

Service Framework

컴포넌트 개발을 위한 "Service Framework"

(주)삼성SDS(대표:김홍기
www.sds.samsung.co.kr)가
개발한 Service Frame-
work은 웹 기반의 비즈니스
어플리케이션을 컴포넌트로
신속하고 견고하게 개발할
수 있도록 다양한 서비스(컴
포넌트)와 개발 표준을 제공
한다. 이 프레임워크가 제공
하는 재사용 가능한 20개의
컴포넌트와 MVC기반의 아
키텍처는 J2EE 환경 위에서

개발되는 모든 어플리케이션에 수평적으로 적용될 수 있다.

Service Framework은 컴포넌트 개발/조립, 사용자 인터페이스 구성에 필
요한 기반 환경(자원접근 통제, 인증, 로깅, 풀링 등)의 모든 요구사항을 충
족시킬 수 있다. Service Framework은 기능성, 가변성, 성능향상의 3가지
에 관점에서 모든 어플리케이션 개발 시 요구되는 기반 서비스를 컴포넌트
의 집합으로 제공한다. 또한 유연한 확장 구조를 통해 도메인에 특성에 맞
는 컴포넌트를 지속적으로 추가하여 업종별 어플리케이션 프레임워크를 구
축할 수 있는 기본적인 토대를 제공한다.

Service Framework은 이미 금융(농협, 우정국)과 의료(원자력병원) 같은
대형 프로젝트 적용 사례를 확보하고 있다. 적용 사례를 통해 검증된
Service Framework의 효과에는 개발 생산성 향상, 품질과 성능 향상, 유지
보수 비용 절감 등이 있다. Service Framework의 효과적인 적용은 아키텍
처와 컴포넌트의 재사용을 통해 어플리케이션 개발 비용과 시간을 크게 단
축시킨다. 또한 어플리케이션 개발 시 공통으로 사용되는 기본적인 기능요
소의 제공은 개발자가 오직 비즈니스 영역에만 집중하도록 함으로써 어플
리케이션의 품질 향상시킨다. 견고한 프레임워크 위에서 개발된 시스템은
유지보수를 최소화 할 수 있는 기본요건을 가지며, 또한 시스템의 변경을
대비한 다양한 기능(Configuration, 이식성 등)의 사용을 통해 유지보수 비
용을 감소시킨다. 문의 : 031)785-4505 담당 : 김창제 과장



Service Framework

1. 작품명 : Service Framework

2. 제작자 : (주) 삼성 SDS

대표자 : 김흥기

개발참여자 : 김창제, 원성호, 정광현, 최연정

주소 : (463-810) 경기도 성남시 분당구 구미동 159-9번지
성지빌딩 5층

전화 : 031) 785-4505

팩스 : 031) 785-4451

E-mail : changje.kim@samsung.com

3. S/W 요약설명

Service Framework은 서버에서 운영되는 웹 기반의 비즈니스 어플리케이션을 MVC 아키텍처 기반으로 신속하고 견고하게 개발할 수 있도록 다양한 서비스와 설계 패턴을 제공한다. 이 프레임워크는 J2EE 환경 위에서 개발되는 모든 어플리케이션에 수평적으로 적용될 수 있는 재사용 가능한 SW 컴포넌트들로 구성된다. Service Framework은 개발자가 오직 비즈니스 로직에만 전념하여 어플리케이션을 개발하도록 할 뿐만 아니라, 비 기능적인 요소에 대한 개발 영역 분리 통해 어플리케이션의 복잡도를 줄일 수 있도록 한다. 이러한 프레임워크의 효과적인 적용은 SW의 생산성과 품질을 향상시킨다.

4. 개발 배경

SW의 품질과 생산성을 높이는 핵심적인 요소 중 하나는 재사용이다. 재사용될 수 있는 산출물에는 실행 가능한 바이너리 코드, 분석/설계 사양, 테스트 데이터, 문서, 아키텍처 등 여러 가지 형태가 존재한다. 이러한 재사용 단위 중 바이너리 코드와 소스 코드는 가장 흔히 재사용되는 단위이다. Service Framework은 모든 도메인에서 재사용될 수 있는 J2EE 기반의 컴포넌트를 포함하는 라이브러리와 설계 패턴을 잘 정의된 문서와 바이너리 코드로 제공한다.

