На правах рукописи

Бутенко Игорь Всеволодович

Разработка моделей и методов построения и автоматизированного наполнения системы метаданных

Специальность 05.13.11 –
Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Автореферат
диссертации на соискание ученой степени кандидата технических наук

Санкт – Петербург 2011

Работа выполнена в федеральном государственном бюджетном образовательном учреждении высшего профессионального образования «Санкт-Петербургский государственный политехнический университет».

Научный руководитель – доктор технических наук, профессор
Устинов Сергей Михайлович

Официальные оппоненты – доктор технических наук, профессор
Гаврилова Татьяна Альбертовна,
– кандидат технических наук
Абдрахманов Руслан Леонидович

Ведущая организация – Федеральное государственное унитарное предприятие «Научно-исследовательское объединение «Импульс»

Защита состоится « » декабря 2011 г. в 16 часов на заседании диссертационного совета Д 212.229.18 в федеральном государственном бюджетном образовательном учреждении высшего профессионального образования “Санкт-Петербургский государственный политехнический университет” по адресу: 195251, Санкт-Петербург, Политехническая ул., д.29, 9 уч. корп., ауд. 325.

С диссертацией можно ознакомиться в Фундаментальной библиотеке университета.

Автореферат разослан « » ноября 2011 г.

Ученый секретарь
диссертационного совета Васильев А.Е.

Общая характеристика работы

Актуальность проблемы

В современных информационных системах накоплено большое количество данных, и извлечение нужной информации из таких систем часто связано с недопустимыми затратами времени и средств. Для того, чтобы получить информацию, необходимо знать какие именно данные уже есть, где они находятся и как могут быть получены.

В связи с этим при создании новых информационных систем разработчики используют аппарат, базирующийся на метаданных (МД). Он предоставляет возможности описания и манипулирования метаданными в рамках либо общей модели CWM (Common Warehouse Metamodel), сформированной группой OMG (Object Management Group), либо модели конкретной метасистемы (МС). Этим вопросам посвящены многочисленные работы ученых, как в нашей стране, так и за рубежом (Д. Марко, А. Танненбаум, Р. Кимбалл, Б. Инмон), а также такие программные продукты как Oracle Data Integrator, IBM Information Server, SAP BusinessObjects Metadata Manager, CA ERwin Saphir Option, SAS® Metadata Server.

Однако, в настоящее время существует большое количество информационных систем, в которых отсутствуют МД либо их набор не является полным. Такие системы или развиваются с большим трудом (очень большие вложения при минимальных результатах), или не развиваются вовсе, поскольку отсутствует информация не только о том, что, где и как хранится, но и как используется. При этом многочисленные подходы формирования метаданных, применяемые при построении новых информационных систем, не могут быть впрямую использованы для наполнения МД информацией из уже существующих систем.

Выходом из создавшегося положения является построение МС на основе уже используемых информационных систем. Это, в свою очередь, требует создания сопутствующих методов, алгоритмов и программного обеспечения. В свете изложенного, разработка методики построения репозитория метаданных для существующей информационной системы является весьма актуальной.

Цель работы

Целью работы является разработка методики, алгоритмического и программного обеспечения поддержки этапов проектирования информационной аналитической системы на базе аппарата метаданных. Это позволит для уже созданных систем обеспечить их развитие и модернизацию.

Для достижения поставленной цели в работе ставились и решались следующие задачи:

1) Разработка методики построения МС на основе существующих информационных систем.
2) Построение моделей описания МД и классификаторов.
3) Создание алгоритмов наполнения МС данными и классификации этих данных.

4) Реализация на практике МС для решения реальных прикладных задач.

Методы исследования

В диссертации используются методы теории множеств, реляционной алгебры, реляционного исчисления, проектирования и нормализации баз данных (БД), а также методы системного структурного анализа.

Научная новизна

Научную новизну работы, в первую очередь, составляет методика построения МС на основе существующих SQL-ориентированных информационных систем. В рамках этой методики были предложены следующие модели и алгоритмы:

1) Формальная модель описания МД, согласованная с общей моделью CWM, и поддерживающая связи пользовательских понятий и объектов базы данных.
2) Формальная модель описания классификатора на базе предоженной модели МД.
3) Алгоритм наполнения МС данными из внешних источников на основе предложенной модели.
4) Алгоритм построения дерева классификаторов на основе их моделей и данных.

Достоверность результатов

Достоверность полученных результатов обеспечивается использованием утверждений, доказанных в реляционной алгебре, корректным доказательством и непротиворечивостью предлагаемых утверждений, а также подтверждается опытом эксплуатации ПО, разработанного с использованием предложенных методик.

