УДК 598.87

Н. В. Медведев, Г. А. Гришин,
Д. П. Кацыв

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ И МЕТОДИКА РАЗРАБОТКИ ЗАЩИЩЕННЫХ СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА НА БАЗЕ ТЕХНОЛОГИИ IBM LOTUS NOTES/DOMINO

Приведена формальная математическая модель защищенного документооборота и технология ее разработки, вытекающая из этой модели. В основу построения модели положены: нереляционные базы данных, дискреционная модель разграничения доступа к данным. Рассмотрены подходы к построению систем, обеспечивающих безопасный электронный документооборот. Предложен метод построения защищенного электронного документооборота. Приведен пример реализации подобной системы с использованием технологии IBM Lotus Notes/Domino.

В настоящее время интенсивное развитие средств телекоммуникации и глобальных компьютерных сетей позволяет строить распределенные информационные системы для информационной поддержки различных процессов и организации единого рабочего потока для работы групп и целых коллективов как единого целого. Поскольку указанные процессы разнообразны по содержанию, то решение проблемы в области предоставления универсального механизма построения защищенного рабочего потока является актуальным. Одним из вариантов решения подобной проблемы является использование формального подхода логического структурирования. В настоящей работе предложен вариант построения защищенной системы электронного документооборота.

Рабочий поток: интегрированная модель. Жизнеспособность бизнеса определяется возможностью использования, управления, разделения и защиты информации. Важно обратить внимание на то, что данные не являются информацией. Разработчики приложений должны нести накладные расходы для превращения исходных данных в знания, т.е. в полезную информацию. Термин группового программного обеспечения (программное обеспечение для управления групповой информацией) — это свободное определяемое понятие, которое относится к типу прикладного программирования. Рабочий поток —

данные и информация, которая циркулирует в компьютерных сетях, использующих групповое программное обеспечение. Групповое программное обеспечение развивается исходя из общей модели и модели пересылки.

Общая модель основана на том, что документ или приложение базы данных находится в области, доступной для всех пользователей, т.е. используются совместно. При этом документ или приложение базы данных обычно размещается на файловом сервере. Если все пользователи имеют доступ к каталогу на файловом сервере, где размещается файл, все они могут работать с ним. Большинство приложений баз данных не поддерживает одновременный доступ на уровне отдельных записей. Подобный доступ возможен только на уровне файлов. Этим и характеризуется общая модель.

Пользователь должен получать доступ к этому приложению для того, чтобы воспользоваться какой-либо информацией [1]. Это может стать недостатком, поскольку для выполнения какого-либо действия необходимо полагаться на пользователя, объявлять об одобрении требования своевременно и по своему усмотрению. Фактически, если пользователь не осуществляет доступ к базе данных, то не следует и беспокоиться об одобрении требований. Хотя и существует централизованный репозитарий для всех документов, приложение описанного типа не способно к принятию каких-либо действий, основанных на изменении состояния.

В модели пересылки информация перемещается или же отсылается пользователю электронной почтой. Примеры прикладных программ подобного типа — это маршрутизаторы форм, требований и утверждения документов. Использование электронной почты для направления форм близко по смыслу пересылке бумажного документа в ведомстве.

Недостаток этой модели заключается в том, что нет какого-либо удобного способа для определения статуса вашего требования или даже идентификации его хозяина [2]. Отсутствует какое-либо централизованное местоположение для наблюдения за процессом и нет какой-либо общедоступной базы данных, содержащей требования. Отсутствие централизованного репозитария для хранения форм требований означает, что нет какого-либо способа сохранять и отслеживать хронологию подобных форм. После того как форма требования удаляется из последнего почтового ящика, она теряется навсегда. Говоря другими словами, отсутствует способ управления документами.

Lotus Notes/Domino проблема, присущая как моделям пересылки, так и общим моделям, решается путем объединения общедоступной базы данных с механизмом электронной почты. Кроме того, сервер имеет возможность составлять расписание для агентов, которые могут его использовать в определенное время, а также предпринимает

действия на основе некоторых условий, независимых от участия пользователя. Процесс выдвижения требований представляет собой типичное решение рабочего потока. Активация рабочего потока предполагает создание баз данных, наделенных возможностями электронной почты.

