

# 비즈니스 사례를 통한 웹 서비스 적용지침

대우정보시스템 백종현 · 김형석 · 강병철 · 김제욱

## 1. 서 론

웹 서비스 기술 표준이 성숙함에 따라, 실제 비즈니스 문제를 해결하기 위한 기술로서 웹 서비스를 채택하는 기업들이 늘어가고 있다. 현재 웹 서비스는 금융, 통신, 제조업체 등을 중심으로 기업 내부 통합, 기업 간 협업, 기업 포털 구축 등의 분야에서 광범위하게 적용되고 있으며, 새로운 비즈니스 모델 구축에도 적용되고 있다. 하지만 아직까지 웹 서비스 기술을 적용하기 위한 표준 개발 방법론 및 절차 등은 부재한 실정이며, 이는 기업들이 웹 서비스 기술을 적용하는데 위험요인이 되고 있다.

이에 본 고에서는 기업들이 웹 서비스를 이용하여 그들의 비즈니스 문제를 보다 효과적으로 해결할 수 있도록 웹 서비스 적용 지침을 제시한다. 이를 위해 가상의 B2B 시나리오를 제시하며, 시나리오를 이용하여 시스템 개발 단계 별로 웹 서비스 적용 절차 및 지침을 소개한다. 또한 웹 서비스를 적용하는데 따른 다양한 기술적 이슈들을 살펴보고, 이에 대한 해결책을 제시함으로써 기업이 보다 쉽게 웹 서비스를 도입할 수 있도록 한다.

본문의 구성은 다음과 같다. 2장에서는 국내외 기업들의 웹 서비스 적용 사례를 소개한다. 3장에서는 웹 서비스 적용 시스템을 위한 비즈니스 시나리오를 알아보고, 4장에서는 이 시나리오를 이용하여 웹 서비스 적용 절차 및 지침을 제시한다. 5장에서는 웹 서비스 기술 적용 시 제기된 이슈를 살펴보고, 이에 대한 해결책을 제시한다. 마지막으로 6장에서는 본 논문을 요약하고 결론을 내린다.

## 2. 관련 연구

이 장에서는 국내외 기업들이 웹 서비스를 적용하여 기업 간 통합 및 협업 시스템을 구축한 실제 사례를 살펴본다.

### 2.1 델(Dell)

델은 다이렉트 모델(Direct Model)로 기존의 경쟁사

들과 차별화된 비즈니스를 기반으로 업계 최고의 자리에 올랐다. 다이렉트 모델은 유통체계를 공급자, 델, 사용자(최종 고객)로 구성하여 비즈니스의 민첩성과 유연성을 확보할 수 있는 것으로 기존 모델과의 차이점은 그림 1과 같다.

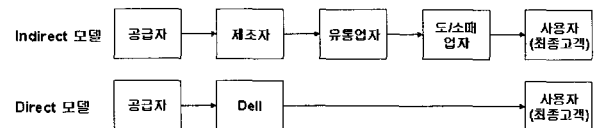


그림 1 델의 다이렉트 모델

델은 다이렉트 모델을 구현하기 위해 좀더 유연한 IT 아키텍처가 필요했으며, 이를 위해 웹 서비스를 적용하였다. 델은 우선 견고한 협력관계에 있는 협력사를 대상으로 웹 서비스를 적용하였으며, 구체적인 적용 분야는 다음과 같다.

- 부품재고 및 가격정보 제공 - 실시간으로 거래 공급자들의 부품재고 상황과 가격 정보를 확보하여 고객 및 3자 파트너에게 제공하기 위한 수단으로 웹 서비스를 활용하였다. 예를 들어 델은 가격정보를 제공하는 웹 서비스를 구축하여 고객이 Dell.com 이외에 Yahoo, C Net, ZDNet 등에서도 동일한 서비스를 이용할 수 있게 하였다.
- 세액과 배송비용 계산 - 국가 별로 다르게 적용되는 세액과 배송비용을 계산하기 위한 수단으로 델은 웹 서비스를 활용하였다. 델은 세금 및 배송비 웹 서비스를 구축하여 제3자 물류업체에게 국가별 세금정보와 배송비 관련 정보를 전달하였다. 이를 통해 델이 서비스하는 어느 국가에서도 정확하고 일관된 정보를 얻을 수 있게 하였다.
- 주문상황 및 배송상태 추적 - 델은 실시간으로 주문 상황 및 배송 상태를 모니터링하기 위한 수단으로 웹 서비스를 활용하였다. 델은 주문 추적 웹 서비스를 구축하여 제3자 물류업체로부터 실시간 주문 및 배송정보를 받아 고객이 Dell.com 또는 전자우편

을 통해 주문 추적 정보를 수시로 확인할 수 있게 하였다.

## 2.2 GM

GM은 세계 최대 자동차 제조업체로 여러 국가에 걸쳐 연구개발 및 생산 시설을 두고 있다. 일반적으로 제조업에서는 제품의 성능 개선 등의 이유로 빈번한 설계변경이 발생하는데, 이러한 변경 사항은 관련된 내부 부서 및 공급사 등에 즉각 반영되어야 한다. 특히 자동차는 인간의 안전과 직접 관련된 제품이므로 미미한 변경일지라도 관련된 모든 곳에 즉각 반영되어야 한다. GM은 기존에 이러한 변화를 반영하기 위해서 전자우편, 전화, 팩스 또는 직접 대면 등의 방법을 사용하였다. 하지만 이러한 방법은 전달과정의 오류로 인한 손실 및 비용문제 등을 야기하였다. GM은 이러한 문제를 해결하고 유연한 생산 협업체계를 구축하기 위해서 웹 서비스를 적용하였다.

