

О ТРЕБОВАНИЯХ К ИНТЕЛЛЕКТУАЛЬНОЙ МОДЕЛИ ДАННЫХ

Сергей Львович Гладков, технический директор,

E-mail: gladkovs@list.ru,

Общество с ограниченной ответственностью «Айго»,

http://www.aigeo.ru

В работе введено понятие о модели баз данных, состоящей из предиката (пропозициональной функции) и множества данных, как интерпретации предиката.

Ключевые слова: предикаты баз данных, модели баз данных, концептуальная модель данных, семантика баз данных, онтологии.

Введение

Традиционные системы баз данных, независимо от типа модели (дореляционной, реляционной или постреляционной), обладают общим недостатком: множество концептов (понятий, терминов, сущностей) составляющих их метаданные должно быть предопределенным. То есть, не прибегая к программированию или внесению изменений в метаданные, нельзя динамически создать новый (виртуальный) концепт. Концепт, его атрибуты и отношения включаются в состав метаданных при помощи языка описания данных, обозначаемого DDL¹, и остается неизменным до тех пор, пока в это описание не будет внесено изменение. Поэтому концепты можно определить, но ими нельзя манипулировать, в отличие от непосредственных данных, работа с которыми осуществляется при помощи языков манипулирования данными, обозначаемых DML². С другой стороны, существует ряд задач, для решения которых необходима некоторая степень манипулирования метаданными.



С.Л. Гладков

Ниже рассмотрены некоторые из них.

Пример 1. Классификация тематики книг

Таблица 1

Кодирование тематики книг в системе УДК³

Автор	Название	УДК
И. Экланд	Элементы математической экономики	51.38+519.9
А. Ахо, Дж. Хопкрофт, Дж. Ульман	Построение и анализ вычислительных алгоритмов	519.682.1+681.142.2

Таблица 1 содержит примеры книг, тематику которых нельзя отнести к одной базовой позиции классификатора. Так тематика книги «Элементы математической экономики» относится к разделам «экономика» (51.xx) и (+) «прикладная математика» (519.xx). А тематика книги «Построение и анализ вычислительных алгоритмов» относится к разделам «прикладная математика» (519.xx) и «вычислительная техника» (681.xx).

Для присвоения кода УДК книге с синтетической тематикой создается синтетический, код, как выражение над кодами базовых позиций УДК. Знак «+» в системе УДК [7] называется знаком присоединения и создает новую ad hoc позицию классификации, являющуюся обобщением тематик с кодами, расположенными справа и слева.

Пример 2. Сравнение бюджетов

¹ DDL – Data Define Language (язык описания данных).

² DML – Data Manipulation Language (язык манипуляции данными).

³ УДК (Универсальная десятичная классификация): код тематики издания в универсальной десятичной классификации.

Сравнение расходов по КБК⁴

№	Год	Код раздела КБК	Название	КБК
1	2009	900	Здравоохранение, физическая культура и спорт	2012.900+2012.1100
2	2012	900	Здравоохранение	
3	2012	1100	Физическая культура и спорт	

Таблица 2 показывает, что в 2009 году данные о расходах на «Здравоохранение, физическая культура и спорт» содержались в одном разделе (900) расходов бюджетной классификации. В 2012 году данные об этих расходах содержатся уже в двух разделах: «Здравоохранение»(900) и «Физическая культура и спорт» (1100)

Для того чтобы иметь возможность изучения динамики расходования средств на «Здравоохранение, физическая культура и спорт» из разделов 2012 необходимо создать ad hoc позицию условный раздел (900+1100), «Здравоохранение, физическая культура и спорт».

Пример 3. Слияние территорий

Таблица 3

Динамика развития региона

№	Год	Код региона	Название	КБК
1	2006	24	Красноярский край	
2	2006	84	Таймырский (Долгано-Ненецкий) автономный округ	
3	2006	88	Эвенкийский автономный округ	
4	2007	24	Красноярский край	2006.24+2006.84+2006.88

Таблица 3 показывает, что на территории Красноярского края в 2006 году располагались три субъекта федерации: Красноярский край, Таймырский (Долгано-Ненецкий) автономный округ, Эвенкийский автономный округ.

Для того чтобы иметь возможность изучения динамики экономических показателей Красноярского края» из данных 2006 необходимо создать ad hoc позицию условный субъект федерации (24+84+88), «Красноярский край».

Все примеры демонстрируют задачи, в которых необходимо из нескольких концептов (понятий, терминов, сущностей) «на лету» создать третий концепт, который является общением исходных концептов.

