

Моделирование разреженных существенно-предметных подсхем баз данных

А. Н. Родионов

Вычислительный центр Дальневосточного отделения Российской академии наук, Хабаровск, Россия

Аннотация. В процессе создания информационно-управляющих систем уровня предприятия всегда стремится разместить предметные списки в одной таблице, которая будет задействована во всех ассоциациях, в которых принимают участие предметы. Однако большой диапазон требований, предъявляемых к структурной организации предметных списков, заставляет дробить предметную таблицу на множество локальных таблиц, ориентируясь, в первую очередь, на максимальную компактность последних. В данной работе ставится под сомнение догмат компактности, приводящий к лавинообразному росту количества предметных таблиц, и в качестве альтернативы выдвигается концепция управляемой разреженности, следование которой допускает, на усмотрение пользователя, сведение и близкородственных, и дальнеродственных предметных множеств в один предметный тип. Разрабатывается рамочный структурный примитив, заключающий в себе кроме собственно предметного блока, объединяющего предметные типы и классификаторы-словари последних, стандартные мета-типы и мета-ассоциации со всеми сопутствующими ограничениями, совместно гарантирующими корректность данных о предметах, размещаемых в базе данных.

Ключевые слова: компактность, контролируемая разреженность, семантические и физические домены, мета- и предметная подсхемы.

DOI 10.14357/20718632240405 EDN CPWUGK

Введение

Компактность как предельная заполняемость таблиц (структур, типов) баз данных, в которых размещаются списки предметов, – определяющее, но не единственное требование к соответствующим, как правило многоуровневым, подсхемам, объединяющим подобные специализированные таблицы. За последние десятилетия было разработано, продолжает разрабатываться и совершенствоваться множество методов построения соответствующих многоуровневых абстрактных схем [1-10], в разной степени удовлетворяющих и требованию компактности, и целому ряду других показателей, среди которых часто выделяют: прямой доступ к типам произвольных уровней, модифицируемость без потерь многоуровневой

организации, поддержку многоуровневых абстракций [2], переменное число классификационных уровней [3]. Тем не менее, именно на компактность ориентируются в первую очередь в процессе создания предметных подсхем.

Обратная сторона компактности – слабо контролируемый рост количества предметных таблиц баз данных, который, к тому же, в разы увеличивает число меж объектных ассоциаций и ограничений, присутствующих в схемах. Фактически, стремление к компактности оборачивается значительным усложнением схем и падением их эффективности в широком смысле этого слова – от возрастания трудоемкости обработки данных до снижения достоверности сведений, накапливаемых базами данных. Компактность – это еще и фактор, сдерживающий появление новых методов

разработки информационных моделей, в которых компактности отводится второстепенная роль. Этим, в частности, можно объяснить отсутствие публикаций, посвященных данной тематике.

Очевидной альтернативой сказанному выступает концепции управляемой, контролируемой разреженности, когда близкие по своим свойствам и назначению, но максимально компактные предметные списки, объединяются в группы – элементарные типы, размещаемые в собственных, но уже разреженных именованных структурах – интегральных типах. Основная сложность манипулирования содержимым интегральных структур заключается в очевидном требовании обеспечивать доступ исключительно к актуальным свойствам конкретного предмета, но при этом игнорировать свойства всех других предметов, приписанных той же самой интегральной группе.

Возникающая в этом случае необходимость в создании блока мета-типов, которые инкапсулируют сведения о принадлежности свойств отдельным элементарным типам и раскрывают состав интегральных типов, ведет к необходимости развертывания мета-подсхемы, сопряженной с собственно предметной подсхемой, предназначеннной для размещения интегральных типов. Отсюда целью настоящей работы становится разработка рамочной предметной схемы (модели данных), позволяющей без участия пользователя – потребителя данных – автоматически контролировать корректную разреженность интегральных предметных типов. Разрабатываемый примитив априори должен быть сконфигурирован как структурный прототип, базовый для последующего многоуровневого представления предметов, и при этом удовлетворять расширенному набору требований по сравнению с общепринятыми. В число таких требований входят: обеспечение одномоментного, через обращение к единственной структуре, доступа ко всем предметным экземплярам, присутствующим в моделируемых доменах, многозначный формат представления предметных свойств, выделение обособленных групп семантических и физических характеристик предметов с одновременным моделированием зависимости, существующей между физическими и семантическими свойствами.

Статья организована следующим образом. Начало работы предваряет изложение ключевых положений и элементов архитектуры категорированных типов, на которые опирается исследование, перечисляются и раскрываются применяемые термины, вводятся используемые обозначения. Далее, рассматриваются и анализируются альтернативные подсхемы разреженной организации предметных списков со всеми сопутствующими ограничениями, порождаемыми этими схемами. В третьем разделе работы устанавливается, исследуется и структурно моделируется многозначная функциональная зависимость, связывающая значения, принимаемые семантическими признаками, и физические свойства предметов. Четвертый раздел отведен для рассмотрения расширений мета-подсхемы, полученной на предшествующем этапе, в которую добавлены структуры, содержащие физические характеристики предметов. Параллельно решаются вопросы консолидации предметной и мета подсхем. В заключении перечисляются основные результаты работы и их приложение для многоуровневого моделирования объектов.

1. Предварительные сведения, терминология и обозначения

Работа строится на основе открытой архитектуры категоризированных типов [11], как одного из семантических направлений проектирования структур баз данных, и реляционного способа логического представления последних. Согласно обозначенной архитектуре каждое именованное множество объектов, в моделях данных также называемое типом, должно, по совокупности формальных признаков, принадлежать к какой-либо категории: сущностей – *E*-типов, процессов – *prc*-типов, классификаторов – *spl*-типов, физических и семантических доменов – соответственно *domPh* и *domSm*-типов, слабых сущностей – *we*-типов, объектов-ассоциаций – *rlt*-типов (аналогов *we*-типов) и так далее. Вследствие наличия большого числа трактовок такой категории как сущности, будем к последней относить только те объекты, существование которых не зависит от других объектов. Очевидно, что появление объектов, причисленных к любой из оставшихся категорий, находится

в зависимости от наличия в предметной области таких объектов как сущности.

Среди обширного конгломерата сущностных объектов несложно выделить достаточно емкое подмножество объектов, созданных человеком. Такое подмножество назовем “Предметы (*THings*)”. Предметное множество в повседневной деятельности принято разбивать на именованные подмножества, элементы которых характеризуются примерно одинаковым набором свойств и схожим назначением. Например, это могут быть подмножество “Электроинструментов”, подмножество “Светильников” и так далее. Для ссылок на такие подмножества в статье используется нотация – Th_i , где i -порядковый номер подмножества. Очевидно, что $TH = \bigcup_{i=1}^I Th_i$, и для любых двух отличных Th_i и Th_{i+n} -типов, где $n > 0$ и $i+n \leq I$, $Th_i \cap Th_{i+n} = \emptyset$. Под Th_i также будут подразумеваться компактные элементарные предметные множества, которые не подвергаются дальнейшему разбиению. В свою очередь, элементарные множества могут объединяться, формируя интегральные множества TH_k . Как и для объединения всех элементарных множеств, для всех интегральных множеств будет справедливо: $TH = \bigcup_{k=1}^K TH_k$, $TH_k \cap TH_{k+n} = \emptyset$. Здесь I и K – количество соответственно элементарных и интегральных предметных множеств в моделируемом домене.