GM은 EDS의 PLM(Product Lifecycle Management) 솔루션인 E-VIS 4.0 솔루션(.NET과 XML기반에서 개발됨)과 GM의 네트워크(VPN과 사설 ATM망)를 이용해서 설계변경 사항에 관한 CAD/CAM 데이터를 실시간으로 관련된 부서 및 공급사에 제공하였으며, 데이터 공유를 위해 SOAP, WSDL, UDDI를 적용하였다. 이를 통해서 GM은 유연한 생산 협업체계를 구축하였으며 신제품 개발 기간 단축, 재고 비용 감소, 기업간 커뮤니케이션 강화 등의 효과를 거두었다.

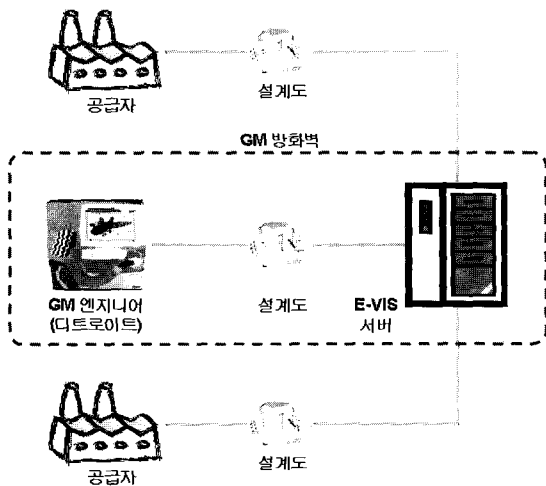


그림 2 GM의 웹 서비스 적용 아키텍처

## 2.3 삼성전자

삼성전자는 24개의 생산 법인과 36개의 판매 법인, 20개의 지점을 세계 곳곳에 갖추고 있는 글로벌 기업이다. 삼성전자는 글로벌 경영의 효율성 제고를 위해 현재 7개의 해외 지역별 총괄 체제를 운영하며 지역별로 차별화된 연구개발, 마케팅 및 서비스를 시행하고 있다. 삼

성전자는 2001년에 전 해외 법인을 통합 운영하기 위한 ERP를 도입했고 2002년에는 해외 법인과 본사 간의 거래 자동화를 위한 WTN(World Trade Network)을 구축해서 업무 협력을 위한 인프라를 다져 놓았지만 정보의 공유에 한계가 있어 협력사 간의 진정한 협업을 구현하지 못하였다.

이에 삼성전자는 협업 지향적인 글로벌 조직을 갖추기 위해 해외 법인은 물론 해외 거래선까지 협업 프로세스를 정착시켜 대규모 글로벌 무역 프로세스 기반을 다지기 위해 GSBN(Global Samsung Business Network) 구축에 착수했다. GSBN 프로젝트는 기존의 IT 시스템 인프라에 별도 수정 없이 구현하는 것을 목표로 했고, 이를 위해 EAI나 B2Bi 등 패키지 방식이 아닌 웹 서비스를 적용하였다. 삼성전자는 GSBN의 구축으로 원가 절감, 업무 효율성 증대와 같은 가시적 효과를 거두었으며, 협력사와의 협업체계 정립과 거래선 및 지법인의 만족도 향상을 얻을 수 있었다.

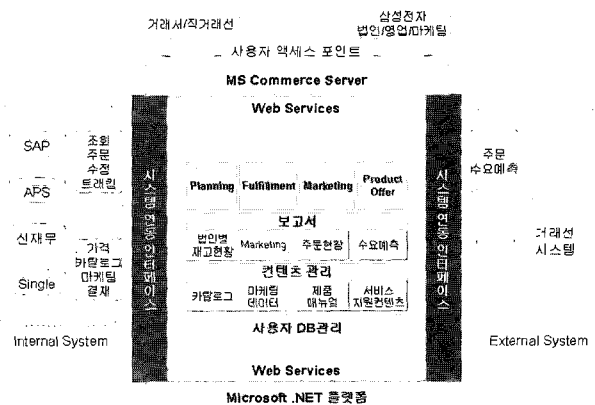


그림 3 삼성전자 GSBN 시스템 아키텍처

## 3. 웹 서비스 적용 비즈니스 시나리오

### 3.1 개요

이 장에서는 웹 서비스 적용 대상이 되는 시스템의 비즈니스 시나리오를 제시한다. 여기서 제시하는 시나리오를 이용하여 이어지는 장에서는 웹 서비스 기술을 적용하기 위한 절차 및 지침 등을 소개한다.

웹 서비스 적용을 위한 비즈니스 시나리오로 여기서는 협업 기반의 여행사 포털 시스템을 선정하였다. 그림 4는 여행 포털 시스템의 개략적인 구성도를 보여준다. 그림에서 보듯이 사용자는 포털을 통해 여행 예약 등의 기능을 사용하며, 포털 시스템은 항공사, 보험사, 은행 등과 협업을 수행한다. 4장에서 자세히 살펴보겠지만, 포털 시스템과 관련 기업 간의 협업을 위해서 SOAP 메시지를 이용한 통신기술, WS-Security를 이용한 보안

기술, BPELWS를 이용한 비즈니스 프로세스 실행 기술 등을 적용한다. 다음 절에서는 유즈케이스를 통해 자세한 비즈니스 시나리오를 소개한다.

