Р.Р. Кеуно

Hierarchical Permissions Model in the Declarative Web-Application Modeling Language in Context of Blockset Methodology

Keywords: web, access permissions, CRUD, BlockSet, BML.
The article describes the access permissions method in BlockSet methodology. The main feature of technique is universalization. On the one hand, access permissions are divided between individual users and groups with their priorities. On the other hand, access may have a different scope and can be inherited.

П.П. Кейно

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ПРАВ ДОСТУПА В ДЕКЛАРАТИВНОМ ЯЗЫКЕ МОДЕЛИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ ПО МЕТОДОЛОГИИ BLOCKSET

Повышение эффективности разработки программного обеспечения – одна из фундаментальных задач отрасли информационных технологий. Решение данной задачи является комплексным и может быть достигнуто с помощью поиска новой методологии разработки и проектирования. Первый аспект научного поиска методологии – сильная теоретическая база, грамотное выделение основных сущностей, с которыми будет происходить работа. Второй – предоставление эффективного инструментария, способного корректно работать с этими сущностями, что является практической составляющей работы.

Представленная методология BlockSet призвана повысить эффективность разработки веб-приложений. Основное свойство методологии сводится к соблюдению принципа доступности, заключающегося в переходе от простейших понятий до комплексных составляющих, и сохранению при этом гибкости и мобильности используемых технологий. Исходя из этого тезиса, заметим, что разработка простейшего проекта не требует профессиональных знаний, при этом данное обстоятельство минимально затрагивает гибкость, требующуюся при создании сложных проектов, где важным критерием является максимальный уровень кастомизации.

Так, методология оперирует ограниченным количеством сущностей с широким набором свойств. Базовые сущности – это блок и набор. Блок представляет собой отдельную ячейку данных определённого типа и является атомарным, но модулируемым элементом. Исходя из этого, несмотря на то, что блок принимает только конкретные типы данных, их поведение при обработке может настраиваться. Другой элемент, получивший название «набор», предназначен для объединения блоков, хранящих логически связанные данные. С некоторыми допущениями можно сказать, что блок – поле в базе данных, а набор – таблица. Разница в том, что эти элементы представляют собой высокоабстрактные сущности с широким набором свойств-модификаторов. Тем не менее наборы, как и таблицы, могут быть в отношениях друг с другом.

Крупный проект может содержать большое количество блоков и наборов, выполняющих самые различные функции. Наборы могут хранить статьи, комментарии, структуру веб-ресурса, пользователей и многое другое. Проекту подобного масштаба требуется функционал, разграничивающий доступ к тем или иным наборам для различных групп пользователей. Таким образом, разграничение прав доступа – неотъемлемая и одна из самых важных частей методологии.

Практическая реализация методологии представляет собой декларативный язык моделирования BML (*англ. BlockSetModelingLanguage*), задающий логику работы серверной стороны. Язык выполнен на основе XML-подобного синтаксиса, а все сущности методологии задаются с помощью своих свойств и элементов [1].

Виды прав доступа и их область действия

В проекте внедрены права доступа следующего типа: создание, чтение, обновление и удаление (англ. CRUD, Create, Read, Update, Delete). Такое разделение положительно зарекомендовало себя во многих программных решениях, включая СУБД [2], REST-архитектуры [3], моделирование интернет-вещей [4] и др. Не стала исключением и методология BlockSet, перенявшая опыт других программных систем. Окончательным аргументом в сторону использования CRUD стало её использование в смежных проектах [5].

В методологии BlockSet права доступа устанавливаются на экземпляры набора, то есть на данные, содержащиеся в этих наборах [1]. Они хранятся в специализированном блоке типа «permissions» («разрешения»). Этот блок хранит информацию о группах (о них – ниже) и пользователях с указанием их прав. Причём индивидуальные права пользователя имеют более высокий приоритет, чем группы, в которых состоит данный пользователь.

Область действия прав доступа может быть как на экземпляры текущего набора, так и на его дочерние наборы. Для этого существует несколько режимов, устанавливаемых атрибутом «scope»: self – устанавливает права доступа на самого себя, branch – на все потомки набора, child – только на прямых потомков. Допустимо комбинировать эти режимы, указывая их в атрибуте через пробел (безусловно, использование «child» вместе с «branch» не имеет смысла).