На протяжении всей работы вводятся и анализируются альтернативные схемы (структурные модели, диаграммы) данных. Для их сопоставления на интуитивном уровне привлекается несколько неформальных критериев. (Задача детальной проработки формальных критериев в статье не ставилась.) Оптимальной будет признаваться стационарная модель, содержащая одновременно: минимальное число структурных элементов и минимальное количество ограничений, порождаемых этими элементами. Понятие стационарности следует трактовать следующим образом. Модель считается стационарной, если она способна отслеживать ранее не охваченные моделью объекты в предметной области без включения в модель дополнительных элементов: структур, ассоциаций, ограничений. Строго говоря, возможность получения абсолютно стационарных моделей находится под большим вопросом, как и вывод соответствующего количественного

критерия. Поэтому из двух альтернативных диаграмм более стационарной будет признаваться та, которая содержит постоянное количество таблиц одной и той же категории и одного и того же назначения.

В качестве логической модели, по умолчанию, будет фигурировать реляционная модель. В то же время, для лучшего представления и восприятия всех актуальных объектных типов и взаимодействий между ними, в работе задействована полу-концептуальная схема, допускающая, в том числе, и размещение на ней ассоциаций с кардинальностью $M:M$. Также, наряду с термином “тип объекта” или “объектный тип” будут встречаться близкие ему по смыслу термины “таблица”, “структура” и “отношение”. Соответственно, строка таблицы может именоваться записью или кортежем, в зависимости от контекста, а столбцы таблицы называться атрибутами. Также, на полу-концептуальные схемы принято ссылаться как на диаграммы (схемы, модели) баз данных или как на структурные диаграммы.

Для формализации возникающих ограничений целостности привлекаются теоретико-множественные операции и операции реляционной алгебры. И те, и другие оперируют множествами, которые в реляционной алгебре трактуются как отношения. Поэтому при записи ограничений, названия объектных типов будут обозначать названия соответствующих им отношений. Чтобы не перегружать получаемые соотношения избыточной символикой, которая затрудняет восприятие этих соотношений, там, где это допустимо, названия ключевых атрибутов, появляющихся в предикативных операциях, таких, например, как селекция σ или проекция π , совпадают с названиями отношений.

Все предметы на схемах однозначно идентифицируются посредством вспомогательного атрибута – суррогатного ключа. Названия всех суррогатных ключей начинаются с префикса $Id_$. Верхний индекс * сопровождает атрибуты простого или составного первичного ключа.

Каждая слабая сущность обозначается через перечисление названий типов, участвующих в ее формировании. Например, если $Spl-Att-Dom$ – название слабой сущности, то это означает, что последняя представляет результаты взаимодействие трех типов: Spl , Att и Dom . Если слабая

сущность переходит в *rlt*-категорию, то вместо символа “-” используется символ нижнего подчеркивания, например, *Spl_Att_Dom*.

В работе также вводятся несколько классов новых ограничений, которым присваиваются условные названия.

2. Ядро структурного примитива для представления предметных списков

Две диаграммы данных воспроизводят два принципиально отличных подхода к структуризации предметного пространства: с абсолютной компактностью (Рис. 1) и с относительной компактностью, допускающей произвольную степень разреженности предметных таблиц (Рис. 2). В первом случае все предметные типы автономны, какие-либо ассоциации между ними исключены.

Вариант с относительной компактностью (Рис. 2) отличает от предыдущего и гораздо

меньшее число предметных таблиц за счет перехода от *Th* к *TH*-типов, и появление элементов связности между ними. В схему (Рис. 2а) внедрены также две разновидности мета-типов: классификаторы *PH*-типов – ${}^{MT}Split_TH_k$ и генеральный список атрибутов – ${}^{MT}Attributes$, единственный на всю базу данных. Каждый ${}^{MT}Split_TH_k$ содержит список элементарных *Th*, что совместно с ассоциацией ${}^{MT}Split_TH_k$ – *TH*_{*k*} позволяет определиться с принадлежностью произвольного экземпляра *TH*_{*k*}-го типа конкретному *Th*_{*i*}. Контроль доступа исключительно к актуальным для *Th*-типов атрибутам обеспечивается за счет считывания данных, которые несет в себе ассоциация MAttributes – MSplitTH (все мета-элементы помечены надстрочным знаком – *Mt*).

Диаграмму с относительной компактностью можно оптимизировать (Рис. 2, б), упаковав все ${}^{MT}Split_TH_k$ в один тип ${}^{MT}Split_TH$, снабдив последний рекурсивной связью. Одновременно можно расширить ее функциональность,

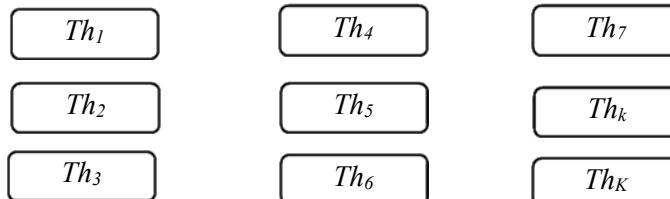


Рис. 1. Организация сущностного пространства с абсолютно компактными предметными типами

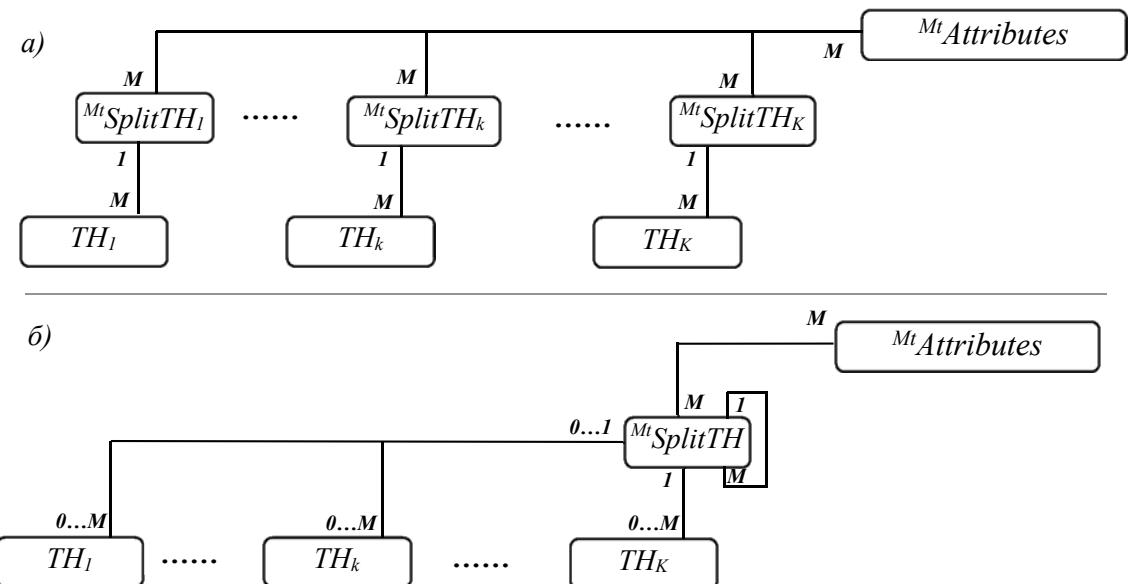


Рис. 2. Организация сущностного пространства с относительно компактными предметными типами