Service Framework은 삼성SDS의 J2EE 기반의 프로젝트, 상업용 컴포넌트, 오픈 소스 프로젝트, 설계 패턴 등을 분석 하여 비즈니스 어플리케이션의 개발과정에서 반복적으로 요구되는 서비스와 WEB 기반의 견고한 아키텍처 모델을 제공한다. Service Framework이 가지는 12가지의 기본 서비스는 모든 도메인에 공통적으로 적용되어 쉽게 사용될 수 있는 인터페이스와 사양을 제공한다. 또한 Service Framework은 특정 프로젝트에서만 적용되는 서비스를 추가할 수 있는 기법을 제공한다. 이와 같은 성질은 Service Framework이 특정 도메인에 적합한 형태로 쉽게 확장될 수 있는 유연한 아키텍처를 제시함으로써 서비스 중심의 프로그램을 가능하게 한다. 서비스 중심의 프로그램은 서비스를 사용하는 비즈니스 어플리케이션에 그 인터페이스만을 공개하고 내부 구현을 은닉함으로써 어플리케이션의 기능적인 요구사항의 변경에 쉽게 대처할 수 있는 방안을 제시할 뿐만 아니라, 표준화된 구조의 제시로 시스템의 복잡성을 크게 줄일 수 있게 한다. Service Framework은 비즈니스 어플리케이션 구축 시 요구되는 다음과 같은 목적을 달성하기 위해 개발되었다.

1) 가변성 제공

비즈니스 컴포넌트가 코드의 수정 없이 재사용되기 위한 핵심적인 요건 중 하나는 기능적인 요구사항의 변경이나 외부 운영환경의 변화에 쉽게 적용할 수 있는 구조이다. Service Framework에서는 변경 가능성이 높은 정보는 외부에서 존재하게 하고 운영시점에 동적으로 이 정보에 접근할 수 있는 체계적인 방법을 제공한다.

2) 기능제공

Service Framework은 어플리케이션 개발 과정에서 항상 요구되는 Logging, Sign-on, Error-handling과 같은 기본적인 기능을 제공한다. 이것은 개발자가 비즈니스 문제에만 전념할 수 있도록 할 뿐만 아니라,

시스템 영역의 기능을 분리함으로써 견고하고 신뢰성 있는 소프트웨어를 빠른 시간 내에 개발 가능하도록 한다.

3) 성능향상

어플리케이션 내에서 빈번하게 사용되는 데이터가 외부에 존재하는 경우는 Service Framework의 Cache에 이 데이터를 저장함으로써 시스템의 데이터에 대한 접근 부담을 줄일 수 있다. 또한 공유되는 객체에 대한 재사용이 높은 경우에도 그 생성 비용을 줄이기 위해 Cache를 사용할 수 있다. 어플리케이션 내의 복잡한 형태의 공통 요구 사항은 Service Framework의 서비스로 구현하여 소프트웨어의 성능을 향상시킬 수도 있다. Service Framework의 모든 서비스는 성능 향상을 위해 singleton을 기반으로 하며, 어플리케이션에서 서비스를 요청한 이후에 그 서비스에 대한 자원을 할당함으로써 사용되지 않는 서비스에 대한 자원 할당을 통제한다.

4) 표준화 제공

Service Framework의 서비스와 Web 기반의 아키텍처 모델 적용은 프로젝트 구성원이 동일한 관점으로 시스템을 바라볼 수 있는 내부 표준 시각을 제시한다. 이것은 어플리케이션의 근간이 되는 시스템 아키텍처의 초기 구축 부담을 줄이고, 같은 관점을 가지고 원활하게 의사소통을 할 수 있도록 한다.

5. 시스템 개요

1) 비즈니스 어플리케이션 개발과정에서 반복적으로 요구되는 서비스(컴포넌트)와 WEB 기반의 견고한 아키텍처 모델을 제공한다.

- 모든 도메인에 공통적으로 적용되어 쉽게 사용될 수 있는 인터페이스와 사양 제공
- Model2 아키텍처를 기반으로 하는 Web Application Framework 제공
- J2EE 기반의 어플리케이션 개발과정에서 빈번하게 사용되는 다양한 유틸리티를 제공

2) 특정 프로젝트에서 적용되는 서비스(컴포넌트)를 추가하는 기법을 제공하여, Service Framework의 확장이 가능하다.

3) 코드의 수정 없이 Configuration 정보의 수정으로 기능적인 요구사항 변경에 쉽게 대처할 수 있는 방안을 제시한다.

6. 시스템 특징

Service Framework은 J2EE 기반의 비즈니스 어플리케이션이 기본적으로 갖추어야 하는 성능, 가변성, 기능 요소를 완벽하게 제공한다. 또한 모든 서비스는 상속과 그 인터페이스를 통해 변경과 확장이 기존의 프레임워크에 영향 주지 않으면서도 가능하다. 이러한 특징은 비즈니스 어플리케이션이 가져야 하는 다양한 측면 중 비즈니스 영역을 제외한 나머지는 프레임워크 통해 모두 제공 받을 수 있도록 한다.