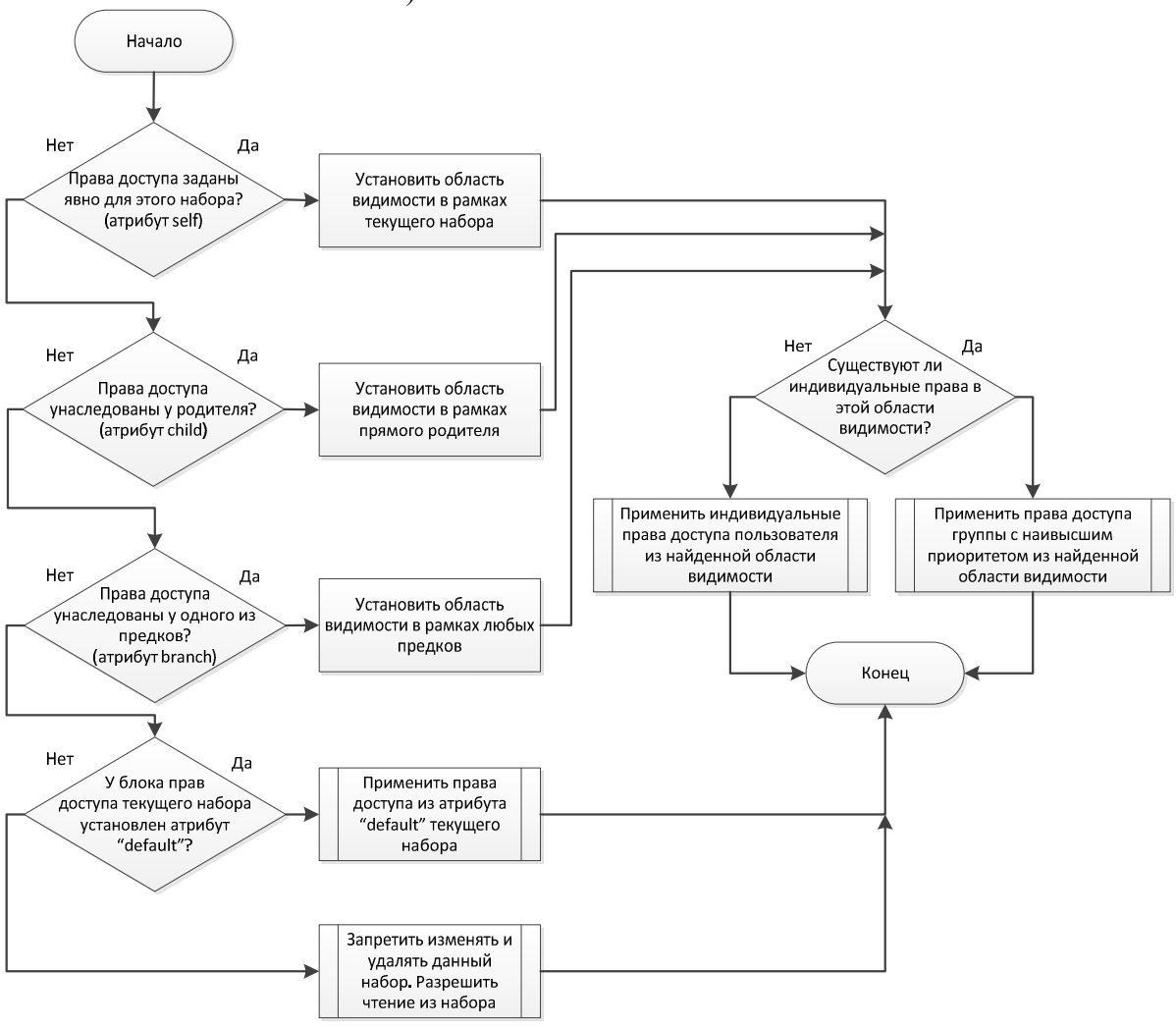

Рис. Алгоритм установки прав доступа к экземпляру набора

В случае если в том или ином наборе для пользователя или его групп нет информации о разрешениях, устанавливаются права, указанные в атрибуте «default» блока «permissions». Но если атрибут не установлен, пользователь получает права только на чтение содержимого набора. Важно отметить, что заданный в явном виде, но пустой атрибут «default» позволяет снять и эти права. Общая схема алгоритма нахождения прав доступа представлена на рисунке.

Авторизационный набор

В сфере веб-технологий используется множество средств проверки подлинности, но как самый простой метод наиболее распространена аутентификация путём ввода посетителем ресурса комбинации логин/пароль. Пользователь вводит данную комбинацию в специализированную форму, после чего отправляется запрос на сервер, где происходит проверка этой комбинации и сопоставление её с информацией, хранящейся на сервере. Если такое соответствие удаётся найти, сервер отправляет клиенту специализированный ключ, который клиент обязан хранить до истечения срока его действия. Таким образом, переходя от страницы к странице одного и того же ресурса, пользователю не требуется каждый раз вводить свой логин и пароль. Вместо этого клиентское приложение автоматически отправляет полученный ранее ключ, каждый раз подтверждая, таким образом, подлинность авторизованного пользователя. Полученный ключ, как правило, хранится в HTTPCookie [6].

