SaaSpia 에서 연관 관계를 지원하는 메타데이타

구경이, 정문영, 이상민, 원희선, 허성진
한국전자통신연구원 SW 연구부문 클라우드컴퓨팅연구부 SW 서비스연구팀
e-mail : {kiku, mchung, sangmin2, hswon, sjheo}@etri.re.kr

Metadata to Support the Relationship on SaaSpia Platform

Kyong-I Ku, Moonyoung Chung, Sang-Min Lee, Hee Sun Won, Sungjin Hur
SW Service Research Team, Cloud Computing Research Department, SW Research Laboratory, ETRI

요 약

기존의 SaaSpia 플랫폼에서는 오브젝트간의 연관 관계를 표현하지 않아, 개발자가 SaaSpia 플랫폼 기반 어플리케이션을 개발하는 데 어려움을 겪는다. 본 논문에서는 오브젝트간 연관 관계를 메타데이타로 표현하고, 해당 정보를 이용해 기본 비즈니스 로직 및 사용자 인터페이스(UI)를 동적으로 구성하는 개발 방안에 대해 제안한다. 동적으로 구성된 기본 비즈니스 로직과 UI 는 다시 메타데이타화하여 저장되어 다중 사용자 지원 및 용이한 환경 설정을 가능하게 한다.

1. 서론

SaaSpia 플랫폼은 다양한 어플리케이션을 온라인으로 서비스할 수 있는 SaaS 플랫폼이다[1]. SaaSpia 플랫폼에서는 어플리케이션의 다중 사용자 지원 및 용이한 환경 설정을 가능하게 하기 위해 다양한 정보를 메타데이타로 관리한다[2]. 하지만, 어플리케이션을 구성하는 오브젝트간에 연관 정보를 메타데이타로 제공하지 않아, 테넌트 관리자, 또는 개발자가 연관된 정보를 사용자에게 제공하는데 어려움을 겪는다.

본 논문에서는 SaaSpia 플랫폼에서 오브젝트간의 연관 관계를 표현하는 메타데이타를 연구한다. 이를 이용하면, 개발자와 테넌트 관리자는 관련된 정보를 이용해 동적으로 제공되는 비즈니스 로직 및 UI 을 변경하거나, 설정할 수 있기 때문에 SaaSpia 어플리케이션 사용자는 좀 더 높은 사용 확장성을 보장받는다.

2. 관련연구

일반적인 데이터베이스 시스템에서는 데이터 사전에 엔티티간에 연관 정보를 가지고 있다. Oracle[3]에서는 USER_CONSTRAINTS, USER_CONS_COLUMNS 에 정보를 관리하기에 다음과 같은 SQL 문으로 관계에 관련된 부모, 또는 자식 테이블명, 컬럼명, 삭제 시 무결성 제약조건 등을 알 수 있다.

```
SELECT 'M', COB.TABLE_NAME, CC.COLUMN_NAME, COA.DELETE_RULE
FROM USER_CONSTRAINTS COA,
        USER_CONSTRAINTS COB,
        USER_CONS_COLUMNS CC
WHERE COA.TABLE_NAME=테이블명
      AND COA.CONSTRAINT_TYPE ='R'
      AND COA.R_CONSTRAINT_NAME = COB.CONSTRAINT_NAME
      AND COA.CONSTRAINT_NAME = CC.CONSTRAINT_NAME
UNION
SELECT 'C', COA.TABLE_NAME, CC.COLUMN_NAME, COA.DELETE_RULE
FROM USER_CONSTRAINTS COA,
        USER_CONSTRAINTS COB,
        USER_CONS_COLUMNS CC
WHERE COA.R_CONSTRAINT_NAME=COB.CONSTRAINT_NAME
    AND COB.TABLE_NAME=테이블명
    AND COA.CONSTRAINT_NAME = CC.CONSTRAINT_NAME;
```

하지만, 해당 정보는 데이터베이스내에서 데이타 무결성을 위한 것이며, 개발자가 연관 관계를 파악해서 어플리케이션에 적용하기는 어렵다.

