Архитектура Домино

- Основные принципы построения платформы
- Принципы разработки в среде Домино
- Технологические особенности и средства разработки
- Модель данных
- Особенности работы с БД
- Средства разработки
- С точки зрения пользователя

Основные принципы построения платформы

Технологически платформа Домино состоит из следующих основных элементов:

- ядро платформы, включающее среду исполнения, средства разработки приложений, набор базовых объектов, функций, процедур и т.п.;
- библиотеки прикладных решений
- отдельные модули и программы

Кратко о внутренней реализации платформы:

- написана на C/C++ (MS Visual C++)
- собственная компонентная архитектура
- создана и развивается исключительно программистами фирмы Софт-Вест и, соответственно, не зависит от сторонних разработчиков

Платформа предназначена для быстрой разработки приложений и их последующей адаптации к специфике предприятия. Имеющиеся средства обеспечивают не только высокую скорость первоначальной разработки продукта, но и его быструю адаптацию в дальнейшем, в соответствии с изменяющимися требованиями предприятий.

Формализация модели данных, процесс проектирования и методология бизнес-решений позволяют сторонним разработчикам быстро разобраться в логике типовых процессов и построить на их основе свои приложения. Также это помогает системным администраторам достаточно быстро понять и освоить принципы работы программы. Обучение конечных пользователей (менеджеры, продавцы, экономисты и т.д.) не занимает много времени, поскольку процессы стандартизованы.

Разработчик прикладного приложения избавлен от от необходимости заниматься тонкостями взаимодействия с базой данных, сложностями транзакционных блокировок, нюансами программирования пользовательских интерфейсов. Платформа берет на себя

эти и многие другие технические вопросы, позволяя разработчику сконцентрироваться на решении прикладной задачи. При этом код платформы открыт для изучения и понимания. Сторонним разработчикам доступны все те же инструментальные средства, что и разработчикам фирмы Софт-Вест. Они могут ознакомиться со всеми деталями структуры и логики прикладного решения и, при необходимости, внести изменения.

Прикладное решение рассчитано на возможность изменения и адаптации в соответствии со спецификой конкретного предприятия как силами сотрудников предприятия, так и сторонними специалистами.

Для постоянного развития решения, использования новых тенденций управления, учёта, изменений законодательства и растущих требований пользователей имеется технология обновления решений без потерь изменений, ранее внесённых
для адаптации решения к специфике предприятия.

Принципы разработки в среде Домино

Разработка приложения в среде Домино отличается от разработки в универсальных системах тем, что приложение Домино разрабатывается в рамках проблемно-ориентированной модели с применением метаданных.
При обычном программировании необходимо описать целый ряд сущностей и способы их регистрации и взаимодействия. При проектировании на платформе Домино разработчик использует объекты (сущности), заранее описанные в платформе. Это товары, документы, каталоги, партнёры, регистры и т.д. Каждый из этих объектов имеет внутренние классификацию и типизацию.

Выделение объектов, типов и классов определялось следующими соображениями:

- понятны назначение и роль объекта в модели данных
- типизация и классификация должны обеспечивать унификацию разработки прикладных решений
- понятны способы применения объектов, классов и типов

Полученные наборы объектов, классов и типов описаны в платформе и все вместе составляют метаданные, которые в значительной степени определяют структуру проектирования приложения. Каждое приложение Домино основывается и опирается на структуру метаданных. При этом, непосредственно в приложении можно добавить свои объекты, типы и классы.

Описание объекта включает такую информацию, как используемая таблица базы данных, поля таблицы, типовые интерфейсные элементы, наборы прав, которые будут использоваться в системе разграничения прав доступа. На основе имеющихся объектов разработчик может создать свои собственные объекты. Эти новые объекты с момента создания будут наследовать свойства от объекта-родителя. При необходимостиразработчик может указать особенности, которой должна обладать новая сущность вотличие от родителя.

В платформе уже заложена мощная функциональность, которая позволяет, не добавляя никакой новой (по сравнению с заложенной в платформе) функциональности, быстро и удобно поддерживать объекты на уровне прикладного решения.

Имеются следующие объекты:

- Каталоги
- Товары
- Партнёры (подразделения)

Это справочники, имеющие общие свойства: внутренняя идентификация объекта в БД, необходимость поддержки иерархии и группировки элементов, участие в хозяйственных операциях предприятия. Перечисленные свойства поддерживаются на уровне платформы.

Справочники - это товары, поставщики, товарные группы и категории, подразделения, банки и т.п.

В справочниках также можно хранить сведения о значениях различных величин, которые сами по себе не имеют объектной семантики. Такими значениями могут быть, например, курсы валют на определённую дату.

