Подсистема управления доступом

Подсистема управления доступом (ПУД) CM4/5 обеспечивает разграничение доступа пользователей к доменным объектам в системе. ПУД реализует дискреционный принцип доступа. Пользователь получает доступ к конкретному объекту только в том случае, если существует правило, предоставляющее пользователю разрешение на доступ данного типа к этому объекту. Помимо разграничения доступа к доменным объектам, ПУД позволяет ограничивать доступ к общесистемным операциям (наподобие создания доменного объекта). Для обобщения подходов объектом доступа в этом случае считается система, а типом доступа – выполнение соответствующей операции.

Правила выдачи разрешений (матрица доступа) определяются в конфигурации системы. Для общесистемных (глобальных) операций всё просто: правила определяют группы пользователей, которые получат разрешение на выполнение данной операции. О группах речь пойдёт ниже, пока только необходимо подчеркнуть, что разрешения выдаются не отдельным пользователям, а группам.

Для контекстных операций вводится понятие контекстной роли. Под ней мы понимаем набор групп (опять групп!) пользователей, имеющих отношение к данному объекту. При этом отношение может быть совершенно любым, способ его определения отдаётся на откуп настройщику или разработчику: из содержимого полей данного доменного объекта или любых связанных с ним, с помощью SQL-запроса или даже собственного Java-класса. Как бы это ни происходило, контекстная роль, получая параметром доменный объект (контекст), раскрывается в набор групп, которым могут быть предоставлены права на тот или иной тип доступа к этому объекту. Таким образом, доступ к объекту может зависеть от его содержимого.

Теперь, наконец, о группах пользователей. Они бывают двух видов: статические и динамические. Состав статических групп определяет администратор через свой интерфейс: он может добавить пользователя в группу или удалить из неё. (Следует отметить, что он не может создавать или удалять сами группы, так как они используются в правилах доступа, определяемых в конфигурации.) Состав динамических групп вычисляется системой. И вновь у настройщика/разработчика имеется большой простор в выборе способа вычисления: SQL-запрос, Java-класс. Но самое интересное в динамических группах то, что они также могут быть контекстными, т.е. зависеть от доменного объекта! Фактически, при определении в конфигурации одной контекстной динамической группы пользователей ПУД создаёт множество реальных групп – столько, сколько есть подходящих доменных объектов. И состав каждой вычисляется исходя из содержимого своего контекстного объекта, т.е. можно снова использовать его поля или связанные объекты.

Важно отметить, что разрешения на конкретные типы доступа выдаются конкретным группам пользователей. И алгоритм, вычисляющий состав контекстной роли, должен определять для контекстной динамической группы её реальный экземпляр (контекст). Чаще всего, именно те доменные объекты, на которые ссылается контекстный или связанный с ним объект, будут, в свою очередь, становиться контекстом для динамических контекстных групп пользователей. Правила доступа для глобальных операций могут ссылаться на контекстные группы пользователей, но тоже должны указывать какой-либо метод определения контекста для такой группы.

## Примеры

Примеры статических групп: Администраторы, Делопроизводители. Живой человек (администратор) при необходимости вручную (через административный интерфейс) добавляет и удаляет пользователей в эти группы. (См. также «О проблеме курицы и яйца»).

Примеры динамических групп без контекста:

* Все пользователи – система автоматически добавляет туда каждого пользователя (доменный объект «Пользователь»), заведённого в системе.
* Делопроизводители – альтернативный вариант реализации группы, добавляющий пользователя в неё на основании, например, некоторого флага (поля) в его профиле (доменном объекте).

Примеры динамических контекстных групп:

* Пользователь и его заместители – по одной группе на каждый доменный объект «Пользователь», в которую, помимо самого этого пользователя, добавляются также временно или постоянно замещающие его пользователи на основе доменных объектов «Замещение». Алгоритм вычисления группы может учитывать и увольнение сотрудника, удаляя его самого из его же группы, при этом его заместители смогут выполнять его функции.
* Сотрудники подразделения – по одной группе на каждый доменный объект «Подразделение», вычисляемой на основании ссылок на подразделение в доменных объектах «Сотрудник».
* Сотрудники подразделения пользователя – вычисляется аналогично предыдущей, но по одной на каждый доменный объект «Сотрудник».

