Юрченко Василий Васильевич
Технологический институт федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.
E-mail: esqayr@inbox.ru.
347939, г. Таганрог, ул. Петровская 9, кв.81.
Тел.: 88634360484
Кафедра математического обеспечения и применения ЭВМ; студент.

Yurchenko Vasily Vasilievich
Taganrog Institute of Technology – Federal State-Owned Educational Establishment of Higher Vocational Education "Southern Federal University"
E-mail: esqayr@inbox.ru.
11, Petrovskay, Taganrog, 347900, Russia.
Phone: 88634360484.
Department of Computer Software; student.

УДК 519.7 : 007 + 06

А.В. Белых, С.М. Ковалев, О.В. Ольховик

СОКРАЩЕНИЕ ИЗБЫТОЧНОТИ СХЕМЫ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ БАЗ ДАННЫХ*

В работе рассматриваются вопросы, связанные с сокращением избыточности схемы объектно-ориентированных баз данных. Представленное решение позволяет помимо сокращения избыточности схемы сократить и избыточность данных, а так же отразить идеи, заложенные при проектировании схемы ООБД, минимальным набором визуальных конструкций.

Объектно-ориентированные базы данных; нормализация; дерево наследования; алгоритм нормализации; сокращение избыточности схемы ООБД.

A.V. Belykh, S.M. Kovalev, O.V. Olhovik

THE REDUCTION OF REDUNDANCY OF THE SCHEME OF OBJECT-ORIENTED DATABASES

In work the questions connected with reduction of redundancy of the scheme of object-oriented databases are considered. The presented decision allows to reduce besides reduction of redundancy of the scheme and redundancy of the data and as to reflect the ideas put at designing of scheme OODB, the minimum set of visual designs.

Object-oriented databases, normalization; inheritance tree; normalization algorithm; reduction of redundancy of the scheme of OODB.

Введение. При проектировании схемы ООБД проектировщик редко может сразу построить схему, которая бы содержала минимальное количество элементов, то есть спроектировать не перегруженную схему. Для устранения избыточности ему требуется пройти не одну итерацию процесса проектирования, и при этом не всегда удается устранить все избыточные элементы в схеме, сохранив при этом общую идею. Однако наличие таких элементов в схеме ООБД зависит не только от проектировщика, но и от того визуального языка, на котором ведется проекти-

* Работа выполнена при поддержке: РФФИ (проекты № 07-01-00075, № 09-07-00192).

рование. Например, в работе [1] обращено внимание на то, что популярный визуальный язык UML обладает громоздкостью, что в свою очередь является причиной создания излишних конструкций на схеме при проектировании. Есть в UML также языковые конструкции, которые просто сложны для восприятия. Таким образом, сегодня актуальным является решение проблемы сокращения избыточности и сложности схемы ООБД.

С одной стороны для устранения избыточности схемы требуется наличие метода проектирования позволяющего ее сократить, а с другой стороны требуется наличие визуального языка проектирования с минимальной элементной базой, который позволил бы отобразить идею проектировщика минимальным набором визуальных конструкций.

Для решения вышеописанной проблемы вполне применим разработанный метод нормализации схемы ООБД, который представлен в [2]. Этот метод является полностью формализованным и основывается на N-модели данных [3]. Предложенные в работе [2] правила и алгоритмы, производят нормализацию, как всей схемы, так и отдельно взятой группы классов и отношений между ними. За счет этого процесс ОО нормализации становится более динамичным и позволяет производить нормализацию только для подвергшихся изменениям классов и отношений между ними. При этом алгоритм является простым как для понимания, так и для реализации в реальной среде проектирования информационных систем.

Опишем действия проектировщика для выработки входных данных для этого алгоритма:

1) описать классы и терминальные атрибуты на этих классах;

2) для каждого описанного класса:

- Применить Правило 1 метода нормализации ООБД [2], благодаря которому функционально зависимые атрибуты выделяются в отдельный класс.
- В каждом появившемся классе, следует определить принадлежность атрибутов к классу. Принадлежность определяется только для атрибутов, не вошедших в новый класс. Принадлежность может быть следующая: атрибут не определен – не имеет определенного значения в классе, атрибут определен – имеет определенное значение в классе, терминальный атрибут.

