

# Контекстный подход в системах сопровождения открытых концептуальных моделей предметной области

**Аннотация.** В статье рассматривается метод управления контекстом, ориентированный на задачу организации интеллектуального интерфейса БД в системах моделирования на основе открытой концептуальной модели предметной области. Основное внимание уделяется различным стадиям семантического анализа нерегламентированных запросов к БД и роли, которую в этом играют контексты отдельных объектов, участвующих в обмене данными моделирования.

**Ключевые слова:** структурирование знаний, модель онтологии, нерегламентированные запросы, интеллектуальный интерфейс БД.

## 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 средств управления контекстом для обработки данных в системах моделирования рассмотрено

на примере системы ситуационного концептуального моделирования (ССКМ) [2].

## 1. Контекстный подход к обработке информации в системах концептуального моделирования

Рассмотрим возможности модификации методологии управления контекстом [3] при организации интеллектуального 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, организация имитационного режима и так далее [2]. Открытость модели предметной области предполагает возможность подключения но-

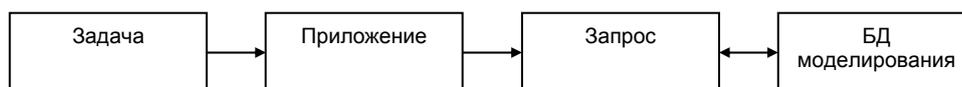


Рис.1. Объекты ОДМ

вых 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 существенно упростить их анализ, причем основная часть процедур анализа осуществляется еще на этапе оптимизации запроса [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, подразделяется на абстрактный и прикладной контексты. Абстрактный контекст объекта соответствует интенционалу этого объекта. Прикладной контекст объекта позволяет описать некоторое множество его экземпляров.

Контексты при их совместном рассмотрении накладывают дополнительные ограничения как на запрос, адресованный к БДМ, так и на структуру самой БДМ. Это дает возможность, например, производить различные процедуры проверки корректности запроса до стадии его исполнения.

*Контекст базы данных 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 не-

которого экстенционала онтологии, то есть для отдельных концептуальных моделей.

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

*Контекст запроса.* В [6] показано, что любой адресный запрос может быть разложен в набор конъюнктивных запросов. Абстрактный контекст конъюнктивного адресного запроса можно представить в виде неориентированного графа  $G$ , где ребра соответствуют связям, не описанным в онтологии, а каждая вершина является древовидным ориентированным графом  $W$ , где между различными вершинами присутствуют только регламентированные онтологией связи. Прикладной контекст адресного запроса получается путем назначения вершинам графа  $W$  условий фильтрации, представимых в виде логических формул над одноместными предикатами. Другими словами, прикладной контекст адресного запроса получается из абстрактного после назначения атрибутов 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 дублирующиеся запросы и, соответственно, многократная проверка их корректности.

Далее рассмотрим процесс формирования абстрактного и прикладного контекстов объектов ОДМ. Для формализации этих контекстов используется созданная авторами модель онтологии структурных ограничений.

## 2. Модель онтологии структурных ограничений

Потребность в создании новой модели онтологии для систематизации знаний об объектах

ОДМ обусловлена следующими причинами. Известные стандартизированные 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. 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 нельзя обойтись средствами, предоставляемыми известными и широко распространенными моделями онтологий. В XML-документах, которые предназначены для работы со слабоструктурированными данными, нет средств, позволяющих 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, формулы над одно- и двуместными предикатами, 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:

$$S_0 ::= \langle CL, A, AT, CLA, AAT, AGR, REF, CARD, COND \rangle, \quad (1)$$

где:  $CL ::= \{c_i\}$  – множество классов, классы одновременно являются агрегатами атрибутов и семейства возможных экземпляров, то есть

можно сказать, что  $c_i = \{ex_m^i\}$ , где  $ex_m^i$  – некоторый экземпляр класса  $c_i$ ;

$A ::= \{a_j\}$  – 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» (здесь и далее  $B(*)$  – булеан множества «\*»);

$AAT \subseteq A \times AT$  – 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:

$cla: CL \rightarrow B(A)$ ,  $\{a_j\} = cla(c_i)$ , где  $c_i$  – 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.

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 допус-

тимых значений атрибута, поскольку для задач данной работы не существенны ни лексическое представление типа, ни множество операций, применимых к некоторому типу.

