МОДЕЛИРОВАНИЕ МНОГОМЕРНЫХ ДАННЫХ В СИСТЕМЕ METAS BI-PLATFORM

Павел Мальцев

***Аннотация**: Представлено формальное описание многомерной модели данных, реализованной в программном комплексе METAS BI-Platform. В статью включено описание объектов многомерной модели (измерений и множеств измерений и т.д.), их свойств и организации, а также операций, выполняемых над ними. Описаны методы агрегации многомерных данных, позволяющие эффективно агрегировать массивы числовых показателей. Программный комплекс METAS BI-Platform предназначен для многомерного анализа данных, получаемых из гетерогенных источников, и позволяет упростить разработку BI-приложений. Программный комплекс представляет собой многоуровневое приложение с архитектурой «Клиент-сервер». Каждый уровень комплекса соответствует степени абстракции данных. На самом низком уровне расположены драйверы доступа к специфическим физическим источникам данных. Следующий уровень – уровень виртуальной СУБД, позволяющей осуществлять унифицированный доступ к данным, что избавляет от необходимости учитывать специфику конкретных СУБД при разработке BI-приложений. Реализован программный интерфейс комплекса (API). В распоряжение разработчиков предоставляется набор готовых компонентов, которые могут быть использованы при создании BI-приложений. Это позволяет разрабатывать на основе комплекса BI-приложения, отвечающие современным требованиям, предъявляемым к подобным системам.*

***Ключевые слова**: Business Intelligence, BI, бизнес-анализ, OLAP, системы поддержки принятия решений, DSS, модель многомерных данных, многомерный анализ.*

***ACM Classification Keywords**: H.2 Database Management: H.2.1 Logical Design – Data models, H.2.4 Systems – Distributed databases; H.4 Information Systems Applications: H.4.2 Types of Systems – Decision support (e.g., MIS).*

***Conference**: The paper is selected from Sixth International Conference on Information Research and Applications – i.Tech 2008, Varna, Bulgaria, June-July 2008*

Введение

В настоящее время всё более широкое применение находят так называемые средства Business Intelligence (BI), которые позволяют облегчить процесс принятия решений за счёт получения необходимых количественных характеристик, являющихся результатом обработки больших объёмов данных, и применения математических методов анализа этих характеристик с целью выявления закономерностей. Естественно, решения принимаются человеком, средства Business Intelligence способны лишь «дать рекомендации», помочь обосновать принимаемые решения.

Работа BI-приложений, как правило, основана на анализе больших объёмов информации, чем больше объём анализируемых данных, тем выше доверие к результатам. Данные для анализа зачастую берутся из реляционных баз данных, но работа с данными в BI-средствах обладает определённой спецификой и реляционная модель данных не всегда отвечает им. Одной из наиболее подходящих для Business Intelligence моделью данных на сегодняшний день является многомерная модель.

В данной статье приводится формальное описание многомерной модели данных, реализованной в программном комплексе METAS BI Platform.

Программный комплекс METAS BI-Platform

Программный комплекс METAS BI-Platform предназначен для проведения многомерного анализа данных, получаемых из гетерогенных источников. Использование данного средства позволит облегчить процесс создания BI-приложений за счёт того, что в комплексе уже реализованы модули сбора и многомерного анализа данных, и разработчик может свободно использовать функции данных модулей.

С точки зрения архитектуры программный комплекс представляет собой многоуровневое клиент-серверное приложение (рис. 1). Это позволяет разрабатывать на основе комплекса BI-приложения, отвечающие современным требованиям, предъявляемым к подобным системам. С другой стороны, архитектуру METAS BI-Platform можно охарактеризовать как иерархическую – модули комплекса разбиты на уровни и модули более высоких уровней используют функции модулей более низких уровней. На рис. 1 схематично представлена архитектура комплекса.

