
ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

УДК 004.654

НОВЫЕ ТЕНДЕНЦИИ В РАЗВИТИИ АРХИТЕКТУРЫ СОВРЕМЕННЫХ КОРПОРАТИВНЫХ СИСТЕМ

Р.Ж. Кушалиев, А.А. Ветрова

В статье рассматривается проблематика и актуальность разработки новых подходов к организации данных, хранящих информацию о структуре предприятий и взаимосвязи их бизнес-процессов с целью разработки автоматизированных приложений, обладающих достаточной гибкостью и не требующих частой модернизации.

Ключевые слова: база данных, объектно-ориентированное программирование, бизнес-приложения, модернизация приложения, проектирование базы данных.

Keywords: database, object-oriented programming, business applications, re-engineering applications, database design.

В современных условиях рыночных отношений основным звеном экономики являются предприятия, поскольку именно на этом уровне создается нужная обществу продукция, оказываются необходимые услуги. Но все больше предприятий в ходе своей деятельности сталкиваются с проблемами улучшения управляемости компании:

- улучшения контроля и ускорения бизнес-процессов;
- улучшения возможности их отслеживания и возможности оценки параметров, характеризующих качество исполнения бизнес-процессов.

Единственным способом реализации подобных задач является внедрение автоматизированных систем, выполняющих вышеуказанные функции. Однако в настоящее время попытки разработки и внедрения подобных систем наталкиваются на существенные сложности, связанные с решением сложных задач формализации управленческих процессов.

Основной задачей автоматизации документооборота является перенос в область автоматизации формализованной части управленческого процесса. Следует отметить, что сам процесс формализации является периодическим и итеративным. По мере внедрения определенного процесса выявляются его недостатки и необходимость внесения изменений в его структуру. При этом уже внедренные приложения по автоматизации документооборота, которые разрабатывались для конкретного предприятия-заказчика и решали необходимые для предприятия задачи, могут потерять актуальность, в результате чего требуется их модернизация.

Существующие подходы к разработке приложений, использующие архитектуру реляционных баз данных, не обладают достаточной гибкостью, так как ориентированы на хранение внутри таблиц базы данных (БД) информации о сущностях информационной системы и связях между ними. Поэтому изменения в работе самого предприятия, которые неизбежно происходят с течением времени, требуют внесения изменений в задачи, которые решает имеющаяся система автоматизации. При этом, как правило, перед разработчиками ставятся новые задачи, и они занимаются реинжинирингом ранее разработанной системы. Такой принцип работы долгое время устраивал всех, так как изменения на предприятиях происходили не так часто, и в среднем система без изменений могла просуществовать более 5 лет.

Однако в условиях рыночных отношений процесс изменения как структуры самих предприятий, так и бизнес-процессов, требующих автоматизации, происходит все чаще, а постоянный реинжиниринг систем автоматизации становится весьма дорогостоящим фактором, снижающим эффективность работы предприятия и отрицательно влияющим на его конкурентоспособность на рынке. Подобная ситуация перестает устраивать как разработчи-

ков, которые вынуждены постоянно сокращать сроки разработки приложений, чтобы они не потеряли своей актуальности, так и предприятия, которые вынуждены нести высокие затраты, связанные с автоматизацией бизнес-процессов.

Таким образом, актуальной становится задача разработки новых подходов к организации данных, хранящих информацию о структуре предприятий и взаимосвязи их бизнес-процессов с целью разработки автоматизированных приложений, обладающих достаточной гибкостью и не требующих частой модернизации в стремительно меняющихся рыночных условиях.

Недостатки традиционного (реляционного) подхода к организации БД и разработки приложений на его основе. Структуры многих предприятий состоят из дочерних компаний, филиалов, отделов, рабочих групп и т.д. Любая группа объектов, в которой один объект может быть «родителем» для произвольного числа других объектов, организована в виде иерархического дерева.

В свою очередь, архитектура реляционных баз данных ориентирована на хранение внутри таблиц БД информации о сущностях информационной системы и связях между ними. Каждая из записей таблицы содержит информацию об одном экземпляре объекта. Организация хранения информации о независимых друг от друга экземплярах сущностей (т.е. так называемых «плоских» данных) не вызывает затруднения. Однако организация хранения в реляционных БД информации о «вложенных» друг в друга сущностях, т.е. иерархических данных, не совсем проста, связана с рядом сложностей.

