Министерство образования Республики Беларусь

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра интеллектуальных информационных технологий

В.В.Голенков, Н.А.Гулякина, О.Е.Елисеева

ГРАФОДИНАМИЧЕСКИЕ МОДЕЛИ И ЯЗЫКИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ ОБУЧАЮЩИХ СИСТЕМАХ

УЧЕБНОЕ ПОСОБИЕ

по курсу "Интеллектуальные тренажерные и обучающие системы" для студентов специальности “Искусственный интеллект”

Минск 2000

УДК 519.68:681.3
ББК 32.813
Г 60
Авторы:
В.В.Голенков, Н.А.Гулякина, О.Е.Елисеева

Рецензенты:
акад. РАЕН, д-р техн. наук, вед. науч. сотр. Института проблем передачи информации, зам. президента Российской ассоциации искусственного интеллекта В.Л.Стефанюк;
доц. кафедры педагогики БГУ, канд. пед. наук М.Ф.Поснова

Голенков В.В. и др. Графодинамическе модели и языки представления знаний в интеллектуальных обучающих системах: Учеб. пособие по курсу "Интеллектуальные тренажерные и обучающие системы" для студентов специальности "Искусственный интеллект". - Мн.: БГУИР, 2000. - 75 с.: ил. 42.
ISBN 985-444-109-1

Рассматриваются графодинамические модели представления знаний в интеллектуальных обучающих системах (ИОС) нового поколения. Указанный подход является одним из наиболее перспективных для создания интеллектуальных систем различного назначения, в том числе и для ИОС. Описывается множество графовых семантических языков, соответствующих графодинамическим моделям представления знаний различных подсистем ИОС.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Анализ требований, предъявляемых к интеллектуальным обучающим системам (ИОС) нового поколения, показывает, что комплексное выполнение всех этих требований предполагает поддержку а) БЗ весьма сложной структуры, б) самых различных моделей рассуждений и, как следствие, в) параллельной переработки информации [14]. Очевидно, что для удовлетворения всех этих условий необходимо наличие некоторого единого подхода к проектированию всех компонентов системы, на базе которого следует разработать соответствующий инструментарий [17].

В свою очередь, каждый из компонентов ИОС, помимо знаний специфических, описывающих только конкретную предметную область или конкретного пользователя, содержит также и некоторые метазнания, т.е. знания, общие для любой предметной области и для любого пользователя. Следовательно, эти знания могут быть внесены в систему заранее. Механизмы обработки тех или иных знаний также поддаются унификации и могут являться общими для различных подсистем ИОС. Таким образом, в состав инструментальных средств проектирования ИОС должны включаться средства описания знаний и наиболее общие механизмы их переработки. На базе этого инструментария технология проектирования конкретной ИОС значительно упрощается по сравнению с тем, как если бы создавалась некоторая отдельно взятая ИОС без использования базовых инструментальных средств.

Главной целью данного пособия является рассмотрение одного из перспективных подходов к проектированию интеллектуальных систем (ИС) различного назначения, основанного на графодинамической парадигме представления и переработки сложно структурированной информации [3], применительно к созданию ИОС нового поколения по различным предметным областям.

Согласно данному подходу каждая из подсистем в составе ИОС представляет собой конкретную ИС. Все знания для каждой ИС, являющейся компонентом ИОС, описываются на семантических графовых языках SC (Semantic Code) [3] и SCL (Semantic Code Logic) [9; 10; 19], а также их расширениях, которые более подробно будут рассмотрены ниже. Для реализации операций пе-

реработки знаний в ИОС (SCL-операций [8] и их аналогов) используется язык параллельного программирования SCP [6; 7].

1. ГРАФОДИНАМИЧЕСКАЯ ПАРАДИГМА ПЕРЕРАБОТКИ ИНФОРМАЦИИ КАК ОСНОВА ДЛЯ СОЗДАНИЯ ИОС НОВОГО ПОКОЛЕНИЯ

Исходя из вышесказанного ИОС нового поколения представляют собой более развитый и, как следствие, более сложный класс КСО, в котором решаются следующие проблемы:

1) обработка больших объемов сложно структурированной информации различного типа;

2) обеспечение гибкости и легкой модифицируемости системы с целью адаптации к пользователю;

3) работа в режиме реального времени;

4) интеграция различных программных систем в составе одной системы.

В связи с этим, инструментальные средства проектирования (ИСПр) ИОС нового поколения ориентированы на следующее:

1) поддержку сложно структурированных баз знаний, включающих описание информации различного типа;

2) обеспечение легкой интегрируемости баз знаний с внешними по отношению к ним прикладными программами;

3) обеспечение легкой интегрируемости различных видов знаний и различных механизмов их переработки, т.е. интегрируемости баз знаний различных прикладных ИС в составе одной системы;

4) сочетание синтаксической простоты и семантической мощности языков представления знаний;

5) открытость (легкую расширяемость) языков представления и переработки знаний;

6) поддержку параллельной переработки знаний;

7) наличие технологии применения ИСПр для разработки конкретных

ИОС.

В настоящее время существует несколько подходов к проектированию прикладных ИС, разработаны также различные типы языков представления и переработки знаний, таких как, например, фреймовые языки, системы продукций, семантические сети и др. Наиболее перспективным многими разработчиками ИС считается подход к представлению знаний в виде однородных семантических сетей, благодаря, во-первых, наглядности и внешней простоте описания информации, а во-вторых, предоставлению возможности описывать и обрабатывать знания сложной структуры. До недавнего времени существовала проблема программной реализации средств интерпретации языков данного класса. Однако сейчас эта проблема снимается в связи с развитием графодинамической парадигмы параллельной асинхронной переработки знаний, представленных однородными семантическими сетями [4].

Использование в качестве основы для реализации ИОС нового поколения графодинамической парадигмы параллельной асинхронной переработки знаний является одним из возможных перспективных вариантов решения данной задачи, так как суть указанной парадигмы заключается в следующем:

1) ориентация на семантическое представление информации, а именно на графовый семантический язык представления знаний;

2) представление информации с использованием языков на теоретико-множественной основе;

3) обеспечение возможности глубокого распараллеливания процессов обработки информации;

4) приспособленность к будущей аппаратной поддержке в рамках специальных параллельных компьютеров.

Таким образом, указанный подход к проектированию ИС является одним из наиболее перспективных для ИОС нового поколения, так как он специально ориентирован на поддержку сложноструктурированных знаний и смешанных гибких стратегий решения неформализованных задач. Основой для реализации указанных инструментальных средств является графодинамический параллельный ассоциативный компьютер [4; 12], архитектура которого значительно облегчает процесс разработки на его основе инструментальных средств, ориентированных на создание прикладных ИС.

Знания в графодинамическом параллельном ассоциативном компьютере описываются в виде графодинамических моделей, представляющих собой специальным образом организованные семантические сети [5]. Базовым языком описания графодинамических моделей является графовый язык SC (Semantic Code) [3; 11]. Конструкции языка SC (SC-конструкций) состоят из SC-элементов: SC-узлов и SC-дуг. Каждый SC-узел является знаком некоторого предмета или отношения описываемой предметной области (и называется ключевым узлом формальной теории, описывающей предметную область), либо знаком некоторого множества, состоящего из SC-узлов и/или SC-дуг. Каждая SC-дуга семантически трактуется как высказывание о принадлежности некоторого элемента (того элемента, в который дуга входит) некоторому множеству, которое обозначается другим или тем же элементом SC-конструкции (тем элементом, из которого рассматриваемая дуга выходит). В соответствии с типом дуги указанное высказывание может быть либо позитивным, либо негативным, либо нечетким, а также может быть как константным (конкретным) высказыванием, так и переменным высказыванием, которое представляет собой произвольное высказывание из некоторого множества высказываний о принадлежности. SC-элемент может быть константным (иметь тип const) или переменным (иметь тип var). SC-узел может иметь содержимое. SC-конструкции могут быть представлены как в графическом, так и в линейном (текстовом) виде.

Рассмотрим несколько простейших примеров представления информации в виде SC-конструкций.

ПРИМЕР 1

Введем некоторое множество, описывающее всех учащихся. Назовем его **student**. Каждый элемент указанного множества будет являться знаком конкретного обучаемого, т.е. высказывание **Иванов** Î **student** означает, что некто Иванов является учащимся. На языке SC данное высказывание имеет следующий вид.

В линейной (текстовой) форме:

student -> **Иванов**;

В графической форме данное высказывание представлено на рис. 1.1.

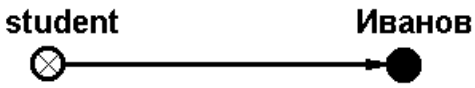

Рис. 1.1. Представление на языке SC высказывания:
"Иванов является учащимся"

Согласно денотационной семантике языка SC, здесь SC-узел **student** является знаком унарного отношения (имеет тип rel), а SC-узел **Иванов** является объектным (имеет тип obj) и обозначает некоторого конкретного учащегося, которого в данном описании кратко назвали **Иванов**. Оба введенных SC-узла являются константными. Типология указанных узлов задается путем явного указания их принадлежности ко множествам rel и obj соответственно:

rel -> **student**;
obj -> **Иванов**;

При описании знаний в графической форме типология SC-узлов задается конкретным внешним видом SC-узла [3].

ПРИМЕР 2

Введем теперь новый SC-узел **Петров**, обозначающий некоторого индивида, не являющегося учащимся. Если такую информацию требуется в явном виде хранить в базе знаний, то формируется SC-конструкция с использованием негативной SC-дуги (см. рис. 1.2), соответствующая высказыванию **Петров** Ï **student** .