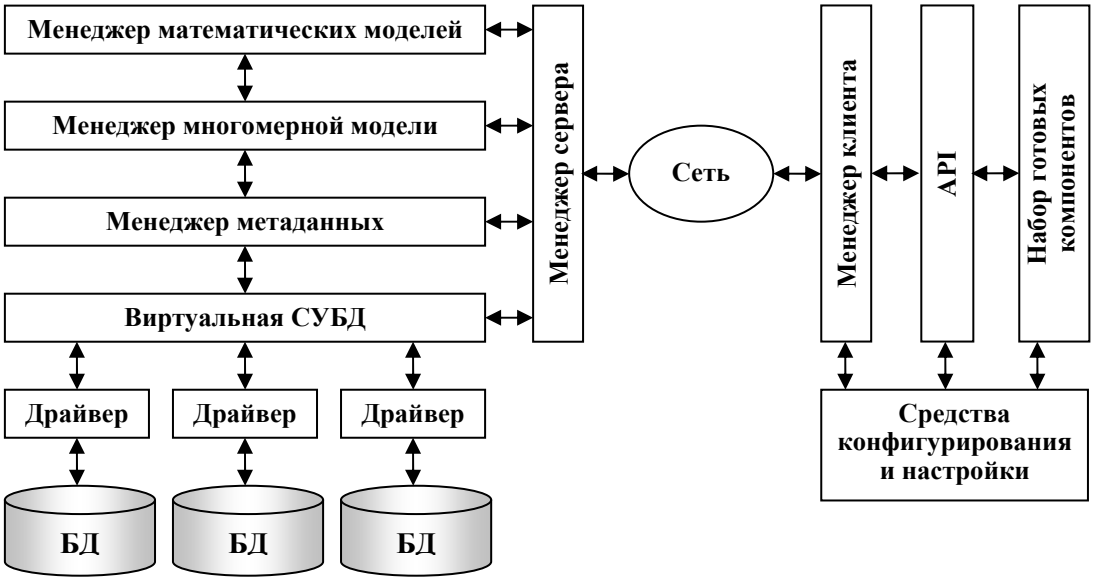

Рис. 1. Архитектура комплекса METAS BI-Platform

Каждый уровень комплекса соответствует степени абстракции данных. На самом низком уровне расположены *драйверы доступа к специфическим физическим источникам данных*. Следующий уровень реализован в виде *виртуальной СУБД*, которая позволяет осуществлять унифицированный доступ к реляционным данным и избавляет модули вышестоящих уровней от необходимости учитывать специфику конкретных СУБД.

Менеджер метаданных осуществляет ведение *базы метаданных* (*репозитория*) комплекса и разработанных на его основе приложений. Метаданные в репозитории представляются с позиции объектного подхода.

В основе работы комплекса лежит технология OLAP. В соответствии с современными требованиями к средствам, реализующим эту технологию, данные должны представляться в многомерной модели. За реализацию этого представления отвечает *менеджер многомерной модели.* Описание многомерной модели, реализованной в METAS BI Platform, приведено ниже.

Иногда возникает потребность в построении некоторых математических представлений данных, получаемых из базы, с целью их анализа. Для решения этой задачи в комплексе реализован *менеджер математических моделей*, частично упрощающий реализацию математических методов анализа данных в конечных приложениях.

Для того чтобы разработчики BI-приложений могли использовать комплекс, реализован программный интерфейс комплекса (API), а также набор готовых компонентов, которые могут быть использованы при разработке BI-приложений.

Описание многомерной модели данных в METAS BI-Platform

Можно сказать, что в многомерной модели данные представляются в виде *вектор-функций* многих переменных: $f_i(x_1,...,x_n)$, где f_i – некоторые числовые показатели (например: объём продаж, сумма сделки), а x_i – параметры. *Параметры* организуются в виде *измерений*, имеющих иерархическую структуру. Будем считать, что *элемент измерения* является некоторым подмножеством множества допустимых значений соответствующего параметра. Мощность многомерной модели заключается в агрегации. На пример, если мы запросили у OLAP-средства общую сумму контрактов за 2006 г., то OLAP-средство само просуммирует суммы всех контрактов из БД, дата которых принадлежит 2006 г.

