



С.Г. Нуралиев

**АРХИТЕКТУРА
"1С:ПРЕДПРИЯТИЯ"
КАК ПРОДУКТ
ИНЖЕНЕРНОЙ МЫСЛИ**



С.Г. Нуралиев

**АРХИТЕКТУРА
"1С:ПРЕДПРИЯТИЯ"
КАК ПРОДУКТ
ИНЖЕНЕРНОЙ МЫСЛИ**

Москва
2009

Нуралиев С.Г.

Архитектура "1С:Предприятия" как продукт инженерной мысли.

В статье руководителя отделения разработки экономических программ фирмы "1С" Сергея Нуралиева подробно рассматриваются реализованные в системе "1С:Предприятие" архитектурные и технологические инновации, которые в совокупности определили ряд совершенно новых технологий разработки бизнес-приложений и новых качеств самих этих приложений.

Статья опубликована в газете PC Week/Russian Edition (издается по лицензии международного издательского дома Ziff-Davis Media Inc.), NN 46, 47 и 48 за 2004 г. Данный вариант статьи публикуется с разрешения PC Week/RE. В него внесен ряд дополнений, по техническим причинам не вошедших в газетный вариант, а также добавлены иллюстрации.

Фирма "1С" благодарит редакцию PC Week/RE за сотрудничество и помощь при подготовке статьи к публикации.

(с) ЗАО "1С", 2004 г

(с) PC Week/RE, 2004 г.

Архитектура "1С:Предприятия" как продукт инженерной мысли

ОГЛАВЛЕНИЕ

□	Взгляд с другой стороны	4
□	За что боремся	4
□	Платформа и бизнес-приложения	5
□	Метаданные – способ описания бизнес-приложения	7
□	Построение приложения на основе модели	8
□	Управление данными	8
□	Стандартные прототипы прикладных объектов	13
□	Прикладные объекты и механизмы	15
□	Высокоуровневая модель интерфейса	18
□	Интеллектуальные механизмы подготовки отчетов	21
□	Построение распределенных и интегрированных информационных систем	24
□	Поставка и обновление прикладных решений	25
□	А также...	26
□	Подводя итоги	28

Архитектура "1С:Предприятия" как продукт инженерной мысли

Взгляд с другой стороны

Сегодня практически каждый второй продукт человеческой деятельности, предлагаемый покупателю, имеет отношение к высоким технологиям. А это значит, что он не только удовлетворяет конкретные потребности человека, но и является воплощением научной и инженерной мысли. Очевидно, что в некоторых случаях заложенные в продукт идеи и достижения, полностью определяют его сущность и потребительскую ценность.

В этой статье мы решили сконцентрироваться не на описании возможностей продукта "1С:Предприятия", а именно на технологических инновациях, которые в совокупности определили ряд совершенно новых технологий разработки бизнес-приложений и новых качеств самих этих приложений.

Сначала сделаем небольшое замечание по терминологии. Эта статья не ориентирована исключительно на тех, кто имеет дело с системой программ "1С:Предприятие". Как нам кажется, она может быть интересна многим специалистам, интересующимся развитием технологий экономических и корпоративных приложений, независимо от того с какими продуктами они работают. Поэтому в данной статье мы постараемся лишь в минимальном объеме использовать терминологию, специфическую для "1С:Предприятия". В некоторых случаях мы будем использовать для близких понятий различные термины, приводя их аналоги и необходимые разъяснения.

Небольшое уточнение требуется по самому названию продукта. "1С:Предприятие" это система программ, включающая платформу и набор построенных на ее основе бизнес-приложений (прикладных решений), предназначенных для множества отраслей и предприятий разного масштаба. В данной статье мы будем говорить в основном о платформе, на которой строятся бизнес-приложения.



За что боремся

В начале хочется кратко определиться с тем, что считать основной целью создания таких технологий как платформа "1С:Предприятие". Если посмотреть на историю развития компьютеров и программирования, то можно заметить, что, наряду с повышением производительности, увеличением объемов обрабатываемой информации, повышением эргономичности и т.д. прослеживается достаточно четкое стремление к повышению уровня абстракции программных систем. Эта тенденция является в какой-то степени уникальным свойством, присутствующим именно компьютеру, тогда как в других областях человеческой деятельности стратегические цели развития носят более утилитарный характер. Проследить историю повышения уровня абстракции очень легко – начиная от программирования на уровне соединительных шнуров, к машинным кодам, ассемблеру, структурным языкам программирования и т.д. Каждый этап знаменовал повышение уровня абстракции взаимодействия человека (и разработчика, и конечного пользователя) с компьютером.

Так вот, основная задача платформы "1С:Предприятие" заключается, прежде всего, в реализации данного подхода при разработке и использовании бизнес-приложений. Разумеется, заодно решаются и традиционные задачи, связанные с производительностью, эргономикой, функциональностью и т.д. Но именно повышение уровня абстракции позволяет переи-

ти от технических и низкоуровневых понятий к более содержательным и высокоуровневым, а значит приблизить их к языку пользователей и специалистов в предметной области. В конечном итоге это позволяет значительно ускорить и унифицировать как саму разработку системы, так и ее сопровождение.

Платформа и бизнес-приложения

Пожалуй, наиболее концептуальными в архитектуре "1С:Предприятия" являются само наличие платформы и понятие бизнес-приложения.

Разумеется, любая фирма-разработчик бизнес-софта имеет достаточно мощный инструментарий и набор технологий, с помощью которых строит свои решения. Однако, при разработке "1С:Предприятия" мы изначально ориентировались на создание полнофункциональной, целостной платформы, которая будет использоваться для построения самых разнообразных бизнес-приложений не только самой фирмой "1С", но и множеством других компаний-разработчиков, знакомых со спецификой тех или иных отраслей. Вот почему платформа изначально создавалась как тиражируемый продукт, включающий как все необходимые технологии для эксплуатации бизнес-приложений, так и инструменты для их разработки и модификации.

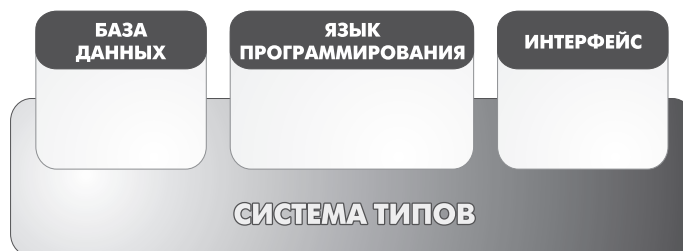
В "1С:Предприятии" было введено четкое разделение на платформу и бизнес-приложение. Платформа представляет собой так называемый framework, в котором функционирует бизнес-приложение. Мы не смогли найти точного перевода этого слова на русский язык. С одной стороны framework можно считать фундаментом для построения приложений, а с другой – средой исполнения. Кроме того, платформа содержит, разумеется, и инструментарий, необходимый для разработки, администрирования и поддержки бизнес-приложений. Такое приложение является самостоятельной сущностью и может выступать в качестве отдельного программного продукта, но полностью опирается на технологии платформы.



Заметим, что понятие платформы и платформенно-ориентированного построения приложений сейчас является общепринятым и трактуется гораздо шире, чем просто способность работать в определенной операционной системе. Под платформой понимается среда исполнения и набор технологий, используемые в качестве основы для построения определенного круга приложений. Фактически, приложения базируются на нескольких платформах, образующих многослойный пирог. При этом важно, что платформа предоставляет разработчику определенную модель, как правило, изолирующую его от понятий и подробностей более низкоуровневых технологий и платформ. Такова и платформа "1С:Предприятие", позволяющая использовать самые разные технологии более низкого уровня, не меняя кода бизнес-приложений.

Например, она предлагает разработчику собственную модель работы с данными и изолирует его от особенностей конкретного хранилища данных, а это позволяет, не изменяя бизнес-приложение, использовать в нем различные хранилища. К примеру, в качестве БД при решении задач небольшого масштаба может применяться собственный файловый движок, а для работы в масштабе предприятия – MS SQL Server.

Ключевым качеством платформы "1С:Предприятие", пожалуй, является достаточность ее средств для решения задач, стоящих перед бизнес-приложениями. Это позволяет обеспечить очень хорошую согласованность всех технологий и инструментов, которыми пользуется разработчик. Ведь часто именно наличие "швов" между различными технологиями становится причиной самых серьезных проблем. Простейший пример – система типов. В платформе "1С:Предприятие" разработчик использует одну систему типов данных и для взаимодействия с БД, и для реализации бизнес-логики, и для построения интерфейсных решений:



Поэтому у него нет проблем, связанных с преобразованием типов при переходах между разными уровнями прикладной системы. Другим очень важным моментом является стандартизация. Наличие единой платформы для большого количества прикладных решений способствует формированию общего "культурного слоя", включающего и людей (программистов, аналитиков, пользователей) и методологию (типовые структуры данных, алгоритмы, пользовательские интерфейсы). Опираясь на этот "культурный слой", разработчик тратит минимум усилий на поиск необходимого решения, практически, в любой ситуации, начиная от включения в проект нового специалиста и кончая реализацией какой-либо подсистемы бизнес-приложения по типовой методологии.

Одним из существенных преимуществ четкого разграничения между платформой и бизнес-приложением является высокий уровень адаптируемости решений под требования клиента.

Следует заметить, что именно для экономических задач особо важна возможность эффективного изменения готового решения разработчиком, не участвовавшим в его создании. В индустрии разработки бизнес-приложений, в отличие от многих других областей, существенная часть разработчиков не создает программы "с чистого листа", а дорабатывает и развивает типовые решения.

Это обстоятельство определяет особые требования к обеспечению наглядности и простоте понимания разработчиком существующих решений и максимально учитывается во всех механизмах платформы.

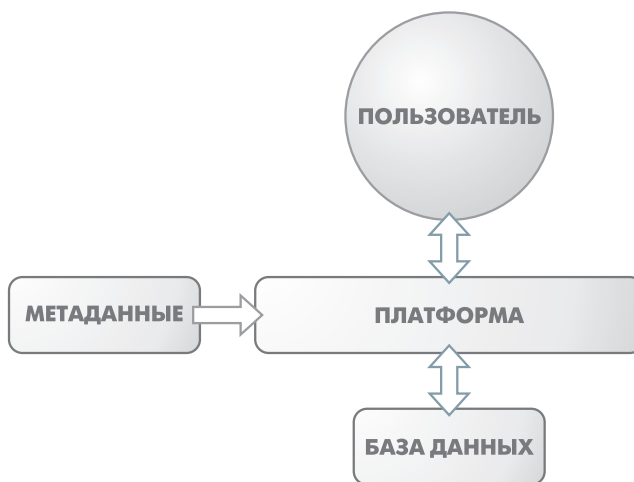
Инструментальные средства "1С:Предприятия" представляют собой не некий дополнительный "toolkit", а являются неотъемлемой составляющей платформы. Они ориентированы в равной степени, как на разработку решений, так и на их адаптацию при внедрении на конкретном предприятии. Эти средства поставляются с каждым комплектом 1С:Предприятия и применяются как для внесения небольших изменений, например, в макет печатной формы, так и для существенной доработки прикладного решения включая структуры данных и бизнес-логику. Возможности эффективного внесения изменений в приложение при его внедрении заложены в самих этих инструментах, а, кроме того, этому способствует и архитектура построения прикладного решения. В следующих разделах статьи мы постараемся проиллюстрировать данный тезис.