1) 기능 차별성

Service Framework은 새로운 서비스를 Plug-in할 수 있는 기능을 지원한다. 기본적으로 제공하는 하는 12개의 서비스는 모든 도메인에 수평적으로 적용 가능한 서비스로 구성되지만, 이 프레임워크를 사용하는 프로젝트에서는 도메인 종속적인 형태의 서비스를 추가하여 프레임워크의 확장이 가능하다. 또한 사용되지 않는 서비스를 프레임워크 내에서 제거할 수 있는 기능을 제공한다.

WAF은 프레젠테이션 레이어 개발에 필요한 모든 기능적인 요소를 아키텍처에서부터 Logging까지의 전 부분을 제공한다.

2) 독창성

기존에 존재하는 프레임워크는 대부분 라이브러리 중심의 소규모 기능을 지원하고, 상업용 프레임워크는 다양한 기능을 지원하나 그 구조가 매우 복잡하고, 어플리케이션 서버에 종속적인 형태를 지닌다. Service Framework은 가볍고 견고한 틀을 가지고 있으면서도, 어플리케이션 서버에 독립적인 시스템의 개발을 가능하게 할 뿐만 아니라 개발에 필요한 기능적인 모든 요소를 제공한다. 이러한 기능적 요소에 대해 개개의 항목으로 분리될 수 있는 서비스 중심의 프로그램을 가능하게 함으로써, 어플리케이션의 복잡도를 줄이고 각 단위의 재사용을 획기적으로 증대시킨다.

3) 성능 및 적용효과

Service Framework은 일반 비즈니스 어플리케이션과 달리 이 프레임워크를 적용하여 개발하는 어플리케이션의 성능을 보장하기 위한 Cache, JNDI, Pool과 같은 서비스를 제공한다. 프레임워크 그 자체의 성능 향상을 위해 서비스는 Service Framework이 시작될 때 오직 하

나의 인스턴스만을 생성하도록 하며 모든 시스템은 운영 시점에 이 서비스를 공유하는 기본 구조로 구성되어 있다. 이것은 객체 생성이나 그 객체가 가지는 자원할당에 대한 비용을 줄임으로써 전체적으로 시스템의 성능을 향상시킨다. 또한 Service Framework은 그 내부에서도 빈번하게 공유되는 객체나 데이터를 캐쉬에 저장함으로써 원격 서버나 하드디스크 같은 매체에 존재하는 정보의 접근에 따른 비용을 감소시킨다.

이 프레임워크를 적용하여 삼성 SDS의 금융개발 UNIT에서 개발한 농협 e-banking 시스템은 동시 사용자 200명을 기준으로 평균 3~4초 미만의 처리 응답 속도를 얻었다. e-banking 시스템의 테스트장비 IBM의 RS6000 Series이며, Application Server는 JEUS를 사용하였다. 또한 생산성 측면에서도 프레임워크가 가지는 WEB 아키텍처와 서비스 지향의 프로그램 설계 방식의 적용으로 전체 업무 중 20~30% 정도 대대 코드의 재사용 효과를 얻었다.

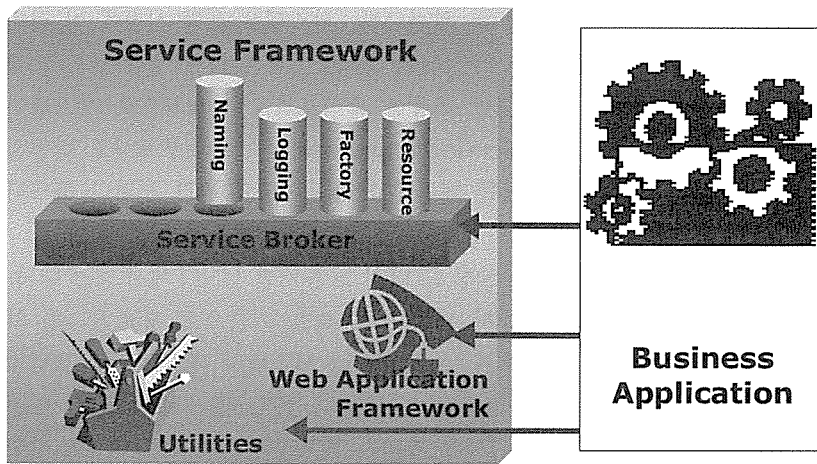
4) 시장성 및 우수성

최근의 엔터프라이즈 어플리케이션 개발 프로젝트의 기술환경은 크게 .Net Framework과 J2EE의 2가지 기술 환경으로 양분된다. J2EE의 시장은 2004년까지도 .Net Framework에 비해 시장 규모가 다소 우세할 것으로 예측된다. Service Framework은 이러한 J2EE 기술환경의 모든 프로젝트에 적용 가능한 서비스로 구성되어 있다. 또한 기존의 WAS 중심의 업체가 제공하는 대형 서버에 비해 가볍고 견고한 구조로 그 적용 범위는 점점 확대될 것으로 예측된다.