Объекты многомерной модели

Будем называть *множествами точек измерения* непустые множества строковых значений, а элементы этих множеств, соответственно, – *точками измерения*. Множества точек измерения будем обозначать X^i, где $i \in N$, а точки измерения из множества X^i будем обозначать x_j^i, где $j \in N$.

Рассмотрим некоторое множество точек измерения X, будем обозначать через $\tilde{X}$ такое подмножество $\mathcal{P}(X)$, что:

$$\varnothing \in \tilde{X} \quad (1)$$

$$X \in \tilde{X} \quad (2)$$

$$(\forall x \in X) : \{x\} \in \tilde{X} \quad (3)$$

$$((\forall X^1, X^2 \in \tilde{X}) : (X^1 \bigcap X^2 = \varnothing) \vee (X^1 \bigcap X^2 = X^1) \vee (X^1 \bigcap X^2 = X^2) \quad (4)$$

Замечание: Из (4) видно, что подмножества X в $\tilde{X}$ организованны в виде иерархии. Приведём пример: на рис. 2 изображена схема организации подмножеств измерений.

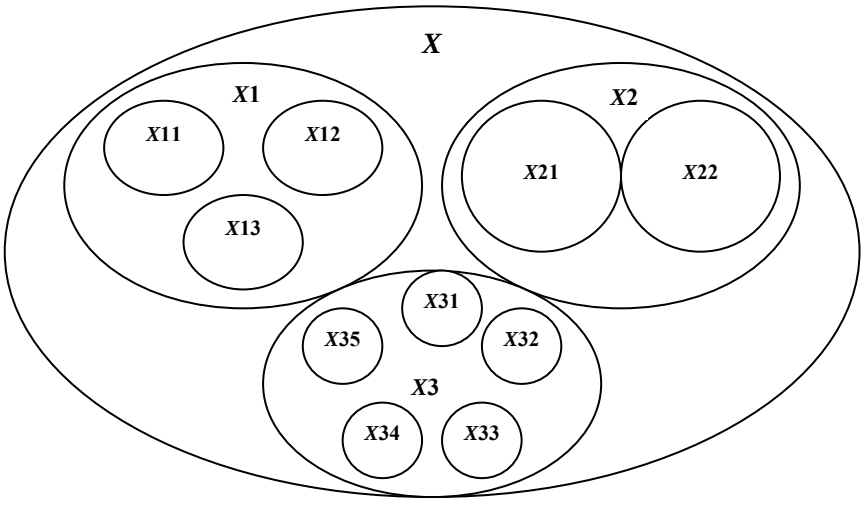

Рис 2. Схема организации подмножеств измерении

Такую организацию подмножеств можно интерпретировать как иерархию, которая изображена на рис. 3.

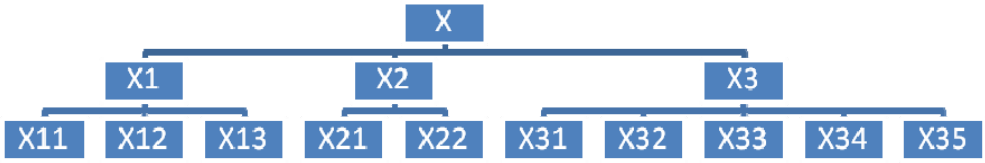

Рис. 3. Иерархия измерений

Пусть дано некоторое множество точек измерения *X*, для него построено измерение $\tilde{X}$. Веткой иерархии измерения $\tilde{X}$ для некоторой точки $x \in X$ будем называть такое подмножество $\tilde{X}$ (обозначим его $Branch(\tilde{X}, x)$), что

$$((\forall X^1 \in Branch(\tilde{X}, x)) : x \in X^1) \wedge ((\forall X^2 \in \tilde{X} \setminus X^1) : x^1 \notin X^2).$$

Длиной ветви иерархии $Branch(\tilde{X}, x)$ $x \in X$ будем называть величину *Card Branch*($\tilde{X}$, *x*), т.е. количество подмножеств из $\tilde{X}$, входящих в *Branch*($\tilde{X}$, *x*).