Выделение бизнес-приложения как самостоятельного элемента, фактически, позволило нам сформировать целую индустрию создания, распространения и поддержки разнообразных прикладных систем, концентрирующую свои усилия на специфике этого класса задач и не требующую глубокого понимания большей части технологических деталей и подробностей. Например, для фирмы, имеющей специалистов, опыт и репутацию в конкретной отрасли, стало вполне реальным в короткие сроки выйти на рынок бизнес-приложений с собственным продуктом, не тратя несколько лет на создание технологической базы.

Метаданные – способ описания бизнес–приложения

В "1С:Предприятии" прикладное решение не пишется в прямом смысле на языке программирования. Язык программирования, конечно, используется, но только там где это действительно необходимо.

В основе бизнес–приложения лежат метаданные. Они представляют собой структурированное декларативное его описание. Метаданные образуют иерархию объектов, из которых формируются все составные части прикладной системы и которые определяют все аспекты ее поведения. Фактически, при работе бизнес–приложения платформа "проигрывает" (интерпретирует) метаданные, обеспечивая всю необходимую функциональность.



Метаданными описываются структуры данных, состав типов, связи между объектами, особенности их поведения и визуального представления, система разграничения прав доступа, пользовательский интерфейс и т.д. В метаданных, фактически, сосредоточены сведения не только о том, "что хранить в базе данных", но и о том, "зачем" хранится та или иная информация, какова ее роль в системе, и как связаны между собой информационные массивы. Использование языка программирования ограничено в основном решением тех задач, которые действительно требуют алгоритмического описания, например, расчета налогов, проверки корректности введенных данных и т.д.

Что дает такой подход к построению бизнес–приложения? Во–первых, при описании метаданных широко используется визуальное редактирование. Это позволяет свести существенную часть разработки к визуальному проектированию, не требующему кропотливого написания кода. Однако у данного подхода есть и другие не менее важные преимущества. Описывая прикладное решение в терминах метаданных, разработчик "сообщает" платформе много очень полезной информации, которую та может эффективно использовать в самых различных целях. На основе метаданных система автоматически "выстраивает" большую часть механизмов и объектов, обеспечивающих функционирование прикладного решения. Например, описания метаданных платформе достаточно для того, чтобы автоматически сформировать пользовательский интерфейс системы, обеспечивающий ввод и редактирование взаимосвязанной информации. Другой пример – возможность построения даже конечным пользователем, не имеющим навыков программирования, достаточно сложных отчетов.

Идеология использования метаданных в самых общих словах сводится к простому тезису: "Давайте не будем программировать все функции разрабатываемого решения. Расскажем платформе о составе, структуре, особенностях и взаимосвязи различных его частей, и пусть остальное она сделает сама".

Эта идеология (Metadata Driven) сегодня находит все большее применение во многих перспективных разработках.

Построение приложения на основе модели

В "1С:Предприятии" изначально заложена строгая ориентация на построение прикладного решения на основе определенной модели.

Этот подход является весьма перспективным и по нашей оценке будет доминирующим в обозримом будущем в современных средствах разработки. Идеи построения бизнес-приложений на основе модели, например, нашли воплощение в архитектуре MDA (Model Driven Architecture) консорциума OMG.

Под моделью понимается вся идеология построения прикладного решения. Сюда относятся способы построения структур данных, типы связей между данными, принципы манипулирования данными, формы описания бизнес-логики, способы связи данных с интерфейсными объектами, разделение функциональности по уровням системы и многое другое.

Важно, что все бизнес-приложения неукоснительно следуют принятой модели и этим обеспечивается единообразие и предсказуемость их поведения. Фактически, разработчик, желающий отразить в прикладном решении специфику той или иной предметной области, имеет вполне определенный набор способов решения этой задачи средствами, заложенными в платформу. С одной стороны, такой подход ограничивает (вполне осмысленно) свободу разработчика, но с другой – защищает его от множества ошибок и позволяет в сжатые сроки получать работоспособное решение, которое сможет в дальнейшем развиваться и поддерживаться как им самим, так и, при необходимости, другим специалистом.

Очевидным следствием этого подхода является изоляция разработчика бизнес-приложения от деталей технологий хранения информации, организации трехуровневой архитектуры и т.д. Например, как уже отмечалось выше, все прикладные решения, базирующиеся на "1С:Предприятии", без каких-либо изменений работают как с собственным файловым движком БД, так и с сервером БД. При этом необходимые структуры данных создаются и изменяются системой автоматически на основе описания метаданных, и разработчику не приходится вникать в детали форматов хранения конкретных СУБД. Манипулирование данными в приложении также описывается в высокоуровневой модели и автоматически исполняется с учетом особенностей используемого хранилища.

По нашей оценке, предоставление разработчику определенной модели, абстрагированной от конкретных используемых средств, будет доминирующим в обозримом будущем в современных средствах разработки.

Наличие единой модели ключевым образом сказывается и на простоте освоения системы. Вся разработка ведется в рамках одной сквозной системы понятий и в едином пространстве типов данных. У разработчика не возникает необходимости осваивать несколько моделей представления и тратить усилия на реализацию переходов между ними на разных уровнях.

Например, описание в метаданных тех или иных объектов (сущностей) сразу определяет и соответствующие типы встроенного языка программирования и необходимые для их хранения структуры БД. Все последующие манипуляции этими объектами, как в памяти, так и в БД выполняются единообразно, не требуя преодоления "барьеров" между различными нотациями, принятыми при работе с СУБД и с универсальными языками программирования.

Управление данными

Одним из ключевых отличий бизнес-приложений от программ иного назначения является парадигма, используемая в них для манипулирования данными.

Очевидно, что для экономических задач работа с данными является основным содержанием с одной стороны и наиболее проблемным вопросом с другой. Это можно объяснить тем, что если во многих областях создания программных систем уже достигнут уровень, при котором программные продукты "близки к совершенству" (например, в текстовых процессах), то в области экономического софта до этого, мягко говоря, далеко.

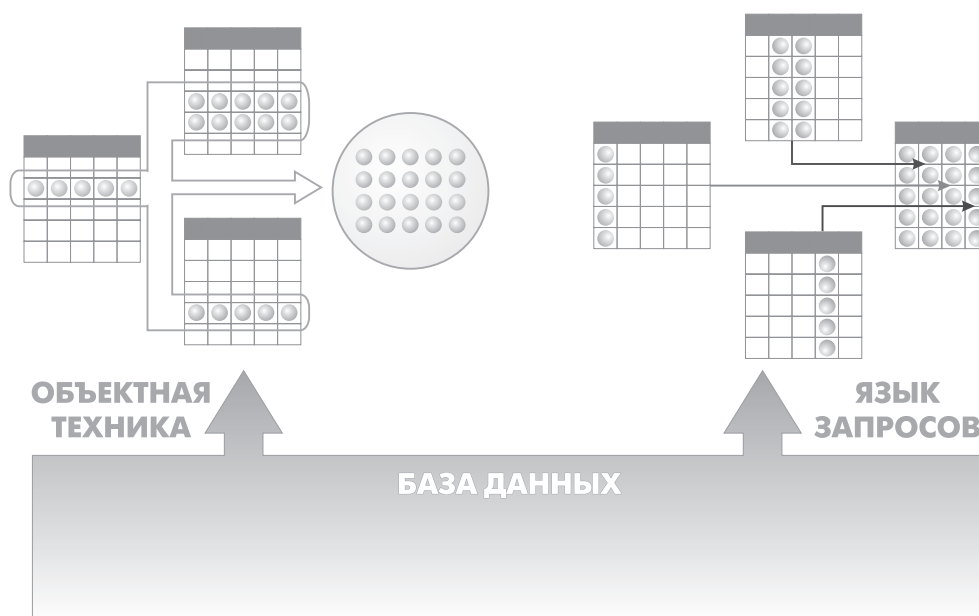
Экономические прикладные системы постоянно находятся "под гнетом" противоречивых требований. С одной стороны необходимо обрабатывать большие объемы данных, а с другой – обеспечивать широкую функциональность и высокую производительность. Объемы обрабатываемых данных растут, растут требования к разнообразию решаемых задач, и повышаются требования к масштабируемости. Не следует забывать и о повышении удобства разработки, эргономичности систем, возможностях их обновления и доработки и т.д. В современной мировой практике существует несколько парадигм манипулирования данными, которые используются в средствах разработки бизнес-приложений в различных комбинациях. При этом до идеала еще далеко – все имеющиеся и создаваемые технологии являются компромиссными решениями, нацеленными на улучшение показателей систем по нескольким указанным критериям.

Разумеется, эти противоречия касаются всех аспектов системы, но на способах манипулирования данными они проявляются наиболее ярко.

В современной мировой практике существует несколько парадигм, которые используются в средствах разработки бизнес-приложений в различных комбинациях.

В "1С:Предприятии" используется смешанный подход, который с одной стороны имеет много общего с подходами, заложенными в перспективных разработках других фирм, но с другой стороны обладает и существенными отличиями.

Для всех операций модификации данных (создания, изменения и удаления) в "1С:Предприятии" применяется исключительно объектная техника. Это означает, что разработчик взаимодействует с БД не на уровне записей, а с помощью объектов, соответствующих хранимым в БД сущностям. Для изменения хранимых данных, ему не нужно писать сложные запросы и преобразовывать результаты их обработки в объекты языка программирования. Достаточно получить объект из базы данных, изменить его свойства и снова сохранить. Разработчик имеет при этом возможность написать обработчики событий, связанных с изменением данных, выполняя с их помощью различные проверки и изменяя при необходимости другие данные. Система обеспечивает эффективную технологическую поддержку объектного подхода, осуществляя, например, кэширование объектов, контроль объектной и ссылочной целостности и т.д. Для чтения данных может использоваться как объектная техника, так и декларативный язык запросов, который основывается на классическом SQL, но имеет ряд существенных расширений. Расширения направлены с одной стороны на поддержку работы с объектами, хранящимися в базе данных, а с другой – на эффективное решение экономических задач. Ниже мы рассмотрим язык запросов более подробно:



Одним из ключевых преимуществ объектно-реляционной парадигмы управления данными является наглядность и простота разработки приложений. Объектная техника, используемая в основном для модификации данных, обеспечивает очень хорошую читаемость алгоритмов бизнес-логики, существенно уменьшает количество ошибок при разработке, а также обеспечивает высокий уровень целостности данных. Кроме того, важной особенностью объектной техники является то, что она, благодаря строгой гранулярности манипулирования данными, упрощает переход к распределенным и интегрированным системам.