В методологии BlockSet существует специализированный набор, выполняющий перечисленные выше функции. Такой набор получил наименование авторизационного (*англ. Authorization Set*). Авторизационный набор, в силу своей специфичной и самодостаточной функции, в проекте может быть только один, поэтому для его объявления используется уникальное зарезервированное имя «*_users»*. Авторизационный набор содержит список пользователей, их пароли в зашифрованном виде, а также дополнительные данные по усмотрению разработчика проекта. В функции авторизационного набора входят получение от пользователя пары логин/пароль, проверка этой пары с одним из экземпляров набора и выдача ключа сессии клиенту в случае успешной авторизации.

В авторизационном наборе должны присутствовать как минимум два блока. Один блок – для хранения пользовательского логина – имеет зарезервированное имя «*_login*» и может быть любого строкового типа: «*string»*, «*email*», либо «*phone*». Во втором блоке хранится пользовательский пароль. Этот блок имеет зарезервированное имя «*_password*» и тип «*password*» (без нижнего подчёркивания).

Авторизационный набор хранит информацию о пользователях, поэтому допустимо добавлять и другие блоки, относящиеся к пользовательской информации: дата рождения, личные предпочтения, место работы и др. Хорошей практикой считается объявлять блоки, сохраняющие дату и IP-адрес регистрации пользователя и те же данные о его последней авторизации.

Группы и набор групп

В методологии BlockSet права доступа к определённым частям проекта назначаются для групп и пользователей. Любой пользователь может состоять в неограниченном числе групп. Группы предназначены для определения уровня доступа того или иного пользователя. Для реализации подобного функционала внутри авторизационного набора необходимо объявить другой специализированный набор – набор групп. Данный набор использует зарезервированное имя «*_groups*». Для того чтобы реализовать возможность добавлять множество пользователей во множество групп, необходимо

установить для набора групп связь «многие ко многим» с родительским (здесь – всегда авторизационным) набором (атрибут *relation="multi"*).

Название группы представляет собой последовательность символов латиницы и цифр. Название группы определяется зарезервированным именем «*_groupname*» блока строкового типа. Группы создаются и удаляются администратором системы. Все группы имеют собственный приоритет, выраженный в числовой форме. Приоритет указывается в блоке типа «*number*» с зарезервированным именем «*_priority*». Права доступа у пользователя, являющегося членом нескольких групп, определяются группой с наивысшим приоритетом. Групп с одинаковым приоритетом существовать не может.

Пользователь, как правило, состоит в группах, созданных вручную администратором системы. Им же и присваиваются группы определённым пользователям. Однако, помимо стандартных групп, созданных администратором, существуют группы, присваиваемые пользователям в зависимости от тех или иных обстоятельств, внешних факторов. Эти группы получили название событийных. Такие группы создаются автоматически при объявлении набора групп, их удалить нельзя, но допустимо менять приоритет.

Кроме того, системой автоматически создаётся и ряд стандартных групп, рекомендуемый к использованию для разграничения прав доступа. Эти группы ничем не отличаются от тех, которые могут быть созданы администратором вручную: администратор точно также может менять их приоритет, название, а также удалять. В таблице представлен перечень всех групп, создаваемый при объявлении соответствующего набора, их начальные приоритеты и права доступа.

Таблица. Перечень групп, создаваемых при инициализации группового набора с указанием начальных приоритетов и прав доступа

Имя группы в системе	Название	Права доступа	Приоритет	Описание
admin	Администратор	CRUD	1000	Стандартные группы, создаются автоматически
supermod	Супермодератор	CRUD	900	
moder	Модератор	CRUD	800	
banned	Заблокированный	----	700	
owner	Владелец	-RU-	600	Устанавливает разрешения для владельца (создателя) экземпляра набора
user	Пользователь	-R--	0	Любой авторизованный в системе пользователь
search	Поисковый робот	-R--	–600	Устанавливается всем поисковым машинам, индексирующим ресурс
guest	Гость	-R--	–700	Любой неавторизованный пользователь
all	Все	-R--	–1000	Любой пользователь

Пример проекта с комплексным разграничением прав доступа

В листинге представлен проект веб-приложения «Диссертационные советы» в виде описания динамических структур на языке BML с разграничением прав доступа. Целью гипотетического проекта является организация документооборота между соиска-