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

Согласно терминологии работы [6], многосортная алгебра  $A = \langle (s^A)_{s \in S}, (f^A)_{f \in F} \rangle$  состоит из семейства множеств-носителей  $s^A$  (сортов) и семейства частичных функций  $f^A$ :

$$f^A : s_1^A \times s_2^A \times \dots \times s_{n-1}^A \rightarrow s_n^A, n \geq 0.$$

В рассматриваемой задаче в качестве сортов многосортной алгебры примем классы онтологии. Семейство частичных функций формализует связи, заданные в онтологии. Так как связи онтологии определяются, в основном, бинарными отношениями схемы концептуальной модели и ее фрагментов, то эти функции имеют схему  $f: s_1 \rightarrow s_2$ .

С каждым сортом  $s_i \in S$  свяжем счетное множество  $X_s$  переменных и определим множество термов TR и функцию тип:  $TR \rightarrow S$  следующим образом:

- всякая переменная  $x \in X_s$  есть терм, причем  $\text{тип}(x) = s$ ;
- всякий нуль-арный функциональный символ (константа)  $f \in F$  со схемой отображения  $f: \rightarrow s$  есть терм, причем  $\text{тип}(f) = s$ ;
- если  $f \in F$  имеет схему отображения  $f: s_1 \rightarrow s_2$  и  $t_1, t_2$  – термы, у которых  $\text{тип}(t_1) = s_1$ ,  $\text{тип}(t_2) = s_2$ , то  $f(t_1, t_2)$  – терм,  $\text{тип}(f(t_1, t_2)) = s_2$ .

**Определение 1.** Назовем *фактом* упорядоченный список вида  $(f, t_1, t_2)$ , где  $f \in F$ ,  $f: s_1 \rightarrow s_2$ ,  $t_i \in TR$  ( $TR$  – множество допустимых термов),  $\text{тип}(t_i) = s_i$ ,  $i = 1, 2$  или список вида  $(\text{тип}, t, s)$ , где  $t \in TR$ ,  $s \in S$ ,  $\text{тип}(t) = s$ . В этих обозначениях *ситуацией* назовем конечную конъюнкцию фактов. В дальнейшем через D будем обозначать множество возможных ситуаций.

**Определение 2.** Ограничением отношения онтологии назовем пару  $\langle \text{decl}, \text{expr} \rangle$ , в которой decl – ситуация, называемая декларацией переменных (переменные соответствуют классам онтологии), expr – логическая формула над элементарными одно- и двуместными предикатами, где переменные – атрибуты классов онтологии, присутствующих в декларации.

**Определение 3.** Системой ограничений назовем конечное множество кортежей  $\text{COND} = \{ \langle \text{decl}, \text{expr} \rangle \}$ .

Каждая онтология, созданная в соответствии с описанной моделью, является неоднородной семантической сетью в терминологии работы [8] (фактически интенционал сети). Неоднородной экстенциональной семантической сетью будет сеть следующего вида:

$H_0 ::= \langle E, A, AT, CLA, AAT, AGR, REF, CARD, COND \rangle$ , которая получается из схемы  $S_0$  (1) путем следующих преобразований:

1. каждому классу  $cl_i \in CL$  поставим в соответствие некоторое подмножество одного из декартовых произведений  $e \subseteq at_{k1} \times at_{k2} \times \dots \times at_{km}$ , где  $at_i \in \text{aat}(\text{cla}(cl_i))$ ;  $e$  будем называть экстенционалом (или множеством экземпляров, или множеством примеров) класса  $cl_i$ . Множество экстенционалов всех классов из CL обозначим через E;
2. в схеме (1) заменим CL на E;
3. определим на  $E^2$  отношения из  $S_0$ .

Отношение «is-a» в спецификацию модели онтологии не включено, оно реализуется при переходе от интенционала к экстенционалу семантической сети (в процессе автоматического согласования структуры БДМ с открытой концептуальной моделью). Экстенционалы онтологии хранятся в базе данных моделирования и содержат информацию, относящуюся к некоторому множеству концептуальных моделей.

Далее на примере БДМ ССКМ иллюстрируются понятия абстрактного и прикладного контекстов базы данных моделирования как объекта ОДМ.