Декларативное описание запросов, в свою очередь, позволяет получать сколь угодно сложные выборки данных и предоставляет мощные возможности агрегирования, необходимые для решения аналитических задач.

По нашему мнению такая смешанная техника в том или ином виде начнет в обозримом будущем достаточно широко использоваться в перспективных разработках экономических систем.

Здесь, немного забежав вперед, следует сказать, что в модели "1С:Предприятия" реализована наиболее современная концепция работы с информацией, сочетающая три способа представления данных – хранение сущностей в базе данных, их представление в языке программирования в виде объектов и отображение в формате XML. Фактически любая информация может в зависимости от текущего режима работы представляться одним из этих трех способов.

Долговременное хранение сущностей (persistence) осуществляется в БД, что обеспечивает надежность и эффективную обработку больших объемов информации. Для внесения изменений, данные предварительно преобразуются в объекты встроенного языка. При внутреннем обмене в распределенной БД или взаимодействии с другими информационными системами данные переносятся в формате XML:



Заметим, что все три способа представления опираются на единую систему понятий и от разработчика не требуется усилий для трансформации данных из одного способа представления к другому.

Такой подход позволяет органично вписывать "1С:Предприятие" в будущие гетерогенные решения, которые по нашей оценке (и она совпадает с оценкой многих экспертов) получат со временем самое широкое распространение.

Одной из отличительных особенностей модели "1С:Предприятия", не имеющей, как нам кажется, прямых аналогов в других подобных системах является деление всех прикладных данных на те, что имеют объектную природу и не имеющие таковой. Заметим, что для манипулирования теми и другими используется объектная техника.

Такое деление, соответствует реальной природе данных. В предметной области всегда есть сущности, имеющие объектную природу, например, "клиенты", "физические лица", "товары". Здесь объект имеет определенную "самодостаточность" не зависящую от данных, которые его описывают. Например, у человека может поменяться фамилия, имя и номер паспорта, но нам важно знать, что это именно то же самое физическое лицо (уникальный объект). С другой стороны есть сущности, не имеющие объектной природы. Например, запись о приходе некоторого товара на некоторый склад является лишь информацией о движении товара и не имеет никакого другого содержания, кроме того, что зафиксировано в записи. Если заменить в такой

записи один товар на другой, смысл записи о товародвижении полностью изменится. Иными словами, для такой сущности запись без указания конкретных значений полей не имеет никакого смысла.

В результате такого смыслового разделения сущностей платформа "1С:Предприятие" фактически предлагает разработчику готовую методологию для наиболее естественного решения различных задач манипулирования данными в соответствии с природой этих сущностей.

Работа с объектными сущностями поддерживается представлением сущностей базы данных в виде объектов встроенного языка программирования, а также специальными типами данных, служащими для представления объектных ссылок (ссылок на объекты базы данных). При этом, зная объект, легко получить его ссылку, а, зная ссылку, извлечь из БД объект. Такая техника обеспечивает наглядный и естественный способ описания в исходном коде алгоритмов бизнес-логики, манипулирующих объектами, а, кроме того, гарантирует логическую целостность данных при любых операциях. Она напоминает написание приложений объектных БД, с той лишь разницей, что сохранение данных происходит в таблицах реляционной СУБД. При этом в модулях, написанных на встроенном языке, может одновременно присутствовать несколько объектов, представляющих одну сущность БД.

Механизмы платформы обеспечивают поддержку уникальных объектных идентификаторов (ссылок), контроль версий объектов, пессимистическую и оптимистическую их блокировку. Оптимистическая блокировка гарантирует логическую целостность изменения объектов, а пессимистическая позволяет организовывать одновременное редактирование пользователями одних и тех же объектов в интерфейсе "1С:Предприятия". Платформа оптимизирует операции считывания объектов за счет использования механизма их кэширования как внутри транзакций, так и вне их. При модификации объектов реализована технология "умной записи": система следит за их изменениями и реально записывает на диск только модифицированные данные, обеспечивая, тем не менее, целостность данной операции.

Работа с неobjектными сущностями осуществляется с помощью наборов записей, представляющих собой по сути коллекции записей. Такие наборы поддерживают чтение и модификацию данных с требуемой для прикладной логики гранулярностью.

Важно, что оба указанных подхода позволяют придерживаться простого и естественно-го стиля написания алгоритмов бизнес-логики. Разработчик, которому необходимо модифицировать имеющуюся в приложении бизнес-логику, читая такой алгоритм, быстро понимает суть дела и может легко производить отладку и корректировку исходного кода. Теперь для взаимодействия с базой данных разработчику не приходится самому программировать сложные преобразования информации из объектных структур в реляционные, что заметно повышает эффективность создания и отладки ПО.

Еще одной важной особенностью объектной техники, принятой в платформе "1С:Предприятие", является то, что те же объекты, которые присутствуют в модулях на встроенном языке (как для объектных, так и для не объектных сущностей) используются и для отображения данных в интерфейсе. Элементы управления форм непосредственно связываются с нужными объектами, и обеспечивают их отображение и редактирование пользователем без какой-либо помощи со стороны разработчика.

Для объектных сущностей платформа "1С:Предприятия" поддерживает механизм представлений. Он отвечает за отображение в интерфейсе значений, заданных ссылками на сущности базы данных. При необходимости отобразить ссылочное значение система автоматически формирует представление на основании свойств метаданных, оптимизируя, по возможности, получение информации из БД с помощью кэширования и других механизмов. В процессе обработки запросов и построения отчетов также широко используются представления. Это позволяет универсальным образом получать представления ссылочных полей, если запрос формируется для отображения данных в пользовательском интерфейсе, и автоматически включать отображения представлений в отчеты для полей, содержащих ссылочные значения. Важно, что механизм представлений дает возможность разработчику просто и естественно манипулировать объектными ссылками, минимизируя в то же время число обращений к БД.

Наряду с описанными выше методами манипулирования данными и формирования запросов, система предлагает еще один способ доступа к данным – динамические выборки. Этот механизм позволяет обращаться к очень большим объемам данных, обеспечивая считывание информации порциями. При этом разработчик, только указывает, какие данные необходимо получить, а система автоматически выполняет обращения к БД с необходимой гранулярностью. Важно, что для решения этой задачи не используются какие-либо специфические средства динамического считывания конкретной СУБД, требующие удержания в памяти открытой выборки, а осуществляется автоматическое формирование запросов, последовательно выбирающих блоки записей.

Еще одним важным решением в части работы с данными в "1С:Предприятии" является поддержка в полях таблиц составных типов данных. Эта возможность, насколько нам известно, не имеет близких аналогов в других системах. При описании типа поля какого-либо объекта можно выбрать не только один из доступных типов, но и практически любую (с некоторыми ограничениями) их комбинацию. Например, в поле "Плательщик" в документе, отражающем операцию с банком, допускается хранение ссылки на юридическое или физическое лицо в зависимости от конкретной операции. Хотя приведенный пример является достаточно простым, возможность работы с составными типами позволяет решать такие задачи, как хранение произвольных характеристик товаров, ведение аналитического учета на бухгалтерских счетах по любому составу аналитических разрезов, настраиваемых пользователем и т.д. Важно, что система не просто предоставляет возможность хранения в одном поле разнородных значений, а делает это прозрачным для разработчика способом. Прежде всего, необходимо отметить полную поддержку работы с полями составных типов "движка" базы данных и языка запросов. Для этих полей поддерживается весь набор стандартных операций (сравнение, агрегирование и т.д.). Другим важным моментом является поддержка составных типов в интерфейсных механизмах системы. Например, поле ввода, связанное с данными такого составного типа, предоставляет весь набор возможностей редактирования (выбор типа; редактирование значений всех типов, входящих в указанное поле; ограничение выбираемых типов).

Отдельного внимания заслуживают также средства формирования запросов "1С:Предприятия". Мы уже упомянули, что они основаны на конструкциях стандартного языка SQL, но имеют ряд существенных расширений.

Прежде всего, следует отметить поддержку в языке запросов объектов, хранящихся в БД. Все операторы языка запросов обеспечивают работу со ссылочными типами (полями, хранящими ссылки на объекты БД). Например, поддерживается обращение к полям в нотации "через точку" без ограничения количества уровней. Можно указать в запросе выборку такого поля, как "Товар.Производитель.Страна.Наименование". Для объектов базы данных допускается обращение к вложенным таблицам и как к отдельным таблицам, и как к обычным полям объекта, содержащим коллекции записей.

Другим важным качеством является штатная реализация функции формирования многомерных итогов с произвольным порядком обхода измерений. При этом доступны такие возможности, как сочетание многомерного обхода измерений и многоуровневого обхода иерархии значений каждого измерения (например, многоуровневой структуры подразделений или многоуровневой группировки товаров). Поддерживается также ряд специальных режимов обработки итогов.

Механизм запросов способен на основании свойств прикладных объектов, определенных в метаданных, упорядочивать выборки автоматически. Это позволяет разработчику при создании запроса, предназначенного для получения отчета или визуализации данных, не указывать, по каким полям производить упорядочивание, а, включив автоматический режим, получить стандартную для выбираемых данных сортировку.

Еще одним мощным средством механизма запросов являются виртуальные таблицы. Они обеспечивают доступ к производным данным, предоставляемым различными прикладными подсистемами, не требуя для этого написания сложных запросов. Например, можно обратиться к виртуальной таблице для получения распределения остатков и оборотов товаров по складам и номенклатуре, а также по календарным периодам. Следует отметить, что работа с

виртуальными таблицами не является аналогом простого хранения типовых запросов (view). При использовании виртуальных таблиц разработчик указывает набор параметров, описывающих необходимую выборку, беря в их качестве не только конкретные значения, но и, например, сложные условия. Разработчик прикладного решения работает с виртуальной таблицей, практически, так же как и с обычной, но система формирует запрос к БД таким образом, чтобы обеспечить максимальную эффективность. В частности, при обращении к данным, обрабатываемым учетными механизмами, могут использоваться хранимые ими промежуточные итоги.

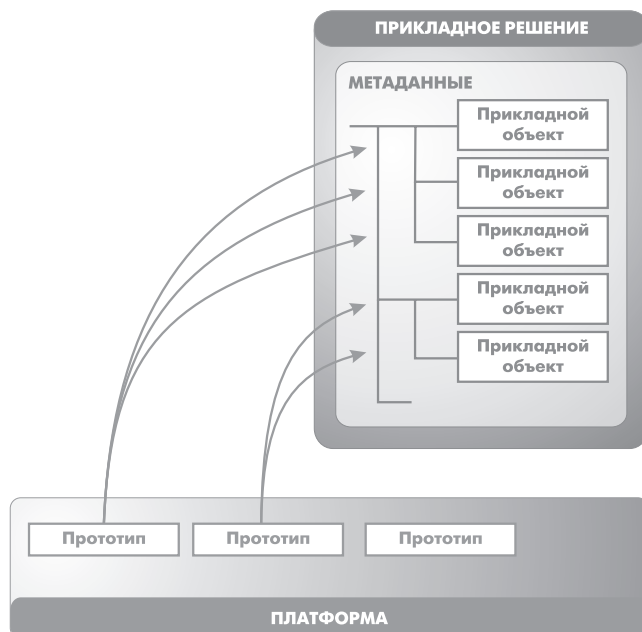
Стандартные прототипы прикладных объектов

Если говорить о различиях моделей бизнес-приложений и средств их разработки то, пожалуй, наиболее существенным окажется то, в каких понятиях (можно даже сказать "в какой парадигме") описывается приложение. Разумеется, в каждом средстве разработки может использоваться несколько способов описания, но какой-то один набор понятий всегда является основополагающим.

