

ИНФОРМАЦИОННЫЕ СИСТЕМЫ ЭЛЕКТРОННОЙ ИДЕНТИФИКАЦИИ АВТОМОБИЛЬНЫХ ТРАНСПОРТНЫХ СРЕДСТВ

Андреев А. Я., Гинько А. Г.

БНТУ, Минск, andreev@bntu.by, andreev_ay@mail.ru

БНТУ, Минск, ginko.ag@yandex.by

Эффективное использование средств электронной идентификации невозможно без их интеграции в системы управления предприятием или технологическими процессами доставки грузов.

Характерные для систем электронной идентификации большие объемы разнообразных по направлению информационных потоков порождают значительное количество данных. Для принятия правильных и эффективных решений эти данные должны быть своевременно получены, обработаны и доведены до соответствующих исполнителей и руководителей. Для решения таких непростых задач используются современные информационные технологии.

Современные технологии обработки данных рассчитаны на применение в самых разнообразных условиях. Для обработки данных используются компьютерные программы, объединяемые в большой класс транзакционных систем.

В простейшем случае информационная система может быть реализована на отдельном компьютере. Это приемлемое решение для выполнения простых задач одним пользователем. Если в работе системы участвуют несколько пользователей на нескольких компьютерах, то требуется вручную постоянно выполнять актуализацию данных, чтобы изменения, выполняемые одними пользователями, становились доступными другим пользователям, что снижает оперативность и надежность работы системы и делает недоступной обработку данных в режиме реального времени.

В системах доставки грузов такая технология работы приемлема в очень ограниченных вариантах. Например, работа бухгалтера в небольшой компании, обработка отдельной складской операции и т. п.

Проблему работы нескольких пользователей информационной системы можно решить, если они будут работать с одними и теми же данными, а их компьютеры будут объединены в сеть. Для совместной обработки данных в сети может использоваться несколько сетевых архитектур.

Архитектура файл-сервер выделяет в сети компьютеры, которые могут использоваться как серверы и (или) клиенты. На серверах выделяются общие ресурсы (файлы, принтеры и т. п.), а клиентские компьютеры могут использовать эти ресурсы.

Поток информации, требующей обработки, постоянно растет, и в компьютерных сетях все чаще используется архитектура клиент - сервер, где для обработки данных используется мощный сервер данных, а для представления и изменения нужных данных – пользовательское приложение, которое работает на клиентском компьютере. Таким образом, основная, наиболее ресурсоемкая работа с данными выполняется в месте их хранения, а на пользовательский компьютер передаются только данные, необходимые для получения информации или требующие изменения.

Если архитектура файл-сервера отличается от локальной обработки данных на одном компьютере в основном местом хранения данных и необходимостью обеспечения доступа к данным нескольких пользователей, то в архитектуре клиент-сервер обработка, хранение и доступ к данным обеспечивается сервером базы данных. Последнее решение имеет следующие преимущества:

- независимость данных от пользовательского приложения, которое может быть реализовано на различных языках программирования;
- возможность централизованного управления бизнес-правилами обработки данных;

- возможность распределения ресурсов между несколькими серверами;
- более надежная система обеспечения секретности и безопасности данных;
- легкая организация доступа к гетерогенным и распределенным данным.

В последнее время для работы с данными все шире применяются технологии World Wide Web (WWW). Эта технология в глобальных масштабах реализуется в сети Интернет и основывается на поиске данных с помощью гипертекстовых ссылок, которые обеспечивают доступ к различным файлам с данными.

Обычно одноуровневые прикладные программы обращаются к таблицам с данными на сервере так, как это происходит в технологии, основанной на файл-сервере. Это означает, что типичный запрос прикладной программы клиента выбирает строки непосредственно из таблиц, относящихся к основной структуре данных. Это происходит, когда прикладная программа разработана на основе локального варианта в архитектуре ISAM (индексно-последовательный метод доступа). Для доступа к данным открывается таблица, выбирается требуемый индекс, и ищутся строки с необходимыми данными. Бизнес-логика встроена в каждое пользовательское приложение, использующее данные сервера.