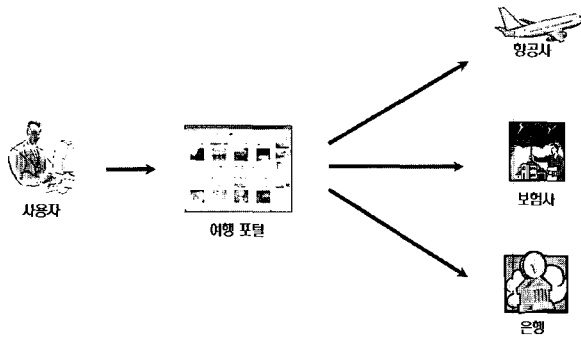


그림 4 여행 포털 시스템의 구성도

### 3.2 유즈케이스 시나리오

유즈케이스 시나리오를 통해 여행 포털 시스템의 상세한 비즈니스 시나리오를 살펴보자. 그림 5는 여행 포털 시스템의 유즈케이스 다이어그램이다. 그림에서 보듯이 여행 포털 시스템은 사용자, 여행 포털, 항공사, 보험사, 은행 등의 액터(actor)들로 구성된다. 각 액터가 가진 유즈케이스들은 서로 연계되는데, 여행 포털은 주로 사용자 인터페이스(UI)와 관련된 기능을 처리하며, 항공사, 보험사, 은행 등의 시스템은 실제 비즈니스 로직

를 포함한다. 표 1은 각 액터들을 설명한 것이다.

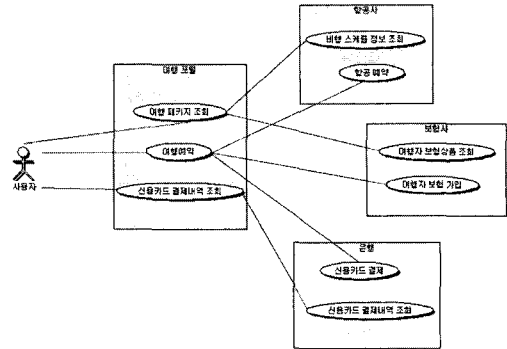


그림 5 여행 포털 시스템의 유즈케이스 다이어그램

표 1 액터 설명

액터	설명
사용자	여행 포털에 접속하여 여행 정보를 검색하고 항공을 예약하며 신용카드 정보를 조회하는 고객
여행 포털	사용자를 위해 여행예약, 신용카드 정보조회 등의 기능을 UI를 통해 제공하는 시스템. 업무의 구현을 위해 은행, 항공사, 보험사 등과 연동함
항공사	항공 노선, 스케줄, 예약 기능 등을 제공하는 시스템
보험사	특정 여행과 관련된 보험의 가입, 탈퇴 등의 기능을 제공하는 시스템
은행	신용카드 정보조회, 신용카드 결제 등의 기능을 제공하는 시스템

표 2 여행 포털의 유즈케이스 시나리오

액터	유즈케이스	관련 액터	기능적 요구사항	비기능적 요구사항
여행 포털	여행패키지 조회	사용자, 항공사, 보험사	패키지 상품명, 출발지, 도착지 등의 조건으로 여행 패키지 조회 (해당 여행에 대한 보험상품 조회 포함)	없음
	여행 예약	사용자, 항공사, 보험사, 은행	조회한 여행 패키지에 대해 예약자 정보, 동반자 정보 등을 입력하여 여행 예약 (항공사와 연계한 항공권 예약, 보험사와 연계한 보험가입, 은행과 연계한 신용카드 결제 기능 포함)	<ul style="list-style-type: none"> <li>은행과의 연동 시 인증, 메시지 암호화 및 무결성, 부인방지</li> <li>비즈니스 프로세스 자동화 기능 적용</li> <li>트랜잭션 보장</li> </ul>
	신용카드 결제내역 조회	사용자, 은행	사용자가 신용카드로 결제한 내역을 은행으로부터 조회	<ul style="list-style-type: none"> <li>인증, 메시지 암호화 및 무결성</li> </ul>
항공사	비행 스케줄 정보 조회	여행 포털	포털로부터 패키지 상품명, 출발지, 도착지 등의 조건을 받아 해당 여행 패키지 리스트를 반환	없음
	항공 예약	여행 포털	포털로부터 항공 번호, 예약자 및 동반자 정보를 받아 항공권을 예약	없음
보험사	여행자 보험 상품 조회	여행 포털	포털에게 보험상품 번호, 상품명, 보험료 등의 여행자 보험상품 정보 제공	없음
	여행자 보험 가입	여행 포털	포털로부터 보험상품 번호, 가입자 정보 등의 정보를 받아 보험 가입	<ul style="list-style-type: none"> <li>비즈니스 프로세스 자동화 기능 적용</li> <li>트랜잭션 보장</li> </ul>
은행	신용카드 결제	여행 포털	포털로부터 카드번호, 비밀번호, 유효기간, 결제금액 등의 정보를 받아 신용카드 결제	<ul style="list-style-type: none"> <li>인증, 메시지 암호화, 메시지 무결성 및 부인방지</li> <li>비즈니스 프로세스 자동화 기능 적용</li> <li>트랜잭션 보장</li> </ul>
	신용카드 결제내역 조회	여행 포털	포털로부터 카드번호, 조회기간 등의 정보를 받아 해당 카드 결제 내역을 반환	<ul style="list-style-type: none"> <li>인증, 메시지 암호화, 메시지 무결성</li> </ul>

각 액터 별 유즈케이스 시나리오를 통해 시스템의 기능적 요구사항 및 비기능적 요구사항을 살펴보자. 표 2 는 여행 포털, 항공사, 보험사 및 은행이 제공하는 유즈 케이스 시나리오이다.

#### 4. 웹 서비스 적용 지침

본 장에서는 3장에서 소개한 여행 포털의 비즈니스 시나리오를 이용하여 여행 포털 시스템 구축 시 웹 서비스 기술 적용을 위한 절차 및 지침을 제시한다. 여기서는 웹 서비스 기술 적용에 관한 절차를 웹 서비스 식별 및 명세, 웹 서비스 아키텍처 설계, 웹 서비스 개발, 웹 서비스 테스트 등의 단계로 분류하였다.