В качестве примеров существующих подходов можно привести описания в терминах реляционных таблиц, классов объектного языка программирования, сохраняемых сущностей (Entity Persistent) и т.д.

В модели разработки "1С:Предприятия" используется подход, которому мы не нашли явного аналога в других системах. Здесь все прикладное решение описывается метаданными в виде совокупности прикладных объектов, выбираемых из жестко определенного набора прототипов (классов). Можно было бы назвать создаваемые объекты бизнес-компонентами, а их прототипы шаблонами (patterns). Каждый такой прототип отвечает за отражение в прикладном решении определенной совокупности объектов или процессов предметной области, имеющих схожие поведенческие характеристики и сходную роль в общей картине решения.

Примерами таких прототипов в "1С:Предприятии" являются "Справочники", "Документы", "Регистры накопления". Каждый прототип имеет некоторую базовую реализацию, которая определяет особенности функционирования, создаваемых на основе данного прототипа объектов: структуру хранимых сущностей вместе с некоторыми predetermined полями, набор типов языка программирования, методы, свойства и события, а также типовые для решаемой задачи операции, способы отображения и редактирования, методы регулирования прав доступа и т.д.



Если в ходе разработки необходимо создать объект, отражающий особенности предметной области, программист выбирает подходящий прототип и создает на его основе объект метаданных. Далее он может задать различные свойства, определяющие особенности поведения объекта, основанного на этом прототипе, дополнить при необходимости структуру данных объекта, реализовать необходимый набор методов, определить обработчики событий, уточнить способы отображения объекта в интерфейсе и т.д.

Для простых объектов такое доопределение может не потребоваться совсем, и вся процедура создания объекта будет сведена к указанию его имени в свойствах метаданных. В более сложных случаях разработчик имеет возможность достаточно подробно определять и структуры данных, и алгоритмы бизнес-логики, но при этом стандартная функциональность реализуется автоматически без каких-либо усилий с его стороны.

Важно, что количество таких прототипов в платформе не велико, около двух десятков: их легко изучить, с тем, чтобы впоследствии эффективно применять для решения любых задач предметной области.

Таким образом, структура метаданных "1С:Предприятия" является не просто набором описаний объектов в определенных унифицированных терминах. Все прикладное решение, фактически, состоит из объектов, четко разделенных по тем ролям, которые они играют в бизнес-приложении. Такой подход существенно усиливает эффект и от описания системы в терминах метаданных, и от построения приложения на основе модели.

Наличие у объектов, описанных метаданными, стандартной функциональности и predetermined роли позволяет системе автоматически решать гораздо более широкий круг задач, возлагаемых как непосредственно на эти объекты, так и на общие механизмы, работающие с ними. Фактически, зная о назначении (роли в прикладном решении) того или иного объекта, платформа сама, без какого-либо участия программиста, обеспечивает для него в любой ситуации соответствующий "индивидуальный подход". Например, механизм форм "знает", как лучше редактировать те или иные данные, а стандартный механизм отчетов может автоматически "подстроиться" с тем, чтобы проводить анализ информации наиболее эффективно.

Использование predetermined прототипов обеспечивает стандартизацию модели построения прикладных решений. Такая стандартизация, возможно, не очень существенна для одного отдельно взятого бизнес-приложения, но она имеет огромное значение при создании индустрии разработки и поддержки множества прикладных решений. Стандартизация позволяет существенно упростить задачи проектирования и в несколько раз снижает сложность и трудоемкость сопровождения. Например, если необходимо подключить к проекту еще одного специалиста или передать другому сотруднику поддержку и развитие решения, новый разработчик может после беглого просмотра структуры метаданных буквально за несколько минут составить общее представление о строении и основной функциональности прикладного решения, так как в структуре метаданных все прикладные объекты сгруппированы по принятым в модели "1С:Предприятия" функциональным ролям. В дальнейшем он сможет также быстро разбираться в устройстве и бизнес-логике отдельных подсистем. По опыту эти показатели существенно отличаются от тех, которые достигаются при использовании других подходов к описанию приложения.



Прикладные объекты и механизмы

Выше мы описали только принцип построения прикладного решения, основанный на использовании стандартных прототипов прикладных объектов.

Разумеется, здесь не место для подробного описания всех имеющихся прототипов. Но ниже, чтобы продемонстрировать те идеи, которые заложены в имеющуюся модель построения бизнес-приложения, мы дадим краткий обзор прототипов и предоставляемых ими возможностей.

Сначала мы приведем примеры достаточно простых прототипов – справочников и документов. Справочники описывают каталоги, содержимое которых более или менее постоянно. Это может быть перечень выпускаемой продукции, список клиентов компании, перечень валют и т.д. Справочники обеспечивают поддержку иерархических структур, позволяют относить данные к отдельным объектам и их группам, предоставляют ряд других сервисных возможностей.

Документы отражают в системе события, происходящие в жизни предприятия: поступление материалов, перечисление денег через банк, прием сотрудника на работу и т.д. Этот прототип обеспечивает их отражение в различных учетных механизмах, поддерживает контроль последовательности обработки событий, реализует сквозную нумерацию объектов разного типа и т.д.

Остановимся немного подробнее на некоторых возможностях этих двух прототипов.

Известно, что классическая реляционная модель не имеет готовых средств для поддержки иерархии данных и ее реализация всегда требует кропотливой работы. Справочники (а также некоторые другие прототипы прикладных объектов "1С:Предприятия"), поддерживают многоуровневую иерархию изначально. Включается этот механизм простой активизацией соответствующего свойства в метаданных. При этом поддержка иерархии распространяется сразу на все аспекты использования прикладного объекта. Например, прототип обеспечивает поддержку необходимых свойств и методов в объектной модели манипулирования данными (определение уровня объекта, контроль за цикливанием иерархии и т.д.). В интерфейсных механизмах реализуется представление данных в виде иерархического списка или дерева с навигацией по уровням и интерактивным изменением иерархии. В механизмах отчетов обеспечивается формирование иерархических документов такого рода и получение многоуровневой иерархии итогов в любых отчетах, в которых объекты этого типа, выступают, в качестве параметров для группировки.

Другим примером штатных возможностей прототипов является поддержка механизма проведения документов. Этот механизм предлагает разработчику стандартную модель организации связи между информацией о событиях, происходящих на предприятии, и различными учетными механизмами. Любая вводимая пользователем в виде документов информация может отражаться в любых учетных механизмах (планировании, управленческом учете, бухгалтерском учете и т.д.). Разработчик должен только указать в свойствах метаданных связь между документами и учетными механизмами, а также описать алгоритм проведения документа. Все необходимые действия по проведению и отмене проведения система будет выполнять автоматически. При этом системой предоставляются дополнительные возможности, такие как поддержка отражения событий в реальном времени, поддержка восстановления последовательности отражения событий, происходящих на предприятии, при изменении их задним числом и т.д. В результате предоставляется единая модель связи исходных данных и учетных механизмов, которая не просто облегчает разработку, но и обеспечивает единообразное предсказуемое поведение всех прикладных решений, что существенно облегчает их освоение и поддержку.

Весьма интересными являются прототипы объектов "1С:Предприятия", применяемые в механизме сложных периодических расчетов. Этот механизм можно рассматривать как универсальный инструмент для решения задач расчета зарплаты любой сложности. На самом деле, расчет зарплаты является наиболее типичным применением данного механизма, но сам механизм не имеет ориентации именно на эту задачу и успешно используется для решения

других задач, требующих описания периодических расчетов со сложными взаимосвязями, например, расчета дивидендов, стоимости коммунальных услуг и т.д. Он содержит набор типовых стратегий, примером которых может служить вытеснение одних видов расчетов другими при их пересечении на шкале времени. Применительно к задаче расчета зарплаты эта возможность решает весь комплекс проблем, связанных с пересекающимися по времени начислениями и удержаниями, для которых определены сложные правила взаимного исключения. Другим примером является возможность установки взаимосвязи между различными видами начислений и удержаний по периоду действия. Таким образом, механизм представляет определенную математическую модель, устанавливающую связи между различными видами расчетов и предоставляет разработчику действующие на основе этой модели механизмы обработки информации.

Не углубляясь далее в подробности этого механизма, стоит отметить, что он позволяет полностью описать схему расчета зарплаты на основании предоставляемых прототипов прикладных объектов и заложенных в них возможностей. На примере данного механизма можно достаточно наглядно продемонстрировать эффект от использования предлагаемого подхода. Мы не имеем на данный момент информации о наличии в каком-либо средстве разработки экономического софта подобного механизма. Во всех известных нам случаях модуль расчета зарплаты (Payroll) реализуется ценой достаточно больших затрат и на проектирование и на реализацию. Причем этот модуль приходится существенно или полностью переписывать для каждой страны, так как он сильно привязан к местному законодательству. Применение механизма сложных периодических расчетов в "1С:Предприятии" позволяет перевести разработку такого модуля от проектирования "с чистого листа" к реализации по стандартной схеме. По нашей оценке это сокращает затраты на порядок, но самое главное, существенно упрощается внесение изменений в процессе поддержки системы.

Еще одним характерным примером служит механизм характеристик, предназначенный для решения задач, которые плохо согласуются с классической реляционной моделью базы данных. Одной из таких задач, является построение структуры для отражения разнообразных свойств (характеристик) номенклатуры товаров или изделий. Если создается решение для одной организации с четкой ориентацией на определенную группу товаров, то все необходимые реквизиты можно заложить непосредственно в структуру каталога товаров. Например, для дилерской фирмы по продаже автомобилей ими будут цвет, объем двигателя и т.д. Однако если создается тиражное бизнес-приложение, или решение для организации, занимающейся производством или продажей разнообразной номенклатуры, этот вариант не подходит, так как очевидно, что состав характеристик должен зависеть от конкретной группы номенклатуры и добавление каждой новой группы не должно сопровождаться доработкой прикладного решения. Очевидно, что реализация такого решения плохо ложится в реляционную структуру и требует особого подхода. Реализованный в "1С:Предприятии" механизм характеристик предоставляет разработчику типовое средство решения подобных задач. Специализированный прототип предназначен для организации хранения списка видов характеристик. При этом каждому такому виду присписывается свой тип данных. Например, объем двигателя будет задаваться числом, а цвет выбираться из справочника. Важно, что конкретные виды характеристик могут не только определяться в самом приложении, но и добавляться пользователем в процессе эксплуатации системы без изменения прикладного решения. При этом разработчик может задействовать данный механизм в самых различных ситуациях, например, для хранения логически связанных комбинаций характеристик номенклатуры и расчета себестоимости в разрезе имеющихся комбинаций.