Измерение $\tilde{X}$ будем называть регулярным, если $(\forall x \in X) : Card\ Branch(\tilde{X}, x) = k$, где $k \in N$. Договоримся понимать под глубиной иерархии некоторого измерения $\tilde{X}$ величину $Depth(\tilde{X}) = \max_{x \in X} Card\ Branch(\tilde{X}, x)$. Очевидно, что если измерение регулярно, то глубина его иерархии равна длине любой его ветки.

Рассмотрим некоторое измерение $\tilde{X}$ и построим группу множеств подмножеств:

$$L_0(\tilde{X}) = \{\varnothing\}$$

$$L_i(\tilde{X}) = \{X^1 \mid X^1 \in \tilde{X} \setminus \bigcup_{j=0}^{i-1} L_j(\tilde{X}), (\forall X^2 \in \tilde{X} \setminus \bigcup_{j=0}^{i-1} L_j(\tilde{X}) \mid X^1 \neq X^2) : (X^2 \subset X^1) \vee (X^1 \bigcap X^2 = \varnothing)\}$$

$$i = \overline{1, m}, m = Depth(\tilde{X})$$

Множества $L_i(\tilde{X})$ будем называть уровнями измерения $\tilde{X}$, а i – номером уровня.

Будем называть набором измерений упорядоченное множество m измерений. Обозначать его будем следующим образом: $\breve{X} = \langle \tilde{X}^1, \tilde{X}^2, ..., \tilde{X}^m \rangle$, $m \in N$. Элементом набора измерений, соответственно, будем называть m-ку $\breve{x} = \langle X^{11}, X^{21}, ..., X^{m1} \rangle$, где $X^{i1} \in \tilde{X}^i$, $i = \overline{1, m}$.

Рассмотрим некоторую предметную область (ПрО). Пусть для этой предметной области у нас разработана некоторая OLTP система, будем обозначать её $OLTP($ПрО, $data)$, где $data$ – набор данных. Объединим все возможные для ПрО наборы данных в множество D. Далее построим множество S, такое, что $Card\ S = Card\ D$ и каждому элементу из D поставим в соответствие один элемент из S. Элементы множества S будем называть состояниями предметной области ПрО, а само множество S – множеством состояний ПрО. При измерении набора данных будет меняться и состояние ПрО.

Событием будем называть изменение состояния предметной области. Понятно, что событие вызывается изменением набора данных. *Фактами* будем называть те события, записи о которых заносятся в базу данных (возможно, это те записи, которые и привели к событию), и мы можем их проанализировать. Факты будем обозначать строчными буквами греческого алфавита, кроме того, выделим особый факт,

назовём его нулевым фактом и договоримся обозначать его буквой o, договоримся также ни один другой факт так не обозначать.

Назовём *показателем факта* его числовую характеристику, т.е. по правилам нашей предметной области каждому факту поставлено в соответствие одно или несколько чисел. Обозначать показатели фактов будем следующим образом: $f_i(\alpha) \in R$, где α – некоторый факт, а $i \in N$. Договоримся, что если это специально не оговорено, то все показатели нулевого факта равны 0. Адресной функцией Adr, будем называть такое биективное отображение, что:

1. Прообразом этого отображения является множество $\left\{\left\langle x^1, x^2, ..., x^m \right\rangle \mid x^i \in X^i\right\}$, где $m \in N$, X^i – множество точек измерения.
2. Образом этого отображения является некоторое множество фактов.

В одной и той же предметной области могут рассматриваться факты различной природы. Факты, на возникновение которых по правилам данной предметной области следует реагировать одинаковым образом, будем объединять в *группы* (*классы фактов*). Классом фактов назовём такое множество фактов A, что:

1. $(\exists R)(\forall \alpha \in \mathrm{A}) : \alpha \in D(R)$, где R – некоторая реакция предметной области на некоторую группу фактов, $D(R)$ – область определения реакции предметной области (множество фактов которые составляют ту группу фактов, на которую распространяется определённая реакция предметной области).
2. Существует такая адресная функция, что её образ совпадает с A.
3. У фактов в классе одинаковый набор показателей, т.е. показатели фактов в рамках класса это отображения класса на множество действительных чисел.

