

Управление контекстом при организации интеллектуализированного интерфейса БД в системах моделирования на основе концептуального подхода

А. А. Зуенко, А. Я. Фридман

ИИММ ТП КолНЦ РАН, Апатиты

1. Введение

При создании многих программных систем, в частности, систем моделирования, возникает необходимость интеграции базы данных (БД) с остальными программными модулями системы. Цель интеграции — наделить компонент взаимодействия с БД некоторыми дополнительными возможностями семантического анализа информации предметной области. Такой анализ может касаться как самой БД, так и запросов, адресованных к ней. Например, проведение анализа корректности запроса может гарантировать адекватность его результатов с точки зрения различного рода ограничений, накладываемых как на источник данных, так и на функциональные возможности приложения, реализующего запрос. При выполнении семантического анализа запросов необходимо учитывать специфику объектов, участвующих в обработке информации БД. Для представления такой информации об объектах в настоящей работе применяется контекстный подход [1].

Предложена формализация контекстов различных объектов взаимодействия и метод совместного использования этих контекстов для семантического анализа запросов. Применение методологии управления контекстом для обработки данных в системах моделирования рассмотрено на примере системы ситуационного концептуального моделирования (ССКМ) [2].

2. Контекстный подход к обработке информации в системах концептуального моделирования

Рассмотрим возможности модификации методологии управления контекстом [3] при организации интеллектуализированного интерфейса обработки данных моделирования в системах моделирования на основе концептуального подхода. В рамках модифицированной методологии в обработке данных моделирования (ОДМ) участвуют следующие объекты: задача, прикладная программа (приложение), запрос приложения и база данных моделирования (БДМ), предназначенная для хранения элементов концептуальной модели предметной области и результатов процесса имитации (рис. 1).

Задачами любой системы моделирования на основе концептуального подхода являются построение открытой для оперативных модификаций концептуальной модели предметной области, осуществление автоматических процедур проверки корректности модели и так далее (см. [2]). Открытость модели предметной области предполагает возможность подключения новых программных модулей для обработки информации. Вследствие этого одна и та же задача может быть реализована разным набором *приложений*. Каждое приложение для обработки данных использует определенный набор специализированных *запросов*. Для того чтобы структура запроса отражала структуру БДМ, которая является иерархической, в работе [4] предлагается использовать путевые (адресные) запросы. Адекватность структуры запросов и *БДМ* позволяет сильно упростить их анализ [5, 6].

Все перечисленные объекты ОДМ характеризуются собственным контекстом. Таким образом, контекст взаимодействия складывается из контекста задачи, контекста приложения, контекста запроса и контекста БДМ, каждый из которых, в свою очередь, разделяется на абстрактный и прикладной контексты. Абстрактный контекст объекта соответствует интенционалу этого объекта. Прикладной контекст объекта позволяет описать некоторое множество его экстенционалов.

Контексты при их совместном рассмотрении накладывают дополнительные ограничения как на запрос, адресованный к БДМ, так и на струк-

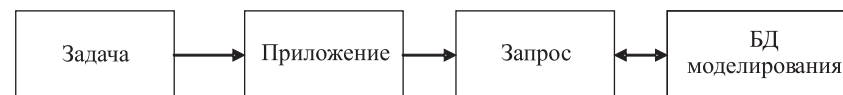


Рис. 1. Объекты ОДМ

туру самой БДМ. Это дает возможность, например, производить различные процедуры проверки корректности запроса до стадии его исполнения.

Контекст БД моделирования. Абстрактный и прикладной контексты БДМ предлагается задавать с помощью формального аппарата онтологии структурных ограничений, модель которой приводится ниже. Интенционал онтологии структурных ограничений описывает класс концептуальных моделей, которые являются допустимыми в рамках той или иной системы моделирования, и соответствует абстрактному контексту БДМ. Прикладной контекст БДМ задается посредством наложения на интенционал онтологии дополнительного набора ограничений, которые специфичны для некоторого экстенционала онтологии, то есть для отдельной концептуальной модели.

При описании контекстов других объектов взаимодействия активно используются концепты онтологии структурных ограничений и их свойства.