##### 4.1 웹 서비스 식별 및 명세

이 단계에서는 구축하고자 하는 웹 서비스 적용 시스템의 요구사항을 분석하여 시스템을 구성하는 웹 서비스들을 식별하고 이에 대한 명세서를 작성한다. 웹 서비스 식별 및 명세 단계의 목적은 다음과 같다.

- 요구사항, 유즈케이스 모델을 기반으로 웹 서비스 후보를 식별한다.
- 식별된 웹 서비스를 정제하고 역할 및 기능을 명확히 하기 위해 인터페이스를 정의한다.
- 웹 서비스 간의 상호작용을 분석하여 인터페이스를 정제하고, 식별된 컴포넌트의 상세 오퍼레이션을 도출하여 웹 서비스 명세서를 작성한다.

##### 4.1.1 웹 서비스 식별

웹 서비스 식별 활동에서는 기존에 분석한 요구사항의 내용을 바탕으로 웹 서비스 후보를 식별하고, 상위 수준의 컴포넌트 아키텍처를 도출한다. 상세한 활동 절차는 다음과 같다.

- ① 웹 서비스 도출 기준 결정 - 개발하고자 하는 시스템의 특성 및 기 추출된 웹 서비스 요구사항에 따라 웹 서비스 도출 기준을 결정한다.
- ② 웹 서비스 식별 - 유즈케이스 모델을 이용하여 액터 간 데이터 흐름을 분석하고, 프로세스 통합이나 서비스 통합의 경우 웹 서비스 후보로 식별한다.
- ③ 웹 서비스 컴포넌트 아키텍처 구성 - 도출된 웹 서비스 후보를 중심으로 웹 서비스 간의 의존성을 파악하여 웹 서비스 컴포넌트 아키텍처를 작성한다.

여행 포털 시스템에서의 웹 서비스 도출 기준은 다음과 같다.

- 비즈니스 시나리오로부터 추출된 유즈케이스를 기반으로 하여 시스템들(여행 포털, 항공사, 보험사,

은행 등) 간의 의존관계를 파악한다.

- 타 시스템에게 제공하는 기능을 위주로 웹 서비스 후보를 도출한다.
- 유즈케이스에 제시된 업무 플로우를 웹 서비스 관점에서 상세히 분석하여 추가적으로 필요한 웹 서비스를 도출한다.

위 기준에 따라 표 3과 같이 웹 서비스를 식별하였다.

표 3 여행 포털 시나리오로부터 식별된 웹 서비스 목록

분류	사이트	서비스 명	설명
서비스 소비자	포털	여행 예약	여행 예약을 위한 인터페이스 제공
		신용카드	신용카드 업무를 위한 컴포넌트
		여행 패키지	여행 상품 조회를 위한 컴포넌트
서비스 제공자	은행	신용카드 결제	항공 예약을 위한 신용 카드 결제 서비스
		신용카드 사용	신용카드의 사용 내역 조회 서비스
	항공사	항공기 예약	항공기 예약 서비스
		비행스케줄 조회	비행 스케줄 조회 서비스
	보험사	여행자 보험가입	보험 가입 처리 서비스
		여행자 보험상품 조회	보험 상품 조회 서비스

다음으로는 식별한 웹 서비스들 간의 의존 관계를 파악하여 웹 서비스 컴포넌트 아키텍처를 구성한다. 그림 6은 여행 포털 시스템의 웹 서비스 컴포넌트 아키텍처이다.

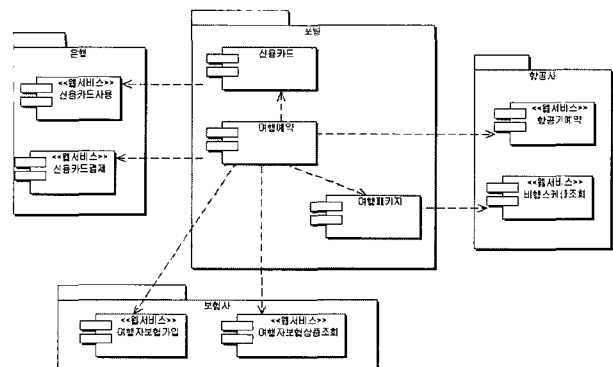


그림 6 여행 포털 시스템의 웹 서비스 컴포넌트 아키텍처

##### 4.1.2 웹 서비스 명세

웹 서비스가 추출되면 각 웹 서비스의 인터페이스, 즉 웹 서비스가 제공하는 오퍼레이션 및 이에 대한 시그니처(Signature) 등을 정의해야 한다. 또한 웹 서비스들 간의 상호작용을 분석하여 이들간의 의존관계 및 개발 시 필요한 선후 관계를 파악하고, 필요에 따라 기 추출된 웹 서비스를 수정 및 삭제 하거나 새로운 웹 서비스 후보를 추출한다. 그 후 최종적으로 추출된 웹 서비스를 대상으로 명세서를 작성한다. 웹 서비스 명세서에는 다

음의 내용이 포함된다.

- 웹 서비스 개요-서비스 명, 간단한 설명
- 개발/조달 여부-서비스를 직접 개발하는지 또는 외부에서 조달하는지 여부
- 기능 설명-서비스가 제공하는 각 오퍼레이션을 정의
- 비기능 요구사항-웹 서비스 보안, 트랜잭션, 신뢰성 등의 비기능 요구사항 명시
- 기술 표준-해당 서비스에 적용될 기술 표준을 명시

표 4는 항공 예약 서비스에 대한 웹 서비스 명세서이다.

