

УДК 519.2

ББК 22.17

**Обработка данных и управление в сложных системах:** Сборник статей / Под ред. Глуховой Е.В. Томск: Изд-во Том. ун-та, 2002. Вып. 4. 112 с.

ISBN 5-7511-1594-5

Сборник содержит статьи сотрудников и аспирантов факультета информатики, экономики и математики филиала Кемеровского государственного университета в г. Анжеро-Судженске и факультета прикладной математики и кибернетики Томского государственного университета, посвященные статистической обработке временных рядов, актуарной математике, а также вопросам управления в системах массового обслуживания и в измерительных системах.

Для студентов, аспирантов, научных работников, занимающихся вопросами временных рядов и управления в измерительных системах.

УДК 519.2

ББК 22.17

ISBN 5-7511-1594-5

© Филиал Кемеровского государственного университета в г. Анжеро-Судженске, 2002

# ПРИМЕНЕНИЕ МЕТАДАНЫХ ПРИ ПРОЕКТИРОВАНИИ УРОВНЯ ХРАНЕНИЯ В СИСТЕМАХ, ИСПОЛЬЗУЮЩИХ ООРБД

О.А. ЗМЕЕВ, А.Н. МОИСЕЕВ, Д.В. НОВИКОВ

## Введение

Метаданные – это «данные о данных». Они содержат информацию о том, что представляют собой данные: типы, структура, процессы возможного преобразования и т.д. Метаданные используются с момента появления систем управления базами данных (СУБД). Однако в последнее время, при разработке сложных систем документооборота, имеющих открытые интерфейсы к уровню хранения данных, роль и место метаданных в структуре СУБД существенно меняется.

Это происходит по двум причинам. Во-первых, структуры баз данных в современных системах становятся значительно сложнее, и ориентироваться в них без дополнительной информации становится очень сложно. Во-вторых, меняются подходы к использованию самих данных. Если раньше основной целью формирования баз данных являлось накопление информации и последующий поиск, то сейчас основная цель состоит в том, чтобы использовать эту информацию при решении различного рода аналитических задач.

Несмотря на то, что метаданные, используются сейчас гораздо шире и интенсивнее, способ их применения практически не изменился. Как и раньше, они предоставляют разработчикам и пользователям информацию справочного характера. В то же время потенциальные возможности использования метаданных гораздо шире. С их помощью можно не только определять (и переопределять) существующие структуры логических связей данных, но и проектировать новые, оптимальные для решения нестандартных аналитических задач. В настоящей работе рассмотрены способы организации гибкого механизма настройки данных под конкретные аналитические задачи с применением метаданных.

## Требования к уровню хранения данных в информационной системе

Сформулируем стандартные требования к уровню хранения в современной информационной системе. Как уже говорилось выше, механизм настройки данных под конкретные аналитические задачи должен быть гибким. При этом в большинстве случаев желательно не менять логику приложения. Более того, необходимо, чтобы все запросы к базе данных строились относительно текущей схемы, определенной в пользовательском описании метаданных.

Попытаемся суммировать также требования, предъявляемые к информационным системам нового поколения:

- возможность внесения изменений и расширения функциональности в короткие сроки без остановки работы системы (предпочтительно использование высокоуровневого объектного интерфейса) с накоплением информации об обновлениях версий прикладной системы (например, в виде базы знаний);
- открытая архитектура, позволяющая легко осуществлять не только локальную, но и глобальную интеграцию через механизмы Internet/Intranet;
- прогрессивная технология разработки, позволяющая создавать новые версии систем в короткие сроки меньшим числом разработчиков;
- наличие интегрированных простых в использовании, основанных на промышленных стандартах средств разработки.

Реализация этих требований на уровне хранения определяет необходимость от жесткой привязки формирования запросов к фиксированной структуре логических зей. Использование метаданных на уровне хранения позволяет принципиально упростить процесс формирования запросов путем оптимизации для конкретных задач «времени структуры данных с помощью их метаописаний».

Естественно, такой вариант предполагает, что в СУБД должен быть реализован механизм использования полученных от пользователя метаданных при навигации по данным и при поиске информации.