Для решения этой задачи на множестве концептов (понятий, терминов, сущностей) необходимо определить операции «+» и «-», позволяющие создавать новые концепты. Например, «здание» + «сооружение» – общее понятие для зданий или сооружений. А «постройка⁵» – «сооружение» общее понятие для всех построек, не являющихся сооружениями.

Шаги создания модели данных

Множество базовых концептов

Предметная область любой базы данных, т. е. множество базовых концептов (понятий, терминов, сущностей) – это динамически неизменяемая⁶ часть значений ее метаданных.

Предметная область выделяется из универсального множества концептов набором предикатов или пропозициональных функций [15]. В концептуальной модели «Сущность-связь» Питера Чена [3] концепты представляются «сущностями», предикаты – «связями». На стадии проектирования логической модели концепты и сущности представляются отношениями (таблицами, узлами и т. д.).

⁴ КБК - Код бюджетной классификации.

⁵ Постройка – то, что построено – здание, сооружение [5].

⁶ Правильнее сказать, что множество базовых концептов является слабо изменяемым в пределах возможностей языка описания данных (DDL – Data Define Language). [Прим. автора]

В логической схеме стираются различия между отношениями, созданными на основании концептов, и отношениями, соответствующими предикатам предметной области. Но это не значит, что таких отличий не существует, примерами отношений последнего типа являются «обобщение» и «агрегация», подробно описанные в статье Джона и Дианы Смит [2].

Поэтому множество базовых концептов далее рассматривается как пара, называемая далее онтологией [9]:

$$O = \langle X, R \rangle,$$

где X – конечное множество концептов предметной области, которую представляет онтология O ;

R – конечное множество отношений между концептами заданной предметной области.

Кроме того, будем считать, что для онтологии O определена одна или несколько общезначимых пропозициональных функций C , которые в дальнейшем изложении будут называться контекстом концепта, группы концептов, и всей онтологии.

Множество порождённых концептов

Определим на множестве базовых концептов X решетку⁷, как универсальную алгебру, так как это описано в монографии Биркгофа «Теория решеток» [5].

Для этого на множестве X введем две операции: «+» и «-», обладающие следующими свойствами:

- $\forall x_i, x_j | x_i \in X \text{ и } x_j \in X \Rightarrow \exists x_{g+}, x_{g-} \text{ такие что, } x_{g+}=x_i, + x_j, x_{g-}=x_i, - x_j$
- $\forall x \in X; x + x = x, x - x = x$ (идемпотентность) P1
- $\forall x, y \in X; x + y = y + x, x - y = y - x$ (коммутативность) P2
- $\forall x, y, z \in X; x + (y + z) = (x + y) + z, x - (y - z) = (x - y) - z$ (ассоциативность) P3
- $\forall x, y \in X; x - (x + y) = x + (x - y) = x$ (поглощение) P4

Операция «+» на множестве X является операцией «обобщения»⁸ имеет здесь тот же смысл что и в известной работе Джона и Дианы Смит [2], а созданный концепт x_{g+} называется «обобщением» x_i, x_j .

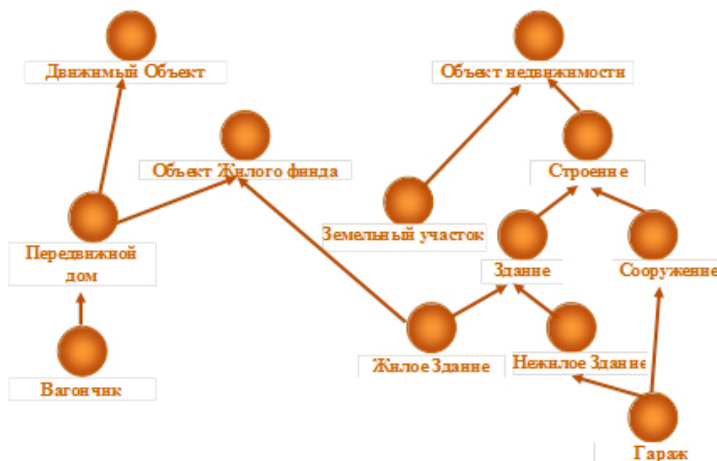


Рисунок 1 – Пример детерминированной решетки как части интеллектуальной модели управления имуществом

онтология [9]:

$$O = \langle \bar{X}, \{\vee, \wedge\}, R \rangle,$$

где \bar{X} – замыкание конечного множества концептов предметной области по отношению к операциям « \vee » и « \wedge »; R – конечное множество отношений между концептами заданной предметной области.

