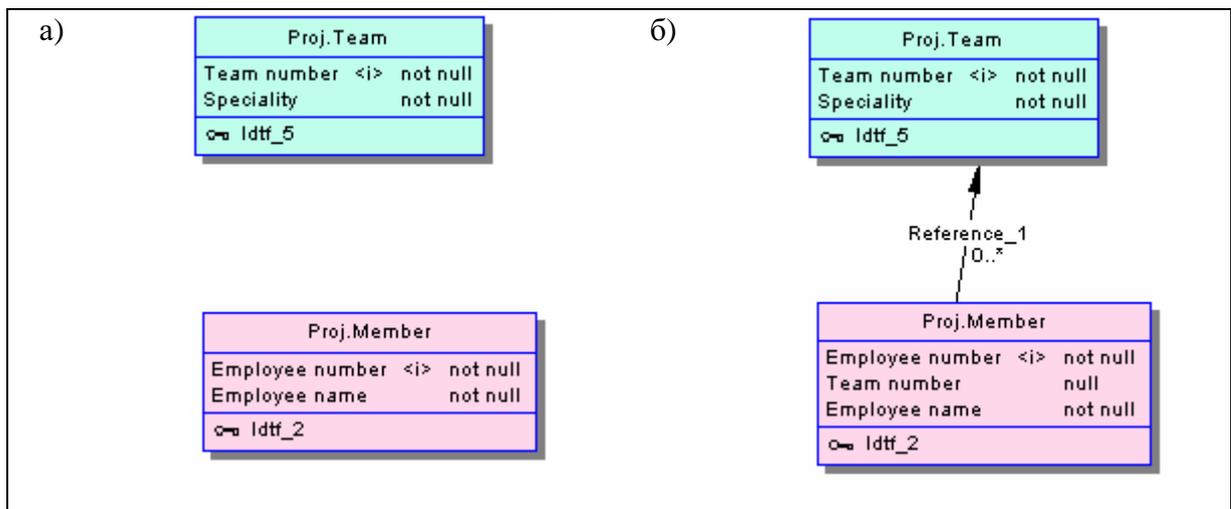


## ИНСТРУМЕНТАРИЙ АВТОМАТИЗИРОВАННОЙ НОРМАЛИЗАЦИИ ОТНОШЕНИЙ ЭЛЕМЕНТОВ ПРЕДМЕТНОЙ ОБЛАСТИ

А. Ж. Каливанов (Санкт-Петербург)

### Введение

При построении схем данных (в частности, баз данных) администраторы систем пользуются такими известными CASE-средствами, как PowerDesigner (Sybase), ERwin (CA), Designer/2000 (Oracle) и др. [1] Решение вопроса о «правильности» (нормализации, оптимизации, соответствия предметной области) схем целиком зависит от опыта работы администратора. К сожалению, современные CASE-средства позволяют лишь устранять ошибки формирования связей двумя объектами (рис. 1) [2].



**Рис. 1. Формирование связи между двумя объектами:**  
 а – объект Proj.Member не содержит атрибутов для связи с объектом Proj.Team;  
 б – автоматическое добавление атрибута Team\_number для устранения ошибки при формировании связи между объектами

Такой подход приводит к временному решению, поскольку не устраняет избыточности элементов в схеме и не гарантирует нормализации отношений между элементами (свойствами) внутри объекта [3, 4]. Поэтому предполагается, что существует систематический подход к анализу аналитических отчетов по изучению исследуемой предметной области (ПО), формированию объектов ПО (и их взаимосвязей), программных интерфейсов (реализующих информационные запросы) к ним и интеграции формальных описаний в CASE-средства.

В настоящее время нами реализован ряд модулей, выполняющих анализ взаимосвязей между объектами и их свойствами, позволяющий приводить отношения ко второй нормальной форме.

### Основы формализации аналитических отчетов

Процесс проектирования информационных систем и структур данных, составляющих основу информационного фонда информационной системы (ИС) начинается с обследования ПО. Результатами обследования являются такие документы, как карты (вопросники) обследования, протоколы и отчеты.

Анализ документов обследования позволяет выявить объекты  $\{A_j\}_{j=1(I)N}$  ПО  $U=\{A_j\}_{j=1(I)N}$  и их свойства – элементы данных (атрибуты)  $a_j \in \{A_j\}_{j=1(I)N}$   $[j=1(I)M]$ . Взаимо-

связи между объектами и их свойствами могут быть представлены бинарными отношениями  $F_A = A_i \times A_n$  (связь *Reference\_1* на рис. 1) и  $f_a = a_j \times a_m$ ,  $m = I(I)M$ ;  $i, n = I(I)N$ . При этом  $a_j$  – основное свойство, а  $a_m$  – зависимое [3, 5, 6].