Если рассматривать Хранилище Данных как логическую конструкцию, то данные – это внешнее представление Хранилища. Без наличия актуальных, максимума полных и легко понимаемых пользователем метаданных Хранилище Данных превращается в обычный, но очень дорогостоящий архив. Из всего множества стандартов особый интерес для разработки систем, имеющих дело с метаданными, в настоящее время представляют:

- Unified Modeling Language (UML), язык моделирования, де-факто, являющийся стандартом модели и, следовательно, основным стандартом представления метаданных
- XML вообще и XMI, в частности, как стандартное «транспортное» средство для передачи метаданных;
- серия стандартов, направленных на построение инфраструктуры взаимодействия приложений через Internet: SOAP, eXML, WSDL, WSFL, UDDI.

Систему, которая помимо жестко запрограммированного каркаса содержит неглубоко настраиваемый уровень хранения, управляемый знаниями, сформулированными в виде метаданных, гораздо проще и быстрее модифицировать. По нашему мнению, системы, которые используют метаданные, лучше других соответствуют перечисленным в требованиях к информационной системе нового поколения. С появлением общепринятых языков общения, системы смогут получать новые знания не только от системных администраторов и аналитиков, но и обмениваясь информацией с другими системами.

### Варианты метаданных

Проанализируем варианты использования метаданных в информационных системах. В настоящее время общепринятыми являются следующие типы метаданных [2]:

1. Навигационные метаданные, которые можно назвать «данными о данных». Помогают конечным пользователям, выполняющим запросы к Хранилищу Данных. Этот тип метаданных, также называемый семантическими метаданными, обеспечивает конечному пользователю возможность восприятия информации Хранилища в виде понятных ему категорий и составляющих их сущностей.
2. Управляющие метаданные облегчают процесс извлечения, преобразования, перемещения и загрузки (так называемый ETL-процесс). Этот слой метаданных используется работниками, когда данные собираются в Хранилище Данных из различных источников. Этот тип метаданных идентифицирует расположение исходных данных и их формат, также содержит логику, необходимую для преобразования исходных данных в формат Хранилища, и согласование данных различных источников. Этот тип метаданных важен еще и потому, что процесс обновления Хранилища из внешних источников должен проводиться автоматически по заранее заданному регламенту, поэтому на этапе формирования этих метаданных требуется учитывать способы разрешения большого количества потенциально возможных рисков.

3. Реляционные метаданные – это традиционные метаданные, используемые администратором базы данных, чтобы поддерживать реляционную структуру БД в целостном и непротиворечивом состоянии.

### Стандартная семантическая модель хранилища данных

Описание Хранилища Данных семантическими моделями, входящими в состав метаданных, выполняется администратором системы. При этом метаданные одного Хранилища Данных могут включать теоретически неограниченное количество концептуальных представлений, что объясняется различными потребностями в получении информации отдельных категорий пользователей [3]. Формальное описание любых данных, предназначенных для размещения в хранилище, может быть осуществлено с применением следующего шаблона, изображенного на рис. 1.

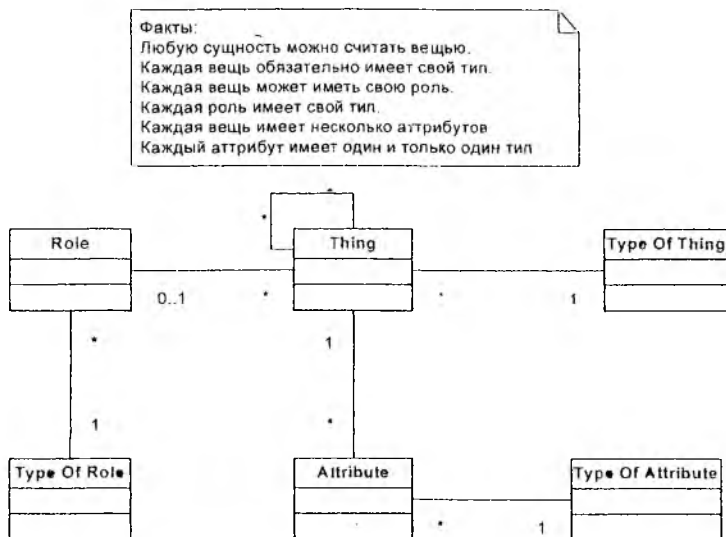


Рис. 1. Общий шаблон описания данных

В основе понятия метаданных лежит иерархия (дерева) смысловых атрибутов логической структуры БД. При этом считается, что смысловые атрибуты, являющиеся физическими полями реляционных таблиц, логически связаны друг с другом отношениями одного из четырех типов.