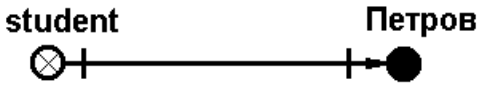

Рис. 1.2. Представление негативных высказываний на языке SC:
"Петров не является учащимся"

ПРИМЕР 3

Пусть теперь требуется описать начальную информацию об обучаемом путем задания его полного имени, включающего фамилию, имя и отчество. Для этой цели может быть использовано более сложное четырехарное асимметричное отношение. Обозначим его SC-узлом **fullName**. Это отношение включает четыре компонента, три из которых помечены атрибутами **name_**, **patronomic_**, **family_name_**, уточняющими семантику вхождения элементов в рассматриваемое множество (рис.1.3). Так как в базе знаний может присутствовать множество подобных высказываний, то обязательным является введение отдельного SC-узла, являющегося знаком кортежа каждого отдельно взятого отношения. В данном примере знаком кортежа рассматриваемого отношения является SC-узел name_Иванов.

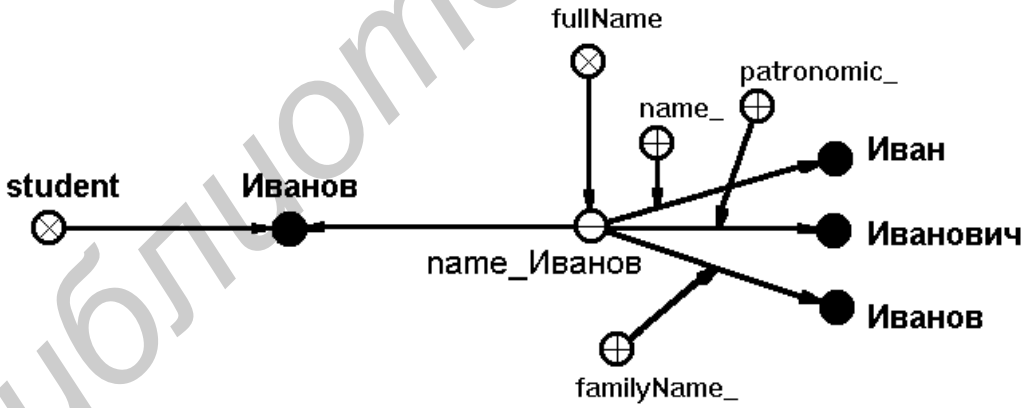

Рис. 1.3. Описание на языке SC информации о полном имени учащегося с использованием асимметричного отношения

Язык SC хорошо приспособлен к описанию сложно структурированных предметных областей, а также к описанию синтаксиса и семантики собственных языков. Абстрактные машины переработки информации, использующие в

качестве внутреннего языка различные SC-подъязыки, называются абстрактными SC-машинами.

Графовый язык SC имеет простой синтаксис, простую базовую денотационную семантику, открытый характер денотационной семантики, обусловленный открытым набором ключевых SC-узлов, и открытую операционную семантику, определяемую открытым набором операций абстрактных SC-машин.

Для представления знаний о предметной области (ПОб), т.е. логических соотношений между понятиями ПОб, в качестве основы используется SC-подъязык, называемый языком SCL (Semantic Code Logic) [9], на базе которого строится абстрактная SC-машина, определяющая операции переработки знаний, представленных на языке SCL - SCL-машина. В основе языка SCL лежит логический язык графового типа, т.е. графовый способ представления логических высказываний. Кроме средств представления логических высказываний в состав языка SCL входят также средства описания всевозможных целей и средства описания текущего состояния реализуемых семиотических моделей.

Язык SCL является языком открытого типа, позволяющим легко расширять и модифицировать средства представления различных знаний, что позволяет рассматривать язык SCL как ядро целого семейства языков представления знаний.

Таким образом, благодаря открытости языков SC и SCL, на их основе создаются дополнительные расширенные средства для описания информации и знаний в ИОС. Расширение указанных языков производится путем добавления к существующим наборам ключевых SC- и SCL-узлов новых узлов, которые позволят использовать расширенные таким образом возможности указанных языков в составе инструментальных средств проектирования ИОС. Кроме того, аналогичным образом расширяется операционная семантика соответствующих абстрактных машин, а именно, путем добавления к существующему набору операций переработки знаний добавляются операции, позволяющие интерпретировать знания, представленные с использованием расширенного набора ключевых узлов.

В качестве средства интерпретации абстрактных SC- и SCL-машин, т.е. реализации операций переработки знаний, представленных на соответствующих языках, используется язык программирования SCP (Semantic Code

Programming) [7], который также является SC-подъязыком, т.е. относится к разряду графовых языков, и специально ориентирован на переработку SC-конструкций. Кроме того, это язык параллельного программирования, благодаря чему в результате реализации операций абстрактных машин на языке SCP обеспечивается параллельная переработка знаний, что способствует повышению быстродействия готовой системы. В языке SCP имеется возможность через содержимое SC-узла интегрировать процедуры, обрабатывающие SC-конструкции, с процедурами, написанными на традиционных языках. Благодаря этому свойству появляется возможность интегрировать различные программные продукты, такие как, например, средства мультимедиа, готовые тестирующие программы и т.д.

В памяти графодинамического параллельного ассоциативного компьютера (SC-памяти) тело SCP-программы представляется в виде SC-конструкций. Это дает возможность SCP-программам преобразовывать самих себя. Благодаря этой особенности представления SCP-программ имеется возможность реализовывать механизмы самообучения ИС.

Заметим также, что в основу реализации графодинамического компьютера положены принципы не только параллельной, но и распределенной переработки информации. Это позволит в перспективе обобщить рассматриваемые в данной работе инструментальные средства на случай сетей. Таким образом, приложив минимальные усилия, можно будет получить инструментальные средства для проектирования интеллектуальных систем дистанционного обучения (ИСДО). Способ представления информации в виде семантических сетей является наглядной и следовательно удобной основой для описания принципов кооперативного взаимодействия пользователей ИСДО.

2. СЕМАНТИЧЕСКИЙ ГРАФОВЫЙ ЯЗЫК ПРЕДСТАВЛЕНИЯ УЧЕБНОГО МАТЕРИАЛА С СЕМАНТИЧЕСКОЙ ИНТЕГРАЦИЕЙ СО СРЕДСТВАМИ МУЛЬТИМЕДИА

База знаний (БЗ) предметной области (ПОб) описывается и перерабатывается средствами языка SCL. Подробно способы представления и переработки знаний на языке SCL рассмотрены в [4; 9; 10], и здесь их подробное рассмотрение опускается. Указанные средства включаются в состав инструментальных средств проектирования ИОС. Примеры решения задач из области геометрии подробно рассмотрены в [13].

Ниже дополнительно рассматриваются особенности представления информации в БЗ предметной области с целью генерации понятных пользователю объяснений, а также способы активизации механизмов решения задач и объяснения в результате диалога с пользователем.

Для иллюстрации выразительных средств языка SCL рассмотрим рис.2.1, позаимствованный из [4], на котором представлено SCL-высказывание определения отрезка. Здесь узлы с идентификаторами ***theory***, ***const_***, ***eqExpr_***, ***atomExpr_***, ***conj_*** являются ключевыми узлами языка SCL, с помощью которых описывается логический уровень высказывания; **TheoryGeom** - введенный инженером знаний узел для описания геометрической предметной области, являющийся знаком соответствующей SCL-теории (на это указывает дуга, проведенная в данный узел из узла ***theory***). Константы SCL-теории **point**, **segment**, **between**, **between_** являются знаками отношений, обозначающими множество геометрических точек, отрезков и отношения “Лежать между” соответственно. Далее на этом же примере будут рассмотрены способы дополнения БЗ предметной области информацией из учебной базы знаний (УБЗ).

