8 МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ

УДК 004.056.52

МНОГОМЕРНАЯ МОДЕЛЬ РАЗГРАНИЧЕНИЯ ДОСТУПА К ОБЪЕКТАМ В СИСТЕМЕ КОНТРОЛЯ ВЕРСИЙ

А.И. Спивак, А.В. Разумовский, И.А. Зикратов

Предложена новая модель разграничения доступа для систем контроля версий файлов. Рассматривается трехмерная модель разграничения доступа на основе дискреционной модели Харрисона–Руззо–Ульмана. Для решения задачи управления и контроля базовые операции модели Харрисона–Руззо–Ульмана дополнены новыми операциями, учитывающими доступ субъектов к версиям объектов.

Ключевые слова: модель, безопасность, разграничение доступа, контроль версий.

Введение

Разработка и эксплуатация сложных IT-систем предполагает участие в этих процессах коллектива разработчиков и пользователей. В частности, при разработке программного обеспечения 3D-анимации объектами являются исходные тексты программ, отдельные графические сцены и т.д. Для координации усилий по разработке одних и тех же объектов со стороны множества разработчиков применяются системы контроля версий. Известно, что разграничение доступа в таких системах, как Subversion [1], CVS [2] обеспечивается на уровне объекта доступа и подразумевает все возможные его версии, которые были получены при совместной работе разработчиков. Однако реализовать разграничение доступа субъектов на уровне версии не представляется возможным, так как такой функционал не реализован в существующих системах контроля версий. В этом случае особую актуальность приобретает задача обеспечения совместной работы различных субъектов над одними и теми же объектами, что подразумевает независимое появление различных версий исходного объекта. Для решения этой задачи авторами взята за основу известная из теории информационной безопасности модель дискреционного разграничения доступа Харрисона–Руззо–Ульмана (HRU) [3, 4]. Доработка этой модели путем введения в ее состав совокупности матриц доступа, описывающих разграничение субъектов к версиям объектов, позволила реализовать механизм контроля версий при производстве мультимедийного контента в системе 3D-анимации.

Дискреционная модель разграничения доступа в системах контроля версий

При разработке мультимедийного контента особое внимание уделяется наличию отдельных версий одних и тех же сцен, а также возможность существования отличающихся прав доступа к отдельным версиям сцен со стороны субъектов-разработчиков. Традиционные системы контроля версий обеспечивают распределение доступа к версиям бинарных файлов посредством механизма блокировок. Механизм блокировки позволяет одному из разработчиков захватить в монопольное использование файл или группу файлов для внесения в них изменений. На то время, пока файл заблокирован, все права у остальных пользователей отбираются, он остается доступным всем остальным разработчикам только на чтение, и любая попытка внести в него изменения отвергается сервером. Недостатки использования блокировок очевидны:

- блокировки мешают продуктивной работе, поскольку вынуждают ожидать освобождения блокированных файлов;
- блокировки создают административные проблемы, когда разработчик может забыть снять блокировку с занятых им файлов.

Для разрешения подобных проблем приходится применять административные меры, в том числе включать в систему технические средства для сброса неверных блокировок, но и при их наличии на приведение системы в порядок расходуется время.

Основным отличием предлагаемой модели разграничения доступа является представление версий объектов не в виде древовидной структуры, количество узлов и ветвей в которой является случайными и трудно прогнозируемыми величинами, а виде совокупности матриц доступа. Согласно модели HRU, матрица доступа описывает права доступа именованных субъектов к именованным объектам, которые записаны на пересечении соответствующих строк и столбцов.

В качестве субъектов доступа в предлагаемой модели рассматриваются разработчики, работающие с различными версиями сцен. Объектами являются результаты работы – файлы-сцены. Тогда матрица прав доступа имеет в качестве столбцов объекты, которыми являются файлы-сцены, а строками – субъекты-разработчики. В ячейках-пересечениях находятся права доступа данного субъекта к данному объекту. При появлении i-й версии объекта создается дополнительная i-я матрица доступа, которая отражает

права к этой версии объектов со стороны субъектов. В случае отсутствия версий для какого-либо объекта ячейки, соответствующие его версиям, будут пустыми.

В соответствии с формальным описанием модели HRU обозначим: O – множество объектов; S – множество субъектов; R – множество прав доступа субъектов к объектам; V – множество версий объектов O. Для описания отношений между субъектами в модели принимается следующее отношение принадлежности множества субъектов к множеству объектов: $S \in O$.

Пространство состояний такой системы представляется в виде $S \times O \times R \times V$.