Этот принцип доступа к данным и их поиску вызывает ряд проблем:

- при изменении структуры данных или правил обработки данных логика работы приложения должна быть разработана заново;
- если запросы не ограничивают размер возвращаемых наборов результатов, масштабирование прикладной программы может оказаться непростой задачей. Это означает, что добавление дополнительных пользователей может вызвать ухудшение эффективности работы системы, в то время как прикладная программа успешно работала с небольшим количеством пользователей;
- особенно усложняется работа над большими проектами, так как переписывание большого объема кода и перекомпилирование проекта очень трудоемки. Это означает, что работа для групп, программирующих большие проекты, становится особенно трудной;

В одноуровневых прикладных программах клиентское приложение выполняет много функций, требования к производительности рабочей станции достаточно высоки.

Двухуровневое приложение предполагает размещение бизнес-логики на сервере. В этом случае клиентское приложение содержит только средства интерфейса пользователя, а алгоритм обработки данных располагается на сервере. При использовании данных сервера несколькими приложениями существенно облегчается контроль и изменение правил обработки данных, так как их изменение не требует вмешательства в клиентские приложения, которые могут быть установлены на большом количестве рабочих станций. При увеличении количества пользователей трудоемкость поддержки приложений увеличиваться не будет.

Построение двухуровневых приложений требует переноса кода, управляющего данными, на сервер. Вместо таблиц клиентское приложение должно иметь дело с логическими объектами, а для обновления данных выполнять внешние (удаленные) процедуры. Следует лишь обратить внимание, что в большинстве случаев это не является чисто механическим действием.

Перенос правил обработки данных на сервер позволяет также снизить нагрузку на клиентское приложение. Требования к рабочей станции могут быть снижены. Вместе с тем при наличии высокопроизводительного многопроцессорного сервера можно повысить производительность обработки данных, так как такие серверы БД, как MS SQL Server, способны к выполнению одновременно нескольких хранимых процедур, причем каждая может выполняться в индивидуальном потоке.

Одним из наиболее современных подходов к построению двухуровневых прикладных программ является использование компонентов ActiveX, которые могут быть выполнены через вызовы расширенных хранимых процедур. Эти компоненты могут, используя свои свойства и методы, выполнять специфические задачи обработки данных. При изменении

бизнес-логики достаточно использовать другой компонент, оставив без изменения набор свойств и методов, доступный пользовательскому приложению.

Формирование прикладных программ из компонентов имеет огромную привлекательность, но отсутствие инструментария интегрирования объектов, созданных различными разработчиками, в одну прикладную программу создает подчас непреодолимые трудности в реализации такого подхода. Появившаяся в последнее время Component Object Model (COM) предназначена для ликвидации этих трудностей.

Современные информационные системы, следуя за тенденциями в экономике, становятся все более сложными, в процессе функционирования могут охватывать несколько предприятий, что вызывает изменение данных на нескольких серверах. Когда проект охватывает несколько серверов, в приложении появляется необходимость использования дополнительных уровней. В этом случае такой сервер, как Microsoft Transaction Server (MTS), может играть роль координатора работы объектов, выполняя множество сложных операций интегрирования, координации и взаимодействия, которые оказываются за пределами действия диспетчера распределенных транзакций в SQL Server. MTS не только управляет транзакциями между серверами, но также выполняет и функции объединения потоков и межобъектного взаимодействия.

Учитывая, с одной стороны, современные требования к качеству логистического обслуживания, необходимость получения большого количества данных из внешних источников, а с другой стороны, слишком большие затраты на полностью интегрированную обработку данных в масштабе реального времени в глобальной компьютерной сети необходимо стремиться соблюдать три основных принципа:

- данные должны быть доступны. Весьма эффективно создание веб-сервера, доступ к которому внутри организации может быть обеспечен через внутреннюю сеть интранет, а для сотрудников, работающих вне организации, – через Интернет. Средства защиты данных должны быть более конкретными. Вместо запрета доступа к таблицам лучше защищать конкретные колонки и записи данных;
- данные должны быть точными. При добавлении данных в информационную систему они должны подвергаться автоматической логической проверке;
- данные должны быть понятны. Все операции с конкретными данными (кроме просмотра) лучше выполнять в одном и том же месте, в основном там, где они были впервые введены в компьютер.

Функционирование такой информационной среды поддерживается специальным программным обеспечением, которое называется средствами групповой работы. Функционируя совместно с почтовыми программами и серверами баз данных, такие средства обеспечивают коллективный доступ к документам, поддерживают деятельность дискуссионных групп, календарные функции, контактную информацию, временные графики и перечни задач для групп пользователей.