- Документы

Документы - это счета, накладные, заказы и т.п. Посредством документов фиксируются различные события, происходящие в хозяйственной жизни организации. Важными свойствами документа являются его привязка ко времени и месту создания/применения/хранения (подразделение). Документ может иметь состав - строки.

Обычно в бизнес-приложениях разделяют два состояния документа: черновик и

подтверждённый (акцептованный) документ. Если документ не подтверждён, то считается, что документ ещё не готов, находится в процессе ввода, проверки, согласования и т.п. Только после подтверждения документ учитывается в различных системных регистрах.

- Регистры

Регистр "отвечает" за учёт движений ресурсов (финансов, товаров, материалов и т.д.). В регистрах хранится информация о поступлении и расходе тех или иных ресурсов. Заложенная в платформу функциональность этих регистров позволяет получить остатки ресурсов на определённый момент времени, рассчитать итоги и т.д.

- Проводки и сальдо

Система двойной записи бухгалтерского учёта представляет собой отдельную модель учёта со своей спецификой, поэтому для хранения данных выделены отдельные классы объектов. Причём механизмы, связанные с данными объектами, никак не навязывают разработчику собственно принципов ведения бухгалтерского учёта.

Технологические особенности и средства разработки

Платформа Домино для всех прикладных решений, независимо от отраслевой специфики, обеспечивает:

- возможность использования системы от локального компьютера до сотен пользователей в локальной сети;
- возможность использования архитектуры клиент-сервер;
- возможность развёртывания работы на нескольких территориально удалённых точках с периодическим обменом
  информацией.

Важно, что все эти возможности могут быть реализованы без переработки прикладного решения. Разработчик тиражных решений может выпускать одно и то же приложение для использования в организациях самого разного масштаба. Это может быть и однопользовательский вариант для персонального использования в совсем небольшой организации, и многопользовательская клиент-серверная версия, которая обеспечивает эффективную работу и
надёжное хранение информации при наличии десятков и сотен одновременно работающих пользователей, и распределённый вариант работы с данными для территориально разнесённых филиалов и подразделений, не связанных локальной сетью.

В варианте клиент-сервер реализована двухуровневая архитектура. Программа, работающая у пользователя, взаимодействует с сервером приложений, который при необходимости обращается к системе управления базой данных (СУБД). При этом физически сервер приложений и СУБД могут быть установлены как на одном компьютере, так и на разных, что позволяет администратору при необходимости распределять нагрузку между серверами.

Модель данных

Модель базы данных Домино, с которой работает разработчик прикладного решения, имеет ряд особенностей, отличающих её от классических моделей СУБД (например, основанных на реляционных таблицах), с которыми имеют дело разработчики в универсальных системах.

Модель базы данных полностью абстрагирована от формата хранения.

Разработчик прикладного решения может манипулировать объектами, хранящимися в базе данных и ссылками на эти объекты, а также обращаться к данным, используя табличную парадигму.

Особенности работы с БД

Разработчик прикладных решений работает непосредственно с платформой Домино, при этом он может:

- описывать структуры данных в редакторе приложений;
- манипулировать данными с помощью объектов встроенного языка;
- составлять запросы к данным.

Программный слой ядра платформы обеспечивает операции исполнения запросов, описания структур данных и манипулирования данными, транслируя их в соответствующие команды СУБД.

Для прикладного разработчика реализована объектная модель работы с базой данных. Разработчик оперирует объектами встроенного языка. Обращения к объекту, например, документу, происходят как к единому целому. Платформа обеспечивает сохранение целостности объектов, кэширование объектов, вызов соответствующих обработчиков событий и т.д.

В платформе реализован механизм работы с территориально распределёнными информационными базами. При этом в каждом из удалённых подразделений поддерживается автоматический обмен и синхронизация данных на уровне экземпляров объектов. Интеллектуальный механизм репликации позволяет переносить целостные объекты, описывать
правила переноса объектов между базами. Переносятся только изменённые данные. Система обеспечивает достаточно высокую устойчивость к сбоям и защиту от потерь части информации при передаче.

Средства разработки

Редактор приложений - это инструментальное средство для разработки приложений на платформе Домино.

Редактор позволяет описать:

- объекты метаданных (сущности, с которыми работает прикладное решение)
- пользовательские интерфейсы (меню, панели инструментов, списки, формы)
- систему прав доступа
- процедуры обработки данных
- отчётные формы

Средства редактора ориентированы не только на профессиональных программистов, но и на широкий круг пользователей. При их разработке особое внимание уделялось простоте освоения и использования.