이 프레임워크의 적용은 어플리케이션이 가지는 다양한 측면의 문제 영역을 서비스와 WAF를 통해 제공함으로써 높은 품질과 향상된 성능을 가지는 어플리케이션을 저비용으로 빠른 시간 내에 개발 가능하게 한다. 기본적으로 Service Framework은 그 자체의 재사용뿐만 아니라 이 프레임워크를 적용하여 개발하는 어플리케이션도 재사용 가능한 컴포넌트로 개발할 수 있도록 유도하는 잠재된 설계패턴을 제시하고 있다.

7. 시스템 구성

Service Framework은 JVM위에서 비즈니스 어플리케이션이 사용 가능한 공통 영역의 레이어를 대상으로 개발되었다. Server Side 어플리케이션 개발을 지원하는 Service Framework은 다음 그림과 같이 세 부분으로 구성된다.



■Service Broker와 서비스

Service Broker는 서비스를 관리하며 서비스의 내부 구현은 어플리케이션 으로부터 감추어진다. 비즈니스 어플리케이션은 Service Broker를 통해 가져온 서비스의 인터페이스만을 사용하여 서비스의 기능을 제공 받는다. Service Broker가 관리하는 서비스는 운영시점에 Plug-in된다.

■Web Application Framework

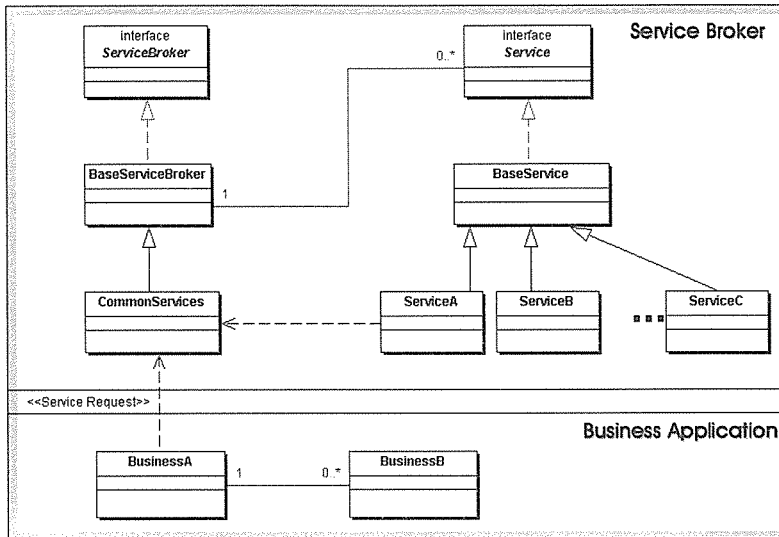
Model2 Architecture를 기반으로 하여 Web 기반의 어플리케이션 개발에서 요구되는 Sign-on, Page Template, Role, Screen-work flow 통제와 같은 기능을 제공한다.

■Utility

J2EE 기반의 어플리케이션 개발과정에서 빈번하게 사용되는 다양한 유틸리티를 제공한다.

1) Service Broker Model

Service Broker는 Plug-in되어 있는 다양한 서비스를 관리한다. Service Framework 내에 있는 서비스에 비즈니스 어플리케이션이 접근하기 위해서는 Service Broker에 서비스 요청을 해야 한다. Service Broker는 요청된 서비스를 찾아 그 인터페이스를 비즈니스 어플리케이션에 제공한다. 다음 클래스 모델은 Service Broker와 Service의 관계를 보여준다.



Service Framework의 Service Broker와 Service의 설계 원칙은 다음과 같다.

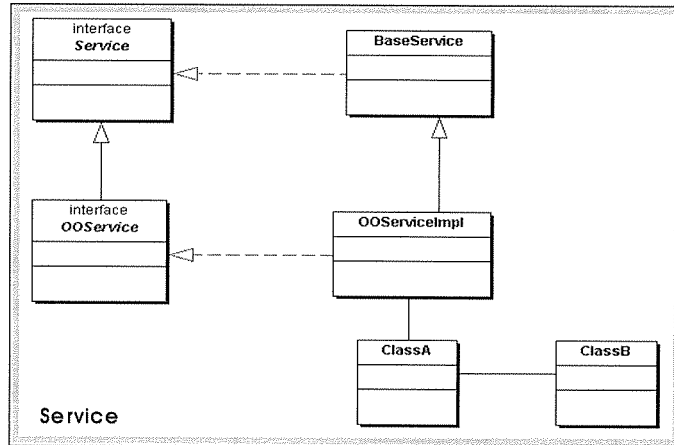
- interface를 정의한다.
- interface를 구현하는 default implementation class 정의한다.
- interface를 구현하는 클래스나 default implementation을 상속 받는 클래스를 정의한다.