표 4 항공 예약 서비스 명세서

서비스명	AirlineReservation		
서비스 설명	항공기 예약을 위한 서비스	개발/조달	직접 개발
기능	getAirlineSchedule	설명	항공 스케줄을 조회
		파라미터	int airlineID
		반환값	AirlineSchedule
	reservation	설명	항공 예약
		패라미터	int airlineID int scheduleID String userID
반환값		boolean	
비기능	웹 서비스 보안	요구사항	메시지 수준의 보안
		기술표준	WS-Security
	웹 서비스 트랜잭션 보장	요구사항	Long Term 트랜잭션 보장
		기술 표준	WS-Coordination, WS-Transaction

## 4.2 웹 서비스 아키텍처 설계

웹 서비스 아키텍처 설계 단계에서는 시스템의 비기능 요구사항을 만족시키기 위해 서비스가 갖추어야 할 구조와 필요한 기술을 식별한다. 이 단계의 목적은 다음과 같다.

- 비즈니스 기능을 만족하도록 서비스들 간의 의존관계를 표현하는 웹 서비스 아키텍처를 설계한다.
- 시스템의 비기능 요구사항을 만족하도록 웹 서비스가 채택해야 할 기술 표준 아키텍처를 정의한다.
- 적용할 웹 서비스 기술을 실제 시스템 상에 구현하기 위한 개발도구 및 서버 등의 솔루션을 식별하기 위해 웹 서비스 소프트웨어 아키텍처를 정의한다.

### 4.2.1 웹 서비스 컴포넌트 아키텍처

여기서는 웹 서비스 식별 시 작성한 웹 서비스 컴포넌트 아키텍처를 웹 서비스 설계 및 구현 패턴 등을 이용하여 정제한다. 그림 7은 여행 포털 시스템의 웹 서비스 컴포넌트 아키텍처를 보여준다. 그림에서 JCF는 J2EE 개발 프레임워크를 의미하며, BPEL은 BPEL4WS로 정의한 여행 예약 비즈니스 프로세스를 말한다.

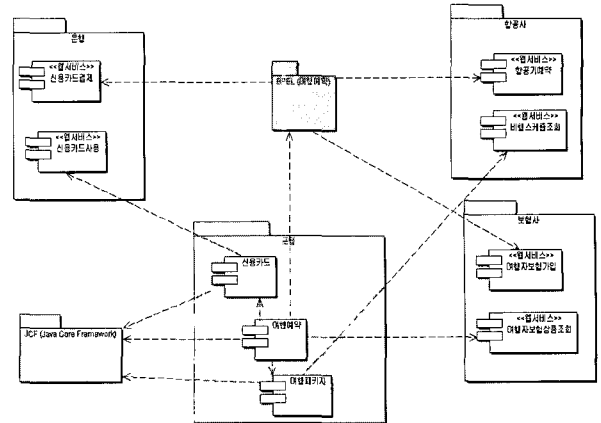


그림 7 정제된 웹 서비스 컴포넌트 아키텍처

### 4.2.2 웹 서비스 기술 아키텍처

웹 서비스 기술 아키텍처 단계에서는 식별한 웹 서비스를 구현하기 위해 필요한 웹 서비스 기술 스펙을 선정하고, 이들 간의 관계를 적절히 설정한다. 이를 위해 웹 서비스 서버와 클라이언트 간의 기술적 요구사항을 파악하고, 이를 구현하는 기술 스펙을 추출한다. 표 5는 그림 7에 제시된 웹 서비스 간의 관계에 기초하여 필요한 웹 서비스 기술을 식별한 결과이다.

표 5 웹 서비스 기술 스펙 식별 결과

서비스 클라이언트	웹 서비스	기술적 요구사항	기술 스펙
포털의 여행예약	BPEL(여행 예약)	서비스 기술 (descriptoin), 메시징	SOAP, WSDL
	보험사의 여행자 보험상품 조회		
포털의 여행패키지	항공사의 비행스케줄 조회		
포털의 신용카드	은행의 신용카드 사용	서비스 기술, 메시징, 보안(암호화, 부인방지, 무결성)	SOAP, WSDL, WS-Security
BPEL(여행 예약)	항공사의 항공기 예약	서비스 기술, 메시징, BPM(비즈니스 프로세스 자동화 및 분산 트랜잭션)	SOAP, WSDL, BPEL4WS, WS-Transaction, WS-Coordination
	보험사의 여행자 보험 가입		
	은행의 신용카드 결제		

### 4.2.3 웹 서비스 소프트웨어 아키텍처

웹 서비스 소프트웨어 아키텍처 단계에서는 앞 절에서 구성한 기술 아키텍처의 기술 요건을 구현할 소프트웨어들을 식별하여 전체 시스템 아키텍처를 구성한다. 이를 위해서는 웹 서비스 관련 개발도구 및 서버들을 기능 별로 구분하는 카테고리가 필요하다. 표 6은 웹 서비스 소프트웨어 카테고리를 제시하며, 각 카테고리 별로 여행 포털 시스템이 채택한 개발도구 및 서버 목록을 보여준다.

표 6 웹 서비스 소프트웨어 카테고리

SW 구분	기능	구현 기술	도입한 SW
웹 서비스 개발 도구	Java, .NET 등으로 구현된 객체를 웹 서비스로 구현 및 웹 서비스 프로세스 구현	SOAP, WSDL	eclipse 1.2.1 (with Web Services Plug in), GLUE
SOAP 서버	웹 서비스 배치 및 실행	SOAP	AXIS 1.2 (with TOMCAT)
웹 서비스 보안 실행엔진	서비스 메시지 암호화, 무결성, 인증 및 부인방지	WS-Security	IBM WSDK 5.1
웹 서비스 비즈니스 프로세스 서버	웹 서비스 비즈니스 프로세스 실행 환경 제공	BPELAWS, SOAP	Collaxa 2.0
웹 서비스 비즈니스 프로세스 편집기	웹 서비스 비즈니스 프로세스 실행 언어 편집	BPELAWS	Collaxa JBPEL
웹 서비스 트랜잭션 서버	웹 서비스의 분산 트랜잭션 보장	WS-Transaction, WS-Coordination	Collaxa 2.0

그림 8은 식별한 웹 서비스 소프트웨어를 중심으로 표현한 여행 포털 시스템의 아키텍처이다.

