

На правах рукописи

Макунин Алексей Анатольевич

ТЕХНОЛОГИЯ ПОСТРОЕНИЯ МОДУЛЬНЫХ
АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ
ДЛЯ СЛОЖНЫХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ И ЕЕ ПРИМЕНЕНИЕ
НА ПРИМЕРЕ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ
СИСТЕМЫ МУНИЦИПАЛЬНОГО ЗАКАЗА
ОРГАНОВ МЕСТНОГО САМОУПРАВЛЕНИЯ

Специальность 05.13.11 —
«Математическое и программное обеспечение вычислительных
машин, комплексов и компьютерных систем»

Автореферат
диссертации на соискание учёной степени
кандидата технических наук

Томск – 2006

Работа выполнена в Томском государственном университете.

Научный руководитель – д.т.н., доцент Скворцов Алексей Владимирович.

Официальные оппоненты:

д.т.н., профессор Янковская Анна Ефимовна;

к.т.н., доцент Гриценко Юрий Борисович.

Ведущая организация:

Сибирский государственный аэрокосмический университет.

Защита состоится 23 марта 2006 г. в 10:30 на заседании Диссертационного совета Д 212.267.08 при Томском государственном университете (634050, г. Томск, пр. Ленина, 36).

С диссертацией можно ознакомиться в библиотеке Томского государственного университета.

Отзывы на автореферат (2 экз.), заверенные печатью, высылать по адресу: 634050, г. Томск, пр. Ленина, 36, ученому секретарю ТГУ.

Автореферат разослан 7 февраля 2006 г.

Ученый секретарь
диссертационного совета, д.т.н., доцент

Скворцов А.В.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы. Мир вокруг нас бесконечно разнообразен и широк. Сложные предметные области содержат в себе очень большое количество объектов разных типов, большое число связей между ними, а также сложные процессы обработки информации. Разработка информационных систем для таких предметных областей также очень сложна и трудоемка. Как правило, для их построения применяется методология «сверху-вниз», что позволяет полностью охватить всю предметную область и точнее ее формализовать, правда, потратив достаточно большое время и много других ресурсов. Такой подход к разработке не единственный — используя ряд иных методик, можно также разрабатывать самые разные информационные системы. Выбор методики зависит от задачи, от привычек и общих условий работы коллектива авторов, а также от корпоративных стандартов организации-разработчика.

Объем проектных работ и кодирования определяется сложностью предметной области. Поэтому управление сложностью является важным моментом разработки почти всех программных систем. При этом чаще всего используется фундаментальный прием ограничения сложности — разделение на части (блоки, модули) по функциональному назначению. Но не все существующие методики модульного построения просты и удобны. Более того, зачастую они не обеспечивают механизмов развития и расширения программного продукта в целом. Безусловно, есть технологические решения, позволяющие строить расширяемые программы за счет дополнительных модулей (plug-in, add-in) или других механизмов, но большинство из них весьма узко специализированы.

Существует широкий класс задач, связанный с разработкой информационных систем для предметных областей, чья структура слишком сложна или полностью не определена, а существующие условия предполагают ограниченные сроки разработки. Для решения таких задач требуется технология, позволяющая поэтапно создавать программный продукт частями, запуская их в эксплуатацию по мере разработки компонентов и расширения рамок охвата предметной области. Подобный инструментарий позволяет упростить создание приложений для автоматизации сложных предметных областей, а также их развитие и сопровождение.

Использование модульного подхода в качестве основы для такого инструментария позволяет не только просто строить сложные приложения, собирая их из кирпичиков, но и обеспечивать их взаимозаменяемость для доработки и расширения возможностей информационных систем. Основные преимущества модульной архитектуры этим не ограничиваются. Следует упомянуть возможность выборочной компоновки *автоматизированных информационных систем* (АИС), многократное использование кода и разработанных классов.

Чтобы все преимущества модульной технологии стали доступными, необходимо решить ряд задач, связанных с организацией среды функционирования, формализацией межмодульного взаимодействия и принципиально иными способами определения содержательной стороны предметной области, создающими условия для структурного и функционального слияния ее частей при поэтапной разработке модульного программного обеспечения.

Большой вклад в развитие модульных технологий и расширяемых программ внесли Н. Вирт, М.М. Горбунов-Посадов, Д. Рамодин. Распределенные модели модульных программных продуктов развивались в работах К.В. Ахтырченко, В.В. Леонтьев, Д. Бокс, Е. Игумнов. Другими методами структуризации в программных системах занимались А. Бельченко, Ю.А. Загорулько, И.Г. Попов, Л.А. Калинин, Н.А. Колчанов, Н.Л. Подколотный, А. Поточкин, О.И. Росеева, Дж. Майопулос, Р. Мочинг-Питрик, Х. Такеда, М. Такаи, Т. Нишида, Т.Р. Грубер.

Еще один актуальный вопрос — это разработка принципов совместного использования систем хранения информации и модульных технологий. В развитие теории и методологии реляционных баз данных значительный вклад внесли Э. Кодд, Дж. Дейт, Дж. Мартин, Д. Мейер; объектно-ориентированные базы данных развивались М. Аткинсоном, Ф. Бансильоном, Д. ДеВиттом, К. Дитрихом, Д. Майером, С. Здоником; над объектно-реляционными моделями и объектно-ориентированными базами данных наиболее плодотворно работали С. Кузнецов, Е. Григорьев, С. Савушкин, М. Стоунбрейкер.

Применение модульного варианта архитектуры программных продуктов и использование систем управления базами данных дает возможность автоматизировать сложные предметные области, например, такие как *экономика муниципальных образований*, охватить которые сразу не удастся по причине огромного числа процессов и взаимодействующих объектов, и главное — кардинальных различий функциональности.

Цель работы заключается в разработке технологии построения гибких, модульных, легко конфигурируемых и расширяемых автоматизированных информационных систем, позволяющей сократить временные и ресурсные затраты на создание программного обеспечения, упростить его сопровождение и развитие.