Примеры контекстных ролей:

* Автор документа (для доменного объекта «Документ») – включает группу «Пользователь и его заместители», контекстом для которой является пользователь из поля «Автор».
* Исполнители поручений (для доменного объекта «Документ») – включает экземпляры группы «Пользователь и его заместители» для всех пользователей, содержащихся в полях «Исполнитель» каждого дочернего объекта «Поручение».
* Подписант документа-основания (для доменного объекта «Поручение») – включает группу «Пользователь и его заместители» с контекстом пользователя из поля «Подписант» дочернего объекта «Подпись» родительского объекта (типа «Документ»).

## О типах доступа

Матрица доступа к доменным объектам определяется с точностью до его типа и статуса. Определены следующие типы доступа к доменным объектам:

* чтение
* изменение (запись)
* удаление
* создание дочернего объекта заданного типа
* выполнение заданного конфигурируемого действия

Типы доступа для общесистемных операций перечислены ниже:

* создание доменного объекта заданного типа
* выполнение заданного конфигурируемого действия

Из перечисления видно, что под большинством перечисленных элементов скрывается множество действительных типов доступа, отличающихся конкретными типами объектов или действиями. Права доступа на каждую для каждого такого типа (например, «Создание дочернего объекта „Поручение“», «Создание дочернего объекта „Подпись“» и т.д.) назначаются независимо друг от друга.

Также необходимо отметить, что в списке отсутствует важнейшая контекстная операция «Изменение статуса». Причина заключается в том, что основным механизмом управления жизненным циклом доменного объекта в системе должны быть бизнес-процессы. Они являются (важной) частью бизнес-логики и работают от имени системы. Ручное изменение статуса пользователем (или иным клиентом) системы рассматривается в качестве исключения, и при наличии такой необходимости оно может быть выставлено в качестве конфигурируемого контекстного действия, операция выполнения которого уже является объектом назначения прав доступа. Конечно, при этом подходе операция изменения статуса доменного объекта не может быть вынесена в публичный API системы.

Та же логика может быть применена и к глобальному действию «Создание доменного объекта заданного вида», но отсутствие этой операции в публичном API видится более нелогичным, поэтому она всё же включена в список доступных для назначения прав операций. Тем не менее, это решение таит в себе ряд неудобств. В интерфейсе системы создание недочерних доменных объектов должно быть конфигурируемой функцией (предусмотрено много различных способов их создания в различных местах), для чего весьма удобную концепцию предоставляют конфигурируемые действия. Однако, их выполнение также является операцией, управляемой ПУД, и мы должны либо требовать от настройщика дублировать правила доступа для этих операций, либо предусматривать способ их логического связывания (и автоматического распространения правил с одной на другую).

Наконец, тот же вопрос стоит и в отношении контекстной операции «Назначение (запуск) заданного бизнес-процесса» (для доменного объекта). Если функции управления бизнес-процессами выносятся в публичный API, то они должны быть под управлением ПУД, а список дополнен соответствующими операциями. Или же эти функции генерализуются через интерфейс конфигурируемых действий, и тогда дополнительные операции в ПУД не нужны.

## Разница между контекстными группами и контекстными ролями

Собственно, зачем же в ПРД нужны контекстные роли? Они очень похожи на контекстные группы пользователей. Внешняя разница между ними заключается лишь в том, что контекстные группы включают в себя пользователей, а контекстные роли – группы пользователей (не обязательно контекстные, но в первую очередь именно их). Любые правила можно технически реализовать с использованием только контекстных групп.

Важнейшее различие между этими объектами заключается в их внутренней реализации. Контекстные группы пользователей существуют физически – в виде записей в БД, по одной на каждый контекстный объект для каждой группы. Хранится также состав каждой такой группы – список пользователей, входящих в неё. При каждом изменении контекста (доменного объекта или связанных с ним, влияющих на состав группы) производится повторное вычисление состава этой группы.

Контекстные роли, с другой стороны, существуют только в конфигурации. Их состав не хранится в БД, а вычисляется непосредственно при вычислении списка доступа к доменному объекту.