⁷ В «Лекциях по общей алгебре» А.Г. Куроша [4] понятие «решетка» введено под другим названием – «структура».

⁸ В теории семантических сетей для отношения «обобщение» принято обозначение *is a*.

Оси и пространство данных

Ось (s) – это обобщение атрибута, а также домена в теории реляционных баз данных. Ось определяется независимо от концептов из множества \bar{X} . Каждая ось s определяет множество значений D_s . Совокупность осей составляет универсальное пространство $S = \{s_n | 0 < n \leq N\}$.

Замечание 1. Множество осей $S = \{s_n | 0 < n \leq N\}$ – является подмножеством \bar{X} , т. е. каждая ось s_n представляет собою концепт, к которому в частности могут быть применены операции «+» и «-». Цель определения оси как концепта станет особенно понятной после рассмотрения последовательности порождения концептов (Семантика концептов).

Данные – это точки пространства S . Каждая точка представляет собою набор значений, соответствующих осям пространства: $\{d_{ij} | \forall j d_{ij} \in D_i\}$.

Определим на множестве \bar{X} отношение $*$, которое каждому концепту ставит в соответствие подпространство $S_x \subseteq S$. S_x – пространство, на котором определены данные концепта x . Пара $\{x; S_x\}$ эквивалентна отношению реляционной алгебры, где x – имя отношения, а $\{x * S_{x_i}\}$ – множество атрибутов.

Для обозначения отношения из множества концептов \bar{X} в множество данных D универсального пространства S будем использовать знак операции «композиция»⁹ для объединения концепта и соответствующего ему подпространства осей $x * S_x = D_x$.

Пусть определены композиции $x_1 * S_{x_1}$ и $x_2 * S_{x_2}$, тогда подпространство, которое соответствует обобщению $x_1 + x_2$, может быть получено пересечением подпространств исходных концептов, т. е.

$$S_{x_1+x_2} = S_{x_1} \cap S_{x_2}.$$

Поясним то, как осуществляется «композиция» с $x_1 + x_2 * S_{x_1+x_2}$. Для этого с каждой осью сопоставляется «свойство» или «метод» отображения, которое соответствует реляционным функциям группировки SUM(), MAX(), MIN(), AVG()¹⁰. Так, если оси «количество» соответствует «свойство» SUM(), то «композиция» «компьютеры + столы» * «количество» преобразуется в отношение «компьютеры» + «столы» * «количество». Каждый экземпляр (кортеж) динамически созданного отношения будет сумму значений «компьютеры» * «количество» + «столы» * «количество».

Результату этого шага должна соответствовать онтология:

$$O = \langle \overline{X * S}, \{\vee, \wedge\}, R \rangle,$$

где \bar{X} – замыкание конечного множества концептов предметной области по отношению к операциям « \vee » и « \wedge »; R – конечное множество отношений между концептами заданной предметной области.

Классификация, верификация, фальсификация концептов

Введением пространства S было задано отображение множества \bar{X} на множество значений D_s , $\forall x \in \bar{X} \exists S_x \Rightarrow S_x(x) = D_x$.

Теперь рассмотрим обратную задачу: нахождения концепта $x \in \bar{X}$, которому соответствует точка $d \in D$.

Задачи такого типа часто возникают в процессе Добычи данных (англ. **Data Mining**), например, нахождение мест остановки (стоянки) транспортного средства в результате анализа трека с ГЛОНАС (GPS) приемника, в процессе информационного обмена.

⁹ Такая композиция должна напоминать о необходимости механизма динамического связывания при реализации перехода от концепта к данным.

¹⁰ Подобный механизм реализован, например, в модели Geodatabase, реализованной в семействе продуктов ArcGIS, поставляемой Институтом исследования систем окружающей среды США от английского *Environmental Systems Research Institute (ESRI)*. Описание этого механизма можно найти в [10] или [11].

Для решения этой задачи введем понятие функции принадлежности $\mu_x(d) \in [0..1] \mid d \in D \vee x \in X$, которая точкам множества данных ставит в соответствие вероятность того, что данная точка d принадлежит концепту x .

Так как каждому концепту x поставлено в соответствие подпространство S_x , то все точки за пределами этого пространства априори не будут соответствовать концепту, т. е. $\mu_x(d) = 0 \mid \forall d \notin D_x$.

