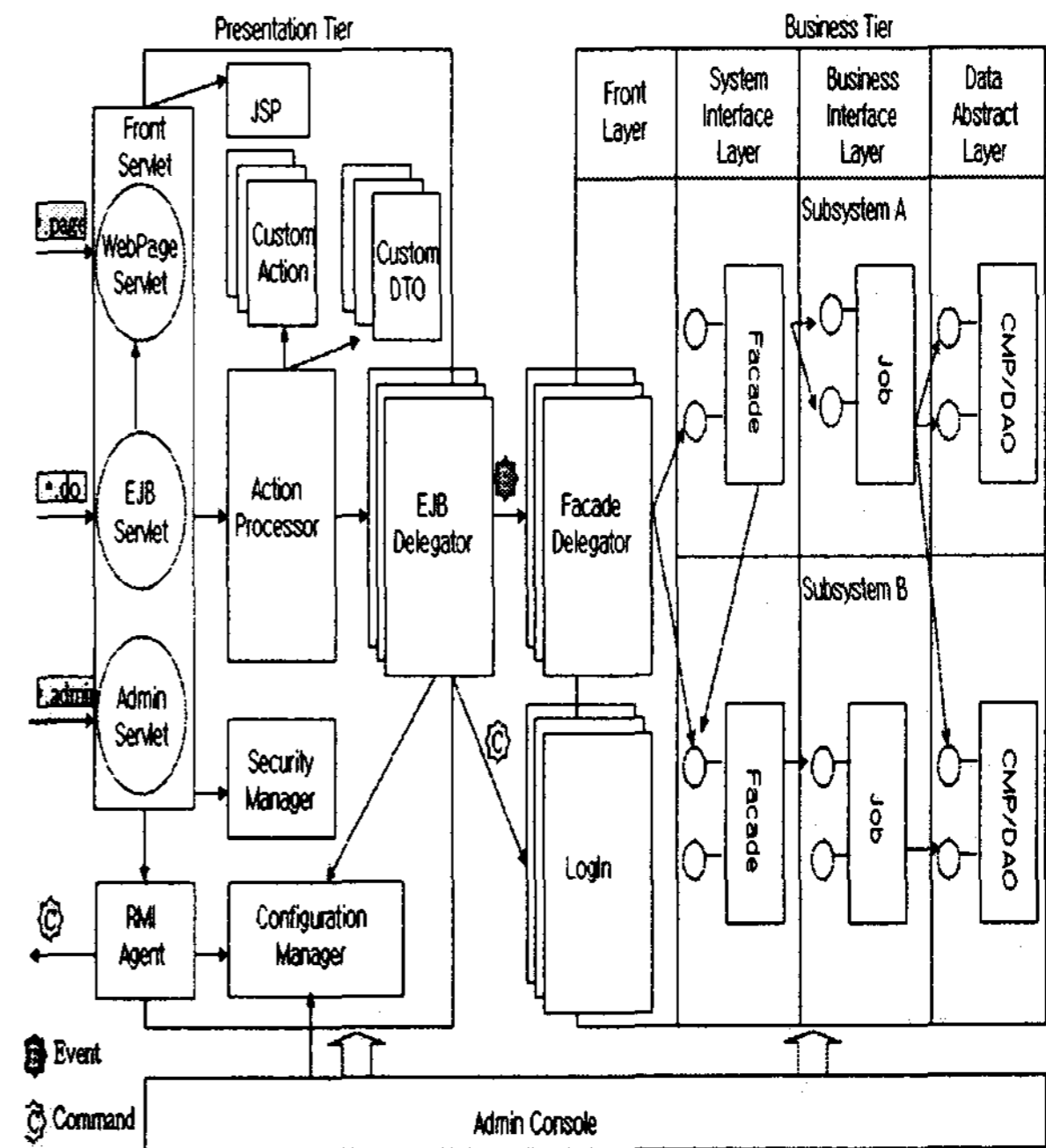


그림 2 CBD 프레임워크 전체 아키텍처

프리젠테이션 단의 FrontServlet은 *.page, *.do, *.admin, *.login 같은 요청 URL 패턴에 따라 클라이언트 요청을 처리하는 각각의 서블릿들이다. *.do인 경우에는 EJBServlet이 요청을 받아서 비즈니스 로직을 처리하며 *.page는 PageServlet이 받아서 화면 구성을 위한 처리를 수행하는데 어드민 콘솔에서 각 업무 구분 별로 설정한 페이지 구성 정보에 따라 UI쪽 작업을 수행한다.