Рис. 1. Три "кита" рабочего потока

Применяя рабочий поток после составления требования и сохранения в базе данных, приложение определяет участника, а затем сообщает ему о требовании. Сообщение обычно включает ссылку на документ, который можно открыть, щелкнув кнопкой мыши. Участник оценивает требование и может тут же одобрить его.

Три "кита" рабочего потока — это связь, сотрудничество и координация. Связь означает передачу сообщений или модель пересылки, сотрудничество определяет использование общедоступной модели, а координацию осуществляет коллектив сотрудников, работающих вместе для достижения определенных целей. Связь и сотрудничество поддерживают координацию. В этом суть группового программного обеспечения. Описанные три "кита" рабочего потока показаны на рис. 1.

Пакет Lotus Notes/Domino прекрасно подходит для проектирования рабочего потока приложения, поскольку позволяет объединить передачу сообщений и общедоступные базы данных, преодолевая ограничения общих моделей и моделей пересылки.

Построение системы защищенного документооборота. При построении любой системы, в том числе и системы электронного документооборота, необходима формализация задачи, что позволяет разработать определенные методы ее решения.

На рис. 2 приведена обобщенная схема построения системы электронного документооборота. Для формализации задач в области электронного документооборота целесообразно использовать метод логического структурирования (Logical Frameworks). Непосредственное применение указанного метода для формирования структуры информационной сети подробно описано в работе [3]. В результате применения указанного метода, обозначенного на рис. 2 оператором L (шаг a), получим множество частных задач $\{A_j\} : j \in [1..J]$, $j \in Z$, где J — количество частных задач, результат решения которых может

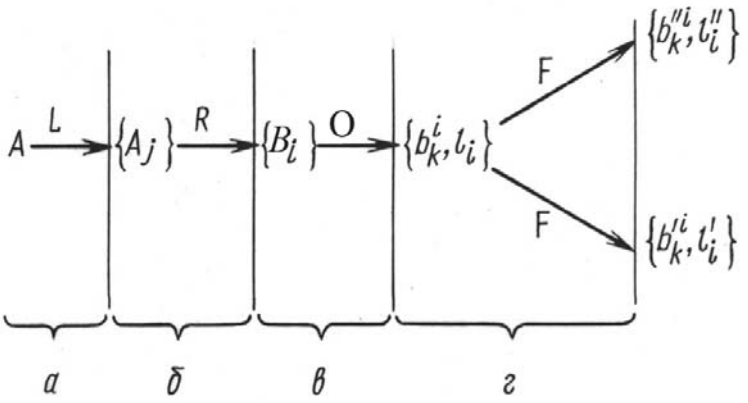

Рис. 2. Схема построения системы электронного документооборота

быть описан двумя состояниями — “истинно” или “ложно”. Множество $\{A_j\}$ не содержит информации о конкретной технологии решения задачи и является техническим заданием на разработку системы.

Для выбора оптимального метода решения задачи необходимо учесть особенности технологии, на основе которой обеспечивается решение поставленной задачи. В нашем случае особенности технологии учитываются характеристиками электронного представления информации.

Перед проведением дальнейшей детализации модели необходимо отметить, что многие задачи A_j могут иметь варианты решения с использованием других технологий, и возникает естественное желание перенести этот вариант решения на новую технологию. Использование идей старых технологий далеко не всегда удачно и часто требует пересмотра. Процесс пересмотра идеи реализации решения частной задачи называется реинжинирингом.

В результате проведения реинжиниринга (рис. 2, шаг *б*) получаем новое множество частных задач $\{B_i\} = R(\{A_j\}) : i \in [1..I]; j \in [1..J]; i, j \in Z$, где I — количество частных задач, полученных после реинжиниринга; R — отображение, описывающее процесс реинжиниринга.