Еще одним интересным решением является механизм бизнес-процессов (work-flow), позволяющий разработчику организовать совместную работу пользователей при выполнении типовых последовательностей деловых операций. Во многих существующих информационных системах для решения задач work-flow используются специализированные продукты, которые приходится интегрировать с приложениями, решающими экономические задачи. В платформе "1С:Предприятие" механизм бизнес-процессов полностью интегрирован в систему таким образом, что ни разработчик ни пользователь не видят "швов" разделяющих этот механизм и другую функциональность. Этот механизм включает средства для описания в прикладном решении схем бизнес-процессов, и их ролевой маршрутизации, для формирования заданий ,

выполняющихся в каждой точке маршрута, для управления бизнес-процессом и организации его связи с другими функциями прикладного решения. Данный механизм предоставляет разработчику гибкие возможности управления ветвлением процесса и формирования задач. Например, кроме обычного условного ветвления разработчик может визуальным образом описать параллельное прохождение нескольких веток маршрута, и указать точку их слияния. Допускается направление одного задания группе потенциальных исполнителей, например, если выписать счет должен один из менеджеров отдела. И наоборот, в некоторой точке маршрута можно инициировать несколько заданий, например, если финансовые отчеты должны представить все руководители отделов. Ролевая маршрутизация позволяет формировать задачу не только непосредственно конкретному сотруднику, но и распределять задачи по ролям, подразделениям и другим критериям, которые может описать разработчик прикладного решения. При осуществлении ролевой маршрутизации разрешается указывать текущее распределение обязанностей сотрудников с учетом временных замещений, совмещений нескольких должностей и т.д. Важно отметить, что данный механизм предлагает готовую стратегию автоматизации совместной деятельности работников предприятия. Для описания простейших бизнес-процессов достаточно визуального задания схемы маршрута и указания условий ветвления в их узловых точках. Все остальные действия выполняются системой автоматически. При реализации сложных бизнес-процессов усилия разработчика требуются в основном для тесной их увязки с функциями прикладного решения.

На приведенных примерах мы продемонстрировали, что каждый из прототипов прикладных объектов и объединяющих их механизмов закрывает определенную область задач предметной области существенно снижая затраты на разработку и, что очень важно, обеспечивая стандартизацию прикладных решений.

Чтобы дать представление о спектре инструментов предоставляемых разработчику, ниже мы кратко опишем еще несколько механизмов.

Так, для решения задач, связанных с учетом наличия и движения средств (материальных и денежных), предлагается механизм регистров накопления.

Он поддерживает многомерную систему учета с произвольным составом измерений и ресурсов и обеспечивает оптимизацию получения итогов на различные моменты времени, тем самым позволяя эффективно решать задачи материального учета, производственного и финансового планирования, расчета с контрагентами и т.д.

Механизм бухгалтерского учета представляет собой универсальный "движок" для решения задач автоматизации бухгалтерского учета в различных моделях счетоводства.

Он может использоваться в различных странах и имеет большой запас гибкости для дополнительной настройки. В его основе лежит принцип двойной записи, причем допускается работа с финансовыми транзакциями (операциями), включающими как простые корреспонденции (один дебет и один кредит), принятые в российском учете и в моделях учета некоторых других стран, так и сложные, применяемые в большинстве западных стран (несколько дебетов и несколько кредитов).

Механизм поддерживает многомерную систему учета с произвольным составом измерений и ресурсов. Уникальным его свойством является возможность настройки состава дополнительных измерений независимо для каждого счета. При этом допускается формирование состава измерений как разработчиком при создании прикладного решения, так и пользователем в процессе работы с системой. Многоуровневые и многомерные итоги в разрезе периодов, перекрестных корреспонденций и т.д. генерируются автоматически.

Механизм сведений предназначен для решения задач, связанных с хранением разнообразной информации об объектах в различных разрезах.

Например, данные о ценах на товары могут храниться в разрезе их номенклатуры, категорий клиентов и т.д. При необходимости может поддерживаться хранение истории внесенных изменений, что позволяет получать наряду с актуальными данными, еще и их значения на любой прошедший момент времени. Механизм обеспечивает всю логику работы с хранимыми сведениями, начиная от манипулирования данными до их автоматического контекстного отображения в пользовательском интерфейсе.

Еще раз подчеркнем, что в модели "1С:Предприятия" прикладные механизмы и лежащие в их основе прототипы – это не просто набор шаблонов, которые разработчик может использовать "по мере надобности". Всё прикладное решение с необходимостью основывается на использовании предлагаемого набора прототипов.

Этого, достаточно компактного, набора вполне хватает, чтобы закрыть все потребности предметной области. Имеющимся набором прототипов система фактически навязывает разработчику стандартную модель проектирования, и это позволяет, по нашей оценке, существенно снизить затраты на построение и поддержку прикладных решений .

Высокоуровневая модель интерфейса

Очевидно, что для бизнес-приложения вопрос организации пользовательского интерфейса имеет особое значение. Во-первых, потому, что для существенной части пользователей эти приложения являются основным инструментом, и такие пользователи работают с ним большую часть дня, не являясь зачастую профессиональными пользователями компьютера. Во-вторых, интерфейс бизнес-приложений обычно очень "объемный". В отличие от многих других классов систем бизнес-приложения, как правило, содержат от нескольких сотен, до нескольких тысяч форм. Кроме того, эти формы могут периодически меняться, например, в ходе развития системы или при необходимости изменения бизнес-логики. Соответственно, затраты на разработку интерфейса прикладного решения весьма высоки. Именно поэтому в платформу "1С:Предприятие" включен целый ряд механизмов, ориентированных на быструю разработку эргономичного пользовательского интерфейса. В них реализована собственная оконная модель, система форм, набор элементов управления и т.д.

Основной идеей построения интерфейса, реализованной в "1С:Предприятии", является максимальное использование информации из метаданных, а также объектов манипулирования данными с тем, чтобы вся конструкция не требовала детальной настройки со стороны разработчика и функционировала по большей части автоматически . Мы уже говорили выше, что объекты встроенного языка, используемые для манипулирования данными , применяются и для их отображения в интерфейсе. Разработчику достаточно связать такой объект с элементом управления и механизм интерфейса полностью возьмет на себя организацию просмотра и модификации данных. Однако, наибольший эффект достигается за счет установления связи между объектом манипулирования данными и самой формой, а не с отдельными элементами управления. В таком случае вся функциональность формы, относящаяся к просмотру и редактированию данных, а также ее связь с другими формами, обеспечивается системой автоматически . Для этого в "1С:Предприятии" реализован механизм расширений форм и расширений элементов управления. Расширения автоматически "включаются в работу" с учетом типа данных, с которыми связан элемент управления или форма.

Например, поле ввода обеспечивает все необходимые действия по редактированию ссылочных значений (выбор из специальных форм, множественный подбор, очистку, подсветку и автоматический выбор незаполненных значений, переход к формам редактирования выбранных объектов и т.д.). Очень эффективен так называемый ввод по строке, который позволяет в несколько раз ускорить массовый ввод информации, исключая необходимость выбора значений из форм списков . Для его реализации в свойствах метаданных задается перечень полей, по которым пользователи могут быстро находить нужные объекты, например, код, артикул, наименование. При наборе в поле ввода нескольких символов система автоматически подставляет то значение, одно из указанных в метаданных полей которого начинается с тех же символов. Если таких значений найдено несколько, пользователю предлагается осуществить выбор из списка. От разработчика при этом не требуется никаких усилий, так как интерфейсный механизм опирается на свойства метаданных.

Другим интересным примером является табличное поле. Этот элемент управления представляет собой мощный механизм визуализации и редактирования данных, предоставляющий пользователю возможность просмотра и редактирования больших списков, за счет динамического считывания. Мы уже говорили выше о механизме динамического считывания, как одном из средств манипулирования данными. В механизме интерфейса "1С:Предприятия"

он позволяет просматривать массивы данных практически неограниченного размера без считывания их в память целиком. При этом с точки зрения пользователя навигация по списку практически не отличается от просмотра данных, находящихся в памяти компьютера. Данный механизм поддерживает гибкие возможности поиска и фильтрации, просмотра иерархических структур, редактирование данных в списке и в отдельных формах, настройку состава колонок и их сортировки, печать списков и их экспорт в различные форматы и т.д.

Для форм, списков и других элементов система автоматически формирует командный интерфейс (кнопки, меню и целые панели управления), обеспечивающий все действия по просмотру и редактированию, а также связи с другими формами и сервисные функции.

Описав объект метаданных в прикладном решении, разработчик может вообще не строить для него форму, все необходимые формы будут создаваться автоматически в процессе работы системы. Если же он создает форму, чтобы задать ей специфические свойства, то конструктор генерирует стандартную форму, которая не содержит изначально никакого кода на встроеном языке., так как расширения элементов управления и формы обеспечивают всю необходимую функциональность. Разумеется, разработчик может и без конструктора вставить в любую форму элементы управления, связать их с данными и получить полностью работающую конструкцию, не написав ни строки кода. Дописывать что-то ему придется лишь тогда, когда он захочет переопределить или расширить стандартную функциональность.

Важно что, интерфейс "1С:Предприятия" обеспечивает работу не просто отдельных форм и элементов управления. Штатная механика управления формами прикладных объектов обеспечивает различные варианты связей между формами используемые при выборе значений, множественном подборе, детализации информации и т.д. Фактически, платформа "1С:Предприятия" предлагает разработчику готовую стратегию организации всего интерфейса бизнес-приложения имеющую способы реализации практически всех необходимых сценариев работы пользователей.

За счет использования хранящихся в метаданных знаний о связях между различными объектами, поддерживается автоматическая генерация команд перехода между логически связанными формами. Так пользователь, работая с некоторым документом, может без каких-либо усилий со стороны программиста получить доступ к записям, отражающим этот документ в различных учетных механизмах. Кроме связей определяемых в метаданных бизнес-логикой прикладного решения, разработчик может определить (в виде критериев отбора) и дополнительные связи для перехода между объектами, которые так же автоматически будут представляться в командном интерфейсе форм.

Другим стандартным интерфейсным решением является так называемый механизм "ввода на основании". Он позволяет при вводе данных по одному объекту, заполнять автоматически целый ряд полей, извлекая их значения из данных других объектов. Например, выписав счет, пользователь может нажатием одной кнопки сформировать на основании этого счета накладную: для этого достаточно установить в метаданных связь между упомянутыми объектами и описать правила заполнения полей.

