

УДК 681

КОНЦЕПЦИЯ ОБЪЕКТНО-АССОЦИАТИВНОЙ БАЗЫ ДАННЫХ

*канд. техн. наук Д.О. ГЛУХОВ, А.В. ПРАНОВИЧ
(Полоцкий государственный университет)*

Рассмотрена концепция объектно-ассоциативной базы данных. Показано, что такая модель данных идеально подходит для создания базы данных в условиях начальной неопределенности относительно многообразия объектов в системе, полных спецификаций объектов и возможных связей между объектами.

Реальность никогда не может быть вписана в рамки некоторой упрощенной схемы, она всегда богаче, всегда шире. Как бы мы не обобщали эту схему, всегда возникает ситуация, которая перечеркивает всю нашу работу. Эта предполагаемая неопределенность никогда еще не была так близка к тому, чтобы стать ключевым аспектом концепции баз данных.

Чтобы уловить тенденцию развития теории баз данных, остановимся на ключевых моментах истории.

Экскурс в историю

Реляционный подход, предложенный Эдом Коддом в 1964 году и развитый в работах Дейта, Мейера, Ульмана и других известных теоретиков реляционной теории, стал доминировать в сознании специалистов по информационным технологиям в силу его высокой прагматичности. Реляционный подход опирается на понятие схемы реляционной базы данных (РБД), изначально жестко заданной, неизменной на стадии эксплуатации. Информация об объектах, помещаемых в базу данных, обрезается до вмещаемой в схему части, конечно, наиболее существенной, необходимой и достаточной для конкретной прикладной задачи. Как будет показано далее, последний тезис справедлив далеко не всегда. Разнообразие объектов определяется множеством отношений - таблиц базы данных. Оно фиксированно и неизменно. Связи между объектами разных типов также неизменны и определяются схемой [3, 7].

Объектно-ориентированный подход (ООП), разработанный в конце 80-х - начале 90-х годов XX века такими известными теоретиками, как Г. Буч, Б. Страустрап, Э. Йордон, С. Шлеер, С. Меллор, Б. Мейер, и другими никак не снимает проблемы ограничения модели, заданной проектировщиком схемой. Только на смену отношению связывания по ключу приходит несколько новых отношений, в том числе: наследование, агрегация, ассоциация, инкапсуляция, использование. При этом база данных к тому же должна вписываться в изначально заданную схему, только уже схему объектно-ориентированной базы данных (ООБД) [2, 4, 5].

Объектно-реляционный подход (ОРП) сложился как попытка усовершенствовать реляционный подход и применить его для манипулирования сложноструктурированными данными [9].

Основоположник данного подхода - известный американский ученый Майкл Стоунбрейкер. Реализацией данного подхода явилась система управления базами данных (СУБД) Postgres университета Беркли штата Калифорния. В настоящее время на рынке конкурируют такие СУБД, как Oracle8, IBM DB2 Universal Database и Informix Universal Server, имеющие объектные расширения. Однако пока не выработан единый стандарт таких расширений. Как результат - расслоение серверного программного обеспечения (ПО) на сервер баз данных и сервер приложений.

Большинство ведущих поставщиков РСУБД активно продвигают свое видение ОРСУБД. Доминирует представление, что ОРСУБД - это попытка адаптировать реляционное хранилище данных под задачу хранения объектно-ориентированной системы, или, что то же, дать возможность посредством оперирования объектами, взаимодействовать с реляционной СУБД.

Для нас важно, что проблему неизменности абстрактной схемы не решает и данный подход.

Ассоциативная модель данных (АМД). Доктор Саймон Вильямс в 2000 году ввел понятие ассоциативной модели данных [1].

В такой модели атомарные значения выступают в качестве элементов, которые имеют связи с другими элементами. По сути, формируя сеть ассоциаций, сделан был первый шаг от уникампетентного программирования к омникмпетентному. Особенностью АМД является его направленность на обработку языковых конструкций. Сущности в ассоциативной модели остаются самодостаточными, они не классифицируются, не группируются, не подвергаются какой-либо другой абстракции. Передавая эту логическую надстройку, по-видимому, на уровень программирования. То есть понятие абстрактной схемы данных было Саймоном уничтожено до основания. Хотя именно абстрактная схема являлась тем инструментом, который позволил реляционному подходу владеть миром.

Философия двойственности. В данной работе проводится важная философская мысль о первичности объектов и вторичности отношений, классов, групп и любых других абстракций. Это сразу дает нам почву для размышления.

Если исходить из вторичности абстрактного уровня описания, то нужно предположить существование универсального репозитория, в который можно поместить любой объект, характеризуемый любым набором свойств и имеющий произвольные связи с другими объектами. В этом случае классификация есть динамический процесс, позволяющий на универсальном множестве объектов сформировать, возможно, и пересекающиеся подмножества объектов разных классов. Что в существующих парадигмах просто нереализуемо. Хотя естественно и не противоречит строгой логике.