Объекты  $A_i$  и связи  $F_A$  фиксируются на диаграмме бизнес-процессов (BPM). Далее производится отображение BPM на концептуальную (CDM) и физическую (PDM) модели данных [2].

В основе автоматизированного анализа документов может лежать синтаксический разбор предложений. Тогда в качестве элементов  $a_j$ ,  $A_i$  могут выступать подлежащие (основное свойство) и дополнения (зависимое свойство), а в качестве отношений  $f_a$ ,  $F_A$  – сказуемые предложений, описывающих существенные для реализации бизнес-логики свойства ПО.

Аналогичным образом могут быть представлены информационные запросы к системе. При этом запросы  $Q_A$  будут являться совокупностью отношений  $\{F_A\}$  между объектами.

Таким образом, созданы предпосылки для покрытия информационного фонда системы  $S$  множеством элементов и связей между ними:  $S = \langle \{a_j\}, \{f_a\} \rangle$  [3]. Кроме того, можно говорить об адекватности системы  $S$  заданной предметной области, или, по крайней мере, ее описанию, документам обследования.

Примерами автоматизированных средств, реализующих синтаксический разбор, являются системы лингвистического перевода текстов (например, ABBYY Lingvo), а также средства проверки правописания в текстовых редакторах (например, MS Word).

### Порядок формирования объектов предметной области

Дальнейшее проектирование ИС приводит к необходимости устранения аномалий манипулирования данными. Указанные аномалии являются следствием избыточности, присутствующей в исходном логическом описании объектов  $\langle \{A_i\}, \{F_A\} \rangle$ , и количестве предлагаемых к хранению данных [4].

Основной подход к устранению аномалий – упорядочение связей между объектами ПО и их свойствами, выявление возможных ключей и нормализация полученных отношений между свойствами внутри объекта и самими объектами [7, 8]. Предлагаемый далее порядок действий идентичен как для объектов, так и для их свойств, поэтому он будет рассмотрен на примере свойств объектов.

Пусть свойства и отношения между основными и зависимыми свойствами сведены в квадратную матрицу  $MF_{[M]}$  [4]. Например, для  $M = 4$ , где

		1	2	3	4
1.	Team_number	1	1		
2.	Speciality		1		
3.	Employee_number	1		1	1
4.	Employee_name				1

$$MF_{[4]} \Rightarrow$$

основные свойства представлены в строках, а зависимые – в столбцах матрицы.

Наличие свойства и отношения между двумя свойствами в матрице  $MF_{[M]}$  задано единичным значением соответствующей ячейки. При наличии множественной связи, соответствующая единица (не находящаяся на главной диагонали) может быть заменена значением  $N > 1$ . Тогда, подсчитав количество единиц в строках и перенумеро-

вав строки в порядке убывания количества единиц в каждой строке, можно построить регулярную (упорядоченную) матрицу  $\Delta MF_{[M]}$  (рис. 2).

$$MF_{[4]} \Rightarrow \begin{array}{c} \begin{array}{cccc} & 1 & 2 & 3 & 4 \\ 1 & \left[ \begin{array}{cccc} 1 & 1 & & \\ & 1 & & \\ 3 & 1 & & 1 & 1 \\ & & & 1 \end{array} \right] \end{array} \\ \begin{array}{cc} \text{Число} & \text{Новая} \\ \text{ед.} & \text{позиция} \end{array} \\ \begin{array}{cc} 2 & 2 \\ 1 & 3 \\ 3 & 1 \\ 1 & 4 \end{array} \end{array} ; \Rightarrow \Delta MF_{[4]} = \begin{array}{c} \begin{array}{cccc} & 3 & 1 & 2 & 4 \\ 3 & \left[ \begin{array}{cccc} 1 & 1 & & 1 \\ & 1 & 1 & \\ 2 & & 1 & \\ & & & 1 \end{array} \right] \end{array} \end{array}$$

Рис. 2. Схема построения регулярной матрицы  $\Delta MF_{[M]}$

Из строк, содержащих более одной единицы, могут быть сформированы возможные ключи  $\{K_i\}_{i=1(I)N}$  объектов  $A_i$ ;  $[K_i \subset \{A_i\}_{i=1(I)N}]$  [5].