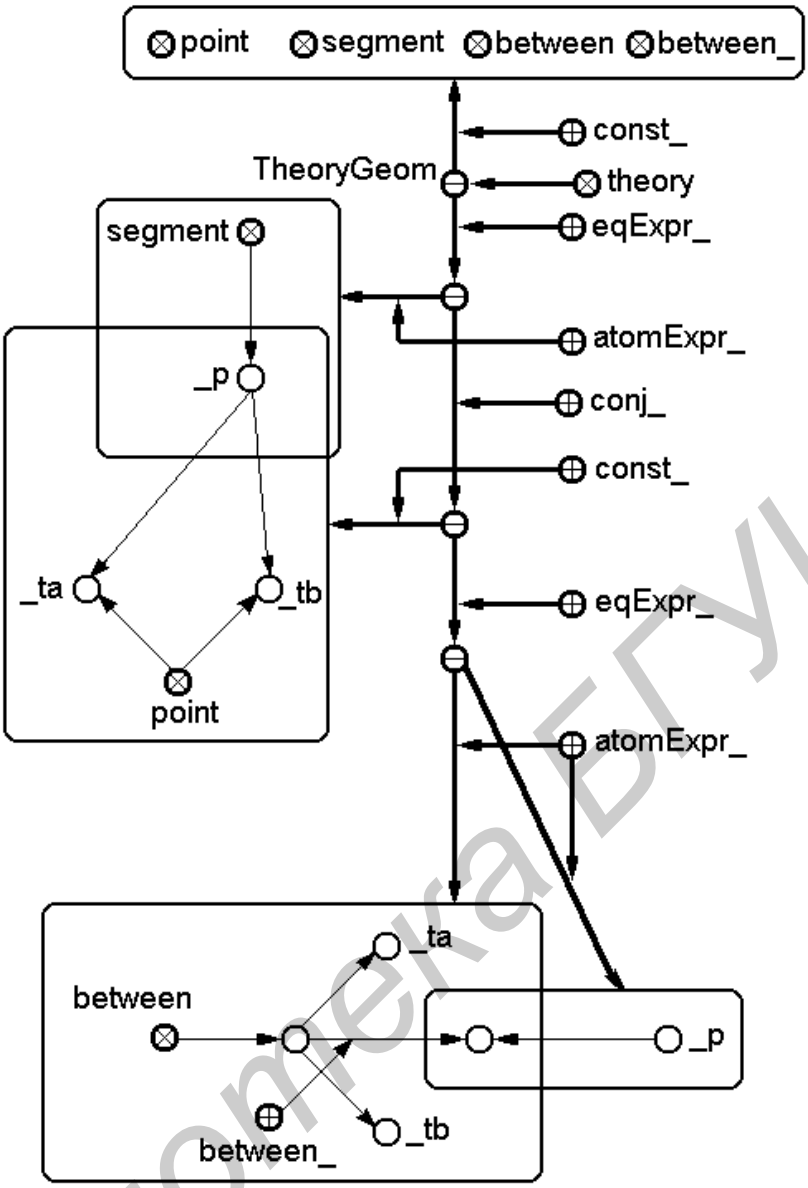

Рис.2.1. Представление на языке SCL высказывания об определении отрезка

2.1. Описание семантической роли элементов базы знаний

Согласно технологии формирования БЗ для подсистемы решения задач и ознакомления с предметной областью [14] после формализации предметной области следует этап дополнения БЗ учебным материалом, элементарными единицами которого являются тексты определений выделенных основных понятий ПОб, комментарии к ним, а также выделение семантически связанных понятий. Для отражения этой информации в УБЗ используются базовые средства языка SC, а именно набор соответсвующих знаков отношений: ***wordDef*** -

для ввода текста определения понятия, ***comm*** - для ввода комментариев к понятиям, ***nearSem*** - для выделения семантически связанных понятий.

Рассмотрим использование перечисленных ключевых узлов языка SC на примере. Пусть для описания предметной области выделены основные понятия “точка” и “отрезок” и введены соответствующие этим понятиям ключевые узлы SCL-теории **point** и **segment**. Для этих понятий в УБЗ необходимо ввести соответствующие определения. На языке SC это будут следующие высказывания:

wordDef -> (**point, *text_: /”Точка - это основная геометрическая фигура.”/);**

wordDef -> (**segment, *text_: /”Отрезок - это множество точек на прямой, лежащих между двумя заданными точками, которые называют концами отрезка.”/);**

Соответствующие им SC-конструкции изображены на рис.2.2.

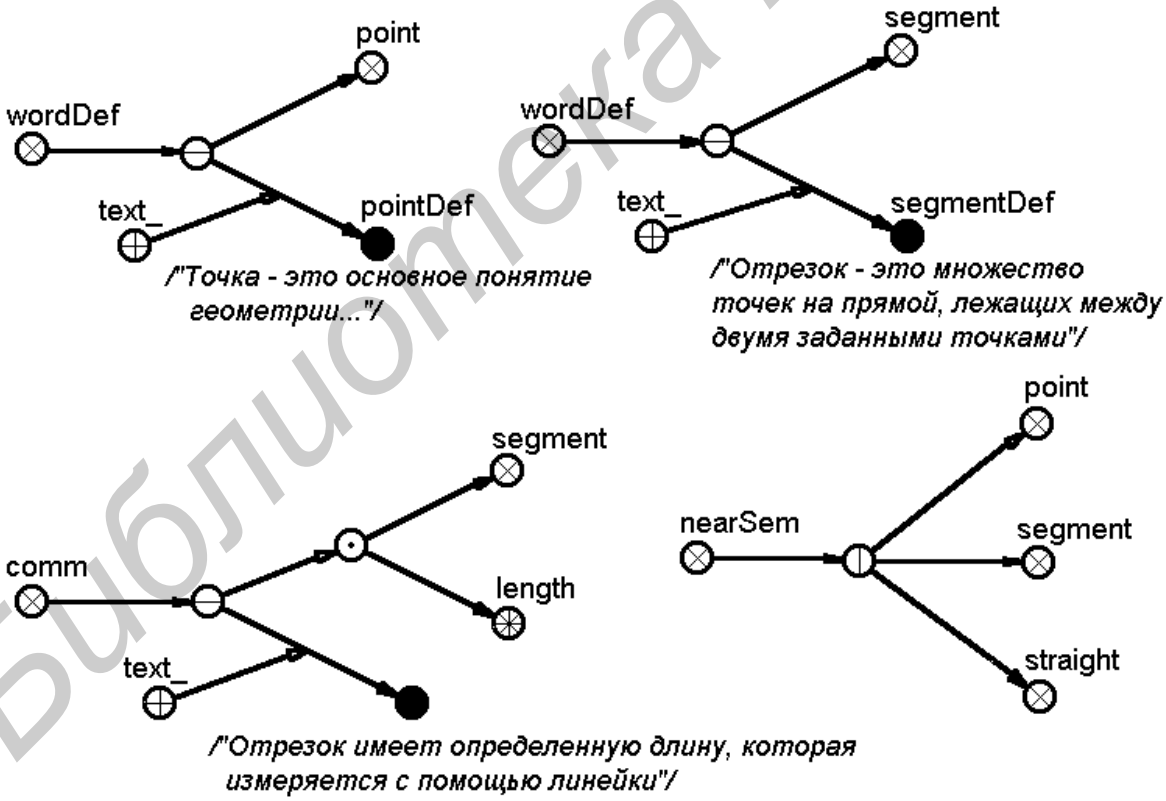

Рис.2.2. SC-конструкции для ***wordDef***, ***comm***, ***nearSem*** в графовом виде

Понятия “точка” и “отрезок” являются семантически связанными, что отражается в виде следующей SC-конструкции (см. также рис.2.2.):

nearSem -> (*point, segment, straight*); ,

где **straight** - знак множества геометрических прямых.

Для указания некоторых свойств описываемых понятий вводятся соответствующие комментарии, например (в графовом виде эта SC-конструкция представлена на рис.2.2):

comm -> ((*segment, length*), text_: /"Отрезок имеет определенную длину, которая измеряется с помощью линейки."/);

Загрузка представленных на рис.2.1 и 2.2 SC-конструкций в SC-память приведет к существованию в ней фрагмента БЗ по геометрии, описывающего в текстовом и графовом виде информацию об основных понятиях геометрии - точке и отрезке.

Для реализации взаимодействия с пользователем необходимы также операции обработки SC-конструкций рассмотренных выше типов. Подробнее эти оерации рассмотрены ниже.

2.2. Описание структуры и содержания учебного материала

Для представления структуры и содержания учебного материала имеются соответствующие расширенные средства языка SC, которые подробно рассматриваются ниже.

contents - знак отношения, связывающего элементы структуры учебного курса, последовательность которых указывается с помощью атрибутов ***1_***, ***2_***, ... ***n_*** (рис.2.3). Знаком кортежа этого отношения является SC-узел, обозначающий некоторый раздел, а элементами этого кортежа являются знаки соответствующих указанному разделу подразделов. В качестве идентификаторов SC-узлов, обозначающих разделы и подразделы учебного материала рекомендуется указывать их полные названия. Это значительно облегчит как формирование соответствующих SC-текстов, так и их обработку и понимание.

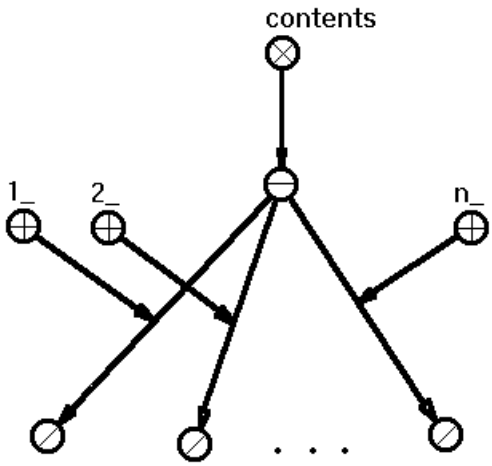

Рис.2.3. Общий вид SC-конструкции, отражающей структуру учебного материала

В линейном виде использование данного отношения будет иметь следующий вид:

***contents* -> Планиметрия ->**
***1_:* Основные_свойства_простейших_геометрических_фигур,**
***2_:* Углы,**
***3_:* Признаки_равенства_треугольников,**
***4_:* Сумма_углов_треугольника,**
5_:* Геометрические_построения, *... ;

***contents* ->**
Основные_свойства_простейших_геометрических_фигур ->
***1_:* Точка_и_прямая,**
***2_:* Основные_свойства_принадлежности_точек_и_прямых,**
***3_:* Основные_свойства_взаимного_расположения_точек_на_прямой_и_на_плоскости,**
4_:* Основные_свойства_измерения_отрезков_и_углов, *... ;
и т.д.

section - знак отношения, задающего содержание некоторого раздела учебного материала (см. рис. 2.4). С помощью атрибута ***name***_ данного отношения указывается название раздела, атрибут ***text***_ указывает на тектовое содержание раздела, атрибут ***keyWord***_ указывает на множество ключевых понятий, описываемых в рамках данного раздела, *picture*_ - на множество рисунков к тексту раздела, ***animation***_- демонстрационных роликов, ***sound***_ - звукового сопровождения к разделу, которое будет выводиться пользователю в случае обращения к данному разделу. В качестве такого звукового сопровождения может быть либо некоторая музыкальная заставка, либо речевое краткое сообщение о содержании раздела и рекомендациях по его прочтению, а, при желании, разработчик ИОС может озвучить содежимое раздела и выводить его полностью в речевом виде. Заметим также, что при описании содержимого раздела необязательно вводить все элементы.