### 3. 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 схемы КМ [4, 6]. Концептуальная модель (класс M) состо-

ит 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).

### 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.

### 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;  $O^M.basetype$  – встроенный (базовый) тип 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;  $P^M.basetype$  – встроенный (базовый) тип 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;  $R^M.basetype$  – встроенный (базовый) тип 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.

### Отношения и ограничения на связи между классами

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 одно- и двуместными предикатами.

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 (базовый тип “buf”) должны быть одинаковыми.

2. Имена соответствующих однотипных ресурсов на входах и выходах процессов-разветвителей и преобразователей (базовый тип “bra” и “tra”) должны быть различными.

3. На входные ресурсы процессов с базовым типом “pro” ограничений не накладывается.

Ниже приведем формальное описание данного ограничения  $cond = \langle decl, expr \rangle$ :

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

$expr = (P.basetype = 'buf') \wedge (R^{in}.usertype = R^{out}.usertype) \wedge (R^{in}.rname = R^{out}.rname) \vee ((P.basetype = 'bra') \vee (P.basetype = 'tra')) \wedge (R^{in}.usertype = R^{out}.usertype) \wedge (R^{in}.rname \neq R^{out}.rname) \vee (P.basetype = 'pro')$ .

## 4. 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 поясним, как происходит генерация прикладного контекста БДМ.

В отличие от ограничений интенционала, ограничения, специфичные для конкретных КМ, формируются преимущественно автоматически, например, на основе информации о пользовательских типах элементов [9] или любой другой информации, которую разработчик системы моделирования посчитает нужным учитывать. В частности, они могут генерироваться из ограничений абстрактного контекста путем замены базовых типов конкретными именами и пользовательскими типами или посредством специализированных алгоритмов, выявляющих интересующие ситуации в КМ. Далее опишем пользовательские типы элементов ССКМ и поясним, как на их основе получают ограничения прикладного контекста.

Ресурсы в ССКМ могут иметь только целочисленные, вещественные или строковые значения. Пользовательские типы ресурсов позволяют на основании этих базовых типов создавать новые типы данных, которые получают 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.

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 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. Информация о типах представляется в виде системы ограничений, где каждое ограничение включает логическую формулу над элементарными одно- и двуместными предикатами. Например, списки типов входных и выходных ресурсов формируют дополнительные ограничения на отношения онтологии «объект (процесс) – входные ресурсы», «объект (процесс) – выходные ресурсы». Пусть имеется промышленный объект с пользовательским типом «ТЭС» (теплоэлектростанция), который может потреблять либо уголь, либо мазут (пользовательские типы ресурсов «Уголь» и «Мазут»), тогда на основе описания типа объекта будет автоматически сгенерировано ограничение  $cond = \langle decl, expr \rangle$ , где:

$decl = \text{inresobj}(R^{\text{in}}, O) \wedge \text{тип}(O, \text{объект}) \wedge \text{тип}(R^{\text{in}}, \text{ресурс})$ ;

$expr = (O.\text{usertype} = \langle \text{ТЭС} \rangle) \wedge ((R^{\text{in}}.\text{usertype} = \langle \text{Уголь} \rangle) \vee (R^{\text{in}}.\text{usertype} = \langle \text{Мазут} \rangle))$ .

Подобные ограничения определяются составом конкретной модели предметной области, а не являются следствием схемы концептуальной модели. Объединяясь с системой ограничений абстрактного контекста, такие ограничения образуют контекст БДМ.

## 5. Абстрактный и прикладной контексты запроса

В силу того, что классы элементов онтологии упорядочены в иерархию, то и структура запроса должна отражать иерархичность, чтобы можно было строить запросы в терминах онтологии. Поэтому для обращения к БДМ предла-

гается использовать язык, опирающийся на понятие пути, как способ адресовать данные в рамках иерархии.

