

# SaaS 플랫폼 기술 및 개발 동향

The Technology Trends and Development of SaaS Platform

소프트웨어 기술의 미래전망 특집

## 목 차

- I . 서론
- II . SaaS 플랫폼 기술동향
- III . SaaS™ 플랫폼 개발 현황
- IV . 결론

이지현 (J.H. Lee)	SW서비스연구팀 선임연구원
박경현 (K.H. Park)	SW서비스연구팀 선임연구원
이상민 (S.M. Lee)	SW서비스연구팀 선임연구원
이원재 (W.J. Lee)	SW서비스연구팀 선임연구원
오병택 (B.T. Oh)	SW서비스연구팀 선임연구원
강성주 (S.J. Kang)	SW서비스연구팀 연구원
정문영 (M.Y. Chung)	SW서비스연구팀 선임연구원
양경아 (K.A. Yang)	SW서비스연구팀 선임연구원
원희선 (H.S. Won)	SW서비스연구팀 선임연구원
허성진 (S.J. Hur)	SW서비스연구팀 팀장
최 완 (W. Choi)	클라우드컴퓨팅연구부 부장

본 고에서는 SaaS를 서비스로 제공하기 위한 기술인 SaaS(Software as a Service) 플랫폼 기술과 개발 동향을 소개한다. 이를 위해 국내외 SaaS 개발 시장의 동향을 살펴보고 국가 플랫폼 개발의 일환으로 2009년부터 ETRI에서 연구개발 중인 SaaS 플랫폼의 개발 및 기술 현황에 대해 살펴본다.

## I. 서론

새로운 소프트웨어(이하 SW) 유통모델로서 등장한 Software as a Service(SaaS)는 원격 공유 호스팅 환경에서 사용자들에게 온라인 서비스 형태로 SW를 제공하는 모델을 의미한다[1]. SaaS는 때때로 온라인에서 요구형 SW(on-demand software)를 제공하는 기술로 불리기도 하는데, 다수의 사용자는 SW를 온라인 상에서 사용하고 그에 대한 비용을 정액 또는 사용한 서비스 사용량만큼 지불함으로써 SW나 운용 서버를 운영하거나 유지 보수하는 비용을 거의 필요로 하지 않는다. 따라서 사용자는 초기 투자 비용으로부터 자유롭고 관리 비용을 낮춰 경제적이 수 있다는 장점을 갖는다.

### 1. SaaS 시장 전망

시장조사기관 IDC 2010년 발표한 전 세계 SaaS 전망 2010-2014 보고서[2]에서 SaaS 시장규모는 2009년 연간 131억 달러 규모에서 매년 25%씩 증가하여 2010년에는 171억 달러, 2014년에는 405억 달러 규모의 시장으로 고성장할 것으로 전망했다. 그리고 2009년부터 2014년까지 SaaS를 통한 수익은 패키지 SW 성장의 5배 이상으로 증가할 것으로 전망하였다. 또한 SW 시장이 SaaS의 비중이 높아지면서 2012년까지 신생 SW 업체 가운데 패키지 SW를 출시하는 업체는 15% 미만이고 85%의 SW업체가 SaaS에 집중할 것으로 전망하였다.

Gartner의 2010년 SaaS에 대한 하이프 사이클 보고서는[3] 다양한 SW가 SaaS 형태로 개발되고 있으며 고객관리(CRM: Customer Resource Management), 인력관리(HRM: Human Resource Management) 등의 다양한 엔터프라이즈 시장에 SaaS가

도입되고 있는 것을 보여준다.

### 2. SaaS 애플리케이션의 특징

SaaS 애플리케이션은 개인보다는 기업용 애플리케이션으로 개발되고 있는데 데이터 중심의 애플리케이션이 대다수며 여러 사용자가 속한 회사나 기관에서 사용되므로 애플리케이션 사용자가 많다는 특징을 갖는다.

상기 언급된 애플리케이션 사용자 집단은 SaaS에서 테넌트(tenant)라 불리는데 테넌트의 관리자는 애플리케이션의 사용자 인터페이스나 특정 기능을 변경하고자 할 때 SW 공급자를 대신해 직접 애플리케이션의 구성을 변경할 수 있다. 따라서 SaaS 애플리케이션은 여러 테넌트(이하 멀티테넌트)가 애플리케이션의 구성을 변경하여 맞춤 서비스를 생성할 수 있는 구성변경성(customizability)을 제공해야 한다.

SaaS 애플리케이션의 또 다른 특징은 멀티테넌시(multitenancy)를 지원한다는 점이다. 멀티테넌시는 하나의 동일한 애플리케이션 인스턴스를 동작시켜 멀티테넌트에게 필요한 서비스를 제공함을 의미한다.

### 3. SaaS 플랫폼

SaaS 플랫폼은 애플리케이션을 온라인 서비스로 제공하며 테넌트 측 개발자의 개발 용이성을 확대하는데 필요한 기술들을 스택으로 나타낸다.

SaaS 플랫폼에 대해 서로 다른 구성 요소를 갖는 3가지 참조 아키텍처를 정리하고 각 아키텍처의 특징을 살펴본다[4].

(그림 1)의 (1)은 공유 프로세스 기반의 아키텍처로서 다수 테넌트의 트랜잭션이 하나의 특정 애플리케이션 서버에서 수행되고 데이터 플랫폼으로 불리는 데이터베이스는 물리적으로 분리된 구조이다. 테



(그림 1) SaaS 플랫폼 참조 아키텍처[5]

넌트별 애플리케이션에 대한 동작 요청 시 테넌트의 트랜잭션은 메타데이터를 데이터베이스에서 가져와 동적으로 사용자 인터페이스, 비즈니스 로직, 데이터를 구성한다. 이 때 물리적 메타데이터를 해석할 수 있는 특별 가상 머신이 필요하며 이 아키텍처의 대표적인 예는 Codys 플랫폼, Relationals의 LongJump이다.

(그림 1)의 (2)는 SaaS 플랫폼의 전체를 공유하는 아키텍처로 위에 살펴본 공유 프로세스 기반의 아키텍처와 다른 점은 테넌트별로 물리적인 데이터베이스를 테넌트별로 분리하지 않고 논리적으로 하나의 데이터베이스를 사용하여 모든 기술을 공유함으로써 멀티테넌시를 지원한다는 점이다. 이 예로는 Google App Engine, RollBase의 Platform as a Service(PaaS), Salesforce.com의 Force.com, 한국전자통신연구원에서 개발하고 있는 SaaSpia™가 있다.

