

УПРАВЛЕНИЕ РАЗРАБОТКОЙ ПРОЕКТНОЙ ДОКУМЕНТАЦИЕЙ

С. П. Бонадаренко

Кафедра технологий программирования, Белорусский государственный университет

Минск, Республика Беларусь

E-mail: {svetabon}@gmail.com

В данной работе проводятся результаты разработки модуля управления данными, входящего в состав системы управления разработкой проектной документации. Исследуются основные модели безопасности и выбираются наиболее подходящие из них для реализации управления доступом к данным системы. Проектируется модель для централизованного хранения данных на сервере. Реализовано приложение для разграничения и управления доступом к этим данным.

ВВЕДЕНИЕ

Целью функционирования проектно-исследовательских организаций является создание пакета проектно-сметной документации, описывающей совокупность бизнес-процессов необходимых для достижения конкретного результата. Это может быть строительство жилых зданий, электростанций и т. п. Однако своевременное исполнение подобного вида работ может срываться из-за ряда причин, к которым можно отнести: нарушение сроков выполнения проектов, сложность обеспечения своевременности получения согласований и технических условий [1]. Существенным недостатком является также децентрализованное хранение электронной проектной документации, вызванное отсутствием единого хранилища и единых правил обработки и хранения электронной проектной документации. Отсутствует также разграничение прав доступа к проектной документации и механизм версииности для документов. .

1. ОПИСАНИЕ МОДУЛЯ УПРАВЛЕНИЯ ДАННЫМИ

В силу вышеуказанных причин возникает необходимость в разработке системы, предназначенной для автоматизации управления разработкой проектной документации в организациях подобного типа. Основной целью подобной системы является автоматизация планирования и контроля хода выполнения каждого проекта организации в реальном режиме времени.

Информационная часть системы включает в себя такие объекты как файлы, документы, папки. Документ – объект системы, который представляет контейнер для хранения файлов, и этим он подобен папке. Папка – объект системы, представляющий собой контейнер в электронной библиотеке для хранения документов, файлов и других папок, аналог папки в файловой системе и служит для систематизации документов и управления правами доступа. Совокупность файлов, документов и папок образует электронную библиотеку. При каждом изменении любого документа и файла в электронной библиотеке

должна со-храняться предыдущая версия и информация об авторе и времени изменения.

Система управления разработкой проектной документации в общем случае должна включать в себя модуль, обеспечивающий контроль совокупности выполняемых в организации проектов в целом; модуль, позволяющий реализовать управление конкретным проектом; осуществлять редактирование плана проекта; модуль, обеспечивающий работу с электронной библиотекой проектно-сметной документации, представляющей совокупность множества различных электронных документов, необходимых для реализации проекта.

Следует отметить, что проекты, выполняемые проектно-исследовательскими организациями, могут быть типизированы. Это означает, что каждая организация выполняет проекты одного или более типов, в рамках которых подобны структуры работ, роли исполнителей, содержание проектно-сметной документации, которая является результатом проекта. В связи с вышесказанным особенно актуальным в системе управления разработкой проектной документации является приложение, обеспечивающее управление электронной библиотекой. Существенным в таком приложении является выбор используемых методов разграничения доступа к данным, централизованно хранящихся на сервере. Решение этой задачи связано с исследованием и выбором соответствующей модели безопасности [2, 3].

При реализации модуля управления данными в качестве объектов были выбраны папка, файл, документ. Папка может содержать файлы и документы. Кроме этого, файл может иметь несколько версий, причем расположенных в хронологическом порядке. Документ – это абстрактная сущность, которая хранит в себе несколько файлов. По сути, документ это та же папка, но обладающая дополнительными свойствами. Документ, так же как и файл, имеет несколько версий, этим достигается версияльность документов.

Все операции записи над файлами должны быть атомарными, что исключает появле-

ние неверных или противоречивых данных. При этом если изменяется файл, создается новая версия, а старая продолжает храниться. Если изменяется, добавляется или удаляется файл в документе, то создается новая версия документа. Атомарность операций достигается механизмом блокировок.

