

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

УДК 004.652.8

DOI 10.17223/19988605/31/8

А.М. Бабанов

ПЕРСПЕКТИВЫ ПРОЕКТИРОВАНИЯ БД, ОТКРЫВАЮЩИЕСЯ С ПРИМЕНЕНИЕМ СОВРЕМЕННЫХ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ДАННЫХ

Освещаются задачи проектирования схем БД, которые зачастую решаются проектировщиками неосознанно, что в дальнейшем усложняет работу и приводит к некорректным и неэффективным результатам. Новые перспективы открываются с использованием современных семантических моделей, обладающих дополнительными возможностями фиксации семантики. За счет более полного описания предметной области в семантической схеме можно существенно автоматизировать процесс проектирования и улучшить качество получаемых схем данных.

Ключевые слова: семантическая модель данных; OR-модель; ERM-модель; проектирование схем БД; задачи, перспективы.

Семантические модели и методика их использования при проектировании схем БД многими незаслуженно недооцениваются. Действительно, если рассматривать их как возможность наглядного представления готовой схемы БД, построенной человеком напрямую в модели конкретной СУБД, то так оно и есть – эффект невелик. В таком случае те многочисленные проблемы проектирования, о которых пойдет речь в статье, преодолеваются часто неосознанно, интуитивно, без глубокого их анализа и использования системного подхода. Многие проектировщики о них даже не подозревают. Важна уже сама по себе констатация этих задач проектирования. Представляют интерес также способы разрешения этих проблем, которые становятся возможными благодаря именно семантическим моделям.

В настоящей работе рассматриваются задачи проектирования схем БД и способы их решения (возможно, автоматического) с использованием современных семантических моделей «Объект – Роль» (OR-модель) [1] и «Сущность – Связь – Отображение» (ERM-модель) [2].

1. Фиксация представлений и требований пользователей отдельно в удобной для них форме

Начинается работа по созданию системы баз данных с анализа бизнес-процессов предметной области (ПрО) и используемых в них данных. Этой информацией во всей полноте, как правило, не владеет ни один эксперт по ПрО, каждый отчетливо представляет только свои задачи. А спроектировать в конечном счете необходимо единую интегрированную БД, обеспечивающую информацией всех участников всех бизнес-процессов. Причем каждый должен пользоваться своим, удобным ее представлением.

Следует отметить, что ошибки этапа анализа требований к будущей системе – самые дорогие из всех ошибок разработчиков. Их запоздалое исправление зачастую приводит к существенным объемам переделок на последующих этапах. Поэтому все требования, полученные от экспертов по ПрО на естественном языке, должны быть сразу же переведены в ясный, точный, формальный вид и перепроверены у экспертов.

Второй аспект, который необходимо учитывать при знакомстве проектировщиков с ПрО, связан с «человеческим фактором». Следует «бережно» относиться к экспертам по ПрО – стараться общаться с ними на их языке, не утруждать их визитами, переспросами об одном и том же. А значит, надо изначально максимально полно зафиксировать семантику ПрО, желательно в формальном виде. И на последующих этапах использовать этот артефакт, а не обращаться к экспертам повторно.

Порядок выполнения работ по анализу ПрО сложился давно и достаточно традиционен. Т. Халпин видит процесс формализации представлений о ПрО следующим образом:

«Этот этап жизненного цикла информационных систем называется концептуальным проектированием. Иногда его называют анализом, чтобы отличить от последующих этапов логического и физического проектирования. Для больших приложений могут выделяться подпроцессы или компоненты, легче поддающиеся анализу, и для них проектируются подсхемы. Впоследствии эти подсхемы интегрируются в глобальную концептуальную схему всей ПрО» [1. С. 51]. «Проектировщики достигают консенсуса в терминологии, поэтому для совпадающих понятий в подсхемах используются одни и те же термины» [Там же. С. 62].

Осознавая положительные стороны ОР-моделирования, самого выразительного среди зарубежных аналогов нашей модели [3], упомянем о недостатках процедуры проектирования Халпина.

1. Немногочисленность и низкий уровень ОР-форм представления моделируемого мира (объекты и роли) требует изначального приведения многообразных других форм человеческого восприятия к этим понятиям.