Проведение реинжиниринга позволяет скорректировать исходную постановку задачи в части ее реализации с помощью электронного представления информации.

Для получения структуры системы электронного документооборота предлагается использовать методы лингвистического анализа и объектно-ориентированного анализа (обозначенного на рис. 2 как отображение O (шаг *в*)) [5]. В результате для каждой задачи B_i получим множество действий $\{b_k^i\} : i \in [1..I]; k \in [1..K]; i, k \in Z$, и правило взаимосвязи между ними l_i, которые совместно решают задачу B_i.

Множество $\{l_i\} : i \in [1..I]; i \in Z$ описывает уровень информационного обмена проектируемой системы (рис. 2, шаг *в*). В работе [3]

введен термин “информационная политика”, описывающий рассмотренное множество. В свою очередь, множество $\{b_k^i\} : i \in [1..I]$; $k \in [1..K]$; $i, k \in Z$ включает набор действий, которые необходимо совершить.

Для получения окончательной структуры системы необходимо сформулировать критерий F, определяющий типовое действие из множества $\{b_k^i\}$. Решающим моментом при определении типа действия (типовое или специализированное) является частота его использования для описания элементов множества $\{b_k^i\}$. Если частота использования превышает единицу, то элемент множества следует отнести к множеству типовых действий. Полученное множество типовых действий $\{b_k'^i\}$ определяет соответствующий уровень системы, $\{l_i'\}$ описывает взаимосвязи между ними. Соответственно, множество $\{b_k''^i\} = \{b_k^i\} \backslash \{b_k'^i\}$ является множеством специализированных действий (рис. 2, шаг *г*), $\{l_i''\}$ описывает взаимодействие между ними.

Таким образом, модель системы электронного документооборота можно представить в обобщенном виде:

$$F(O(R(L(A)))) = \{b_k'^i, l_i'\} \cup \{b_k''^i, l_i''\}, \qquad (1)$$

где A — цель проекта; F — критерий выделения типовых действий; O — отображение, описывающее применение методов лингвистического и объектно-ориентированного анализа; R — отображение, описывающее процесс реинжиниринга; L — отображение, описывающее применение методов логического структурирования.

Перейдем к рассмотрению вопроса обеспечения безопасности рабочего потока.

Математическая модель обеспечения безопасности рабочего потока. Цель данной работы — изложение формальной математической модели безопасной системы электронного документооборота и технологии ее разработки, вытекающей из этой модели. В основу построения модели положены нереляционные базы данных, дискреционная модель разграничения доступа к данным, отношения подчиненности и вложенности между объектами и репликация данных. Понятие безопасности электронного документооборота подразумевает невозможность получения субъектами непредусмотренных прав доступа и целостность данных. Электронный документооборот в модели рассматривается как совокупность нереляционных баз данных, взаимодействие между которыми заключается в асинхронном (т.е. выполняемом время от времени) реплицировании некоторых данных из одних баз данных в другие. Отношение вложенности ограничивает выбор данных для репликации; его сохранение между подчиненными

объектами обеспечивает целостность данных в защищенном документообороте. Доступ к данным каждой базы регулируется ее матрицей прав доступа, которая исключает возможность модификации данных, являющихся собственностью других баз данных. Свойство сепарабельности нереляционных баз данных позволяет однозначно идентифицировать ее собственные данные.

Любое подмножество множества {*ins, sel, del, upd*} называется правом доступа. Элементы в нем называются правами записи, чтения, стирания и обновления данных соответственно. Права *sel, del, upd* называются правами модификации данных. Прямоугольная матрица M, в которой столбцы поставлены во взаимно однозначное соответствие данным некоторой базы данных B, представленным их идентификаторами, а элементами являются права доступа, называется матрицей прав доступа к этой базе. В ней строки называются субъектами базы, а элемент $M[u,(o,q)]$ в строке u и столбце, соответствующем (o,q), – правом доступа субъекта u к данному (o,q). Если это право пустое, то говорят, что субъект u не имеет прав доступа к данному (o,q); если же какая-то из операций *ins, sel, del* или *upd* принадлежит к M$[u,(o,q)]$, то говорят, что субъект u имеет по отношению к данному (o,q) право записи, чтения, стирания или обновления соответственно. Все столбцы, соответствующие данным (o,q) с фиксированным o, называются столбцами для объекта o.