Краткие итоги истории

1. Реляционный подход предполагает наличие конечного разнообразия неизменных по своей структуре сущностей.

2. Объектно-ориентированный подход обогащает конечное разнообразие дополнительными отношениями.

3. Ассоциативный подход претендует на решение проблемы омникомпетентного проектирования и допускает неограниченное многообразие сущностей, но вообще снимает задачу абстрактного моделирования предметной области.

Идея об ограничении при формировании абстрактного уровня описания проблемной области не нова. Она активно развивается в публикациях [8, 9].

Авторы подчеркивают то, что реляционная модель данных не допускает естественного представления данных сложной структуры. Многочисленные важные связи приходится реализовывать на уровне бизнес-логики.

Преимущества неизменяемой схемы:

- во-первых, имеется возможность проанализировать статистику обращений и построить индексы, ускоряющие работу сервера баз данных на порядки;

- во-вторых, декларирование схемы как объекта проектирования привело к появлению значительного количества совершенных во многих отношениях средств проектирования. *Case-средства* возвели схему базы данных в ранг непререкаемого стандарта при проектировании баз данных. С появлением UML был сформирован международный стандарт в области проектирования объектно-ориентированных систем;

- в-третьих, соответственно, системы приобрели общий язык описания и возможность унификации.

Однако схема базы данных, став тем инструментом, который позволил эффективно проектировать базу данных, явилась для нее и самым мощным ограничением.

Ограничения:

- всегда появиться объект, который обладает свойством, важным, но отсутствующим в схеме, о котором информацию придется записать в ежедневник;

- всегда потребуется построить такой отчет, который непонятно как строить по имеющейся схеме данных;

- всегда возникнут такие связи, которые нельзя зафиксировать в базе данных заданной схемы.

Поэтому, предлагаемый нами **объектно-ассоциативный подход** (ОАП) исходит из начальной неопределенности относительно многообразия вариаций объектов, предоставляет инструмент формирования абстракций в ассоциативной среде, и притом допускает построение разных абстрактных моделей для одной и той же области данных, выделяемых по различным признакам классификации. В этом концептуальное отличие предлагаемого подхода от известных теорий.

Яркие примеры ограниченности известных подходов

Критике реляционных принципов, как и объектно-ориентированного подхода, посвящено много публикаций [6, 8, 9, 10, 11, 12].

Первые работы, в которых отмечались эти и ряд других недостатков, появились уже в начале 80-х годов. Одна из наиболее ярких статей «Системы баз данных третьего поколения: Манифест» была опубликована в 1990 г.

Вслед за первым манифестом последовали: Манифест систем объектно-ориентированных баз данных (М. Аткинсон, Ф. Бансилон, Д. Де Витт, К. Диттрих, Д. Майер, С. Здоник // СУБД. - № 4. - 1995) и Третий манифест (Х. Дарвин, К. Дэйт // СУБД. - № 1. - 1996).

В большинстве случаев отмечается, что сами принципы таковы, что невозможно создать средство полностью их удовлетворяющее.

Например, второй принцип реляционности утверждает отсутствие упорядоченности атрибутов. Тем не менее в языке SQL очевидно, что упорядоченность присутствует.

Пример 1

```
INSERT INTO 'tablename' VALUES (12, 'Задвижка 2', 34, '12.05.1998');
```

иначе, как бы мы узнали, какое значение, какому атрибуту устанавливается в соответствие.

Даже первый, основной, принцип реляционности не реализуется ни одной СУБД.

Пример 2

```
SELECT 'name' FROM 'table name';
```

Утверждается, что реляционная алгебра замкнута относительно реляционного отношения, однако в примере тривиальный запрос порождает нереляционное отношение, отношение, не гарантирующее отсутствие кортежей дубликатов. Если бы основной принцип выполнялся, стало бы невозможно выполнять групповые операции, формировать сложные запросы связывания.

Пример 3

Любое отношение можно свести к реляционному отношению, добавив некоторый уникальный счетчик.

Но тогда в чем же смысл этого принципа? Смысл прост. Это самый простой способ связывания - связывание по ключу. Проще придумать невозможно.

Кстати, Саймон Вильямс в первую очередь разрешил дублирование данных в ассоциативной модели!

В объектно-ориентированной базе данных ситуация выглядит несколько иначе. Здесь уже присутствуют отношения между классами. И предполагается отношение агрегации между объектами.

Но проблема возникает на стадии проектирования с выявлением абстракций.

Пример 4

Атомарность значений атрибутов - четвертый краеугольный камень реляционной теории - становится препятствием для реализации современных баз данных для геоинформационных систем (ГИС). баз мультимедиа данных, систем искусственного интеллекта. Исчезает роль СУБД как универсального хранилища данных [10, 11, 12].

Пример 5

При разработке объектно-ориентированной системы проектируется система классов, которая не даст сохранить избыточную информацию об объекте, информацию, которая не вписывается в предложенную для него абстракцию, или какие-либо связи придется выявлять проверкой массы условий на уровне бизнес-логики. Кроме данных ограничений, часто указывается на то, что чисто объектно-ориентированная модель данных лишена эффективных, высокопроизводительных средств манипулирования данными и формирования запросов.