<i>M¹SplitTH</i>		
<i>Id sp</i> ^{**}	<i>Name</i>	<i>Id sp rec</i>
1	Электроинструменты	
2	Перфораторы	1
3	Дрели	1
4	Мебель	
5	Диваны	4
6	Столы	4
7	Шкафы	4
8	Брус	

Электроинструменты (TH₁)	
<i>Id th₁</i> [*]	<i>Id sp</i>
Пр1	2
Пр2	3
Др1	3
Др2	3
Др3	3
Шк3	7

Брус (TH₃)	
<i>Id th₃</i> [*]	<i>Id sp</i>
Бр1	8
Бр2	8

Мебель (TH₂)	
<i>Id th₂</i> [*]	<i>Id sp</i>
Дв1	4
Дв2	4
Шк1	7
Шк2	7
Ст1	6
Пр3	2

Рис. 3. Пример распределения элементарных предметных списков среди *TH*-типов

разрешив одновременное присутствие в диаграмме как типов с относительной, так и с абсолютной компактностью. Для этого достаточно перейти от кардинальности *1:M* ассоциаций *M¹SplitTH – TH_k* к кардинальности *0...1:0...M*. Тип *M¹SplitTH* также обладает по сравнению с набором *M¹SplitTH_k*-типов как минимум одним неоспоримым преимуществом, выполняя роль словаря данных. Уникальные названия элементарных *Th*-типов, сосредоточенные в *M¹SplitTH*, позволяют отказаться от стандартного обязательного атрибута “наименование предмета”, который должен присутствовать в любом предметном списке.

Иллюстрацией сказанного может служить фрагмент заполненной базы данных (Рис. 3), соответствующий диаграмме на Рис. 2, б.

В отличие от схемы с абсолютной компактностью, схема на Рис. 2, б порождает по меньшей мере два ограничения: ограничение “эффекта рекурсии” и ограничение “запрета на пересечение *TH*-типов”. Первое ограничение сводится к тому, что элементом ассоциации *M¹SplitTH – TH_k* со стороны *M¹SplitTH* может выступать только идентификатор записи самого нижнего уровня. Для отношения *M¹SplitTH* на Рис. 3 указанное ограничение накладывает запрет на участие в *M¹SplitTH – TH_k* кортежей с *Id sp*=1 и *Id sp*=4.

Второе ограничение уже было formalизовано ранее как *TH* = $\bigcup_{k=1}^K TH_k$, $TH_k \cap TH_{k+n} = \emptyset$. Оно исключает появление экземпляров одного и

того же *Th*-типа более чем в одном *TH*-типе. Вычеркнутые на Рис. 3 кортежи в *TH₁*- и *TH₂*-типах иллюстрируют действие данного ограничения.

Схема на Рис. 2, б будет неполной, если не определиться с обеспечением общей точки доступа к предметным данным. Воспользуемся фрагментом диаграммы, который содержится в [12]. Основная идея – разместить все существенные экземпляры в единственной таблице с атрибутами, общими для всех предметов, а уникальные атрибуты, характеризующие отдельные существенные группы, сконцентрировать в специализированных таблицах, например, таких, которые показаны на Рис. 2. С учетом сказанного базовый структурный примитив примет окончательный вид (Рис. 4). Теперь каждый предмет появится в базе данных дважды. Один раз – в *Things*, а один – в одном из *TH*-типов. С учетом того, что содержимое интегральных типов не пересекается, мощность *Things* будет всегда равняться сумме мощностей всех интегральных типов: $M(Things) = \sum_{k=1}^K M(TH)_k$. Для диаграммы на Рис. 4 это станет одним из обязательных структурных ограничений. Нетрудно заметить, что подобная организация схемы предполагает дублирование записей о предметах, приводящее к избыточности схемы. Но в данном случае избыточность – это компромисс, обеспечивающий замыкание всех ассоциаций, в которых могут участвовать предметы, на уникальном в своем

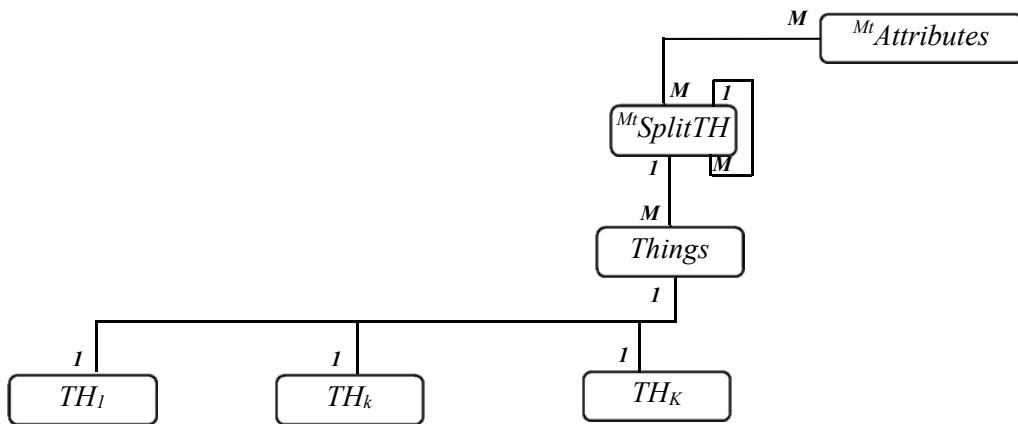


Рис. 4. Базовый примитив для представления предметных типов в базах данных

роде *Things*-типа, а не распределение ассоциаций среди всех *TH*-типов.

3. Зависимости между физическими и семантическими свойствами предметов. Ограничение соответствия физических и семантических свойств

Содержимое многочисленных сайтов, например, тех же торговых площадок, демонстрирует широкий спектр признаков, подключаемых пользователями для описания предметов. Опираясь на столь масштабную выборку, находящуюся к тому же в свободном доступе, несложно выявить ряд зависимостей, которые объективно имеют место среди некоторых характеристик. Вынося за рамки работы вопросы однозначной семантической идентификации предметов,

остановимся на зависимости, устанавливающей соответствие между семантическими и физическими свойствами предметов. Для наглядности, на Рис. 5 приводятся некоторые из многочисленных физических и семантических характеристик. Одна из таких зависимостей действует между элементами двух множеств, одно из которых представлено значениями, принимаемыми семантическим признаком “Привод”, и подмножеством физических характеристик (Рис. 5).

Вопрос информационного моделирования зависимостей между атрибутами-свойствами предметов – это один из аспектов корректности, достоверности данных.