Для поиска записи-цели в иерархической структуре вначале необходимо определить путь доступа, а затем просмотреть все записи, лежащие на этом пути, – шаг за шагом. Насколько запутанной являются схемы представления иерархических данных в реляционных БД, настолько и трудоемким является проектирование конкретных прикладных систем на их основе. Как показывает опыт, длительные сроки разработки прикладных систем нередко приводят к тому, что они постоянно находятся в стадии разработки и доработки. Кроме того, принцип навигации не позволяет существенно повышать уровень манипулирования данными, чтобы сделать его доступным пользователю.

Очевидно, что доступ к данным по путям, не предусмотренным при создании БД, может потребовать неразумно большого времени. Повышая эффективность доступа к данным и сокращая таким образом время ответа на запрос, принцип навигации вместе с этим повышает и степень зависимости программ и данных. Обработывающие программы оказываются жестко привязанными к текущему состоянию структуры БД и должны быть переписаны при ее изменениях. Операции модификации и удаления данных требуют переустановки указателей, а манипулирование данными остается ориентированным на запись.

Таким образом, системам, разработанным в реляционных БД и работающим с большими, постоянно увеличивающимися объемами данных, постоянно грозит опасность стать громоздкими, застывшими и слишком сложными системами. Поэтому очевидно, что старые принципы разработки приложений в области автоматизации документооборота принципиально не могут быть реализованы в современных условиях и не устраивают как заказчиков, так и разработчиков.

Объектно-ориентированная модель организации данных, ее преимущества. Исходя из всего выше представленного, ясно, что методология структурирования данных должна иметь объектную направленность, связанную с выделением объектов и применения объектно-ориентированного подхода и программирования.

Объектно-ориентированный подход – это возможность работать со сложноструктурированными данными и преодолевать ограничения, связанные с использованием реляционной технологии СУБД. Реляционные БД вынуждают пользователей представлять иерархические данные в терминах кортежей многих отношений. А для выборки данных, разбросанных таким образом по многим отношениям, реляционная БД должна выполнять дорогостоящие операции соединения. Иерархические данные в объектно-ориентированных языках

ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

представляются естественным образом благодаря тому, что значение атрибута объекта, в свою очередь, может быть объектом.

Объектно-ориентированные технологии обеспечивают естественный переход от концептуальной структуры БД к логической структуре БД. В отличие от реляционных БД при проектировании объектно-ориентированных БД (ООБД) не требуется декомпозиция и нормализация объектов, выделенных на этапе концептуального проектирования. Объекты представляются в том же виде, что наделяет объектно-ориентированные структуры наглядностью и прозрачностью и позволяет значительно сократить время на их разработку и повысить степень преемственности. ООБД определяют возможность создания и использования сложных типов данных. При этом не требуется модификации ядра ООБД, и для создания нового типа необходимо унаследовать характеристики любого имеющегося типа, наиболее подходящего по своему поведению и состоянию, расширить недостающие операции и атрибуты и переопределить уже имеющиеся. Полученные объектно-ориентированные структуры обладают высокой степенью модульности, что позволяет вносить в них изменения наиболее простым и безболезненным способом. При этом изменения влияют на один класс (или связанную подсистему классов) и могут эффективно управляться и проверяться.

Нет необходимости говорить, насколько широкое распространение получила в последнее время идеология объектов. Термин «объект» в объектно-ориентированной (ОО) системе означает комбинацию данных и программ, представляющих некоторую сущность реального мира. Данные состоят из компонентов произвольного типа, называемых атрибутами. Каждая программа называется методом. Пользователи не могут увидеть, что у объекта внутри, но могут им пользоваться, обращаясь к его программной части. Это немногим отличается от обычного вызова процедуры, когда пользователи обращаются к ней, подставляя значения входных параметров и получая результаты в виде выходных параметров.

Каждый объект имеет состояние и поведение. Состояние объекта – набор значений его атрибутов. Поведение объекта – набор методов (программный код), оперирующих над состоянием объекта. Значение атрибута объекта – это тоже некоторый объект или множество объектов. Состояние и поведение объекта инкапсулированы в объекте; взаимодействие между объектами производится на основе передачи сообщений и выполнении соответствующих методов.