В рамках указанной цели были поставлены **следующие задачи**:

1. Изучить и проанализировать российский и мировой опыт технологий разработки и архитектуры класса информационных систем, в основу которых положены принципы расширяемости, изменяемости и интеграции отдельных АИС в программные комплексы.

2. Разработать технологию построения автоматизированных информационных систем для широкого класса задач, ускоряющую и упрощающую

щую создание и сопровождение комплексов программ для сложных предметных областей.

3. Апробировать разработанную технологию на примере создания программного комплекса для информационной поддержки системы муниципального заказа органов местного самоуправления в Российской Федерации.

Метод исследования. При выполнении диссертационной работы использовались методы системного анализа, теории построения реляционных баз данных, объектно-ориентированного проектирования, теории графов, теории сложности алгоритмов.

Научная новизна:

1. В результате анализа проблем построения информационных моделей для сложных предметных областей с большим количеством информационных связей и объектов, играющих различные роли в разных АИС, для расширения объектно-реляционной модели предложена новая интерпретация понятий *аспекта* и *контекста*, упрощающих и ускоряющих построение информационных систем за счет введения новых принципов структуризации в моделях данных. На основе предложенных понятий разработана модель *дерева объектов (О-дерева)* для отображения объектов и формализм правил его построения, позволяющие создавать гибкие и легко настраиваемые АИС для широкого класса задач.

2. Разработан набор эффективных по времени алгоритмов для работы с О-деревьями, в т.ч. поиска, добавления, раскрытия и удаления узлов. Предложена новая структура данных «*кэш узлов О-дерева*», позволяющая эффективно управлять оперативной памятью, занимаемой объектами предметной области, при необходимости выгружая их из памяти. Предложен ряд эвристик для алгоритмов поиска и раскрытия узлов О-дерева. Получены оценки их трудоемкости.

3. На основе анализа требований и условий функционирования информационных систем разработаны технология построения и внутренняя среда функционирования АИС в виде модульной объектно-ориентированной надстройки над различными типами реляционных СУБД, упрощающая и ускоряющая разработку широкого класса задач, их сопровождение и дальнейшее развитие.

Теоретическая и практическая ценность:

1. Предложенное понятие *контекста* позволяет ввести новые способы структуризации данных сложных предметных областей. Применение этого понятия позволяет *поэтапно* (в зависимости от степени значимости и актуальности), *неконфликтно* и *неизбыточно* (не дублируя определения объектов и повторно используя программный код) создавать программные приложения для решения различных задач.

2. Предложенная технология построения АИС на основе О-дерева позволяет создавать масштабируемые, расширяемые и переносимые модульные комплексы программ для разных реляционных СУБД.

3. Предложенные эффективные алгоритмы поиска, добавления, раскрытия и удаления узлов О-дерева позволяют существенно увеличить эффективность работы с динамически строящимися древовидными структурами. Предложенная автором структура данных «кэш узлов О-дерева» позволяет эффективно управлять памятью при работе с динамически строящимися деревьями.

4. Разработан комплекс программ для управления планированием, реализацией и для контроля за исполнением муниципального заказа в органах местного самоуправления. Предложенная автором технология построения комплекса позволяет легко и гибко его переконфигурировать под изменяющиеся условия, сохраняя при этом высокую эффективность работы. Разработанный комплекс программ применим не только для управления закупками муниципалитета, но и в иных крупных организациях и предприятиях.

Внедрение результатов работы: разработанный автором комплекс программ «Муниципальный заказ» внедрен в промышленную эксплуатацию в администрации г. Томска (280 рабочих мест) и в администрации Чаинского района Томской области (с. Подгорное, 34 рабочих места).

На защиту автором выносятся:

1. Новая интерпретация понятий *контекста* и *аспекта* в объектных и объектно-реляционных СУБД и схема их применения для проектирования и разработки АИС широкого класса.

2. Модульная архитектура и технология построения АИС, базирующаяся на механизме *О-дерева*.

3. Ряд эффективных алгоритмов и структур для работы с *О-деревьями*.

4. Программный комплекс для автоматизации системы муниципального заказа в Российской Федерации.

Апробация работы и публикации:

Основные положения работы докладывались и были одобрены на следующих научных и специализированных конференциях:

1. Международной научно-практической конференции «Геоинформатика – 2000» (Томск, 2000).

2. Всероссийской научно-практической конференции «Новые технологии и комплексные решения: наука, образование, производство» (Анжеро-Судженск, 2001).

3. Межрегиональной конференции Ассоциации Сибирских и Дальневосточных городов «Повышение эффективности управления бюджетными ресурсами через внедрение системы муниципального заказа» (Омск, 2001).

4. Всероссийской научно-практической конференции «Информационные технологии и математическое моделирование» (Анжеро-Судженск, 2002).

5. Научно-практическом семинаре Ассоциации Сибирских и Дальневосточных городов «Муниципальный заказ как система управления материальными, финансовыми и социальными ресурсами» (Томск, 2004).

По результатам выполненных исследований автором опубликовано 9 печатных работ, в том числе 8 статей. Кроме того, автором разработаны и переданы заказчикам технические и методические материалы по программному комплексу «Муниципальный заказ» общим объемом в 300 стр.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе проведен обзор технологий разработки программного обеспечения и современных архитектур построения автоматизированных информационных и иных программных систем, приводятся примеры использования модульной технологии на примере ряда коммерческих программных продуктов. На основе проведенного анализа автором выдвигаются следующие основные требования к технологии построения автоматизированных информационных систем:

- в технологии должна быть предусмотрена возможность расширения охвата предметной области за счет добавления новых ее фрагментов, причем такое расширение должно быть максимально простым;
- технология должна помогать создавать модульные рабочие места с возможностью замены, добавления и изменения блоков и частей без перекомпиляции всего программного продукта;
- разработанная технологическая основа (среда функционирования) должна базироваться на объектно-ориентированной парадигме;
- среда функционирования должна по возможности быть максимально не зависимой от выбранной СУБД, обеспечивая при этом переносимость;
- интерфейс системы должен быть простым, наглядным и интуитивно понятным.