Практическая ценность и внедрение результатов работы

1) Разработанная методика построения МС, модели описания МД и классификаторов носят формализованный характер и могут быть использованы при построении различных баз данных МС.
2) Предложенные алгоритмы могут быть использованы при построении модуля загрузки и классификации данных в различных прикладных задачах.
3) Реализованная МС может эффективно применяться в организациях, часто имеющих дело с изменениями аналитической отчетности, либо автоматизирующих различные бизнес-процессы..

В частности, построенная МС была включена в коммерческие программные продукты:

- «СКАУТ-Навигатор» в качестве средства поддержки формирования различных отчетов,
- «СКАУТ-УКОИ» в качестве средства построения многомерных OLAP-отчетов,
- «СКАУТ-Сервисный центр» в качестве элемента модуля генератора отчетов,
- другие продукты, производимые ООО «Деловые консультации, СПб».

Результаты диссертационной работы используются в рамках учебного курса «Базы данных» кафедры «Информационные и Управляющие Системы» СПбГПУ.

Самостоятельную практическую ценность представляют отдельные фрагменты созданной МС, в частности:

- база данных МС,
- модуль выгрузки МД из исходных систем.

Эффективность предложенных в диссертации разработок подтверждена актами соответствующих предприятий о внедрении и научно-технической значимости работы.

Положения, выносимые на защиту

На защиту выносятся следующие научные и практические результаты:

1) Методика построения МС на основе существующих SQL-ориентированных информационных систем.
2) Формализованные модели описания МД и классификаторов МД.
3) Алгоритм наполнения МС данными из внешних источников на основе предложенной модели.
4) Расширяемая база данных МС, основанная на описанных моделях.
5) Модуль выгрузки МД из исходных информационных систем.

Апробация работы и публикации

Основные результаты диссертационной работы обсуждались в рамках научно-практических конференций «Технологии Microsoft в теории и практике программирования» (СПб, 2004, 2005, 2006, 2009, 2010 гг.) и «Фундаментальные исследования в технических университетах» (СПб, 2005 г.), на семинарах «Неделя науки» СПбГПУ (XXXI – 2003 г., XXXII – 2004 г.), а также были опубликованы в сборниках «Труды молодых ученых» и «Научно-технические ведомости СПбГПУ».

По материалам диссертации опубликовано 15 печатных работ, в том числе 3 из Перечня ВАК.

Структура и объем работы

Диссертация содержит 144 страниц основного текста, 35 рисунков, 4 таблицы и состоит из введения, четырех глав, заключения, списка литературы и одного приложения.

Содержание работы

Во введении обосновывается актуальность работы, формулируются цель, задачи, объект и предмет диссертационного исследования, показана научная новизна и практическая ценность полученных результатов. Отражается структура диссертации.

Первая глава посвящена изучению различных подходов к построению метаданных в информационных аналитических системах (ИАС). Детально рассматривается процесс построения ИАС с использованием метаданных (МД) и репозиторий, как средство хранения метаданных. Проводится обзор существующих моделей ИАС: хранилищ данных, федеративных систем, основным источником данных для которых является информация из внешних источников. Именно поэтому одним из важнейших этапов построения ИАС является этап интеграции и загрузки данных. В работе рассматриваются три основных метода интеграции данных: консолидация, федерализация и распространение. Какой бы способ интеграции не был выбран, необходимо решать вопросы переноса данных. Обычно перенос состоит из извлечения, преобразования и загрузки данных. Отмечается важный момент: каждое из этих решений требует четкого понимания структуры исходной системы. Таким образом, для построения ИАС требуется механизм просмотра объектов в исходной системе, поскольку аналитику необходимо работать не с самими данными, а с их описанием. Для него предпочтительней работать в понятной для себя предметной области и не вникать в технические детали БД. А это, в свою очередь, означает, что необходим инструментарий работы с метаданными.

В работе производится анализ основных свойств и требований к метаданным с точки зрения различных классов пользователей и различных типов информационных систем, проведен обзор современных средств работы с метаданными. Отмечается, что в настоящее время на рынке представлено большое количество средств для работы с метаданными, которые должны взаимодействовать между собой и поэтому необходима стандартизация метаданных.

В качестве стандарта для обмена метаданными ИАС принят Common Warehouse Metamodel основанный на стандартах XML, XMI и UML. Это обеспечивает мощную объектную модель с набором API, форматов обмена и услуг, которые охватывают весь спектр метаданных. Для возможности обмена метаданных с внешними источниками одним из требований к потенциальной метасистеме должна быть возможность поддержки стандарта CWM.