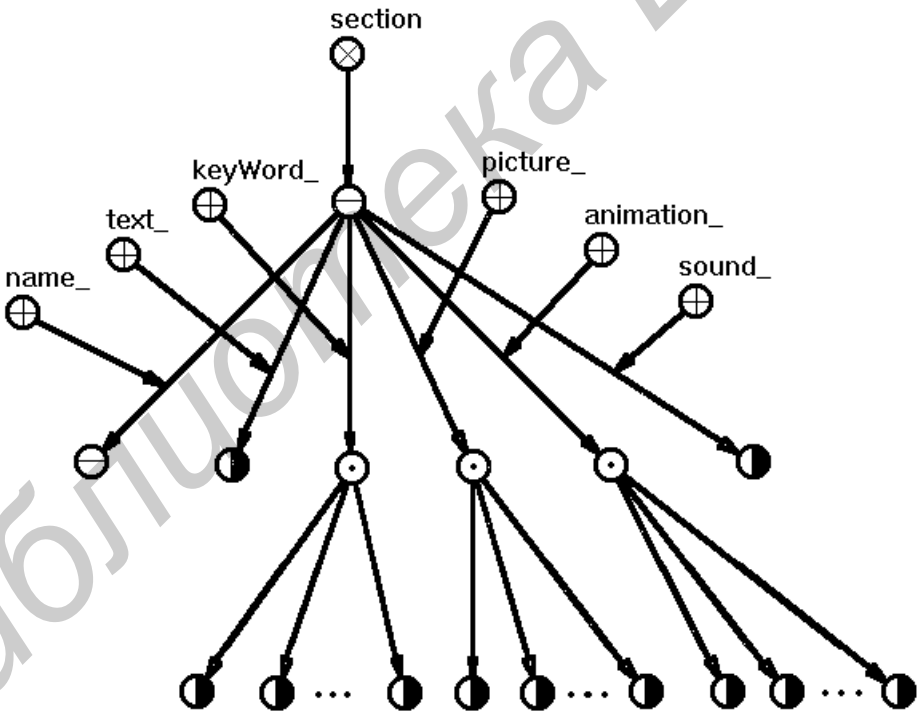

Рис. 2.4. Общий вид SC-конструкции, отражающей содержимое раздела учебного материала

Например, раздел может содержать только текст, и соответствующая SC-конструкция будет иметь вид:

section -> (name_: **Точка_и_прямая,** ***text_: /”Основными геометрическими фигурами на плоскости являются точка и прямая. ...”/***);

Однако для того, чтобы отразить взаимосвязь разделов и вводимых в них понятий рекомендуется описывать среди компонентов данного отношения множество ключевых понятий. Это значительно облегчит дальнейшую обработку фрагментов учебного материала и, заметим, заставит разработчика представить учебный материал в более структурированном виде. Так, приведенная выше конструкция при добавлении множества ключевых понятий примет следующий вид:

section -> (name_: **Точка_и_прямая,** ***text_: /”Основными геометрическими фигурами на плоскости являются точка и прямая. ...”/, keyWord_: (***point, straight***)***);

Заметим также, что содержимое узла, помеченного атрибутом ***text_***, может быть оформлено как текстовый файл, сформированный с помощью некоторого текстового редактора, либо как гиперсреда, включающая текстовые, графические, анимационные, звуковые фрагменты (см. ниже отношение ***hiperMedia***). В последнем случае SC-узел, обозначающий текст раздела, будет являться знаком кортежа отношения ***hiperMedia***.

hiperMedia - знак отношения, описывающего гиперсреду, т.е. семантически связанный фрагмент учебного материала, включающий основной текст (помечаемый атрибутом ***text_***) с явными ссылками на некоторые ключевые понятия (выделенные с помощью атрибута ***keyWord_***), о которых можно будет при работе в гиперсреде запросить дополнительную информацию, а также элементы мультимедиа, выделяемые с помощью атрибутов ***picture_***, ***animation_*** и ***sound_***. Все компоненты отдельного фрагмента гиперсреды включаются в него в определнной последовательности, которая задается с помощью порядковых атрибутов ***1_***, ***2_***, ... ***n_***. Количество элементов отношения ***hiperMedia*** никак не ограничивается, однако, обязательным является наличие

элементов, помеченных атрибутами ***text_*** и ***keyWord_***. Общий вид данного отношения приведен на рис.2.5.

В отличие от элементов отношения ***section***, в котором некоторые элементы являются знаками множеств ключевых понятий, рисунков, роликов, каждый из элементов отношения ***hiperMedia*** обозначает конкретное ключевое понятие, рисунок, ролик, звук и обязательно должен иметь содержимое в виде соответствующего текста или файла. Следует также заметить, что элемент, помеченный атрибутом ***sound_***, может быть дополнительно помечен порядковым атрибутом, которым уже помечен некоторый элемент этого же отношения. Это будет означать то, что при выводе пользователю содержимого этого элемента должно выводиться одновременно и звуковое сопровождение из файла в содержимом узла, помеченного атрибутом ***sound_***.

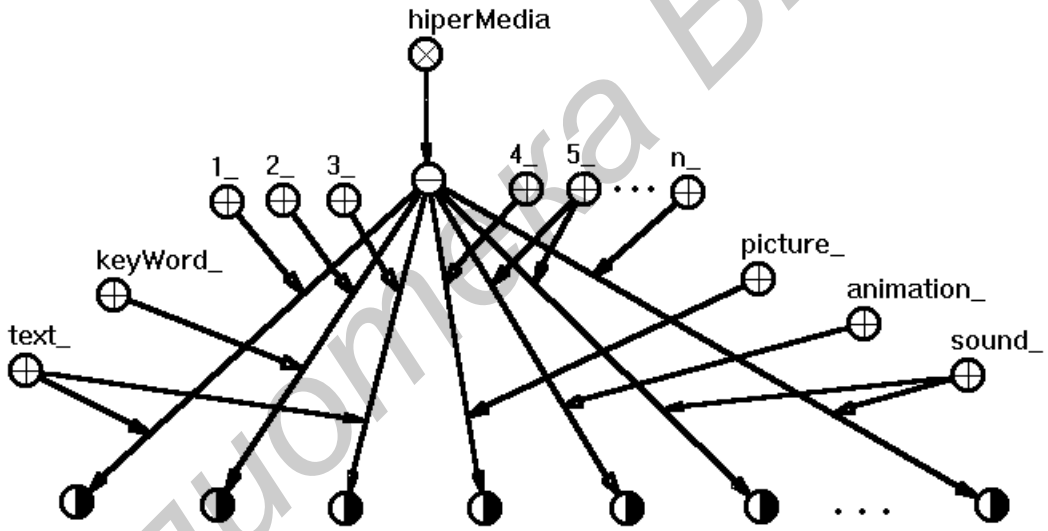

Рис.2.5. Общий вид SC-конструкции для представления информации в гиперсреде

themePlan - знак отношения, описывающего тематический план, с помощью которого задается последовательность подачи учебного материала при изучении некоторого конкретного понятия, знак которого помечается атрибутом ***object_*** (см. рис. 2.6.). Последовательность подачи учебного материала задается атрибутами ***1_***, ***2_***, ... ***n_***. Элементами данного отношения, как правило, являются знаки определений, разделов, фрагментов гипресред, рисунков и

т.д. При этом логично предположить, что указанные в рамках отдельного тематического плана элементы упорядочены в порядке возрастания сложности соответствующего учебного материала. Внесение в УБЗ тематических планов не является обязательным и рекомендуется для того, чтобы обеспечить более гибкое и эффективное функционирование ИОС при поиске ответов на вопросы пользователя. Так, например, обучаемый может попросить “рассказать” ему что-нибудь о треугольниках. В случае наличия соответствующего понятию треугольника тематического плана система для начала выберет его первый элемент и выведет пользователю. Затем, получив подтверждение на вывод дополнительной информации, приступит к дальнейшему ознакомлению пользователя с информацией о запрошенном понятии согласно существующего тематического плана. В случае же отсутствия такового системе придется предварительно сгенерировать соответствующий тематический план в результате поиска по всей УБЗ информации, имеющей отношение к вопросу пользователя.

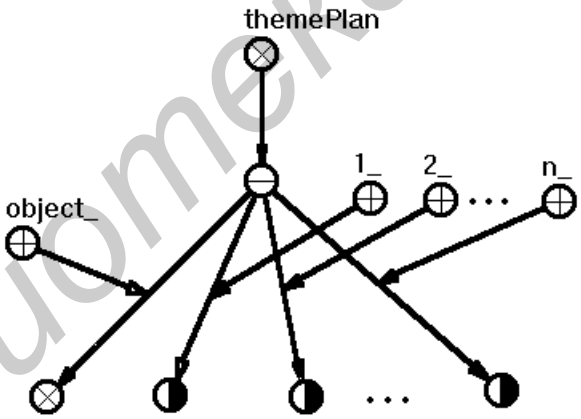

Рис.2.6. Общий вид SC-конструкции для представления тематического плана

complication - знак отношения, задающего степень сложности элементов учебного материала (см. рис. 2.7). Степень сложности задается с помощью некоторой абсолютной или условной числовой шкалы (см. ключевые узлы языка SC ***scale***, ***absScale***, ***convScale***). Тип шкалы выбирает разработчик ИОС, однако, при выборе необходимо учитывать, что при наращивании или модификации УБЗ в дальнейшем должна будет использоваться ранее выбранная шкала. Элементами данного отношения являются SC-узлы, обозначающие

те или иные элементы учебного материала, например, знаки разделов, текстов определений, комментариев и т.п.