Таким образом, класс фактов составляют не только те факты, записи о которых имеются в базе данных, но и все те, которые по правилам предметной области могут иметь место, и реагировать на них следует одинаково. Договоримся классы фактов обозначать прописными буквами греческого алфавита. Пусть имеется некоторый класс фактов A, через $\overline{\mathrm{A}}$ будем обозначать подмножество A такое, что в него входят только те факты, записи о которых уже имеются в БД.

Кубом будем называть множество $Cube(\overline{A}) = \overline{\mathrm{A}} \bigcup \{o\}$, где A – некоторый класс фактов. Кроме того, договоримся, что o имеет тот же набор показателей, что и факты из A.

Агрегирующей функцией назовём отображение, которое совокупности действительных значений M ставит в соответствие действительное значение: $\varphi(M) \in R$. Пару «показатель факта – агрегирующая функция» будем называть *параметром факта*. Параметры факта будем обозначать следующим образом: $F_i^{\alpha} =< f_j(\alpha), \varphi_k >$, где $i, j, k \in N$, α – некоторый факт, $f_j(\alpha)$ – показатель факта, φ_k – некоторая агрегирующая функция.

Адресная функция не позволяет получить агрегированные данные, а позволяет получить лишь конкретный факт, поэтому построим такое отображение, которое позволяло бы по элементу набора измерений получить агрегированные данные.

Пусть у нас имеется некоторый куб $Cube(\overline{\mathrm{A}})$, для которого мы построили адресную функцию $Fact_{\overline{\mathrm{A}}}^{-}$. Рассмотрим прообраз этой функции: $D(Fact_{\overline{\mathrm{A}}}^{-}) = \left\{< x^1, x^2, ..., x^m > \mid x^i \in X^i\right\}, m \in N$, по определению X^i – некоторое множество точек измерения. Для каждого множества точек измерения X^i построим измерение $\tilde{\tilde{X}}^i$, получим набор измерений $\breve{X} = \left\langle \tilde{\tilde{X}}^1, \tilde{\tilde{X}}^2, ..., \tilde{\tilde{X}}^m \right\rangle$; данный набор измерений будем называть *системой координат куба* $Cube(\overline{A})$, а элемент этого набора измерений – *координатой*

ячейки куба; *нулевой координатой* будем называть такую координату, в которой хотя бы один элемент является пустым множеством и обозначать её будем $\breve{0}$.

Таким образом, адресная функция с возможностью агрегации имеет вид

$$Fact_{\overline{A}}(\breve{x}) = \begin{cases} Adr(< x^1, x^2, ..., x^m >), < x^1, x^2, ..., x^m > \in \overline{D}(Adr) \\ o, < x^1, x^2, ..., x^m > \notin \overline{D}(Adr) \\ o, \breve{x} = \breve{0} \end{cases}$$

$$\breve{x} =< x^1, x^2, ..., x^m >$$

Осталось ввести понятие самой *ячейки куба*. Выберем один из показателей для фактов куба $Cube(\overline{A})$, пусть это будет f_i. Возьмём в качестве агрегирующей функции φ_k. Построим для куба систему координат (построим измерения, которые составляют систему координат по множествам точек измерения из прообраза адресной функции). Пусть это будет $\breve{X} = \left\langle \tilde{\tilde{X}}^1, \tilde{\tilde{X}}^2, ..., \tilde{\tilde{X}}^m \right\rangle$, $m \in N$. Возьмём произвольно координату ячейки куба $\breve{x} = \left\langle X^{11}, X^{21}, ..., X^{m1} \right\rangle$, $X^{i1} \in \tilde{\tilde{X}}^i$, $i = \overline{1,m}$ и построим для этой координаты множество $A = \left\{ < x^1, x^2, ..., x^m > | \, x^i \in X^i \right\}$. Построим теперь на основании множества A множество $B = \left\{ f_i(\alpha) \mid \alpha = Fact_{\overline{A}}(< x^1, x^2, ..., x^m >), < x^1, x^2, ..., x^m > \in A \right\}$. Значение $\varphi_k(B)$ и будет ячейкой куба.