Контекст запроса. В работе [6] показано, что любой адресный запрос может быть разложен в набор конъюнктивных. Абстрактный контекст конъюнктивного адресного запроса представим в виде неориентированного графа G , где ребра соответствуют связям, не описанным в онтологии, а каждая вершина является древовидным ориентированным графом W , где присутствуют только регламентированные онтологией связи. Прикладной контекст адресного запроса получается путем назначения вершинам графа W условий фильтрации, представимых в виде логических формул над предикатами первого порядка. Другими словами, прикладной контекст адресного запроса получается из абстрактного после назначения атрибутам уровней запроса конкретных значений (где это необходимо).

Контекст приложения описывает возможности прикладной программы. Основным свойством этого контекста является список запросов, которые могут быть выполнены приложением.

Контекст задачи определяется совокупностью контекстов приложений, необходимых для ее осуществления. При этом исключаются дублирующие запросы и, соответственно, многократная проверка их корректности.

Что представляет собой контекст запроса, подробно описано в работе [6]. Далее рассмотрим процесс формирования абстрактного и прикладного контекстов БДМ. Для формализации этих контекстов используется созданная авторами модель онтологии структурных ограничений.

3. Модель онтологии структурных ограничений

Потребность в создании новой модели онтологии для систематизации знаний о БДМ обусловлена следующими причинами. Известные стандар-

тизированные модели онтологий и схем данных (OWL, XML Schema и т. д.) позволяют накладывать определенные ограничения только на отдельные концепты онтологии, приписывая их атрибутам определенные значения, поэтому для формализации таких онтологий применяется логика первого порядка. Однако в ряде случаев необходимо формализовать ограничения как на отдельные связи, так и на наборы связей между концептами. Такие ограничения можно рассматривать как условия существования связей онтологии. Очевидно, что для описания подобных ограничений недостаточно формул над предикатами первого порядка, то есть нельзя обойтись средствами, предоставляемыми известными и широко распространенными моделями онтологий. Даже в XML-документах, которые предназначены для работы со слабоструктурированными данными, отсутствует поддержка логических формул, в результате чего нельзя удовлетворительно задавать ограничения на иерархическую структуру документа в следующих случаях:

- появление необязательного атрибута/элемента влияет на значение другого атрибута/элемента;
- значение одного атрибута/элемента обуславливает наличие или значение другого атрибута/элемента;
- наличие ссылки внутри документа с жестко фиксированной структурой определяется исходя из значений отдельных атрибутов/элементов документа.

Вместе с тем, такого рода формулы, как будет показано далее, способны описывать достаточно широкий класс ограничений, которые например, накладываются на отношения концептуальной модели. Необходимость в использовании ограничений описываемого класса появилась, в частности, при создании онтологии структурных ограничений в рамках системы ситуационного концептуального моделирования (ССКМ) [2]. На примере этой онтологии ниже в работе иллюстрируется подход к описанию ограничений на наборы связей между концептами.

Предлагаемая модель онтологии структурных ограничений имеет вид:

$$S_o ::= \langle CL, A, AT, CLA, AAT, AGR, REF, CARD, COND \rangle, \quad (1)$$

где: $CL ::= \{cl_i\}$ — множество классов, классы одновременно являются агрегатами атрибутов и семейства возможных экземпляров, то есть можно сказать, что $cl_i = \{ex_m^i\}$, где ex_m^i — некоторый экземпляр класса cl_i ;

$A ::= \{a_j\}$ — множество атрибутов классов;

$AT ::= \{at_k\}$ — множество типов данных атрибутов, включающее элементарные типы $\{string, real, integer, boolean\}$ и типы, производные от элементарных, то есть получающиеся посредством их усечения; тип

данных at_k трактуется как множество допустимых значений, то есть $at_k = \{v_p^k\}$, где v_p^k — некоторое допустимое значение для типа at_k , причем множество допустимых значений не обязано быть конечным;

$CLA \subseteq CL \times B(A)$ — отношение «класс — атрибуты класса»;

$AAT \subseteq A \times AT$ — отношение «атрибут — тип атрибута»;