Кроме того, на использование технологии должны накладываться следующие условия:

- функционирование даже в небольшом коллективе разработчиков;
- ориентированность на сложные предметные области;
- разработка программного обеспечения, ориентированного на сбор, обработку и анализ разнородных данных.

Во второй главе приведено комплексное описание разработанной автором технологии построения модульных АИС с использованием открытой архитектуры и объектно-ориентированной надстройки над реляционной СУБД.

Сложность моделируемого информационными системами реального мира и тенденция к расширению охвата ими предметных областей, построение баз данных для систем, работающих на стыке разных отраслей знания, а самое главное — наличие в них сущностей, представленных с некоторой степенью абстракции с разных точек зрения, приводит к тому, что возникают трудности с целостным, обобщенным, непротиворечивым и одновременно избыточным представлением требуемых для потребителей сведений. Для формализации предметной области автором дается новая интерпретация понятий *контекста* и *аспекта* в применении к объектно-реляционным базам данных, и показываются механизмы его применения к построению гибких и легко настраиваемых АИС.

Однозначно можно сказать, что на любой сложной предметной области можно построить иерархию ее подобластей с нужной степенью детализации. При дроблении всей предметной области неизбежно возникают ситуации, когда одни и те же физические объекты попадают одновременно в более чем одну часть исходной предметной области, ведут себя по-разному и имеют множество связей с другими объектами, фактически играя разные роли.

Работать с набором предметных областей труднее, чем с одной. Поэтому для большей строгости и точности информационного моделирования автором предлагается применить понятие *контекста* в онтологическом смысле, формализующего понятие предметной области. С онтологическими понятиями достаточно трудно работать, т.к. они не формализуемы иначе как на естественном языке, и поэтому иногда могут допускать неоднозначное толкование, для чего введем ниже несколько неформальных понятий и определений.

Точку зрения (набор внутренних структур и поведение — отклики на внешние воздействия с этой точки зрения) на некоторый моделируемый объект реального мира в понимании некоторой предметной области будем называть *аспектом*, а саму предметную область, охватывающую рассматриваемые объекты в некотором аспекте — *контекстом*.

Рассмотрим приложение контекстов к объектной модели. В ней основными структурными элементами являются объекты. Объекты представляются совокупностью своих атрибутов и набором операций. Объекты с одинаковой внутренней структурой и набором операций объединяются в классы.

Пусть $\{c_i\}$ — множество контекстов всей предметной области. Пусть p — понятие или категория, используемая для именованя атрибутов и методов объектов. Введем функцию sem и предикат SEM . Функция $sem(p)$ формально возвращает *семантику* переданного параметра p , а двуместный предикат $SEM(p, c)$ возвращает истину, если параметр p осмыслен в контексте c , и ложь в противном случае. Под *семантикой* будем понимать

формализованный смысл параметра (в общем случае понятия или категории).

Далее будем рассматривать внутреннюю организацию отвлеченного класса объектов O . Пусть $A = \{a_j\}$ — множество атрибутов класса объектов O , а $M = \{m_k\}$ — множество операций с объектами.

Назовем *аспектом* класса объектов O в контексте c подмножество самостоятельных осмысленных атрибутов A_c и операций M_c , такое, что

$$A_c = \{a_j \mid \text{SEM}(a_j, c) = \text{true}\};$$
$$M_c = \{m_i \mid \text{SEM}(m_i, c) = \text{true}\}.$$

Будем обозначать *аспект* класса объектов O в контексте c как O_c :

$$O_c = A_c \cup M_c.$$

Назовем пару атрибутов a_i и a_j *семантически тождественными*, если одинаков их смысл, т.е. семантическая нагрузка в точности совпадает, а значит, выполняется условие: $\text{sem}(a_i) = \text{sem}(a_j)$.

Зададим операцию семантического пересечения множеств следующим образом: в семантическое пересечение двух множеств войдут семантически тождественные элементы обоих множеств. Если среди множества атрибутов не существует семантически тождественных пар, то подмножества атрибутов всех контекстов могут задавать семантическое полное разбиение исходного множества атрибутов. Предметные области, чьи контексты образуют семантические разбиения, будем считать вырожденными случаями, и в дальнейшем их рассматривать не будем. Для таких случаев лучше подходят классические подходы к разработке информационных систем.

Для подготовки основы практической реализации механизма идентификации введем еще несколько понятий.

Системный контекст S — частный (специальный) случай контекста: формальное представление точки зрения системы управления базами данных, обеспечивающей механизмы идентификации объектов, их хранения и доступа к ним.

Системный аспект класса O_S — это унифицированная и стандартизованная (по построению) структура данных для идентификации экземпляров, а также механизмов ее хранения в долговременной памяти. Такая унификация может удобным и логичным способом задаваться в структуре и методах базовых классов объектной модели разрабатываемой технологии. Следует выдвинуть обязательное требование к аспектам классов для обеспечения механизма идентификации:

$$O_A \cap O_B \supseteq O_S,$$

т.е. пересечение аспектов класса должно включать его системный аспект.

Иерархия классов. Корень иерархии классов — TEntity реализует наиболее полный набор абстракций, обеспечивающий именование и идентификацию объектов, общие принципы их хранения и отображения.

Концептуально все множество базовых классов, как видно на рис. 1, делится на три группы: классы для отображения единственного объекта (TEntityInstance), классы множества однородных объектов с возможным условием их отбора (TEntityClass) и классы объектов, которые отображают множества объектов, но ведут себя как экземпляры — контейнерные объекты (TEntityContainer).