Рассматриваются существующие метасистемы: IBM Information Server, SAS® Metadata Server, CA ERwin Saphir Option, Oracle Data Integrator, SAP BusinessObjects Metadata Manager. Вводятся критерии сравнения этих метасистем. В качестве основного критерия была выбрана возможность «обратного проектирования», т.е. возможность строить по схеме БД ее модель. Даже те системы, которые формально поддерживают обратное проектирование, на деле предоставляют только пользовательский интерфейс и возможность частичной загрузки начальных данных. Основной этап непосредственно ассоциирования данных той или иной предметной области с конкретными метаданными ложится на плечи пользователя. Серьезный недостаток с точки зрения практики – цена (все рассматриваемые системы дороги). Таким образом, на основе анализа делается вывод о необходимости разработать свою собственную систему по работе с МД. При

этом данная система может быть частично интегрирована в уже существующие программные средства для расширения их функциональности и уменьшения стоимости новой системы. Но это не должно быть ограничением системы, т.е. не должно быть ориентированности на конкретные продукты. Необходим универсальный подход для решения задачи построения метасистемы по уже существующим на предприятии данным. Она должна обеспечивать формирование общего информационного пространства и дальнейшее развитие системы. Это, в свою очередь, требует разработки сопутствующих методов, алгоритмов и программного обеспечения. Предлагаемая МС должна решать следующие задачи:

- Построение единого хранилища и описание всех объектов существующей системы.
- Возможность классификации и поиска необходимой информации.
- Обеспечение отслеживания версий объектов БД на сервере.
- Повышение уровня языка доступа к данным – возможность строить запросы из МС.
- Создание базиса для построения федеративных систем и ХД.

Во второй главе предлагается строить репозиторий МД, в качестве исходных данных для которого используются МД, загружаемые из уже существующих информационных систем. Реализация данного подхода может быть осуществлена в результате выполнения следующих этапов:

1) **Разработка структуры БД МС**. Результатом данного этапа является информационная структура МС, для создания которой необходимо построить модель объектов, хранящихся в системе.
2) **Разработка модели классификатора**, как элемента структуры БД МС.
3) **Загрузка метаданных**. Необходимо разработать алгоритм загрузки данных в структуры соответствующих моделей, созданных на этапе 1. Для этого также надо описать модель загрузки объектов в МС.
4) **Построение дерева классификатора**. Для качественного функционирования репозитория МД необходима подсистема настройки классификаторов. Для этого требуется построить модель, которая будет опираться на модель данных из п.2.
5) **Дальнейшее использование (орг. мероприятия).** Кроме задачи построения системы немаловажную роль играет и поддержка работоспособности системы. Для эффективного функционирования МС формируются требования к пользователям систем, взаимодействующих с МС.
6) **Выбор средств реализации МС.** Для реализации МС необходимо выбрать следующие программные средства: СУБД, средства загрузки данных, средства реализации клиентского приложения.

Во второй главе подробно рассматриваются модели описания объектов БД и классификаторов, а также проектирование структуры БД МС по предложенным моделям.

В рамках ИАС разработчики оперируют следующим набором сущностей БД, которые и являются объектами МС: таблицы, хранимые процедуры (ХП), колонки таблиц, параметры процедур, типы данных и их описания.

В работе предлагается следующая модель описания этих объектов:

O=<n,t,{O},P,C>, где n – имя объекта (в терминах БД); t – тип объекта (таблица, ХП, колонка, параметр ХП); P – параметры объекта, необходимые (и, может быть, достаточные) для построения объекта O на базе; C – описания объекта.

Описание объекта может включать описание других объекты (таблица описывается через колонки), что позволяет конструировать описание объектов разной сложности. Для того, чтобы установить, какие элементы векторов *P* и *C* отвечают за какое свойство, введем векторы типов параметров *П* и *K*.

P – четко заданная последовательность параметров.

$P=\{p_i\}$, где p_i – значение параметра i *из вектора типов параметров $П=\{n_i\}$.*

C –четко заданная последовательность описаний.

$C=\{c_i\}$, где c_i – значение параметра i *из вектора типов описаний $K=\{k_i\}$.*

Говоря о параметрах объектов отметим, что количество элементов P может изменяться в зависимости от прикладной области, текущей задачи, требований к системе, среды окружения. Но при этом можно расширять это множество без больших затрат, поскольку все данные, которые хранятся в P, четко определимы. Это данные, которые можно выбрать из БД или среды окружения. Все элементы множества P определяются автоматически при загрузке объекта в МС. Набор элементов матриц П и P однозначно определяется типом объекта.

На основе P строятся предустановленные классификаторы. При необходимости расширить набор классификатора требуется добавить новый параметр в описание МС, загрузить данные в этот параметр и на его основе построить классификатор.

На уровне реализации задача сводится к добавлению еще одного столбца в таблицы описаний объектов в МС и программированию наполнения этого столбца. В рамках предложенной модели объектов никаких изменений в исходную систему вноситься не будет.

По мере развития МС увеличивается количество описанных параметров и для новых проектов минимизируется вероятность расширения этого множества. Это обеспечивается благодаря тому, что для каждого проекта $П_{i+1} \supseteq П_i$, где i – порядковый номер проекта.