이와 같은 설계 방식은 Service Framework 그 자체를 특정 프로젝트에 맞게 쉽게 확장할 수 있는 기본적인 틀을 제공할 뿐만 아니라, 어플리케이션이 Service Framework에게 요구하는 기능적인 변경사항에 대해서도 어플리케이션의 수정 없이 쉽게 대응할 수 있다는 장점을 갖게 한다.

2) Service Model

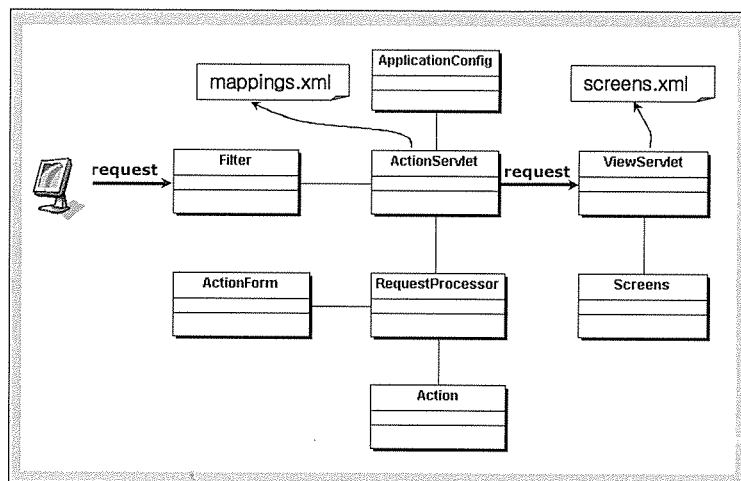
서비스는 비즈니스 어플리케이션 전체가 공유하여 사용되는 컴포넌트이다. Service Framework은 시스템이 제공해야 하는 다양한 기능을 재사용 가능한 서비스들로 분할하여 서비스 중심의 프로그램이 가능하도록 하는 개념을 제공한다. Service Framework은 이와 같은 형태의 12가지 서비스를 제공한다. 각 서비스는 모든 도메인에서 사용 가능한 기능을 대상으로 하고 있으며 아래와 같은 구조로 새로운 서비스를 프레임워크 내에 추가 가능하다. 다음 클래스 모델은 Service Broker가 관리하는 서비

스의 기본구조를 보여준다.



3) Web Application Framework

Enterprise Application의 개발범위 중 Presentation Layer는 서버사이드에 존재하는 로직이나 데이터 모델만큼 큰 비중을 차지한다. 특히 View Model에 대한 아키텍처의 부재는 개발에 상당한 혼선을 일으킬 뿐만 아니라, 고객의 잦은 요구사항 변화에 대응하기 어려운 단점을 가지게 된다. Service Framework내에 존재하는 Web Application Framework(WAF)은 View모델 자체를 다시 MVC 아키텍처 기반으로 구성할 수 있게 하는 기본적인 설계 패턴을 제공한다. 개발자는 내부의 복잡한 처리의 흐름에 대해서는 모두 프레임워크를 통해 기능을 제공할 수 있도록 구성되어있다. 다음은 WAF의 개념 모델이다.



8. 주요기능

Service Framework은 Service, Web Application Framework, Utilities로 구성된다. 여기에서는 각 구성 항목별로 개략적인 기능을 소개한다.

1) Service

① Resource Service

프로퍼티 파일내에 존재하는 데이터를 읽고 접근하게 하는 서비스로 Service Framework의 서비스에 대한 환경정보를 가지고 있다.

② Logging Service

Logging Service는 Application의 수행과정에 대한 정밀한 내용을 Logging을 통해 제공한다. Logging 결과는 후에 확인할 수 있도록 퍼 시스템스 매체에 저장되어야 한다.

LoggingService는 기본적으로 Jakarta Project의 Log4j-1.2.3을 Wrapping하여 개발하였다.

이 Service는 다음과 같은 기능을 제공한다.

■시스템이 가지는 기능 영역에 따라 다양한 종류의 Logger를 정의할 수 있다.

■Logging 결과를 저장하기 위해 6가지의 Destination을 가지고 있다. (File, Email, System Daemon, Database, Remote Server, Console)

■Logging은 내용에 따라 Debug, Info, Warn, Error Level중 하나를 선택하여 남길 수 있다. 이 Level은 Debug<Info<Warn<Error의 순서를 지니며, 특정 Level 이상에 대해서만 Logging을 하도록 제약할 수 있다.

