УДК 004.652.8

В.М. Левыкин, В.А. Филатов, Н.В. Черненко

Метод синтеза единой структурной составляющей реляционной модели данных

Предложен метод структурной интеграции схем независимо спроектированных и функционирующих баз данных различных реляционных моделей в единую. Исследована реляционная модель и возможные варианты интеграции нескольких схем реляционных баз данных, описан метод синтеза единой структурной составляющей реляционной модели данных.

A method of structural integration of the schemes of the projected and functioning data bases of various relational models into a single one is suggested. A relational model and the possible variants of the integration of several schemes of relational databases are investigated. A method of the synthesis of a single structural component of the relational data model is described.

Розглянуто метод структурної інтеграції схем реляційних баз даних, що спроектовані та функціонують незалежно. Проведено дослідження реляційної моделі, можливі варіанти інтеграції кількох схем реляційних баз даних, запропоновано метод синтезу єдиної структурної складової реляційної моделі даних.

Введение. Ключевая роль в достижении успеха большинства компьютеризированных систем принадлежит программному обеспечению. В последнее десятилетие прикладные программы проделали путь от маленьких и сравнительно простых приложений из нескольких строк кода до очень больших и сложных приложений, состоящих из нескольких миллионов строк. Многие из этих приложений требовали постоянного сопровождения, включая исправление выявленных ошибок, реализацию новых требований пользователей, а также перенос программного обеспечения на новые или модернизированные вычислительные платформы. Усилия и ресурсы, затрачиваемые в настоящее время на сопровождение программного обеспечения, возрастают высокими темпами.

Основные проблемы проектирования информационных систем

Разработка и реализация многих современных крупных информационных проектов имеет, как правило, затяжной характер, их стоимость превосходит запланированную, а окончательный продукт получается ненадежным и сложным в использовании. Все это привело к ситуации, известной под названием «кризис программного обеспечения». Хотя первые упоминания о кризисе были в конце 80-х годов, даже спустя 30 лет ситуацию изменить полностью не удалось.

Некоторые причины общей проблемы проектирования сложных информационных систем состоят в следующем:

– разработка около 40% систем заканчивается неудачно или прекращается до завершения работ;

– рационально интегрировать интересы бизнеса и используемой информационной технологии удается не более чем в 25% систем;

– только 20–30% информационных систем отвечают всем критериям достижения успеха.

Основные неудачи при создании программного обеспечения вызваны отсутствием приемлемой методологии разработки, полной спецификации всех требований на этапе проектирования или достаточного разделения общего глобального проекта на отдельные компоненты, поддающиеся эффективному контролю и управлению.

В случае частичной реализации требований пользователей информационной системы (ИС) или изменения бизнес-процесса до такой степени, что система перестает отвечать требованиям пользователей, возможны несколько вариантов развития:

– разработка новой системы;

– модификация (развитие) существующей (унаследованной – *legacy system*) системы;

– реинжиниринг существующей (унаследованной – *legacy system*) системы.

Первый вариант наиболее простой и предпочтительный для разработчика, но этот путь менее всего удовлетворяет требованиям пользователей, так как потребуются затраты дополнительного времени и финансовых ресурсов, а также существуют риски разработки и риски потери накопленной информации за время существования эксплуатируемой ИС.

Реинжиниринг унаследованных информационных систем требует привлечения экспертов в области информационных систем и технологий, что соответственно приводит к крайней сложности таких работ.

В большинстве случаев существует такое мнение: проще разработать систему заново, нежели прибегнуть к ее реинжинирингу. Это связано с квалификацией специалистов, которых необходимо привлечь для проведения работ. Она должна быть достаточно высокой для решения комплекса задач проектирования и создания модифицированной информационной системы [1].

Постановка задачи

Цель проводимых исследований – разработка метода синтеза реляционной модели данных, полученной на основе интеграции моделей данных различных независимых предметных областей.

Важнейшую роль в реализации проекта ИС занимает информационное обеспечение, а именно внутримашинное информационное обеспечение – базы и банки данных. Сегодня подавляющее большинство ИС используют системы управления базами данных, в основе которых заложена реляционная модель хранения данных, предложенная Э.Ф. Коддом [2]. Теоретически данная модель описана более 40 лет назад, однако широкое практическое применение получила последние 15 лет. На данном этапе развития информационных технологий широко известен строгий математический аппарат, позволяющий описать структуру базы данных, операционную составляющую, алгоритмы поиска функциональных зависимостей и нормализации схем баз данных.

Существует классическое описание реляционной модели данных, выделяющее три основные функциональные компоненты:

– структурная компонента;

– ограничения целостности;

– операционная спецификация.