Таким образом, задав вектор P для таблицы и колонок таблицы, мы можем:

- однозначно идентифицировать каждую такую таблицу и колонку в ней,
- построить на любой базе таблицу аналогичной структуры,

- строить запросы к исходной таблице, основываясь только на ее описании.

Значения из вектора С – это всевозможные описания объекта О на языке пользователя. Таким образом, С – это основа МС, именно он обеспечивает отображение описания системы с языка БД на язык пользователя. Элементы С должны полностью описать объект О. Количество элементов С зависит от многих факторов: предметная область, полнота описания уже существующей БД, требования к системе и др. Но при этом, по мере развития МС и появления новых элементов С, которые не будут использоваться в других проектах, будем предоставлять возможность пользователю при необходимости их настраивать. Таким образом, будет поддерживаться возможность развивать исходные системы в плане расширения описаний их объектов. Так же, как и для множества П, получаем, что для каждого проекта $K_{i+1} \supseteq K_i$, где i – порядковый номер проекта.

Определим классификатор метасистемы следующим образом:

$$\Omega = \{\omega\} \to O_\omega \subseteq O$$

Здесь ω – ограничения, определяемые на множестве всех объектов O и позволяющие однозначно идентифицировать подмножество O_ω.

Поскольку любой объект множества О может быть записан в виде *O=<n,t,{O},P,C>*, то получаем, что описание ограничений должно включать пять подмножеств.

$$\omega = \langle \omega_n, \omega_t, \{\omega\}, \omega_P, \omega_C \rangle$$

Из этого описания может быть исключено ω_n путем объединения с ω_P, поскольку наименование объекта является подмножеством *P*.

Рассмотрим рекурсивное описание объектов и ограничений. Для объектов рекурсия получается из того, что описывается объект через другие объекты МС. При этом важно, что эти объекты всегда другого типа. Описание параметра – в описании типа параметра.

Таким образом, для задания ограничения на объект достаточно задать тип объекта, на который накладывается ограничение, и само ограничение. В нашей системе все описания объектов хранятся в двух множествах: *P* и *C*. Поэтому описание ω сводится к следующему:

$$\omega = \{\omega_t, \omega_P, \omega_C\}$$

В итоге получаем общую формулу описания классификатора:

$$\Omega = \{\omega_t, \omega_P, \omega_C\}$$

Первым шагом построения МС является анализ прикладных задач исходной системы. В результате можно получить задачи самой МС и сформировать набор характеристик объектов, который будет храниться в репозитории. Предложенная модель говорит только о том, как представляются объекты системы, но не говорит о том, какие именно данные мы будем хранить. Есть жесткие ограничения модели на хранимые данные. Так, описания объекта, содержащиеся в *P*,

должны быть предметнонезависимыми и включать описания самого объекта как физического файла на диске и объекта БД, но без привязки непосредственно к предметной области. Таким образом, получается, что свойства *P* должны быть заполнены всегда, причем заполняться они должны в автоматическом режиме. Свойства, описываемые через *C*, наоборот – полностью зависимы от предметной области. Здесь основная нагрузка ложится на пользователя. Именно он должен сформулировать, откуда получать то или иное значение. Если же данных пока нет, то пользователю придется их вводить.

Одной из важных особенностей разрабатываемой МС является ее расширяемость. При этом первые реализации МС выявили набор атрибутов, которые в большинстве своем покрывают основные необходимые свойства элементов системы.

Поскольку МС разрабатывалась на основе данных реляционных СУБД и предполагалось, что в качестве ядра для МС будет выступать также реляционная СУБД, то проектирование БД МС производилось с использованием аппарата реляционной алгебры.

При построении БД за основу был взят принцип деления типов объектов на классы, применяемый в стандартных репозиториях, поставляемых вместе с промышленными СУБД. При этом он был расширен в рамках предлагаемой модели. Были выделены следующие классы:

- процедуры, функции, таблицы и представления (1 класс),
- параметры процедур и столбцы таблиц (2 класс),
- типы данных (3 класс).

В результате анализа свойств объектов, входящих в каждый из классов, сформирован эффективный вид связи объектов в БД (рисунок 1).

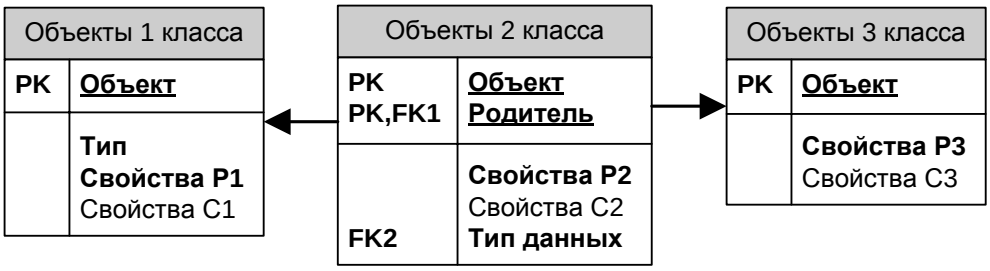

Рисунок 1

Именно для такой архитектуры была реализована БД МС.