$AGR ::= \{agr_i\} \subseteq CL \times CL$ — множество отношений агрегации между классами (иерархическое отношение «часть-целое»);

$REF ::= \{ref_j\} \subseteq CL \times CL$ — множество связей (ссылочных, ассоциативных) между классами;

$CARD \subseteq (REF \cup AGR) \times TCD$ — отношение «связь-кардинальность», где $TCD ::= \{tcd_k\} ::= \{oo, om\}$ — кардинальность (множественность) связи (варианты сочетаний «one» и «many»);

$COND ::= \{cond_m\}$ — множество ограничений (система ограничений) на связи, формат представления $cond_m$ описан ниже.

Некоторые отношения, описанные в рассмотренной модели онтологии, удобно представлять в виде частичных функциональных отображений:

$cla : CL \rightarrow B(A)$, $\{a_j\} = cla(cl_i)$, где cl_i — некоторый класс, $\{a_j\}$ — множество его атрибутов;

$aat : A \rightarrow AT$, $at_j = aat(a_i)$, где a_i — некоторый атрибут, at_j — тип этого атрибута;

$card : (REF \cup AGR) \rightarrow TCD$, $tcd_j = card(ref_i \vee agr_i)$, где в качестве аргумента выступает либо ref_i — некоторая ссылочная связь, либо agr_i — некоторое отношение агрегации, а tcd_j — кардинальность этой связи.

Классы в описанной модели представляют собой одновременно семейства и агрегаты других классов. При помощи отношений агрегации классы можно выстраивать в иерархии. Отношения агрегации соответствуют иерархии контейнеров, принятой в некоторых языках объектно-ориентированного программирования [7].

Атрибуты являются неделимыми информационными объектами. Атрибутам сопоставляется тип. Тип трактуется как множество допустимых значений атрибута, поскольку для задач данной работы не существенны ни лексическое представление типа, ни множество операций, применимых к некоторому типу.

Отношения REF используются, прежде всего, для моделирования связей между сортами элементов концептуальной модели (ее фрагментов) и поэтому предполагаются функциональными, поскольку каждое отношение схемы модели (ее фрагментов) можно отнести к одному из двух видов: $el_i \rightarrow B(el_j)$ или $el_i \rightarrow el_j$, где el_i, el_j — некоторые множества элементов концептуальной модели. Подобно модели онтологии, принятой в OWL, в настоящей модели используются ограничения на отношения. Но в отличие от OWL для задания семантических ограничений на отношения

здесь могут также применяться логические формулы над элементарными предикатами первого и второго порядков без кванторов. Каждому отношению $ref_i \in REF$ может быть сопоставлен некоторый набор логических формул, задающих ограничения на множества отбытия (cd_x) и прибытия (cd_y) функционального отношения $ref_i : cd_x \rightarrow cd_y$. Выполнение этих логических ограничений (аксиом) для экземпляров $ex_m^x \in cd_x$ и $ex_n^y \in cd_y$ означает, что экземпляр ex_m^x класса cd_x ссылается на экземпляр ex_n^y класса cd_y , а невыполнение указывает на отсутствие ссылочной связи между ними. Таким образом, набор ограничений задает условие существования связи $ref_i : cd_x \rightarrow cd_y$ между всеми возможными экземплярами классов cd_x и cd_y .

Также для связей $ref_i \in REF$ можно задавать кардинальность. Кардинальность указывает точное количество элементов в связи при условии ее наличия, то есть при выполнении логических ограничений.

Отношения AGR отражают иерархию данных вычислительного эксперимента. Для них не задается логических ограничений, поскольку структура данных вычислительного эксперимента должна быть жестко определена, то есть не должна содержать «структурных нулей». Отношения агрегации устанавливают связь между одним классом, который будем называть владельцем, и другим классом, который будем называть свойством. Свойство соответствует слабой сущности в терминах ER диаграмм. На связь «часть-целое» (или «владелец-свойство») не накладывается никаких дополнительных ограничений, поскольку факт существования экземпляра владельца (например, наличие записи в определенной таблице) определяет и факт существования экземпляра свойства. Для отношений, принадлежащих множеству AGR, могут задаваться ограничения кардинальности.