1. Один к одному (1:1). Рассматриваемый атрибут является простым реквизитом вышестоящего по иерархии атрибута, например, полное наименование организации по отношению к ее уникальному регистрационному номеру в справочнике юридических лиц.

2. Многие к одному (M:1). Рассматриваемый атрибут может быть элементом некоторого классификатора, однозначно соответствующим экземпляру вышестоящего атрибута, например, связь между описанием административно-территориального образования (код ОКАТО), в котором находится организация, и уникальным кодом этой организации;

территория может включать, теоретически, неограниченное множество организаций, но местом расположения организации может быть лишь одна территория: при этом атрибут «территория» описывает организацию.

3. Один ко многим (1:M). Экземпляру вышестоящего атрибута может соответствовать неопределенное количество экземпляров рассматриваемого атрибута, но экземпляр рассматриваемого атрибута связан только с одним экземпляром вышестоящего, например, связь множества банковских счетов с одним владельцем - юридическим лицом: при этом «банковский счет» является множественным атрибутом, характеризующим организацию.

4. Многие ко многим (M:N). Рассматриваемый и вышестоящий атрибут связаны множественным отношением, что позволяет экземпляру одного из них соответствовать произвольному множеству экземпляров другого, например, список юридических лиц и список возможных учредителей; каждое юридическое лицо может создаваться несколькими учредителями, а каждый учредитель может принимать участие в создании ряда юридических лиц.

Также поддерживается возможность рекурсивного описания смысловых сущностей. Так, учредитель юридического лица тоже является юридическим лицом и имеет своих учредителей, которые в свою очередь сами - юридические лица, и так далее. Кроме этого, допускается наличие смысловых сущностей, описываемых одинаковым набором атрибутов. Например, сущности «основной вид деятельности» и «дополнительный вид деятельности» по сути своей есть элементы одного и того же общестатистического справочника ОКОНХ и, следовательно, имеют одинаковый атрибутивный состав.

Описывая по этим правилам смысловые сущности предметной области, отраженные в структуре БД, можно перейти к построению послышней иерархии атрибутов, где ветвями графа будут логические отношения между вершинами - вышестоящими и нижестоящими атрибутами. Метаданные формально описывают такое дерево атрибутов и объявляют отношения между ними в терминах ссылочной целостности реляционной модели. Тем самым появляется возможность представлять конечному пользователю информацию предметной области в ее действительном логическом виде, производя неявно все внутренние операции с реляционными таблицами базы данных. Естественно, что сами метаданные содержат всю необходимую информацию о БД или, точнее, о соответствии между концептуальным и физическим представлением информации.

Понятие иерархической структуры (дерева) строго вводится в теории графов, как сеть частного вида, в которой каждая вершина имеет только одну входящую ветвь. Вершина, которая не имеет ни одной входящей ветви, называется корнем дерева. В каждой построенной в рамках метаданных семантической модели корневая вершина может быть только одна. Но в чистом виде такой моноцентризм, казалось бы неизбежный в иерархической модели, накладывает слишком жесткие ограничения на сценарий обработки содержащейся в ней информации, так как предусматривает единственный вход в ее структуру. Подобным недостатком страдает, в частности, широко распространенная сегодня методология проектирования информационных объектов IDEF.

### Модификация стандартной модели

Рассмотрим другой подход, который позволяет избежать описанной выше статичности. Основная идея заключается в том, что каждый смысловой атрибут может быть объявлен как потенциально корневой, даже если он является промежуточной вершиной основного дерева. Для каждой такой вершины может быть автоматически получена альтернативная иерархия атрибутов, в которой идиграф, расположенный ниже вершины, оста-

нется без изменений, а остальная часть дерева будет инвертирована. Так как каждая связь определяется типом отношения (1:1, 1:M, M:1, M:N), при инверсии подграфа тип каждой входящей в него ветви меняется на противоположный. Таким образом, из одного основного дерева может быть порождено множество альтернативных иерархий – лес. Тем самым достигается множественность возможных точек входа в модель, то есть полниерархический взгляд на единое Хранилище Данных.