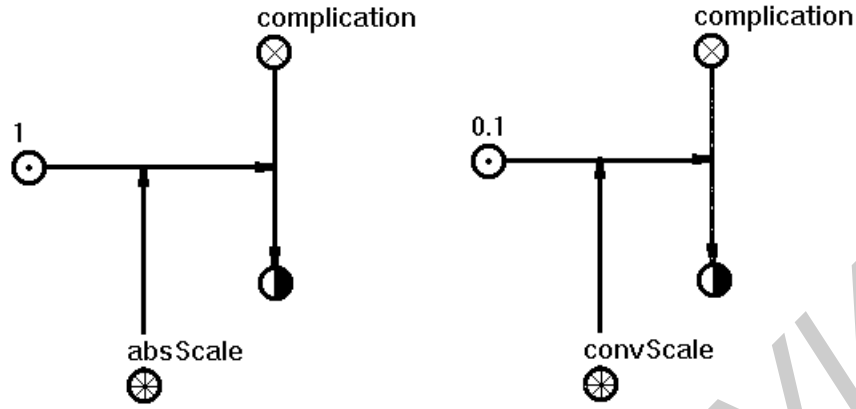

Рис.2.7. Способы представления на языке SC информации об уровне сложности элементов учебного материала

Заметим также, что в составе одной и той же подсистемы решения задач и ознакомления с предметной областью может быть создано несколько формальных моделей, описывающих различные предметные области. Это позволяет осуществлять режим обучения по различным предметным областям.

2.3. Операции переработки информации об учебном материале в графодинамической памяти

Как было сказано выше, механизмы переработки знаний в подсистеме решения задач и ознакомления с предметной областью представляют собой набор SCL-операций [8], с помощью которых в результате обработки БЗ предметной области осуществляется решение задач, и их аналогов, которые предназначены для обработки УБЗ с целью ознакомления пользователя с учебным материалом. Применение набора SCL-операций для решения геометрических задач подробно рассмотрено в [13].

Ниже будут кратко рассмотрены операции переработки УБЗ, которые делятся на следующие основные группы:

1) операции навигации по разделам учебного материала в соответствии с

его структурой;

2) операции вывода пользователю содержимого заданного раздела;

3) операции поиска ответов на вопросы справочного характера;

4) операции ранжирования учебного материала по степени сложности и определения средней степени сложности заданных совокупностей элементов учебного материала.

Выполнение каждой из операций происходит в том случае, когда в базу знаний ИОС попадает соответствующее данной операции задание. Это задание формулируется в виде SC-конструкции, описывающей запрос типа ***goalView_***. Вид и типология запросов более подробно рассмотрены ниже.

2.3.1. Операции навигации по разделам учебного материала в соответствии с его структурой

Операции навигации по разделам учебного материала в соответствии с его структурой предназначены для ознакомления пользователя со структурой учебного материала с различной степенью подробности и для обеспечения быстрого перехода от одного раздела учебного материала к другому. Суть этих операций сводится к поиску элементов отношения ***contents***, связь между которыми отражается путем включения знаков кортежей данного отношения в состав этого же отношения, но уже в качестве элементов другого кортежа.

Активизация указанных операций происходит в результате поступления соответствующего запроса в подсистему решения задач и ознакомления с предметной областью. В этом запросе указывается:

- структуру какого раздела необходимо отобразить пользователю (путем указания знака этого раздела в структуре запроса),
- с какой степенью подробности отображать запрашиваемую структуру (путем указания глубины поиска, т.е. количества вложений подразделов в состав указываемого раздела);
- направление поиска (вверх, вниз или в обе стороны по структуре учебного материала);
- в каком виде выводить найденную информацию - в тектовом или в графовом (т.е. в виде соответствующей SC-конструкции). В последнем случае

пользователь будет иметь возможность самостоятельно переходить от одного раздела к другому путем указания одного из интересующих его разделов в составе выведенных на экран;

- режим навигации (с возможностью редактирования структуры или без).

Данный запрос может быть сформирован либо самой ИОС в соответствии с текущим состоянием процесса обучения, либо сгенерирован ею в результате поступления вопроса пользователя.

К группе операций навигации по разделам учебного материала относятся также операция просмотра структуры учебного материала и операции ее модификации. К операциям модификации структуры учебного материала относятся операции добавления новых элементов, удаления элементов, редактирования наименований элементов структуры учебного материала.

Ниже даны краткие спецификации указанных операций.

OPViewContents

Операция просмотра структуры учебного материала (рис. 2.8). В результате реализации данной операции пользователю на экран выводятся элементы, задающие структуру разделов темы, знак которой указан в качестве элемента запроса.

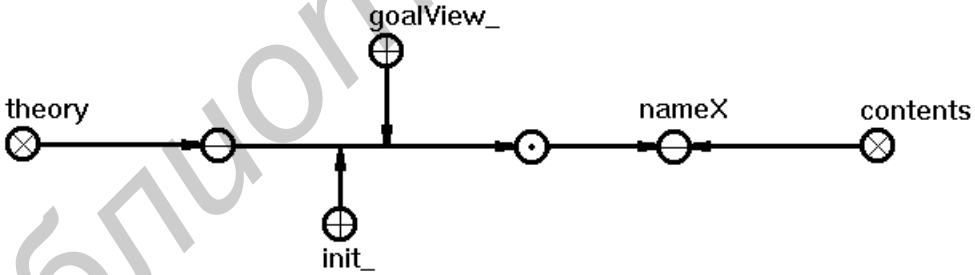

Рис. 2.8. Вид запроса на реализацию операции просмотра структуры учебного материала

OPAddContents

Операция добавления новых элементов в структуру учебного материала (рис.2.9, 2.10). В результате реализации данной операции в SC-памяти генерируется SC-конструкция, заданная в структуре запроса. Если знак кортежа указанного в структуре запроса отношения ***contents*** является SC-константой, то в составе этого кортежа генерируются указанные дополнительные элементы. В случае, если указанные в структуре запроса элементы присутствуют в составе

заданного кортежа, то производится их модификация, т.е. замена на указанные в запросе элементы.

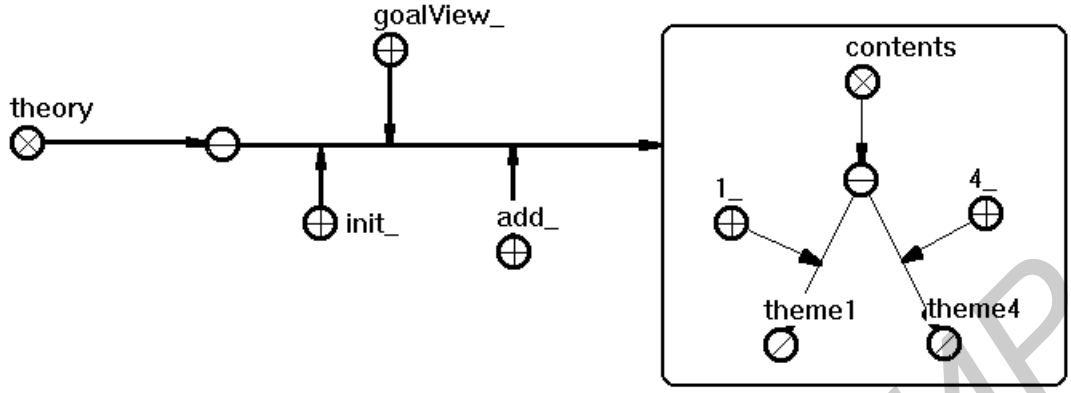

Рис. 2.9. Вид запроса на реализацию операции добавления новых элементов в структуру учебного материала

Кроме того, существует модификация данной операции, осуществляющая вставку некоторого подраздела между двумя заданными (см. рис. 2.10). В результате реализации данной операции происходит добавление нового подраздела с последующей перенумерацией последовательности подразделов в составе заданного. Следует также заметить, что поиск элементов, между которыми необходимо вставить новый подраздел, может осуществляться по двум признакам: либо по их порядковым номерам (заданным с помощью порядковых атрибутов ***1***_, ***2***_, ... ***n***_, либо по наименованиям. В последнем случае порядковые номера заданных подразделов в структуре запроса указывать необязательно).

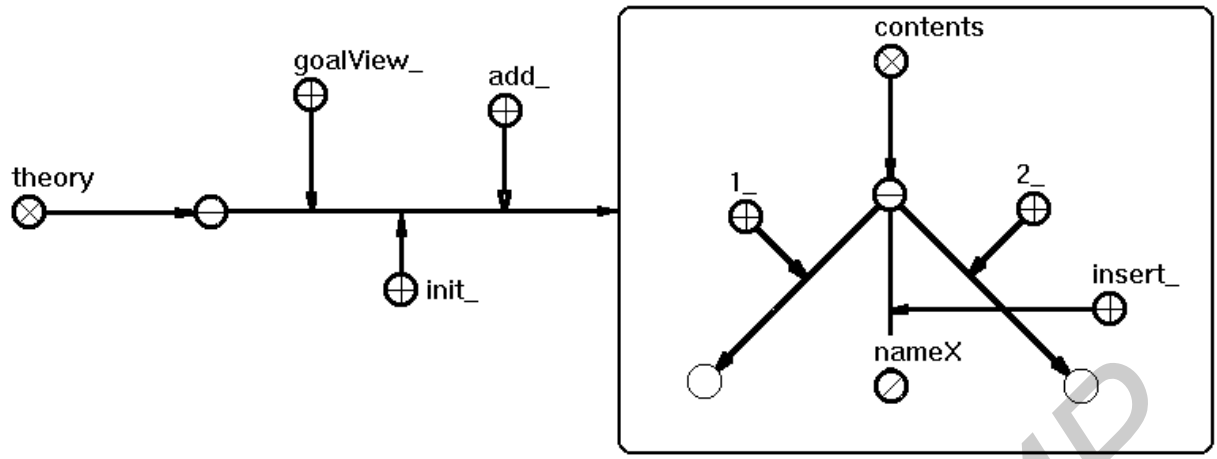

Рис. 2.10. Вид запроса на реализацию операции вставки нового элемента в структуру учебного материала между двумя заданными

OPDeleteElemContents

Операция удаления элементов из структуры учебного материала (рис. 2.11). В результате реализации данной операции из SC-памяти удаляются указанные в структуре запроса элементы.