(그림 1)의 (3)은 위의 SaaS 플랫폼의 전체를 공유하는 아키텍처 외 멀티테넌트들에게 애플리케이션의 구성변경성을 지원하는 방법으로 맞춤형 멀티테넌시 지원 아키텍처를 추가적으로 제공한다. 이는 애플리케이션 설계 및 구현에 구성 변경 알고리즘을 반영하고 이러한 설계 및 구현 조건들이 해석되는 범용 애플리케이션 플랫폼(즉, 범용적인 SaaS 플랫폼)을 제공하는 것을 의미한다. 이 아키텍처는 제시된 세 가지 SaaS 플랫폼 중 설계 및 구현 수준이 가장 높다

[5]. Salesforce.com, ForeSoft, Archer Technologies는 이 아키텍처를 지원하고 한국전자통신연구원의 SaaSpia™ 역시 범용적인 플랫폼을 지원한다.

## II. SaaS 플랫폼 기술동향

### 1. SaaS 플랫폼 핵심 기술

#### 가. 멀티테넌트 지향 애플리케이션 개발 및 설정

SaaS 플랫폼의 핵심 기술은 멀티테넌시를 지원하여 테넌트별 요구사항이 반영되도록 하는데 있다. 테넌트의 요구사항은 애플리케이션의 구성을 테넌트가 원하는 대로 변경할 수 있게 반영되어야 하므로 플랫폼은 애플리케이션 설정 기능을 제공하고 테넌트의 요구사항은 SaaS 플랫폼으로 전달되어야 한다. 즉, SaaS 플랫폼은 테넌트들에게 테넌트별로 구성된 서로 다른 서비스와 독립된 서비스 개발 환경을 제공한다[6],[7],[8].

테넌트 요구사항들은 SaaS 플랫폼에서 메타데이터로 관리되고 테넌트별 서비스가 메타데이터 기반 아키텍처(metadata-driven architecture)에 기반하여 동적으로 서비스를 생성하여 테넌트에게 제공된다[9]. 이 때 SaaS 플랫폼은 하나의 인스턴스를 통해 테넌트들에게 독립된 서비스를 제공한다.

멀티테넌시의 장점은 멀티테넌트를 위한 애플리케이션의 설정 기능 이외 서버 자원의 공유 측면에서 상당한 비용을 줄일 수 있다는 것이다. 기존의 단일 테넌트 SW 개발의 한계점은 테넌트들이 가상 서버를 통해 서비스를 제공받기 때문에 테넌트별로 분산된 리소스와 독립된 서버 환경을 가지고 있기 때문에 테넌트의 수가 증가할 수록 유지보수 비용이 증가한다는 점이다. 하지만 SaaS에서와 같이 공유 호스팅

환경에서 리소스를 공유하고 하나의 애플리케이션 인스턴스를 통해 멀티테넌트에게 서비스를 제공하면 테넌트의 수가 증가할 수록 유지보수 비용은 상대적으로 낮아질 수 있다[8].

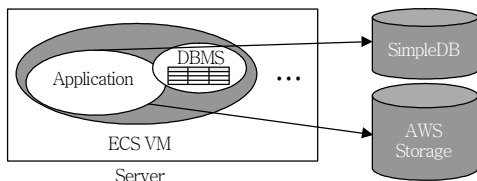
나. 테넌트 데이터 관리

SaaS 플랫폼 핵심 기술로 테넌트 데이터 관리가 있다. 테넌트 데이터는 테넌트별 애플리케이션에 대한 논리적 데이터로서 공유 호스팅 환경에서는 물리적 데이터 저장 구조에 정형화된 형태로 저장 및 관리되어야 한다. 테넌트 데이터 관리는 이와 같이 효율적인 멀티테넌시를 지원하는 방법으로 필수적이다. 또한 테넌트 데이터는 SaaS 애플리케이션에 대해 다양한 구성변경성을 제공할 때 이용되므로 테넌트 관리자가 필요에 따라 자사의 테이블을 확장 또는 변경할 수 있어야 한다. 아래에 테넌트 데이터를 관리할 구현하는 3가지 방법을 살펴본다.

1) 공유 서버

공유 서버 기법은 테넌트별 데이터를 공유 서버에 개별 가상 머신(이하 VM: Virtual Machine)을 설치하여 이 곳에 테넌트 데이터를 저장 및 관리하는 기법이다. 각 테넌트는 공유 서버에 자신의 데이터베이스 시스템을 소유한다. 이 기법은 데이터베이스 변경이 필요하지 않기 때문에 널리 이용된다. 대표적인 예로 Amazon 클라우드 플랫폼의 Amazon Web Services (이하 AWS: Amazon Web Services)가 있다[10].

(그림 2)의 AWS는 비관계형 데이터 저장소로 도

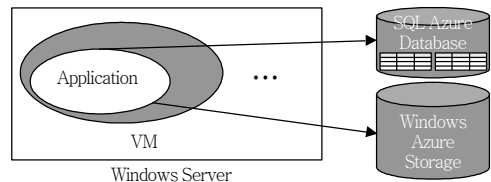


(그림 2) Amazon Web Services

메인이라 부르는 단위로 데이터를 관리하는 SimpleDB를 함께 제공한다. 그러나, SimpleDB는 일반적으로 많이 사용하는 관계형 데이터베이스가 아니므로 관계형 데이터 베이스 애플리케이션을 만들 경우에는 테넌트 VM에 관계형 데이터베이스를 설치하여 사용한다.

2) 공유 프로세스

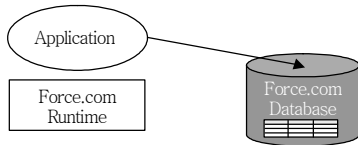
공유 프로세스(DBMS: Database Management System) 방식은 테넌트들이 하나의 데이터베이스를 공유하고, 각 테넌트가 하나의 자신의 데이터베이스 공간을 가지는 방법이다. 이 기법은 공유 서버 기법보다 메모리와 데이터베이스 연결 풀을 공유함으로써 보다 효율적인데, 이 기법의 예로 마이크로소프트의 Windows Azure Platform의 SQL(Structured Query Language) Azure가 있다. (그림 3)의 SQL Azure[11]은 클라우드 플랫폼의 고가용성을 보장하는 관계형 데이터베이스 인프라이다. 각 테넌트는 하나의 데이터베이스 공간을 할당 받아서 지역 데이터베이스 사용과 같이 접근할 수 있다. 마이크로소프트의 관계형 데이터베이스인 MS-SQL와 같이 개발 가능하고 쿼리 모델도 자사의 T-SQL의 대부분을 지원하며 MS-SQL와 연동하는 대부분의 도구와 호환 가능하다.