Таким образом, проектировщик описывает классы и определяет принадлежность атрибутов к ним. Эти данные являются входными для алгоритма нормализации. Для реализации новых требований следует дополнить метод нормализации из работы [2] Правилом 4.

Правило 4. Если терминальные атрибуты класса C1 могут быть описаны в классе C2, а терминальные атрибуты в классе C2 могут быть описаны в классе C1, то между данными классами может быть установлена наследственная связь.

Исходя из этого, Алгоритм 1 [2] следует изменить следующим образом:

```
B={};
For i:=1 to Length(T) do
 Begin
 {Последний элемент множества(массива) T или одноэлементное множество не рассматриваются}
  If i=Length(T) Then Break;
 {рассмотрим каждый последующий элемент множества T - (T[j]) в сравнении с элементом  текущим T[i]}
  For j:=i+1 to Length(T) do
   Begin
```

{Если текущий рассматриваемый элемент T[i], разложим на общую часть T[j] и на остаток (T[i]\T[j]), то он далее не рассматривается, а замещается остатком (T[i]\T[j]) и ребром, ссылающимся на общую часть T[j]. Т.к. только общая часть может рассматриваться как еще не делимая. Другими словами T[j] входит в T[i]}

```
If T[j] включено в T[i] Then
 Begin
  T[i]:=T[i]\T[j];
   {Если для T[j] и T[j] выполняется Правило 4 }
  If (T[j] определено в T[i])and(T[i] определено в T[j]) Then
     {Строится наследственная связь между T[j] и T[j] }
    B=B+([i],[j])
  Else
     {Строится отношение между T[j] и T[j] }
    R=r+{r[i], r[j], OneToOne}
  Break;
 End;
End;
End;
```

Рассмотрим, как алгоритм нормализации ООБД влияет на сокращение избыточности схемы на следующем примере.

Пусть проектировщик выполнил вышеописанные действия и определил следующие входные данные для алгоритма нормализации ООБД, т.е. классы и их терминальные атрибуты:

- АВТОМОБИЛЬ (Наименование марки, Наименование модели, Производитель, Класс, Год выпуска, Номер партии, Регистрационный номер, Коробка передач, Мощность, Цвет, Тип кузова, Усилитель руля);
- МОТОЦИКЛ (Наименование марки, Наименование модели, Производитель, Класс, Год выпуска, Номер партии, Регистрационный номер, Коробка передач, Мощность, Цвет, Высота седла);
- ПАРТИЯ (Наименование марки, Наименование модели, Номер партии, Производитель, Класс, Год выпуска);
- МОДЕЛЬ (Наименование марки, Наименование модели, Класс);

При этом будем учитывать, что для каждой пары классов выполняется Правило 4, кроме пары АВТОМОБИЛЬ и МОТОЦИКЛ.

Для краткости обозначим классы как: T1-АВТОМОБИЛЬ, T2-МОТОЦИКЛ, T3-ПАРТИЯ, T4-МОДЕЛЬ. Атрибуты классов при этом обозначим следующим образом: 1-Наименование марки, 2-Наименование модели, 3-Производитель, 4-Класс, 5-Год выпуска, 6-Номер партии, 7-Регистрационный номер, 8-Коробка передач, 9-Мощность, 10-Цвет, 11-Тип кузова, 12-Усилитель руля, 13-Высота седла. Тогда классы можно описать как:

- T1={1,2,3,4,5,6,7,8,9,10,11,12};
- T2={1,2,3,4,5,6,7,8,9,10,13};
- T3={1,2,3,4,5,6};
- T4={1,2,4}.

Для построения нормализованной схемы по данным классам необходимы только алгоритмы, связанные с нормализацией дерева наследования классов, так как нормализация отношений между классами не требуется. Не рассматривая подробно каждую отдельную итерацию, из которых состоит алгоритм нормализации дерева наследования классов, представим только их входные и выходные параметры.