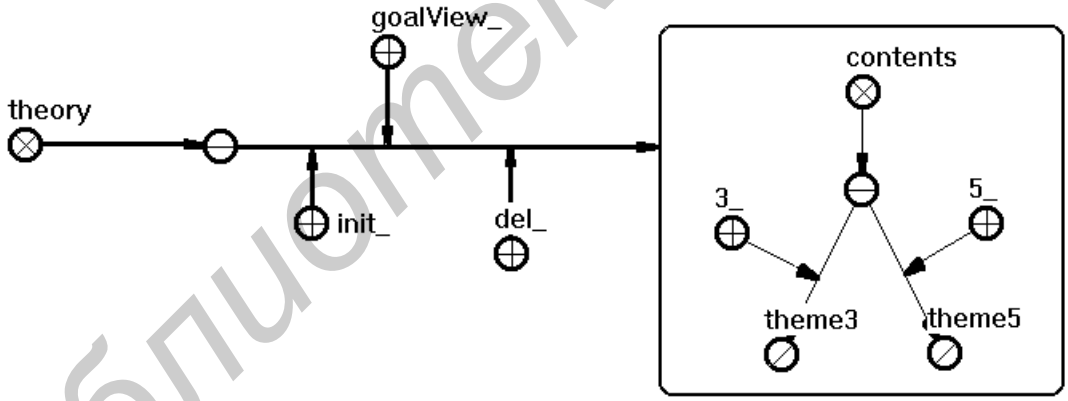

Рис. 2.11. Вид запроса на реализацию операции удаления элементов из структуры учебного материала

2.3.2. Операции вывода пользователю содержимого заданного раздела

Операции вывода пользователю содержимого заданного раздела осуществляют обработку кортежей отношения ***section*** и выводят на экран пользо-

вателю соответствующий заданному разделу текст, а также дополнительную информацию, относящуюся к этому разделу и указанную в составе рассматриваемого отношения. Эти операции, так же, как описанные выше, активизируются в случае наличия в БЗ соответствующего запроса, в котором указывается:

- текст какого раздела необходимо вывести (путем указания в структуре запроса SC-узла, обозначающего его наименование, либо знаков ключевых слов, которые встречаются в запрашиваемых разделах);

- какие дополнительные компоненты к тексту указанного раздела необходимо вывести;

- какой степени сложности материал выводить пользователю.

Таким образом, к операциям вывода содержимого заданного раздела относятся операции вывода текста раздела по его наименованию, поиска и вывода содержимых разделов, имеющих заданный в запросе набор ключевых слов, вывода содержимого указанного одним из способов раздела в виде гиперсреды, вывода отдельных рисунков и/или демонстрационных роликов, отосящихся к заданному разделу.

Реализация операций данного класса сводится к обработке SC-конструкций, описываемых с помощью отношений ***section*** или ***hiperMedia***, и инициируется активным запросом типа ***goalView_***. В соответствии со структурой запроса набор операций указанного класса подразделяется на следующие рассматриваемые ниже типы.

OPOutputSection

Операция вывода текста раздела по его наименованию. Единственным элементом запроса (см. рис. 2.12, а) в этом случае является SC-узел с наименованием соответствующего раздела.

OPFindSection

Операция поиска и вывода содержимых разделов, имеющих заданный в запросе набор ключевых слов. В данном случае в структуре запроса (см. рис.2.12, б) указывается множество ключевых узлов, в соответствии с которым будут найдены разделы, содержащие указанный набор ключевых слов. Данный тип запроса, как правило, формируется автоматически в процессе поиска

дополнительной информации о понятиях предметной области, обозначенных заданным набором ключевых слов.

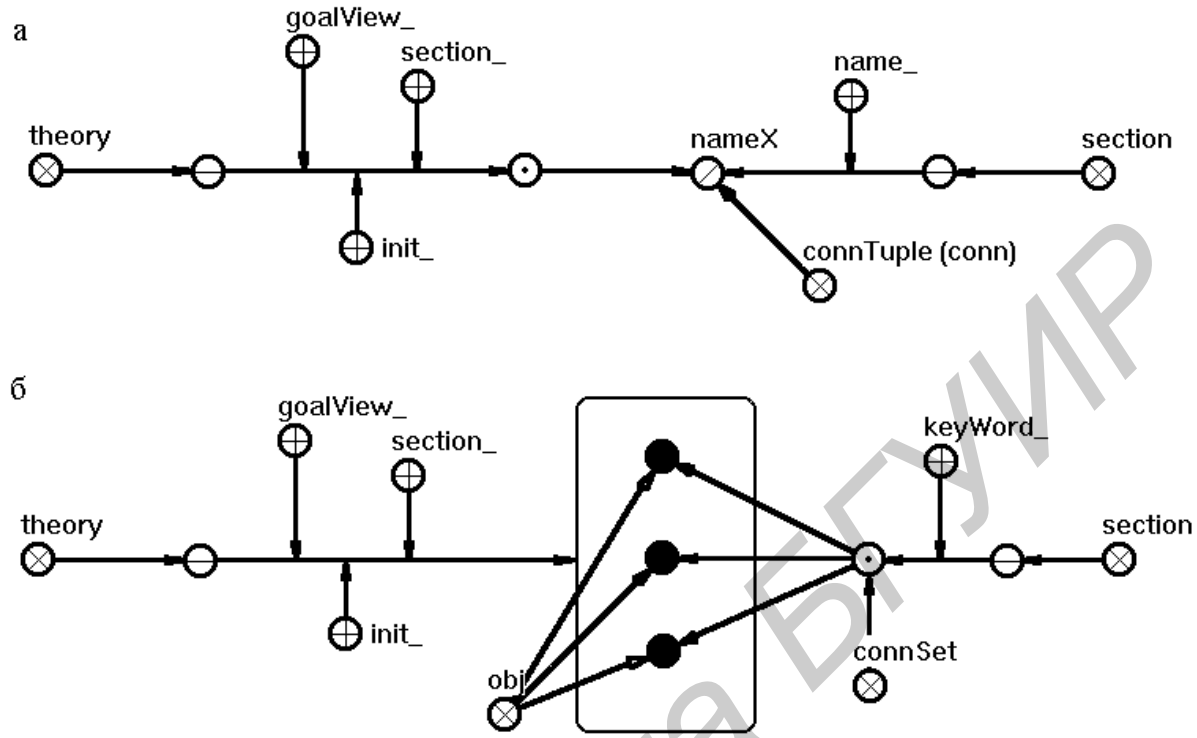

Рис. 2.12. Вид запроса на реализацию операции вывода текста раздела:
а - по его наименованию, б - по набору ключевых слов

OPOutputHiperSection

Операции вывода содержимого раздела, указанного одним из способов (по наименованию или набору ключевых слов), в виде гиперсреды. Данный набор операций предназначен для поддержки работы с гиперсредой, т.е. для осуществления диалогового взаимодействия с пользователем в режиме просмотра элементов гиперсреды. Виды запросов, обрабатываемых данной операцией, приведены на рис. 2.13.

OPOutputSectionIllustration

Операция вывода отдельных рисунков и/или демонстрационных роликов, относящихся к заданному разделу. Данная операция инициируется запросом, элементом которого является знак(и) конкретного рисунка и/или ролика.

