

# 접근제어 기법을 이용한 주소록, 일정 관리 시스템의 설계 및 구현

서경진\*, 이은정\*\*

\*, \*\*경기대학교 컴퓨터학과

e-mail:logen0104@gmail.com\*, ejlee@kyonggi.ac.kr\*\*

## Design and Development Address & Schedule Management System Using Access Control Technique

Kyong-Jin Seo\*, Eun-Jung Lee\*\*

\*Dept of Computer Science, Kyonggi University

### 요 약

접근제어는 권한이 있는 사용자에게 데이터에 접근할 수 있도록 제한하는 것을 말한다. 본 논문에서는 웹 주소록과 일정관리 기능을 여러 사용자가 공유할 수 있게 하기 위해서 접근제어 기법을 적용한 웹사이트를 개발하였다. 접근권한을 기술하기 위하여 그룹의 개념을 도입하여 그룹일정에 대해서 공유범위를 설정할 수 있게 하였다. 한편 주소록과 일정관리 데이터를 vCard와 vCalendar 등의 마이크로포맷 표준을 도입 하였다. 그리고 최근 웹 2.0 기술로 주목받고 있는 루비 온 레일스(Ruby on Rails) 프레임워크를 사용하여 구현 하였다.

### 1. 서 론

최근 웹상에서 여러 사용자들이 공동으로 데이터를 사용하고 접근하는 시스템이나 서비스가 많아지면서 접근제어가 주요한 이슈로 주목받고 있다. 접근 제어는 권한이 있는 사용자에게 허용되는 데이터만 접근할 수 있도록 제한하는 시스템의 기능으로 접근규칙을 통해 기술된 허용조건을 기술한다. 접근제어 분야에서 웹에서의 접근 제어 방법과 XML데이터에 대한 접근 제어 방법 등의 연구되었다[5,6].

본 논문에서는 웹 주소록과 일정관리 기능을 여러 사용자가 공유할 수 있게 하기 위해서 접근제어 기법을 적용한 웹사이트를 개발하였다. 접근권한을 기술하기 위하여 그룹의 개념을 도입하여 그룹일정에 대해서 공유범위를 설정할 수 있게 하였다. 한편 주소록과 일정관리 데이터를 vCard와 vCalendar 등의 마이크로포맷 표준을 도입하였다. 그리고 최근 웹 2.0기술로 주목받고 있는 루비 온 레일스(Ruby on Rails)프레임워크를 사용하여 구현 하였다.

주소록과 일정관리 기능을 위한 표준으로 마이크로포맷의 vCard와 vCalendar는 microformats.org가 제창하는 XML 포맷으로 XHTML에 메타데이터를 넣어 콘텐츠 정보를 보다 상세하게 구조화 하려는 것이다.

마이크로포맷은 간단한 XHTML, XML을 이용하여 블로그 기사를 구조화 함으로써 사람에게나 프로그램에게나 보기 쉬운 것을 만들고 있다. 그 시도의 가장 첫 단계가 사람과 조직에 관해 기술하는 포맷의 vCard, 달력과 이벤

트에 관한 vCalendar이다.

최근 빠른 웹 사이트의 개발을 위하여 프레임워크를 이용한 개발 방법이 광범위하게 사용되고 있는데 그 중에서 루비 온 레일스(Ruby on Rails) 프레임워크는 2004년 7월에 발매되었으며 덴마크의 데이빗 하이네마이어 한슨(David Heinemeier Hansson)이 오브젝트 지향 스크립트 언어인 루비(Ruby)를 사용해 개발한 오픈 소스 웹 어플리케이션 프레임워크이다.

이 루비 온 레일스의 큰 특징은 코드의 생산성이 매우 높다는 점이다. 이름을 정하는 규칙에 따라 변수 이름 등을 사용하면 표준적인 데이터베이스로의 액세스 프로그램이나 검색 프로그램, 업데이트화면 등을 자동적으로 생성할 수 있다. 단순한 데이터베이스 액세스만을 위한 웹 어플리케이션의 경우 프로그램을 짜지 않고도 자동적으로 소스 코드를 생성할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 본 논문의 접근제어 모델 설명을 위해 접근제어 규칙, 사용자 설정을 통한 필터링, 데이터 모델을 정의한다. 3장에서 설계를 위한 시스템 구성도, 액티브 레코드 모델, 컨트롤러, 뷰를 설명한다. 4장에서 설계한 내용을 토대로한 결과를 보인 후, 5장에서 결론 및 향후 방안에 대해 논의 하도록 한다.