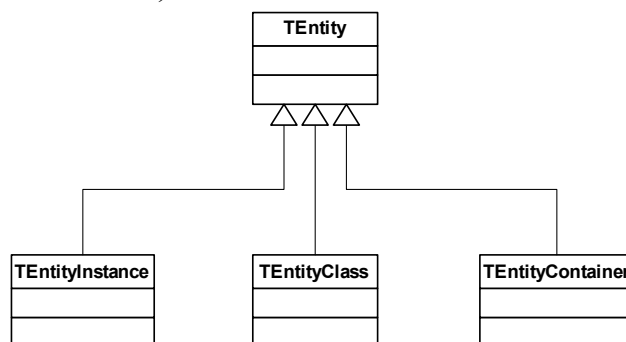


Рис. 1. Базовые классы системы

Классы для реальных предметных областей являются потомками этих трех базовых классов. Надстройка над СУБД из объектной получается объектно-ориентированной, если потомки трех базовых классов сами будут образовывать поддеревья с иерархией наследования.

Для каждого типа сущности реального мира создается еще один класс и соответствующая ему визуальная форма, что полностью соответствует концепции MVC (*model — view — controller*). Идея концепции заключается в том, что сущность разделяется на три составляющие:

- модель (*model*), которая отвечает за бизнес-логику (в нашем случае это не абстрактный класс-потомок TEntity);
- представление (*view*), реализующее отображение информации;
- управляющие элементы (*controller*) — объекты, преобразующие действия с представлением (*view*) в операции с моделью (*model*).

Долговременное хранение объектов. Рассматривая классы системы, следует отметить, что на них возлагаются функции по чтению данных из базы, их интерпретации и обеспечения семантических преобразований (синонимов, омонимов и пр.). Эти механизмы скрываются в реализации внутренних методов, что целиком и полностью возлагается на разработчиков и экспертов.

Однако внедрение объектно-ориентированных принципов в сами реляционные СУБД не совсем удобно для применения. Альтернативой является многоуровневая архитектура, в которой на каждом уровне выполняются свои функции и используются свои эффективные методики их осуществления. Пример трехуровневой архитектуры:

Уровень 1. Реляционная СУБД (эффективное хранение информации).

Уровень 2. Объектно-ориентированная надстройка (структурированное представление информации в виде объектов).

Уровень 3. Пользовательское приложение (выполнение требуемых задач — бизнес-логика и интерфейс).

Здесь важным для нас является принцип *сборки-разборки* объектов при считывании/записи данных из/в СУБД на втором уровне. Прослойка может это обеспечивать сама, либо просто предоставлять соответствующие интерфейсы для классов хранимых объектов.

Чтобы избежать трудностей, которые возникают при разработке универсальной (в смысле подходящей для разных СУБД) схемы хранения данных в реляционной модели автором был предложен ряд принципов для именования единиц хранения информации (таблиц и представлений) и их внутренних полей.

Преобразования (сборка-разборка) из реляционной модели в объектную (объектно-ориентированную) и обратно производится внутри методов базовых классов, доступных в системном контексте.

Дерево объектов и их визуализация. Для задания структуры хранения объектов во внутреннем представлении, а также для обеспечения связи между визуальными и невизуальными компонентами автором предлагается структура дерева объектов (*О-дерева*). Для оптимизации поиска и доступа к объектам, загруженным в оперативную память, автором разработан менеджер объектов, реализующий функции объектного кэша.

В целом совокупность объектов отдельной предметной области визуально представляется в виде графа, где вершинами являются объекты, а ребрами — ссылки. Однако это неудобно для пользователя. Варианты представлений показаны на рис. 2. На рис. 2,*а* показан граф ссылок между объектам, на рис. 2,*б* показан тот же граф, но с представлением в виде дерева — остова графа (жирные дуги); отброшенные дуги изображены пунктиром. На рис. 2,*в* показан вариант представления графа в виде *О-дерева*, где жирными дугами показаны ребра остова, пунктирными — ребра, не вошедшие в остов, а прямоугольниками (показаны частично) обведены кратные вершины, соответствующие одним и тем же объектам.

По последовательности рисунков видно, что фактически задача визуализации графов объектов предметной области формально может быть сведена к построению остова и последующим дублированием узлов, требующимся для включения в результирующее дерево ребер, выброшенных при скелетизации (построении остова). Однако на практике все выглядит несколько иначе: такой формальный подход не учитывает специфики межобъектных ссылок, их семантики и степени значимости. Поэтому автором предлагается другой конструктивный подход — представление объектов предметной области в виде динамически строящегося *О-дерева*.

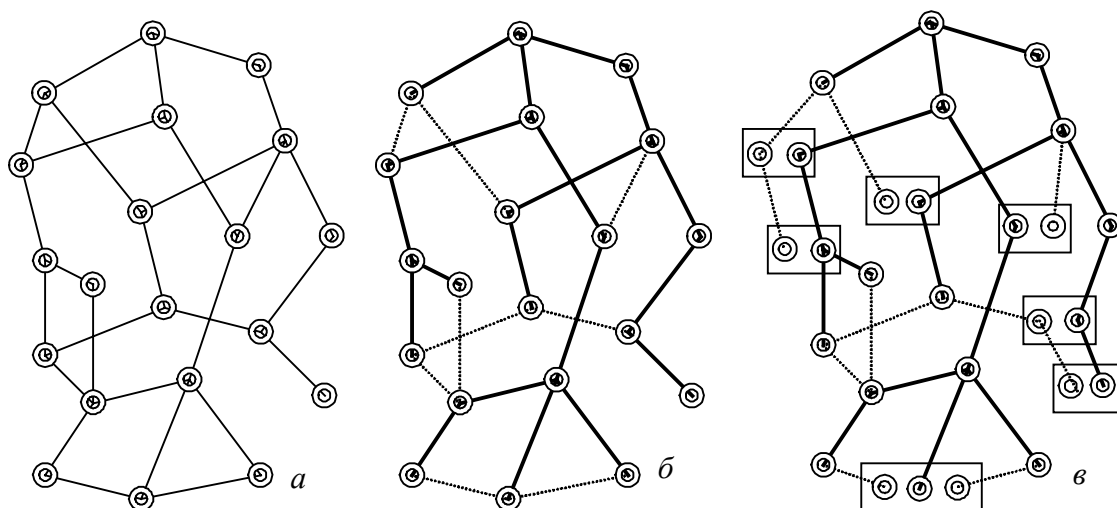


Рис. 2. Варианты представления графовых структур

O-дерево представляет собой сильноветвящееся дерево в классическом понимании, но интегрированное со средствами графического представления, и ориентированное на пользовательский интерфейс. С каждым его узлом однозначно связан конкретный экземпляр объекта определенного класса. Приведем основные принципы работы с *O-деревом*:

1. Дерево объектов строится по ходу работы с ним (по мере необходимости при раскрытии узлов).

2. Правила раскрытия узлов и построения дерева разрабатываются в соответствии со структурой предметной области и хранятся в служебной таблице. В дереве могут отображаться только объекты зарегистрированных классов.

3. При активизации узла дерева производится отображение связанного с ним объекта в панели для форм. Для этого привлекается ассоциированная с классом объекта активного узла форма (компонент view концепции MVC).

4. При добавлении или удалении объектов в базу данных они либо появляются в дереве, либо из него удаляются. Эти операции с деревом производятся только в случае их потенциальной видимости, т.е. если объекты находятся в памяти и потенциально достижимы без повторного доступа к базе данных.

5. Связи между узлами обязательно учитывают контекст родительского объекта.

Узлы дерева могут находиться в трех состояниях:

– *неопределенно свернутом* — узел не раскрывался, наличие и количество потомков узла неизвестно (не запрашивалось);

– *определенно свернутом* — узел раскрывался, но был свернут, количество и наличие потомков известно;

– *развернутом* — узел раскрыт, потомки при их наличии отображаются в дереве и доступны для выполнения различных операций.

В ходе построения дерева используются четыре типа правил (способов навигации). Эти правила задают порядок и регламентируют механизмы (в том числе и параметризованные) организации связи между объектами способа навигации. Перечислим возможные типы связей между узлами:

– *Раскрытие корня* — правило, которое выполняется однократно, для организации основы — корня дерева при начале его построения.

– *Абстрактная* — связь, которая структурирует начальные уровни дерева, задавая основные его ветви обычно в самом начале ветвления. В основном такого рода связи задаются между корнем и абстрактными классами, либо между абстрактными классами и классами множествами. В этом случае связь может быть параметризована. Параметры задаются в правилах служебной таблицы.

– *Связь «множество – экземпляр»* — связь определяет отношение «целое – часть», либо «есть некоторый». Связь не имеет параметров.

– *Связь «по внешнему ключу»* — связь между экземпляром и множеством объектов, построенная на отношении между ними типа «один ко многим». Связь осуществляется по принципу равенства значений внешнего ключа в дочернем множестве объектов значению первичного ключа в экземпляре родительского объекта. Имя внешнего ключа задается как параметр связи.

Правила, определяющие связи между объектами, задаются как набор кортежей вида *<контекст, предок, потомок, тип связи, параметры>*.

Раскроем значения компонентов этого кортежа:

– *контекст* — имя контекста, в котором данная связь актуальна;

– *предок* — идентификатор класса из таблицы зарегистрированных в системе классов, являющийся предком в правиле построения дерева;

– *потомок* — идентификатор класса из таблицы зарегистрированных в системе классов, являющийся потомком в правиле построения;

– *тип связи* — идентификатор типа связи, соответствующего семантике отношений между предком и потомком (из вышеперечисленного списка);

– *параметры* — строковое значение определенного формата, определяющее дополнительные ограничения и условия на связь.

Для отображения объектов в виде дерева и предоставления места для визуализации форм объектов используется навигатор объектов (рис. 3). Внешне навигатор представляет собой две панели: в левой части — *О-дерево* (визуальная его часть), в правой — панель для форм объектов.

При активации узла в дереве в правую панель при помощи специальных внутренних механизмов автоматически встраивается форма, соответствующая объекту, ассоциированному с выбранным узлом, и управление передается самому объекту для выполнения специальных процедур отображения.