Разработчик прикладного решения предоставляется стандартное средство реализации "сменного" командного интерфейса. В прикладных решениях, имеющих большой объем функциональности, это дает пользователю возможность переключаться на другой режим работы с полной или частичной сменой командного интерфейса, не выходя из системы.

Механизм форм обеспечивает стратегию смысловой идентификации форм, позволяющую не открывать повторно уже открытые формы, а активизировать открытые ранее формы при обращении пользователя к той или иной информации.

Таким образом, с помощью расширений форм и других механизмов автоматически поддерживается целостная стратегия пользовательского интерфейса, включающая различные варианты взаимодействия между формами, автоматическое формирование командного интерфейса для навигации по различным режимам и функциям системы, а так же всевозможные сервисные режимы, обеспечивающие комфортную работу пользователя.

Мы уже отмечали, что в интерфейсе "1С:Предприятия" реализовано много специальных элементов, не использующих стандартные средства операционной системы. Это сделано для

того, чтобы можно было строить интерфейс, максимально отвечающий специфике деловых приложений.

В частности, в "1С:Предприятии" реализована собственная модель оконной системы, ориентированная на работу с большим количеством разнородной информации. Например, максимизация окна управляется пользователем для каждой формы отдельно, а не как единый режим для всего приложения (как это принято в стандартных MDI приложениях). Кроме традиционных видов окон, в ней используются прикрепленные и прячущиеся окна. В интерфейсе "1С:Предприятия" из модального окна разрешается вызов немодальных окон, благодаря чему, фактически, образуется несколько слоев многооконного интерфейса. Разработчик может открывать модально любые формы, не изменяя логики их работы.

Поддерживается автоматический режим корректировки размеров элементов управления при изменении размера формы и при перемещении разделителей внутри нее. Таким образом, пользователь может установить желаемые размеры для каждой формы и отдельных ее частей, а формы автоматически используют все предоставленное пространство для эффективного отображения информации. Важно, что этот механизм не требует какой-либо настройки со стороны разработчика, но допускает, разумеется, и более тонкий тюнинг.

Формы и элементы управления имеют дизайн, ориентированный с одной стороны на отображение максимального количества информации, а с другой стороны на снижение утомляемости при длительной работе пользователя. Для этого используется "плоский" дизайн интерфейса приближенный к Web-дизайну. Для управления дизайном интерфейса предусмотрен механизм стилей, позволяющий централизованно изменять общий вид прикладного решения. Практически каждый элемент управления "оснащен" функциональностью, ориентированной на специфику экономических задач. Кроме того, предусмотрен ряд элементов управления непосредственно нацеленных на аналитические задачи, например, это диаграмма, диаграмма Ганта и т.д.

В интерфейсных механизмах "1С:Предприятия" имеется еще много интересных элементов, выше мы перечислили только некоторые решения. Важно, что весь набор интерфейсных механизмов предоставлен разработчику в достаточно высокоуровневых понятиях. Как уже говорилось, важным качеством концепции разработки в "1С:Предприятии" является изолирование прикладного решения от различного рода технических деталей. Например, разработчик не имеет доступа к управлению движениями мыши или детальной отрисовке элементов управления. Даже достаточно сложные механизмы предоставляются разработчику в таком виде, чтобы с одной стороны они могли бы функционировать вообще без какой-либо настройки, а с другой стороны – обеспечивали бы возможность настройки в простых высокоуровневых категориях, не требующих специальных знаний. Благодаря этому в прикладных решениях поддерживается современный полнофункциональный интерфейс, и в то же время, только незначительная доля исходного кода относится к его визуальной части а большая часть реализует собственно бизнес-логику интерфейса прикладного решения.

Следует упомянуть и о Web-расширении – механизме, позволяющем реализовывать Web-интерфейс к прикладным решениям "1С:Предприятия". Мы не будем рассказывать о деталях технологического устройстве этого механизма. Наиболее интересным в его реализации, на наш взгляд является то, что он также как и основной (rich) интерфейс максимально использует для автоматического формирования Web-форм информацию из метаданных, а также знания о назначении и устройстве прототипов прикладных объектов. Форма для редактирования любого объекта или просмотра списка объектов создается Web-расширением автоматически или может быть определена разработчиком вручную в проекте. Если разработчик описывает форму самостоятельно, то в его распоряжении специализированные элементы управления, которые не только позволяют организовать связь с теми или иными данными, но и предоставляют всю необходимую функциональность по просмотру и редактированию. Например, в полях ввода поддерживается выбор из форм-списков и подбор по первым буквам, в списках поддерживается иерархический просмотр, настройка пользователем фильтрации и сортировки. Система предоставляет большой набор стандартных действий по "обслуживанию" данных и самостоятельно организует связь между формами приложения. Таким образом обеспечивается единообразие двух вариантов пользовательского интерфейса (разумеется, с поправкой

на особенности среды Web), а также максимальная автоматизация их разработки, основанная на использовании информации из метаданных.

Одним из наиболее важных, на наш взгляд, следствий построения пользовательского интерфейса на основе стандартных средств предоставляемых платформой является то, что интерфейс всех прикладных решений "1С:Предприятия" построен единообразно. Это позволяет пользователям максимально применять накопленный ранее опыт. Освоив приемы работы с интерфейсом одного бизнес-приложения, пользователь легко начнет работать с любым другим прикладным решением "1С:Предприятия".

Интеллектуальные механизмы подготовки отчетов

В любом бизнес-приложении имеются два основных механизма, обеспечивающих взаимодействие пользователя с системой. Это интерфейс, предназначенный для ввода информации, и средства подготовки отчетов. Поскольку механизмы получения отчетов отвечают, прежде всего, за представление пользователю информации, необходимой для принятия управленческих решений, естественно, что в "1С:Предприятии" им уделено особое внимание.

Технология подготовки отчетов, примененная в "1С:Предприятии", содержит, по нашему мнению, целый ряд уникальных решений, не имеющих прямых аналогов в других продуктах. Прежде всего, следует отметить, что в платформе "1С:Предприятие" нет традиционного для большинства систем отдельного средства "генератор отчетов". Для подготовки отчетов разработчику предлагается целый ряд механизмов, которые могут использоваться в различных комбинациях и в сочетании с другими механизмами. Благодаря этому в "1С:Предприятии" отчеты органично вписываются в общий интерфейс приложения. Фактически, пользователь в процессе работы не видит грани между общим интерфейсом и механизмом отчетности. По нашему мнению, в современной экономической прикладной системе и не должно быть иначе: ведь аналитическая информация может использоваться во всех режимах и формах, а отчеты формируются по большей части уже не для вывода их на печать, а для интерактивного анализа. Поэтому в "1С:Предприятии" средства подготовки отчетности тесно интегрированы с другими механизмами и имеют мощные возможности для интерактивной работы.

Одним из наиболее интересных механизмов такого рода, предоставляемых платформой, является построитель отчетов, предоставляющий возможность с минимальными усилиями получить отчет с развитой функциональностью.

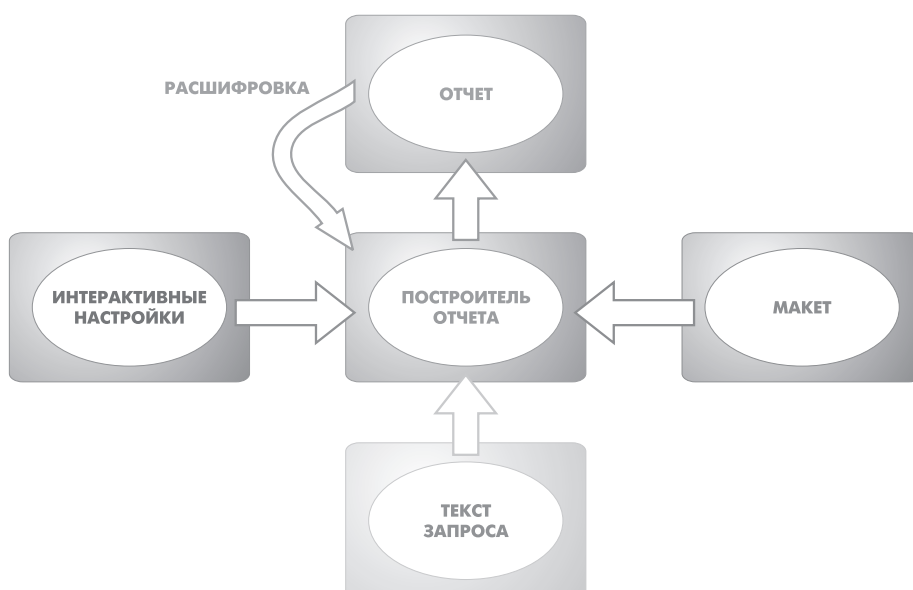
Нам хотелось добиться того, чтобы разработчик или пользователь кратко описывал, какую информацию необходимо проанализировать, а все остальное для получения полнофункционального управляемого отчета построитель делал бы самостоятельно.

Выборка исходной информации осуществляется на основе описаний, выполненных с помощью языка запросов "1С:Предприятия". Поскольку при этом активно используются готовые виртуальные таблицы учетных механизмов, текст запроса для достаточно сложного отчета будет занимать всего несколько строчек. Его вовсе не обязательно писать вручную. В большинстве случаев запрос формируется средствами визуального конструктора и зачастую вся процедура сводится к выбору источника данных и указанию полей итогов. Данный инструмент позволяет визуально формировать логические конструкции практически неограниченной сложности, включая объединения запросов и вложенные запросы, а также способен "поднимать" для редактирования текст запроса, расположенный в любом месте модуля.

На основании всестороннего анализа текста запроса построитель самостоятельно формирует всю инфраструктуру необходимую не только для генерации отчета, но и для тонкой его настройки пользователем, а также для навигации между отчетами. В результате создается форма, содержащая как собственно отчет, так и разнообразные средства для управления им – выбора включаемых в отчет полей, фильтрации информации по сложным критериям, группировки по строкам и колонкам, настройки упорядочивания данных и т.д. Отображаться отчет может в виде таблицы с многоуровневой иерархией строк и колонок, диаграммы, сводной диаграммы или сводной таблицы.

Заметим, что интерактивная настройка фильтрации обеспечивает такие возможности, как установка отбора по полям связанных таблиц (полученных "через точку"), указание в качестве критерия отбора множества значений или групп объектов, в иерархии которых нужно отбирать информацию. Например, пользователь может настроить отчет по закупкам товаров таким образом, чтобы в него вошли только накладные, содержащие товары, производители которых относятся к группам "Отечественные" и "Зарубежные". При этом построитель позволяет сохранять текущие пользовательские настройки отчета для последующей работы.