Далее приведем формальное описание системы ограничений разработанной онтологии с использованием многосортной алгебры, определений факта и ситуации, представленных в работе [6].

В рассматриваемой задаче в качестве множества сортов S многосортной алгебры примем множество классов онтологии.

Определение 1. Ограничением отношения онтологии назовем пару $\langle decl, expr \rangle$, в которой $decl$ — ситуация, называемая декларацией переменных (переменные соответствуют классам онтологии), $expr$ — логическая формула над элементарными предикатами первого и второго порядка без кванторов, где переменные — атрибуты классов онтологии, присутствующих в декларации.

Определение 2. Системой ограничений назовем конечное множество кортежей $COND = \{\langle decl, expr \rangle\}$.

Определенная таким образом онтология является неоднородной семантической сетью в терминологии работы [8] (фактически интенционал сети). Неоднородной экстенциональной семантической сетью будет сеть следующего вида:

$$H_o ::= \langle E, A, AT, CLA, AAT, AGR, REF, CARD, COND \rangle,$$

которая получается из схемы S_o (1) путем следующих преобразований:

1. Каждому классу $cl_i \in CL$ поставим в соответствие некоторое подмножество одного из декартовых произведений $e \subseteq at_{k1} \times at_{k2} \times \dots \times at_m$, где $at_i \in aat(cla(cl_i))$; e будем называть экстенсионалом (или множеством экземпляров, или множеством примеров) класса cl_i . Множество экстенсионалов всех классов из CL обозначим через E .
2. В схеме (1) заменим CL на E .
3. Определим на E^2 отношения из S_o .

Экстенсионалы онтологии хранятся в базе данных моделирования и представляют собой информацию, относящуюся к некоторой концептуальной модели. Далее на примере БДМ ССКМ иллюстрируются понятия абстрактного и прикладного контекстов базы данных моделирования как объекта ОДМ.

4. Абстрактный контекст БДМ ССКМ

ССКМ предназначена для моделирования и прогнозирования поведения сложных промышленно-природных комплексов (ППК). Концептуальная модель (КМ) исследуемого объекта в рамках ССКМ используется не только для декларативного описания сущностей предметной области и связей между ними, но и является основным средством организации и управления всем процессом моделирования.

Онтология структурных ограничений ССКМ отражает следующие основные факты о классификации и отношениях элементов КМ (см. также [4, 6]). Концептуальная модель (класс М) состоит из множеств объектов (класс O^M), процессов (класс P^M) и ресурсов (класс R^M). Объекты формализуют организационную структуру исследуемого ППК. Каждый объект является оболочкой набора процессов — преобразователей данных (ресурсов). Как объекты, так и процессы связаны между собой потоками ресурсов. Над множествами определены отношения принадлежности (например, ProcObj — процессы, приписанные к объекту), отношения типа «вход-выход» (например, InResObj — входные ресурсы объекта, ProcOutRes — выходные ресурсы процесса).

4.1. Основные классы

В качестве классов предлагается использовать следующие понятия, формирующие концептуальную модель:

M — класс «модель», который трактуется как множество всех возможных моделей объекта исследования, созданных на основе схемы КМ;
 O^M, P^M, R^M — классы, соответствующие понятиям объекта, процесса и ресурса.

4.2. Свойства классов

Далее приведены описания только тех свойств, которые непосредственно используются при формировании ограничений на связи между классами. Эти свойства выводятся из схемы концептуальной модели. Прежде всего, к таким свойствам относятся атрибуты объектов, процессов, ресурсов модели. В связи с этим, ниже не приводится описание доменов атрибутов, поскольку они либо очевидны, либо описаны в схеме КМ.

Класс O^M : $O^M.oname$ — имя объекта; $O^M.typedec$ — тип декомпозиции объекта; $O^M.usertype$ — пользовательский тип; $O^M.typeexec$ — тип исполнителя объекта; $O^M.nameexec$ — имя исполнителя объекта; $O^M.cat$ — категория объекта.