Итак, принципы реляционного и объектно-ориентированного подхода ограничивают в основном именно проектируемую схему базы данных.

Принципы объектно-ассоциативного подхода

1. Система состоит из объектов.
2. С объектами могут быть связаны другие объекты отношением агрегации.
3. Связи между объектами рассматриваются двух типов:
 - а) агрегация: родительский - дочерний;
 - б) ссылка: родительский объект - дочерний объект-ссылка - сторонний объект. Причем объект-ссылка связан с родительским объектом отношением агрегации.
4. Объект может играть одну из трех ролей: ссылка, значение или корневой объект.
5. Каждый объект значение или ссылка имеют интерпретации. Интерпретация объекта есть смысловое содержание представленного объектом значения или ссылки. В данном случае тип описывает не только способ хранения и отображения данных, но и несет оттенок назначения для предметной области. Интерпретация объекта-ссылки суть роль или тип связи.
6. Существует неопределенность относительно того, каким перечнем свойств и связей обладают объекты, какое разнообразие объектов предполагается в системе.
7. На множестве объектов могут динамически формулироваться абстракции - классы.
8. Абстрактная модель является способом доступа и манипулирования данными.
9. Класс есть некоторое подмножество объектов, выделяемое по некоторому признаку. Причем признаком, по которому формируется класс, может выступать как наличие некоторых свойств или связей по их интерпретациям, так и условия, формулируемые на значениях свойств или связей.
10. Объект может принадлежать произвольному количеству классов и иных абстракций.
11. Существует неопределенность относительно того, сколько классов представлено в системе, каковы отношения между этими классами.

Формализм объектно-ассоциативной модели данных

Базовые множества. Универсальным способом представления значений будем рассматривать строковое представление. Для алфавита char множество допустимых строк (цепочек символов) будем обозначать как char^* . Такая нотация употребляется в теории формальных грамматик и, кроме этого, хорошо понятна программистам, работающим на языке C++.

Для определения доступа к элементам множеств будем использовать точечный оператор:

$$o.a : a \in o.$$

Обозначим множество объектов O .

Определение объекта. Каждый объект универсального множества объектов является сложным по своей структуре и для реализации сформулированных принципов объектно-ассоциативного подхода может быть представлен в следующей конфигурации:

$$o = \langle \text{value}, \text{reference}, \text{interpretation}, \text{parent} \rangle.$$

Ведем соответствующие операторы отношений.

Уровни представления ассоциативной системы объектов

Будем рассматривать ассоциативную систему объектов на трех уровнях представления:

- интерфейсное представление в виде совокупности подмножеств разметки;
- объектное представление с отношением агрегации на множестве объектов;
- смысловое представление или интерпретация в пространстве смыслов EXPLAIN.

Определение агрегации. Объекты могут связываться отношением агрегации.

Если объект o_1 входит в объект o_2 , то

$$o_1.\text{parent} = o_2,$$

или будем записывать

$$\begin{aligned} o_2 &= \text{parentOf}(o_1), \\ o_1 &\in \text{childrenOf}(o_2). \end{aligned}$$

Определение интерпретации. Смысловое окрашивание объекта в рамках некоторой предметной области есть его интерпретация. Множество допустимых интерпретаций – EXPLAIN. Одним и тем же смыслом могут обладать разные объекты. Объект не может быть двусмысленным. Интерпретация свойственна всем объектам и

$$o.\text{interpretation} \in \text{char}^*.$$

Введем оператор интерпретации:

$$\text{explain}(o) = o.\text{interpretation}.$$

Определение подмножества разметки. Объекты могут быть сгруппированы в подмножества разметки. Подмножеством разметки мы будем называть некоторое произвольное подмножество объектов, сформированное по некоторому смысловому признаку. Мы можем определять подмножества по своему усмотрению. Примером подмножества разметки будет являться набор объектов значений и ссылок, на наш взгляд, необходимый для определения абстракции.

В зависимости от признака выделения подмножества можно рассмотреть 3 типа подмножеств разметки:

1. Некоторое произвольное подмножество L .
2. Подмножество объектов, объединенных тем, что они все являются агрегированными в один и тот же родительский объект $o - L(o)$.
3. Подмножество объектов, ассоциированных с некоторой интерпретацией $l(e), e \in \text{EXPLAIN}$.

Подмножества разметки могут пересекаться, быть равными, не пересекаться, но их объединение всегда равно универсальному множеству объектов (рис. 1).

Иными словами, любой объект всегда принадлежит некоторому множеству разметки, но он также может принадлежать и нескольким различным подмножествам одновременно

$$\bigcup_i L_i = O.$$

Подмножества разметки выступают в качестве тех «страниц», через которые мы воспринимаем систему объектов. Они выступают интерфейсными составляющими системы.

Поскольку разметка системы объектов допускает формирование новых групп объектов на уже существующем множестве данных, то именно разметка может служить средством формулировки абстракций.