2. Унификация терминологии бизнес-процессов вряд ли обрадует отдельные группы экспертов и пользователей.

3. Первоначально построенные подсхемы данных бизнес-процессов служат лишь полуфабрикатом для интегрированной схемы ПрО. Дальнейшая их судьба не обсуждается. А ведь они являются важным источником информации для многих последующих задач проектирования.

Халпин, сравнивая свою модель с реляционной моделью, правильно утверждает, что «на концептуальном уровне следует использовать понятия, которые близки и понятны людям» [1. С. 7]. Конечно, объекты и их роли понятнее кортежей и отношений. Но человеческому представлению, кроме объектов и ролей, свойственны понятия «взаимосвязь объектов», «характеристика», «значение характеристики». Вместо простой фиксации семантики ПрО в схеме проектировщик вынужден транслировать описания моделируемого мира, сообщенные ему экспертом, на скудный язык структур данных ОР-модели.

ERM-модель предлагает широкий набор взаимосвязанных структурных понятий (в том числе простых и понятных для человека). Выделение среди них базовых и производных понятий с правилами их взаимного преобразования [4] дает возможность проектировщику в каждом случае использовать наиболее подходящие структурные понятия. Система проектирования в любой момент может автоматически преобразовать схему к нужному понятийному базису.

При ERM-проектировании на первом этапе создаются подсхемы данных в точном соответствии с требованиями пользователей подсистем и с сохранением их представлений и терминологии. Но определяются они не по отдельности, а все вместе составляют единую ERM-схему. На последующих этапах эти сведения будут положены в основу решения многих задач проектирования.

Для иллюстрации предлагаемых идей воспользуемся фрагментами схемы медицинской ПрО, описанной в [5] (рис. 1).

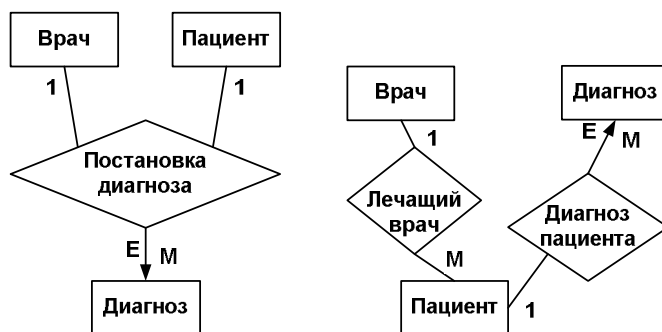


Рис. 1. Фрагменты исходной ERM-схемы медицинской ПрО

На рисунке приведены два различных представления о поставленных пациентам диагнозах: первое – для бизнес-процесса постановки диагноза (наиболее информативное), второе – для процесса лечения пациента в стационаре. Во втором случае знания о враче, который поставил диагноз, не требуется.

Обратите внимание на разницу во множествах связей между пациентами и диагнозом: на левой диаграмме оно тернарно, на правой – бинарно.

2. Приведение подсхем к базовым понятиям и определение взаимоотношений между элементами разных подсхем

Как уже отмечалось, все современные методики проектирования БД либо изначально предполагают создание интегрированной схемы БД, либо интегрируют ранее созданные подсхемы. И в том и в другом случае осуществляется унификация форм представления и наименований явлений ПрО. Это приводит к потере семантики, локализованной в подсхемах, и ее повторному выяснению у экспертов по ПрО в дальнейшем в ходе создания внешних схем пользователей. ERM-модель позволяет сохранить представления и наименования, сложившиеся в разных группах пользователей, и указать, как они взаимосвязаны между собой.

Источниками различий в подсхемах разных пользователей являются их несовпадения во взглядах на те бизнес-процессы, в выполнении которых они участвуют (или не участвуют). Что касается данных этих бизнес-процессов, то люди зачастую по-разному решают задачи структурирования информации, исходя из своих знаний и интереса к ней.