Класс P^M : $P^M.pname$ — имя процесса; $P^M.usertype$ — пользовательский тип процесса; $P^M.typeexec$ — тип исполнителя процесса; $P^M.nameexec$ — имя исполнителя процесса; $P^M.cat$ — категория процесса.

Класс R^M : $R^M.rname$ — имя ресурса; $R^M.usertype$ — пользовательский тип ресурса; $R^M.typeexec$ — тип исполнителя ресурса; $R^M.nameexec$ — имя исполнителя ресурса; $R^M.cat$ — категория ресурса; $R^M.proctype$ — программный тип ресурса.

4.3. Отношения и ограничения на связи между классами

Отношения между классами и некоторые ограничения, накладываемые на них, удобно представлять в виде ориентированного графа бинарных отношений $G ::= \langle V, E \rangle$, где множество вершин V — это множество классов онтологии (то есть $V = CL$), а множество ребер E является объединением множеств AGR и REF (то есть $E = AGR \cup REF$).

Далее перечислим наиболее существенные отношения семантической сети:

$Hierarchy \subseteq O^M \times B(O^M)$ — отношение иерархии между объектами модели;

$ProcObj \subseteq O^M \times B(P^M)$ — отношение «объект модели — процесс модели»;

$InResProc \subseteq P^M \times B(R^M)$ — отношение «процесс модели — его входные ресурсы»;

$OutResProc \subseteq P^M \times B(R^M)$ — отношение «процесс модели — его выходные ресурсы»;

$InResObj \subseteq O^M \times B(R^M)$ — отношение «объект модели — его входные ресурсы»;

$OutResObj \subseteq O^M \times B(R^M)$ — отношение «объект модели — его выходные ресурсы».

На некоторые из описанных отношений накладываются логические ограничения в виде формул над элементарными предикатами первого и второго порядков без кванторов.

В качестве примера приведем предикатную формулу, описывающую ограничения, которые неформально можно выразить следующим образом [9]:

- 1) Имена соответствующих однотипных ресурсов на входах и выходах процессов-буферов (тип «buf») должны быть одинаковыми.
- 2) Имена соответствующих однотипных ресурсов на входах и выходах процессов-разветвителей и преобразователей (тип «bra» и «tra») должны быть различными.
- 3) На входные ресурсы процессов с пользовательским «рго» типом ограничений не накладывается.

Ниже приведем формальное описание данного ограничения

$$cond = \langle decl, expr \rangle :$$

$$decl = inresproc(R^{in}, P) \wedge outresproc(R^{out}, P) \wedge \text{тип}(P, \text{процесс}) \wedge$$

$$\wedge \text{тип}(R^{in}, \text{ресурс}) \wedge \text{тип}(R^{out}, \text{ресурс});$$

$$expr = (P.usertype = 'buf')(R^{in}.usertype = R^{out}.usertype)$$

$$(R^{in}.rname = R^{out}.rname) \vee (((P.usertype = 'bra') \vee$$

$$\vee (P.usertype = 'tra'))(R^{in}.usertype = R^{out}.usertype)$$

$$(R^{in}.rname \neq R^{out}.rname) \vee (P.usertype = 'pro').$$

5. Формирование прикладного контекста БДМ ССКМ

Выше было показано, как задавать ограничения, которые свойственны всем концептуальным моделям, то есть ограничения интенционала онтологии. Эти ограничения формируют абстрактный контекст БДМ. Теперь опишем, как происходит генерация прикладного контекста БДМ.

В отличие от ограничений интенционала, ограничения, специфичные для каждой конкретной КМ, не задаются экспертом, а формируются

автоматически. Они появляются в результате использования в аппарата пользовательских типов ее элементов [9].

Ресурсы в модели могут иметь только целочисленные, вещественные или строковые значения. Пользовательские типы ресурсов позволяют на основании этих базовых типов создавать новые типы данных, которые получаются усечением базовых. Здесь можно провести аналогию с диапазоном значений в OWL. Для строкового ресурса пользовательский тип можно охарактеризовать следующим образом:

< имя_пользовательского_типа_ресурса >

< список_возможных_значений > .