FrontServlet이 처리하는 URL 패턴에는 업무 카테고리 정보가 있다. 예를 들어 클라이언트가 http://localhost/user.list.do와 같은 *.do 형태의 URL 패턴을 사용했다면 어드민 콘솔상의 트리 계층 구조상의 업무 분류에는 User라는 업무 카테고리 이름이 존재하며 그 하위 트리에는 list라는 서버 업무 카테고리가 존재한다. Admin Console 상의 계층형 각 업무 카테고리 별로 혹은 메소드를 가지고 있는 클래스 이름, 룰 그리고 설정 정보들

을 지정할 수 있으며 또한 업무 카테고리 별로 Action, CDTO, EJB Delegator 그리고 비즈니스 로직을 가지는 EJB 빈 코드를 개발하고 관리할 수 있다.

Action Processor는 UI상의 데이터를 CDTO에 넣은 후에 그 결과를 액션 객체에 넘겨주는 역할을 수행한다. CDTO(Custom Data Transfer Object)는 UI의 데이터를 담기 위한 객체이다. 액션객체는 ActionProcessor에서 호출하는 혹은 메소드를 가지고 있으며 EJB를 사용하지 않을 시 프리젠테이션 단의 비즈니스 로직을 구현한다. CDTO나 액션객체는 요청시점에 생성하지 않고 어드민 콘솔에서 각 업무 카테고리 별로 구현하고 컴파일 한 후에 객체 풀에 들어가 있는 객체를 사용한다.

Configuration Manager는 어드민 콘솔상에서 트리 계층 형태의 업무 카테고리 별로 설정된 XML 기반 정보를 관리한다. 어드민 콘솔상에서 설정 정보가 변경되며 Configuration Manager의 정보도 동적으로 변경된다. FrontServlet은 바로 URL상의 해당 업무 카테고리를 추출하고 이 카테고리 별로 설정된 정보들을 바로 Configuration Manager을 통해 가져온다. EJBDelegator는 EJB 빈과 통신을 담당하며 CDTO나 액션객체와 같이 어드민 콘솔상에서 각 업무 카테고리 별로 구현해서 컴파일 한 후에 객체 풀에 들어간다.

Event 객체의 Request는 UI에서 넘어온 값을 저장하는 객체이고 Response는 EJB에서 비즈니스 로직 처리후의 결과 값을 저장하는 객체이다. CommonDTO는 비즈니스 로직 요청시 마다 가지고 가야 하는 정보를 담고 있는 객체이다. 문자열 타입의 EJB Identifier는 특정 서브시스템의 Facade 컴포넌트 메소드를 호출하기 위해 필요한 Reflection 기법에 필요한 정보를 가져오는 구분자이다. Exception Manager는 예외가 발생한 경우 예외 처리를 담당한다.

EJB을 호출하고 난 후에 결과 값은 CDTO 형

태로 반환되어서 Admin Console에 지정된 다음 URL로 전송된다. 예를 들어 다음 URL 값이 *.page 패턴이면 EJB에서 반환된 데이터를 통해 화면을 구성하기 위한 WebPageServlet으로 제어가 이동하며 해당 CDTO 데이터도 전송된다.

Facade Delegator는 모든 EJB 요청 진입 점으로서 프레임워크에서 제공하는 EJB 컴포넌트이다. 로그 처리, 예외처리 그리고 날짜 계산과 같은 유스케이스들이 공통으로 처리해야 하는 작업을 수행한다. Facade Bean은 분석 모델의 각 유스케이스 별로 존재하는 시스템 인터페이스를 구현하는 EJB 빈으로서 주로 비즈니스 로직을 가지고 있는 Job 빈을 호출하거나 다른 서브시스템의 비즈니스 로직과의 연동과 같은 비즈니스 프로세스를 조정하는 역할을 수행한다.