АЛГОРИТМ 1. Нахождение ребер и элементов дерева наследования.

ВХОД: T={T1,T2,T3,T4}, где T1={1,2,3,4,5,6,7,8,9,10,11,12}, T2={1,2,3,4,5,6,7,8,9,10,13}, T3={1,2,3,4,5,6}, T4={1,2,4}.

ВЫХОД:

- T={{7,8,9,10,11,12}, {7,8,9,10,13},{3,5,6},T4}, где T4={1,2,4};
- B={(T1,T3), (T2,T3), (T3,T4)}.

АЛГОРИТМ 2. Построение дерева наследования;

ВХОД:

- T={{7,8,9,10,11,12}, {7,8,9,10,13},{3,5,6},T4}, где T4={1,2,4};
- B={(T1,T3), (T2,T3), (T3,T4)}.

ВЫХОД: Дерево наследования классов. Получившееся дерево наследования классов показано на рис. 1, где в прямоугольниках указаны описываемые атрибуты классов.

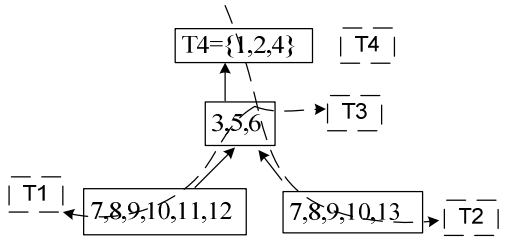

Рис. 1. Дерево наследования классов

Для упрощения восприятия результата представим дерево наследования классов на визуальном языке NVL предложенном в работе [4]. Данный язык обладает простым и удобным синтаксисом, который включает в себя всего пять визуальных элементов, что делает язык легким в освоении и позволяет оперировать сложными структурами и большими массивами данных посредством их визуальных аналогов. Визуальные элементы NVL аналогичны элементам из стандартов UML 2.2 [5] и IDEF1X [6], и поэтому интуитивно понятны не только разработчикам ИС, но и опытным пользователям. Теоретические основы NVL описаны в статье [3]. Декларативная составляющая языка представлена в работе [7].

Результат алгоритма нормализации, представленный на языке NVL, показан на рис. 2.

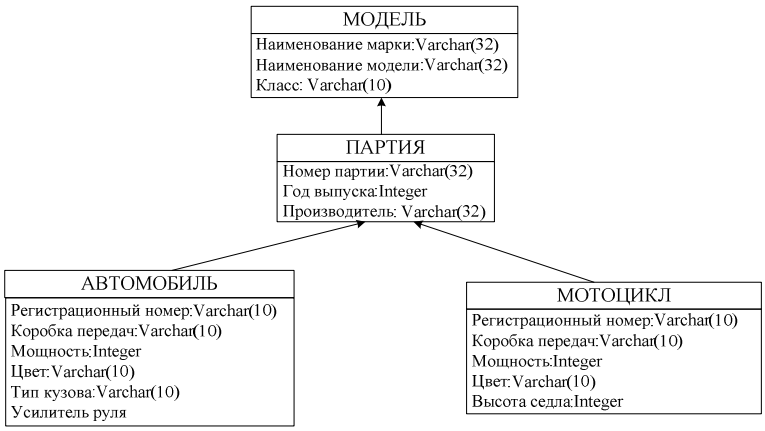

Рис. 2. Схема наследования классов на языке NVL

Анализируя полученную схему, можно отметить, что алгоритм нормализации позволил устранить избыточность за счет сокращения количества атрибутов на схеме относительно входных данных. Схема для данного примера, конечно, могла выглядеть иначе, но схема (см. рис. 2) является наименее избыточной из тех, который могли быть созданы исходя из входных данных. Полученная схема обладает наглядностью, компактностью и простотой. Такая схема позволяет сконцентрировать факты в одном месте. Так же алгоритм нормализации за счет декомпозиции позволил построить дерево наследование классов с минимальным количеством атрибутов. При этом алгоритм помимо сокращения избыточности атрибутов позволяет сократить в общем случае избыточность классов и данных соответственно, что в свою очередь сказывается на общем восприятии схемы.

