

Дрождин В.В.

Drozhdin V.V.

## Открытость структур в эволюционной модели данных

### Openness of structures in the evolutionary data model

Рассмотрена проблема открытости структур данных в эволюционной модели данных и предложены методы реализации открытости структур как вверх до формирования единой структуры данных локальной системы, так и вниз до байтов и битов, имеющих, по крайней мере, конструктивную семантику.

The article deals with the problem of openness of structures in the evolutionary data model and the methods of realization the openness of structures, both up to the formation of the united data structure of the local system, and down to bytes and bits having, at least, constructive semantics.

**Ключевые слова:** самоорганизующаяся система, информационная система, модель данных, эволюционная модель данных, структура данных, преобразование структур данных, операции над структурами данных, семантика данных

**Key words:** self-organizing system, information system, data model, evolutionary data model, data structure, transformation of data structure, operations on data structure, data semantics

Эволюционная модель данных (ЭМД) предназначена для создания активных самоорганизующихся информационных сред (СИС), способных самостоятельно поддерживать информационные модели с высокой степенью адекватности отражаемой предметной области в течение длительного времени. Это требует наличия в ЭМД очень мощных средств для организации информации об объектах различной структуры и сложности и возможности формирования представлений объектов с требуемой степенью детализации, включающих обобщенные (укрупненные, интегральные) показатели.

В [1] на основе системного подхода определяется пятислойная организация данных локальных систем. Эта организация данных является достаточно гибкой и может быть адаптирована для создания СИС. При этом различают структуры  $S^i$ , являющиеся целостными объектами-системами и представляющими один объект  $i$  уровня, и структуры  $R^i$ , являющиеся множествами допустимых структур  $S^i$  [2]. В таблице 1 приведены слои организации данных локальной системы с их описаниями.

Таблица 1.

Номер слоя	Структуры слоя	Описание структур	Отношения между данными	Ограничения на данные
0	$S^0, R^0$	Абстрактный тип данных	---	---
1	$S^1, R^1$	Допустимое подмножество абстрактного типа данных	---	Ограничения на атомарные данные
2	$S^2, R^2$	Структура с жесткими (логическими) связями	Отношения (взаимосвязи) между данными типа функциональных и многозначных зависимостей	Ограничения на наличие и определенность ключей и др.
3	$S^3, R^3$	Структура, представляющая совместно используемые данные	Отношения, определяющие совместное использование данных	Ограничения на совместное использование данных
4	$S^4, R^4$	Структура, представляющая всю совокупность данных локальной системы	Отношения, определяющие автономную совокупность данных	Ограничения на автономность данных

Приведенные структуры конструктивно имеют следующие характеристики:

-  $R^0$  – тип данных языка программирования или абстрактный тип данных определенный и реализованный в системе, элементами которых являются атомарные объекты  $S^0$ ;

-  $R^1$  – подмножество базового типа  $R^0$ , объекты  $S^1$  которого получены по закону  $f$  (в частном случае тривиальному) из объектов  $S^0$ ;

-  $R^2$  – множество сложных объектов  $S^2$ , каждый из которых является композицией объектов  $S^1$ ;

-  $R^3$  – более сильно связанная (совместно используемая) часть объектов  $S^2$  или совместно используемые  $S^2$  и ранее созданные объекты  $S^3$ ;

-  $R^4$  – единственный объект  $S^4$ , представляющий всю взаимосвязанную совокупность данных  $S^2$  и  $S^3$  локальной системы.

Можно провести определенную аналогию между структурами ЭМД и структурами реляционной модели данных (РМД), представленную в таблице 2.

Таблица 2.

Структуры ЭМД	Структуры РМД
$S^1$	элемент домена
$R^1$	домен
$S^2$	кортеж
$R^2$	отношение
$S^3$	кортеж представления
$R^3$	представление
$S^4, R^4$	база данных

Как видно из характеристики структур и проведенной аналогии их со структурами РМД открытость структур ЭМД вверх до формирования единой структуры, содержащей всю базу данных локальной системы, не содержит серьезных (логических) ограничений и, следовательно, может считаться вполне приемлемой для организации данных в рамках ЭМД. Однако открытость структур вниз (до байтов и битов) ограничивается структурами  $R^1$ , объекты  $S^1$  которых обладают изначально ми-

нимальной семантикой в моделируемой предметной области (ПО) и представляют, например, фамилии, адреса, даты рождения и др. Но возникшая потребность в более точном моделировании ПО часто требует выделения отдельных компонентов из принятых изначально минимальных объектов, например, названия населенного пункта из адреса или фамилии из ФИО. Поэтому необходима разработка методов и средств, позволяющих декомпозировать целостные структуры  $S^1$  на более мелкие подструктуры  $S^{-1}$  с возможностью восстановления из них исходных структур  $S^1$ .