Для доступа к данным БДМ используются путевые (адресные) запросы следующего вида:

$$\langle \text{Имя\_объектной\_структуры} \rangle [ \langle \text{условия\_фильтрации} \rangle ] \{ \langle \text{Имя\_объектной\_структуры} \rangle [ \langle \text{условия\_фильтрации} \rangle ] \}^{\geq 0} [ \text{Имя\_атрибута} ] \quad (2)$$

Данное адресуется как свойство некоторой объектной структуры (или как сама объектная структура), которая занимает определенное место в общей иерархии включения. Для формирования условий выборки нужных данных используются фильтры. Условия фильтрации можно разделить на следующие виды:

1. одноместные предикаты;
2. двуместные предикаты ( $<$ ,  $>$ ,  $=$ ,  $\neq$ ), где переменные – атрибуты уровня;
3. переход к вложенному запросу при помощи заранее оговоренных связей (ролевых связей онтологии);
4. переход к вложенному запросу по нерегламентированным онтологией связям.

**Определение 4.** Адресный запрос вида (2) называется конъюнктивным, если все условия фильтрации имеют вид  $F \& P$ , где  $F$  – конъюнкция элементарных одно- и двуместных предикатов (переменные – атрибуты уровня), а  $P$  – конъюнкция межуровневых переходов, как регламентированных, так и нерегламентированных.

В процессе выполнения программного приложения, реализующего некоторую задачу моделирования, запросы к БДМ, как правило, формируются динамически, то есть с использованием данных, которые становятся известными только при работе приложения (передаются

конечным пользователем или другими приложениями). Часть запроса, которая не зависит от подстановки конкретных значений, в рамках настоящей работы называется абстрактным контекстом запроса.

Рассмотрим пример адресного запроса “получить все процессы модели с именем «M1», объекты которых имеют категорию «COMP» (имеющие подчиненные объекты [2, 9]), и входные ресурсы которых являются входными ресурсами объектов с категорией «GISC» (состоящих их набора стандартных ГИС-элементов)”:

```
models[name='M1'].processes[(objowner= models
[name='M1'].objects[cat='COMP'].id)and(id=models
[name='M1'].resources[conso=models[name='M1'].
objects[cat='GISC'].id].consp)].
```

Граф этого запроса показан на Рис.2. Он не содержит нерегламентированных связей и соответствует, в общем случае, одной компоненте связности. На графе для наглядности не нанесены атрибуты уровней запроса.

Абстрактный контекст этого запроса можно записать в виде следующего ограничения онтологии  $cond_1 = \langle decl_1, expr_1 \rangle$ :

```
decl1 = point(Process1, Model1) ∧ point(Object1,1,
Model1,1) ∧ point(Resource1,2, Model1,2) ∧
point(Object1,2,1, Model1,2,1) ∧ objproc(Process1,
Object1,1) ∧ procinres(Resource1,2, Process1) ∧
objinres(Resource1,2, Object1,2,1) ∧ тип(Model1, модель) ∧
тип(Model1,1, модель) ∧ тип(Model1,2, модель) ∧
тип(Model1,2,1, модель) ∧ тип(Process1, процесс) ∧
тип(Resource1,2, ресурс) ∧ тип(Object1,1, объект) ∧
тип(Object1,2,1, объект) ∧ attr(name, Model1) ∧
attr(name, Model1,1) ∧ attr(name, Model1,2) ∧
attr(name, Model1,2,1) ∧ attr(cat, Object1,1) ∧ attr(cat,
Object1,2,1);
```

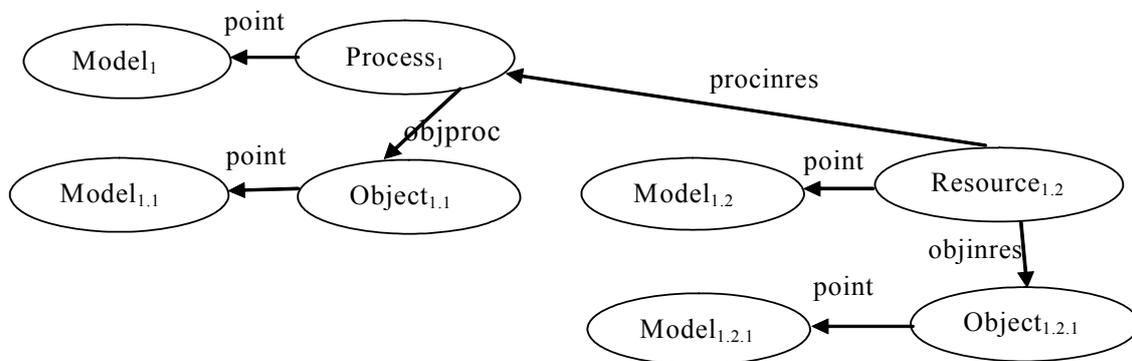


Рис.2. Пример графа адресных запросов

$\text{expr}_1 = \text{true}$  – означает, что на атрибуты из ситуации  $\text{expr}_1$  не налагается ограничений.