(그림 3) Windows Azure Platform

3) 공유 스키마

테넌트 테이블을 확장시키기 위해 각 테이블은 추가적인 테이블 칼럼이 있고 모든 테넌트의 데이터는



(그림 4) Force.com

(그림 4)의 하나의 스키마에 저장된다.

공유 스키마의 칼럼은 여러 데이터 형을 수용하기 위해서 문자열 형태로 되어 있고 이 기법은 모든 리소스를 공유할 수 있는 최고의 기법이다.

이 기법을 사용하는 대표적인 클라우드 플랫폼은 Salesforce.com의 Force.com 플랫폼이다. 공유 스키마에 모두 저장된 데이터는 테넌트별로 파티션되고 개별 디스크에 저장되어 지역성을 높인다. Force.com의 런타임 엔진은 애플리케이션 실행 시 SOQL(Salesforce Object Query Language)와 APEX를 통해 테이블에 접근한다.

## 2. 해외 기술 개발 동향

### 가. Salesforce.com의 Force.com 플랫폼

SaaS 플랫폼 선두업체들 중의 하나인 Salesforce.com은 엔터프라이즈 클라우드 컴퓨팅 플랫폼으로 Force.com을 제공하고 있으며, 어떤 경쟁업체보다도 많은 수의 기업용 애플리케이션이 Force.com을 통해 제공되고 있다. 앞에서 살펴 본 범용적이며 SaaS 플랫폼의 전체를 공유하는 SaaS 플랫폼 아키텍처를 통해 멀티테넌시, 높은 확장성(scalability), 높은 생산성을 지원하는 Force.com은 가장 성숙한 클라우드 애플리케이션 플랫폼으로 평가 받고 있다[12].

Force.com에서는 사용자 인터페이스 개발을 위해 Visualforce 언어, 비즈니스 로직 개발을 위해 Apex 언어, DB 질의를 위해 SOQL을 제공한다. 이러한 언어를 통해 높은 생산성과 테넌트 분리(tenant isolation) 기능을 제공한다. Force.com을 사용하여 Sales-

force.com CRM 애플리케이션의 구성을 변경할 수 있고, 또한 별도의 비즈니스 애플리케이션을 구현할 수 있다.

### 나. WaveMaker

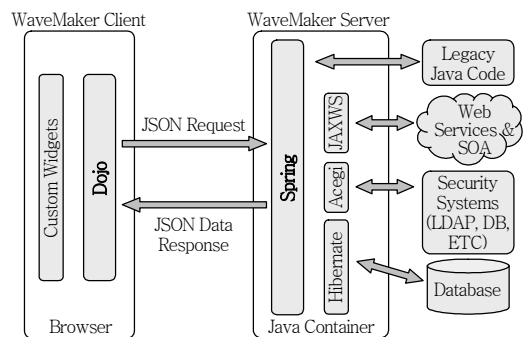
WaveMaker는 웹 2.0 애플리케이션 개발을 고속화하는 애플리케이션 개발 환경(rapid application development environment)으로 WaveMaker Studio와 WaveMaker Enterprise Server로 구성된다[13].

개발 도구인 WaveMaker Studio는 웹 브라우저 상에서 동작하며 위지위그 드래그 & 드롭(WYSIWYG Drag and drop)을 통한 쉬운 개발을 지원한다. 개발 결과는 자바(Java) 애플리케이션 저장되며 자바 IDE를 사용하여 수정 및 확장 개발이 가능하다.

WaveMaker Enterprise Server는 개발된 WaveMaker 애플리케이션을 실행할 수 있는 표준 기반 자바 서버로 표준 자바 컴포넌트를 사용하여 보안, 데이터 접근, 확장성(scalability)을 지원한다.

개발된 웹 애플리케이션을 웹 애플리케이션 서버 또는 Amazon EC2, Rackspace, OpSource, Eucalyptus 등 RightScale에 의해 관리되는 클라우드 인프라에 배포할 수 있다.

(그림 5)의 WaveMaker는 내부적으로 Dojo, Spring, Hibernate, JAXWS(Java API for XML Web



(그림 5) WaveMaker 개념도

Services), JSON-RPC, Acegi 등과 같은 공개 SW 기술 및 표준 기술을 사용한다.

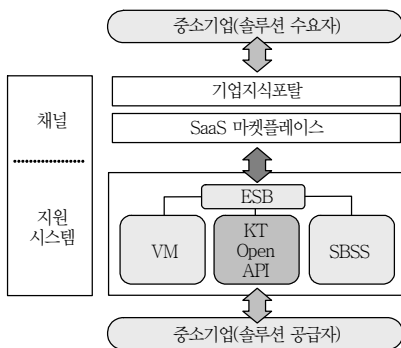
### 3. 국내 기술 개발 동향

#### 가. KT의 SaaS 플랫폼

KT의 비즈메카는 소프트웨어가 중심인 IT 컨버전스를 주도하기 위한 임대 플랫폼 서비스[14]로 중소기업 및 중소기업에게 필요한 e-비즈니스 솔루션, 콘텐츠, 컨설팅, 교육 등을 통합적으로 제공하고 관리를 대행해 주는 플랫폼 서비스이다.

비즈메카 플랫폼은 새로운 서비스가 웹 서비스 기술을 통해 손쉽게 추가되고 연계되어 개발되도록 지원하기 때문에 고객이 필요로 하는 맞춤형 서비스를 제공할 수 있다. 또한 KT는 지난해 지식경제부 주관으로 추진된 SaaS 마켓인 오아시스(OAASYS: Office as a Service System)를 기반으로 비즈메카 서비스를 확장하여 2011년 10월에 정식 서비스할 계획이며, 2011년 말까지 유·무선 통신 인프라를 오픈 API 형태로 개방하고 이를 활용하여 기업용 소프트웨어 기업들이 다양한 서비스를 제공할 수 있도록 지원할 계획이다[15]. (그림 6)은 KT 오픈마켓 서비스의 기반인 OAASYS의 구조도를 보여준다.