Выделим S_x такое минимальное подпространство I_x такое, что функция принадлежности принимает одно и тоже значение в точке d_i подпространства I_x , и в каждой точке S_x , проекция которой на I_x равна d_i .

Таким образом I_x – пространство минимума различающих данных для концепта x .

Тогда под классификацией точки $d \in D$ будет пониматься задача соотнесения ее одним или несколькими концептами x с использованием набора функций принадлежности $\{\mu_x(d) \mid d \in I_x\}$.

Под верификацией точки d подпространства S_x будет пониматься задача проверки соответствия этой точки концепту x при помощи функции принадлежности.

Под фальсификацией точки d подпространства S_x будет пониматься задача нахождения и внесение в d таких минимальных изменений, что поле такого преобразования точка перестает соответствовать концепту x .

Семантика концептов

Отношения между концептами заданной предметной области можно рассматривать как способ создания или порождения новых концептов. Например, концепты «город» и «улица», связанные отношением «улицы города», создают концепт «улица в городе».

Вследствие этого «созидающего» свойства отношений между концептами их удобнее использовать в функциональной форме, что и будет сделано в дальнейшем изложении без дополнительных комментариев.

Рассмотрим множество взаимно-однозначных функций порождения $R: \overline{X} \rightarrow \overline{X}$, определенное на множестве концептов. Будем называть концепт $x_n \in \overline{X}$ порожденным от концепта $x_0 \in \overline{X}$, если существует последовательность $\{r_i \in R \mid 1 \leq i \leq n\}$ такая, что $x_n = r_n * \dots * r_2 * r_1(x_0)$. Таким образом, композиция функций $r_n * \dots * r_2 * r_1$ преобразует концепт x_0 в другой концепт x_n , который в общем случае принадлежит другому множеству \overline{X} .

Положим, что среди функций множества R существует $r_{\text{пор}} \in R$ такое, что $r_{\text{пор}}(x_0) \equiv \{x_0\}$. Другими словами, функция $r_{\text{пор}}$ концепту x_0 ставит в соответствие его же самого.

Далее $r_{\text{пор}}$ будет называться нуль-порождением (0-порождением) концептов.

Ввод функции нуль-порождения дает возможность далее считать множество \overline{X} замкнутым относительно операций порождения концептов и исключить множество $\overline{\overline{X}}$, так как множество функций может быть переопределено как: $R: \overline{X} \rightarrow \overline{X}$.

Определение 1. Пусть $O = \langle \overline{X}, R \rangle$, где \overline{X} - есть множество концептов,

а $R: \overline{X} \rightarrow \overline{X}$ - множество взаимно-однозначных функций, определенных на концептах, тогда концепт $x_n \in \overline{X}$ будем называть порожденным от концепта $x_0 \in \overline{X}$, если существует последовательность $\{r_i \in R \mid 1 \leq i \leq n\}$ такая, что $x_n = r_n * \dots * r_2 * r_1(x_0)$. Таким образом, композиция функций $r_n * \dots * r_2 * r_1$ преобразует концепт x_0 в другой концепт x_n .

Замечание 1.1. Из настоящего определения нетрудно видеть, что по мере порождения концепта x_n была порождена целая последовательность концептов $x_0, x_1=r_1(x_0)$,

$x_2=r_2*r_1(x_0), \dots, x_{n-1}=r_{n-1}*\dots*r_2*r_1(x_0), \quad x_n=r_n*\dots*r_2*r_1(x_0)$. Такую последовательность $\langle x_0, x_1, \dots, x_n \rangle = P$ вслед за Чарльзом Хоаром [14] будем называть **протоколом порождения концепта x_n** .

Замечание 1.2. Так как оси образуют подмножество концептов \bar{X} , то элементом последовательности протокола порождения может быть ось подпространства S_x концепта x . Например, пусть «цвет» название одной из осей, тогда концепту «цвет кузова автомобиля» соответствует следующий протокол порождения: $x_1=r_1(x_0)$, $x_2=r_2(x_1)$. Здесь x_0 – соответствует понятию «автомобиль», x_1 – «кузов автомобиля», x_2 – «цвет кузова автомобиля».

Определение 2. Последовательность $\{x_{n-1} = r_{n-1}(x_{n-2}); x_{n-2} = r_{n-2}(x_{n-3}); \dots x_1 = r_1(x_0)\}$ будем называть контекстом концепта x_n или $C(x_n)$. Тогда концепт x_n может быть представлен тройкой: $\langle \text{Name}(x_n); r_n(x_{n-1}); C(x_n) \rangle$. Где $\text{Name}(x_n)$ – имя или идентификатор концепта, $r_n(x_{n-1})$ отношение между концептом и его контекстом.