Структурная компонента реляционной модели данных (*SRM*) – *n*-арное отношение: $SRM = \{R, D, A, dom\}$, где R – множество имен отношений; D – множество доменов; A – множество имен атрибутов; dom – отображение из A в D.

Прежде чем перейти к разработке метода синтеза интегрированной схемы базы данных, кратко рассмотрим основные понятия реляционного подхода.

Любой элемент D_i множества D называется доменом и позволяет именовать бесконечное множество элементарных данных или значений. Для элемента A_i множества A определено множество значений атрибута, совпадающее с одним из доменов. В реляционной модели данных задается отображение $dom: A \to D$, пара $< A_i, dom(A_i) >$ называется атрибутом с именем A_i и областью значений $dom\ A_i = D_j$.

Выражение $R_i(A_i ... A_n)$, в котором все имена A_i различные, называется схемой отношения. Множество имен атрибутов в схеме отношения называется носителем отношения. Каждой схеме отношения R_i модель ставит в соответствие множество кортежей декартового произведения: $r_i = dom\ A_1 \times \ldots \times dom\ A_n$.

Схемой реляционной базы данных называется конечный набор схем отношений $B = \{R_1, \ldots, R_p\}$. Реляционной базой данных со схемой B называется множество реализаций схем $R_1, \ldots, R_p$ отношений $b = \{r_1, \ldots, r_p\}$.

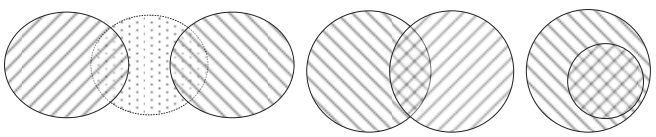
Рассмотрим информационное пространство, в границах которого функционируют множество информационных систем, спроектированных в разное время и решающих разные задачи информационной поддержки. В основе каждой ИС используется реляционная база данных и, соответственно, известна ее реляционная модель. Целью проводимых исследований рассматривается задача построения единой интегрированной реляционной модели базы данных, которая позволила бы по возможности объединить схемы различных БД без утраты их функцио-

нальности и тех данных, которые в них хранятся. В настоящее время методы и алгоритмы решения такого класса задач известны только для частных случаев и для определенных предметных областей. Однако единой методики и технологии для решения такого рода задач на формальном математическом уровне на сегодняшний день нет [3].

Разработка метода синтеза схемы реляционной модели данных

Как уже отмечалось, в информационном пространстве организации, учреждения, предприятия могут функционировать различные системы, спроектированные на основе реляционного подхода, предметные области которых произвольным образом могут соотноситься между собой.

Введем некоторые обозначения: пусть S_n – схема базы данных ИС, обеспечивающая информационную поддержку n-й предметной области. Рассмотрим возможные случаи взаимного влияния различных ИС на примере двух систем, представленных на рисунке.



Вариант 1 Вариант 2 Вариант 3

Варианты интеграции двух предметных областей

Вариант 1. Предметные области не пересекаются, схемы БД не содержат данных об одном и том же объекте реального мира. В случае принятия решения об интеграции таких схем БД, вводится дополнительная подсхема ($S_{\text{общ}}$), позволяющая связать представленные схемы в единую.

Вариант 2. Предметные области частично пересекаются, некоторые отношения БД содержат данные об одном и том же объекте двух предметных областей. Это наиболее часто встречающийся случай.

Вариант 3. Предметные области пересекаются, все отношения одной БД содержат данные о том же объекте, что и отношения другой (все или некоторые).

Частным случаем третьего варианта может выступать полное соответствие $S_{\text{ПР1}}\, S_{\text{ПР2}}$. Такое возможно, например, при внедрении нового варианта системы [4]. В таком случае вопрос об интеграции данных особо важен.

Для интеграции различных схем баз данных разнородных ИС необходимо преобразовать их схемы в одну, с сохранением данных. Для этого предлагается метод синтеза интегрированной реляционной модели данных: $SRM_s = (R_s, D_s, A_s, dom_s)$ из множества отдельных схем баз данных – $SRM_1 = (R_1, D_1, A_1, dom_1)$, $SRM_2 = (R_2, D_2, A_2, dom_2), \ldots, SRM_n = (R_n, D_n, A_n, dom_n)$.

Данный метод разработан для второго и третьего вариантов. Метод позволяет синтезировать структуру интегрированной реляционной модели данных.

Этапы реализации метода:

• Определить степень нормализации существующих схем БД.

• Если необходимо, привести их ко второй нормальной форме, применяя известные алгоритмы нормализации.