Этими свойствами групп и ролей обуславливается разница в их использовании. Контекстные группы пользователей предназначены для использования с относительно редко изменяющимися объектами в качестве контекста, однако позволяют задавать относительно сложные правила их формирования. Для СЭД такими объектами являются пользователи и связанные с ними сущности (профили), организационно-штатная структура и т.п. Контекстные роли связываются с динамичными объектами системы (документы, поручения и т.п.), существующими в большом количестве, но должны иметь максимально лёгкие и быстрые алгоритмы для их вычисления.

## Списки доступа

Списки доступа (Access control list, ACL) – внутренний объект ПУД, предназначенный для оптимизации запросов определения доступа.

Списки доступа для контекстных операций хранятся в таблицах БД – по одной таблице на каждый тип доменных объектов. Списки доступа хранят *актуальные* разрешения (permissions) на доступ различного типа к каждому доменному объекту. Субъектами разрешений выступают группы пользователей. Актуальность означает хранение только тех разрешений, которые действительны для текущего статуса доменного объекта. При изменении статуса объекта производится повторное вычисление списка доступа к нему.

ПУД также отслеживает все изменения доменных объектов и другие события, которые могут привести к изменению списков доступа, и вносит соответствующие изменения в списки.

Наличие списков доступа позволяет реализовать проверку прав пользователя на выполнение той или иной операции (включая операции над множеством однотипных объектов, например, чтение списка) через обращение лишь к двум таблицам БД: списку доступа (частному для этого типа объектов) и составу групп пользователей.

С учётом тотального преобладания количества разрешений на чтение над количеством разрешений на любые другие операции над объектом, а также количества запросов прав чтения над всеми прочими, списки доступа разбиваются на две отдельные таблицы (для каждого типа доменных объектов, конечно): списки чтения и списки доступа (остальные операции).

## Архитектура

ПУД состоит из нескольких компонентов, включённых в ядро системы. Ключевой из них – служба контроля доступа (СКД), обеспечивающая проверку прав. Она основана на схеме маркеров доступа (access tokens). Все сервисы DAO-слоя (базовые сервисы), осуществляющие непосредственный доступ к защищаемым объектам, требуют от вызывающего их кода предоставления специального объекта – маркера доступа. Это объект инкапсулирует данные о субъекте, объекте и типе доступа (операции). Они обращаются к СКД для проверки подлинности и соответствия маркера доступа запрошенной операции, и только в случае положительного ответа производят нужный доступ. Для получения маркера доступа код бизнес-слоя системы также должен обратиться к СКД, которая при этом производит проверку прав (авторизацию) субъекта доступа. Такой подход блокирует возможность доступа к доменным объектам в обход СКД.

Следует отметить, что субъектом доступа может быть не только пользователь. Любой код, выполняющийся в бизнес-слое, может запросить маркер доступа как от имени пользователя (любого), так и от имени системы, указав при этом идентификатор процесса. Права доступа системы к любым объектам не ограничены, но идентификатор процесса, явно указанный в его коде, сохраняется в системных журналах, протоколирующих доступ к защищаемым объектам. Это позволяет не давать пользователям лишних прав на изменение тех объектов, которые должны изменяться только бизнес-правилами при внесении изменений пользователем в другие объекты (связанные), упрощая конфигурацию прав и защищая объекты от несогласованных «ручных» изменений. Кроме того, протоколирование источников изменений с точностью до процесса упрощает отладку системы и проведение расследований по обращениям заказчиков.

В СКД также реализуется возможность делегирования базовым сервисам функций по проверке прав доступа. Такие сервисы получают возможность встраивания запросов проверки прав в основные запросы, осуществляющие доступ к объектам. Эта возможность используется лишь для некоторых (доверенных) сервисов и для ограниченного набора типов доступа, в основном – для чтения отдельных доменных объектов и коллекций. Принимая на себя такую ответственность, доверенные сервисы становятся, по сути, частью ПУД.

СКД и доверенные базовые сервисы осуществляют проверки прав с использованием списков доступа и таблиц членства пользователей в группах.

Другая важная часть ПУД – компоненты, обеспечивающие актуальность списков доступа. В их число входят классы, реализующие различные алгоритмы формирования контекстных ролей и динамических групп пользователей. Помимо собственно вычисления их состава, эти компоненты реагируют на различные события, происходящие в системе, которые могут повлиять на состав вычисляемых ролей или групп (прежде всего – изменение доменных объектов), осуществляя немедленный пересчёт. И снова некоторая доля ответственности ложится на базовые сервисы, которые обязаны информировать компоненты ПУД о важных для них событиях.