Поэтому вполне естественно, что в составе структурных диаграмм должны появиться элементы, нацеленные на фиксацию и отслеживание соотношений подобного рода. Между тем, одного только списка атрибутов, присутствующего в



Рис. 5. Некоторые физические и семантические признаки предметов. Многозначная зависимость между значениями семантических признаков и соответствующими им физических величин

составе базового примитива на Рис. 4, для решения поставленной задачи будет явно недостаточно. Но перед тем как преобразовать диаграмму на Рис. 5 и привести ее к искомому виду, уточним некоторые детали, касающиеся особенностей взаимодействий между предметными и доменными типами.

Возьмем в качестве эталонного модельного предмета условную “дрель”. Если эта дрель “сетевая” (семантический признак “привод” принял значение “сетевая”), то актуальными для такой дрели окажутся такие физические характеристики как “напряжение”, “мощность” и “длина кабеля”. Если эта дрель “аккумуляторная”, то к перечисленным признакам добавится “емкость батареи”, а признак “длина кабеля” будет исключен из списка. Выделим, пока еще только вербально, описанное ограничение в отдельный класс, присвоив последнему сокращенное название *oФзС* – ограничение зависимости физических величин от

значений семантического признака. Уникальность *oФзС* в том, что оно принимает в расчет связь между значениями семантической величины с самими физическими величинами, а не со значениями физических величин, как это обычно принято. Именно последнее обстоятельство создает трудности в получении структур, способных оптимальным образом отразить действие *oФзС*. Если взять за основу диаграмму на Рис. 4, то для поддержки *oФзС* диаграмму придется дополнить произвольным числом собственно семантических таблиц – *Dom_s*, каждая из которых будет отведена под список значений конкретной семантической характеристики, а также снабдить диаграмму двумя дополнительными мета-структурами (Рис. 6).

Первая структура – *Mt-const Attributes' kinds* – будет классифицировать атрибуты либо на физические – *Ph*-атрибуты, либо семантические – *Sm*-атрибуты и, следовательно, содержать

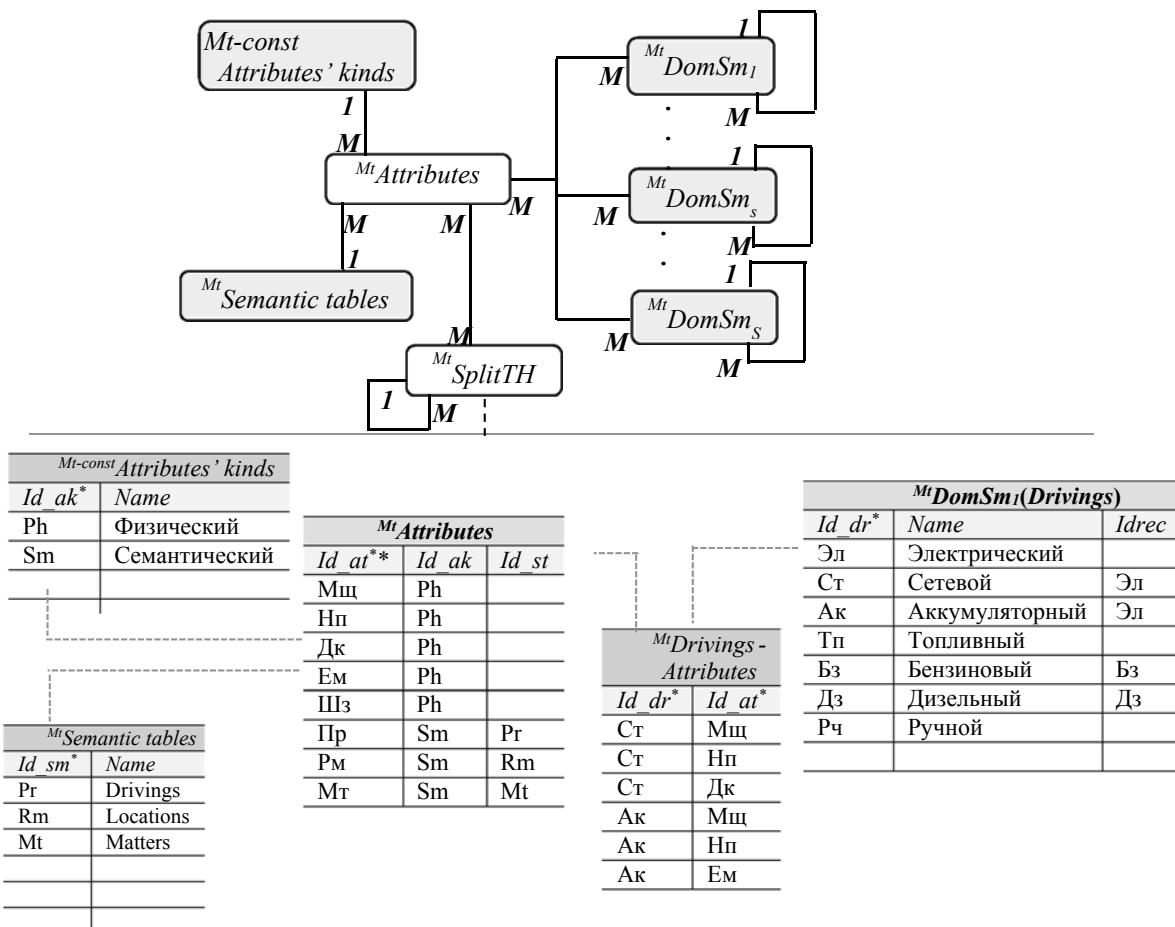


Рис. 6. Один из вариантов организации доменных списков

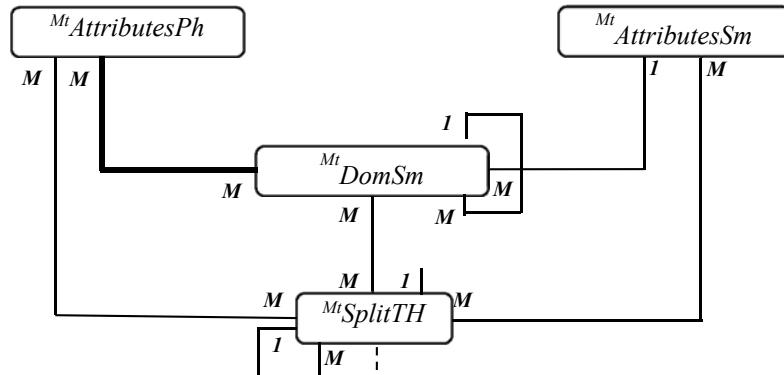


Рис. 7. Дифференциация атрибутов и моделирование оФзС в предметном примитиве

только две записи, и эти записи станут константами базы данных, на что непосредственно указывает надстрочный знак “*Mt-const*” в наименовании *Attributes' kinds*. Другая – *Mt Semantic tables* – включать переменный список семантических таблиц. Полученная в результате диаграмма мало того, что перегружена однотипными структурами в лице *DomSm*-типов и связанными с ними слабыми сущностями, она и крайне нестационарна, так как при появлении новых семантических атрибутов требует расширения и реорганизации вследствие включения в ее состав дополнительных структурных элементов. И это только один из ее недостатков, к которым также следует отнести ряд ограничений, которые порождает схема. Например, ассоциация *Mt Semantic tables* – *Mt Attributes* должна инициироваться только для *Sm*-атрибутов, а атрибут *Id_at* в слабых сущностях *Mt DomSm-Attributes* может принимать значения только из подмножества физических атрибутов.