Так, во многих семантических моделях для элементов ПрО предлагается три формы данных – сущность, связь и значение характеристики (или их аналоги). И для каждого явления ПрО проектировщик должен выбрать лишь одну из них и зафиксировать ее в интегрированной схеме хранимых данных (так называемая проблема триализма [6]). Второй задачей структуризации данных является задача правильного определения структуры связей – важно точно определить, сколько и какие сущности их образуют [Там же]. Помимо унификации форм данных и структуры связей при интеграции подсхем безвозвратно теряются некоторые ограничения целостности, определяющие специфические бизнес-правила.

В ERM-моделировании предлагается на этом этапе не отказываться от подсхем данных, а наряду с изначальными (возможно, производными) формами данных и ограничений целостности автоматически порождать их базовые формы (классы и отображения) и на них с помощью операций и отношений между классами и отображениями задавать взаимосвязи между элементами различных подсхем. Для этого используются отношения «равенство», «включение», «непересекаемость» – для классов, и «следствие», «эквивалентность», «несовместность» – для отображений.

В нашем примере множества связей ПОСТАНОВКА ДИАГНОЗА и ДИАГНОЗ ПАЦИЕНТА явно близки по смыслу, но отличаются структурно. В ERM-схеме это можно представить на базовом, более выразительном уровне. При переходе на этот уровень явно вводятся реляционные отображения, определяемые множествами связей (рис. 2).

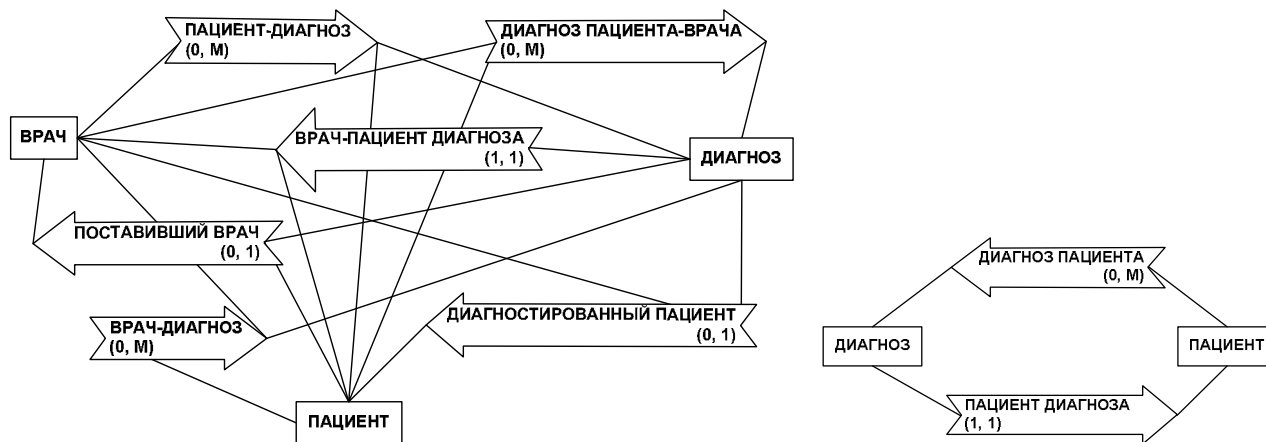


Рис. 2. Диаграммы реляционных отображений ERM-схемы медицинской ПрО

Далее с использованием операций над отображениями определяются взаимосвязи между ними (рис. 3).

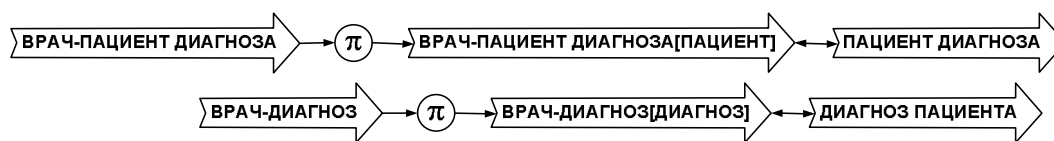


Рис. 3. Диаграммы взаимоотношений отображений ERM-схемы медицинской ПрО

На рисунке указано, что производное отображение, являющееся проекцией базового отображения ВРАЧ-ПАЦИЕНТ ДИАГНОЗА на роль ПАЦИЕНТ, эквивалентно базовому отображению ПАЦИЕНТ ДИАГНОЗА. Это означает, что для каждого диагноза его связь с пациентом определяется одинаково обоими множествами связей. Второй подграф на рис. 3 говорит о том, что и множества диагнозов, полученные с помощью этих множеств связей для любого пациента, совпадают.