KT의 오픈마켓 서비스는 기업용 솔루션 오픈마



(그림 6) KT 오픈마켓 서비스의 기반인 OAASYS 플랫폼 구조도[16]

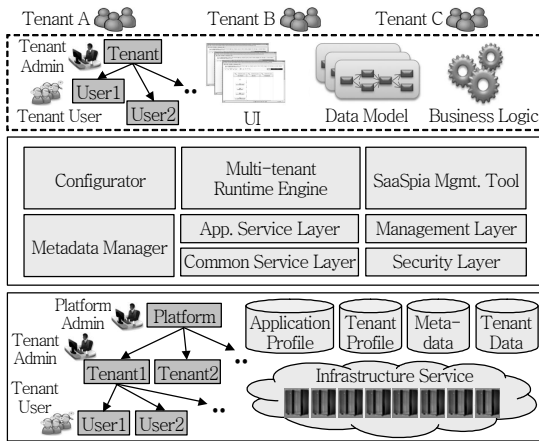
켓 형태로 개발사가 자사 솔루션을 마켓에서 자유롭게 판매하고, 기업 고객은 온라인 상에서 마켓에 접속해 필요한 솔루션을 찾아 구매할 수 있다. KT는 기존 인프라 서비스(IaaS: Infrastructure as a Service)에 이어 SaaS까지 클라우드 서비스의 사업 영역을 확대하였으며, 당분간 SaaS로 제공할 솔루션을 늘리기 위해 국산 SW업체와의 협력 관계를 확대하고 기존 ASP(Application Service Provider) 사업인 KT 비즈메카에서 판매했던 솔루션까지 추가할 계획이다[17]. KT에서 제공하는 SaaS 서비스는 아직 공유 프로세스 기반의 멀티테넌시 지원 아키텍처를 지원하지 못하며 기존 ASP 사업과 유사한 SaaS 기술 성숙도 레벨 1의 수준을 제공하고 있다.

#### 나. ETRI의 SaaSSpia™ 플랫폼

한국전자통신연구원은 다양한 애플리케이션 SW를 중소기업에 저비용, 고효율로 온라인 서비스화하기 위한 SaaS 플랫폼 개발과제를 2009년부터 수행하고 있다. SaaS 플랫폼에 대한 민간기업들의 관심과 기술 수요가 큰 만큼 과제 초기에 민간 분야의 SaaS 관련 단체인 SaaS 코리아포럼과의 협조를 통해 요구사항을 정의하였다.

한국전자통신연구원에서 개발하고 있는 SaaSSpia™는 SaaS와 Utopia를 결합한 것으로 개발된 SaaS 플랫폼의 명칭이며, 범용적인 SaaS 플랫폼 아키텍처를 통해 SaaS 기술 성숙도 레벨 3 수준의 애플리케이션을 개발하고 운영할 수 있도록 지원하여 동일한 SW와 데이터베이스를 이용하여 서비스하면서도 테넌트별로 접근 독립성과 설정 환경을 지원하여 자사만의 SW처럼 사용하게 하는 멀티테넌트 지원 플랫폼이다.

이 플랫폼을 기반으로 자유무역협정(이하 FTA) 원산지관리시스템 등의 적용 서비스를 선정하고 개



(그림 7) SaaS™ 구조도

발하여 시범 서비스 단계에 있다.

SaaS™의 대표적인 기능 모듈로는 SaaS 애플리케이션 설정 도구, 멀티테넌트 실행 엔진, 메타데이터 관리자, 플랫폼 통합 관리도구 등이 있으며 (그림 7)과 같다.

SaaS 애플리케이션 설정 도구는 SaaS™기반의 서비스를 이용하는 테넌트의 관리자를 위한 도구이다. 기능으로 SaaS™ 기반으로 개발된 애플리케이션을 코드 변경 없이 테넌트의 필요에 맞게 설정할 수 있는 환경을 제공한다. 애플리케이션에서 설정할 수 있는 대상은 사용자 인터페이스, 데이터 스키마, 비즈니스 로직 및 워크플로우에 해당한다. SaaS™는 애플리케이션 설계 및 구현 시 구성 변경을 미리 고려하여 표현 및 해석될 수 있도록 알고리즘과 처리 도구를 제공하고 이로써 범용적인 SaaS 플랫폼 아키텍처를 지원한다.

멀티테넌트 실행 엔진은 멀티테넌트 지원 아키텍처의 메타데이터를 해석할 수 있는 특별 가상 머신 기술의 구현이다. 이로써 테넌트별 메타데이터를 기반으로 테넌트별 독립적인 서비스를 생성하여 사용자에게 제공하는 기능을 한다.

메타데이터 관리자는 하나의 데이터베이스로 멀

티테넌트의 논리적 데이터를 저장하고 관리하기 위한 기술을 구현한 것이다. SaaS™ 설정 도구와 실행 엔진을 통해 메타데이터 접근을 지원하고 테넌트별로 설정 환경을 통해 변경된 요소들을 메타데이터로 저장하고 관리한다. 또한 애플리케이션 실행 시 메타데이터 지원 API를 이용해 테넌트별로 분리된 데이터를 제공하며 테넌트별 추가 확장 가능한 여분의 데이터 필드를 제공한다.

플랫폼 통합 관리 도구는 SaaS™ 플랫폼 관리자를 위한 시스템, 테넌트, 사용자 및 서비스의 통합 관리 도구로서 실행 중인 서비스별 플랫폼 사용 정보의 실시간 모니터링 제공, 테넌트별 플랫폼 사용 정보 관리 기능 등을 제공한다.

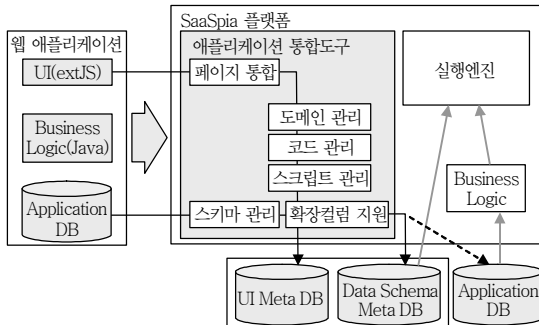
### III. SaaS™ 플랫폼 개발 현황

#### 1. 애플리케이션 통합 생성 기술

기존의 웹 애플리케이션은 단일 테넌트 구조이기 때문에 SaaS 애플리케이션으로 변환하기 위해서는 SaaS™ 플랫폼 상에 웹 애플리케이션을 통합해야 한다. SaaS™ 플랫폼 상의 웹 애플리케이션은 범용적인 SaaS 플랫폼 아키텍처를 통해 생성 및 구성변경될 수 있다. 이를 위해 SaaS™ 플랫폼은 기존의 웹 애플리케이션을 SaaS™ 플랫폼 상에 통합하기 위해 SaaS™ 통합도구라는 애플리케이션 통합 인터페이스를 제공한다. 즉, SaaS™ 통합도구는 기존의 웹 애플리케이션에서 설정 가능한 부분을 메타데이터로 변경해주는 기능을 수행한다.

(그림 8)은 SaaS™ 통합도구를 이용하여 웹 애플리케이션을 통합하는 모습을 보여준다.