Допускается порождение нового класса на основе уже существующего – наследование. В этом случае новый класс, называемый подклассом существующего класса (суперкласса), наследует все атрибуты и методы суперкласса. В подклассе, кроме того, могут быть определены дополнительные атрибуты и методы. Наследование классов, набор классов образует древовидную иерархию. При поддержании множественного наследования классы связаны в ориентированный граф с корнем, называемый решеткой классов. Объект подкласса считается принадлежащим любому суперклассу этого класса. Таким образом, такой подход оптимально подходит для решения проблем, связанных с проектированием систем автоматизации документооборота. Иерархия классов в ООБД подобна структуре управления предприятий и позволяет использовать новые языки программирования и внедрять в систему новые средства коммуникации.

Практика показывает, что наиболее распространенная сегодня реляционная технология мало пригодна для работы со сложными объектами. А применение ОО подхода позволит производить изменения в приложениях системы без существенных вмешательств и изменения программного кода.

Новые подходы к организации данных для хранения информации о структуре предприятий и взаимосвязи ее отдельных элементов. Делая вывод о том, что наиболее распространенная сегодня реляционная технология мало пригодна для работы со сложными объектами, а применение ОО подхода позволит производить изменения в приложениях без существенных вмешательств и изменения программного кода, нельзя не отметить, что исследования в данном направлении начаты сравнительно недавно. При всех достоинствах современной объектной технологии разработки БД имеется несколько препятствий, которые

удерживают разработчиков от принятия решения о переходе с реляционной технологии на объектную. Основным препятствием является значительный объем разработок, опирающихся на реляционные СУБД. Ведь при переходе на объектную технологию необходимо многое начинать «с нуля», и поэтому возникает вопрос целесообразности такого перехода. Кроме того, объектная технология, поддерживаемая в ряде постреляционных СУБД, не имеет развитого и стандартизированного языка генерации отчетов и анализа данных, каким является структурированный язык запросов SQL.

Для широкого применения ОО подхода к автоматизации бизнес-процессов на предприятиях необходимо провести серьезные исследования в области формализованного описания организационных структур на основе ОО подхода и разработки коллекций объектов для его реализации. Поэтому для успешного перехода с реляционных СУБД необходимо разработать типовые решения по организации структур данных для хранения информации о предприятии и бизнес-процессах, основанных на объектно-ориентированном подходе.

УДК 622.691.4.01

КОМПЛЕКС ПРОГРАММ ДЛЯ ОЦЕНКИ ОСТАТОЧНОГО РЕСУРСА МАГИСТРАЛЬНЫХ ТРУБОПРОВОДОВ

В.Г. Лим, В.Ю. Чаплыгин

В данной статье затрагивается вопрос оценки остаточного ресурса магистральных трубопроводов, имеющих продольные поверхностные повреждения, обусловленные наружной коррозией или механическим воздействием на трубу. Содержится обоснование и описание основной концепции расчетного метода, известного под названием «критерий B31G», используемого в разработанном программном комплексе.

Ключевые слова: магистральный трубопровод, остаточный ресурс, стресс-коррозия, критерий B31G, минимальный нормативный заданный предел текучести (SMYS), фактор Фолиаса.

Keywords: main pipeline, residual resource, stress-corrosion, criterion B31G, specified minimum yield strength (SMYS), Folias factor.

Трубопроводные системы являются сложными инженерными сооружениями, с помощью которых осуществляется транспортировка нефти, газа и нефтепродуктов. Энергетическая безопасность значительной части регионов нашей страны и многих из зарубежных пользователей напрямую связана с надежностью функционирования российских трубопроводных систем. В результате длительной эксплуатации трубопроводов постоянно снижаются их эксплуатационные параметры, заложенные в проекте, повышается склонность металла к постепенному разрушению, что объясняется последствиями накопления в нем дефектов, внутренних микронапряжений или даже отдельных структурных изменений [1, 3]. Изменение эксплуатационных параметров трубопровода и воздействие различных факторов техногенного характера приводит к снижению ресурсной прочности труб.

По сведениям Госгортехнадзора России, более половины аварий на трубопроводах связано с накоплением повреждений в металле трубы и в сварных швах. Поэтому применение современных средств технической диагностики и своевременное выполнение ремонтных работ могут способствовать значительному сокращению количества аварий. Однако в настоящее время на территории России фактический срок эксплуатации большинства магистральных трубопроводов приближается к тому моменту, когда значительно возрастает ин-