비즈니스 로직 빈은 비즈니스 도메인의 비즈니스 로직을 구현한 EJB 빈이다. Persistent 빈은 비즈니스 로직 빈과 관련해서 관계형 데이터베이스, XML 그리고 파일 시스템등과 관련해 비즈니스 데이터의 CRUD(Create, Read, Update, Delete)와 같은 기능을 제공한다. Persistent와 관련해 EJB의 CMP나 BMP를 지원하기 위한 OR 맵핑 지원 툴킷을 제공함으로써 코드 자동화 기능을 제공하고 있다. 또한 JDBC와 관련된 DAO(Data Access Object) 코드 자동 생성 기능을 제공하기 위해 SQL기반 맵핑 툴킷을 제공한다.

4. 웹서비스 기반 어플리케이션 프레임워크 아키텍처

그림 3은 CBD 기반 어플리케이션 아키텍처를 서비스 기반 아키텍처로 변환하기 위한 구성도이다. EJB Adapter는 3장의 CBD 어플리케이션 프레임워크의 프리젠테이션 단의 EJB Delegator와 비즈니스 로직 단의 Façade Delegator이다.

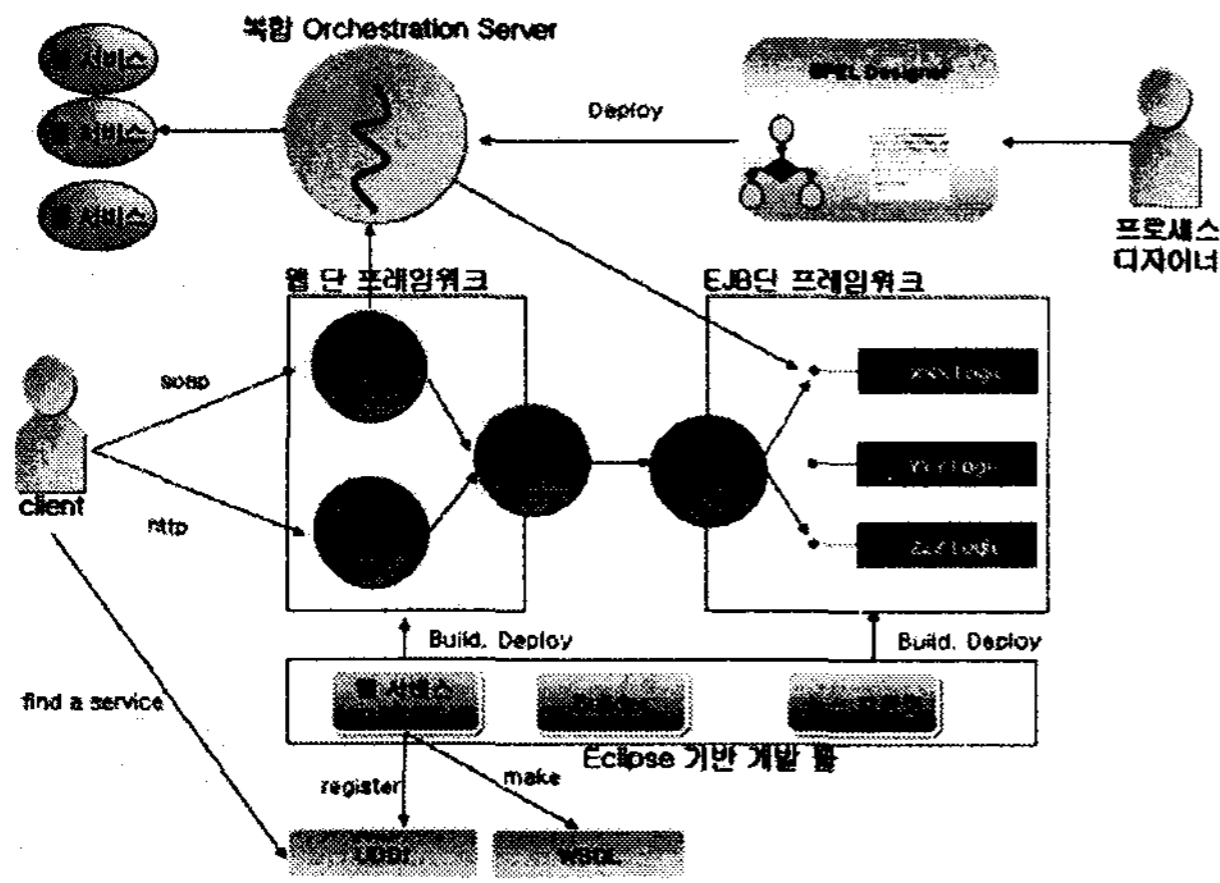


그림 3 웹서비스 기반 어플리케이션 프레임워크

프로세스 디자이너는 BPEL Designer 를 사용해 웹 서비스뿐만 아니라 EJB 까지 프로세스 정의를 수행한다. 정의된 프로세스를 복합 오케스트레이션 서버에 배포함으로써 서버가 서비스를 받을 준비를 한다. 클라이언트는 HTTP 나 SOAP 프로토콜을 사용해서 기존 CBD 어플리케이션 프레임워크가 가지고 있는 Presentation Tier 프레임워크로 요청을 보낸다.

그림 4 는 서비스 기반 어플리케이션 프레임워크의 멀티 채널 확장성을 묘사하고 있다.