Матрица прав доступа M, где столбцами являются объекты, а строками – субъекты, содержит права доступа субъектов к объектам. При создании новой версии объекта (V_i) создается новая матрица $W_{vl}[s, o]$, где содержатся права доступа субъектов к новой версии объекта. Для упрощения получения информации о последнем изменении объекта применяется сквозная нумерация версий объектов, при этом не имеет значения, какой именно субъект создал новую версию объекта. Суть модели поясняется рисунком.

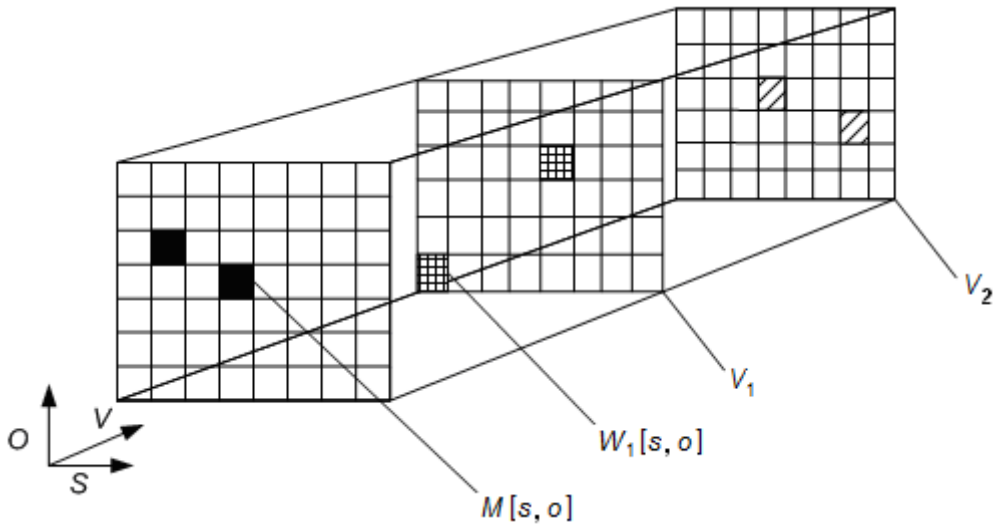

Рисунок. Совокупность матриц доступа субъектов к версиям объектов

На рисунке изображена трехмерная матрица прав доступа в системе контроля версий, в которой присутствуют различные права доступа к разным версиям. Отмеченные ячейки в матрице показывают наличие определенных прав субъекта к данному объекту для версии объекта $W_i[s, o]$.

Формальное описание операций управления системой контроля версий

Положив в основу описания системы формальный аппарат модели HRU, дополним его множеством версий объектов. Тогда состояние системы, добавление и удаление прав доступа, субъектов, объектов и их версий может быть описано следующим образом.

Пусть, учитывая введенные выше обозначения, состояние системы представлено в следующем виде:

$Q=(S, O, V, M, W)$,

где $M[s, o]$ – ячейка, содержащая элементы из множества R, строки – субъекты, столбцы – объекты; $W_v[s, o]$ – ячейка, содержащая элементы из множества R, строки – субъекты, столбцы – версии объектов.

Изменения в состояние системы могут быть внесены посредством команд $a(x_1, \ldots, x_n)$, которые содержат условия выполнения команды и базовые операторы:

if $r_{1n} \in M[x_{s1}, x_{o1}]$ and … and $r_{nn} \in M[x_{sn}, x_{on}]$ … and $r_{1m} \in W_{v1}[x_{s1}, x_{o1}]$ and … and $r_{nm} \in W_{vm}[x_{sm}, x_{om}]$ and…

then

$c_1(\ldots)$

$c_2(\ldots)$

…..

$c_p(\ldots)$

$r_1, \ldots, r_n \in R$ – права доступа;

$c_1, \ldots, c_p$ – набор операторов, в которых в качестве параметров принимаются $x_1, \ldots, x_n$.

При выполнении команды $a()$ система переходит из состояния Q в состояние Q'.

Элементарные операторы модели Харрисона-Руззо-Ульмана с учетом расширения модели приобретают следующий вид:

1. Добавление права субъекту по отношению к объекту **Enter *r M[s,o]***.

Начальное состояние $q=(S, O, V, M, W)$: $s\in S$, $o\in O$, $r\in R$, $v\in V$.

Конечное состояние $q=(S', O', V', M', W')$: $S'=S$, $O'=O$, $V'=V$, $W'[s, o]=W[s, o]$, $M'[s, o]=M[s, o]\cup\{r\}$, если $(x_s, x_o)\neq(s, o)\Rightarrow M'[x_s, x_o]=M[x_s, x_o]$.