Пусть C контекст некоторого концепта x . Рассмотрим подмножество $X_C \subseteq \bar{X}$, всех концептов принадлежащих C . Тогда контекст C может быть использован в качестве основы для построения функции принадлежности подмножества X_C , которую будем обозначать $\overline{\mu X_C}$.

Действительно, контекст C представляет собой последовательность концептов $\{x_i | 0 \leq i \leq n-1\}$, на основе которых создаются элементарные высказывания. Эти элементарные высказывания, в свою очередь, при помощи логических связок («и», «или», «если... то...», отрицания и др.) образуют общее высказывание контекста. В этом случае общее высказывание контекста станет функцией принадлежности множества $\overline{\mu X_C}$.

Контекст концептов, составляющих базу данных, позволяет решать следующий важные задачи:

1 Уменьшение избыточности базы данных за счет создания контекста и функции принадлежности базы данных.

2 Выполнение поиска и выборки данных из нескольких баз данных, для которых может быть построен общий контекст. Например, выбор данных из реестров имущества Мотыгинского и Краснотуранского районов. При этом схемы баз данных каждого из реестров в устоявшемся смысле окажутся одними и теми же. Контекст содержать 3 концепта, как и 3 функции принадлежности:

- «Страна» – Россия;
- «Регион» – Красноярский край;
- «Муниципальный район» – Мотыгинский район для первого реестра и Краснотуранский район для второго. Тогда запросы к каждому реестры должны быть дополнены своей функцией принадлежности.

Определение 3. Если для каждого концепта некоторого множества $X_C \subseteq \bar{X}$ используется один и тот же контекст C . C называется контекстом этого множества, а $\overline{\mu X_C}$ функция принадлежности этого множества. Если один и тот же контекст C используется для всех концептов базы данных DB , то C называется контекстом базы данных, а соответствующая этому контексту функция принадлежности $\overline{\mu DB_C}$ – функция принадлежности базы данных.

Замечание 3.1. Если $X_C \subseteq \bar{X}$ – множество концептов, принадлежащих контексту базы данных, то исходное множество концептов \bar{X} может быть представлено как пересечение множества концептов, принадлежащих контексту базы данных, и множества

концептов, принадлежащих базе данных, так что пересечение этих множеств будет пустым: $\bar{X} = X_C \cup X_{DB} \Rightarrow X_C \cap X_{DB} = \emptyset$

Расширенный контекст множества концептов

Целями выделения контекста С, как общей части протокола концептов базы данных заключается в том, чтобы:

Исключить описание контекста из описаний каждого концепта базы данных и хранить его отдельно;

Создать динамическую процедуру объединения концептов базы данных с контекстом в тех случаях, когда описание концептов может привести к многозначности, например, при необходимости организации взаимодействия различных баз данных.

Таким образом, концепты $\{x_{n-1} = r_{n-1}(x_{n-2}); x_{n-2} = r_{n-2}(x_{n-3}); \dots x_1 = r_1(x_0)\}$, составляющие контекст базы данных, в первую очередь, позволяют динамически увеличивать или уменьшать общие ключевые и различающие данные концептов, входящих в состав баз данных.

Концепты, составляющие контекст базы данных могут быть ключевыми и различающими концептами для справочных данных, относящихся ко всей базе данных. Например, так называемые «северные коэффициенты», увеличивающие или уменьшающие расчетную сумму зарплаты могут быть общими для всех объектов о людях базы данных. Аналогично, коэффициенты, учитывающие территориальное расположение арендуемого объекта, увеличивающие или уменьшающие величину арендной платы. Разные коэффициенты должны применяться к разным объектам аренды в зависимости от территории, на которой они расположены, несмотря на то, что сведения о территории непосредственно не входят в ключи объектов аренды.

Определение 4. Пусть множество концептов контекста базы данных $X_C \subseteq \bar{X}$, а $\bar{X}_{DB} \subseteq \bar{X}$ - множество всех концептов базы данных.

Назовем справочными концептами базы данных $X_{DIR} \subseteq \bar{X}$ концепты, порожденные от концептов контекста, но не принадлежащие множеству концептов базы данных. Другими словами $x \in X_{DIR} \Leftrightarrow x \notin X_{DB} \wedge \exists r_i | 1 \leq i \leq m \wedge \exists x_0 \in X_C \Rightarrow x = r_m * \dots * r_2 * r_1(x_0)$.