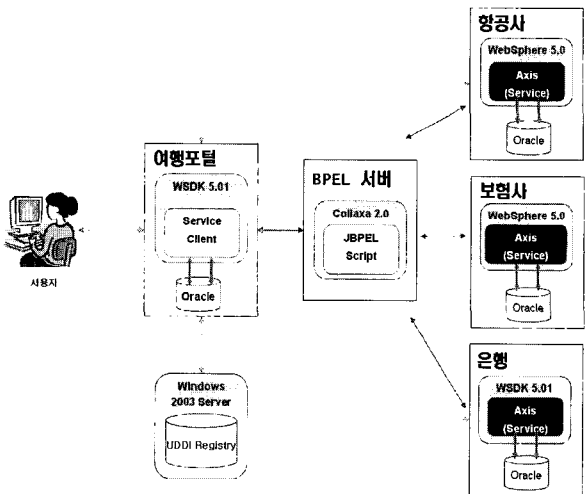


그림 8 웹 서비스 소프트웨어 중심의 여행 포털 시스템 아키텍처

### 4.3 웹 서비스 개발

이 단계에서는 이전 단계에서 도출한 웹 서비스 명세 및 웹 서비스 아키텍처를 이용하여 실제 웹 서비스를 개발하고 물리적인 환경을 구축한다. 웹 서비스 개발 시 고려 사항은 다음과 같다.

- 각 업무 단위로 웹 서비스 적용 범위 및 방법을 고려한다. 이 과정에서 서비스를 구축하기 위한 추가 기술 요소 및 위험 요인을 식별한다.
- 웹 서비스 기술을 구현하기 위한 개발 도구 및 서버의 적용 절차를 고려한다.
- 서비스 제공자 입장에서의 작업 단계 및 서비스 소

비자 입장에서의 작업 단계를 각각 고려한다.

웹 서비스 제공자와 사용자 별 개발절차는 다음과 같다.

- 웹 서비스 제공자 입장
  - ① 식별한 웹 서비스 별로 필요한 기술 요소를 나열하고, 기술 적용을 위한 개발 도구 및 서버를 선정한다.
  - ② 웹 서비스로 노출할 컴포넌트의 비즈니스 로직을 개발하고, 이에 대한 웹 서비스 인터페이스를 개발한다.
  - ③ 웹 서비스 배치(deploy) 도구를 이용하여 웹 서비스 서버에 배치한다.
  - ④ 웹 서비스 테스트 클라이언트를 구축하여 서버에 배치한 웹 서비스를 테스트한다.
  - ⑤ 테스트에 성공한 경우 웹 서비스 인터페이스인 WSDL 문서를 배포한다.
- 웹 서비스 사용자 입장
  - ① 이용할 웹 서비스에 대한 정보를 서비스 제공자로부터 받는다.
  - ② 웹 서비스 프록시를 생성하기 위한 개발 도구를 선정한다.
  - ③ 웹 서비스 제공자 또는 UDDI로부터 이용하려는 웹 서비스의 WSDL 파일을 제공 받는다.
  - ④ 웹 서비스 개발 도구를 이용하여 웹 서비스 WSDL 파일로부터 웹 서비스 프록시를 생성한다.
  - ⑤ 웹 서비스 프록시를 이용하여 웹 서비스 테스트 클라이언트를 구축한 후, 이를 이용하여 웹 서비스를 테스트한다.
  - ⑥ 실제 사용자 측 비즈니스 로직과 웹 서비스 프록시를 통합한다.

### 4.4 웹 서비스 테스트

이 단계에서는 개발한 웹 서비스가 기능 및 비기능 요구사항을 만족하는지 테스트하나, 웹 서비스 테스트 시 작업 지침은 다음과 같다.

- 각 웹 서비스 별로 단위 테스트를 실시한다.
- 업무 단위로 웹 서비스 서비스 통합 테스트를 실시한다. 이때 비기능 요구사항에 대한 만족도를 점검한다.

여행 포털 시스템의 경우, 다음과 같은 절차를 통해 웹 서비스를 테스트한다.

- 항공사, 보험사 및 은행 내의 웹 서비스 별로 기능 위주의 단위테스트를 실시한다.
- 유즈케이스 단위로 웹 서비스 간의 호출을 통해 서

비스가 정확한 업무를 수행하는 지 확인한다.

- 웹 서비스 요청 및 응답 시 교환되는 SOAP 메시지를 모니터링 하여 제대로 동작하는 지 확인한다.
- 웹 서비스 호출 시 비기능 요구사항이 있을 경우 이에 대해 테스트를 수행한다. 여기서는 WS-Security와 BPEL에 대하여 테스트를 수행한다.

## 5. 이슈 및 해결방안

이 장에서는 웹 서비스 기술 적용 시에 겪었던 다양한 이슈들을 제시하고, 이에 대한 해결방안을 제시한다. 이슈를 기술과 개발도구로 분류하여 소개한다.

### 5.1 기술 관련 이슈

#### 5.1.1 표준 적용 시 상호운용성 문제

웹 서비스의 중요한 목표 중 하나로 어플리케이션들 간의 상호운용성 확보를 꼽을 수 있다. 하지만 웹 서비스 기술 표준들의 버전이 다양해짐에 따라, 이를 구현한 개발도구들 간의 상호운용성 문제가 웹 서비스 기술을 적용하는데 매우 큰 걸림돌로 작용하고 있다. 복잡한 표준들의 탄생으로 오히려 웹 서비스 기술이 어플리케이션들 간의 상호운용성 획득을 어렵게 만들고 있는 것이다.