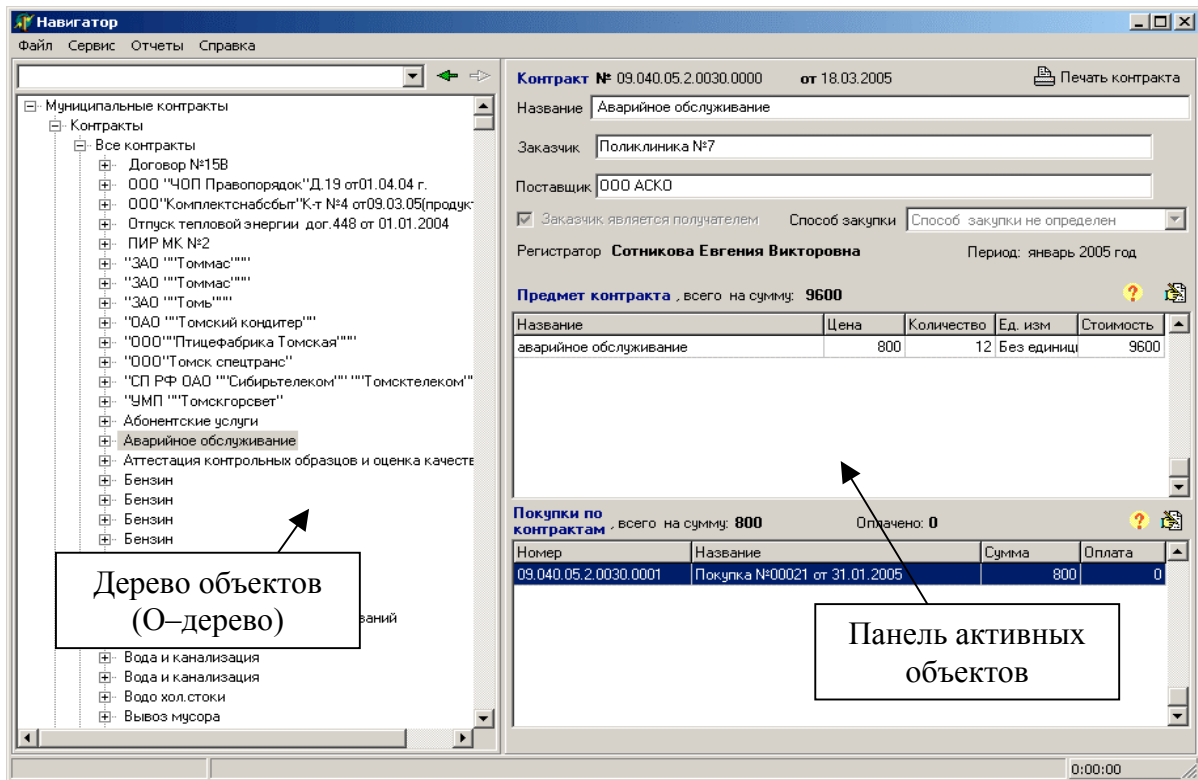


Рис. 3. Внешний вид навигатора объектов со схемой частей

Алгоритмы. Построение полного графа для предметной области, заполненной большим количеством объектов, является очень трудоемкой задачей, но для целей визуализации этого не требуется. Само *O-дерево* реализуется с помощью стандартного компонента TListView Borland Delphi и набора вспомогательных структур. К узлам дерева через специальное поле привязаны объекты в виде экземпляров классов потомков TEntity.

В общем случае предполагается работа с сильно ветвящимся частично заданным и достраиваемым в процессе манипуляций деревом. «Частично заданным» его можно назвать потому, что на каждый момент времени работы пользователя оно будет иметь разную конфигурацию и вид в зависимости от потребностей оператора по работе с информацией.

Типичные операции по работе с таким деревом: раскрытие узла (добавление объектов – прямых потомков), добавление объекта с его последующей визуализацией, активизация объекта, поиск объекта, удаление объекта (в общем случае нескольких узлов).

Для эффективного выполнения этих операций автором был разработан ряд алгоритмов и структур данных. Рассмотрим их.

Помимо самого дерева используется ряд следующих вспомогательных структур:

- список классов объектов системы;
- список правил построения дерева вида $T_i \xrightarrow{P} T_j$, где T_i — класс-предок, T_j — класс-потомок, а P — тип правила (способ порождения);

– индексы для быстрого поиска объектов по идентификатору и классу.

Для работы с *O-деревьями* автором были разработаны следующие алгоритмы (в скобках указана их трудоемкость, где r – число правил, p – общее число объектов в текущем классе, k – коэффициент кратности вершин, n – общее число объектов в БД всех классов, m – обрабатываемое (проверяемое или удаляемое) число узлов):

- **ПРОВЕРИТЬ_И_ОСВОБОДИТЬ** ($T(m) = O(m)$).
- **ВЫДЕЛИТЬ_ДЛЯ_УДАЛЕНИЯ** ($T(m) = O(m)$).
- **РАСКРЫТЬ_УЗЕЛ** ($T(r) = O(r)$).
- **НАЙТИ_ОБЪЕКТ** ($T(p, k) = O(k + \log p)$).
- **УДАЛИТЬ_ОБЪЕКТ** ($T(p, k) = O(k + \log p)$).
- **ВИЗУАЛИЗИРОВАТЬ_ОБЪЕКТ** ($T(k, n) = O(kn)$).

Первые два алгоритма носят вспомогательный характер и используются для работы с кэшем объектов. В нем имеется список последних обращений к объектам, наиболее часто используемые узлы перемещаются к началу списка. По мере его разрастания до критической величины M и при необходимости добавления новых объектов, находящиеся в конце списка сохраняются в базе и удаляются из оперативной памяти и из объектного кэша.

Для алгоритма **ВЫДЕЛИТЬ_ДЛЯ_УДАЛЕНИЯ** автором предложен ряд эвристики, касающиеся вариантов удаления объектов.

Далее рассмотрим добавление объекта с его последующей визуализацией. Фактически необходимо после добавления объекта в базу обеспечить его видимость, что представляется в общем случае нетривиальной задачей. Для этого необходимо раскрыть некоторые узлы для достижения видимости требуемого объекта, т.е. найти последовательность применения порождающих правил.

В общем случае трудоемкость алгоритма в наихудшем случае составляет $O(kn)$, где n — количество объектов в базе, k — коэффициент оценки кратности повторения объектов в дереве ($k \geq 1$). Для достаточно большого n это неоправданно трудоемко. Поэтому сделаем попытку проанализировать алгоритм и попытаемся применить некоторые эвристики, которые могли бы уменьшить трудоемкость визуализации объекта. Очевидно, что оптимизируемым или уменьшаемым параметром здесь служит количество вновь создаваемых узлов. Алгоритм с полным обходом дерева имел бы экспоненциальную трудоемкость.

Автором в качестве возможных рассматривались варианты повышения эффективности работы этого алгоритма. В основном это вариации, основанные на моделировании процесса построения дерева осведомленным пользователем или экспертом. Общими для этих подходов является вы-

борка списка классов узлов-листьев, построение возможных цепочек (потенциально ограниченных по длине) порождающих правил для *O-дерева* и их применение:

1. Выбираются все применимые цепочки в порядке их получения из списка порождающих правил.

2. Цепочки сортируются в порядке возрастания их длины.

3. Цепочки сортируются, исходя из оценок количества объектов, образующихся при их применении.