Прикладной контекст рассмотренного запроса получается из абстрактного путем присоединения к ситуации  $\text{expr}_1$  ситуации  $\text{expr}_1^C = (\text{Model}_{1.1}.\text{name} = 'M1') \wedge (\text{Model}_{1.1.1}.\text{name} = 'M1') \wedge (\text{Model}_{1.2}.\text{name} = 'M1') \wedge (\text{Model}_{1.2.1}.\text{name} = 'M1') \wedge (\text{Object}_{1.1}.\text{cat} = 'COMP') \wedge (\text{Object}_{1.2.1}.\text{cat} = 'GISC')$ .

Контекст запроса более общего вида, а именно конъюнктивного запроса, содержащего нерегламентированные онтологией связи, можно также записать в виде ограничения онтологии, то есть как  $\text{cond}^P = \langle \text{decl}^P, \text{expr}^P \rangle$ . Допустим, что конъюнктивный запрос содержит  $n$  нерегламентированных переходов на другие уровни, тогда компонент связности в нем –  $n+1$ . Контекст каждой компоненты можно описать соответствующим ограничением  $\text{cond}_i^P = \langle \text{decl}_i, \text{expr}_i \rangle$ , где  $i = 1 \dots n+1$ . Тогда ситуация  $\text{decl}^P = \text{decl}_1 \wedge \dots \wedge \text{decl}_{n+1}$ , а  $\text{expr}^P = \text{expr}_1 \wedge \dots \wedge \text{expr}_{n+1} \wedge \text{expr}_1^P \wedge \dots \wedge \text{expr}_n^P$ , где  $\text{expr}_1^P \dots \text{expr}_n^P$  – предикаты, задающие нерегламентированные переходы. Выражение  $\text{cond}^P = \langle \text{decl}^P, \text{expr}^P \rangle$  будем называть конъюнкцией ограничений и обозначать для краткости  $\text{cond}^P = \text{cond}_1^P \wedge \dots \wedge \text{cond}_{n+1}^P$ . Произвольный запрос, допускающий связки не только по «и», но и по «или», представляет собой дизъюнкцию

$$\bigvee_P \text{cond}^P = \bigvee_P \bigwedge_i \text{cond}_i^P \quad (3)$$

ограничений из множества  $\text{COND}^R = \{\text{cond}^P\}$ , соответствующих контекстам конъюнктивных запросов, совокупность результатов исполнения которых идентична результату исходного запроса.

Таким образом, получаем, что и абстрактный, и прикладной контекст произвольного запроса можно описать как ограничение используемой модели онтологии.

Присутствие в запросе межуровневых связей, оговоренных в онтологии, дает возможность анализировать эти связи с точки зрения ограничений онтологии. Однако такой анализ целесообразно производить только после предварительного преобразования структуры запроса. Это обусловлено следующими причинами. Во-первых, различные уровни запроса могут адресовать одни и те же элементы данных, поэтому условия этих уровней целесообразно анализировать совместно, предварительно вы-

полняя их объединение. Кроме того, наличие в запросе некоторых регламентированных связей накладывает на запрос «скрытые» ограничения, явное указание которых позволяет производить более полный анализ с точки зрения различного рода ограничений. Таким образом, в процессе исследований выделено два класса предварительных преобразований запроса, направленных на оптимизацию его структуры:

1. Правила «склейки» вершин – выявление и отождествление узлов запроса, описывающих одно и то же множество элементов КМ.
2. Правила доопределения запроса – выявление «скрытых» ограничений запроса.

Часть таких преобразований хорошо поддается алгоритмизации, поскольку является следствием только иерархичности онтологии. Эти преобразования выполняются над парами вершин, не соединенных ассоциативными дугами. Другая часть преобразований специфична для каждой конкретной онтологии и не может быть задана раз и навсегда, поскольку касается тех пар вершин, между которыми существует путь, состоящий только из ассоциативных дуг.