• Определить множества отношений $R_1, R_2, \ldots, R_N$, установить соответствие $R_1 \leftrightarrow R_2$, $R_1 \leftrightarrow \leftrightarrow R_n$, $R_2 \leftrightarrow R_n, \ldots$ (какие из отношений пересекаются или могут расцениваться как тождественные), если необходимо, произвести переименование имен отношений. Сформировать множество $R_s = R_1 \cup R_2 \cup \ldots \cup R_n$. Определение полного соответствия происходит на основании экспертных оценок. Эксперт должен хорошо знать предметную область, существующий бизнес-процесс и схемы БД, с которыми придется работать. Необходимо определить семантическую нагрузку имен отношений и, при необходимости, провести их транслитерацию.

• Определить множества доменов $D_1, D_2, \ldots, D_N$ и установить соответствие $D_1 \leftrightarrow D_2$, $D_1 \leftrightarrow D_n, D_2 \leftrightarrow D_n, \ldots$. Определить возможные сторонние элементы D_{is} множеств $D_1, D_2, \ldots, D_N$ для их дальнейшего исключения из D_s. На данном этапе также необходимо определить семантическую нагрузку множеств доменов $D_1, D_2, \ldots D_N$, провести их транслитерацию.

• Определить «устаревшие» элементы множеств D_1, D_2, … D_N (какие не могут быть в текущем бизнес-процессе, но присутствовали в прошлом), преобразовать их в множество D^* – в SRM_s множество доменов, определяющее «историческую» составляющую во множествах D_1, D_2, … D_N.

Выделить единое множество $D^{i'} = D_1 \cup \cup D_2 \cup ... \cup D_n \mid D^* \mid D_{is}$.

Результирующее множество доменов состоит из объединения множества фактических доменов всех схем и множества «исторических» доменов $D_s = D^i \cup D^*$.

• Провести оценку множества схем отношений R_1 и R_2, определить множества атрибутов A_1, A_2, …, A_N, с учетом проведенного анализа множеств доменов, при необходимости, исключить избыточные атрибуты в ходе формирования итогового множества

$$A_s = A_1 \cup A_2 \cup ... \cup A_n \mid A_{is}.$$

На данном этапе необходимо определить семантическую нагрузку множеств имен атрибутов A_1, A_2, …, A_N, и, при необходимости, провести их транслитерацию.

• Провести оценку множества схем отношений, определить возможные области значений атрибутов: $dom_1 : A_1 \to D_1$, $dom_2 : A_2 \to D_2$, …, $dom_n : A_n \to D_n$ с учетом сформированных множеств D^*, D_s, A_s. На основании проведенного анализа сформировать dom_s.

• Полученные результаты преобразовать к следующему виду: $SRM_s = (R_s, D_s, A_s, dom_s)$.

Заключение. В статье предложен метод структурной интеграции схем независимо спроектированных и функционирующих баз данных. Для полного преобразования нескольких различных реляционных моделей данных в единую, рассмотрена операционная составляющая и компоненты ограничений целостности. В рассмотренном случае исследована технология преобразования нескольких моделей в одну, операционная составляющая не рассматривалась в виду того, что все операции синтеза ограничены реляционной моделью данных. Синтезированная реляционная модель данных будет обладать тем же набором операций, что и каждая из исходных моделей. Примерами таких операций могут выступать базовые операции реляционной алгебры и производные от них.

Предложенный метод на формальном уровне позволяет описать процесс синтеза структурной составляющей реляционной модели базы данных из множества отдельно существующих схем баз данных, а также минимизировать затраты на проектирование общей схемы БД, обеспечить хранение данных и манипулирование ими для всех функциональных задач в рамках интегрированной ИС. Использование интегрированной ИС снижает риски представления неактуальной, избыточной информации, а также стоимость ее поддержки и сопровождения.

1. *Stein R.E.* Re-Engineering the Manufacturing System: Applying the Theory of Constraints // Marcel Dekker, Inc. – 2003. – 325 p.
2. *Codd E.F.* A Relational Model of Data for LargeShared Data Banks // Communications of the ACM. – 1983. – **38**, № 1. – P. 17–36.
3. *Дейт К.* Введение в системы баз данных. – М.: Изд. дом «Вильямс», 2001. – 1072 с.
4. *Пономаренко Л.А.*, *Танянский С.С.*, *Филатов В.А.* Построение оптимальной последовательности соединения отношений в запросах реляционной базы данных // Системні дослідження та інформаційні технології. – 2003. – № 2. – С. 53–58.

Поступила 08.02.2011

Тел. для справок: (057) 702-1350, 66-0586, 702-1432, 702-1736 (Харьков)
E-mail: *Filatov_val@ukr.net*