2. Удаление права у субъекта по отношению к объекту **Delete r** $M[s,o]$.

Начальное состояние $q=(S, O, V, M, W)$: $s\in S$, $o\in O$, $r\in R$, $v\in V$.

Конечное состояние $q=(S', O', V', M', W')$: $S'=S$, $O'=O$, $V'=V$, $W'[s, o]=W[s, o]$, $M'[s, o]=M[s, o]\setminus\{r\}$, если $(x_s, x_o)\neq(s, o)\Rightarrow M'[x_s, x_o]=M[x_s, x_o]$.

3. Создание субъекта **Create s'.**

Начальное состояние $q=(S, O, V, M, W)$: $s'\notin S$.

Конечное состояние $q=(S', O', V', M', W')$: $S'=S\cup\{s'\}$, $O'=O\cup\{s'\}$, $V'=V$, для $\forall(x_s, x_o)\in S\times O \Rightarrow M'[x_s,x_o]=M[x_s, x_o]$, $W'[x_s, x_o]=W[x_s, x_o]$, $M'[s', x_o]=\varnothing$ для $\forall x_o\in O'$, $M'[s', x_s]=\varnothing$ для $\forall x_s\in S'$, для $\forall v\in V'$, $\forall o\in O'\Rightarrow W_v'[s', o]=\varnothing$.

4. Удаление субъекта **Destroy s'**.

Начальное состояние $q=(S,O,V,M,W)$: $s'\in S$.

Конечное состояние $q=(S',O',V',M',W')$: $S'=S\setminus\{s'\}$, $O'=O\setminus\{s'\}$, для $\forall(x_s, x_o)\in S'\times O' \Rightarrow M'[x_s, x_o]=M[x_s,x_o]$, $W'[x_s, x_o]=W[x_s, x_o]$.

5. Создание объекта **Create o'**.

Начальное состояние $q=(S,O,V,M,W)$: $o'\notin O$.

Конечное состояние $q=(S',O',V',M',W')$: $S'=S$, $O'=O\cup\{o'\}$, для $\forall(x_s, x_o)\in S\times O\Rightarrow M'[x_s, x_o]=M[x_s, x_o]$, $W'[x_s, x_o]=W[x_s, x_o]$, для $\forall x_s\in S'\Rightarrow M'[x_s, o']=\varnothing$, для $\forall v\in V'$ и $\forall x_s\in S'\Rightarrow W_v'[x_s,o']=\varnothing$.

6. Удаление объекта **Destroy o'.**

Начальное состояние $q=(S,O,V,M,W)$: $o'\in O, o'\notin S$.

Конечное состояние $q=(S',O',V',M',W')$: $S'=S$, $O'=O\setminus\{o'\}$, для $\forall(x_s, x_o)\in S'\times O'\Rightarrow M'[x_s, x_o]=M[x_s, x_o]$, $W'[x_s, x_o]=W[x_s, x_o]$.

Наличие в новой модели версий объектов приводит к следующим дополнительным операциям:

1. Добавление права субъекту по отношению к версии объекта **Enter** $r\ W[s, o]$.

Начальное состояние $q=(S, O, V, M, W)$: $s\in S$, $o\in O$, $r\in R$, $v\in V$.

Конечное состояние $q=(S', O', V', M', W')$: $S'=S$, $O'=O$, $V'=V$, $W_v'[s, o]=W_v[s, o]\cup\{r\}$, если $(x_s, x_o)\neq(s, o)\Rightarrow W_v'[x_s, x_o]=W_v[x_s, x_o]$, $M'[s, o]=M[s, o]$.

2. Удаление права у субъекта по отношению к версии объекта **Delete** $r\ W[s, o]$.

Начальное состояние $q=(S, O, V, M, W)$: $s\in S$, $o\in O$, $r\in R$, $v\in V$.

Конечное состояние $q=(S', O', V', M', W')$: $S'=S$, $O'=O$, $V'=V$, $W_v'[s, o]=W_v[s, o]\setminus\{r\}$, если $(x_s, x_o)\neq(s, o)\Rightarrow W_v'[x_s, x_o]=W_v[x_s, x_o]$, $M'[s, o]=M[s, o]$.

3. Создание версии объекта **Create v'.**

Начальное состояние $q=(S, O, V, M, W)$: $v'\notin V$.