<u>В третьей главе</u> рассматриваются вопросы, связанные с загрузкой данных в разработанную БД, построением дерева классификатора, а также способы повышения эффективности работы МС и выбор средств реализации МС.

Для наполнении БД МС данными необходимо построить модели загрузки данных в МС из различных внешних систем.

Рассмотрим работающую систему, построенную на некоторой СУБД. Пусть в такой системе есть множество объектов *S*, хранящееся в БД. Для каждого объекта системы существует своё

описание и, соответственно, свой объект в МС. Таким образом, целью загрузки данных из исходной системы в МС является преобразование исходного объекта S в его описание O.

$S_i \to O_i^j, i \in \{1, n\}, j \geq 1$, где n – количество рассматриваемых объектов в Системе, j – номер описания объекта S.

Для того, чтобы каждый раз не загружать все множество объектов системы в МС, необходимо обрабатывать только те данные, которые изменились с даты последней загрузки. Поэтому для загрузки данных в МС будет рассматриваться подмножество $S_{load} \subseteq S$, где S_{load} – измененные после последней загрузки в МС объекты. Очевидно, что при первоначальной загрузке $S_{load} = S$.

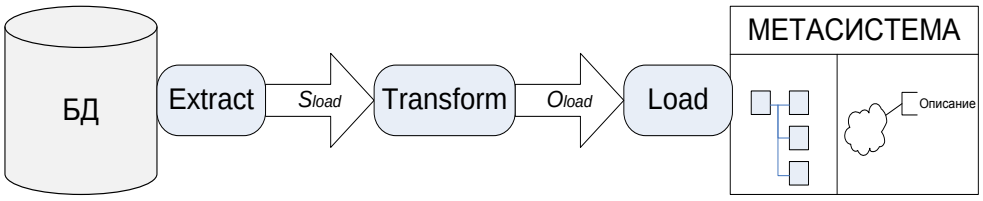

Рисунок 2. Схема загрузки данных в МС

Процедура синтаксического анализатора будет обрабатывать набор S_{load}. На выходе получается множество описаний данных объектов O_{load}. При этом количество элементов в O_{load} не может быть меньше, чем в S_{load}.

Поскольку в рамках построенной модели для классификаторов есть требование, что множество всех подмножеств O_ω должно быть актуальным в любой момент времени, то при загрузке данных в МС должны запускаться процедуры пересчета классификаторов.

Для обеспечения универсальности механизма загрузки разработан специализированный формат, который был промежуточным звеном между МС и исходными данными.

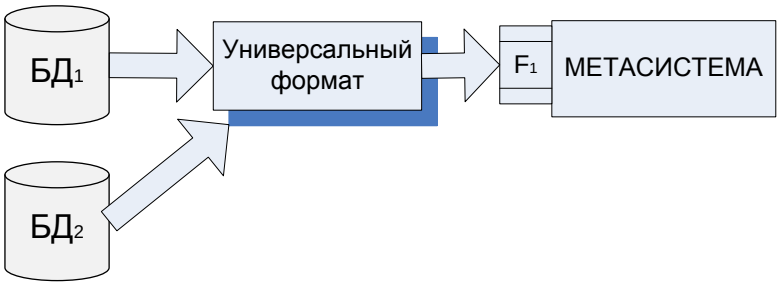

Рисунок 3. Загрузка метаданных

При таком подходе можно разделить этап выгрузки данных из внешних источников и этап загрузки данных в МС. Требуется написать только один загрузчик данных в МС F1 и, поскольку он будет получать данные в универсальном формате, его можно использовать при загрузке данных из любой системы, при этом нет различия в СУБД, на которой построена система-источник.

Поскольку порядок наложения условий на объекты системы с помощью классификаторов не имеет значения, то будем строить систему классификаторов таким образом, чтобы

минимизировать количество условий на каждом уровне. При этом рекомендуется выделить в описании *P* такие свойства, по которым можно было бы разграничить объекты по: области применения, области действия и типу объекта.

Это, в первую очередь, важно при первоначальном наполнении системы данными, когда еще мало описательной информации и строить классификаторы по *C* не всегда корректно из-за того, что многие объекты выпадут из рассмотрения. В то же время, классификация объектов по свойствам *P* гарантирует участие в выборке всех объектов. Отметим, что такой подход построения МС позволяет в дальнейшем в качестве классификатора использовать механизм нечетких классификаторов.

Отметим основные моменты, которые могут упростить построение и дальнейшее использование МС. Их можно назвать организационными мероприятиями, поскольку эти требования должны быть приняты в рамках всей организации, использующей МС. Так, к ним можно отнести:

- Подготовку исходной информации для загрузки. Введение соответствующих требований на оформление кода в плане ведения комментариев.
- Введение пользовательских типов. В исходной системе нужно выделить основные часто используемые домены и для каждого такого домена завести свой пользовательский тип.