Допускаются следующие операции преобразования матрицы прав доступа M:

1) расширение права доступа какого-либо субъекта к какому-либо данному — в некоторый элемент матрицы вписываются дополнительные права;

2) сужение права доступа какого-либо субъекта к какому-либо данному — из некоторого элемента матрицы удаляются некоторые права;

3) создание объекта — введение в матрицу столбцов для нового объекта с пустым правом доступа в них всех субъектов;

4) уничтожение объекта — удаление из матрицы соответствующих столбцов;

5) создание субъекта — введение в матрицу новой строки с пустым правом доступа во всех столбцах;

6) уничтожение субъекта — удаление из матрицы соответствующей строки.

Если φ — любая из этих операций, то пусть $\varphi(\mathrm{M})$ — это матрица, которая получается как результат применения φ к матрице M. Если $\varphi = \varphi_1 \dots \varphi_n$ — произвольная последовательность указанных операций, то пусть $\varphi(\mathrm{M}) = \varphi_n(\dots(\varphi_2(\varphi_1(\mathrm{M})))\dots)$.

Если φ — произвольный набор из операций преобразования матрицы M и $\psi(\mathrm{M})$ — формула алгебры высказываний, построенная из элементарных высказываний вида $r \in \mathrm{M}[u,(o,q)]$, где r — любое из прав записи, чтения, стирания или обновления и $\mathrm{M}[u,(o,q)]$ — произвольный элемент матрицы M, то пара $k = (\psi(\mathrm{M}), \varphi)$ называется командой изменения прав доступа, заданных матрицей M. В ней ψ называется условием для выполнения команды. Результатом применения команды k к M является матрица $k(\mathrm{M})$ такая, что $k(\mathrm{M}) = \mathrm{M}$, если $\psi(\mathrm{M}) = л$, и $k(\mathrm{M}) = \varphi(\mathrm{M})$ в противном случае. Если $k = k_1 \ldots k_n$ — произвольная последовательность команд изменения прав доступа в M, то результат ее применения к M определяется как матрица $k(\mathrm{M}) = k_n(\ldots(k_2(k_1(\mathrm{M})))\ldots)$. Множество результатов применения к матрице M всевозможных конечных последовательностей команд из некоторого множества K команд изменения прав доступа обозначается $\mathrm{K}^*(\mathrm{M})$. Если M — это матрица прав доступа к базе B, то M и матрицы в $\mathrm{K}^*(\mathrm{M})$ называются состояниями доступа базы B.

В случае, когда информационные объекты принадлежат некоторой базе данных и доступ к данным в них ограничивается состоянием доступа этой базы, соответствующим образом ограничиваются и операции манипулирования с объектами *insert, select, delete, update*. Обозначенные соответственно как *ins, sel, del, upd* эти ограничения операций определяются следующим образом. Пусть u — произвольный субъект базы данных, μ — ее состояние доступа, o — ее объект, d — произвольное данное, однотипное с o, s — значение первичного ключа в d и P – логическое выражение, принадлежащее объекту o. Тогда для *nam* $\in \{$*ins, del, upd*$\}$:

$$nam(u,\mu,o,d) = \begin{cases} 0, & \text{если } nam \notin \mu[u,(o,s)]; \\ nam(o,d) \end{cases}$$

и, кроме того,

$$sel(u,\mu,o,P) = select(o - \{(o,q) : sel \notin \mu[u,(o,q)]\}, P);$$

$$del(u,\mu,o,P) = delete(o - \{(o,q) : del \notin \mu[u,(o,q)]\}, P);$$

$$upd(u,\mu,o,dP) = update(o - \{(o,q) : upd \notin \mu[u,(o,q)]\}, dP).$$

Распространим предыдущие операции на базы данных, введя следующие четыре операции: *Sel* — чтение, *Ins* — запись, *Del* — стирание и *Upd* — обновление. Каждая из них является функцией от пяти аргументов и записывается в форме $nam(u,\mu,\beta,o,\alpha)$, где nam — имя операции, $nam \in \{Sel, Ins, Del, Upd\}$, $\beta = (O,\rho,\chi)$ — база данных, u — субъект базы; β, μ — ее состояние доступа, O — объект в β и α —