## 2. 제안된 접근제어 모델

### 2.1 접근제어 규칙

사용자의 집합을  $U$ , 그룹의 집합을  $G$ , 레코드의 집합을  $R$ 이라고 하자.

$$U = \{u_1, u_2, \dots, u_n\}$$

$$G = \{g_1, g_2, \dots, g_m\}$$

$$R = \{r_1, r_2, \dots, r_p\}$$

집합  $A(u)$ 는 사용자  $u$ 에 허용되는 접근권한 이라고 한다면 다음과 같이 표현할 수 있다.

for  $r \in R, u \in U, r \in A(u)$  if

$$i) \text{level}(r) = \text{public}$$

$$ii) \text{level}(r) = \text{group and } \text{group}(r) \cap \text{group}(u) = \emptyset$$

$$iii) \text{level}(r) = \text{private and } \text{author}(r) = u$$

### 2.2 사용자 설정을 통한 필터링

본 논문에서는 웹상의 공유 데이터를 사용자에게 제공하는 방법으로 권한 규칙을 통한 접근제어 단계와 사용자 설정을 통한 필터링의 단계로 나누어 페이지를 생성한다. 권한 규칙은 그 사용자에게 허용되는 데이터 집합을 구하는 것이라면 사용자 관점에서 그 데이터 중에서 관심을 가지는 데이터 집합을 다시 필터링하는 것이 가능하다. 본 시스템에서는 데이터 의존적인 필터링 조건을 사용자가 기술할 수 있게 설정화면을 제공한다.

사용자  $u$ 에 대한 필터링 규칙들이 정의된 경우 이를 통한 필터링은 다음과 같이 표현할 수 있다.

$F(u)$  : 사용자  $u$ 에 대한 필터링 규칙의 집합, 규칙들을 모두 만족하여야 필터링 결과에 속한다.

$$\text{filtered}(u) = \{r | r \in A(u), f(r) = \text{true for all } f \in F\}$$

### 2.3 데이터 모델

본 논문에서 사용한 vCard 데이터 모델은 다음과 같다.

<표 1> vCard 데이터 모델

vCard 레코드 이름	설명
VERSION	vCard명세의 버전
REV	vCard의 생성일자를 지정
PROPID	자신만의 고유한 이름(식별자)
FN	사람의 성과 이름

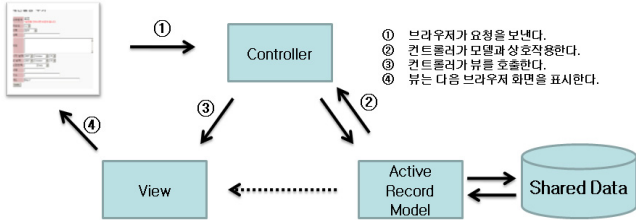
Family name	성
Given name	이름
Additional name	추가적인 이름
Honorific prefix	경칭 접두사
Honorific suffix	경칭 접미사
NICKNAME	별명
TITLE	업무 직함을 지정
ROLE	업무 범주 혹은 직위
BDAY	생일
ADR	주소 (반복부)
Post office box	사서함
Extended addr.	확장된 주소
Street addre.	거리 주소
City, town	시
Statem province	도
Postal code	우편번호
Country	국가
<b>TYPE</b>	
dom	국내 주소
intl	국제 주소
postal	우편 주소
parcel	소포 수취 주소
home	거주지 주소
work	직장 주소
<b>TEL</b>	전화번호 (반복부)
<b>TYPE</b>	
home	집 전화번호
msg	메시지 저장이 되는 전화번호
work	직장 전화번호
voice	음성 전화번호
fax	팩스번호
cell	이동 전화번호
video	화상 전화번호
pager	호출기 번호
bbs	음성 사서함 번호
modem	모뎀이 연결되어 있는 번호
car	차량 전화번호
isdn	ISDN번호
pcs	개인통신 서비스 번호
<b>EMAIL</b>	이메일 (반복부)
<b>TYPE</b>	
SMTP	인터넷 메일
X400	
URL	개인홈페이지
TZ	vCard주체의 지리적 위치
CATEGORIES	조직에관한 정보
NOTE	보충 정보와 일반적인 주석

## 3. 설 계

본 논문에서 개발된 웹사이트는 루비 온 레일스 프레임워크를 따르므로 모델 - 뷰 - 컨트롤러라는 세 부분으로 나누어 시스템을 구성하였다. 아래에서는 각 부분에 대하여 시스템 설계와 개발 방법을 살펴본다.