Методика построения метасистемы

Результаты, представленные в главах 2 и 3, позволяют сформировать методику построения МС, которая базируется на моделях описания объектов исходной базы данных (п. 2.2) и описания классификаторов (п. 2.3), схеме предложенной базы данных (п. 2.4) и использует в качестве инструментальной поддержки разработанный программный модуль загрузки данных (п. 3.1).

Методика представлена в виде последовательности шагов, при реализации которых могут быть использованы:

- уже существующие модели данных;
- алгоритмы наполнения данными МС;
- предложенные рекомендации.

В качестве основы при построении методики использовался анализ задач и требований из главы 2.1, а также введенные в главах 2.2, 2.3 модели объектов, описаний классификаторов и загрузки данных соответственно. Основными шагами методики являются:

1) **Выбор средств реализации МС.**

Для реализации МС необходимо выбрать следующие программные средства: СУБД, средства загрузки данных, средства реализации клиентского приложения. Эти вопросы рассмотрены подробно в главе 4.

2) **Реализация структуры БД МС**.

В качестве структуры БД следует воспользоваться результатом п. 2.4. Как показал опыт автора, этой структуры достаточно для реализации МС в различных предметных областях и для различных конечных целей (построение аналитических отчетов, использование в качестве справочника по системе). При этом в моделях из п. 2.2, на основе которых построена структура БД, допускается расширение базового набора параметров объектов, что может привести к изменению структуры БД. На практике расширение свойств объектов в рамках моделей п.2.2 влечет за собой увеличение количества параметров у объекта МС. А это, в свою очередь, реализуется добавлением необходимого количества столбцов в таблицу описания.

3) **Загрузка метаданных.**

В качестве средства загрузки метаданных рекомендуется использовать алгоритм, описанный в п. 3.1.5. Этот алгоритм может быть недостаточен, когда требуется загрузка данных в МС из источников, не рассматриваемых в предложенной работе: структурированных документах, изображениях, специальных программах и т.п. В этом случае необходимо реализовать свой собственный загрузчик. При этом новый загрузчик должен обладать следующими свойствами:

- для уменьшения объемов загрузки данных будут обновляться данные только по объектам, измененным после последней загрузки,
- для уменьшения числа затрагиваемых при загрузке параметров будут обновляться только реально обновленные данные.

4) **Построение дерева классификатора.**

При построении дерева классификаторов следует использовать рекомендации из п. 3.2. Так, при первоначальной загрузке необходимо выделить в описании *P* такие свойства, по которым можно было бы разграничить объекты по:

- области применения,
- области действия,
- типу объекта.

При первоначальном наполнении системы данными, когда еще мало описательной информации, строить классификаторы по *C* не всегда корректно из-за того, что многие объекты выпадут из рассмотрения. В то же время, классификация объектов по свойствам *P* гарантирует участие в выборке всех объектов.

5) **Дальнейшее использование (организационные мероприятия).**

Для полноценного использования метасистемы и для минимизации затрат на ее поддержку рекомендуется использовать алгоритмы, состоящие из следующих шагов.

Подготовка исходной информации для загрузки

В рамках данной методики предлагается два варианта подготовки информации для загрузки. Первый заключается в том, что необходимо выполнить всестороннее описание значений *C* для выбранных типов последовательности *K*. Он включает в себя следующие этапы:

- Описание значений *C* для всех объектов Системы
- Загрузка данных в МС
- Первичная классификация данных
- Доопределение объектов, если это необходимо в самой МС.

Второй вариант отличается от первого тем, что мы не будем предварительно заполнять значения *C* в самих объектах. Загрузка данных МС и первичная классификация будут осуществляться только по значениям вектора *P*, который всегда заполнен. После загрузки данных необходимо будет вручную максимально полно описать значения *C* для как можно большего количества типов описаний *K*. Только после этого можно будет проводить классификацию объектов по их описаниям.

Выбор подхода ложится на пользователя, но реально на практике получается комбинация двух вариантов, поскольку обычно в текстах скриптов встречаются комментарии, которые в нашем случае и являются исходным описанием объекта. Они автоматически загружаются в МС, а те данные, которых не хватает и те, по которым не было соответствующего комментария, дополнительно вводятся вручную.

Введение пользовательских типов

Рекомендуется использовать механизм пользовательских типов, встроенный в стандартные поставки большинства промышленных СУБД. Для этого в исходной системе нужно выделить основные часто используемые домены и для каждого такого домена завести свой пользовательский тип. Выделение доменов может производиться в рамках самой МС. В этом случае, после первоначальной загрузки данных начинается выделение и классификация основных типов, которые в дальнейшем переносятся в сами объекты БД.

Построенная по предложенной методике МС обладает следующими свойствами:

1) Непротиворечивость метаданных в МС.