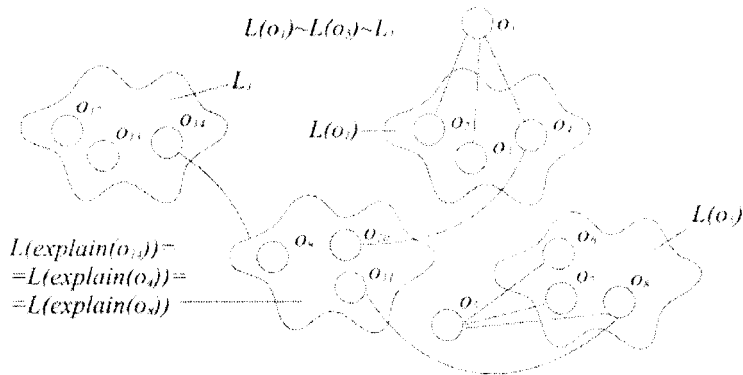


Рис. 1. Иллюстрация отношения подобия на подмножествах разметки

Определение подобия подмножеств разметки. Два объекта, не агрегирующие другие объекты, подобны, если

$$o_1 \sim o_2 : \begin{matrix} explain(o_1) = explain(o_2) & И \\ L(explain(o_1)) = L(explain(o_2)) \end{matrix} \quad (1)$$

Два подмножества разметки подобны в том и только в том случае, если

$$L_1 \sim L_2 : |L_1| = |L_2| \quad И \quad \forall o_i \in L_1 \quad \exists! o_j \in L_2 \quad .o_i \sim o_j . \quad (2)$$

Два объекта, содержащие агрегированные объекты, подобны, если

$$o_1 \sim o_2 : \begin{matrix} explain(o_1) = explain(o_2) & И \\ L(explain(o_1)) = L(explain(o_2)) & И \\ L(o_1) \sim L(o_2) \end{matrix} \quad (3)$$

Данное определение рекурсивно.

Подобие множеств разметки является ограничительным отношением в силу своей симметричности. Однако на его основе мы введем менее ограничительное отношение включение подобия.

Множество разметки L_1 включает подобие множества разметки L_2 в том и только том случае, если

$$L_1 \supseteq L_2 : \forall o_i \in L_2 \quad \exists! o_j \in L_1 \quad .o_i \sim o_j . \quad (4)$$

Основным отличием данного определения от определения (2) является отсутствие ограничения на равенство мощностей множеств. То есть множество L_1 может содержать, помимо объектов, подобных объектам множества L_2 , объекты, никоим образом не относящиеся к множеству L_2 .

Определение значения. Значением объекта является цепочка символов, в силу своей универсальности:

$$o.value \in char^* ,$$

или в операторной форме:

$$valueOf(o) = \omega \in char^* .$$

Определение ссылки. Объект o_1 ссылается на объект o_2 , если

$$o_1.reference = o_2 \quad И \quad o_1.value = o_2.value ,$$

что выражается оператором

$$destinationOf(o_1) = o_2 .$$

Определение клона. Клоном считается ссылка, но без ограничения по совпадению значения объектов:

$$o_1.reference ;$$

$$o_2 = clone(o_1) .$$

Свойства объекта

Итак, объект обладает свойствами, которые выражаются агрегированными в нем значениями или ссылками. Мы будем говорить о некотором подмножестве свойств объекта, если некоторое подмножество агрегированных в нем значений и ссылок принадлежит одному подмножеству разметки. Если это подмножество разметки содержит только агрегированные объекты объекта o , то мы будем отражать это в виде $L(o)$. Если некоторое подмножество разметки содержит только агрегированные объекты объекта o , и причем все такие объекты, мы будем записывать $L_x(o)$:

$$\bigcup L_i(o) = L_x(o). \quad (5)$$

Хотя для дальнейших рассуждений нам совсем необязательно рассматривать исключительно полные наборы свойств.

Классификация

Подмножества разметки с определенным на них отношением подобия и включения подобия являются простейшим способом формулирования классов на множестве объектов.

Если объект имеет свойства, то будем иметь в виду, что мы можем сформировать подмножество разметки, такое, что подмножество разметки свойств объекта будет обладать подобием ему или включать его подобие. Иными словами, мы выделим тип объекта в виде отдельного подмножества. Отношение подобия установит сходство объектов по наличию у них сходного состава свойств и выступит инструментом классификации.

Тогда классом C объектов системы, заданным некоторым множеством разметки L , будет множество объектов, обладающих свойством:

$$C(L) = \{o_i\} | \forall o_i \in C(L) \quad L(o_i) \ni L. \quad (6)$$

Говорить о том, что мы, обладая инструментом классификации, предложим язык манипулирования данными в такой ассоциативной среде, еще рано. И тем более рано говорить об эффективных реализациях операций над динамическими классами. Однако необходимо подчеркнуть, что классы, которые появились как вторичный продукт на множестве объектов, могут естественным образом эксплуатировать реляционную алгебру и реляционное исчисление как готовый, хорошо изученный математический аппарат манипулирования данными.