기존 웹 애플리케이션을 SaaS™ 플랫폼 상에 통합하기 위해서 웹 애플리케이션이 ExtJS(Extended



(그림 8) SaaSPIA™ 통합도구의 동작 예

JavaScript)와 Spring 기반의 자바로 구성되어 있어야 한다. ExtJS로 구성되는 사용자 인터페이스 부분은 통합도구의 페이지 생성 기능을 통해 메타데이터로 변환되어 저장된다. 그 외 (그림 8)에서처럼 SaaSPIA™ 통합도구는 웹 애플리케이션을 통합하기 위해 도메인관리, 코드관리, 스크립트 관리 등을 기능을 함께 지원한다.

## 2. 애플리케이션 설정 기술

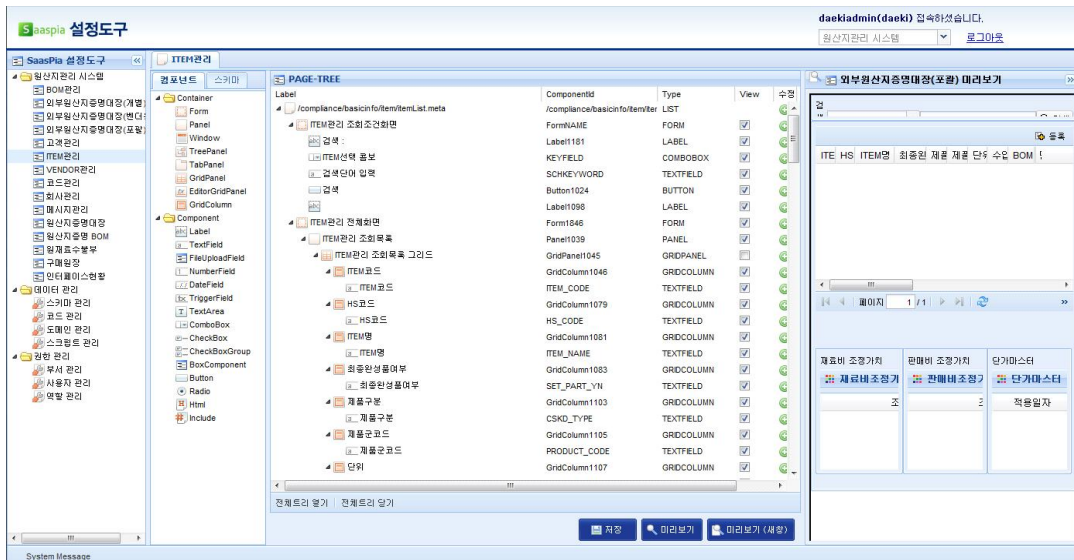
SaaSPIA™ 설정도구는 테넌트별 요구사항을 SaaS 애플리케이션에 반영할 수 있는 환경을 제공하는 인

터페이스로 사용자 인터페이스, 데이터 스키마, 비즈니스 로직 설정 기능을 제공한다.

### 가. 사용자 인터페이스(UI) 설정 기술

사용자 인터페이스는 ExtJS 기반의 자바스크립트로 구현된 웹 페이지로 SaaSPIA™에서는 트리 모델로 관리된다. SaaSPIA™ 설정도구는 테넌트별로 독립된 페이지를 설정 가능하게 하며 하나의 페이지는 사용자 인터페이스를 구성하는 컴포넌트들로 저장되기 때문에 페이지 내 컴포넌트의 위치 변경, 레이블 변경, 테이블 필드 변경 등 컴포넌트 단위의 설정이 용이하다. 설정한 페이지는 페이지를 구성하는 컴포넌트들을 동적으로 조합하여 웹 페이지로 렌더링하여 보여준다. (그림 9)는 SaaSPIA™ 설정도구의 모습을 보여준다.

(그림 9)에서 보는 것처럼 SaaSPIA™ 설정도구는 페이지 설정 이외에도 애플리케이션 스키마를 관리하기 위한 스키마 관리 기능을 제공하고 웹 페이지를 통한 데이터 유효성 검사 및 다양한 비즈니스 로직을



(그림 9) SaaSPIA™ 설정도구



제공하기 위해 스크립트 관리 기능을 함께 제공한다.

나. 데이터 스키마 설정 기술

SaaSSpia™는 관계형 데이터베이스 인프라를 이용한 공유 스키마 기반의 테넌트 데이터 관리를 지원함으로써 애플리케이션에 공통적으로 제공되는 데이터 스키마와 테넌트별 필요에 의해 확장 가능한 스키마를 제공한다.

데이터 테이블의 고정필드는 모든 테넌트들에게 공통적으로 제공되고 확장필드는 테넌트별로 설정되어 확장될 수 있는 필드이다. 하지만 하나의 확장필드는 테넌트에 따라 다른 타입으로 정의되어 사용되므로 동일한 확장필드라고 해도 테넌트별로 다른 용도로 사용된다.

SaaSSpia™ 설정도구는 도메인 관리를 통해 확장 필드에 대한 데이터 타입, 범위, 유효성에 대한 검사를 지원하며 코드 관리 기능을 통해 애플리케이션에서 사용할 열거형 타입들을 제공한다.

다. 스크립트 기반 비즈니스 로직 설정 기술

비즈니스 로직은 UI와 데이터베이스 간의 정보 교환을 통해 일어나는 논리를 표현한 것으로, SQL로 작성되어 모델-뷰-컨트롤러(이하 MVC: Model-View-Controller) 구조로 구성된 SaaS 애플리케이션의 컨트롤러에 의해 실행된다.

일반적으로 사용자는 웹 애플리케이션 개발자가 제공하는 비즈니스 로직의 범위 내에서 기능을 동작시킬 수 있으나 SaaSSpia™에서는 개발자가 제작한 비즈니스 로직을 설정하여 변경할 수 있는 기능을 제공한다.

SaaSSpia™의 비즈니스 로직 설정 기술은 자바 스크립트를 이용한 방법과 자바 컨트롤러를 이용한 방법으로 구분된다.

성명	직급	입사년도	연차
김○○	팀장	1985	27
이××	사원	2004	8

(그림 10) 비즈니스 로직 설정 개념

SaaSSpia™의 스크립트 기반 비즈니스 로직 설정 기술은 UI 메타로 저장된 서비스 페이지가 렌더링 될 때, 페이지 내에 특정 UI 컴포넌트에 들어갈 값을 처리하는 비즈니스 로직을 위한 스크립트를 설정하는데 그 목적이 있다. 예를 들어 (그림 10)과 같이 개발자가 제작한 임의의 페이지 내에 사용자의 정보를 표시하는 컴포넌트에서 스크립트 기반의 비즈니스 로직을 설정하여 본다. 개발자가 제공한 그리드의 열은 ‘성명’, ‘소속’, ‘입사 년도’로 3개이지만 특정 테넌트는 ‘연차’라는 새로운 필드를 만들고자 한다면, 앞 절에서 설명한 사용자 인터페이스 설정 기술을 통해 새로운 필드를 선언하고, 관련 실제 데이터를 넣어 보일 수 있다. 하지만 실제로 이 열에 들어갈 값은 ‘현재 년도 - 입사 년도 + 1’과 같은 연산을 수행한 후에 결정된다.