Force.com[4]은 SaaS 기반 플랫폼 관련 선두업체로서 관계형 테이블과 같은 오브젝트 객체들을 생성하고, 오브젝트 및 오브젝트간 관계들을 명세한 meta data 정보를 어플리케이션 개발 도구에 제공한다. 오브젝트간 관계로 lookup, 또는 master-detail 을 선택하도록 하고, 관계를 이용해 오브젝트를 조작할 수 있는 동적 메소드 및 UI 를 설정해준다[5]. 따라서, 개발자가 Force.com 플랫폼상에서 동작하는 어플리케이션을 쉽게 구현할 수 있다.

3. 연관 관계를 지원하는 메타데이타

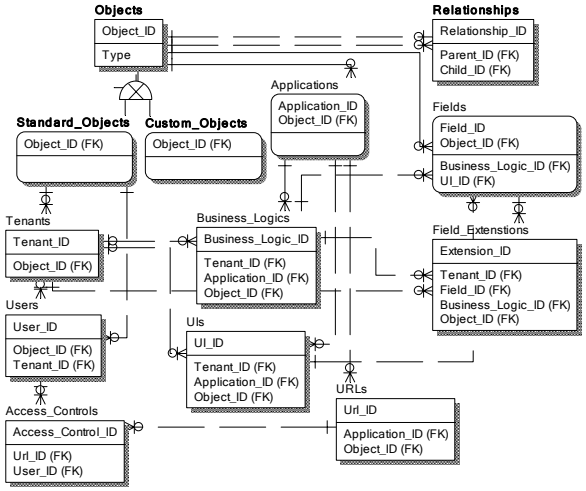
SaaS 플랫폼은 여러 고객이 하나의 서비스 인스턴스를 공유하고, 데이터베이스 안의 정보를 각자 관리할 수 있어 서비스로서 어플리케이션의 비용을 절감할 수 있다. SaaSpia 플랫폼에서 메타데이타는 일반 웹 어플리케이션을 SaaSpia 어플리케이션으로 변환해 다중 사용자 지원을 가능하게 하고, 운용중인 SaaSpia 어플리케이션에서 사용자마다 설정을 변경할 수 있도록 지원해 용이한 환경 설정을 가능하게 한다[2]. 이를 위해 메타데이타 관리기는 어플리케이션의 사용자 UI, 데이타스키마, 비즈니스 로직을 설정하는 설정도구에게 메타데이타를 제공한다. SaaSpia 어플리케이션은 SaaS 플랫폼 속성을 위해 테넌트별 데이터 접근 및 테넌트별 확장 가능한 필드를 제공하도록 구성되어 있다. 하지만, 어플리케이션을 구성하는 오브젝트간에 연관 정보를 메타데이타로 제공하지 않아, 테넌트 관리자, 또는 개발자가 연관된 정보를 사용자에게

본 연구는 지식경제부 및 정보통신연구진흥원의 IT 산업원천기술개발사업의 일환으로 수행하였음. [2009-S-033-01, 중소기업 SW 서비스를 위한 SaaS 플랫폼 개발]

제공하는데 어려움을 겪는다.

본 논문에서는 Force.com 처럼 오브젝트간 연관 관계를 메타데이타로 지원한다. 또한, 오브젝트에서 기본 제공 오브젝트(테넌트 정보, 사용자 정보 등)들과 개발자가 추가한 오브젝트간에 관계 설정도 제공한다. 따라서, 연관 관계를 이용해 개발자가 SaaSpia 에서 어플리케이션을 구현하거나, 테넌트 관리자가 어플리케이션을 설정 시, 해당 정보를 이용해 설정함으로써 SaaSpia 어플리케이션 사용자는 좀 더 높은 사용 확장성을 보장받을 수 있다.

SaaSpia 플랫폼 기반 어플리케이션 및 오브젝트간 연관 관계를 표현하는 메타데이타를 도시한 것은 (그림 1)과 같다.



(그림 1) SaaSpia 플랫폼내 어플리케이션 및 연관 관계를 표현하는 메타데이타의 예

본 논문에서 제안된 오브젝트간 연관 관계는 다음과 같은 정보를 메타데이타로서 제공한다.