Более предпочтительным будет решение, основанное на разбиении списка атрибутов на два подмножества: подмножество физических атрибутов и подмножество семантических атрибутов с одновременным сосредоточением значений всех семантических признаков в одной структуре с условным наименованием – *Mt DomSm* (Рис. 7).

4. Подключение физических доменов

Физические свойства предметов не могут принимать произвольные значение. Как минимум они должны лежать в границах предопределенных интервалов. Для задания актуальных значений отдельных свойств, обязательно

соотнесенных с экземплярами *Mt Split TH*-типов, что важно, привлекается специальная категория типов – доменные типы [13]. Доменные типы – *domPh* – могут быть в зависимости от способа представления физических значений дискретными или непрерывными, а также интервальными или точечными. Поскольку всестороннее рассмотрение всех возможных комбинаций физических доменов не входит в задачи статьи, ограничимся подключением к предметным типам только доменов, содержащих дискретные и точечные значения. Но при этом сконцентрируемся на общем случае, когда физическое свойство предмета может одновременно принимать несколько значений, будучи многозначным свойством. Моделирование общего случая, что будет показано ниже, приводит к достаточно компактным и малоразмерным схемам с минимальным числом структур.

Собственно, сама задача подключения физических доменов распадается на две части. Прежде всего, необходимо определиться со структурами, задающими значения физических доменов, а затем подключить последние к предметным типам и формализовать все сопутствующие ограничения, обусловленные конфигурацией полученной схемы.

Чтобы задать множество значений, которые может принимать какая-либо характеристика предмета, отнесенного к элементарному *Th_i*-типу, потребуется “разорвать” ассоциацию *Mt AttributesPh* – *Mt Split TH* на Рис. 7 и сформировать *we*-тип *Mt Spl Att Dom* с участием *Mt AttributesPh*, *Mt Split TH* и *Mt DomPh*, как это сделано на усовершенствованной конструкции (Рис. 8).

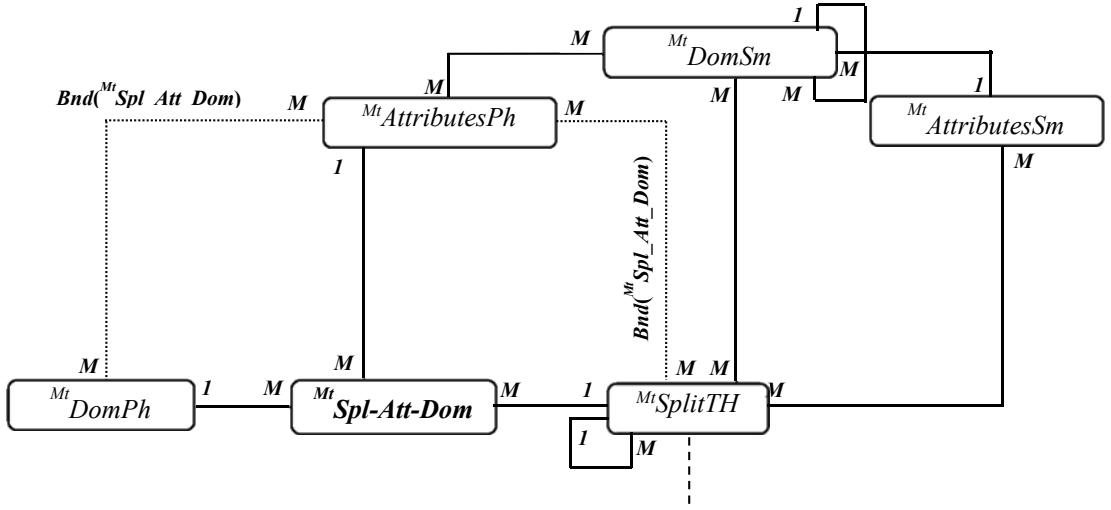


Рис. 8. Подсхема с комплексным физическим доменом

Множество значений всех физических характеристик с условным наименованием *Value* можно задать на универсальном (в контексте области определения физических свойств) множестве действительных чисел, которое является одним из обязательных типов данных, встроенных в состав любой системы управления базой данных. При этом полученное множество разместится в единственной структуре ${}^{Mt}DomPh$, представляющей собой своего рода комплексный физический домен, что избавит от необходимости создавать отдельные ${}^{Mt}DomPh$ -типы для каждой физической характеристики. Добавим в схему на Рис. 8 также необязательную ассоциацию ${}^{Mt}AttributesPh - {}^{Mt}DomPh$, тем самым сведя к минимуму риск ввода некорректных значений в ${}^{Mt}Spl-Att-Dom$. Слабая сущность, которая будет соответствовать ${}^{Mt}AttributesPh - {}^{Mt}DomPh$, выступит в роли ограничения (1) на содержимое ${}^{Mt}Spl-Att-Dom$:

$$\pi_{Att,Dom}^{Mt} {}^{Mt}Spl-Att-Dom \subseteq {}^{Mt}AttributesPh - {}^{Mt}DomPh, \quad (1)$$

где $\pi_{Att,Dom}({}^{Mt}Spl-Att-Dom)$ – операция проекции на атрибуты *Att* и *Dom*, входящие в состав отношения ${}^{Mt}Spl-Att-Dom$. В [13] подобного рода ограничения маркируются как ограничительные – *bounded (bnd)* – с прикреплением к ним названия отношения, на которое распространяется действие этого ограничения. Фрагмент экземпляра базы данных (Рис. 9), соответствующий схеме на Рис. 8, иллюстрирует сказанное.

Подобная структурная формализация физических доменов (Рис. 8) предопределяет два варианта подключения последних к предметным типам: фиксированное (Рис. 10 и 11) и гибкое (Рис. 12).

Тип ${}^{Mt}Spl-Att-Dom$ на Рис. 10 приведен к *rlt*-категории, поскольку участвует во взаимодействии с *Things*. Он также становится эксклюзивным поставщиком данных для формирующейся слабой сущности ${}^{Mt}Spl-Att-Dom$, которая аккумулирует значения всех физических характеристик предметов. Заметим, что отражение многозначности свойств привело к тому, что привычные атрибуты-свойства, которыми принято наделять предметные типы, исчезли из состава последних.