Для целочисленных ресурсов пользовательский тип выглядит так:

< имя_пользовательского_типа_ресурса > < a > < b > < h > ,

где a и b — соответственно нижняя и верхняя границы интервала возможных значений типа, h — шаг, задающий промежуток между возможными значениями.

Пользовательский тип для вещественных ресурсов имеет следующий формат:

< имя_пользовательского_типа_ресурса > < a > < b > < ε > ,

где a и b интерпретируются так же, как и у целочисленных ресурсов, а $ε$ — точность вычислений (количество разрядов после запятой, которые учитываются в вычислениях).

Общим в определениях целочисленных и вещественных пользовательских типов является то, что их возможные значения могут принадлежать лишь одному непрерывному интервалу.

Фреймы, задающие пользовательские типы ресурсов, используются для формирования ограничений на хранимые значения тех или иных ресурсов.

Поскольку объекты могут рассматриваться как комплексные процессы (агрегаты процессов), то для объектов и процессов фреймы, представляющие пользовательские типы, описываются следующим образом:

< имя_пользовательского_типа >

< список_типов_входных_ресурсов >

< список_типов_выходных_ресурсов > .

Набор используемых пользовательских типов элементов специфичен для каждой конкретной КМ. Анализируя конъюнктивный адресный запрос (его прикладной контекст), который, как правило, работает с данными единственной модели, можно сформировать списки пользовательских

типов элементов этой модели. Далее информация о типах представляется в виде системы ограничений, где каждое ограничение включает логические формулы над элементарными предикатами первого и второго порядков без кванторов. Например, списки типов входных и выходных ресурсов формируют дополнительные ограничения на отношения онтологии «объект (процесс) — входные ресурсы», «объект (процесс) — выходные ресурсы». Эти ограничения определяются составом конкретной модели предметной области, а не являются следствием схемы концептуальной модели. Объединяясь с системой ограничений абстрактного контекста, такие ограничения образуют прикладной контекст БДМ.

6. Формирование обобщенного контекста запроса

Теперь рассмотрим использование контекстного подхода для решения задачи анализа корректности запросов к БДМ. Обобщенным контекстом назовем контекст запроса, на который накладываются контексты всех других объектов ОДМ (задачи, приложения, БДМ).

Процесс формирования обобщенного контекста (рис. 2) происходит следующим образом. После предварительных преобразований конъюнктивного запроса, описанных в работе [6], осуществляется генерация прикладного контекста БДМ. Системы ограничений как абстрактного, так и прикладного контекстов БДМ оперируют переменными, то есть являются схемами ограничений, а оптимизированный запрос представляет собой ситуацию, содержащую исключительно константы. Формируя всевозможные подстановки на основании конкретного запроса (его прикладного контекста), получаем набор конкретизированных ограничений контекста БДМ.

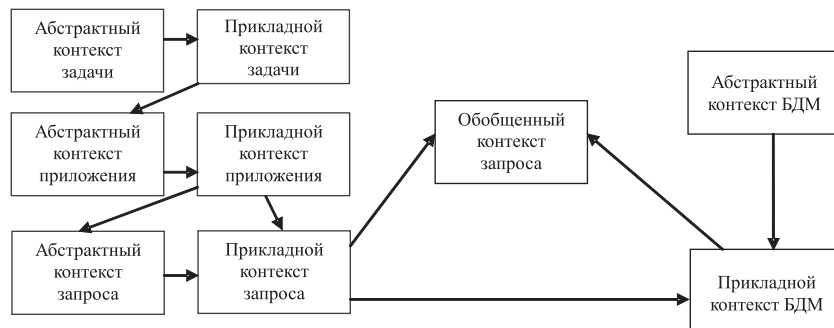


Рис. 2. Схема формирования обобщенного контекста запроса

Накладывая эти ограничения на прикладной контекст запроса, получаем некий обобщенный контекст. Если налагаемые ограничения не противоречат друг другу и ограничениям самого запроса, то запрос является корректным с точки зрения структурных ограничений онтологии. В противном случае запрос признается некорректным и не поступает на исполнение СУБД.