В приложении все пользователи могут выполнять одни и те же операции с файлами, если у них есть права на их выполнение. Если пользователь запрашивает файл на открытие, система сначала проверяет, авторизован ли он для этой операции, затем проверяет, не заблокирован ли файл другим пользователем, и только после этого блокирует файл текущим пользователем и открывает его.

Если пользователь добавляет файл, система сначала проверяет, авторизован ли он для этой операции, затем записывает файл с локального диска пользователя (с рабочей машины пользователя) в централизованное хранилище и создает файл.

Если пользователь удаляет файл, система сначала проверяет, авторизован ли он для этой операции, затем проверяет, не заблокирован ли файл другим пользователем, и только после этого удаляет файл. Причем, физического удаления не происходит, файл просто помечается как удаленный.

Для разграничения доступа к данным было решено использовать модель безопасности на основе списков прав доступа (ACL). Для улучшения механизма управления пользователями и их правами было решено совместить модель на основе списков с моделью разграничения доступа на основе ролей.

Было выделено три основных типа объектов доступа: папка, файл, документ. В качестве субъектов безопасности в ACL-списках могут выступать не только пользователи, но и группы пользователей из ролевой модели. Это значительно упрощает управление правами доступа, улучшает понимание модели, снижает расходы на хранение списков прав доступа в хранилище данных и увеличивает производительность. Для хранения пользовательских данных используется хранилище FILESTREAM. Это позволяет получать доступ к файлам как к обычным файловым потокам, причем делая это из хранимых процедур. Хранилище FILESTREAM объединяет компонент SQL Server Database Engine с файловой системой NTFS, размещая данные больших двоичных объектов (BLOB) типа `varbinary(max)` в файловой системе в виде файлов. С помощью инструкций `Transact-SQL` можно вставлять, обновлять, запрашивать, выполнять поиск и выполнять резервное копирование данных FILESTREAM. Интерфейсы файловой системы Win32 предоставляют потоковый доступ к этим данным.

Для кэширования данных файлов в хранилище FILESTREAM используется системный кэш NT. Это позволяет снизить возможное влияние данных FILESTREAM на производительность компонента Database Engine. Вся работа с данными ведётся через ORM EntityFramework, с использованием хранимых процедур, в которых уже идёт проверка прав доступа. В сочетании с запретом прямого доступа на таблицы с данными, можно добиться хорошей защищённости системы.

Для обеспечения совместной работы с файлами применяется механизм блокировок. Суть его состоит в следующем: если объект открывается на запись каким-либо пользователем, то этот пользователь блокирует его, и объект становится недоступным для записи всем остальным пользователям, но остаётся доступным для чтения. В общем, моделируется обычное поведение файловой системы, за исключением того, что ведётся полная история изменений, сохраняются все версии файлов, почти исключается потеря данных, собирается статистика.

Для успешного функционирования системы безопасности обеспечены возможности настройки различных прав доступа для каждого типа объекта (папка, файл, документ); наследования прав доступа родительского объекта (для файлов и документов – папка, для файлов – документ); настройки прав доступа как для отдельных пользователей, так и для групп пользователей.

Формируется несколько различных списков доступа – для каждого типа объекта (папка, файл, документ) и для каждого субъекта (пользователь, группа пользователей). Каждый элемент списка содержит разрешение либо запрет одной операции для одного пользователя, а так же данные о том, наследуется ли это право от родительского объекта.

II. ЗАКЛЮЧЕНИЕ

Приложение разработано на языке C с использованием технологий WindowsForms. Для хранения данных используется база данных на сервере MS SQLServer с включённым хранилищем FILESTREAM. В качестве уровня доступа к данным используется ORM EntityFramework. Приложение моделирует поведение файловой системы, дополняя его ведением версииности файлов, ведением полной истории изменений, статистики и гибкой системой настройки прав доступа.

1. Управление проектами / И.И. Мазур [и др.]. – М.:Омега. – 2010. – 960 с.
2. Шаньгин, В. Ф. Защита компьютерной информации. Эффективные методы и средства / В.Ф.Шаньгин // – М.: ДМК. – 2008. – 554 с.
3. Ховард, М. Защищенный код / М. Ховард, Д.Леблан // – М.: Русская редакция. – 2005. – 704 с.