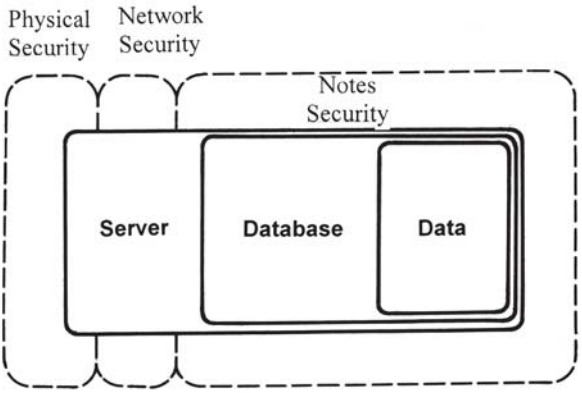

Рис. 3. Схема построения Lotus Security

способ адресации данных. Операция действует лишь на данные в базе, изменяя значения полей в O и сохраняя параметры ρ, χ. Результатом операции является база данных, однотипная с β. Первые две операции — *Sel* и *Ins* — выполняются как *sel* и *ins* независимо от отношения ρ и его характеристики χ в базе B, а именно:

$$Ins(u, \mu, \beta, o, d) = (O^*, \rho, \chi), \quad \text{где} \quad O^* = O - \{o\} \cup \{ins(u, \mu, \beta, o, d)\};$$

$$Sel(u, \mu, o, P) = (O^*, \rho, \chi), \quad \text{где} \quad O^* = \{sel(u, \mu, o, P)\}.$$

Операции же *Upd* и *Del* выражаются через *upd* и *del* в зависимости от параметров ρ и χ базы B.

Операции *Sel*, *Ins*, *Del*, *Upd* называются операциями манипулирования с базами данных, а три последние, т.е. *Ins*, *Del*, *Upd* — операциями модификации баз данных.

Необходимо отметить, что безопасность в Lotus-технологии осуществляется не только путем управления правами доступа и модификации информации, но и за счет многоуровневой схемы (рис. 3) [4]. Данная схема состоит из уровня сервера, уровня базы данных и уровня записей информации.

В дополнение к представленной схеме производитель предлагает уделить серьезное внимание сетевой безопасности Lotus-технологии при подключении ее к Internet сети. Сетевая безопасность осуществляется за счет использования межсетевых экранов (рис. 4).

Выводы. Применение предложенной модели защищенного документооборота позволяет:

1. Организовать поддержку обращений пользователей с учетом особенностей электронного представления документа;

2. Решать задачи управления данными в пределах одного уровня за счет многоуровневой архитектуры;

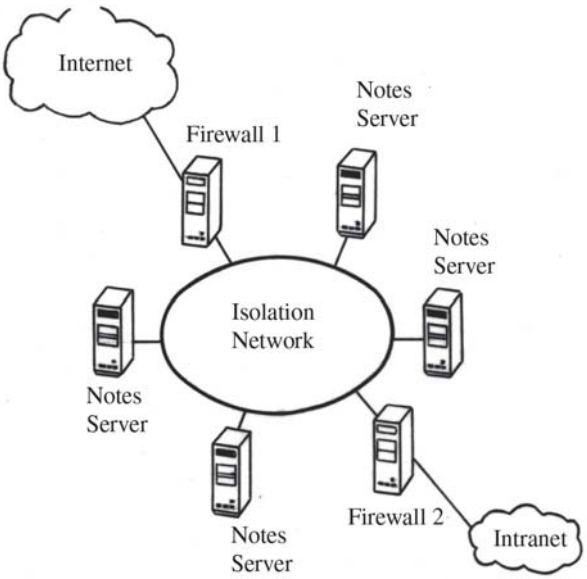

Рис. 4. Схема построения сетевой защиты Lotus

3. Эффективно организовать рабочие потоки в сети, полностью обеспечить жизненный цикл документации;

4. Эффективно организовать защиту рабочего потока как от внутренних, так и от внешних угроз.