Чтобы ощутить все удобства эксплуатации объектно-ассоциативной модели данных, можно рассмотреть наиболее характерные области применения этого подхода, рассмотреть предлагаемую нами реализацию ОАБД NormPro.

Область применения объектно-ассоциативной базы данных

Ряд отраслей испытывает острую потребность в автоматизации задач хранения и обработки большого объема данных в виде схем, карт, технических документов, чертежей и т.д. Примерами подобных баз данных могут служить разнообразные геоинформационные системы, базы данных оборудования и иных объектов, распределенных и увязанных с информацией на чертежах, схемах, картах. Базы данных информации такого характера требуют специфичных возможностей.

В частности, на начальном этапе не представляется возможным определить все характеристики, какими могут обладать учитываемые объекты.

Реализация

Объектно-ориентированная база данных NormPro предлагает профессиональный инструмент подготовки технической документации самого разного назначения. Формируемый таким образом интерфейс можно отнести к гиперсхемам (рис. 2). Посредством интерактивных схем появляется возможность организовать удобное перемещение среди документов.

Элементы схем могут регистрироваться в базе данных. Согласно формальному описанию в NormPro реализован механизм организации страниц свойств объектов, путем порождения подмножеств разметки для конкретных объектов, по подобию эталона.

Имеется возможность обогащать сущности дополнительными свойствами или наборами свойств. Реализуется именно та неопределенность относительно многообразия сущностей, которая является концептуальным отличием объектно-ассоциативной технологии.

Кроме этого, имеется возможность организовать мультиязычный интерфейс. Для поддержки мультиязычности в NormPro реализован механизм переключателей.

Переключатель - это объект объектно-ассоциативной базы данных, который имеет связанное с ним подмножество разметки, состоящее из клонов. Переключатель является инициатором действия копирования значений клонов в оригиналы объектов:

$$L(o) = \{o_i\} | o_i = clone(o_j) \forall o_i \in L(o), \quad o_j \in O. \quad (7)$$

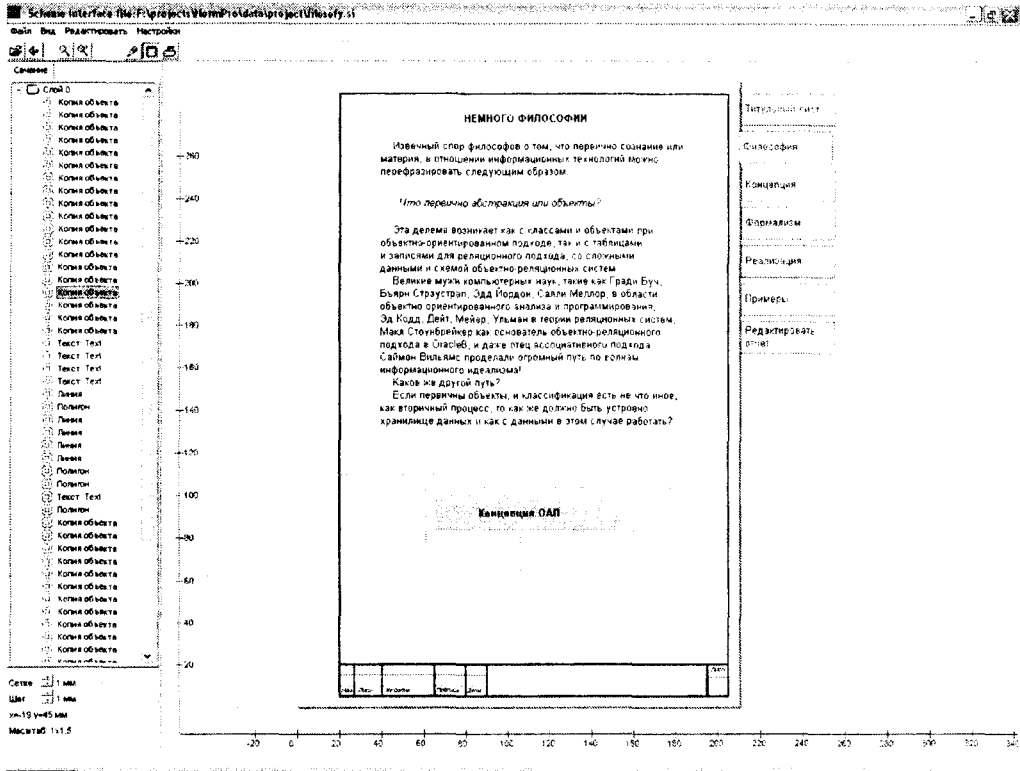


Рис. 2. Объектно-ориентированная база данных NormPro

Если объект o есть переключатель, то согласно (7), переключение – это операция копирования (рис. 3):

$$\forall o, \in L(o) \quad o, \text{reference.value} = o, \text{value} \tag{8}$$