3. Определение статуса данных («хранимые – получаемые – частично получаемые»)

Переходя от подсхем пользователей к интегрированной схеме БД, проектировщик сталкивается с еще одной проблемой – какие данные хранить в БД, а какие получать из них автоматически. Халпин выделяет даже три статуса данных – хранимые, получаемые и частично получаемые.

«Получаемый (derived) факт – это факт, который выводится из других фактов математическими вычислениями или логическим выводом. Факт, который нельзя вывести из других фактов, называется хранимым или утверждаемым пользователем (asserted) фактом. Для каждого получаемого факта в схеме данных задается правило его получения» [1. С. 33]. «Частично получаемый (semiderived) тип фактов определяется в том случае, когда ряд фактов этого типа можно вывести, а другие факты будут заданы пользователем» [Там же. С. 99].

В ОР-методике проектирования БД решение задачи определения статуса данных – исключительная прерогатива человека. Он сам делает конкретный выбор и сам определяет правила получения данных, фиксируя свое решение в ОР-схеме.

ERM-схема к этому моменту уже содержит все взаимосвязи элементов подсхем, и есть надежда, что этой информации будет в большинстве случаев достаточно для автоматического решения проблемы «хранимые – получаемые – частично получаемые» (или, по крайней мере, «хранимые – получаемые»). В редких случаях система ERM-проектирования может проконсультироваться у человека. Также можно автоматизировать процесс определения правил вывода получаемых данных. Вся необходимая для этого информация уже задана в ERM-схеме.

Что касается множеств связей диагнозов с пациентами из нашего примера, то очевидно, что из его тернарного варианта легко получить бинарные связи. Обратного преобразования бинарных связей в тернарные не существует. Таким образом, тип связей ПОСТАНОВКА ДИАГНОЗА – хранимый, а ДИАГНОЗ ПАЦИЕНТА – получаемый (с помощью операции проекции).

4. Интеграция хранимых элементов схемы

Традиционный подход к проектированию БД (разработка подсхем и их интеграция в общую схему) предлагает человеку именно на этапе интеграции решать все те задачи, о которых речь шла выше. Зачастую для сложных ПрО это осуществить отнюдь не просто.

Многие методики вообще не регламентируют этот процесс, апеллируя к интуиции проектировщика. Другие, более детальные, указывают основную операцию интеграции – объединение элементов подсхем, напоминая, что «при этом необходимо разрешить возможные конфликты именования, ликвидировать избыточность и неоднозначность» [5. С. 242].

В ERM-моделировании задача интеграции схемы на уровне базовых понятий «класс» и «отображение» решается сама собой после выделения хранимых структур и ограничений целостности. Особен-

ностью такого моделирования является то, что наряду с этими элементами схемы (и непосредственно с ними связана) имеется информация о получаемых структурах, что является основанием для последующей автоматической генерации внешних схем бизнес-процессов.

5. Определение представлений для пользователей

После получения интегрированной схемы необходимо спроектировать подсхемы отдельных пользователей и групп пользователей. «Каждая внешняя схема определяет информационные структуры и операции над данными, которые доступны одной группе пользователей» [1. С. 31]. Поскольку эта задача предполагает создание инструментов, обеспечивающих работу непосредственно с данными, решать ее приходится с использованием средств СУБД и в рамках ее модели данных.

Обычно в этот момент проектировщики вновь обращаются к экспертам с вопросом, что и в каком виде они хотели бы видеть в диалоге с системой БД. Эти представления и необходимо реализовать в СУБД. Основными инструментами разработчиков в случае реляционной СУБД являются представления (view) и триггеры (trigger). Первые обеспечивают необходимые преобразования данных из интегрированной логической модели во внешнюю схему при чтении информации, вторые реализуют проверки данных и обратное преобразование при вводе и изменении данных. Именно эти объекты БД и надлежит создать разработчикам системы. Окончательный вид информация для пользователей приобретет после разработки специализированных диалоговых и отчетных средств.