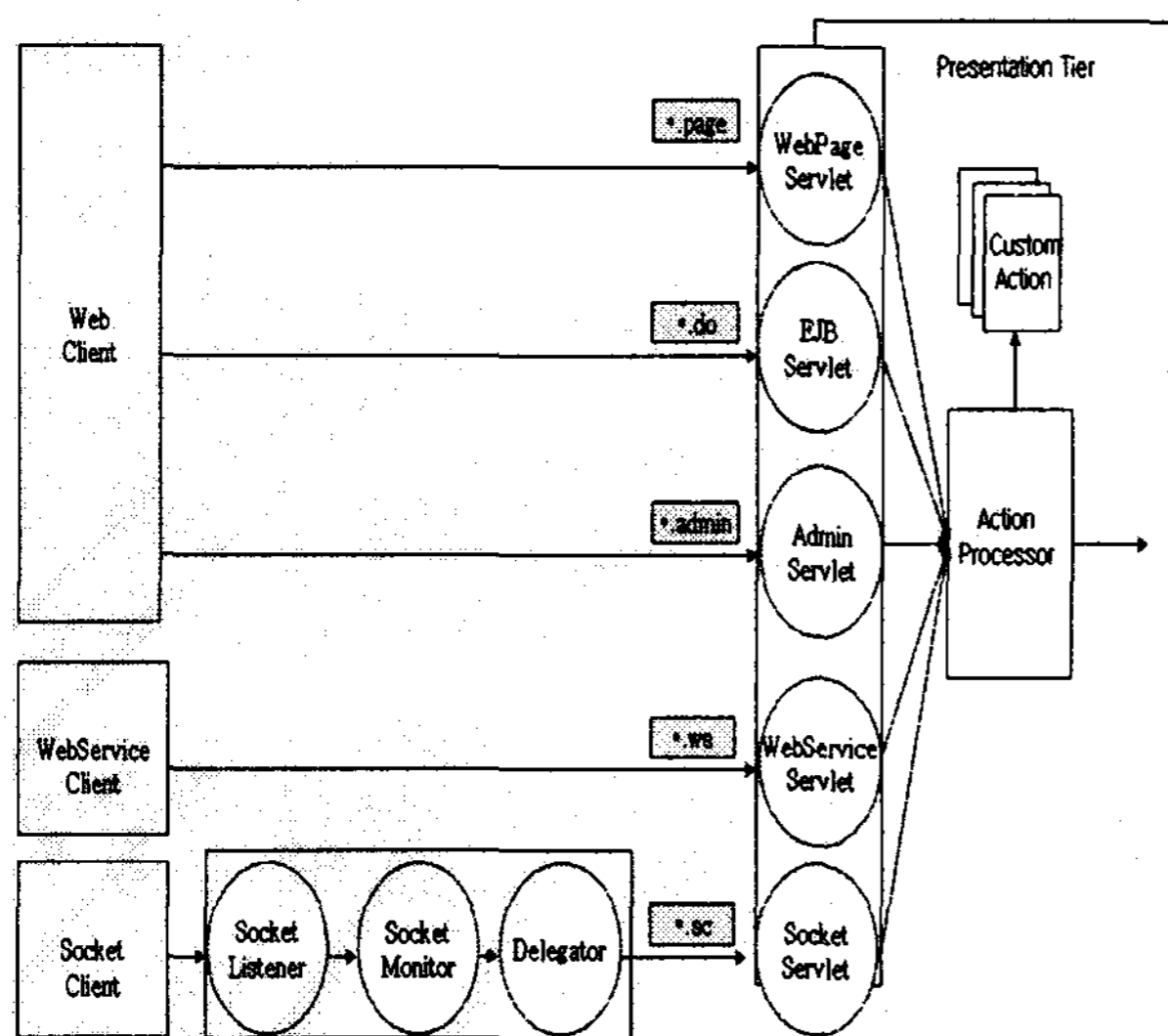


그림 4 멀티채널 구현 확장성

TCP 소켓을 사용하는 클라이언트를 위해 Socket Listener 을 만들어서 웹 어플리케이션 초기화 시 구동 되도록 하고 있다. 소켓으로 들어오는 클라이언트에 대한 모니터링 기능을 제공하기 위해 Socket Monitor 객체를 제공하고 있으며 Socket Monitor 는 받은 정보를 어드민 콘솔에게 제공한다. 다음으로 소켓으로부터 들어온 요청을 처리하기 위해 받은 데이터를 Delegator 를 통해 *.sc URL 패턴에 따라 Socket Servlet 으로 전송해서 처리한다. 웹 서비스 클라이언트가 접속할 경우 웹 클라이언트처럼 HTTP 프로토콜 위에 SOAP 메시지가 전달되기 때문에 WebService Servlet 을 생성해서 SOAP 메시지를 처리한다. 기타 다른 이기종 시스템과 연동 시 HTTP 프로토콜 기반 위에서 통신을 수행하는 경우에는 각 URL 패턴에 따라 해당 서블릿을 생성 함으로서 처리하고 만일에 요청을 받아 들이기 위해 데몬(Daemon)이 필요한 경우에는 TCP 소켓과 같이 요청을 받아들이기 위한 데몬을 생성해서 요청을 받은 후에 다시 특정 URL 패턴에 해당하는 서블릿을 만들어 처리한다.

SOAP 프로토콜로 들어온 메시지는 Front Servlet 의 한 종류인 Service Servlet 이 받아서 처리하는데 만일 SOAP 메시지의 헤더 부분에 BPEL4WS 처리를 요하는 정보가 있다면 이 요청은 복합 Orchestration Server 를 통해서 처리된다. 이러한 복합 Orchestration Server 를 통해서 처리될 수 있는 것은 웹서비스 뿐만 아니라 EJB 까지 포함할 수 있다.

기존 CBD 어플리케이션 프레임워크가 가지고 있는 어드민 콘솔은 Eclipse 기반의 툴이다. 이 툴에는 관련된 소스를 자동으로 생성하는 소스 자동화 툴과 컴파일 및 패키징 그리고 배포를 위한 Builder 툴이 존재한다. 기본 EJB 컴포넌트를 웹서비스 형태로 전환하기 위해서는 어드민 콘솔상에 웹 서비스 환경을 지원할 수

있도록 하는 웹 서비스 지원 틀이 필요하다. 이 틀은 EJB Local 이나 Remote Interface 를 기반으로 WSDL 을 생성하고 UDDI 에 해당 서비스를 등록하는 기능을 수행한다[14].