Таким образом, в рассматриваемом примере можно построить несколько таких альтернативных деревьев, то есть создать полниерархическую модель этой предметной области. При этом в иерархии, где корнем является уникальный номер юридического лица, одной из характеристик будет множество банковских счетов, которые имеет это юридическое лицо (тип отношения 1:M); в то же время в альтернативной иерархии, где корневой вершиной будет банковский счет, юридическое лицо, которому он принадлежит, после инверсии станет соответственно его характеристикой (тип отношений изменится на M:1).

### **Метаданные и объектно-ориентированный подход**

Защита инвестиций в программное обеспечение сводится к максимальному снижению стоимости сопровождения, внесению изменений и развития системы. Зачастую изменение одной части системы вызывает необходимость в модификации многих смежных частей. Архитектура с четким разделением функциональности по логическим слоям (данные – логика – представление) позволяет существенно снизить эти затраты.

Разбиение приложения на слои стало фактическим стандартом для компаний-разработчиков, использующих различного рода компонентные технологии, однако множество других коллективов продолжают создавать монолитные системы. Из-за необходимости разбирать «клубки» запутанных взаимосвязей стоимость даже небольших изменений в таких системах становится слишком высокой, а повторное использование такого кода часто вообще теряет смысл.

Большинство новых приложений сегодня разрабатывается с использованием объектного подхода, поэтому кажется естественным для хранения данных использовать объектное хранилище. Однако применение объектных СУБД не всегда целесообразно и эффективно: нельзя, например, использовать «любимый» генератор отчетов, практически неизбежны проблемы с масштабируемостью и интеграцией с другими системами, имеющими реляционную основу. Кроме того, объектные СУБД не стандартизированы, и придется привязаться к единственному поставщику, рискуя однажды остаться без перспективы развития и поддержки. Да и ситуация, сложившаяся на рынке объектных СУБД, говорит сама за себя.

Широко распространенный сегодня подход – отображение объектных данных на реляционную структуру. При этом используются такие достоинства реляционных СУБД, как удобство работы с данными как с множествами, наличие хорошего теоретического фундамента и стандартизированного языка доступа и управления этими данными, солидная история и широкое распространение реляционных СУБД, доказавших свою надежность и масштабируемость. Естественно, что для реализации интерфейса между реляционным уровнем хранения и объектной бизнес-логикой, можно использовать механизм мета-данных, описанный выше.

## Заключение

Набор объектов, каким бы хорошим он ни был, – еще не все: требуется инфраструктура управления. И чем больше работы она возьмет на себя, тем проще будет каждый конечный объект информационной системы. До появления компонентной технологии сервером приложений обычно считали слой бизнес-логики системы, построенной в трех-уровневой архитектуре. Сейчас же термином «сервер приложений» все чаще обозначают среды разработки и исполнения приложений, предоставляющие широкий набор служб, сокращающих трудоемкость создания конечных продуктов и повышающих их качество.

Следует учесть, что разработка информационной системы обычно начинается не с нуля. Изначально уже имеется система администрирования прав доступа, универсальная клиентская консоль, набор часто используемых объектов, система автоматического обновления клиентской части приложения, система тестирования работы приложений под нагрузкой, система управления проблемами, которую можно использовать не только во время разработки и сопровождения, но и как часть прикладной системы.

Кроме всего вышесказанного, с помощью технологии, использующей метаданные, можно достаточно быстро предоставить заказчику функциональный макет будущей системы, который сформируется с помощью специальных программ-«экспертов» сразу после описания диаграммы классов и переходов состояний. С этого момента заказчик получает текущий работающий вариант системы и становится полноправным участником процесса разработки.

## ЛИТЕРАТУРА

1. Hurwitz J. The Evolution of Metadata // DBMS. 1997. Vol.10. №8. P. 12-15.
2. Леон М. Соревнование стандартов на метаданные // ComputerWorld-Россия. 1996. №41. С. 23.
3. Тью J. Каждому пользователю – свое представление данных // ComputerWeek-Москва. 1996. №38. С. 32-33.