### 3.1 시스템 구성도

Ruby on Rails는 기본적으로 MVC프레임워크를 가진다. 레일스는 애플리케이션 구조를 정리해서, 프로그램을 모델, 뷰, 컨트롤러로 나누고 각각의 기능 덩어리들을 따로따로 개발한다.

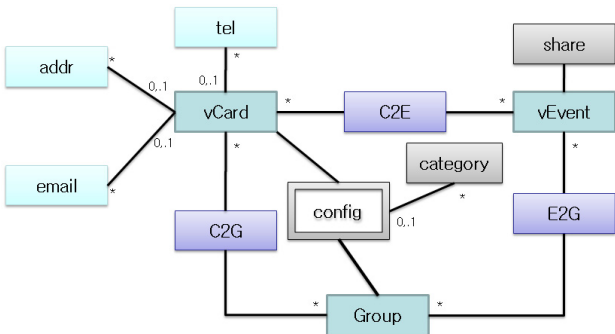


(그림 1) 모델-뷰-컨트롤러 아키텍처

### 3.2 액티브 레코드 모델

액티브 레코드는 데이터베이스 스키마를 정의하고 데이터베이스 질의를 담당하는 코드 부분으로 데이터베이스 테이블이 하나의 클래스에 대응되고 질의는 그 클래스의 멤버함수로 표현된다.

본 논문에서 사용된 시스템 모델은 그림 2와 같다. 여기서 vEvent는 vCalendar의 데이터 일종으로 일정 정보를 가지며, vCard는 이름, 전화번호, 주소 등 사용자 정보를 가지며, 그룹은 접근제어를 위한 사용자 집합을 정의한다.



(그림 2) 시스템 모델

### 3.3 컨트롤러

사용자가 웹 서버에 요청을 보내면 컨트롤러의 액션 함수가 호출된다 이것은 사용자의 요청을 처리하는 코드를 가지며, 모델에서 필요한 데이터를 추출하고 뷰를 통해 사용자에게 내려보낼 페이지 또는 뷰를 가공한다. 본 시스템에서는 다음 아래와 같은 컨트롤러들이 있다.

<표 2> 컨트롤러 이름과 역할

컨트롤러	역할
main_controller	<ul style="list-style-type: none"> <li>로그인.</li> </ul>
vCard_controller	<ul style="list-style-type: none"> <li>vCard 추가, 조회, 수정, 삭제.</li> <li>addr, tel, email 추가, 조회, 수정, 삭제.</li> <li>addr, tel, email 폼 렌더링.</li> </ul>

vEvent controller	<ul style="list-style-type: none"> <li>vEvent 추가, 조회, 수정, 삭제.</li> <li>일정 공유 범위 설정.</li> <li>공유범위에 맞는 일정 필터링 계산</li> </ul>
Group_controller	<ul style="list-style-type: none"> <li>Group 추가, 조회, 수정, 삭제.</li> <li>그룹의 멤버 참여.</li> </ul>
config_controller	<ul style="list-style-type: none"> <li>일정 설정 추가, 수정, 삭제</li> <li>사용자 지정 일정 필터링 계산.</li> </ul>

여기서 접근제어에 참여하는 컨트롤러는 vEvent 컨트롤러와 config 컨트롤러로 vEvent 컨트롤러에서 일정 공유 범위 설정과 접근 권한 계산하고, config 컨트롤러에서 사용자 관심 필터링이 계산된다.

### 3.4 뷰

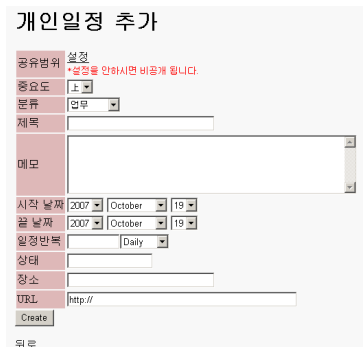
뷰는 보통 모델에 있는 데이터를 기반으로 사용자 인터페이스의 생성을 책임진다. 예를 들어 일정의 목록을 보고 싶을 때, 리스트는 모델을 통해서 접근할 수 있지만, 모델에서 해당 리스트 목록에 접근 하고, 그걸 최종 사용자에게 맞게 형식화 하는 것이 뷰이다. 그룹 리스트를 보여주는 루비 코드는 다음과 같다.

```
<%= link_to '뒤로', :action => 'show', :controller=> 'vcards', :id => @vcard%>
```