클라이언트는 HTTP 프로토콜 상에서 기존 CBD 컴포넌트를 호출할 수 있을 뿐만 아니라 UDDI 에서 해당 서비스에 대한 WSDL 을 찾아 웹서비스를 호출할 수 있다. 만일 클라이언트가 웹서비스를 호출했다면 Service Servlet 이 요청을 받아서 SOAP 헤더 안의 정보가 BPEL4WS 을 처리하는 정보가 있다면 Service Servlet 은 복합 Orchestration Server 를 호출한다. 그렇지 않고 특정 서비스를 요청한다면 Service Servlet 은 EJB 컴포넌트를 호출해 비즈니스 로직을 처리해 클라이언트에게 결과를 SOAP 메시지를 만들어 전송한다.

복합 Orchestration Server 는 프로세스 디자이너가 BPEL Designer 를 사용해 작성한 후에 Orchestration Server 에 배포한 프로세스 정의에 따라 프로세스를 진행한다. Orchestration Server 는 새로이 개발한 웹서비스 뿐만 아니라 기존 EJB 기반 CBD 컴포넌트까지 포함해서 프로세스를 통합할 수 있다. 이때 Orchestration Server 에서는 EJB Adaptor 가 수행하는 것과 같은 Adapter 를 통해 EJB Delegator 를 호출한다.

5. 결론

본 논문에서는 재 사용성과 변경 용이성을 위해 개발한 다양한 업무 EJB 컴포넌트 형태들을 서비스 기반 아키텍처 상으로 전환하기 위한 서비스 기반 애플리케이션 개발 프레임워크 아키텍처를 제시한다. 제시한 서비스 기반 프레임워크는 기존 CBD 기반 업무 컴포넌트를 웹 서비스로 전환하기 위한 자동화 틀을 제공함으로써 기존 EJB 상에 존재하는 Local 이나 Remote 인터페이스를 WSDL 로 변환 후에 UDDI 에 등록 함으로서 웹서비스 클라이언트가

EJB 컴포넌트 형태로 구현된 서비스를 사용할 수 있게 한다. 뿐만 아니라 웹서비스와 EJB 까지 포함한 형태로 복합 어플리케이션을 개발할 수 있는 Orchestration Server 를 제시 함으로서 기업 내외 비즈니스 프로세스를 효과적으로 통합할 수 있는 장점이 있다.

[참고문헌]

- [1] D.C. Luckham, J. Vera and S. Meldal, Key Concepts in Architecture Definition Language, in Foundations of Component-Based System, Ed. Gary T, Leavens and Murali Sitaraman, Cambridge University Press 2000
- [2] Shim, S.S.Y., Pendyala, V.S., Sundaram, M., and Gao, J.Z., "B2B E-Commerce Frameworks", Computer, IEEE, October 2000.
- [3] Mercator Software Inc., " Integration Technology Vision", <http://www.mercator.com>, April 2002. 2000.
- [4] Linthicum D., Enterprise Application Integration, Addison-Wesley 2000.
- [5] Wheatstone Inc., "Application Integration - WhitePaper", <http://www.wheatstone.net/whatwedo/appintwp.htm>.
- [6] IBM, WebSphere Business Integration Server, <http://www-306.ibm.com/software/integration/wbiserver/>
- [7] Francisco Curbera, Yaron Goland, Johannes Klein, Frank Leymann, Dieter Roller, Satish Thatte, and Sanjiva Weerawarana. Business Process Execution Language for Web Service (BPEL4WS) 1.0. Published on the World Wide Web by BEA Corp., IBM Corp. and Microsoft Corp., August 2002.
- [8] OMG, BPMI.org, <http://www.bpmn.org/>
- [9] ebBPMLOrg, <http://www.ebpml.org/bpml.htm>.

[10]Assaf Arkin. Business process modeling language - bpm1.0 last call working draft. Published on the World Wide Web by BPMI.org, 2002.

[11]IBM et al., Business Process Execution Language for Web Services version 1.1, <ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>.

[12]Oracle, Oracle BPEL Process Manager 2.0 Quick Start Tutorial, <http://www.oracle.com/technology/products/ias/bpel/pdf/orabpel-QuickStart.pdf>.

[13] Microsoft, Microsoft BizTalk Server 2004, <http://msdn.microsoft.com/biztalk/>

[14] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web Services Description Language (WSDL) 1.1. Published on the World Wide Web by W3C, Mar 2001.