이러한 웹 서비스 표준들 간의 상호운용성 문제를 해결하기 위해 탄생한 단체가 WS-I이다. 이 단체는 여러 웹 서비스 관련 표준 단체들이 제공하는 스펙들을 통합적으로 관리하여, 웹 서비스 기술 도입 시에 상호운용성을 보장하는 방안을 제시한다. 이를 위해 WS-I는 XML 1.0, XML Schema, HTTP 1.1, SOAP 1.1, WSDL 1.1, UDDI 2.0, 등으로 구성된 WS-I Basic Profile 1.0을 개발하였다. Basic Profile을 준수하는 웹 서비스 개발도구 또는 서버들 간에는 웹 서비스를 이용한 통신 시에 상호운용성이 보장된다.

여행 포털 시스템의 주요 개발도구인 WSDK, AXIS 1.2 등은 모두 Basic Profile 1.0을 준수하므로 웹 서비스를 이용한 통신의 상호운용성을 보장할 수 있었다. 특히 웹 서비스 서버와 클라이언트 간의 플랫폼이 상이할 경우, 반드시 WS-I의 Profile을 준수하는 개발도구 및 서버의 사용을 권고한다.

#### 5.1.2 SOAP 메시지 스타일

SOAP 메시지에는 RPC와 Document 방식 등 두 가지 종류의 메시지 스타일이 존재하는데, 이들은 WSDL 문서 내에 기술되어 웹 서비스의 호출 방식을 규정한다. RPC 스타일은 SOAP 요청 및 응답 시 <soap:body> 내에 호출할 오퍼레이션의 이름과 이것이 요구하는 파라미터들을 명시한다. 따라서 주로 이는 RPC 프로그래밍

모델을 구현할 때 자주 사용된다. 반면 Document 스타일은 <soap:body> 내에 SOAP이 규정하는 요소를 포함하고 있지 않으며, 웹 서비스 사용자와 제공자 간에 동의하는 스키마 형식의 문서를 포함하고 있다. 이는 주로 메시징 방식의 프로그래밍 모델을 구현할 때 주로 사용된다.

여행 포털 시스템의 경우 모든 웹 서비스 호출이 RPC 방식으로 이루어졌으므로, RPC 방식의 SOAP 메시지 스타일을 선택하였다.

#### 5.1.3 서비스 바인딩 방식

웹 서비스의 실제 주소가 바인딩 되는 시점에 따라 두 가지 바인딩 방식이 존재한다. 설계 시점에 주소가 정해지는 경우를 정적(static) 바인딩이라고 하며, 실행 시점에 정해지는 경우를 동적(dynamic) 바인딩이라고 한다. 정적 바인딩에서는 설계 시점에 호출할 웹 서비스의 주소가 클라이언트의 소스 코드에 반영된다. 실제 서비스의 유형과 물리적인 주소가 모두 설계 시점에 정해지는 것이다. 반면 동적 바인딩에서는 설계 시점에서 서비스의 유형만 정해지며 실제 주소는 정해지지 않는다. 이 경우 UDDI가 제공하는 API 또는 클라이언트 측 비즈니스 로직을 이용하여 실제 호출할 서비스의 주소를 실행 시간에 결정한다.

표 7 웹 서비스 바인딩 유형

	정적 바인딩	동적 바인딩
서비스 유형 결정 시점	설계 시점	설계 시점
서비스 주소 결정 시점	설계 시점	실행 시점

여행 포털 시스템의 경우, 여행 포털과 항공사, 보험사 및 은행 간의 웹 서비스 통신을 정적 바인딩 방식을 통해 웹 서비스를 구현하였다. 협업 파트너를 실행 시간에 판단해야 할 필요가 없었기 때문이다.

## 5.2 개발도구 관련 이슈

### 5.2.1 WSDL 개발도구

일반적으로 웹 서비스 클라이언트가 서버를 호출할 때, 서비스의 주소는 전송 레이어(예를 들어 HTTP)에 실려서 전송된다. WS-Addressing은 이러한 단점을 극복하기 위해 개발되었다. 이는 SOAP 메시지 자체에 서비스 주소에 관한 정보를 반영함으로써, 서비스 주소와 전송 레이어와의 의존관계를 없애준다.

4장에서 밝혔듯이 여행 포털 시스템의 비즈니스 프로세스 실행 엔진으로 Collaxa 서버를 사용하였다. 그런데 Collaxa에 배치된 웹 서비스는 WS-Addressing 메커니즘에 의해 호출된다. 따라서 Collaxa에 배치된 비즈니스 프로세스 실행 웹 서비스를 호출하기 위해서는

웹 서비스 클라이언트가 WS-Addressing 방식의 주소 매커니즘을 제공해야 한다. 이러한 이유로 여행 포털 측의 웹 서비스 클라이언트는 The Mind Electric이 제공하는 GLUE 개발도구를 이용하여 구축하였다. 이 도구는 WS-Addressing을 이용한 서비스 호출 방식을 지원하기 때문이다.

### 5.2.2 보안 관련 도구

여행 포털 시스템은 암호화, 무결성, 부인방지, 인증 등의 보안 관련 요구사항을 만족하기 위해 WS-Security 기술을 채택하였고, 이를 구현하는 서버로 WSDK 5.1을 사용하였다. 그런데 WSDK는 런타임 시에 서버 내부의 API를 통해서만 WS-Security 관련 프로세싱을 수행한다. WSDK는 별도의 보안 관련 플러그인(plug-in)이나 API를 제공하지 않기 때문에 웹 서비스 클라이언트 측도 WSDK 서버에 배치해야 하는 불편함이 있다. 일반적으로 이러한 시나리오는 부적절하다. 클라이언트는 서버와 다른 플랫폼 상에서 동작하는 경우가 대부분이며, 따라서 웹 서비스 보안을 위해 따로 플러그인 형식의 플랫폼 독립적인 아키텍처를 구성하는 것이 타당하기 때문이다. 향후 확장성을 고려해봐도 이는 자명하다.