телями учёных степеней и диссертационными советами, коих может быть несколько. Названия сущностей указаны в атрибуте «name». Далее по тексту в скобках указаны названия сущностей из листинга.

```
<model>
      <setname="speciality">
            <block-
name="global_permissions"type="permissions"scope="branch"/>
            <blockname="cipher"type="string"/>
            <blockname="name"type="string"/>
            <blockname="description"type="string"/>
            <setname="council"relation="multi">
                  <block-
name="council_permissions"type="permissions"scope="branch"/>
                  <blockname="org"type="string"/>
                  <blockname="city"type="string"/>
                  <blockname="cipher"type="string"/>
                  <setname="thesis"relation="parent">
                        <block-
name="thesis_permissions"type="permissions"default="cr"scope="branch"/>
                        <blockname="subject"type="string"/>
                        <blockname="author"type="string"/>
                        <blockname="supervisor"type="string"/>
                        <blockname="org"type="string"/>
                        <blockname="text"type="string"/>
                        <blockname="pubdate"type="datetime"/>
                        <blockname="status"type="number"/>
                        <setname="related_docs"relation="multi">
                              <blockname="pubdate"type="datetime"/>
                              <blockname="name"type="string"/>
                              <blockname="description"type="string"/>
                              <block-
name="attachment"type="file"ext="pdf"/>
                        </set>
                        <setname="branch"relation="multi">
                              <blockname="name"type="string"/>
                        </set>
                  </set>
            </set>
```

Листинг. Пример проекта «Диссертационные советы»

Модель данных представляет собой базу диссертаций (thesis). Каждой диссертации соответствует определённая специальность (speciality) и определённый диссертационный совет (council), а также пакет сопутствующих документов (related_docs). Диссертация может соответствовать нескольким отраслям специальностей (branch). Статус работы (status) может быть следующим: подано на рассмотрение, рассматривается, принято к защите, защищено. Добавление конкретных пользователей и групп в тот или иной блок прав доступа осуществляется с помощью визуального редактора, а на языке BML описана лишь структура прав доступа.

Политики прав доступа проекта следующие:

1) изменять, удалять, создавать новые шифры специальностей могут только администраторы (указано в global_permissions). Область действия («branch») распространяется на весь проект;

2) добавлять диссертационную работу и сопутствующие документы (дочерняя сущность, права действия распространяются, согласно значению «branch») может любой пользователь (указано в thesis_permissions, а также в атрибуте «default»);

3) после публикации работы изменять информацию о ней пользователь может только после установки соответствующих прав уполномоченным в данном конкретном диссертационном совете (указано в council_permissions);

4) изменять статус (status) может только уполномоченный в данном конкретном диссертационном совете (council), куда была подана работа (указано также в council_permissions).

Заключение

В рамках практической реализации методологии BlockSet и языка BML была использована гибкая система разграничения прав доступа с разделением на пользователей и группы по приоритетам, с одной стороны, и широким разделением области действия, с другой. Разработанная система полностью соответствует проверенной временем методике CRUD. Для авторизации пользователей и добавления их в группы были введены специализированные сущности: авторизационный набор и набор групп. В примере реализации проекта наглядно показаны гибкость разработки и универсальность технологии.

СПИСОК ЛИТЕРАТУРЫ:

1. Кейно П.П., Силуянов А.В. Декларативный подход описания динамическихWeb-узлов // Информационные системы и технологии. 2015. Т. 89. №. 3.
2. Truica C. O. et al. Performance evaluation for CRUD operations in asynchronously replicated document oriented database //Control Systems and Computer Science (CSCS), 2015 20th International Conference on. IEEE, 2015. Pp. 191–196.
3. Esteller-Curto R. et al. Proposal of a REST-based architecture server to control a robot //Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on. IEEE, 2012. Pp. 708–710.
4. Alam S., Chowdhury M. M. R., Noll J. Senaas: An event-driven sensor virtualization approach for internet of things cloud //Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on. IEEE, 2010. Pp. 1–6.
5. Cadavid J. J. et al. A Domain Specific Language to Generate Web Applications //CIbSE. 2009. Pp. 139–144.
6. Squier A. C., Austin C. J. Universal session sharing :пат. 7188181 США. 2007.

REFERENCES:

1. Keyno P.P., Silujanov A.V. Deklarativnyj podhod opisanija dinamicheskih Web-uzlov // Informacionnye sistemy i tehnologii. 2015. V. 89. №. 3.
2. Truica C. O. et al. Performance evaluation for CRUD operations in asynchronously replicated document oriented database //Control Systems and Computer Science (CSCS), 2015 20th International Conference on. IEEE, 2015. Pp. 191–196.
3. Esteller-Curto R. et al. Proposal of a REST-based architecture server to control a robot //Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on. IEEE, 2012. Pp. 708–710.
4. Alam S., Chowdhury M. M. R., Noll J. Senaas: An event-driven sensor virtualization approach for internet of things cloud //Networked Embedded Systems for Enterprise Applications (NESEA), 2010 IEEE International Conference on. IEEE, 2010. Pp. 1–6.
5. Cadavid J. J. et al. A Domain Specific Language to Generate Web Applications //CIbSE. 2009. Pp. 139–144.
6. Squier A. C., Austin C. J. Universal session sharing :пат. 7188181 США. 2007.