Замечание 4.1. Множества концептов базы данных, контекста, и справочных концептов – взаимно не пересекаются, т. е. $X_{DIR} \cap X_{DB} = \emptyset \wedge X_{DIR} \cap X_C = \emptyset \wedge X_{DB} \cap X_C = \emptyset$.

Заключение

В противоположность традиционным моделям данных интеллектуальная модель данных должна основываться на динамической модели концептов, позволяющей с помощью операций « \vee » и « \wedge » преобразовывать существующие и создавать новые концепты. Это свойство обеспечит адаптируемость модели как изменению смысла концептов со временем, так и коренным изменениям внешних объектов, данные о которых хранятся в информационной системе.

Концепты интеллектуальной модели данных должны обладать свойствами верифицируемости, фальсифицируемости. Другими словами интеллектуальная модель данных должна содержать механизм проверки правильности соотнесения внешних данных с тем или иным концептом на всем протяжении времени ее использования. Это свойство особенно важно для информационной системы, предназначенной для хранения и обработки научных и исторических данных.

Определение концепта интеллектуальной модели данных должно содержать контекст концепта, как механизма объединения концептов в множества. Это свойство обеспечит возможность управления созданием и расширением распределенных информационных систем с интеллектуальной моделью данных, а также организации взаимодействия информационных систем с интеллектуальной моделью данных.

Литература

1. Мейер Д. Теория реляционных баз данных / пер. с англ. М.: Мир, 1989.
2. Джон М. Смит, Диана К. Смит. Абстракция баз данных: Агрегация и обобщение // СУБД. 1996. № 2. С. 141–160 (Постоянное место хранения на сайте Центра Информационных Технологий («ЦИТ», «ЦИТ Форум») URL: http://citforum.ru/database/classics/abstraction_generalization/)
3. Питер Пин-Шен Чен. Модель «Сущность-связь» - шаг к единому представлению данных // СУБД. 1995. № 3. С. 137–158. (Постоянное место хранения на сайте «ЦИТ Форум». URL: <http://citforum.ru/database/classics/chen/>)
4. Курош А.Г. Лекции по общей алгебре. М.: Наука; Главная редакция физико-математической литературы, 1973. С. 178–219.
5. Биркгоф Г. Теория решеток: пер. с англ. М.: Наука; Главная редакция физико-математической литературы, 1984.
6. Чёрч А. Введение в математическую логику / пер. с англ. М.: Издательство иностранной литературы, 1960.
7. Универсальная десятичная классификация / Государственный комитет Совета Министров СССР по науке и технике. Межведомственная комиссия по классификации. Государственная публичная научно-техническая библиотека СССР. Среднее издание. М., 1969.
8. Нагаев Р.Т. Недвижимость: энциклопедический словарь. Казань: Изд-во ГУП «ПИК «Идел-Пресс», 2003. 1088 с.
9. Болотова Л.С. Системы искусственного интеллекта: Модели и технологии, основанные на знаниях: учебник. М: Финансы и статистика, 2012. 664 с.
10. ESRI Press (Author), Modeling Our World: The ESRI Guide to Geodatabase design , Chapter 5 «Smart features»
11. ESRI Press (Author), Building a Geodatabase: ArcGIS 9 [Paperback], Chapter 5 «Subtypes and attribute domains».
12. Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы: Обзор // Новости искусственного интеллекта. 1998. № 2. С. 3–19.
13. Коньшева Л. К., Назаров Д. М., Основы теории нечетких множеств: учебное пособие. СПб.: Питер, 2011. 192 с.
14. Хоар Ч. Взаимодействующие последовательные процессы / пер. с англ. М.: Мир, 1989.
15. Клини С. Математическая логика / пер. с англ. М.: Мир, 1973.

On the requirements for intelligent data model

Sergey L'vovich Gladkov, technical Director technical director Aūgeo Limited

The concept of the model in the predicate calculus [15] includes a pair: the predicate (propositional function [6]) and set as an interpretation of the predicate. From this point of view, the database is an interpretation of predicates that are not explicitly specified. At the same time, the explicit inclusion of the predicate in the description of the database models useful for solving a large range of practical tasks. This paper studies the features of the predicates that should be added to the traditional description of the database models.

Keywords: Predicates databases; Model databases; Conceptual data model; The semantics of the database; Ontology.