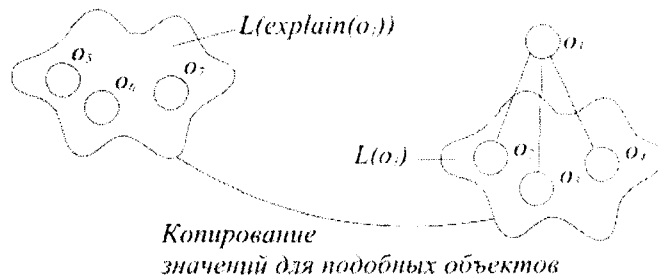


Рис. 3. Иллюстрация действия переключателя

Данный инструмент также обладает возможностью строить документы по шаблонам и предлагает библиотеки элементов для быстрого создания, например, схем размещения оборудования внутри зданий.

Графические возможности NormPro не уступают мощным современным векторным графическим редакторам. В нем реализовано управление произвольными регионами и основные операции над регионами: пересечение, объединение, разность, инверсия. В NormPro реализовано управление прозрачностью графических элементов, вывод текста с альфа-сглаживанием. Поддерживается масштабирование изображения, управление слоями, группировки объектов, управление планами. Поддерживается горизонтальный и вертикальный текст, форматирование текста, управление цветом заливки объектов, цветом линий. Данные возможности дополняют средства объектно-ассоциативной базы данных (рис. 4), по сути, превращая NormPro в средство создания универсальных интерфейсов пользователя.

NormPro поддерживает форматы от A4 до A1 горизонтального и вертикального размещения. Предлагает качественную печать с предварительным просмотром результата (рис. 5).