Фиксированный вариант оптимален и по количеству структурных элементов, вошедших в схему, и по количеству ограничений, которое будет единственным для этой схемы. Этим ограничением станет ограничение на допустимый состав ${}^{Mt}Spl-Att-Dom$, который не может формироваться произвольным образом. Так, экземпляры предметов, берущиеся из *Things*, и экземпляры ${}^{Mt}Spl-Att-Dom$ через ${}^{Mt}SplitTH$ связаны друг с другом. Каждый предмет из *Things* принадлежит некоторому *th*-типу, список которых размещен в ${}^{Mt}SplitTH$. В свою очередь, список свойств *th*-типов, сцепленный со значениями, которые разрешены для этих свойств, заключен в ${}^{Mt}Spl-Att-Dom$. Поэтому любому предмету из ${}^{Mt}Spl-Att-Dom$ можно поставить в

<i>M^t AttributesPh</i>	
<i>Id_at[*]</i>	<i>Name</i>
U	Напряжение
M	Мощность
Ve	Объем двигателя

<i>M^t DomPh</i>	
<i>Id_at[*]</i>	<i>Value[*]</i>
U	12
U	127
U	220
M	1,5
M	2,5
Ve	38

<i>M^t AttributesPh</i>		
<i>Id_at[*]</i>	<i>Name</i>	<i>Value[*]</i>
U	Напряжение	12
M	Мощность	127
Ve	Объем двигателя	220

<i>M^t DomPh</i>		
<i>Id_at[*]</i>	<i>Value[*]</i>	<i>Value[*]</i>
U	12	127
M	1,5	2,5
Ve	38	220

<i>M^t SplitTH</i>		
<i>Id_sp[*]</i>	<i>Name</i>	<i>Value[*]</i>
Пр	Перфоратор	12
Др	Дрель	127
Бз	Бензопила	220

Рис. 9. Экземпляр базы данных с комплексным физическим доменом

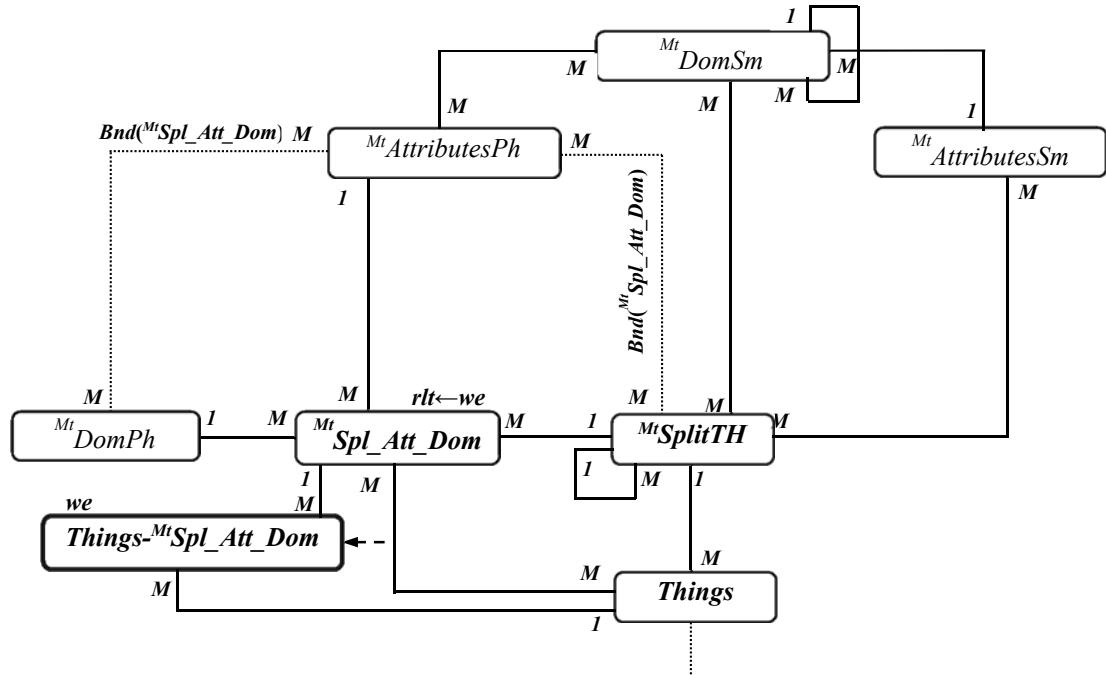


Рис. 10. Фиксированное подключение физических доменов

соответствие элемент только из определенного подмножества ${}^M Spl_Att_Dom$, которое ассоциируется с этим предметом. При записи ограничения на указанное подмножество будем ссылаться как на подмножество с именем *Results*. Поскольку далее для формализации ограничения привлечены операции реляционной алгебры, удобно будет манипулировать реляционной терминологией, в том числе ссылаясь на типы как на отношения.

Обозначим через $tp_n^{App}(Att)$ значение, принимаемое некоторым атрибутом Att в n -м кортеже отношения App . В нашем случае $App = "Things-M^t Spl_Att_Dom"$, а $Att = "Id_th"$. Id_th – атрибут, значение которого однозначно идентифицирует предмет в $Things$. Зная идентификатор предмета и выполнив естественное соединение \bowtie $Things$ с $M^t SplitTH$, а $M^t SplitTH$ с $M^t Spl_Att_Dom$, получим искомое подмножество для выбранного предмета – $Results(tp_n^{App}(Att))$:

<i>M^tSpl_Att_Dom</i>				<i>M^tSplitTH</i>	
<i>ID</i> [*]	<i>Id_sp</i>	<i>Id_at</i>	<i>Value</i>	<i>Id_sp</i> [*]	<i>Name</i>
1	Пр	U	127	Пр	Перфоратор
2	Пр	U	220	Др	Дрель
3	Пр	M	1,5	Бз	Бензопила
4	Др	U	220		
5	Др	M	1,5		
6	Бз	M	2,5		
7	Бз	Ve	38		

<i>Things-^{M^tSpl_Att_Dom}</i>	
<i>Id_th</i> [*]	<i>ID</i> [*]
Пр1	1
Пр1	3
Пр1	4

<i>Things</i>	
<i>Id_th</i> [*]	<i>Id_sp</i>
Пр1	Пр
Пр2	Пр

Рис. 11. Действие ограничения на состав *Things-^{M^tSpl_Att_Dom}*

$$Results(tp_n^{App}(Att) = \sigma_{Id_{th}=tp_n^{App}(Att)}(Things \bowtie^{M^t} SplitTh \bowtie^{M^t} Spl_Att_Dom)).$$

Таким образом, искомое ограничение примет следующий вид:

$$tp_n^{App}(Att) \in Results(tp_n^{App}(Att)).$$

Действие ограничения иллюстрирует еще один фрагмент базы данных (Рис. 11), в который преднамеренно включен кортеж, нарушающий действие данной рестрикции. Некорректный кортеж вычеркнут из *Things-^{M^tSpl_Att_Dom}*.

Упомянутый ранее вариант с гибким подключением (Рис. 12) также как и вариант с жестким присоединением физического домена

предусматривает контроль значений, принимающих физическими свойствами предметов со стороны *Spl-Att-Dom*, но в тоже время предоставляет свободу пользователю в указании произвольных значений свойств, когда, по каким-то причинам, эти значения предварительно не были помещены в *Spl-Att-Dom*. Как следствие, нет необходимости приостанавливать ввод данных из-за неполноты затребованных пользователем доменов и не формировать последние “на лету”.