Следует из выбранного способа хранения данных в МС, способа наполнения МС данными, а также возможностей механизма классификаторов. Данные в МС характеризуются следующими параметрами:

- привязка к месту загрузки самих данных (физически, откуда производилась загрузка)
- дата и время загрузки в МС
- пользователь, производивший загрузку

Также в самом имени объекта МС содержится код исходной информационной системы.

2) Актуальность данных.

Следует из выбранного способа загрузки данных в МС. Процесс обновления бизнес-логики или структур исходной информационной системы совмещен с обновлением данных в МС. Таким образом, в МС попадают только измененные данные и при этом загружаются они туда при изменении внешнего источника данных.

3) Возможность классификации метаданных пользователями МС.

Для этого предоставляется механизм классификаторов в рамках схемы БД, построенной по соответствующей модели данных.

В четвертой главе рассматриваются примеры реальных задач из различных предметных областей, которые были решены с использованием инструментария МС. В основном их специфика направлена на подготовку и анализ данных. Выбор именно такого класса задач обусловлен исходными требованиями к системам, для которых строится МС. Поскольку основная площадка для использования МС – уже существующие информационные системы, то в них относительно редко происходит процесс изменения бизнес-логики, зато очень часто эти системы служат основой для анализа – за счет своего возраста в них накоплены большие объемы данных.

В главе подробно рассмотрены шаги реализации МС по уже готовой структуре БД МС и описанному алгоритму наполнения данных. В качестве СУБД была выбрана Microsoft SQL Server 2005. Клиентские приложения были реализованы с использованием различных средств: настройкой инструмента СКАУТ фирмы «Деловые Консультации, Санкт-Петербург» и программированием доступа к нужным таблицам и интерфейса работы с ними напрямую в C#. Для демонстрации возможностей МС в рамках данной работы был выбран второй вариант. Большая часть бизнес-логики была реализована на серверной части с тем, чтобы иметь широкие возможности работы с данными, используя разные клиентские средства. Для загрузки данных в МС выбран алгоритм, предложенный в главе 3.

В качестве примера рассмотривается система автоматизации учета работы 2500 единиц уборочной техники. От каждой из них ежеминутно поступают данные по 19 параметрам (местоположение, скорость, состояние датчиков и т.п.). Таким образом, в данной системе каждый день появляется $2500 \times 60 \times 24 = 3600000$ записей. Через месяц данных станет больше 100 млн. Кроме сохранения оперативных данных, система должна обеспечивать возможность пользователю получать отчеты в различных срезах за различные периоды времени. При приведенных объемах данных задача получения отчетов за разумное время представляется сложной.

Типичным средством решения является построение OLAP-куба. Поскольку исходные объекты системы лежат в MS SQL Server 2005, то в качестве средства построения OLAP выбран продукт, также входящий в MS SQL Server – Analyzes Services.

Использование предлагаемой МС позволило одновременно предоставить пользователям интерфейс настройки показателей для анализа данных и уменьшить само время на настройку

требуемых отчетных форм на 33% (для 8 отчетов). Но главным достоинством внедрения является резкое снижение требований к обслуживающему персоналу.

В качестве второго примера была выбрана задача построения аналитической подсистемы для коммерческого банка. Сложностью данной задачи было очень большое количество различных объектов системы (более 70 справочников и примерно столько же таблиц с оперативными данными). В этом примере, как и в первом случае, удалось существенно снизить требования к исполнителям, а также сократить время самой настройки отчетов на 44% (для 5 отчетов).

При наличии готовой МС затраты на ее подготовку и настройку к новой предметной области для приведенных задач полностью окупились. Первый проект дополнительно окупил более трети затрат на разработку самой МС.

В заключении 4-й главы сформулирован класс решаемых задач, использование в которых предложенной МС было бы наиболее эффективно.

1. Описание системы, поиск тех или иных объектов по их описаниям или получение информации об объекте по его имени.
2. Построение различного рода отчетов из системы (текстовые, графические, OLAP).
3. Различные варианты разработки импорта-экспорта данных в/из системы.
4. Расширения функциональных возможностей системы.
5. Проектирование новых БД, в качестве источника к которым будет выступать исходная рассматриваемая система.

В заключении приводятся основные результаты работы.

Основные результаты работы

В работе проведен анализ различных ИАС, рассмотрены этапы их построения. Отмечено, что для построения и работы с ИАС необходим механизм метаданных. Рассмотрены различные аспекты использования метаданных: стандарты, классификация по пользователям, классификация по классам систем. Рассматриваются существующие метасистемы и на основе анализа делается вывод о необходимости разработать свою собственную систему по работе с МД. К основным результатам работы можно отнести следующие положения:

1) Предложена последовательность шагов, которая позволяет построить МС на основании уже существующей системы. В процессе рассмотрения каждого из шагов данная последовательность была сформулирована в виде методики построения МС.
2) Предложены формальные модели описания объектов МС и классификаторов.
3) На основе предложенной модели сформулированы и обоснованы критерии построения БД МС. Предложена структура БД МС.
4) Разработан способ наполнения данными МС и предложен универсальный алгоритм такого наполнения.