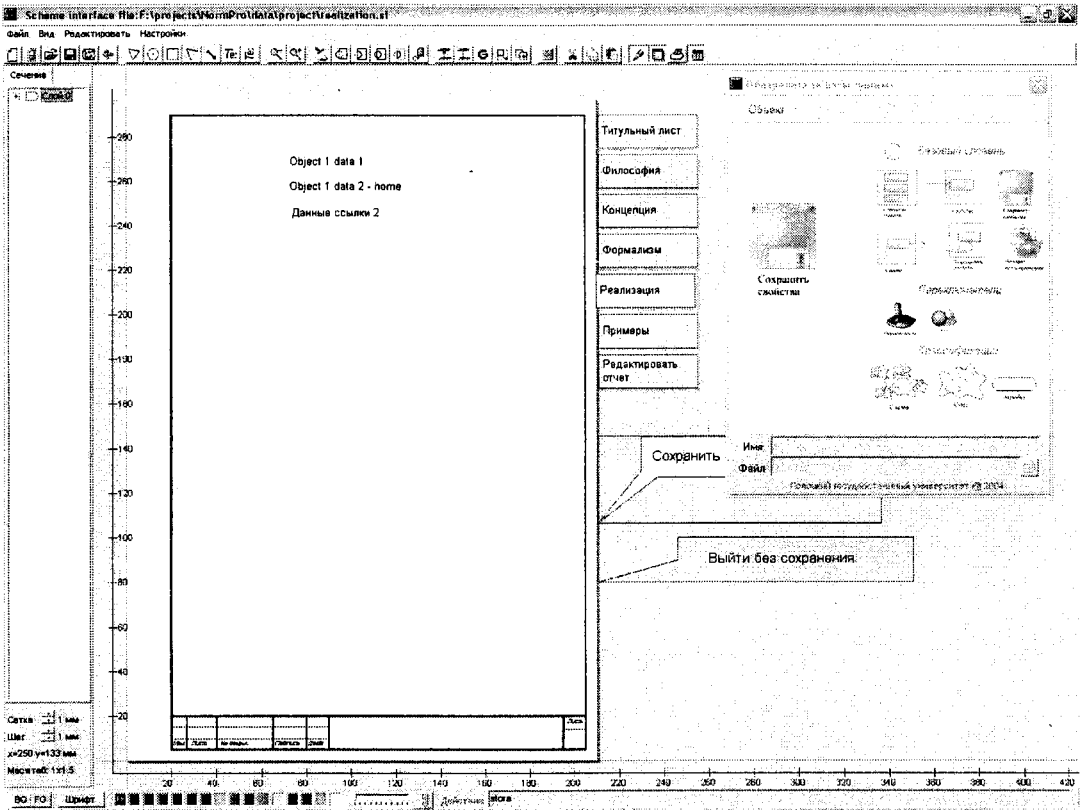


Рис. 4. Реализованные возможности работы со схемами

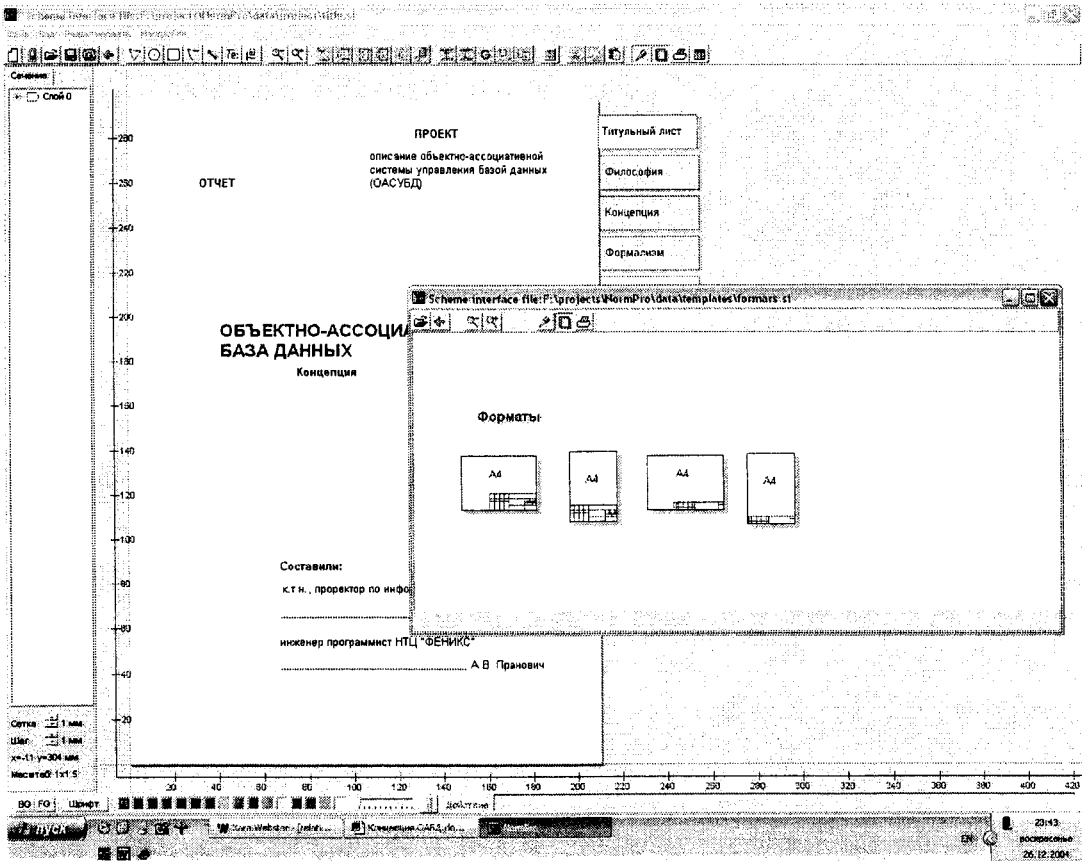


Рис. 5. Библиотеки графических объектов

Результат динамической классификации объектов представляется в виде документа Microsoft Word, который, по сути, является паспортом системы (рис. 6).