Таким образом, на основе одновременного учета контекстов всех объектов, участвующих в обработке данных БДМ, можно производить анализ корректности запроса до стадии его исполнения посредством формирования некоторого обобщенного контекста.

7. Заключение

В статье описан контекстный подход к представлению информации об объектах, участвующих в обработке данных моделирования в системах на основе концептуального подхода. Для формализации контекста разработана модель онтологии структурных ограничений, которая позволяет описывать ограничения как на отдельные связи онтологии, так и на группы связей в виде логических формул над элементарными предикатами первого и второго порядков без кванторов.

Совместное рассмотрение контекстов дает возможность решать, например, задачу семантического анализа запросов. Показано, каким образом происходит формирование обобщенного контекста запроса, который дает возможность осуществлять проверку корректности запроса до стадии его исполнения.

Литература

1. Brézillon P. Context in Artificial Intelligence: II. Key Elements of Contexts // Computer & Artificial Intelligence. 1999. Vol. 18. № 5. P. 425–446.
2. Фридман А. Я. Ситуационный подход к моделированию промышленно-природных комплексов и управлению их структурой. Труды IV международной конференции «Идентификация систем и задачи управления». М., 25–28 января 2005. Институт проблем управления им. В. А. Трапезникова. М.: Институт проблем управления им. В. А. Трапезникова, 2005. С. 1075–1108.
3. Smirnov A., Pashkin M., Shilov N., Levashova T. Constraint-driven methodology for context-based decision support // Journal of Decision Systems. 2005. № 14 (3). P. 279–301.
4. Зуенко А. А. Формализация метаданных для хранилищ информации, поддерживающих открытую концептуальную модель // Информационные технологии в региональном развитии. Апатиты: Изд-во КолНЦ РАН, 2007. Вып. VII. С. 82–87.

5. Зуенко А. А., Фридман А. Я. Развитие алгебры кортежей для логического анализа баз данных с использованием двухместных предикатов // Известия РАН. Теория и системы управления. 2009. № 2. (В печати).
6. Зуенко А. А., Фридман А. Я. Метод семантического анализа нерегламентированных запросов в реляционной базе данных с иерархической структурой (настоящий сборник).
7. Страуструп Б. Язык программирования С++. 3-е изд. СПб.: Невский Диалект — БИНОМ, 1999.
8. Виноградов А. Н., Жиликова Л. Ю., Осипов Г. С. Динамические интеллектуальные системы. 1. Представление знаний и основные алгоритмы // Известия РАН. Теория и системы управления. № 6. М.: Наука, 2002. С. 119–127.
9. Фридман А. Я., Фридман О. В. Контроль корректности вычислений и управление выводом в системах продукций // Имитационное моделирование в исследованиях проблем регионального развития. Апатиты: КНЦ РАН, 1999. С. 93–100.

Метод семантического анализа нерегламентированных запросов в реляционной базе данных с иерархической структурой

А. А. Зуенко, А. Я. Фридман

ИИММ ТП КолНЦ РАН, Апатиты

Статья посвящена разработке метода семантического анализа и оптимизации незапланированных (нерегламентированных) адресных (путевых) запросов к базам данных с фиксированной иерархической структурой. Метод основан на учете семантических ограничений в виде специализированной онтологии и последующем использовании этой информации при анализе запросов с целью упрощения их структуры. Применимость метода ограничена классом онтологий, поддерживающих только иерархические связи «часть-целое» и ассоциативные (предметные) связи. Описана реализация метода для базы данных системы моделирования на основе открытой иерархической концептуальной модели предметной области.

1. Введение

В целях обеспечения полноценной интеграции с объектно-ориентированными приложениями разработчики современных СУБД все более стремятся перенести парадигмы объектно-ориентированного программирования на технологию БД. Современные СУБД используют концепцию объекта (экземпляр сложного типа, определяемого пользователем) для представления знаний о предметной области, снижения сложности запросов к данным, хранения объектов программ и так далее [1, 2].

В СУБД объектный подход широко используется при обработке данных, имеющих иерархическую структуру, в частности, при работе с форматом XML, который фактически представляет собой объектно-ориенти-