③ Factory Service

Factory Service는 클래스의 인스턴스를 얻는 기능을 제공한다. 클래스의 인스턴스를 얻기 위해서는 클래스명만 사용하거나, 클래스명과 ClassLoader를 사용하여 얻을 수 있다. 일반적으로Factory Service를 사용하여 인스턴스를 얻게 되는 클래스는 상속관계를 가지게 되며, Application은 클래스의 인터페이스만 알면, 실제 Concrete Class가 어떤 것이냐에 영향을 받지 않기 때문에, Concrete Class의 변경이 용이하다.

④ Pool Service

Pool Service는 공유하지 않는 객체의 인스턴스를 Pooling하는 기능을 제공한다. Pool Service는 Factory Service를 상속받고 있으며, Factory Service 기능에 객체의 인스턴스를 pooling하는 기능이 추가된 Service라 할 수 있다.

비즈니스 어플리케이션이 Pool Service를 통해서 어떤 클래스의 새로운 인스턴스를 요청할 때, Pool Service는 우선 해당 클래스의 pool에서 가능한 인스턴스가 존재하는지 체크한다. 만약 pool이 비어있다면, Factory Service로 forward하여 새로운 인스턴스가 생성된다.

Pool Service는 다음과 같은 장점을 제공한다.

- 초기화 작업과 관련된 비용을 감소시킨다.
- 객체의 재사용을 통해 메모리를 효율적으로 사용할 수 있다.
- 특정 객체의 수를 선택적으로 제한할 수 있다.

⑤ Cache Service

Cache Service는 상태를 공유할 수 있는 객체를 cache하는 기능을 제공한다.

비즈니스 어플리케이션에서 Cache Service를 사용하는 객체는 변경이 자주 일어나지 않지만 사용 빈도가 많고, 생성하는데 비용이 높은 객체이며, 이 경우에 Cache가 임시 저장 공간으로 사용된다.

이 Service를 사용하게 되면 다음과 같은 장점을 제공한다.

- 자주 접근하는 데이터를 매번 데이터베이스로부터 fetch할 필요가 없으므로 오버헤드가 줄어든다.
- 객체를 매번 생성하지 않기 때문에 메모리를 효율적으로 사용할 수 있다.
- Cache Service를 사용하지만 자주 사용되지 않고 메모리만 점유하고 있는 객체에 대해서는 자체적으로 삭제하는 기능을 제공한다.(각 객체마다 점유시간 설정가능)

⑥ Naming Service

프로퍼티 파일에 주어진 환경정보를 가지고 분산 환경의 원격서버에 존재하는 JNDI Tree에 접근하여 Naming Context의 시작점을 가져온다. 이 서비스는 다른 노드나 어플리케이션 서버에 자원이 할당되는 경우 코드의 수정없이 환경정보의 변경만으로도 쉽게 해당 Context에 접근할 수 있도록 하는기능을 제공한다.

⑦ JNDI Service

JNDI Service는 JNDI Server에 접근하여 객체를 가져오는 과정을 추상화한다. 객체가 원격 서버에 존재하는 경우, 클라이언트가 이 객체를 가져오기 위해 매번 JNDI Service를 사용해 JNDI Server에 접근해야 한다면 성능에 치명적인 손상을 줄 수 있다. JNDI Service에서는 한 번 찾은 객체를 Cache Service를 사용하여 Cache에 저장하고, 이 후 클라이언트 요청에 대해서는 Cache 안에 존재하는 객체를 사용하게 한다. JNDI Server의 접근은 Naming Service를 사용함으로써, 특정 WAS나 네트워크 노드에 종속되지 않는 분산환경의 비즈니스 어플리케이션 구현을 가능하게 한다.

이 서비스가 제공하는 주된 기능은 서버로부터 트랜잭션 객체, EJBHome 객체, Datasource 객체를 찾는 것이다. 이외의 JNDI Server에 존재하는 객체는 lookup() 메소드를 사용하여 찾을 수 있는 기능을 제공한다.

⑧ Upload Service

Upload Service는 파일을 업로드하는 기능을 제공한다.

Java기반의 Web Application에서 파일을 업로드하기 위해서는 RFC1867을 따르는multipart/form-data Content 타입의 Stream을 처리해야 한다.

Service Framework에서 제공하는 Upload Service에서는 HTTP Request를 Parsing하여 파라미터와 파일 내용을 ParameterParser 객체에 저장하고 어플리케이션은 이 객체를 통해 원하는 필드에 대한 값을 얻거나 파일을 이용할 수 있다. 또한 어플리케이션에서 파일의 크기에 따라 메모리에 잠시 저장하거나 디스크에 영구히 저장하는 방식 중 선택할 수 있고, FileItem 객체를 사용하여 파일에 대한 여러 정보를 얻을 수 있다.

⑨ DB Connection Service

DB Connection Pool Service는 Database Connection Pooling 기능을 제공한다.