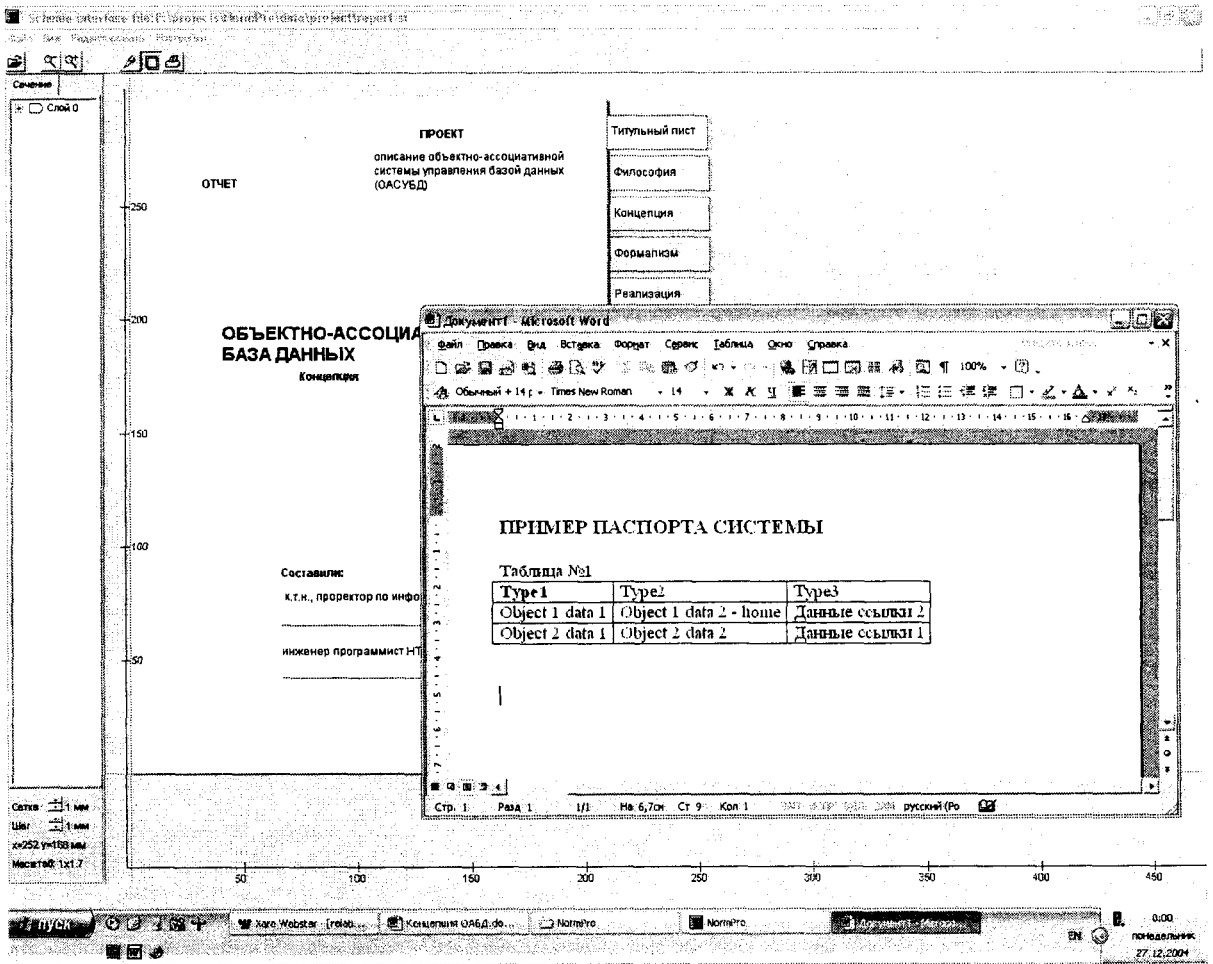


Рис. 6. Пример результата динамической классификации

Потребность в такой базе данных есть практически в любой организации. Конечно, мы находимся на начальном этапе исследования объектно-ассоциативных систем, но перспективы, которые они открывают перед нами, очевидны.

Выводы

Полученное описание достаточно универсально. Все отношения сведены к одному базовому отношению включенности. В силу своей универсальности оно отражает ту концептуальную особенность, что все объекты равноправны, равновесомы, в некотором смысле обезличены. Нет никакой абстракции. Есть только однородная ассоциативная сеть. Абстракция же является продуктом вторичным.

Такая модель данных идеально подходит для создания базы данных в условиях начальной неопределенности относительно многообразия объектов в системе, полных спецификаций объектов и возможных связей между объектами. В связи с тем, что именно такая неопределенность стала ключевым мотивом разработки объектно-ассоциативной базы данных, эта модель позволяет расширять и обогащать систему во время эксплуатации без ущерба для хранимых в ней данных.

Основное направление дальнейшей работы нам представляется в разработке языка управления данными в объектно-ассоциативной среде и создании эффективных реализаций теоретико-множественных и специальных операций в ассоциативной объектной системе.

ЛИТЕРАТУРА

1. Williams S. The Associative Model of Data. - Second edition. - Published in Great Britain: Lazy Software Ltd, 2002. - 272 p.
2. Booch G. Object-oriented analysis and design with applications. - The Benjamin: Cummings Publishing Company, Inc., 1994. - 589 p. (Пер. с рус. Г. Буч. Объектно-ориентированный анализ и проектирование: с примерами приложений на C++. - М.-СПб.: Бином, Невский диалект, 1998. - 560 с.).
3. Codd E.F. Relation Model of Data for Large Shared Data Banks // Comm. ACM. - 1970. - V. 13, № 6. - P. 377 - 383. (Пер. с англ. Кода Е.Ф. Реляционная модель данных для больших совместно используемых банков данных // СУБД. - 1995. - № 1. - С. 145 - 160).
4. Страуструп Б. Язык программирования C++: Пер. с англ. - 3-е изд.- СПб.: Невский диалект; М.: Бином, 1999. - 991 с.
5. Шлеер С., Меллор С. Объектно-ориентированный анализ: моделирование мира в состояниях: Пер. с англ. - Киев: Диалектика, 1993. - 204 с.
6. Taivalsaar A. A critical view of inheritance and reusability in object-oriented programming: Acad. Diss. Univ. Of Jyväskylä - Univ. of Jyraskyla, 1993. - 276 p.
7. Дейт К. Введение в базы данных. - 6-е изд. - Киев: Диалектика, 1998.
8. Пржиялковский В.В. Абстракции в проектировании баз данных // СУБД. - 1998. - № 1,2. - С. 90 - 97.
9. Стоунбрейкер М. Объектно-реляционные системы баз данных // Открытые Системы. - 1994. - № 4 .
10. Васкевич Д. Кризис баз данных и проблема выбора: повестка дня до 2001 года // СУБД, - 1995. - № 1.
11. Кузнецов С.Д. Направления исследований в области управления базами данных: краткий обзор // СУБД, - 1995. - № 1.
12. Зильбершатц А., Стоунбрейкер М., Ульман Дж. Базы данных: достижения и перспективы на пороге XXI столетия // СУБД. - 1996. - № 3.