В случае ERM-моделирования внешние подсхемы фактически совпадают с аналитическими подсхемами. В ходе проектирования элементы этих подсхем ассоциированы с их базовыми эквивалентами и снабжены ссылками на хранимые структуры и ограничения целостности.

Этой информации в ERM-схеме достаточно, чтобы полностью определить внешние схемы пользователей на языке СУБД. В реляционном случае процесс генерации вышеупомянутых представлений и триггеров можно автоматизировать. Для получаемых и частично получаемых типов данных в представлениях определяются способы их вычисления из хранимых типов данных.

Если исходные подсхемы определены в терминах базовых понятий, для удобства восприятия их пользователями автоматически строятся представления для подсхем в традиционных «человеческих» структурных понятиях «сущность», «связь» и «значение».

6. Приближение подсхемы данных к неподготовленному пользователю

Большое внимание создатели и исследователи семантических моделей и методики проектирования БД уделяют донесению семантики ПрО, зафиксированной в схеме, до экспертов и пользователей будущей системы. В этом им видится одна из задач семантического моделирования. Помимо уяснения информационных возможностей схемы эксперты и пользователи могут при этом высказать свои замечания и предложения по ее уточнению и приведению в соответствие с семантикой ПрО.

Лучшему пониманию семантической схемы способствуют:

- близкий к человеческому мировосприятию язык схемы;
- способность представить элементы схемы в виде высказываний естественного языка (вербализация схемы);
- предъявление простых и понятных примеров данных, удовлетворяющих и противоречащих схеме (экземпляризация схемы).

Вот так освещает эти вопросы Халпин:

«Модели ПрО представляются экспертам ПрО для проверки как сами по себе, так и с использованием двух дополнительных возможностей: вербализации структур данных и ограничений целостности, а также предъявления примеров данных, удовлетворяющих или противоречащих схеме (экземпляризации)» [1. С. 10]. «В отличие от UML и ER-модели OR-модель построена на лингвистическом базисе. Для того чтобы извлечь максимальную выгоду от вербализации и экземпляризации при взаимодействии с экспертами по ПрО, лучше использовать язык, который спроектирован специально, в том числе и для этого» [Там же. С. 18].

«Некоторые эксперты в состоянии работать с диаграммами, другие – нет. Некоторые из них хорошо понимают правила, выраженные на естественном языке, другие – нет. Но абсолютно все хорошо работают с конкретными примерами данных. И хотя нет особой необходимости в том, чтобы эксперты работали непосредственно с диаграммами, наличие возможности проиллюстрировать роль непосредственно на диаграмме облегчает задачу проверки проектных решений модельера по поводу тех или иных бизнес-правил» [1. С. 16].

«Для любого типа фактов можно добавить на диаграмме таблицу фактов, заполненную примерами данных для облегчения процесса проверки ограничений целостности. Каждый столбец такой таблицы относится к одной роли» [Там же. С. 10]. «Для двойной проверки ограничений в таблице фактов можно представить также контрпримеры. Конкретные примеры помогают эксперту по ПрО определить, справедливо ли то или иное правило, указанное в схеме данных. Этот дополнительный способ проверки особенно полезен в тех случаях, когда эксперты затрудняются в понимании логических терминов, таких как “каждый”, “по крайней мере”, “не более”, “в точности”, “тот же самый” и т.д.» [Там же. С. 11].

В качестве иллюстрации сказанного можно привести рис. 4, взятый из монографии Халпина [1].

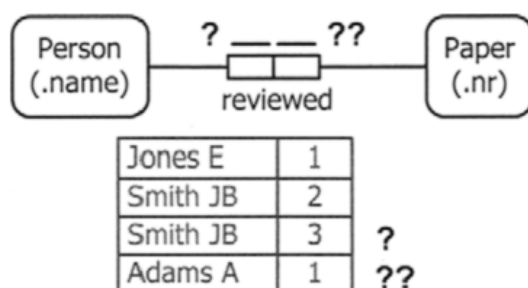


Рис. 4. Экземпляризация с контрпримерами в OR-схеме

Знаками «?» и «??» в OR-схеме и экземпляризации помечены соответствующие друг другу ограничения уникальности роли и контрпримеры. Что касается вербализации, то предполагается, что имена типов сущностей (представлены прямоугольниками с закругленными углами) и предиката (представлен соединенными прямоугольниками ролей) составляют законченное высказывание «Person reviewed Paper» («Человек рецензирует Статью»).