이와 같은 로직은 자바스크립트를 통해 처리되며, ‘연차’라는 필드와 해당 스크립트는 이 필드를 사용하고 하는 테넌트에만 한정되는 내용이므로, 스크립트는 해당 테넌트의 메타데이터로 저장되며 메타데이터를 해석할 수 있는 메타데이터 관리자와 멀티테넌트 실행 엔진을 통해 테넌트별 페이지 렌더링 시 함께 실행된다.

SaaSSpia™는 다양한 ExtJS 컴포넌트(예: Grid, Button, Textfield 등)의 값들 간의 연산을 위한 다양한 스크립트를 제공하며, 테넌트 관리자는 설정도구를 이용해 UI 설정 시 컴포넌트에 스크립트를 할당하여 연산 기반의 비즈니스 로직을 처리한다.

### 라. 컨트롤러 기반 비즈니스 로직 설정 기술

스크립트 기반 비즈니스 로직을 UI 컴포넌트에 연결함으로써 간단한 연산이 처리되도록 하는 방식은 테넌트가 SaaS 서비스에서 제공되지 않는 데이터 처리 관련 비즈니스 로직을 추가하고자 할 때 한계가 있다. 이런 경우 테넌트 관리자는 컨트롤러 기반 비즈니스 로직 설정 방법을 이용하여 테넌트가 원하는 데이터를 처리할 수 있게 처리 로직을 추가하고 실행하게 할 수 있다.

컨트롤러 기반 비즈니스 로직 설정은 두 단계로 수행된다. 먼저 로직 처리에 필요한 새 SQL을 추가한 다음 SQL을 처리할 서비스 처리 모듈(즉, 컨트롤러)을 구현하고 로직 실행을 위한 입출력 변수에 실제 입력 값을 설정한다. 설정된 새 SQL과 비즈니스 로직은 비즈니스 메타 로직 데이터로 저장되고 UI 컴포넌트를 통해 동작이 요청될 때 동적으로 해당 테넌트의 비즈니스 로직 코드로 생성되어 요청한 테넌트에게 제공된다.

### 마. 워크플로우 설정 기술

워크플로우 설정 기능은 SaaSSpia™ 플랫폼 기반 애플리케이션 이용 시 테넌트별로 워크플로우를 다르게 설정하여 테넌트별 프로세스에 따라 업무를 수행할 수 있도록 지원하는 기능이다.

워크플로우 설정은 SaaSSpia™ 기반의 UI 컴포넌트에 처리할 업무들을 프로세스로 정의하고 사용자 역할 등의 업무 상세를 명시함으로써 이뤄진다.

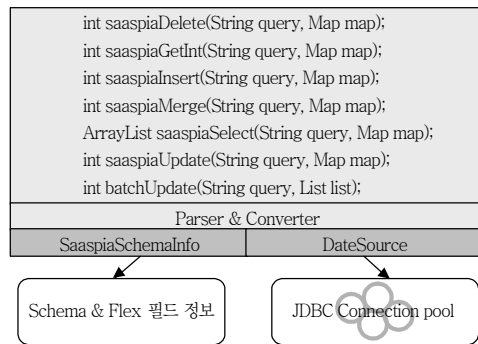
또한 테넌트의 기능에 맞게 워크플로우를 새로 정의하거나 재구성한 후 워크플로우 액티비티의 속성에 기존에 설정된 UI를 매핑하거나 로직을 변경하는 기능을 제공할 수 있다. 이로써 테넌트 사용자는 자신의 역할에 따라 애플리케이션의 UI가 변경되면서 업무를 수행할 수 있다.

## 3. 테넌트 데이터 처리 및 관리 기술

SaaSSpia™ 기반 애플리케이션은 해당 테넌트 데이터 접근을 위해 테넌트 데이터 접근 API를 제공한다. 테넌트 데이터 API는 테넌트 데이터 관련 삽입, 삭제, 갱신, 검색, 배치 갱신 기능을 제공한다.

(그림 11)은 테넌트 데이터 관리기가 수행하는 동작을 보여준다.

SaaSSpia™ 플랫폼에서의 테넌트 데이터의 처리는 공유 테이블로부터 스키마 정보를 읽어 테넌트별 접근 제어와 추가 필드 설정 여부를 파악한 후 테넌트 정보를 포함할 수 있게 애플리케이션의 쿼리를 변환하고 데이터베이스로부터 테넌트 데이터에 접근한다.



(그림 11) SaaSSpia™ 테넌트 데이터 API

### 가. 테넌트 공유 스키마

SaaSSpia™ 플랫폼은 애플리케이션 데이터 스키마를 테넌트별로 공유함으로써 수십 혹은 수백 개의 스키마가 생성되는 것으로 인한 물리적 생성 공간과 쿼리처리 비용을 감소시킨다.

테넌트들이 스키마를 공유함으로써 응용프로그램의 데이터 요청에 대해 해당 테넌트의 데이터 영역으로 접근할 수 있도록 해야 하는데 이를 위해 사용자의 애플리케이션 쿼리를 파싱하고 공유 테이블 접근에 해당하는 부분을 세션 정보를 기반으로 변경하여 실행시킨다.

나. 테넌트 스키마 확장

테넌트가 스키마를 공유하더라도 테넌트별로 데이터 스키마를 설정할 필요가 있다. 이를 위해 SaaS-pia™의 애플리케이션의 테이블은 여분의 확장필드를 추가하여 테넌트별 동적 설정을 지원한다.

예를 들어, 물품 테이블에 한 테넌트는 (그림 12)와 같이 전화번호 필드를 추가하고, 다른 테넌트는 제조사를 추가할 수 있다. 이와 같이 설정도구를 통해 테넌트별 특성화된 테이블 설정을 가능하게 하고 테넌트별 확장필드 관련 연산으로 필드 추가/삭제 및 타입 정의, 필드명 변경의 기능을 제공한다.

Tenant ID	Product ID	-	-	FLEX_1	FLEX_2	제조사
17	10	Acme	St. Paul	NULL	Sandisk	
17	11	Gump	LA	NULL	Transcend	
17	20	Richard	Roma	NULL	LG	
35	100	Kim	Deagu	053-764-2431	NULL	
35	121	Yoon	Busan	051-543-4321	NULL	
35	212	Hyun	Seoul	02-232-1342	NULL	

(그림 12) 테넌트별 스키마 확장필드 설정

4. SaaS 플랫폼 프레임워크

SaaS 플랫폼 프레임워크는 멀티테넌시 지향 SaaS 애플리케이션을 개발하고 설정하기 위한 라이브러리 형태의 개발자 코드를 제공한다. 이 코드들은 SaaS-pia™의 멀티 테넌트 설정 환경, 실행 엔진, 메타데이터 관리, 플랫폼 관리도구 등과 연동함으로써 서비스를 멀티테넌트에게 제공하고 구성 변경하여 사용할 수 있다.