4. Цепочки выбираются на основе детального анализа ссылок объектов в начале и в конце цепочки. В этом случае предпочтение отдается параметризованным цепочкам, которые сокращают размер общего множества объектов.

В случае применения эвристик выигрыш по трудоемкости достигается за счет того, что сильно сокращается выбор возможных узлов для достижения требуемого объекта. Оценки трудоемкостей для первых трех вариантов будут совпадать и составят $O(mr)$, где r — максимальная длина цепочек, а m — количество объектов в самом большом из используемых классов, при этом m много меньше общего количества объектов в базе данных.

Рассмотрим алгоритм поиска объекта. При поиске объектов важную роль играет принадлежность классу. Индексы кэша объектов представлены в виде сбалансированных деревьев, трудоемкость поиска с их использованием будет $O(\log m)$, где m — мощность класса объектов. Трудоемкость поиска всех вхождений объекта в дерево составляет $O(d + \log m)$, где d — максимальное количество вхождений объектов.

Использование предложенной технологии для разработки целой серии автоматизированных рабочих мест требует выполнения определенных стадий:

1) конкретизация структуры предметной области (выделение контекстов);

2) формализация бизнес-процессов и участвующих в них объектов;

3) разработка схемы данных для хранения объектов в СУБД;

4) кодирование классов объектов и их объединение в модули АИС;

5) построение правил навигации, определяющих структуру *O-дерева*;

6) разработка механизмов проведения метаопераций (сервисов), обеспечения настройки отдельных компонентов и создание справочной системы.

Таким образом, весь процесс разработки завершается разработкой модулей создаваемой системы с их последующим связыванием при помощи внутренних средств настройки технологической основы. Общая схема итогового приложения показана на рис. 4.

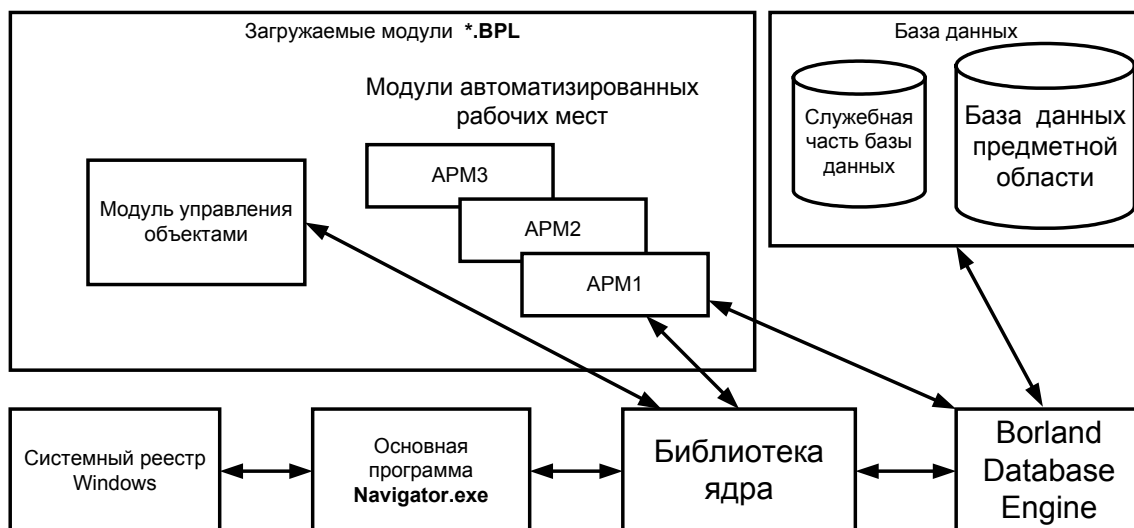


Рис. 4. Общая архитектура системы частей

В третьей главе дается описание принципов организации закупок для нужд органов власти, анализируется мировой опыт автоматизации бюджетных закупок и формулируются требования, условия и основные принципы разработки современной автоматизированной информационной системы автоматизации процессов муниципального заказа в Российской Федерации.

Система муниципального или государственного заказа позволяет существенно экономить финансы за счет более эффективных средств определения и удовлетворения потребностей органов власти и подведомственных учреждений в продукции (товарах, работах и услугах). Однако не сформировавшаяся до конца федеральная законодательная и нормативная база в Российской Федерации в части организации муниципального заказа в настоящее время не обеспечивает устойчивого и надежного фундамента для функционирования системы закупок товаров, работ и услуг для муниципальных нужд.

Проведенный автором анализ современного состояния этого направления показал, что сложность процессов и значительные объемы данных системы муниципального заказа, а также острая необходимость оперативной их обработки заставляют внедрять автоматизированные системы в эту область хозяйственной деятельности. Имеющиеся на рынке программные продукты по автоматизации системы муниципального заказа обладают рядом недостатков, делающих их ограниченно применимыми или же не применимыми вовсе.

Для создаваемого программного комплекса автором были поставлены три главные задачи, решающие вопросы эффективного удовлетворения потребностей муниципалитета в товарах, работах и услугах:

– планирование и формирование сводных объемов годового потребления продукции по видам, выраженных в натуральных показателях;

- размещение этих объемов на товарных рынках региона способами, регламентированными и приносящими экономический эффект (экономия средств и развитие местной сферы производства продукции);

- контроль за фактическим потреблением, приобретением и оплатой товаров, работ и услуг.

Рассмотрение вариантов решения этих задач свелось к необходимости разработки комплекса программ, построенных по общей идеологии с применением малотребовательной к ресурсам технологии, использующей модульную архитектуру и разнородные системы управления базами данных. К самой технологии автором были выдвинуты дополнительные требования на открытость архитектуры, простоту и единообразие пользовательского интерфейса.