Конечное состояние $q=(S', O', V', M', W')$: $S'=S$, $V'=V\cup\{v'\}$, для $\forall(x_s, x_o)\in S\times O\Rightarrow M'[s, o]=M[s, o]$, $W_{v'}'[x_s, x_o]=\varnothing$.

4. Удаление версии объекта **Delete v'.**

Начальное состояние $q=(S, O, V, M, W)$: $v'\in V$.

Конечное состояние $q=(S', O', V', M', W')$: $S'=S$, $O'=O$, $V'=V\setminus\{v'\}$ для $\forall(x_s, x_o)\in S'\times O'\Rightarrow M'[s, o]=M[s, o]$, $W_{v'}'[x_s, x_o]=W_v[x_s, x_o]$.

Таким образом, наличие апробированного формального описания дискреционной системы контроля версий файлов позволяет устранить недостатки, присущие традиционным механизмам блокировки файлов, обеспечить ясную физическую интерпретацию и контролируемое масштабирование процесса создания версий файлов-сцен, обеспечивает простоту разработки программного кода.

Реализация модели разграничения доступа

Описанная в работе дискреционная модель разграничения доступа реализована в программном модуле системы хранения и контроля версий (СКВ) больших бинарных файлов инновационной системы 3D-анимации. Модуль включает в себя несколько обособленных программных блоков.

Функционал модели реализуется компонентом контроля доступа. В его составе выделяются блок управления субъектами и объектами, блок управления правами доступа, блок проверки прав доступа.

В задачи блока управления субъектами и объектами входит внесение изменений в принятую трехмерную матрицу доступа в соответствии с описанными переопределенными операторами, а именно, добавление субъектов, объектов, новых версий и прав к ним.

Блок управления правами доступа ответственен за редактирование прав доступа между субъектами и объектами, в данном случае происходит внесение в ячейки матриц $M[s,o]$ и $W_v[s,o]$ прав r.

Задачи по предоставлению прав субъектов к объектам решаются блоком проверки прав доступа. Благодаря реализованной модели, у пользователей появилась возможность более гибко настраивать права на версии объектов (файлов). Это, в частности, позволяет делать «срез» по правам объектов на определенный момент времени и тем самым избежать использования устаревших версий объектов, что очень важно при работе над сложной сценой в большом фильме с участием множества разработчиков.

Архитектура СКВ имеет распределенный характер. В СКВ выделены три компонента: ядро СКВ, агенты СКВ и клиент СКВ. Благодаря тому, что данные (файлы) передаются только между агентом и клиентом, такая архитектура позволяет снизить нагрузку на ядро СКВ.

Интерфейсы компонентов для общения между собой и программными модулями, использующими СКВ для получения файлов и их версий, унифицированы и представляют собой REST-интерфейс [5].

Все компоненты реализованы на языке Java. Это позволяет развернуть компоненты на базе любой операционной системы, имеющей JVM.

Разработка модели выполнена в рамках ФЦП ГК № 07.524.11.4009 «Разработка инновационной системы 3D-анимации».

Заключение

Разработанная многомерная модель позволяет использующим ее системам контроля версии обеспечивать разграничение доступа на уровне отдельных версий. Такой функционал является конкурентным преимуществом по сравнению с существующими системами контроля версий и дает возможность более гибкого доступа к объектам по сравнению с механизмом блокировки управления.

Литература

1. Система управления версиями Subversion [Электронный ресурс]. – Режим доступа: http://subversion.apache.org/, свободный. Яз. англ. (дата обращения 01.03.2012).
2. Система управления версиями CVS [Электронный ресурс]. – Режим доступа: http://cvs.nongnu.org/, свободный. Яз. англ. (дата обращения 01.03.2012).
3. Девянин П.Н. Теоретические основы компьютерной безопасности. – М.: Радио и связь, 2001. – 192 с.
4. Малюк А.А. Информационная безопасность: концептуальные и методологические основы защиты информации. – М.: Горячая линия – Телеком, 2004. – 280 с.
5. Roy Thomas Fielding. Architectural Styles and the Design of Network-based Software Architectures [Электронный ресурс]. – Режим доступа: http://www.ics.uci.edu/Efielding/pubs/dissertation/top.htm, свободный. Яз. англ. (дата обращения 01.03.2012).

Спивак Антон Игоревич – Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кандидат технических наук, доцент, anton.spivak@gmail.com

Разумовский Андрей Владимирович – Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кандидат технических наук, ассистент, xrew@yandex.ru

Зикратов Игорь Алексеевич – Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, доктор технических наук, доцент, зав. кафедрой, zikratov@cit.ifmo.ru