여행 포털 시스템에서는 웹 서비스 클라이언트, 즉 여행 포털 측도 WSDK에 배치함으로써 웹 서비스 보안 관련 기능을 구현하였다. 하지만 앞에서 언급한 플러그인 또는 별도의 API 형식으로 웹 서비스 클라이언트 측 보안 기능을 구현할 것을 권고한다. 아키텍처의 유연성 및 확장성 측면에서 타당하기 때문이다. 예를 들어 VeriSign의 Trust Gateway, Betrusted의 Unicert XKMS Server 등은 그림 9와 같은 플러그인 방식의 웹 서비스 보안 아키텍처를 제공한다.

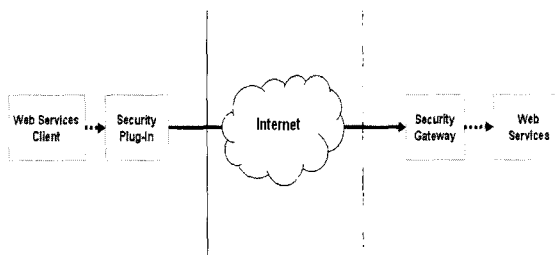


그림 9 플러그인 방식의 웹 서비스 보안 아키텍처

## 6. 결 론

지금까지 기업 간 협업 시에 웹 서비스 기술을 도입하기 위한 절차 및 지침을 여행 포털 비즈니스 시나리오를

통해 알아보았다. 웹 서비스 기술을 적용하려는 기업은 본문에 제시된 웹 서비스 적용 절차 및 지침을 참고하여 기술 도입 시 위험을 줄일 수 있으리라 본다.

웹 서비스는 본문에서 제시한 기업 간 협업 외에도 향후 전사 아키텍처 구축, 기업 포털 구축, 비즈니스 프로세스 관리 등의 분야에 폭넓게 적용될 것으로 예상된다. 따라서 기업은 웹 서비스 기술을 도입하기 위한 각자의 전략을 수립할 필요가 있다. 표준 웹 서비스 개발방법론 수립, 표준 웹 서비스 개발도구 및 서버 선정 등이 그 출발점이 될 수 있을 것이며, 특히 웹 서비스 기술들 간의 상호운용성에 유의해야 할 것이다.

## 참고문헌

- [1] Microsoft, "Dell Commits to a .NET Connected Web Services Architecture," <http://www.microsoft.com/resources/case-studies/CaseStudy.asp?CaseStudyID=14674>, October 2003.
- [2] Steve Ulfelder, "GM gears up with collaboration based on Web services," Network World Fusion, <http://www.nwfusion.com/research/2003/0526gm.html>, May 2003.
- [3] 최진권, "마이크로소프트의 웹 서비스 성공사례", 웹 서비스의 가치창출·사업전략·사례 세미나 발표자료, 전자상거래연구조합·전자상거래표준화통합포럼, February 2004.
- [4] WS-I, "Basic Profile Version 1.0 Final Material," <http://www.wsi.org/Profiles/BasicProfile-1.0-2004-04-16.html>, April 2004.
- [5] Russell Butek, "Which style of WSDL should I Use?," IBM DeveloperWorks, <http://www106.ibm.com/developerworks/webservices/library/ws-whichwsdl/>, October 31, 2003.
- [6] Peter Brittenham, "Web Services Development Concepts," IBM DeveloperWorks, <http://www3.ibm.com/software/solutions/webservices/pdf/WSDC.pdf>, May 2001.
- [7] VeriSign Whitepaper, "Securing Web Services - VeriSign Trust Gateway," <http://www.verisign.com/resources/wp/trustgateway/securingWebServices.pdf>, 2003



**백 종 현**



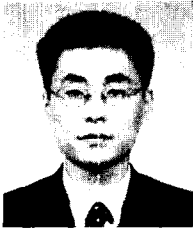
1990 서강대학교 전자계산학과(학사)  
1992 연세대학교 전산학과(석사)  
1996 연세대학교 컴퓨터학과(박사)  
1997~현재 대우정보시스템(주) 기술연구  
소 컨설팅서비스팀장/정보기술연구팀장  
관심분야 : Web Services, EA, CBD,  
ITIL  
E-mail : baegjh@disc.co.kr

**강 병 철**



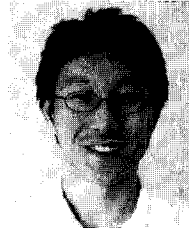
1996 서강대학교 전자계산학과(학사)  
1998 서강대학교 전자계산학과(석사)  
1998~현재 대우정보시스템(주) 기술연구  
소 연구원  
관심분야 : 웹서비스, BPM  
E-mail : bkang@disc.co.kr

**김 형 석**



1992 아주대학교 산업공학과(학사)  
1994 아주대학교 산업공학과(석사)  
1994~1996 (주)동일씨.아이.엠 부설연구소  
연구원  
1996~현재 대우정보시스템(주) 기술연구  
소 차장  
관심분야 : 협업, 물류  
E-mail : hyeoung@disc.co.kr

**김 제 욱**



1999 서울대학교 해양학과(학사)  
2002 서울대학교 컴퓨터공학과(석사)  
2002~현재 대우정보시스템(주) 기술연구  
소 연구원  
관심분야 : XML, 웹서비스, J2EE  
E-mail : jeuk@disc.co.kr

• 11th Asia-Pacific Software Engineering  
Conference(ASPEC 2004)

- 일 자 : 2004년 11월 30일~12월 3일
- 장 소 : 해운대 그랜드 호텔(부산)
- 주 최 : 소프트웨어공학연구회
- 상세안내 : <http://apsec2004.kaist.ac.kr>