Развитие технологии информационных сетей позволит в дальнейшем предоставить качественно новый вид услуг в области телекоммуникационной и сетевой поддержки различных распределенных процессов человеческой жизнедеятельности. Данный подход и технология могут быть востребованы при проектировании виртуальных предприятий, систем дистанционного обучения, электронной коммерции и других распределенных приложений.

СПИСОК ЛИТЕРАТУРЫ

1. Дейт К. Дж. Введение в системы баз данных, 6-е изд. / Пер. с англ. – К.; М.; СПб.: Издательский дом “Вильямс”, 2000. – 848 с.

2. Зегжда Д. П. Ивашко А. М. Основы безопасности информационных систем. – М.: Горячая линия–Телеком, 2000. – 452 с.

3. Коноллли Т. М., Бегг К. Е. Базы данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд. / Пер. с англ. – К.; М.; СПб.: Издательский дом “Вильямс”, 2000. – 1120 с.

4. Скутин А. А. Вопросы защищенности информации в корпоративной информационной системе “КОММЕРСАНТ” // Материалы 3-й межрегиональной науч. практ. конф. “Проблемы информационной безопасности общества и личности”. – Томск, ТУСУР, 2001. – С. 181–184.

Статья поступила в редакцию 7.12.2005

Николай Викторович Медведев родился в 1954 г., окончил в 1977 г. МВТУ им. Н.Э. Баумана. Канд. техн. наук, зав. кафедрой "Информационная безопасность" МГТУ им. Н.Э. Баумана. Автор 45 научных работ в области исследования и разработки защищенных систем автоматической обработки информации.

N.V. Medvedev (b. 1954) graduated from the Bauman Moscow Higher Technical School in 1977. Ph. D. (Eng.), head of "Data Safety" department of the Bauman Moscow State Technical University. Author of 45 publications in the field of study and development of secured systems of automatic data processing.

Георгий Александрович Гришин родился в 1979 г., окончил МГТУ им. Н.Э. Баумана в 2003 г. Доцент кафедры "Информационная безопасность" МГТУ им. Н.Э. Баумана. Автор 4 научных работ в области информационной безопасности.

G.A. Grishin (b. 1979) graduated from the Bauman Moscow State Technical University in 2003. Ph. D. (Eng.), assoc. prof. of "Data Safety" department of the Bauman Moscow State Technical University. Author of 4 publications in the field of the information safety.

Денис Петрович Кацыв родился в 1977 г., окончил в 1999 г. Московский автомобильно-дорожный институт (государственный технический университет). Вице-президент АТК "МиаОйл". Автор пяти научных работ в области автоматизированных систем обработки информации и управления.

D.P. Katsyv (b. 1977) graduated from the Moscow Automobile and Road Institute (state technical university), vice-president of "ATK "MiaOylz". Author of 5 publications in the field of automated systems of data processing and control.

УДК 621.391:53.08

В. Я. Колючкин, А. С. Мачихин

МОДИФИЦИРОВАННЫЙ ИТЕРАЦИОННЫЙ АЛГОРИТМ ВОССТАНОВЛЕНИЯ ИЗОБРАЖЕНИЙ

Предложен модифицированный итерационный алгоритм с ограничениями на положительность решения и нормировкой на область допустимых значений для восстановления изображений, смазанных и искаженных расфокусировкой. Для обеспечения сходимости и стабилизации решения на каждой итерации используется фильтр Винера. Приведены результаты апробации алгоритма на реальных изображениях.

Задача восстановления изображения связана с компенсацией линейных искажений, вносимых системой регистрации. Дополнительные линейные искажения могут быть смазанными или вызваны рас-