5) Предложен механизм работы с пользовательскими классификаторами в рамках МС.
6) Определены организационные мероприятия, облегчающие использование МС.
7) Предложены этапы реализации МС по уже готовой структуре БД МС и описанному алгоритму наполнения данных.
8) Определен общий класс решаемых МС задач, а также рассмотрены примеры реальных задач из различных предметных областей, которые были решены с использованием инструментария МС.

Также в результате работы был программно реализован механизм загрузки данных, который может быть использован отдельно от разработанной в рамках работы метасистемы.

Публикации по теме диссертации

1) **Бутенко И.В. Метасистема как инструмент для построения аналитических отчетов в информационных системах // Научно-технические ведомости СПбГПУ. – СПб.: СПбГПУ, 2011, №2 (120). с. 32-38 (из перечня ВАК).**
2) **Бутенко И.В., Устинов С.М. Методика построения репозитория метаданных для существующей информационной системы. // Научно-технические ведомости СПбГПУ. – СПб.: СПбГПУ, 2010, №4(103). с. 92-99 (из перечня ВАК).**
3) **Бутенко И.В., Зотов А.А., Устинов С.М. Метасистема как основа доступа в неоднородной распределенной базе данных. // Научно-технические ведомости СПбГПУ. – СПб.: СПбГПУ, 2007, №2(50). с. 247-252 (из перечня ВАК).**
4) Бутенко И.В. Принципы построения метаданных при разработке аналитической системы. // Технологии Microsoft в теории и практике программирования. – СПб.: изд-во Политехн. ун-та, 2004. с. 14-15
5) Бутенко И.В., Басин В.В., Дробинцев Д.Ф., Якимайнен А.Ю. Методика описания информационных структур на базе стандарта CWM // Фундаментальные исследования в технических университетах. – СПб.: изд-во Политехн. ун-та, 2005. с. 161-163
6) Бутенко И.В., Бороздин М.А. Использование LINQ и ENTITY FRAMEWORK при написании приложений для работы с базами данных // Технологии Microsoft в теории и практике программирования. – СПб.: изд-во Политехн. ун-та, 2009. с. 77-78
7) Бутенко И.В., Зотов А.А. Архитектура построения автоматизированной банковской системы для многофилиального банка. // XXXI неделя науки СПбГПУ. Ч.IV: Материалы межвузовской научной конференции. – СПб.: СПбГПУ, 2003. С.8 - 9.
8) Бутенко И.В., Зотов А.А., Устинов С.М. Система управления метаданными. // Технологии Microsoft в теории и практике прогр-ия. – СПб.: изд-во Политехн. ун-та, 2006. с. 96-98

9) Бутенко И.В., Дробинцев Д.Ф. Выбор средств организации метаданных для информационно-аналитических систем. // Технологии Microsoft в теории и практике программирования. – СПб.: изд-во Политехн. ун-та, 2005. с. 78-80
10) Бутенко И.В., Дробинцев Д.Ф., Колесник А.С. Принципы построения классического хранилища данных на основании развиваемых витрин данных. // Технологии Microsoft в теории и практике программирования. – СПб.: изд-во Политехн. ун-та, 2005. с. 32-34
11) Бутенко И.В., Кауров И.В., Дробинцев Д.Ф. Разработка модуля анализа и прогнозирования на базе Microsoft Analysis Services // Технологии Microsoft в теории и практике программирования. – СПб.: изд-во Политехн. ун-та, 2010. с. 93-95
12) Бутенко И.В., Ковалевский В.Э. Использование многомерных структур для обработки больших объемов аналитических данных // Технологии Microsoft в теории и практике программирования. – СПб.: изд-во Политехн. ун-та, 2010. с. 96-97
13) Бутенко И.В., Ковалевский В.Э., Дробинцев Д.Ф.Реализация механизма загрузки метаданных в Microsoft Analysis Services // Технологии Microsoft в теории и практике программирования. – СПб.: изд-во Политехн. ун-та, 2010. с. 97-98
14) Бутенко И.В., Устинов С.М. Разработка моделей и методов построения и автоматизированного наполнения системы метаданных. // Выч-ые, измерительные и упр-ие системы: сб. научн. трудов. – СПб.: изд-во Политехн. ун-та, 2009. с.3-10.
15) Бутенко И.В., Якимайнен А.Ю., Дробинцев Д.Ф. Механизм формирования метаданных на базе стандарта CWM. // Технологии Microsoft в теории и практике программирования. – СПб.: изд-во Политехн. ун-та, 2005. с. 71-73