Для декомпозиции и согласования структур  $S^1$  и  $S^{-1}$  будем использовать две операции  $\theta^1$  и  $\theta^{-1}$ , которые определим следующим образом:

$$\begin{aligned}\theta^1 : S^1 &\rightarrow \langle S_1^{-1}, S_2^{-1} \rangle; \\ \theta^{-1} : \langle S_1^{-1}, S_2^{-1} \rangle &\rightarrow S^1.\end{aligned}$$

Операция  $\theta^1$  осуществляет декомпозицию структуры  $S^1$  на две структуры более низкого уровня  $S_1^{-1}$  и  $S_2^{-1}$  путем выделения по определенному закону из структуры  $S^1$  подструктуры  $S_1^{-1}$  и формирования оставшейся части структуры  $S_2^{-1}$ , а операция  $\theta^{-1}$  выполняет композицию структур  $S_1^{-1}$  и  $S_2^{-1}$  в структуру  $S^1$  по закону, обратному  $\theta^1$ . Из объектов  $S_1^{-1}$  и  $S_2^{-1}$  формируются соответственно объекты  $R_1^{-1}$  и  $R_2^{-1}$ , которые в общей структуре объектов будут соответствовать объектам уровня  $R^1$ .

Использование объектов  $R_1^{-1}$  и  $R_2^{-1}$  ничем не отличается от использования обычных объектов  $R^1$  и только для восстановления исходного объекта  $R^1$  необходимо применить ко всем объектам  $S_1^{-1}$  и  $S_2^{-1}$  операцию  $\theta^{-1}$ .

Последовательное применение операций  $\theta_1^1, \theta_2^1, \dots, \theta_m^1$  к остаточной структуре предыдущего разбиения  $(\theta_1^1 : S^1 \rightarrow \langle S_{1,1}^{-1}, S_{2,1}^{-1} \rangle, \theta_2^1 : S_{2,1}^{-1} \rightarrow \langle S_{1,2}^{-1}, S_{2,2}^{-1} \rangle, \dots, \theta_m^1 : S_{2,m-1}^{-1} \rightarrow \langle S_{1,m}^{-1}, S_{2,m}^{-1} \rangle)$  позволяет декомпозировать исходную структуру  $S^1$  на  $m+1$  структур более низкого уровня, а последовательность операций  $\theta_1^{-1}, \theta_2^{-1}, \dots, \theta_m^{-1}$  восстанавливает исходный объект  $S^1$ , т.е.

$$\begin{aligned}\theta_1^1 \circ \theta_2^1 \circ \dots \circ \theta_m^1 : S^1 &\rightarrow \langle S_{1,1}^{-1}, S_{1,2}^{-1}, \dots, S_{1,m}^{-1}, S_{2,m}^{-1} \rangle; \\ \theta_1^{-1} \circ \theta_2^{-1} \circ \dots \circ \theta_m^{-1} : \langle S_{1,1}^{-1}, S_{1,2}^{-1}, \dots, S_{1,m}^{-1}, S_{2,m}^{-1} \rangle &\rightarrow S^1.\end{aligned}$$

В качестве простой операции  $\theta^1$  может использоваться операция выделения одного или  $k$  первых слов из  $S^1$ , в результате чего  $S^1$  разбивается на  $S_1^{-1}$  и  $S_2^{-1}$ , причем в  $S_1^{-1}$  может содержаться не более одного или  $k$  слов, а в  $S_2^{-1}$  – оставшаяся часть строки  $S^1$ . При этом операция  $\theta^{-1}$  будет выполнять конкатенацию, т.е. приписывание в конец строки  $S_1^{-1}$  строки  $S_2^{-1}$ .

В более сложных случаях операция  $\theta^1$ , например, может выделять различные компоненты адреса, заданного строкой символов, выделять некоторую подстроку из строки по заданному условию, декомпозировать числовое значение на два значения путем использования операции взятия по модулю и другие. Операция  $\theta^{-1}$  всегда будет восстанавливать из объектов  $S_1^{-1}$  и  $S_2^{-1}$  исходный объект  $S^1$ .