В системах управления обработка данных, полученных средствами электронной идентификации, должна строиться таким образом, чтобы использовать их основные преимущества в получении достоверной информации в режиме реального времени. С этой целью в информационной системе формируют функционально законченные блоки, которые позволяют автоматизировать отдельные бизнес-процессы. Основной поток данных поступает в систему управления от устройств идентификации. На основании именно этой информации принимаются решения о ходе выполнения плановых заданий и графиков выполнения работ и необходимости коррекции управляющих воздействий.

В этом случае задача проверки соответствия предельным ограничениям габаритов и массы автомобильных транспортных средств (АТС) выполняется путем считывания информации с соответствующих датчиков и идентификации номерного знака автомобиля.

Программное обеспечение для обработки данных электронной идентификации должно обладать следующими свойствами:

- иметь библиотеку протоколов для обеспечения интерфейса нижнего уровня с устройствами RFID. Это позволяет интегрировать в одну систему устройства различных производителей, использовать логические контроллеры, считыватели штрих-кодов и т. п.;
- обеспечивать автоматическую настройку различных стандартов и протоколов для получения данных от устройств RFID без необходимости их детального изучения разработчиками информационных систем;
- позволять быстро разрабатывать модули бизнес-логики с помощью визуальных и программных инструментальных средств;
- иметь возможность встраивать разрабатываемые приложения RFID в технологические и управляющие информационные системы предприятия;
- предоставлять средства контроля информационных потоков в системе;
- обеспечивать генерацию сигналов устройств RFID для тестирования и настройки программного обеспечения;
- позволять легко модернизировать разработанные приложения при расширении систем RFID или замене используемых устройств.

В современных условиях, когда технологии электронной идентификации все шире используются в системах доставки продукции, разработчики программного обеспечения предлагают готовые решения, что позволяет отказаться от дорогостоящей заказной разработки специализированного приложения.

Так, известная фирма по разработке приложений для обработки данных Sybase предлагает комплексное решение для автоматизации процедур электронной идентификации под названием RFID Enterprise.

Приложение RFID Enterprise позволяет пользователям собирать, отслеживать и интегрировать данные с датчиков RFID в системы управления предприятием, а также разрабатывать специализированные приложения для их локального использования.

В состав приложения RFID Enterprise входит несколько компонентов. RFID Edgeware предлагает интерфейс системного управления различными устройствами электронной идентификации – считывателями, сканерами штрих-кодов, принтерами и т. п. Другой компонент автоматизирует ввод информации электронной идентификации в базу данных, а также обеспечивает построение логической модели базы данных для сопутствующей информации. Еще один компонент приложения отвечает за реализацию бизнес-процесса RFID, его интеграцию в систему управления предприятием и мониторинг данных. Он предлагает визуальную среду разработки бизнес-процесса с возможностью его отображения и контроля. И наконец, инструментарий вывода данных RFID на печать позволяет готовить стандартные и специализированные отчеты.

Фирма Manhattan Associates выбрала для развития своего пакета EPC Manager другой путь, поставив во главу угла возможности инвентаризации. В обновленной версии появились функции централизованного слежения за грузами, входящими и исходящими заказами, оформления контейнерных перевозок. Расширена и поддержка соответствия действующим требованиям XML. Главное назначение данного приложения – автоматизировать получение, отслеживание и оценку электронного кода продукции.

Средства обработки данных электронной идентификации включили в свои последние версии приложений и такие известные фирмы-производители программного обеспечения для управления цепочками поставок, как OATSystems и Catalyst International.

Развитие и внедрение средств электронной идентификации в системы управления автотранспортной деятельностью будет способствовать повышению эффективности их функционирования. Появится возможность приблизить управляемость транспортных систем к промышленным за счет получения достоверной информации о состоянии объектов управления в режиме реального времени. Это расширит возможности практического использования современных логистических технологий, когда точная и своевременная

информация является залогом успешной совместной работы нескольких операторов в цепочке доставки грузов или пассажиров.

Список литературы:

1. Горев, А. Э. Информационные технологии на транспорте. Электронная идентификация автотранспортных средств и транспортного оборудования: учеб. пособие для студентов специальностей 190701 – организация перевозок и управление на транспорте, 190702 – организация и безопасность движения (автомобильный транспорт) / А. Э. Горев; СПбГАСУ. – СПб., 2010. – 96 с.
2. Андреев, А. Я. Информационные системы в дорожном движении / А. Я. Андреев // Наука – образованию, производству, экономике : материалы 14-й Международной научно-технической конференции. - Минск : БНТУ, 2016. - Т. 3. - С. 147-148.