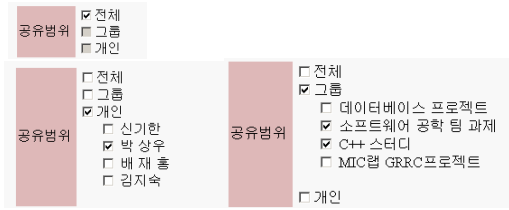
(그림 3) 일정 리스트 코드

## 4. 구현 결과

본 논문에서 구현한 접근제어 기법을 이용한 주소록, 일정관리 시스템의 결과는 아래의 그림과 같다. 그림 4는 일정을 추가하는 화면 이다. 여기에서 그림 5와 같이 일정을 공유할 사용자 범위를 선택할 수 있다.

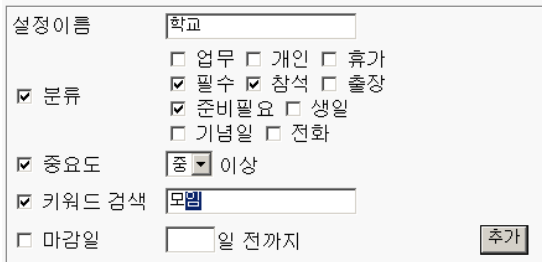


(그림 4) 일정 추가 화면

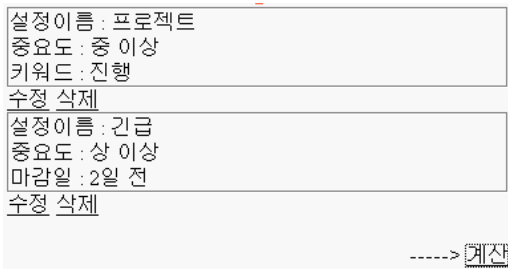


(그림 5) 일정 공유 범위 설정

사용자 관심 일정을 필터링하기 위해서 (그림 6)과 같은 화면을 통하여 사용자가 설정을 추가할 수 있으며, 추가된 관심 목록(그림 7)들이 그림 8과 같이 필터링 되는 것을 볼 수 있다.



(그림 6) 사용자 관심 일정 추가화면



(그림 7) 사용자 관심 일정 목록



(그림 8) 사용자 관심 일정 필터링 결과

## 5. 결론 및 향후 연구 방안

본 논문에서는 접근제어 기법을 이용한 주소록, 일정관리 시스템을 구현하였다. 웹 주소록과 일정관리 기능을 여러 사용자가 공유할 수 있게 하기 위해서 접근제어 기법을 적용한 웹사이트를 개발하였다. 데이터를 표현하기 위하여 vCard와 vCalendar와 같은 마이크로포맷 표준을 도입하였고, 개발 방법에 있어서 최근 주목 받고 있는 루비온레일스(Ruby on Rails) 프레임워크를 사용하여 구현하였다.

본 논문에서는 접근 권한에 따른 일정 공유와 사용자 관심 일정 필터링 계산을 할 수 있는 웹 사이트를 구현하였다. 앞으로 연구에는 XPath형태로 기술된 접근 규칙을 이용하여 다양한 측면에서의 접근 제어를 할 수 있는 연구가 진행될 것이다. 또한 웹 사이트 뿐만 아니라 모바일 환경에서 유용하게 쓸 수 있는 연구가 진행될 것이다.

## 참고문헌

- [1] David Thomas, "Agile Web Development with Rails 2nd Ed. Pragmatic Bookshelf
- [2] David Thomas, "Programming Ruby: The Pragmatic Programmers' Guide" 2nd Ed. Pragmatic Bookshelf
- [3] Chad Fowler "Rails Recipes" Pragmatic Bookshelf
- [4] Brian Suda "Using Microformats" O'REILLY
- [5] J.M. Jeon, Y.D. Chung, M.H. Kim, Y.J. Lee, "Filtering Xpath expressions for XML access control", Computer and Security Vol. 23, P.591 - 605, 2004
- [6] Naizhen Qi, Michinharu Kudo, Jussi Myllymaki, Hamid Pirahesh "A function-based access control model for XML databases". Proc. of the 14th ACM International conference on Information and knowledge management pp 115 - 122. 2005