В [6] предлагается описывать специфичные преобразования запроса декларативно, в виде систем продукций, там же сформулированы и доказаны условия корректности таких систем. В настоящей работе выявляется роль предварительных преобразований запроса при анализе его корректности в применении к абстрактному и прикладному контекстам в отдельности.

Для абстрактного контекста запроса предварительные преобразования целесообразно осуществлять, прежде всего, в рамках компонент связности. В процессе выполнения правил «склейки» вершин запроса для каждой вершины формируется две формулы, одна из которых соответствует конъюнкции одноместных предикатов, а другая представляет собой конъюнкцию двуместных предикатов над переменными уровня. Первая формула является «черным ящиком» в том смысле, что структура ее элементарных предикатов не определена до тех пор, пока не сформирован прикладной контекст запроса. Структура второй формулы, напротив, известна, поскольку она формируется из элементарных предикатов выражения  $\text{expr}$  абстрактного контекста запроса. Истинность второй формулы, как правило, на данном этапе установить невозможно, однако в процессе упроще-

ния формулы можно выявить ее некорректность [5].

Для дальнейшего изложения введем следующие определения.

**Определение 5.** *Типом запроса* называется его абстрактный контекст, где как дизъюнкты формулы (3), так и отдельные конъюнкты, образующие их, располагаются в алфавитном порядке.

В процессе анализа формул над двуместными предикатами может быть установлена некорректность того или иного типа запросов.

**Определение 6.** *Канонической формой* типа запроса называется тип, полученный после предварительных преобразований «склейки» и доопределения запроса.

Набор типов запросов (исходных, канонических форм, вспомогательных формул над элементарными одно- и двуместными предикатами), выполняемых приложением образует *контекст приложения*, который формируется большей частью автоматически. При исполнении приложением запроса, тип которого уже известен, нет необходимости выполнять заново большинство предварительных преобразований запроса, а нужно лишь вычислить сформированные заранее формулы. Контекст приложения формируется с использованием *каталога типов*. Данный каталог является специализированным справочником, позволяющим конструировать новые типы запросов с использованием уже известных «атомов» запросов (компонент связности конъюнктивных запросов), для которых вычислена каноническая форма. Анализируя запрос, тип которого в данном приложении еще не использовался, целесообразно изучить контексты приложений, реализующих ту же самую задачу, то есть *контекст задачи*. Таким образом, контексты приложения и задачи, использующие каталог типов, способствуют повышению эффективности процедур проверок корректности запросов, поскольку исключают многократное дублирование проверок при анализе.

После формирования прикладного контекста запроса может быть выполнено соединение соседних компонент связности графа запроса. Для полученной после соединения компоненты преобразования, предваряющие основной семантический анализ запроса, будут касаться прежде всего цепочек  $X_1 f_1 X_2 \dots X_{n-1} f_{n-1} X_n$ , где

$X_1$  и  $X_n$  – уровни запроса, связанные нерегламентированными дугами. Причина в том, что большинство предварительных преобразований выполняется на уровне абстрактного контекста.

Только после формирования прикладного контекста запроса становится возможным вычисление формулы над элементарными одноместными предикатами, которая конструируется еще на стадии построения канонической формы типа запроса.

Выполнение априорных преобразований прикладного контекста запроса предваряет этап семантического анализа его корректности с учетом конкретных значений.

## 5. 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.3) происходит следующим образом. 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 контекста), получаем набор конкретизированных ограничений контекста БДМ.

Накладывая эти ограничения на 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.