Представленная схема генерирует два ограничения. Одно из них призвано не допустить указания несобственных атрибутов (атрибутов,

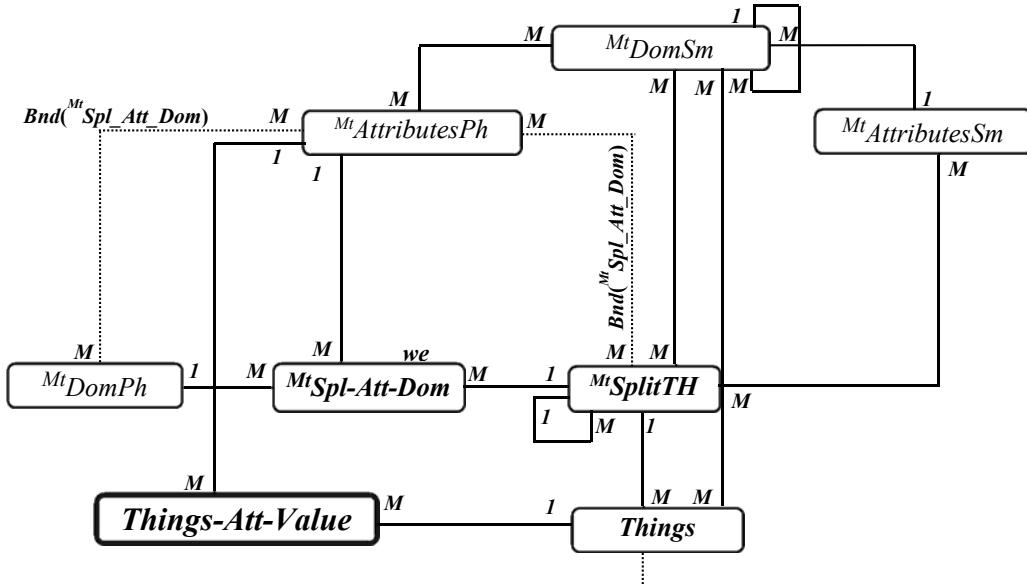


Рис. 12. Гибкое подключение физических доменов

<i>M^tSpl-Att-Dom</i>			<i>M^tSplitTH</i>	
<i>Id_sp</i> ^{**}	<i>Id_at</i> [*]	<i>Value</i> [*]	<i>Id_sp</i> [*]	<i>Name</i>
Пр	U	127	Пр	Перфоратор
Пр	U	220	Др	Дрель
Пр	M	1,5	Бз	Бензопила
Др	U	220		
Др	M	1,5		
Бз	M	2,5		
Бз	Ve	38		

<i>Things-Att-Value</i>			<i>Things</i>	
<i>Id_th</i> [*]	<i>Id_at</i> [*]	<i>Value</i> [*]	<i>Id_th</i> [*]	<i>Id_sp</i>
Пр1	U	220	Пр1	Пр
Пр1	M	1,5	Пр2	Пр
Пр1	Ve	15		
Пр2	M	1,8		

Рис. 13. Проверка на выполнение ограничения “несобственных” атрибутов

не принадлежащих конкретному предмету) при заполнении отношения *Things-Att-Value*. Назовем данную реаликацию ограничением “несобственных атрибутов” и представим алгоритмическую процедуру, позволяющую провести проверку на его возможное нарушение.

Сведения о принадлежности атрибутов *Th*-типам сосредоточены и в слабой сущности *M^tSpl-Att-Dom*, и в *M^tAttributesPh-SplitTh*. Но в *M^tSpl-Att-Dom* могут присутствовать не все атрибуты. Поэтому за базовый источник информации следует принять *M^tAttributesPh-SplitTh*. Обратим внимание, что ограничительная ассоциация *M^tAttributesPh-SplitTh* начинает выполнять в схеме еще одну важную функцию, обеспечивая исчерпывающее представление сведений о закреплении атрибутов за *Th*-типами. Проверка сводится к получению списка (2) собственных атрибутов предмета в виде набора данных *SpisokTh* и последующего использования последнего в качестве ограничения (3):

$$SpisokTh = \pi_{Att}(\sigma_{Id_{th} = current}(Things) \bowtie M^tSplitTh \bowtie M^tAttributesPh-SplitTh), \quad (2)$$

$$tp_{Id_{th}=current}^{Things-Att-Value}(Id_{att}) \in SpisokTh. \quad (3)$$

Запись *Id_th=current* в (2) и (3) означает, что ключевой атрибут *Id_th* отношения *Things-Att-Value* принимает текущее значение, ссылающееся на конкретный предмет.

Фрагмент базы данных на Рис. 13 иллюстрирует действие ограничения “несобственных атрибутов”. В *Things-Att-Value* вычеркнут кортеж нарушающей указанное ограничение.

Следующее ограничение – сигнальное, информационное, призванное предупредить пользователя о потенциальной опасности ввода значений, которые отсутствуют по каким-либо причинам в *M^tSpl-Att-Dom*. Для этого достаточно сформировать запрос, в котором для текущего кортежа из *Things-Att-Value* будет вестись поиск соответствующему ему кортежа в *M^tSpl-Att-Dom*. Естественное соединение *Things-Att-Value*, *Things*, *M^tSplitTH* и *M^tSpl-Att-Dom* обеспечит поддержку данного ограничения. Листинг 1 содержит текст запроса.

Листинг 1

```
SELECT [Spl-Att-Dom].Id_sp, [Spl-Att-Dom].Id_at, [Spl-Att-Dom].Value
FROM SplitTH LEFT OUTER JOIN
    [Spl-Att-Dom] ON SplitTH.Id_sp = [Spl-Att-Dom].Id_sp RIGHT OUTER JOIN
    Things ON SplitTH.Id_sp = Things.Id_sp RIGHT OUTER JOIN
    [Things-Att-Value] ON Things.Id_th = [Things-Att-Value].Id_th
where [Things-Att-Value].Id_th=current-th and [Things-Att-Value].Id_at=current-at
```

Переменным `current-th` и `current-at` в листинге 1 присваиваются текущие поисковые значения для пары (предмет, атрибут).

Наличие альтернативных вариантов схем для подключения доменов закономерно ставит вопрос выбора одного из них. Если дистанционироваться от трудоемкости ввода и контроля данных, то фиксированную схему можно рассматривать в качестве совершенной модели, к которой следует стремиться.

Заключение

В статье излагаются базовые элементы концепции контролируемой разреженности подсхем данных, специализирующихся на размещении предметных списков, обосновываются преимущества концепции в сравнении с распространенным и общепринятым компактным подходом, рассматривается широкий круг вопросов реализации и обеспечения разреженности. Показано, что предметная подсхема должна включать два взаимозависимых блока в лице блока собственно предметных структур и блока мета-данных. Последний представлен комплексом мета-структур, мета-ассоциаций и мета-ограничений и ответственен за достоверность предметных данных.

Обнаружена и исследована многозначная функциональная зависимость, связывающая физические и семантические характеристики предметов. Разработана модель оптимального представления экземпляров этой зависимости в моделирующих структурах.

Полученная в работе разреженная подсхема отличается высоким уровнем стационарности и легко адаптируется и встраивается в глобальную диаграмму базы данных любой предметной области.

Предложенный примитив, после проведения дополнительных исследований, также может быть приведен к многоуровневому формату, обеспечивающему поддержку межуровневых отношений материализации, возникающих между экземплярами предметных типов, которые располагаются на смежных абстрактных уровнях. Тем самым, закладывается фундамент и просматривается реальная перспектива созда-

ния в ближайшее время универсального структурного прототипа, альтернативного по отношению к аналогичным прототипам, разработанных посредством существующих методов, но отличающегося от всех других примитивов высоким уровнем стационарности.