ERM-модель также имеет лингвистические корни. Отображения, по сути, представляют собой предметные функции логики, а последние являются универсальной семантической категорией естественного языка, с помощью которой можно выразить все значимые выражения языка, кроме предложений и единичных имен [7]. С использованием отображений утвердительные предложения приобретают функциональную форму с ярко выраженными подлежащим и сказуемым.

Для нашего примера на рис. 1 можно привести следующие вербализующие правую подсхему высказывания:

- «Врач лечит пациента»
- «Пациент лечится у врача»
- «Пациент имеет диагноз»
- «Диагноз принадлежит пациенту»
- «Врач может лечить нескольких пациентов»
- «Врач может не лечить ни одного пациента»
- «Пациент может лечиться не более чем у одного врача»
- «Пациент может не лечиться у врачей»
- «Пациент может иметь несколько диагнозов»
- «Пациент может не иметь ни одного диагноза»
- «Диагноз должен принадлежать одному и только одному пациенту»

Первые четыре высказывания носят чисто структурный характер и определяют информативность схемы – то, какую информацию БД в состоянии сохранить и вернуть пользователю. Остальные высказывания

звания представляют собой констатацию бизнес-правил ПрО и отражают ограничения целостности, указанные в схеме (рис. 1). А все вместе они позволяют эксперту по ПрО оценить корректность схемы и ее адекватность моделируемому миру.

Заключение

В последнее время, к сожалению, предаются забвению методы структурного анализа и проектирования, а также поддерживающие и автоматизирующие их CASE-средства (Computer Aided Software Engineering – разработка программного обеспечения с помощью компьютера). Их бурное развитие в 1990-х гг. и в начале XXI в. сулило разработчикам информационных систем светлое будущее. Но по каким-то причинам оно не настало. Однако ряд исследователей продолжают развивать это направление исследований, надеясь, что их усилия не напрасны.

Представленная работа затрагивает проблемы, которые часто не замечают современные проектировщики БД. Но от этого их схемы данных не становятся адекватнее, эффективнее и понятнее. Использование семантической методики, подкрепленной CASE-инструментами, в которых реализованы предлагаемые в статье идеи, позволит вывести проектирование БД на качественно новый уровень.

ЛИТЕРАТУРА

1. Halpin T., Morgan T. Information Modeling and Relational Databases. Second Edition. Morgan Kaufman, 2008. 943 p.
2. Бабанов А.М. Семантическая модель «Сущность – Связь – Отображение» // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2007. № 1. С. 77–91.
3. Бабанов А.М. Правила порождения ограничений в семантических моделях данных ORM и ERMM // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2014. № 4(29). С. 68–76.
4. Бабанов А.М. Базовые и производные структурные понятия ERM-модели данных и изоморфное отношение между ними // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2012. № 4 (21). С. 117–126.
5. Цикритзис Д., Лоховски Ф. Модели данных : пер. с англ. М. : Финансы и статистика, 1985. 344 с.
6. Бабанов А.М. Синонимия элементов ERM-схем и ее использование в методике ERM-моделирования для графической нотации // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2014. № 2(27). С. 63–72.
7. Бабанов А.М. Два современных подхода к семантическому моделированию – ORM и ERMM // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2014. № 3(28). С. 46–56.

Бабанов Алексей Михайлович, канд. техн. наук, доцент. E-mail: babanov2000@mail.ru

Томский государственный университет

Поступила в редакцию 12 января 2015 г.

Babanov Alexey M. (Tomsk State University, Russian Federation).

Prospects of database design, opening with application of modern semantic data models.

Keywords: semantic data model, OR-model; ERM-model; DB scheme designing; problems, prospects.