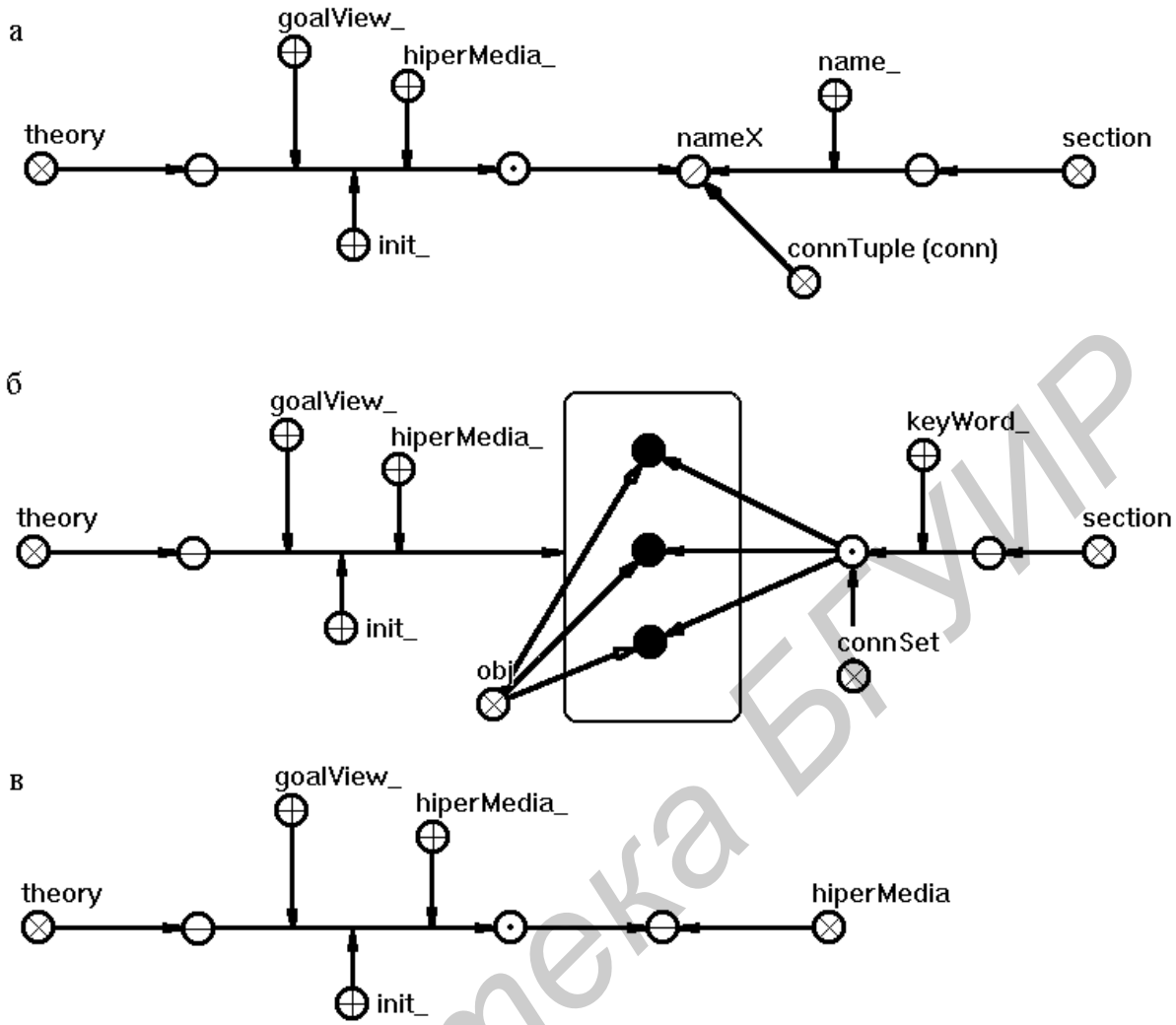

Рис. 2.13. Вид запроса на реализацию операции вывода текста раздела в виде гиперсреды: а - по наименованию раздела, б - по набору ключевых слов раздела, в - по знаку соответствующей гиперсреды

2.3.3. Операции поиска ответов на вопросы справочного характера

Операции поиска ответов на вопросы справочного характера обрабатывают SC-конструкции, описываемые с помощью отношений ***wordDef***, ***wordSem***, ***comm***, ***nearSem***, ***themePlan***, и в соответствии с видом обрабатываемого запроса каждая из этих операций осуществляет поиск в УБЗ структур, задающих соответственно определения понятий предметной области,

комментариев к понятиям, совокупностей семантически связанных понятий, тематические планы.

Набор указанных операций подразделяется на следующие типы.

OPOutputDefinition

Операция вывода определения понятия предметной области. В результате реализации данной операции производится поиск в SC-памяти всех SC-конструкций типа ***wordDef***, задающих текстовые определения указанного в запросе понятия.

OPOutputComments

Операция вывода комментариев к понятиям ПОб. В результате реализации данной операции производится поиск в SC-памяти всех SC-конструкций типа ***comm***, задающих текстовые комментарии к указанному в запросе понятию.

OPOutputNearSem

Операция поиска и вывода на экран совокупностей семантически связанных понятий. В результате реализации данной операции производится поиск в SC-памяти всех SC-конструкций типа ***nearSem***.

Общий вид запроса на реализацию данной группы операций приведен на рис. 2.14. С помощью данного запроса можно формировать задания не только на вывод информации некоторого одного типа (например, определения понятия), но и на вывод совокупной информации о заданном понятии (например, определения, семантики и комментариев). В последнем случае описание типа запроса осуществляется с помощью совокупности соответствующих атрибутов.

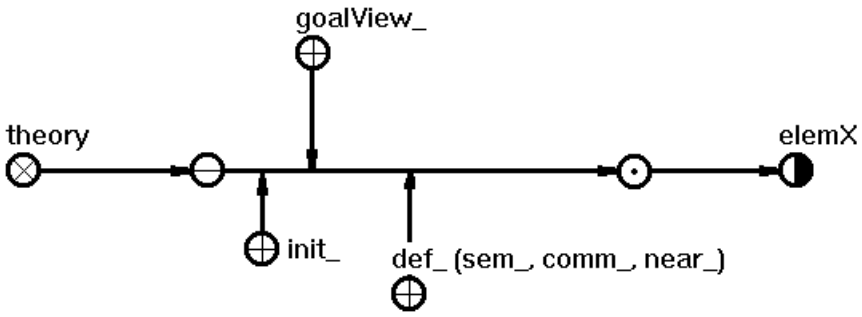

Рис. 2.14. Вид запроса на реализацию операций поиска ответов на вопросы справочного характера

К классу операций вывода пользователю информации справочного характера относятся также следующие:

OPGenerateThemePlan - операция генерации тематического плана;

OPRealizationThemePlan - операция реализации тематического плана.

2.3.4. Операции ранжирования учебного материала по степени сложности и определения средней степени сложности заданных совокупностей элементов учебного материала

Операции ранжирования учебного материала по степени сложности и определения средней степени сложности заданных совокупностей элементов учебного материала используются операциями вывода пользователю фрагментов учебного материала в порядке возрастания (убывания) степени сложности последнего и для выдачи характеристики сложности представляемого системой или освоенного учебного материала. Эти операции могут использоваться также подсистемой управления обучением при анализе уровня знаний обучаемого. Заметим, что если разработчик системы не вносит в УБЗ числовые характеристики сложности учебного материала, то эти операции при реализации ИОС не используются.

3. СЕМАНТИЧЕСКИЙ ГРАФОВЫЙ ЯЗЫК ОПИСАНИЯ СОСТОЯНИЯ ОБУЧАЕМОГО

Для представления знаний об обучаемом при описании модели обучаемого [14] рассмотрим язык SCT (Semantic Code Tutoring), который является подъязыком языка SC.

3.1. Описание информации об обучаемом в виде графодинамических моделей

Для обозначения множества моделей обучаемых в данном языке введен ключевой узел ***student***, который является знаком отношения нефиксированной арности, каждый кортеж которого обозначает конкретного обучаемого. Каждый элемент кортежа дополнительно уточняется атрибутом, указывающим на конкретный тип информации об обучаемом:

startData_ - указывает на элемент ***Bs*** модели обучаемого, содержащий начальные сведения об обучаемом;

aim_ - указывает на цели обучения ***Ba***;

currentState_ - на описание текущего состояния обучаемого ***Bc***;

history_ - на историю взаимодействия обучаемого с системой ***H*** (рис.3.1).

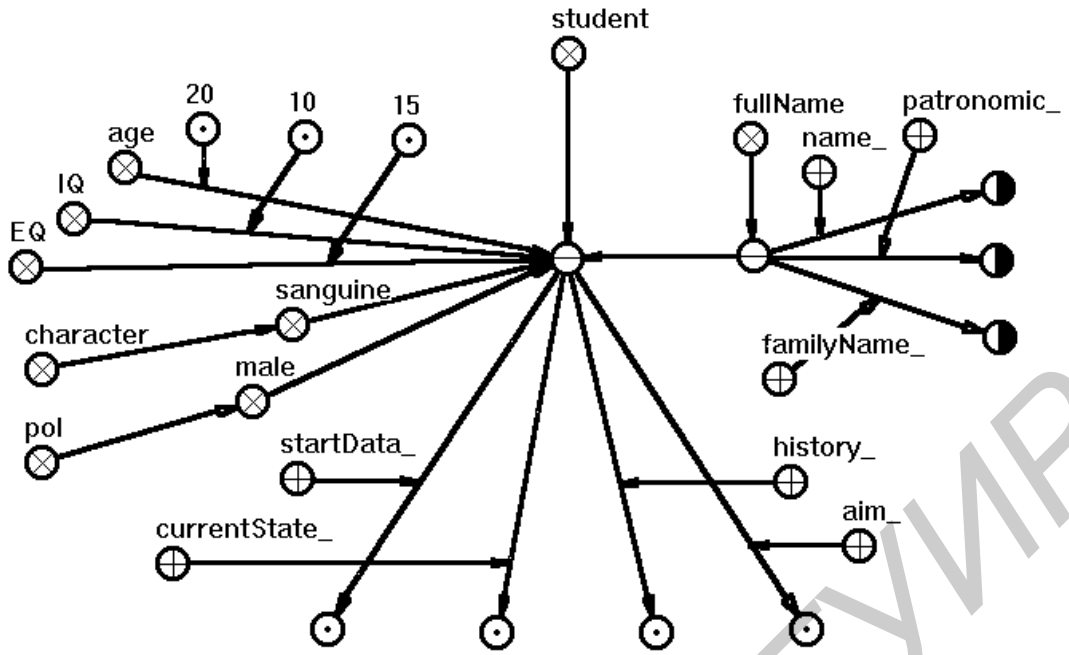

Рис.3.1. Общий вид SC-конструкции, описывающей информацию об обучаемом

SC-узлы, помеченные атрибутами ***startData_***, ***currentState_*** и ***history_***, являются оболочками структур, описывающих некоторый тип знаний об обучаемом, т.е. из этих узлов выходят дуги в узлы и дуги структур, ими обозначаемых. Кроме того, для указания активного в данный момент пользователя, т.е. конкретного обучаемого, который работает в среде ИОС в текущий момент, используется отношение ***initStudent***, единственным элементом которого является знак этого обучаемого. Множество элементов данного отношения становится пустым, когда сеанс работы с обучаемым завершается.

Ниже рассматривается набор ключевых узлов, используемых для описания конкретных знаний об обучаемом.