Типы данных объектов  $S_1^{-1}$  и  $S_2^{-1}$ , т.е. объекты  $S_1^0$  и  $S_2^0$ , из которых формируются объекты  $S_1^{-1}$  и  $S_2^{-1}$ , могут совпадать с объектом  $S^0$ , из которого формировался исходный объект  $S^1$ , или отличаться от него. Например, при выделении фамилии из ФИО как  $S^0$ , так и  $S_1^0$  и  $S_2^0$  являются строками символов, а при выделении почтового индекса из адреса  $S_1^0$  может быть преобразован в целое число, но  $S_2^0$ , как и  $S^0$ , останется строкой символов.

Семантика объектов  $S_1^{-1}$  и  $S_2^{-1}$  может быть двойкой. Если для реализации операции  $\theta^1$  используется поименованный запрос или хранимая процедура, то их имя присваивается объекту  $R_1^{-1}$ , а объект  $R_2^{-1}$  будет иметь имя исходной структуры  $R^1$ . Поэтому после декомпозиции объекты  $S_1^{-1}$  и  $S_2^{-1}$  будут опять обладать семантикой в терминах ПО.

Если же для реализации операции  $\theta^1$  используется общая часть часто задаваемых запросов, то объекту  $R_1^{-1}$  присваивается некоторое стандартное (внутрисистемное) имя и он будет обладать только конструктивной семантикой, которая отражает целостность и способ получения объектов  $S_1^{-1}$ . Интерпретации в ПО объекты  $S_1^{-1}$  в этом случае иметь не будут. Однако если в дальнейшем функционировании СИС встретится поименованный запрос, выбирающий все объекты  $S_1^{-1}$ , то объекту  $R_1^{-1}$  будет присвоено имя этого запроса и объекты  $S_1^{-1}$  приобретут семантику в терминах ПО.

Если операция  $\theta^1$  осуществляет разбиение объекта  $S^1$  на  $S_1^{-1}$  и  $S_2^{-1}$ , то после формирования объектов  $R_1^{-1}$  и  $R_2^{-1}$  путем применения  $\theta^1$  ко всем  $S^1 \in R^1$  операция  $\theta^1$  становится тривиальной, а операция  $\theta^{-1}$  – эквивалентной обычной операции композиции двух объектов  $S^{1'}$  и  $S^{1''}$ , что позволяет удалить специализированные операции  $\theta^1$  и  $\theta^{-1}$ . Наиболее часто разбиением является деление строки на две подстроки. Например, после выделения из всех адресов почтового индекса могут считаться самостоятельными структурами  $R_1^1$  – “почтовый индекс” и  $R_2^{-1}$  – “адрес”, операции выделения почтового индекса и восстановления полного почтового адреса могут быть удалены.

Проблема формирования обобщенных показателей является существенно более сложной, поэтому в данной работе рассмотрим только простейший вариант обобщения объектов.

Простое обобщение можно сформировать на основе принципа совместного использования данных. Например, если имеются отдельные параметры число, месяц и год, часы, минуты и секунды или фамилия, имя и отчество, а их данные в подавляющем большинстве случаев используются совместно, то целесообразно комбинировать их в укрупненные параметры “дата”, “время” и “ФИО” с указанием местоположения каждого отдельного параметра в укрупненном параметре. При этом отдельные параметры будут соответствовать структурам  $S_1^{-1}$ ,  $S_2^{-1}$  и  $S_3^{-1}$ , а укрупненный параметр – структуре  $S^1$ . Осуществление композиции укрупненного параметра из отдельных параметров будет соответствовать операции  $\theta^{-1}$ , а выделение отдельных параметров из укрупненного параметра – операции  $\theta^1$ .

Укрупнение параметров позволяет стандартным способом уменьшать количество компонентов в СИС без организации дополнительных слоев, что повышает эффективность обработки данных.

Взаимобратные операции  $\theta^{-1}$  и  $\theta^1$ , введенные для осуществления композиции и декомпозиции структур, являются операциями-шаблонами, для которых известны: назначение, принцип действия и требования к исходным данным и результату. Особенности выполнения операций зависят от типов обрабатываемых данных и методов преобразования структур, поэтому их реализация на основе запросов пользователей будет определять возможности конкретных СИС в осуществлении эволюции внутренней организации данных и достижении определенного максимума эффективности обработки данных.

Таким образом, открытость структур данных вверх и вниз в ЭМД позволяет создавать информационные модели предметной области с произвольной степенью детализации и последующим уточнением или огрублением этой модели с помощью средств, имеющихся в ЭМД.

### Литература

1. Дрождин В.В. Системный подход к построению модели данных эволюционных баз данных. – Программные продукты и системы, № 3, 2007. – С. 52-55.
2. Система, симметрия, гармония. - М.: Мысль, 1988. - 315 с.