Кроме того, построитель автоматически включает в сформированный отчет всю информацию, необходимую для расшифровки отдельных ячеек отчета (drill-down). Эта расшифровка выполняется в интерактивном режиме – буквально одним щелчком мыши по выбранной ячейке отчета может быть автоматически сформирован более детальный отчет на основании данных этой ячейки и информации, содержащейся в исходном отчете:



Очень важно, что все эти возможности, представляемые разработчику в готовом виде, могут использоваться им в различных комбинациях. В результате он может гибко определять для каждого создаваемого отчета, какие "степени свободы" предоставить пользователю, указывая это в тексте запроса. Так, например, можно указать, по каким полям пользователь может устанавливать фильтрацию, упорядочивание, группировки и т.д. Сами элементы управления отчетом включаются в форму также по усмотрению разработчика. Поскольку все составляющие данного механизма могут использоваться независимо, разработчик может применять построитель не только для подготовки отчетов, но и для решения многих других задач. Например, он может создать процедуру группового изменения данных, в которой для управления выборкой данных будут задействованы все механизмы построителя, но затем полученная выборка будет не отображена в отчете, а пройдет заданную обработку, например, для выбранной группы товаров будут скорректированы цены.

Так же весьма важным эффектом использования этого механизма мы считаем то, что на его основе в прикладных решениях "1С:Предприятия" реализуются инструменты, в которых "продвинутые" пользователи действительно могут самостоятельно создавать достаточно сложные отчеты. На основе этого механизма построена специальная консоль отчетов, используемая в большинстве прикладных решений и предоставляющая пользователю возможность проектировать собственные отчеты, тут же их формировать, настраивать и даже создавать систему связанных, детализирующих друг друга отчетов.

Не были забыты и вопросы эффективности изобразительных средств. В "1С:Предприятии" для визуализации и печати отчетов используется прежде всего метафора табличного

документа (без средств вычислений). Основной идеей данного решения является использование модели электронной таблицы (spreadsheet) в качестве средства оформления отчетных документов, без использования ее как средства вычисления.

В отличие от большинства электронных таблиц табличный документ "1С:Предприятия" может содержать таблицы, имеющие очень большие размеры, как по вертикали, так и по горизонтали. У него богатый набор изобразительных возможностей, позволяющий с одной стороны создавать хорошо оформленные аналитические отчеты, а с другой – подготавливать регламентированные бланки с точной привязкой содержащихся в них объектов к локальным координатам. В отличие от стандартных электронных таблиц здесь разрешается описывать строки с различными настройками ширины колонок. Это необходимо для создания сложных отчетов включающих несколько разноформатных таблиц.

В большинстве случаев заполнение отчета выполняется на основании макета. Для этого в табличном документе предусмотрена гибкая система именования областей и описания параметров. Разработчик визуально проектирует макет, определяющий внешний вид отчета, выделяя в нем различные области, которые должны заполняться данными, и описывая их параметры. При формировании отчета происходит совмещение областей с данными и таким образом содержательная часть отчета объединяется с его внешним оформлением. Следует заметить, что макет отчета может как формироваться строителем автоматически, так и создаваться разработчиком. Допускается использование готовых стилей оформления отчетов и проектирование собственных.

Важно, что средство генерации табличных документов служит не только для получения статических печатных форм. В нем имеется мощный интерактивный механизм для управления многоуровневыми группировками, расшифровки ячеек отчета (drill-down), автонастройки ширины колонок и т.д. Допускается сохранение отчета в различных форматах (html, xls, txt). Кроме того, для интерактивного анализа многомерной информации в табличном документе может использоваться сводная таблица, которая позволяет непосредственно управлять группировкой информации и составом отображаемых данных, не обновляя отчета.

Чрезвычайно важным качеством табличного документа является возможность использования его в качестве полнофункционального механизма ввода и редактирования информации. Такой документ может содержать любые элементы управления, используемые в формах "1С:Предприятия", как в самих ячейках и рисунках, так и поверх ячеек. Это особенно полезно в тех случаях, когда одна и та же форма должна представлять собой и отчет, и инструмент для редактирования информации. Примером может служить отчет о плановых показателях, которые разрешается корректировать.

Для визуального представления аналитической информации разработчику предлагается набор различного вида диаграмм, а также диаграмма Ганта, сводная диаграмма и дендрограмма. Причем эти элементы могут включаться как в формы, наравне с другими элементами управления, так и в табличные документы для получения комплексных аналитических отчетов.

Весьма полезен также механизм интеллектуального анализа данных (data mining), с помощью которого прикладное решение можно с минимальными затратами оснастить такими аналитическими инструментами, как кластерный анализ, дерево решений и т.д.

Важная особенность платформы "1С:Предприятие" состоит в том, что для формирования отчетов любой сложности разработчику не придется подключать дополнительные внешние механизмы. При этом все внутренние инструменты хорошо интегрированы, базируются на единой системе понятий и максимально используют возможности друг друга. Например, сводная таблица, располагаемая в табличном документе, для того чтобы автоматически модифицировать запрос к базе данных по мере того, как пользователь визуально указывает новые уровни группировки, обращается к функциям строителя отчетов. Разумеется, в системе присутствует и конструктор отчетов, который позволяет проектировать их интерактивно, однако важно, что конструктор только "собирает" отчет из имеющихся механизмов ("кубиков"), применяемых и для решения множества других задач. Как показала практика, при таком подходе на разработку достаточно сложного отчета уходит не более нескольких минут.

Построение распределенных и интегрированных информационных систем

Еще одной "козырной картой" платформы "1С:Предприятие" являются механизмы обмена данными.

Изначально решение о создании механизма обмена данными, предназначенного для территориально распределенных информационных баз и не требующего постоянного соединения, было обусловлено в большей степени отсутствием в распоряжении отечественных предприятий качественных высокоскоростных телекоммуникационных каналов. Однако некоторое время спустя стало очевидно, что разработка таких механизмов лежит в русле одного из наиболее перспективных направлений мировой инженерной мысли.

Сегодня в распоряжении разработчика приложений имеется мощный набор механизмов обмена, способный решать самые разнообразные задачи. Кроме традиционной поддержки территориально распределенных информационных систем, с его помощью осуществляется интеграция с другими прикладными программами, а также строятся сложные гетерогенные информационные системы, включающие наряду с решениями на платформе "1С:Предприятие" еще и внешние приложения.

Использование технологий обмена для решения столь широкого спектра задач объясняется тем, что в распоряжении разработчика не монолитное решение, а набор механизмов, которые могут применяться как по отдельности, так и совместно, в любых сочетаниях. Эти механизмы поддерживают работу с XML документами, XML-сериализацию данных, инфраструктуру сообщений, службу регистрации изменений, планы обмена, управление распределенной информационной базой:



Практически уникальным, по нашему мнению, качеством данного набора механизмов является то, что он обеспечивает очень высокий уровень готовности системы к работе в распределенной среде – организация обмена практически не требует дополнительных затрат на разработку. Нужно просто задать в интерактивном режиме состав данных, участвующих в обмене, а механизм обеспечит формирование сообщений и их загрузку. При этом система автоматически организует обмен только измененной информацией, отслеживает получение сообщений, определяет необходимость повторной отправки данных, разрешает коллизии и проверяет целостность загружаемой информации. Гибкие возможности настройки позволяют сформировать практически любую топологию схемы узлов обмена (звезда, снежинка, схемы без центрального узла). Состав данных, участвующих в обмене, и правила разрешения коллизий могут задаваться произвольно. При этом механизмы обмена с одной стороны минимизируют объем передаваемых данных (пересылаются только измененные данные), а с другой – гарантируют устойчивость к потере сообщений. Иными словами, система способна функционировать как в условиях гарантированной доставки сообщений, так и без таковой.

Наличие в платформе эффективных, не требующих сложной настройки механизмов обмена обязано, прежде всего, общей архитектуре построения прикладных решений, реализованной в платформе "1С:Предприятие". Мы уже говорили, о том, что органичный переход к распределенным и интегрированным системам обеспечивается объектной техникой манипулирования данными, используемой в "1С:Предприятии". Система соответствующих объектов обеспечивает штатные возможности сохранения (persistence) любых прикладных данных в формате XML. Кроме того, благодаря наличию стандартных прототипов прикладных объектов платформа способна автоматически задавать для каждого объекта необходимую гранулярность передачи изменений и стратегию обмена в соответствии с его назначением. Основной проблемой большинства известных систем обмена и репликации является то, что в заложенной в них семантике описания данных не содержится сведений о том, какими порциями выполнять обмен, как разрешать коллизии, как обеспечивать логическую целостность и непротиворечивость данных. Из-за этого разработчику при использовании подобных механизмов обычно приходится детально описывать процедуры обмена для каждого информационного массива.

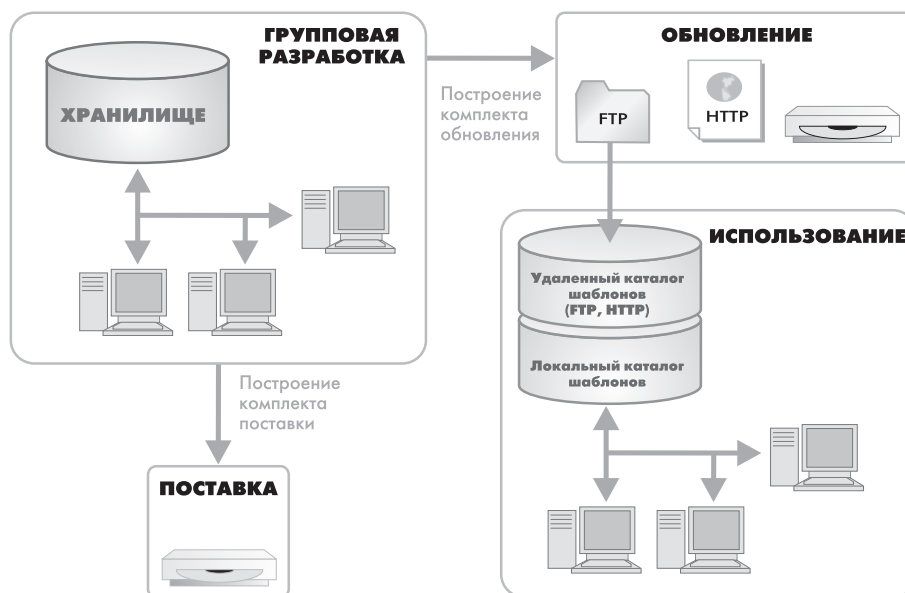
Механизмы обмена "1С:Предприятия" функционируют на уровне стандартных прототипов прикладных объектов и поэтому имеют в своем распоряжении всю необходимую информацию такого рода изначально. Разработчику необходимо только указать, что в обмене участвуют объекты определенного вида. Далее все действия, по регистрации изменений, формированию сообщений, загрузке данных и разрешению коллизий будут выполняться автоматически. Разработчик, конечно, имеет возможность вручную выполнить "тонкую настройку", но она потребует только в отдельных случаях.