가. 프레임워크 계층 기능

SaaS-pia™ 프레임워크는 관리 계층, 공통 서비스 계층, 애플리케이션 서비스 계층, 보안 계층과 플랫폼 관리도구로 분류되어 SaaS 애플리케이션의 전체 라

이프 사이클에 필요한 플랫폼 API를 제공한다.

관리 계층은 시스템 모니터링, 시스템 데이터 관리, 서비스 프로비저닝의 기반이 되고, 공통 서비스 계층은 다양한 애플리케이션에서 발생하는 에러/예외 처리, 로깅, 배치 작업을 처리한다.

애플리케이션 서비스 계층은 테넌트가 서비스의 사용자 인터페이스, 비즈니스 로직을 커스터마이징하고 테넌트 데이터에 접근하기 위한 API를 제공하며 보안 계층은 인증 처리, 사용자 및 역할 기반 권한 관리, 서비스 접근 제어에 대한 기반 API를 제공한다.

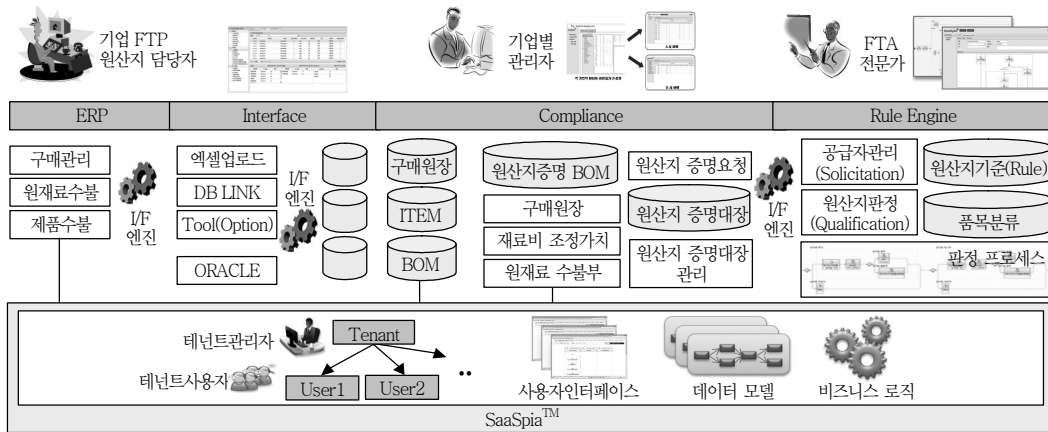
플랫폼 관리도구는 관리 계층과 공통 서비스 계층 API를 활용하여 서비스 프로비저닝, 시스템 모니터링, 서비스 모니터링, 테넌트 관리의 기능을 제공한다.

나. 표준화된 프레임워크 제공

SaaS-pia™ 프레임워크는 전자정부 표준프레임워크 기반 개발 환경 및 실행 환경 기반으로 개발되어 SaaS 애플리케이션 개발 및 실행에 오픈 소스를 이용한 표준 연계 인터페이스를 제공한다. 따라서 개발 및 실행 시 특정 솔루션에 의존하지 않고 독립적으로 SaaS 애플리케이션을 개발 및 실행할 수 있다. 일례로 SaaS-pia™ 프레임워크에서 계층으로 그룹화된 플랫폼 기능들은 Spring MVC 모델에 기반하여 구성되어 있어 SaaS 애플리케이션을 개발하거나 구성 변경할 때 데이터 구조, 사용자 인터페이스, 서비스 처리부에 집중하여 해당 모듈만 개발할 수 있다.

5. 서비스 사례 개발

SaaS-pia™ 적용 사례 연구로 기업 맞춤 원산지관리 시스템을 개발하였다. 해당 시스템은 FTA를 비롯한 국제통상 규제에 효과적으로 대응하기 위해 데이터 마트, 프로세스 기반 원산지 판정, 기업 맞춤 서비스 등을 제공하며 (그림 13)과 같이 구성된다.



(그림 13) SaaSapia™ 원산지관리시스템 구성도

원산지 관리는 90년대 이후 자유무역협정(FTA)이 체결되면서 FTA에 적극적인 대응하고자 필요한 기술인 데 이를 사례 개발 대상으로 선정한 배경을 살펴보자. FTA 협정이 점차 늘고 있고 기업이 수출에서 관세 혜택을 받기 위해서 체계적인 관리가 필요하나 기존의 시스템들이 추가되는 협정마다 다른 판정 절차나 품목마다 다른 기준들을 적용해야 할 때 애플리케이션의 추가 개발이나 수정하는 것이 자금 및 인력이 부족한 중소기업에서 어려운 실정이다[18]. SaaSapia™ 기반 원산지관리시스템은 FTA 대응이 어려운 중소기업의 이러한 한계를 극복하고 저비용·고효율로 SaaS 플랫폼 기반의 온라인 SW 서비스를 제공하고자 개발되었다.

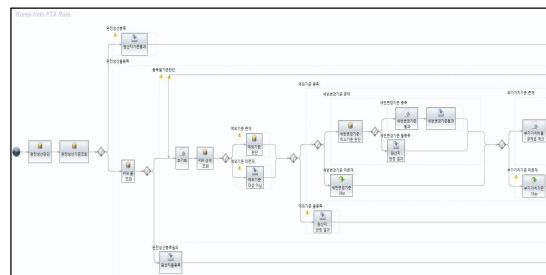
가. 데이터 마트

데이터 마트(Compliance)는 원산지 증명을 위해 기존 ERP에 담겨있는 구매 및 원가 정보를 끌어와 원산지 판정을 가능하도록 데이터를 가공하고 원산지 증명 자료를 관리하는 솔루션이다. 회사의 전산 시스템과 연동 가능한 인터페이스를 제공하여 구매정보, 매출정보, 원가정보를 FTA별 원산지 판정 요건에서 요구하는 수준으로 가공하여 원산지 판정 관련 DB를 구성한다.

또한 협력업체와 연계를 지원하는데 협력업체 사용자가 제출한 원산지 확인서를 완제품업체 관리자가 확인하고 관리할 수 있고 원산지 판정한 데이터를 기반으로 원산지 증명서를 출력 및 저장할 수 있다.