Детальное описание бизнес-логики программируется на встроенном языке редактора приложений. При этом прикладные решения не программируются (кодируются) целиком, большая их часть описывается в виде структур метаданных. Встроенный язык редактора предназначен в первую очередь для программирования бизнес логики в контексте объектной модели Домино. На нём программируются обработчики различных событий, изменяющих состояние объектов системы, например, обработчики команд пользователя, обработчики проведения документов и т.д. Очень существенным моментом является то, что особенности построения языка напрямую соответствуют модели проектирования структур данных, реализованной в Домино. То, что вся разработка приложения ведётся на основе использования стандартных объектов системы, позволяет разработчику прикладных решений применять соответствующие им объекты встроенного языка, имеющие большой набор функций и высокую гибкость.

Средства редактора позволяют:

- задать автоматическую связь форм и элементов управления данными без специального программирования.

- описать поведение элементов, определяемое данными
- перечислить специализированный набор элементов управления
- указать поля ввода с функциональными кнопками (например, выбор, очистка, открытие значений)
- описать эффективные и удобные динамические списки для просмотра информации из базы данных, с поддержкой различных вариантов фильтрации и т.д.
- создать современный эргономичный дизайн пользовательских интерфейсов
- применить механизм автоматического изменения расположения и размеров элементов, который обеспечивает автоматическое выравнивание всех элементов управления, поддерживает разделители внутри окна и тем самым позволяет при разработке формы для объекта не заботиться о том, как она будет отображаться при изменениях размеров окон
- использовать систему стилей оформления

Средства разработки с одной стороны обеспечивают высокую гибкость разработки приложения, с другой стороны позволяют минимизировать объем ручного кодирования.

Многообразие вариантов учёта в зависимости от ситуации, вида деятельности и особенностей работы конкретного предприятия очень велико, и заложенные алгоритмы не всегда подходят. Для таких случаев имеется встроенный скриптовый язык. Он не является прямым аналогом какого-либо языка программирования. Скриптовый язык позволяет разработать необходимые процедуры и функции и заменить ими типовые конструкции.

В редакторе имеется возможность применения как классического генератора отчётов, ориентированного на подготовку исключительно печатных форм, так и создания интерактивных документов. Во втором случае пользователь может не только распечатать отчёт, но и работать с ним практически так же, как с экранной формой. Изменять параметры отчёта, перестраивать его и т.д.

Поддерживаются интерактивные права доступа. Механизм задания интерактивных прав позволяет разработчику вместо кропотливого программирования интерфейсов (закрытия отдельных пунктов меню, кнопок, и т.д.) ограничить интерактивный доступ пользователя к определённым объектам системы, но при этом сохранить возможность программного доступа к ним в случае, если разрешённые действия пользователя вызывают соответствующую

обработку.

Программная проверка ролей и прав позволяет разработчику развивать систему разграничения доступа в соответствии со спецификой конкретного прикладного решения, создавая (программируя) те проверки, которые не предоставляются системой автоматически.

Имеется технология обновления приложения с сохранением внесённых пользователем изменений.

Предусмотрена групповая разработка решений, когда приложение одновременно создают несколько разработчиков.

Домино рассчитано на широкую интеграцию с разными системами. Применяются различные варианты файлового обмена, протоколы http.

С точки зрения пользователя

С точки зрения конечного пользователя Домино - это, в первую очередь, набор прикладных решений, построенных на единой технологической платформе.

Продукты системы ориентированы на решение следующих классов задач:

- Анализ и управление эффективностью работы предприятия. Эти функции системы предназначены для решения задач
  руководителя предприятия и на управленцев, отвечающих за рентабельность бизнеса и его развитие. Их назначение - обеспечить руководителей актуальной информацией, необходимой для оценки ситуации и принятия решений. Например, анализ сбыта товаров.
- Учёт и управление оперативной деятельностью предприятия. Эта функциональность решает задачи менеджеров и
  работников, непосредственно занимающихся торговой, производственной деятельностью или деятельностью в
  области оказания услуг. Она обеспечивает эффективную ежедневную работу предприятия: подготовку документов,
  управление движением товаров, управление производственными запасами и выпуском продукции, приём заказов и контроль их исполнения и т.д.

Современные тенденции развития экономических систем и мировой опыт показывают, что не может существовать единый подход к решению проблем автоматизации. У предприятия должна быть свобода выбора из типовых вариантов и последующая адаптация варианта к специфике своего бизнеса.

Домино предоставляет возможность автоматизации как за счёт внедрения отдельных прикладных решений, которые будут работать автономно или интегрироваться с использованием различных механизмов информационного обмена, так и за счёт использования комплексных решений. Использование обособленных решений проще и

эффективнее, если отдельные задачи автоматизации на предприятии мало пересекаются. Комплексные решения эффективнее при сильной увязке различных задач автоматизации и готовности предприятия к формированию единого информационного пространства. Разумеется, на одном предприятии могут применяться и комплексные решения (например, для автоматизации основной деятельности), и обособленные программы (для вспомогательных или независимых задач).