Таким образом, ячейкой куба $Cube(\overline{A})$ для показателей фактов f_i, агрегирующей функции φ_k с координатой $\breve{x} = \left\langle X^{11}, X^{21}, ..., X^{m1} \right\rangle$ в системе координат $\breve{X} = \left\langle \tilde{\tilde{X}}^1, \tilde{\tilde{X}}^2, ..., \tilde{\tilde{X}}^m \right\rangle$ будем называть значение:

$$Cell^{f_i, \varphi_k}_{Cube(\overline{A})}(\breve{x}) = \varphi_k\left(\left\{ f_i(Fact_{\overline{A}}(< x^1, x^2, ..., x^m >)) \mid x^i \in X^{i1} \right\} \right).$$

Агрегация показателей

Построим некоторое конечное, упорядоченное множество $M =< x_1, ..., x_n >$. Будем его называть *массивом агрегатов*. Договоримся, что для любого массива агрегатов существует отображение $\chi : M \to Y \subseteq R$. *Агрегатом* будем называть число $M[i] = \chi(x_i)$, $x_i \in M$, а i *– номером агрегата*.

При формализации многомерной модели данных мы договорились, что агрегирующей функцией будем называть отображение $\varphi : \mathcal{P}(R) \to R$. Уточним определение: будем под агрегирующей функцией понимать такое отображение $\varphi : \mathcal{P}(M) \to R$, где M – массив агрегатов. Кроме того, сформулируем такую *аксиому агрегирующей функции*: её результат не зависит от выбранного порядка в *M*. Автор предлагает несколько *методов агрегации* (*схем*).

Первой такой схемой является *итерационная схема агрегации*. Договоримся, что *Card M* = *n*, коме этого пусть *n* > 1, полагая $\varphi(\{x\}) = \chi(x)$. Для $\varphi(M)$ найдём такую функцию $\psi(y, x, i) \in [R]$, $y, x \in R, i \in N$, что $\varphi(M) = y_n$, где

$$\begin{cases} y_1 = M[1] \\ y_i = \psi(y_{i-1}, M[i], i) \end{cases}$$

Функцию ψ будем называть *итерирующей функцией* для агрегирующей функции φ.

Что же даёт нам использования итерационной схемы агрегации? Не секрет, что в целях анализа могут понадобиться различные агрегирующие функции и все их предусмотреть невозможно, поэтому при

разработке OLAP-средства необходимо учесть потребность в использовании пользовательских агрегирующих функции и предоставить средства для их разработки. Конечно, можно реализовать средство для разработки всей агрегирующей функции, но при этом оно было бы достаточно сложным, как для реализации, так и для использования, а можно воспользоваться итерационной схемой агрегации, при этом достаточно предоставить возможность проектирования только итерирующей функции, для проектирования которой вполне достаточно средств элементарной математики.

В итерационной схеме агрегации мы за каждый из n шагов обрабатываем всего один элемент массива агрегатов M. С целью повысить производительность построим такую схему, при которой за каждый шаг агрегировалось хотя бы два элемента.

Для начала, для агрегирующей функции φ построим итерирующую функцию второго порядка: $\psi^2(y, x_1, x_2, i) \in [R],\ y, x_1, x_2 \in [R],\ i \in N$, такую, что

$$\varphi(M) = \psi^2(y_{n-2}, M[n-1], M[n], n)$$

$$\begin{cases} y_1 = M[1] \\ y_i = \psi^2(y_{i-2}, M[i-1], M[i], i) \end{cases}$$

Такая схема, выдвигает два требования к множеству M:

1) $n > 2$;
2) *n* – нечётно.

Со вторым требованием можно «справиться» таким образом: если n – чётно, то добавить в качестве $M[n+1]$ ноль, так же можно поступить в случае невыполнения первого требования, т.е. дополнить M нулями до $CardM = 3$, единственное условие, что это необходимо будет учесть при разработке итерирующей функции.