나. 프로세스 기반 원산지판정 룰 엔진

SaaSapia™ 원산지관리시스템은 프로세스 기반 원산지 판정을 지원한다. 프로세스 기반 원산지 판정에 필요한 절차를 BPM(Business Process Management) 기반의 비즈니스 프로세스로 기술하여[19] 원산지판정 절차를 프로그램 소스 코드 대신 BPM 기반의 룰 엔진으로 처리할 수 있다. 이는 새로운 협상에 대한 로직이 추가되어야 할 때 기존의 시스템에 영향 없이 프로세스로 판정 절차를 등록할 수 있게 한다. (그림 14)는 한-인도 CEPA 협정의 원산지 판



(그림 14) 원산지판정 프로세스 예[20]

정 절차를 설계한 프로세스의 일부를 보여준다.

#### 다. SaaS 기반 기업맞춤 서비스

SaaS<sup>TM</sup>는 멀티테넌시를 지원하므로 재료공급 업체, 반제품 공급업체, 완제품 수출업체와 같은 각 기업은 개별적인 요구사항에 맞도록 서비스를 설정하여 기업별 맞춤 서비스를 각 기업에 최적화되게 구성할 수 있었다. 특히 한 시스템을 통한 협력업체와의 원활한 연계가 가능했다.

또한 해당 시스템은 국내 FTA 원산지 시스템을 구축, 관리하기 어려운 환경에 있는 중소기업들이 인터넷을 통해 간편하게 임대하여 사용하게 함으로써 FTA 원산지관리시스템을 신속하게 서비스하고, 시스템 제공 비용을 최소화할 수 있도록 하였다. 이는 자금 및 인력이 부족한 중소기업들에게 복잡하고 어려운 원산지 관리 과정을 체계적이고 철저하게 관리할 수 있도록 하여 저렴한 비용으로 중소기업의 다양한 요구사항에 유연하게 대응하는 서비스임을 입증할 수 있었다.

### IV. 결론

본 고에서는 SaaS를 서비스로 제공하기 위한 SaaS 모델에 대해 살펴 보고 국내외 플랫폼 개발 기술 동향에 대해 설명했다. 그 다음으로 SaaS 플랫폼을 구축하기 위한 핵심 기술에 대해 알아 보았다.

한국전자통신연구원에서 개발되고 있는 SaaS-pia<sup>TM</sup> 플랫폼은 멀티테넌시를 지향하는 플랫폼 핵심 기술로 개발되어 원산지관리 적용 분야로 사례 연구가 되었다. 또한 SaaS<sup>TM</sup> 플랫폼은 FTA 원산지 관리 해당 민간기업으로 기술 이전되어 시범 서비스를 제공 중에 있다.

2011년 3차년도를 맞은 SaaS<sup>TM</sup> 개발은 플랫폼

최적화를 개발 작업 중에 있으며 2차 시범 서비스 개발 사례에 대한 연구가 진행 중에 있다.

#### ● 용 어 해 설 ●

**멀티테넌트**: 조직, 기관, 단체와 같은 애플리케이션 사용자 집단을 하나의 테넌트라고 하고 여러 사용자 집단을 멀티테넌트라고 부름.

**멀티테넌시**: 하나의 동일한 애플리케이션 인스턴스를 동작시켜 멀티테넌트에게 필요한 서비스를 제공하는 소프트웨어 아키텍처 구성 원칙

### 약어 정리

ASP	Application Service Provider
AWS	Amazon Web Services
BPM	Business Process Management
C/S	Client/Server
CRM	Customer Resource Management
DB	Database
DBMS	Database Management System
ExtJS	Extended JavaScript
FTA	Free Trade Agreement
HRM	Human Resource Management
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
JAXWS	Java API for XML Web Services
MVC	Model-View-Controller
OAASYS	Office as a Service System
PaaS	Platform as a Service
SaaS	Software as a Service
SOQL	Salesforce Object Query Language
SQL	Structured Query Language
SW	Software
VM	Virtual Machine

### 참고 문헌

- [1] Mark Turner, David Budgen, Pearl Brereton, "Turning Software into a Service," *IEEE Comput.*, vol. 36, no. 10, 2003.
- [2] IDC, "Worldwide Software as Service 2010-

- 2014 Forecast: Software Will Never Be the Same,” 2010.
- [3] Gartner, “Hype Cycle for Software as a Service,” 2010.
- [4] Yefim V. Natis, “Reference Architecture for Multitenancy: Enterprise Computing in the Cloud,” Gartner, Dec. 2008.
- [5] 이지현, 허성진, “이해당사자의 관점에서 분류한 SaaS 플랫폼 역량,” 제32회 한국정보처리학회 추계 학술대회 논문집, 제16권 2호, 2009.
- [6] Chang Jie Gui et al., “A Framework for Native Multi-tenancy Application Development and Management,” *CEC-EEE*, Tokyo, Japan, July 2007.
- [7] Nitu, “Configurability in SaaS (Software as a Service) Applications,” *ISEC*, Pune, India, Feb. 2009.
- [8] Cor-Paul Bezemer and Andy Zaidman, “Multi-Tenant SaaS Applications: Maintenance Dream or Nightmare?,” *IWPSE-EVOL*, Antwerp, Belgium, Sept. 2010.
- [9] Craig D Weissman and Steve Bobrowski, “The Design of the Force.com Multitenant Internet Application Development Platform,” *SIGMOD* Providence, Rhode Island, July 2009.
- [10] Amazon Web Services, <http://aws.amazon.com/>
- [11] SQL Azure, <http://msdn.microsoft.com/en-us/library/ee336241.aspx>
- [12] Yefim V. Natis, David W. Cearley, and Eric Knipp, “Salesforce.com at an Inflection Point,” Gartner, Jan. 2011.
- [13] WaveMaker, <http://www.wavemaker.com/>
- [14] 조강래, “Bizmeka 플랫폼 & 서비스의 진화와 발전 방향,” SaaSpia 개발 1차 워크숍, 2009.
- [15] 디지털타임즈, “KT, 기업용SW 오픈마켓 사업 본격화,” 2011.
- [16] OAASYS, <http://blog.oaasys.com>
- [17] 전자신문, “국내 첫 SaaS 포털 오아시스, KT가 접수 비즈메카 2.0으로 재탄생,” 2011.
- [18] 한국무역협회 및 지식경제부, “사례를 통해 배우는 FTA 원산지 길라잡이,” 2009.
- [19] UEngine, <http://www.uengine.com>
- [20] 정문영 외, “프로세스기반 원산지관리시스템,” 제 34회 한국정보처리학회 추계학술대회 논문집, 제 17권 2호, 2010.