Литература

1. Atkinson C., Kühne T. The essence of multilevel metamodeling. In.: Gogolla M., Kobryn C. (eds.) UML 2001. Berlin Heidelberg: Springer; 2001. p. 9-33.
2. Neumayr B., Schrefl M., Thalheim B. Modeling techniques for multi-level abstraction. In: Kaschek/Delcambre (eds.): The evolution of conceptual modeling. Berlin Heidelberg: Springer-Verlag; 2011. p. 68-92.
3. Fonseca C. M., Almeida J. P., Guizzardi G., Carvalho V. A. Multi-level conceptual modeling: theory, language and application. Data & knowledge engineering. 2021;134;
4. Neumayr B., Schuetz C. G., Kepler J. Multilevel modeling Encyclopedia of database systems. In: Liu L., Özsu (eds): encyclopedia of database systems. New York, USA: Springer; 2020. 4964 p.
5. Neumayr B. Multi-level modeling with M-objects and M-Relationships. PhD Thesis. Johannes Lepler Universität Linz. 2010.
6. Thalheim B. Entity-relationship modeling: foundation of database technology. Berlin: Springer-Verlag; 2000. 640 p.
7. Gonzalez-Perez C., Henderson-Sellers B. A powertype-based metamodeling framework. Software and systems modeling. 2006; 5(1):72-90.
8. Guerra E., de Lara J. Automated analysis of integrity constraints in multi-level models. Data & knowledge engineering. 2017; 107:1-23.
9. Jacome-Guerrero S. P., de Lara J. TOTEM: Reconciling multi-level modelling with standard two-level modelling. Computer Standards & Interfaces. 2020; 69
10. Macías F., Rutle A., Stolz M. Ecore: combining the best of fixed-level and multilevel metamodeling. In: Grossmann G., Clark T., Atkinson C. (eds) 3rd International workshop on multi-level modelling, MULTI 2016; CEUR-WS, Saint-Malo; France ; 2016. P. 66-75.
11. Rodionov A., Tsoy G. Unveiling and conceptual-logical modeling of phase sequences in data engineering // Proceedings of 6th international conference on information technologies and high-performance computing, Khabarovsk, pp. 54-61.
12. Родионов А. Н. Типология и моделирующие профили процесс-сущностных взаимодействий *moving*-класса. Рамочная информационная модель мультимодальной маршрутной сети // Информационные технологии. 2023. т.29. №8. С. 391–405.
13. Родионов А. Н. Типология и моделирующие профили процесс-сущностных взаимодействий *moving*-класса: базовые типы, ассоциации, ограничения и подсхемы // Информационные технологии. 2023. т.29. №2. С. 72–83.

Родионов Александр Николаевич. Вычислительный центр Дальневосточного отделения Российской академии наук. Хабаровск, Россия. Ведущий научный сотрудник, доктор технических наук, доцент. Область научных интересов: теоретические основы организации систем баз данных. E-mail: alnk.rodionov@mail.ru

Modeling Controlled Sparsity of Databases' Entity-Object Subschemas

A. N. Rodionov

Computer Centre of Far-Eastern Branch of Russian Academy of Sciences, Khabarovsk, Russia

Abstract. In the process of creating information systems at the enterprise level, we always have to solve the same problem - to design a subschema, in the tables of which the lists of objects will be placed. Ideally, there should be one object table in the database, which will be used in all associations in which the objects participate. However, a wide range of requirements for the structural organization of objects lists, such as compactness, modifiability, semantic unambiguous identifiability, and a number of others, force the objects table to be partitioned into many local tables to accommodate in the latter objects with similar properties and identical appointments. The paper calls into question the dogma of compactness, which leads to an avalanche-like increase in the number of object's tables, and as an alternative, the concept of controlled sparsity is put forward. Following this concept allows database creators, at the discretion of the user, to combine into one object type the set of local compact types. We proposed the structural framework, which includes two interrelated blocks: the block of objects' types and meta-block. The later contains meta-types, meta-relationships, and meta-constraints that collectively provide and guarantee the correctness of objects' data placed in the database.

Keywords: compactness, controlled sparsity, semantic and physical domains, meta and object subschemas.

DOI 10.14357/20718632240405 EDN CPWUGK

References

1. Aktinson C., Kühne T. The essence of multilevel metamodelling. In.: Gogolla M., Kobryn C. (eds.) UML 2001. Berlin Heidelberg: Springer; 2001. p. 9-33.
2. Neumayr B., Schrefl M., Thalheim B. Modeling techniques for multi-level abstraction. In: Kaschek/Delcambre (eds.): The evolution of conceptual modeling. Berlin Heidelberg: Springer-Verlag; 2011. p. 68-92.
3. Fonseca C. M., Almeida J. P., Guizzardi G., Carvalho V. A. Multi-level conceptual modeling: theory, language and application. Data & knowledge engineering. 2021;134;
4. Neumayr B., Schuetz C. G., Kepler J. Multilevel modeling Encyclopedia of database systems. In: Liu L., Özsu (eds): encyclopedia of database systems. New York, USA: Springer; 2020. 4964 p.
5. Neumayr B. Multi-level modeling with M-objects and M-Relationships. PhD Thesis. Johannes Lepler Universität Linz. 2010.
6. Thalheim B. Entity-relationship modeling: foundation of database technology. Berlin: Springer-Verlag; 2000. 640 p.
7. Gonzalez-Perez C., Henderson-Sellers B. A powertype-based metamodeling framework. Software and systems modeling. 2006; 5(1):72-90.
8. Guerra E., de Lara J. Automated analysis of integrity constraints in multi-level models. Data & knowledge engineering. 2017; 107:1-23.
9. Jacome-Guerrero S. P., de Lara J. TOTEM: Reconciling multi-level modelling with standard two-level modelling. Computer Standards & Interfaces. 2020; 69.
10. Macías F., Rutle A., Stoltz Mult V. Ecore: combining the best of fixed-level and multilevel metamodeling. In: Grossmann G., Clark T.,Atkinson C. (eds) 3rd International workshop on multi-level modelling, MULTI 2016; CEUR-WS, Saint-Malo; France ; 2016. P. 66-75.
11. Rodionov A., Tsoy G. Unveiling and conceptual-logical modeling of phase sequences in data engineering // Proceedings of 6th international conference on information technologies and high-performance computing, 2021, Khabarovsk, pp. 54-61
12. Rodionov A. N. Typology and modeling profiles of *moving*-class process-entity interactions. Framework for multimodal route network. Information technology. 2023;29(8): 391–405 (In Russ). DOI: 10.17587/it.29
13. Rodionov A. N. Typology and modeling profiles of *moving*-class process-entity interactions: core types, relationships, constraints and subschemas. 2023;29(2): 72-83 (In Russ). DOI: 10.17587/it.29.

Rodionov Aleksandr N. Doctor of technical sciences, Computer Centre Of Far-Eastern Branch of Russian Academy of Sciences, Khabarovsk, 680000, Russia, E-mail: alnk.rodionov@mail.ru