Другая важная особенность механизмов обмена "1С:Предприятия" – их соответствие передовым мировым концепциям интеграции информационных систем. Мы прекрасно понимаем, что сегодня ни один разработчик экономического софта не может считать себя "царем горы", которому нет необходимости заботиться о тесном взаимодействии с системами других вендоров. В данном контексте взаимодействие – не просто умение вызвать какую-то функцию из другой программы или экспортировать в нее какие-либо данные. Речь идет о построении сложных гетерогенных систем, в которых несколько разнородных прикладных решений образуют слаженно действующий ансамбль. Очевидно, что эта тенденция станет в обозримом будущем одной из доминирующих на рынке экономических систем. Механизмы обмена "1С:Предприятия" имеют высокую готовность к работе в гетерогенных системах сегодняшнего и завтрашнего дня не только за счет того, что весь обмен производится на формате XML, но, что еще более важно, в силу идеологического соответствия реализованных решений указанной тенденции. Уже сегодня механизмы обмена "1С:Предприятия" позволяют осуществлять взаимодействие не только с отдельными приложениями, но и с перспективными интеграционными платформами.

Поставка и обновление прикладных решений

Практически все производители прикладных программ предусматривают в них механизмы для обновления версий. Однако для таких систем как "1С:Предприятие" подобные задачи представляют особую сложность. Это объясняется, во-первых, тем, что экономические системы обновляются гораздо чаще, чем другие, а во-вторых, тем, что достаточно часто практикуется изменение прикладных решений "1С:Предприятия" "на местах" с целью адаптации тиражного продукта к потребностям конкретного клиента. Последняя особенность представляет особую сложность, так как при выпуске разработчиком любого обновления прикладного решения возникает необходимость синхронизации сделанных им изменений с теми, что были внесены в процессе внедрения у заказчика.

Для решения этих и множества других задач, связанных с поставкой и обновлением прикладных решений, в "1С:Предприятии" реализован целый комплекс механизмов ориентированных как на разработчиков, так и на пользователей. Они охватывают весь технологический процесс поставки и поддержки: подготовку дистрибутивных файлов, инкрементных обновлений и установочных комплектов поставки, публикацию обновлений в Интернете, автоматический их поиск и выполнение, управление составом поддержки на уровне объектов конфигурации и т.д.:



Одним из наиболее существенных элементов технологии обновления является механизм, обеспечивающий синхронизацию изменений, сделанных поставщиком прикладного решения с изменениями, внесенными при внедрении на конкретном предприятии. Он предоставляет мощные функции сравнения и анализа изменений, а также средства управления их синхронизацией. Администратор или разработчик может детально настроить синхронизацию обновлений вплоть до отдельных объектов, отдельных свойств и отдельных процедур модулей. Например, если специалист, отвечающий за сопровождение прикладного решения на предприятии, отметит объекты, которые намерен поддерживать самостоятельно, они не будут в дальнейшем обновляться при установке очередного обновления от поставщика. Если объекты необходимо объединить, то для упрощения синхронизации изменений можно настроить приоритеты такого объединения.

Механизмы обновления позволяют построить сложную многоуровневую систему сопровождения. Поставщики специализированных вертикальных решений могут находиться на поддержке у разработчиков универсальных приложений, и обеспечивать в свою очередь поддержку собственных клиентов. Пользователь при этом будет эксплуатировать прикладное решение, отдельные части которого сопровождаются разными фирмами.

Поскольку механизм обновления в значительной мере основан на том, что все прикладное решение описывается с помощью метаданных, а бизнес-логика приложения строится на стандартных прототипах прикладных объектов, система способна поддерживать стратегию обновления с учетом специфики архитектуры прикладного решения, контролировать его логическую целостность и предоставлять пользователю информацию об изменениях в наглядной форме.

А также...

Допускается реализация всего интерфейса прикладного решения на нескольких языках. Каждый пользователь может работать с одним из заложенных в него языков. При этом не нужно выносить все строки в отдельные файлы. Элементы интерфейса редактируются на нескольких языках "по месту" – в формах, макетах печатных документов, в меню и т.д. Разработчик может в любой момент переключиться и редактировать интерфейс форм на другом языке. Допускается также поочередное редактирование конкретной надписи на всех поддерживаемых языках. Если необходимо расширить языковую базу, можно собрать все текстовые элементы в общий список с тем, чтобы перевод одинаковых надписей отражался сразу во всех

компонентах интерфейса. Общая стратегия интернационализации включает также перевод системного интерфейса, хранение всех строк в кодировке UNICODE, форматирование дат и чисел в соответствии с особенностями различных стран и языков, построение правил сортировки с учетом национальных стандартов и т.д.

Выполнение алгоритмов бизнес-логики разработчик может по своему усмотрению переносить на сервер "1С:Предприятия". Это позволяет ему управлять распределением нагрузки между клиентом и сервером. При этом от разработчика не требуется специальных навыков построения трехуровневых архитектур, знания сетевых протоколов и т.д. Все технологические детали система берет на себя и обеспечивает рациональное использование серверных ресурсов за счет поддержки stateless-модели, кэширования и разделения системных ресурсов и т.д.

В системе защиты прав доступа "1С:Предприятия" реализована возможность настройки ограничений на уровне отдельных записей, а также полей таблиц. Это позволяет, например, предоставить пользователю доступ к именам сотрудников только в пределах своего департамента, а к данным по окладу и семейному положению – только для своих непосредственных подчиненных. Настройка таких ограничений требует лишь задания формального условия, после чего система контролирует все действия пользователей автоматически, разработчику не нужно учитывать эти ограничения в явном виде при реализации различных интерфейсных режимов и алгоритмов бизнес-логики.

Обеспечиваются две стратегии проверки прав доступа. В зависимости от решаемой задачи, система контроля прав доступа может как отвергнуть обращение пользователя к закрытым для него данным, так и просто исключить недоступные записи из выборки, предоставив пользователю только доступную информацию.

Одним из мощных инструментов разработчика в "1С:Предприятии" является механизм сравнения и объединения прикладных решений. Мы уже упоминали о нем, когда говорили о возможностях поддержки, однако этот механизм может весьма эффективно использоваться и в процессе разработки. Данный инструмент может использоваться как для анализа различий, так и для переноса части функций из одного приложения в другое. Он обеспечивает удобное визуальное представление различий между прикладными решениями и имеет гибкие возможности настройки логики сравнения. При сравнении родственных конфигураций данный механизм автоматически сопоставляет даже переименованные объекты, используя для этого внутреннюю идентификацию метаданных. Соответствие сравниваемых объектов независимо от совпадения их имен можно настроить и вручную, после чего оно будет учтено и во всех ссылках на такие объекты, имеющихся в других объектах прикладного решения. Кроме того, механизм сравнения содержит средства для визуального представления отличий форм интерфейса и макетов печатных документов. Благодаря использованию структуры метаданных и средств визуального сравнения интерфейсных объектов, с помощью нашей платформы удастся решать гораздо больше задач, чем в тех системах, где механизмы сравнения базируются на сопоставлении файлов исходных кодов программ.

В "1С:Предприятии" поддерживается протоколирование действий пользователей в журнале регистрации. Система имеет несколько уровней протоколирования, которые могут функционировать автоматически, не требуя дополнительных усилий со стороны разработчика. Прежде всего, разумеется, фиксируются начало и окончание сеансов работы пользователей, административные операции с информационной базой, а также регистрируемые ошибки. За счет того, что все изменения в БД выполняются исключительно в объектной технике (через объекты, отвечающие за манипулирование данными), система может автоматически регистрировать их в журнале, причем будет делать это независимо от того, как они выполнялись (интерактивно или программно). Разработчик может также реализовать внесение в журнал любых других записей, которые система не способна фиксировать автоматически, например, информацию об отправке факса или формировании отчета. Важно заметить, что журнал регистрации реализован как отдельный системный механизм. Он не использует для хранения информации базу данных "1С:Предприятия", а, следовательно, его ведение не создает существенной дополнительной нагрузки на систему и не замедляет работу пользователей.

Подводя итоги

В этой статье мы попытались перечислить наиболее интересные технологические новшества платформы "1С:Предприятие". При этом мы старались показать, что все реализованные в ней решения основаны на использовании единой модели и единой архитектуры.

Важно и то, что все механизмы платформы в той или иной степени используют возможности других механизмов. Например, почти все они опираются на структуру метаданных и наличие системы прототипов прикладных объектов. Таким образом, хотя каждая из возможностей "1С:Предприятия" может быть достаточно ценна сама по себе, наибольший интерес представляет рассмотрение всей их совокупности в рамках целостной модели.

Сформулируем еще раз несколько общих принципов, лежащих в основе модели "1С:Предприятия":

Архитектура платформы и ее инструментальные средства реализованы таким образом, чтобы разработчик бизнес-приложения мог максимально абстрагируясь от низкоуровневых технологий, сосредоточиться на решении прикладных задач своей предметной области.

Вся разработка от построения структур данных, до проектирования элементов интерфейса и подключения средств интеграции ведется в одной системе понятий, что существенно ускоряет обучение специалистов и повышает производительность их труда.

Платформа содержит готовые ответы практически на все вопросы, возникающие у разработчика прикладного решения, начиная от того, как отражать в информационной базе данные предметной области и кончая процедурами поставки, поддержки и администрирования. Соответственно, разработчику не требуется осваивать разнородные технологии и решать вопросы их совместного использования.

Наличие единой, сквозной высокоуровневой модели и реализующих ее технологий позволяет, по нашей оценке, на порядок сократить затраты на создание и поддержку прикладных решений.

Начиная в 1996 г. работы в этом направлении, и даже выпустив первую версию платформы, мы, конечно, не были абсолютно уверены в правильности выбранного пути. За прошедшее с тех пор время концепция системы успешно развивалась, и сегодня она воплощена в платформе "1С:Предприятие 8.0". Строго говоря, в мире не так уж много специализированных технологий, ориентированных именно на быстрое создание бизнес-приложений. Однако становится все более очевидным, что данный путь лежит на стратегическом направлении развития бизнес-софта.

По оценкам многих специалистов, с которыми нам приходилось общаться, ситуация в России и в странах бывшего СССР, где сложилась целая индустрия разработки и поддержки бизнес-решений, основанная на специализированной технологической платформе, по мировым меркам является уникальной. По нашим наблюдениям, не существует аналогичной специализированной платформы, на которой различными фирмами создавалось бы такое количество разнообразных массовых бизнес-приложений, покрывающих потребности существенной части рынка какого-либо региона. Соответственно, для многих специалистов наши подходы и решения представляют особый интерес, так как реальный эффект от их реализации хорошо заметен на практике.

Разумеется, все изложенные выше соображения (включая и анализ современных тенденций) являются только нашей позицией и результатом наших исследований. Мы ни в коем случае не претендуем на абсолютную истину, которой в этих вопросах, по-видимому, и не существует. Наверняка, у многих читателей есть своя, отличная от нашей, точка зрения на проблемы, затронутые в статье. Мы будем признательны за отзывы (в том числе критические) по данной статье.

Фирма «1С»
Москва, 123056, а/я 64
Селезневская ул., 21
тел. (495) 737-92-57,
факс (495) 681-44-07
www.1c.ru, 1c@1c.ru