<표 1> 오브젝트간 연관 관계

Relationship_ID	Relationships 테이블내 식별자
Parent_ID	부모 Object_id
Child_ID	자식 Object_id
Relationship_Type	식별 관계(식별 or 비식별)
Parent_Input	Parent 입력 시, 무결성 조건
Parent_Update	Parent 수정 시, 무결성 조건
Parent_Delete	Parent 삭제 시, 무결성 조건
Child_Input	Child 입력 시, 무결성 조건
Child_Update	Child 수정 시, 무결성 조건
Child_Delete	Child 삭제 시, 무결성 조건

Java 의 Hibernate, Rails 의 ActiveRecord 와 같은 객체-관계 매핑 프레임워크(ORM)는 관련 데이터에 대해 테이블의 데이터를 조작(생성, 삭제, 검색, 갱신)할 수 있는 동적 메소드들을 제공하고, 해당 데이터와 메소드들을 브라우저에 표시할 부분도 자동 생성해 준다 [6,7]. SaaSpia 플랫폼에서도 본 논문에서 제안하는 오브젝트간 연관 관계를 이용해 관련 비즈니스 로직과 UI 를 동적으로 생성할 수 있다. 즉, 어플리케이션에 포함되는 각각의 오브젝트를 클래스화하고, 식별 관계는 상속으로, 비식별 관계는 해당 클래스의 멤버변수로 설정한 후에 DB 의 무결성을 보존하면서 동적메소드를 생성한다. 또한 UI 템플릿 등을 이용해 브라우저 표시할 부분을 생성하는 로직을 생성한다. 하지만, 기존의 ORM 프레임워크들와 달리, 생성된 동적 메소드와 UI 들은 다시 관련 메타데이타로 변환되어 데이타베이스에 저장된다. SaaSpia 플랫폼에서 어플리케이션 및 오브젝트간 연관 관계를 표현하는 메타데이타를 이용한 어플리케이션의 개발은 다음과 같이 진행된다.

1. 어플리케이션에서 요구사항을 분석한다.
2. 필요한 정보를 오브젝트화한다.
3. 객체에 대해 기본 비즈니스 로직과 UI 를 제공한다(개별 레코드 자세한 정보 읽기, 간략화된 전체 레코드 정보 보기, 삽입, 삭제, 갱신과 같은 비즈니스 로직).
4. 오브젝트간 관계를 표현한다.
5. 연관 관계간 무결성을 해치지 않는 기본 비즈니스 로직과 UI 를 구성해 제공한다(연관된 정보를 볼 수 있는 링크들을 제공).
6. 해당 내용들을 메타데이타화하여 저장한다.

메타데이타를 기반으로 개발된 SaaSpia 플랫폼 기반 어플리케이션은 다중 사용자 지원 및 용이한 환경 설정이 가능하다. 따라서 필요하다면, 테넌트 관리자는 제공된 비즈니스 로직과 UI 를 설정환경에서 재설정해, 터넌트 사용자에게 제공할 수 있다. 어플리케이션 사용자가 브라우저를 통해 어플리케이션의 실행을 요청하면, SaaSpia 플랫폼 실행 엔진에서는 저장된 오브젝트, 비즈니스 로직, UI 등의 메타데이타를 이용해 어플리케이션을 재구성하고 해당 내용을 사용자 브라우저에 전달한다.

4. 연관 관계를 지원하는 메타데이타

본 논문에서는 SaaSpia 플랫폼에서 어플리케이션을 구성하는 오브젝트 및 연관 관계를 메타데이타로 관리하고 해당 데이타를 이용해 어플리케이션을 개발하는 방안에 대해 제안하였다. 제안된 방법은 다중 사용자 지원을 가능한 SaaSpia 어플리케이션을 쉽게 개발하고 운용할 수 있도록 한다. 현재 SaaSpia 에서는 오브젝트가 어플리케이션에 종속되면서, 유연한 공유가 어렵다. 유연한 공유가 될 수 있도록 SaaSpia 에서 메타데이타를 유지한다면, 어플리케이션 사용자는 좀 더 높은 환경 설정성을 보장받을 수 있을 것이다.

참고문헌

[1] 중소기업 SW 서비스를 위한 SaaS 플랫폼 기술 개발 요구사항정의서 0, 한국전자통신연구원, 2010
[2] 구경이, 이상민, 원희선, 허성진, “SaaSpia 플랫폼에서 메타데이타의 지원,” 한국정보처리학회추계학술대회논문집, KIPS, Vol. 17, No. 2, 2010
[3] http://www.oracle.com
[4] http://www.force.com
[5] Craig Weissman, Development with the Force.com Platform, Jason Ouellette, 2010
[6] http://www.hibernate.org/
[7] http://rubyonrails.org/