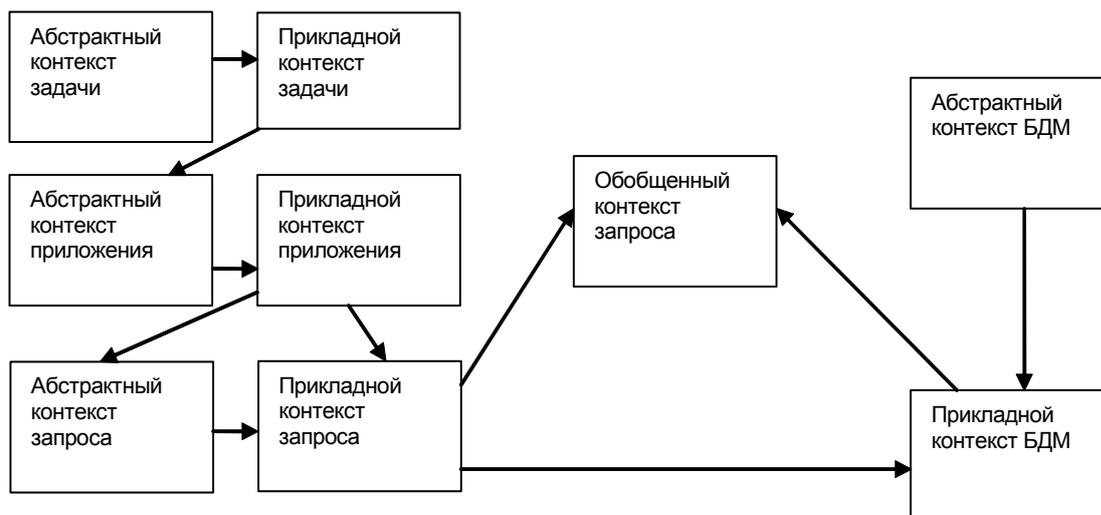


Рис.3. Схема формирования обобщенного контекста запроса

## Заключение

В статье описан контекстный подход к анализу семантических ограничений в системах моделирования на основе открытой концептуальной модели предметной области. Предложена формализация контекстов различных категорий объектов, участвующих в обработке данных моделирования, и метод совместного использования этих контекстов для семантического анализа запросов и контроля корректности базы данных моделирования. Для систематизации 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 одно- и двуместными предикатами. Представлены два вида продукционных правил оптимизации запроса, которая производится при его сопоставлении с контекстными ограничениями базы данных моделирования, и показано, каким образом происходит формирование обобщенного контекста запроса с помощью процедуры логического вывода, что дает возможность осуществлять проверку корректности запроса до стадии его исполнения.

Предложенный подход позволяет применить методы представления знаний, характерные для документальных поисковых систем, в системах моделирования, которые по требованиям к точности реализации запросов относятся к фактографическим информационным системам.

## Литература

1. Brézillon P. Context in Artificial Intelligence: II. Key Elements of Contexts // *Computer & Artificial Intelligence*, 1999. – Vol. 18. - № 5. – Pp. 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". Москва, 25-28 января 2005 г. Институт проблем управления им. В.А. Трапезникова, Москва: Институт проблем управления им. В.А. Трапезникова, 2005 г. - 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 двуместных предикатов // *Известия РАН. Теория и системы управления* (в печати).
6. Зуенко А.А., Фридман А.Я. Метод семантического анализа нерегламентированных запросов в реляционной базе данных с иерархической структурой // VII Всероссийская школа-семинар "Прикладные проблемы управления макросистемами" (Апатиты, 31 марта - 4 апреля 2008 года.): Сборник докладов. – М.: ИСА РАН (в печати).
7. Б. Страуструп Язык программирования C++, 3-е изд. – СПб.: "Невский Диалект" – "Издательство БИНОМ", 1999.
8. Виноградов А.Н., Жилиякова Л.Ю., Осипов Г.С. Динамические интеллектуальные системы. -Ч.1. Представление знаний и основные алгоритмы //

- Известия РАН. Теория и системы управления. -М.: Наука, 2002.
9. Фридман А.Я., Фридман О.В. Контроль корректности вычислений и управление выводом в системах продукции // Имитационное моделирование в исследованиях проблем регионального развития. - Апатиты: КНЦ РАН, 1999.- С.93-100.

**Зуенко Александр Анатольевич**, младший научный сотрудник Института Информатики и Математического Моделирования Технологических Процессов КолНЦ РАН. В 2005 году окончил Петрозаводский Государственный Университет. Имеет 17 печатных работ. Области научных интересов: системы моделирования на основе открытой концептуальной модели предметной области, представление знаний, семантический анализ запросов.

**Фридман Александр Яковлевич**, заведующий лабораторией систем и методов моделирования Института Информатики и Математического Моделирования Технологических Процессов КолНЦ РАН. В 1975 году окончил Ленинградский электротехнический институт. Доктор технических наук (2001). Профессор (2008). Имеет 145 печатных работ. Области научных интересов: моделирование комплексных технологий и их воздействия на окружающую среду, прикладные интеллектуализированные системы.