## Распространение прав доступа

Согласно требованиям по защите информации, система должна содержать средства, ограничивающие распространение прав на доступ. Под распространением мы понимаем неявное предоставление прав доступа к объектам вследствие изменения их содержания. Поясню на примере: пользователю выдано поручение по документу, вследствие чего, по соответствующему правилу («Все исполнители поручений могут читать документ-основание») он имеет доступ к документу на чтение. Предположим, что он перепоручает своё задание другому лицу, или выдаёт связанное поручение. Это лицо автоматически попадает в контекстную роль «Все исполнители поручений» и получает доступ на чтение к тому же документу. В соответствии с бизнес-задачей, всё так и должно быть. Но вот соответствует ли это требованиям безопасности? Имеет ли право пользователь предоставлять таким образом доступ к документу кому угодно? Или круг лиц, кого он может вовлечь, должен быть ограничен? Или так могут делать пользователи, но не все?

Частично защита от неконтролируемого распространения прав обеспечивается строгим подходом к конфигурированию правил доступа. В некоторых случаях может понадобиться дополнительный защищаемый тип доступа – привязка доменного объекта к другому (контекстному). Также возможно создание дополнительного сервиса со своим набором правил, описывающих разрешения на распространение прав (кто – кому). Этот сервис должен контролировать все изменения списков доступа и блокировать запрещённые. (Единственно надёжным вариантом здесь представляется откат транзакций, пытающихся выполнить несанкционированные изменения, т.к. просто удалённые разрешения будут выданы позже при изменении объекта от имени системы.)

Перед окончательным выбором решения необходимо выработать единый подход к ней со стороны специалиста по защите информации и бизнес-аналитика системы.

## Доступ к полям объектов

В некоторых случаях выдвигаются требования дополнительного разграничения доступа к отдельным полям объектов (как на чтение, так и на изменение). Непосредственная реализация этих требований может привести к чрезмерному росту как объёма списков доступа, и без того немалого, так и количества вычислений этих списков.

Для удовлетворения подобных требований можно предложить использовать декомпозицию доменных объектов на такие части, что правила доступа ко всем полям одной части будут одинаковы, и соединение этих доменных объектов в иерархию.

## О проблеме курицы и яйца

Уже упоминалось, что управление составом групп пользователей осуществляет администратор. Но сам администратор – это не кто иной, как пользователь системы, включённый в группу «Администраторы». Но откуда же возьмётся самый первый администратор? (Подобная проблема, кстати, встаёт и с пользователями вообще – управление пользователями обычно осуществляет также администратор, но он сам является пользователем, и кто-то должен добавить его самого.)

Для решения этой проблемы при старте системы производится специальная проверка. Если группа «Администраторы» пуста (не содержит ни одного пользователя), создаётся специальный пользователь system\_administrator (если он ещё не существует) и добавляется в группу «Администраторы». Это не только решает проблему первого старта, но и защищает систему от случайного удаления всех администраторов. Конечно, доменные объекты «Пользователь», «Группа пользователей», а также правила, предоставляющие права на изменение этих объектов группе «Администраторы» задаются в системной конфигурации (недоступной для изменения настройщиком).

## Дополнительные функции

Требуется реализовать возможность включения одних групп пользователей в другие. При этом включение может быть вложенным (на любую глубину). Такая функция была реализована в прототипе системы. Детали конфигурирования и реализации будут изложены позже.

С учётом наличия в системе большого количества служебных доменных объектов (в частности, для реализации полей с множественными значениями) необходимо реализовать возможность «наследования» списков доступа дочерними объектами от родительских. Т.е. для конкретного типа доменного объекта в конфигурации может быть указано использование родительских списков доступа; в таком случае вместо собственных таблиц для него создаётся отдельная таблица (или поле) с указателем на имеющего список доступа родителя (т.к. наследование может быть многоуровневым), а запросы определения прав должны использовать списки доступа этого родителя.

Для следующих версий системы рассматривается возможность дополнительной реализации мандатного принципа доступа. При этом для выполнения операции пользователь должен будет иметь разрешения по обоим принципам: и дискреционному (нынешнему), и мандатному.