Таким образом, проектировщику предоставляется возможность использовать алгоритм нормализации ООБД [2] для сокращения избыточности схемы и данных. Причем применяя в дополнении к нему предложенный визуальный язык проектирования NVL, который не перегружен излишними конструкциями и прост для восприятия, в отличии от аналогичных языков проектирования, проектировщик имеет возможность представлять схемы минимальным количеством визуальных конструкций. Все это дает возможность проектировщику в кратчайший срок строить схемы ООБД с наименьшей избыточностью, как данных, так и самой схемы.

Заключение. Авторы признательны специалистам, высказавшим замечания по данной работе, надеются на сотрудничество в решении еще неразработанных вопросов и будут благодарны за конструктивную критику.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Robert B. France, Sodipto Ghosh, Trung Dinh-Trong*. Model-Driven Development Using UML 2.0: Promises and Pitfalls // Computer. – 2006. – №2. – С. 59-66.
2. *Белых А.В., Ковалев С.М., Ольховик О.В.* Нормализация ООБД на основе N-модели данных // Известия ВогГТУ. Серия: Актуальные проблемы управления, вычислительной техники и информатики в технических системах. – 2009. – №6. – С. 70-78.
3. *Ольховик О.В., Белых А.В.* N-Модель данных // Известия ЮФУ. Технические науки. Тематический выпуск "Интеллектуальные САПР". – 2009. – № 4 (93). – С. 181-188.
4. *Белых А.В., Ковалев С.М., Ольховик О.В.* Визуальный язык проектирования // Вестник ДГТУ. 2009. Т.9. – №4 (43). – С. 381-390.
5. *OMG.* Unified Modeling Language: Superstructure, version 2.2 // OMG.ORG: Портал The Object Management Group URL: http://www.omg.org/cgi-bin/doc?formal/09-02-02.pdf (дата обращения: 28.09.2009).
6. Integration definition for information modeling (IDEF1X) // IDEF.COM: Портал Knowledge Based Systems, Inc. URL: http://www.idef.com/pdf/Idef1x.pdf (дата обращения: 10.09.2009).

Белых Александр Валерьевич
С-Кав. РДОП «Севкавэкспресс» СП ФПД - филиал ОАО «РЖД».
E-mail: white@donses.ru.
г. Ростов-на-Дону, ул. Зорге 29/1 кв. 24.
Тел.: 88632252502; 89094335252.
Аспирант РГУПС, инженер сектора информатизации.

Belykh Alexander Valeryevich
S-Kav. RDOP «Sevkavexpress» organization department FPD – branch of Plc «RZD».
E-mail: white@donses.ru.
24, Apt., 29/1, Zorge street, Rostov-on-Don, Russia.
Phone: 88632252502; 89094335252.
Graduate student of RGUPS, the engineer of sector of informatization.

Ольховик Олег Владимирович
ИВЦ ДГТУ.
E-mail: olvick@spark-mail.ru.
г. Ростов-на-Дону, ул. Нариманова 78, кв.177.
Тел.: 88632421840.
Заведующий сектором внедрения информационных технологий в учебный процесс.

Olhovik Oleg Vladimirovich
Data-processing centre DGTU.
E-mail: olvick@spark-mail.ru.
177, Apt., 78, Narimanova street, Rostov-on-Don, Russia.
Phone: 88632421840.
Technical, manager of sector of introduction of an information technology in educational process of.

Ковалев Сергей Михайлович
РГУПС.
E-mail: white@donses.ru.
344038, Ростов-на-Дону, пл. Ростовского Стрелкового Полка Народного Ополчения 2.
Профессор кафедры Автоматика и телемеханика на железнодорожном транспорте.

Kovalev Sergey Mihaylovich
RGUPS.
E-mail: white@donses.ru.
2, 344038, area The Rostov Shooting Regiment of the National Home guard, Rostov-on-Don, Russia.
Professor of chairs Automatics and Telemechanics on railway transportation.