DB Connection Pool Service는 WAS에서 같은 기능을 제공하므로, 구축하고자 하는 어플리케이션이 WAS를 사용한다면 해당 WAS에서 제공하는 DB Connection Pooling 기능을 사용하는 것이 더 낫다. 이

Service는 WAS를 사용하지 않는 비즈니스 어플리케이션에서 DB Connection Pooling 기능이 필요할 때 사용하게 된다. 또한, 프레임워크 내부적으로는 Message Service나 UniqueId Service에서 사용된다.

DB Connection Pool Service는 여러 개의 connection pool을 관리한다. 각 pool은 특정 데이터베이스와 연계되어 있으며, driver,name, url, username과 password 로 식별된다. 각 pool은 프로퍼트 파일에 정의되어 생성되거나, 런타임시에 registerPool()메소드를 사용하여 생성된다.

⑩ Localization Service

Localization Service는 다국어 지원을 위해 java.util.ResourceBundle을 Wrapping하여 Service Framework에 추가하였다. Localization Service는 클라이언트가 요청하는 Locale과 bundleName에 따라 그에 해당하는 ResourceBundle 객체를 제공한다. 클라이언트에 한 번 서비스된 ResourceBundle 객체는 Hashtable에 저장하고 이후 클라이언트의 요청에 대해서는 Hashtable에서 ResourceBundle 객체를 찾아 제공함으로써 성능을 높일 수 있도록 한다.

⑪ Unique Id Service

Unique Id Service는 시스템에서 필요한 ID를 생성하는 기능을 제공한다. Unique Id Service가 생성하는 ID는 사용 목적에 따라 크게 두 가지 타입으로 구분될 수 있다. 하나는 파일, 객체, 세션 등 임시적으로 시스템에서 생성하는 엔티티들의 이름으로 사용될 수 있는 ID이고 다른 하나는 DB의 key 값으로 사용될 수 있는 ID이다.

⑫ Message Service

Message Service에서 사용하는 message는 message ID와, message 내용, message 타입으로 구성된다. Message는 성능향상을 위해 Cache Service를 사용하여 cache에 저장되며, message 리스트는 데이터베이스나 파일에 저장되어 있다. 흔히 message는 어플리케이션에서 문제가 발생하거나 어플리케이션의 처리 상황을 사용자에게 보여주어야 할 때 사용된다.

2) Web Application Framework

어플리케이션의 View 모델을 처리하기 위한 WAF의 주요 기능은 다음

과 같다.

① Sign-on/Sign-out

대부분의 웹 어플리케이션 서버가 지원하는 Sign-on/Sign-out 모델은 특정 제품에 종속된 형태를 가진다. 이러한 문제는 제품 간에 어플리케이션의 이식성을 떨어지게 한다. WAF은 내부에 Sign-on/ Sign-out 모듈을 가지고 있으며 보호된 페이지의 최초 접근이나 로그인을 통해 Sign-on을 한다. Sign-on은 WAF의 가장 앞 단에 filter로 존재하며 모든 request에 대해 하나의 단일화된 모듈을 통해 통제하도록 한다. Sign-on은 사용자 정보를 기반으로 하고 있으며 이 사용자 정보는 DB나 혹은 XML Configuration에 존재한다.

② Role 기반의 자원 접근 통제

WAF이 Role 기반으로 통제하는 자원은 Page와 Action이다. Page는 사용자에게 보여지는 부분의 처리만을 전담하고, Action은 View단의 처리를 전담한다. 이 Page와 Action은 사용자가 가지고 있는 접근 권한(Role)에 따라 접근 허용 여부가 결정된다.

③ Page Template Service

Template 페이지에 화면 구성 요소가 배치되는 위치 정보를 정의하고 이 Template에 따라 페이지를 생성한다. 이 방법은 사용자에게 가장 민감한 영향을 주는 화면의 배치를 변경 요구 시 코드의 수정 없이 template만을 통해서 해결할 수 있는 장점을 제공한다.

④ Error Handling

WAF의 에러 처리는 Form으로부터 넘어온 request의 데이터가 올바른지 확인하는 기법과 Action 처리시 발생하는 에러를 처리하는 기법을 제공한다. 어플리케이션 처리과정에서 에러가 발생하면, Configuration에 지정된 Error Page나 Default Error Page로 forward한다.

⑤ Screen Workflow Control

request 처리 후에 forward되어야 하는 페이지는 업무 조건에 따라 여러 페이지로 분기될 수 있다. 또한 운영시점에 새로운 페이지로 forward될 수도 있다. 이런 Screen Workflow의 변경에 쉽게 대응하기 위해 XML 기반의 Workflow Control Configuration을 제공한다.

⑥ Localization

사용자에게 보여지는 페이지의 다국어 지원을 위해 locale별로 페이지를 구성하고 사용자가 요구하는 locale에 따라 시스템은 자동적으로 해당 locale에 대한 페이지를 읽는다.