Основную сложность в разработке итерирующих функций второго порядка составляет учёт добавленных нами нулей. В ряде случаев это может составить очень серьёзную проблему, поэтому предложим ещё один путь решения второй проблемы: будем использовать итерирующие функции как первого, так и второго порядков. При этом до ближайшего нечётного номера шага, не превышающего n, будем пользоваться итерирующей функцией второго порядка, а потом, если понадобится, воспользуемся итерирующей функцией первого порядка.

Обобщим теперь всё сказанное на случай *k*-го порядка итерирующей функции. Итерирующей функцией k-го порядка будем называть функцию $\psi^k(y, \bar{x}, i) \in [R]$, $y \in [R], i \in N$, $\bar{x}$ – вектор k-го порядка элементы которого – из R.

Построим следующую схему агрегации:

$$\varphi(M) = \psi^k(y_{n-k}, (M[n-k+1], \ldots, M[n]), n)$$

$$\begin{cases} y_1 = M[1] \\ y_i = \psi^k(y_{i-k}, (M[i-k+1], \ldots, M[i]), i) \end{cases}$$

Назовём эту схему *усовершенствованной итерационной схемой агрегации*. Её применение позволяет в разы снизить количество шагов, сохранив при этом преимущества итерационной схемы агрегации.

Часто при ответе на запрос системе приходится агрегировать уже агрегированные показатели (например, несколько ячеек куба). Эту задачу не получится решить, пользуясь итерационной схемой агрегации. Автор предлагает следующую схему агрегации, при которой возможна агрегация как простых агрегатов, так и уже агрегированных показателей.

Пусть имеется два массива агрегатов $M^1 =< x_1^1, x_2^1, ..., x_{n_1}^1 >, M^2 =< x_1^2, x_2^2, ..., x_{n_2}^2 >$ и агрегирующая функция φ; мы знаем значения $y_1 = \varphi(M^1)$, $y_2 = \varphi(M^2)$, $n_1 = Card\ M^1$ и $n_2 = Card\ M^2$. Требуется найти значение $\varphi(M^1 \bigcup M^2)$. Полагаем при этом, что $\varphi(\{x\}) = \chi(x)$. Будем обозначать $\psi(y_1, y_2, n_1, n_2)$ функцию агрегации агрегированных показателей y_1 и y_2; заметим, что при $n_2 = 1$ получаем итерирующую функцию первого порядка.

Итерационная схема агрегации с использованием функции агрегации агрегированных показателей выглядит следующим образом:

$$\begin{cases} y_1 = M[1] \\ y_i = \psi(y_{i-1}, M[i], i-1,1) \end{cases}$$

Чтобы агрегировать более одного агрегированного показателя, достаточно построить суперпозицию функций агрегации двух агрегированных показателей.

Заключение

В данной статье приведено формальное описание реализации многомерной модели данных в программном комплексе METAS BI-Platform на языке теории множеств и теории функций. Были описаны методы агрегации данных, позволяющие эффективно агрегировать массивы числовых показателей.

Благодарности

Работа выполнена при поддержке гранта РФФИ № 08-07-90006-Бел_а.

Библиографический список

[1] Codd E.F. , Codd, S.B., Salley C.T. Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Technical report, 1993 [PDF] (www.olap.ru).

[2] Мальцев П.А. Разработка компонента визуализации многомерных данных для CASE-технологии METAS // Технологии Microsoft в теории и практике программирования. Материалы конференции. Нижний Новгород: Изд-во Нижегородского университета, 2006. С. 202-204.

[3] Мальцев П.А., Лядова Л.Н. Формализация многомерной модели данных // Математика программных систем: Межвузовский сб. науч. тр. / Перм. ун-т. Пермь, 2006. С. 74-87.

Сведения об авторе

Павел Мальцев *– Пермский государственный университет, аспирант кафедры математического обеспечения вычислительных систем; Россия, г. Пермь, 614990, ул. Букирева, 15;*
e-mail: pavel_maltsev@mail.ru