DOI 10.17223/19988605/31/8

Semantic models and technique of their use at DB scheme designing are wrongly underestimated by many persons. Frequently semantic scheme is only an illustration of DB scheme created directly in DBMS model. In this case, those numerous problems of designing, about which this article narrates, are overcome without their deep analysis and use of the system approach. The article covers these problems of DB scheme designing and ways of their decision (it is possible automatic) with use of the modern semantic models – «Object – Role» (OR-model) and «Entity – Relationship – Mapping» (ERM-model).

All modern DB scheme design techniques either initially assume creation of the integrated DB scheme, or integrate earlier created subschemes. Both in that and in the other case, the unification of representation forms and names of application domain (AD) phenomena is carried out. It leads to the loss of semantics located in subschemes, and its repeated finding-out at AD experts during creation of external user schemes.

At the first ERM-designing stage data subschemes are created in exact accordance with requirements of subsystem users and with preservation of their representations and terminology. But they are defined not separately, and all together make the uniform ERM-scheme. At the following stage it is suggested to not refuse these data subschemes, and along with primary data forms and integrity constraints automatically to generate their base forms (classes and mappings) and to set interrelations between elements of various subschemes by the operations and relations between classes and mappings. In further, similar definitions will allow solving automatically many problems of designing.

Passing from user subschemes to the integrated DB scheme, the designer faces one more problem: what data should be stored in the DB and what data should be derived from stored data automatically. Halpin distinguishes even three statuses of data, namely, stored, derived and semiderived. In OR-technique of DB designing the definition of data status is an exclusive prerogative of a person. He makes a concrete choice and he defines data derivation rules, fixing the decision in the OR-scheme.

By this moment, the ERM-scheme already contains all interrelations of subscheme elements, and there is a hope that this information will be in most cases enough for the automatic decision of a problem «stored – derived». Also, it is possible to automate process of the derivation rule definition. All necessary information for it also is set in the ERM-scheme.

After obtaining of the integrated scheme, it is necessary to design subschemes for separate users and user groups. Usually, during this stage designers again address to experts with the question, what they would like to see in dialogue with DB system and in what form. These representations are also necessary for realizing in DBMS. The basic tools of developers in case of relational DBMS are views and triggers.

In the case of ERM-modeling external subschemes actually coincide with analytical ones. During designing elements of these subschemes are associated with their base equivalents and supplied with references to stored structures and integrity constraints. This information in the ERM-scheme is enough to completely define external user schemes in DBMS language. In the relational case, the generation process of the above mentioned views and triggers can be automated. For derived and semiderived data types rules of their calculation from stored data types are defined in views.

Founders and researchers of semantic data models and DB designing technique give also the much attention to the bringing of AD semantics, fixed in schemes, to experts and users of the future system.

There are features promoted the best understanding of the semantic scheme:

- similar to human perception language of the scheme;
- ability to present elements of the scheme as statements of a natural language (verbalization);
- presentation of simple and clear data examples, satisfying and contradicting to the scheme (fact population).

Use of the semantic technique supported by CASE-tools in which ideas offered in this article are realized, will allow to lead DB designing to qualitatively new and higher level.

REFERENCES

1. Halpin, T. & Morgan, T. (2008) *Information Modeling and Relational Databases*. Morgan Kaufman.
2. Babanov, A.M. (2007) Semantic model “Entity – Relationship – Mapping”. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 1. pp. 77-91. (In Russian).
3. Babanov, A.M. (2014) Constraint specifications generating rules in semantic models ORM and ERMM. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 4(29). pp. 68-76. (In Russian).
4. Babanov, A.M. (2012) Base and derivative structural concepts of ERM data model and isomorphic relation between them. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 4 (21). pp. 117-126. (In Russian).
5. Tsichritzis, D., Lochovsky, F. (1982) *Modeli dannykh* [Data Models]. Translated from English. Moscow: Finansy i statistika.
6. Babanov, A.M. (2014) Synonymy of ERM-scheme's elements and its use in ERM-modeling technique for the graphic notation. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 2(27). pp. 63-72. (In Russian).
7. Babanov, A.M. (2014) Two modern approaches to semantic modeling - ORM and ERMM. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 3(28). pp. 46-56. (In Russian).