Изменчивость законодательной базы еще не совсем сформировавшейся предметной области — муниципального заказа — добавила необходимость в простоте и легкости конфигурации внутренней структуры автоматизированных рабочих мест и при необходимости такого же несложного изменения логики их работы. Для разработки очень удачно подходит разработанная автором технология построения АИС, полностью удовлетворяющая вышеизложенным требованиям.

В четвертой главе приведены основные проектные решения по автоматизации системы муниципальных заказов на основе предложенной автором технологии, приведена детализация предметной области, диаграммы основных бизнес-процессов и потоков данных, приводится иерархия созданной автором объектной модели.

Разработанный автором комплекс программ функционирует под управлением операционных систем Windows 98/Me/2000/XP. Для хранения данных может быть использовано множество современных коммерческих реляционных СУБД, в т.ч. Oracle, Sybase и Paradox, а также другие системы, поддерживаемые протоколом BDE. Весь комплекс в настоящее время состоит из трех частей:

- АИС «Заявка в муниципальный заказ»;
- АИС «План муниципального заказа»;
- АИС «Реестр муниципальных контрактов».

Указанные части программного комплекса обеспечивают почти полный цикл работы. Первоначально на рабочих местах бюджетополучателей формируются заявки на продукцию в пределах бюджетных ассигнований. Заявки на продукцию фиксируются в натуральных величинах с указанием объемов, средней цены и точного наименования с группировкой по предлагаемому справочнику товаров, работ и услуг при помощи АИС «Заявка в муниципальный заказ».

Далее информация посредством механизмов репликации данных сведения по заявке передаются в АИС «План муниципального заказа», где они проходят обработку в части сопоставления товарных наименований. В

итоге получается *сводный план муниципального заказа*. На его основе объявляются конкурсы на позиции, чья общая стоимость превышает определенный нормативно-правовой базой размер. По итогам конкурсов определяются победители.

Все покупки проводятся в установленном законом порядке способами — либо с заключением муниципального контракта, либо посредством покупки без контракта. В обоих случаях производится их регистрация в АИС «Реестр муниципальных контрактов». С определенной периодичностью данные поступают сначала в ведомственный реестр, а затем и в городской. В последнем средстве также производится работа по контролю и анализу фактического исполнения, по соответствию потребленных объемов плановым величинам и пр.

Цикл функционирования всего комплекса программ на этом этапе замыкается — фактически потребленные за год товары, работы и услуги могут составить основу для разработки плановых потребностей на будущий бюджетный год.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Проведено исследование и разработаны структуры данных, базовые алгоритмы и технологии, необходимых для создания АИС для широкого класса информационных систем.

2. Предложена новая интерпретация понятий *аспекта* и *контекста* как механизма структуризации данных для объектных и объектно-реляционных СУБД. Предложен механизм *О-дерева* и формализм правил его построения с использованием контекстов.

3. Предложен ряд алгоритмов для работы с динамическими деревьями. Получены оценки их трудоемкости.

4. Проанализирована и формализована предметная область, касающаяся планирования и управления потреблением товаров, работ и услуг в органах местного самоуправления Российской Федерации.

5. Разработан и внедрен в промышленную эксплуатацию крупный программный комплекс «Муниципальный заказ», позволяющий полнее и быстрее отслеживать информационные потоки, существующие в системе муниципального заказа, что приводит к повышению ее управляемости и эффективности, а также к существенной экономии бюджетных средств.

ОПУБЛИКОВАННЫЕ РАБОТЫ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Макунин А.А., Шилов В.В. Применение модульной технологии разработки автоматизированных рабочих мест в ГИС-приложениях // Геоинформатика–2000. Труды межд. научно-практ. конф. – Томск: Изд-во Том. ун-та, 2000. – С. 84–87.

2. Макунин А.А. Особенности использования иерархических классификаторов в реляционных СУБД // Новые технологии и комплексные решения: наука, образование, производство. Материалы Всероссийской научно-практ. конф. (19 октября 2001 г., Анжеро-Судженск). Часть VI (Информатика). – Кемерово: КемГУ, 2001. – С. 48–50.

3. Макунин А.А. Элементы объектно-ориентированной системы управления базами данных // Вестник Томского ун-та, 2002, № 275. – С. 161–163.

4. Макунин А.А., Сотникова Е.В. Автоматизация регистрации и учета исполнения муниципальных контрактов в Томске // Вестник Томского ун-та, 2002, № 275. – С. 164–166.

5. Сотникова Е.В., Макунин А.А., Сафонов Н.Л. Об автоматизации процессов исполнения и формирования муниципального заказа города Томска // Материалы Межрегиональной конференции АСДГ «Повышение эффективности управления бюджетными ресурсами через внедрение системы муниципального заказа» (26-27 ноября 2001 г.). – Омск, 2001. – С. 43–47.

6. Сотникова Е.В., Макунин А.А. Автоматизация процессов муниципального заказа города Томска // Материалы Всероссийской научно-практической конференции «Информационные технологии и математическое моделирование» (10 ноября 2002 г., г. Анжеро-Судженск). – Томск: «Твердыня», 2002. – С. 304–305.

7. Сотникова Е.В., Макунин А.А. Комплекс информационных систем «Муниципальный заказ» // Материалы Всероссийской научно-практической конференции «Информационные технологии и математическое моделирование» (10 ноября 2002 г., г. Анжеро-Судженск). – Томск: «Твердыня», 2002. – С. 306–307.

8. Макунин А.А., Сотникова Е.В. Комплекс программ по управлению бюджетными и материальными ресурсами муниципального образования // Теоретическая и прикладная информатика / Под ред. А.Ф. Терпугова. – Томск: Изд-во Том. ун-та, 2004. – Вып.1. – С. 43–49.

9. Сотникова Е.В., Макунин А.А. Практика ведения информационной базы системы муниципального заказа в г. Томске // Общественный контроль за конкурсным распределением государственных и муниципальных заказов: Сборник статей. – Томск: Изд-во Том. ун-та, 2004. – С. 33–39.