⑦ Logging

WAF은 개념적으로 Filter, Action Handler, Screen Handler로 구성 되는데 각 처리과정에 대한 로그를 지원한다. 로그는Service Framework의 내부에 존재하는 Logging Service를 사용하며 로그 수준에 따라 info, warn, error 3가지를 남긴다. 이것은 WAF에서 문제가 발생하거나 향후 접속 정보에 대한 통계자료를 얻는데 핵심적인 역할을 한다.

⑧ View Model을 위한 MVC 아키텍처 제공

WAF은 비즈니스 어플리케이션의 View 부분에 대한 처리를 위해 완벽한 형태의 MVC 아키텍처를 제공한다. 이것은 View 모델 내에도 사용자에게 직접적으로 보이는 정보를 다루는 부분과 처리를 위해 필요한 데이터의 영역을 분리한다.

3) Utilities

비즈니스 어플리케이션 개발시 자주 반복되어 사용되는 기능을 Utility API로 구성하였다. Service Broker에 의해 통제를 받는 Service와는 달리 Utility API는 어플리케이션에서 그 API를 제공하는 클래스의 인스턴스를 직접 생성하여 사용하거나 클래스의 정적 메소드에 접근하여 원하는 기능을 제공 받는다.

① Configuration

이 클래스는 프로퍼티 파일에서 key, value 쌍을 읽어 메모리 가지고 있다. value는 이 클래스의 인스턴스를 사용하는 어플리케이션이 key를 가지고 요구하는 메소드에 따라 String, Vector, int, double 등과 값은 타입으로 변환되어 내부적으로 key에 해당하는 값을 유지한다. 한 번 변환된 값은 이후 요청에 대해서는 반드시 같은 타입의 값을 반환하는 메소드를 사용해야 한다.

② Ftp Client

이 클래스는 FTP Server에 접근하여 클라이언트와 서버 사이에서 파일을 송수신하기 위해 사용한다.

이 클래스는 다음과 같은 기능을 제공한다.

■ FTP Server의 현재 디렉토리에 존재하는 파일의 정보를 가져온다.

■ 클라이언트의 파일을 FTP Server로 보낸다.

■ FTP Server의 파일을 클라이언트로 가져온다.

■ FTP Server를 특정 파일을 지운다.

■ FTP Server의 디렉토리를 변경한다.

■ File Transfer Type을 설정한다(binary,ascii).

③ ConvertUtils

이 클래스는 String 객체를 Float, Double, Integer, Byte 와 같은 객체로 변환하는 기능을 제공한다.

④ StringUtil

이 클래스는 비즈니스 어플리케이션 개발 중 자주 사용되는 문자열 처리에 관련된 기능을 제공한다. 비교적 단순한 기능을 구현한 메소드들로 구성되어 있다.

⑤ Mail

Mail Utility는 어플리케이션에서 email을 보내거나 받을 수 있는기능을 제공한다.

Service Framwork 자체에서 Mail Server를 제공하지는 않으며, Mail Server에 대한 설정은 프로퍼티에서 정의하게 된다.

9. 개발단계별 기간 및 투입인원수

개발단계		개발일정	인원	비고
계획		2002.5~2002.6	4	구축계획 수립, 선진사례 분석, 개발환경 수립
1차	분석/설계	2002.6~2002.7	4	아키텍처 수립, 요구사항정의, 정적모델링, 동적모델링, 데이터베이스설계
	개발	2002.7~2002.8	4	코딩, 테스트
	구현	2002.8	6	사용자매뉴얼 작성, 적용사례 확보
2차	분석/설계	2002.8~2002.9	4	아키텍처 수립, 요구사항정의, 정적모델링, 동적모델링, 데이터베이스설계
	개발	2002.10	4	코딩, 테스트
	구현	2002.10	4	사용자매뉴얼 작성, 적용사례 확보

10. 사용 시스템과 개발언어

1) 사용 시스템

- 개발 장비

구분	사양	비고
사용모델	SENS 950	2001년 생산
CPU	Intel Pentium III 1000MHz	
RAM	512 MB	
OS	Microsoft Windows XP Professional Version 2002	

- 개발 서버

구분	사양	비고
CPU	Intel Pentium III 1700MHz X 4	PVCS, DBMS, App. Server설치
RAM	1024 MB	
OS	Microsoft Windows 2000 Server	

2) 개발환경

구분	제품명	제작사
모델링도구	Rational Rose Enterprise Edition 2001A.04.00	Rational
개발언어	JDK 1.3.1	
개발도구	eclipse 6.8	IBM
형상관리도구	PVCS 6.8	MERANT
데이터베이스	Oracle 8i	Oracle
어플리케이션 서버	Weblogic 6.1	BEA