Так, например, на рис. 1 ключами являются атрибуты  $a_1 = K_1$  и  $a_3 = K_3$ , а объектами, построенными на основе указанных ключей, –  $A_1 \supset \{K_1\}$  и  $A_3 \supset \{K_3\}$ .

Проверить правильность формирования ключей можно по имеющимся данным, сведенным в таблицу. Процедура проверки состоит в построении проекций на каждый из указанных атрибутов. Если мощности  $|A_1|$  и  $|A_3|$  множества значений атрибутов  $A_1$  и  $A_3$  равны мощности множества кортежей заданной таблицы, то ключи  $K_1$  и  $K_3$  выбраны правильно [4]. Иногда в исходной таблице нет атрибутов с указанными свойствами. Тогда формируют проекции на два и более (но меньше  $N$ ) атрибутов до тех пор, пока не будут найдены все возможные ключи заданной таблицы.

Определив ключи и подымаясь вверх по матрице  $\Delta MF_{[M]}$ , от свойств, содержащих по одной единице в строке, до строк, имеющих единицу в колонке, соответствующей анализируемому свойству, дополним полученные объекты следующим образом:  $A_3 = \{K_3, a_1, a_4\}$  и  $A_1 = \{K_1, a_2\}$ .

Далее состав свойств объекта уточняется (нормализуется) применением более сложных процедур, основанных на аксиомах вывода функциональных зависимостей [7, 8].

**Возможность формирования интерфейсов к объектам.** Как видно из рассмотренного выше примера, связи между объектами сохраняются:  $F_A = A_3 \times A_1$  или  $f_a = K_3 \times a_1$ . На основе указанных связей и предполагаемых информационных запросов  $Q_A$  к системе  $S$  формируются интерфейсы к объектам [1].

Интегрированное представление объектов, связей и интерфейсов к объектам на основе UML-описаний позволяет сформировать объектно-ориентированную модель (ООМ) информационной системы [1, 2].

**Подход к интегрированию объектов в CASE-средства.** Рассмотренные формализмы позволяют интегрировать полученное описание ПО в требуемое CASE-средство. Это приведет разрабатываемую систему к завершеному виду.

Так, объекты  $A_i$  со свойствами  $a_j$ , дополненными соответствующими характеристиками (тип данных, размер и пр.) могут быть преобразованы в реляционные таблицы. Направление связей  $f_a$  и соответствующие атрибуты позволят сформировать внешние ключи в зависимых таблицах. На основе информационных запросов  $Q_A$  к системе  $S$  могут быть сформированы представления (View) оперативных объектов.

Полученные SQL-скрипты без труда могут быть обработаны практически любым современным CASE-средством [2, 8].

**Заключение.** В настоящее время созданы предпосылки для автоматизированной разработки информационных систем на основе текстовых описаний (аналитических отчетов по обследованию ПО, технических заданий на разработку системы и пр.).

Имеется задел теоретических и программных решений, позволяющий интегрировать полученные формальные описания в современные CASE-средства.

Ожидается, что предлагаемый подход существенно снизит затраты на создание пилотного проекта информационной системы.

### Литература

1. **Харрингтон Д.** Проектирование объектно-ориентированных баз данных: Пер. с англ. – М.: ДМК Пресс, 2001. – 272 с.: ил. (Серия “Для программистов”).
2. PowerDesigner PDM User's Guide//OnlineBooks.1991–2001 Sybase, Inc. and its subsidiaries. [E/E].
3. **Мартин Дж.** Организация баз данных в вычислительных системах. – М.: Мир, 1980.
4. **Каливанов А. Ж.** Основы проектирования реляционных баз данных. Руководство к практическим занятиям. – СПб.: ВКА, 2002. – 60 с.
5. **Мейер Д.** Теория реляционных баз данных. – М.: Мир, 1987.
6. **Озкарахан Э.** Машины баз данных и управление базами данных. – М.: Мир, 1989.
7. **Дейт К.** Введение в системы баз данных. : Пер. с англ. – 6-е изд – К.: Наука, 1998. – 784 с.
8. **Хомоненко А. Д., Цыганков В. М., Мальцев М. Г.** Базы данных: Учебник для высших учебных заведений/ Под ред. проф. А.Д. Хомоненко. – СПб.: КОРОНА принт, 2000. – 416 с.