Для описания базовой информации ***B*** об обучаемом используются следующие ключевые узлы.

fullName - знак отношения, задающего полное имя пользователя, атрибуты которого ***name_***, ***patronomic_***, ***familyName_*** указывают на имя, отчество и фамилию соответственно. Четвертым элементом данного отношения является знак конкретного пользователя.

pol - знак отношения, задающего пол пользователя. Данное отношение содержит два элемента, которые, в свою очередь, также являются знаками отношений: ***male*** - задает мужской пол, ***female*** - женский. Элементами указанных отношений являются знаки конкретных пользователей.

character - знак отношения, задающего типы характера, обозначаемые следующими отношениями: ***sanguine***, ***choleric***, ***phlegmatic***, ***melancholic***. Элементами указанных отношений являются знаки конкретных пользователей. Заметим, что при необходимости этот набор может быть расширен.

age, ***IQ***, ***EQ*** - знаки отношений, с помощью которых описываются числовые характеристики, соотвествующие конкретному пользователю. К таким числовым характеристикам относятся возраст, коэффициент интеллектуальности и коэффициент эмоциональности соответственно. Данные знаки отношений задают соответствующие шкалы измерений и поэтому являются элементами отношения языка SC ***scale***. В процессе создания конкретной ИОС разработчик имеет возможность к указанному набору отношений добавить новые отношения, например, для указания скорости реакции обучаемого и др. индивидуальных характеристик, которые по мнению эксперта необходимо использовать при анализе состояния обучаемого.

Для описания уровня знаний и умений обучаемого используется отношение ***knowCan***, атрибуты которого ***object_***, ***know_***, ***can_*** указывают на элемент учебного материала, знания и умения обучаемого соответственно (см. рис.3.2). Уровень знаний и умений задается путем указания степени достоверности соответствующих высказываний, описываемых с использованием нечетких SC-дуг, выходящих из соответствующих SC-узлов. При этом если степень достоверности не указана, то подразумевается, что обучаемый изучал указанное понятие или обучался умениям, но системе не известно, как он усвоил полученную информацию. Уточнение этого факта происходит в результате тестирования. Если указанная дуга негативная, это свидетельствует о том, что обучаемый не знает (не умеет) соответствующего материала. Так, например, на рис.3.2 представлена информация о том, что некоторый обучаемый по теме

“Понятие треугольника” ознакомлен с понятием “сторона”, не знает ничего о понятии треугольника и знает определение отрезка (причем уровень его знаний определен с достоверностью 0.3), а также не умеет вычислять периметр треугольника.

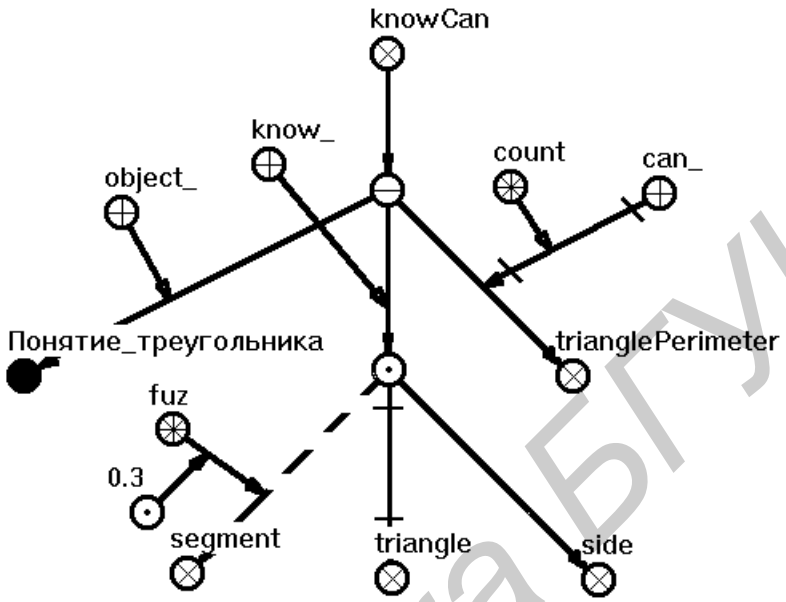

Рис.3.2. Представление на языке SCT информации об уровне знаний и умений обучаемого

Отношение ***knowCan*** может использоваться при описании как стартового уровня знаний и умений пользователя (в startData-компоненте модели обучаемого), так и текущего уровня (в currentState-компоненте). Кроме того, при задании целей обучения в aim-компоненте модели обучаемого может быть задан целевой уровень знаний и умений. В этом случае процесс обучения будет длиться до тех пор, пока стартовый или текущий уровень знаний и умений не станет равен целевому, что определяется путем сравнения SC-конструкций отношения ***knowCan*** в структурах соответствующих компонентов модели обучаемого.

Для описания текущего состояния и истории взаимодействия с системой учитываются основные состояния ИОС по отношению к обучаемому в соответствии с возможными стратегиями и тактиками обучения. Такими страте-

гиями являются:

- решение задачи;
- объяснение решения задачи;
- ознакомление с предметной областью;
- диалог с целью получения ответов на вопросы справочного характера;
- тестирование.

Для обозначения перечисленных стратегий вводятся соответствующие ключевые узлы: ***taskSolving, taskExplanation, theoryView, dialog***, ***testing***. Каждый из этих узлов является элементом множества стратегий, обозначаемого ключевым узлом ***strategy***. Рассмотрим подробно каждое из отношений.

taskSolving - знак унарного отношения, элементами которого являются SC-узлы, каждый из которых обозначает конкретную задачу, которую необходимо решить. Условие и запрос на решение задачи помещается в содержимое SC-узла, обозначающего задачу (см. рис.3.3). Кроме того, в содержимое этого же узла вносится условие, при котором активизированная стратегия решения задачи преобразуется в выполненную.

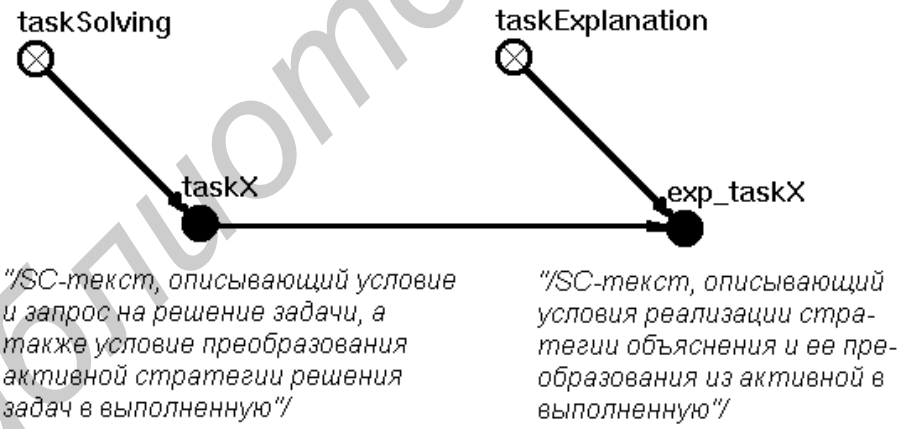

Рис.3.3. Представление на языке SC стратегий решения задач и объяснения

taskExplanation - знак унарного отношения, элементами которого являются SC-узлы, каждый из которых обозначает объяснение некоторой задачи. Если задача решена, то из знака задачи проводится SC-дуга в знак объяснения решения этой задачи (рис.3.3). По аналогии с представлением стратегии решения задач, в содержимое узла, обозначающего объяснение, помещается

SC-текст, описывающий условие реализации стратегии объяснения, а также условие, при котором активизированная стратегия объяснения преобразуется в выполненную.

theoryView - знак отношения, задающего стратегию ознакомления с предметной областью. Каждый кортеж данного отношения обозначает совокупость тактик, применяемых для реализации данной стратегии, а именно, с помощью атрибута ***object_*** указывается элемент учебного материала (в большинстве случаев это знак понятия предметной области), с которым необходимо ознакомить пользователя, а с помощью атрибута ***plan_*** указывается та часть учебного материала, которая соответствует изучаемому понятию, указанному под атрибутом ***object_***. Элемент, помеченный атрибутом ***plan_***, таким образом, может быть знаком кортежа одного из отношений ***themePlan***, ***contents***, ***section***, ***hiperMedia*** (рис.3.4).

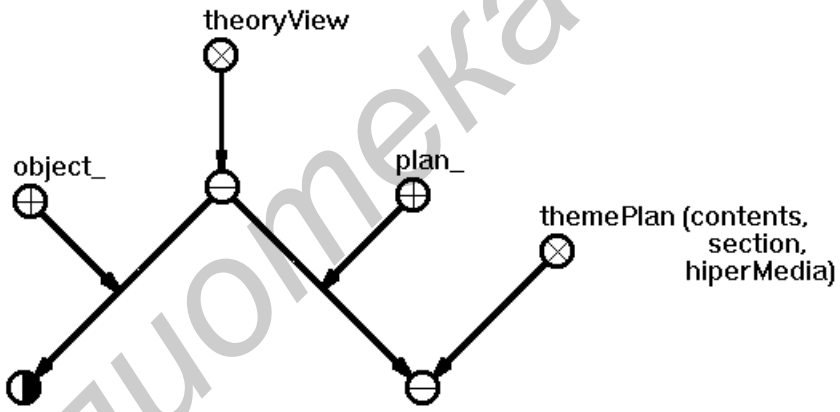

Рис.3.4. Отношение ***theoryView***

testing - знак бинарного асимметричного отношения, задающего применение стратегии тестирования по отношению к рассматриваемому обучаемому.

Наличие SC-конструкции, описывающей некоторую стратегию, в currentState-компоненте модели обучаемого указывает на то, что эта стратегия является активной и либо в текущий момент реализуется, либо завершена и результаты ее реализации должны быть перенесены в history-компонент. Если

же подобная SC-конструкция находится в history-компоненте, это свидетельствует о том, что она уже была применена по отношению к рассматриваемому обучаемому.

Множество активных (примененных по отношению к конкретному обучаемому) и подлежащих реализации стратегий обучения обозначается ключевым узлом ***active***, а множество реализованных (выполненных) стратегий обозначается ключевым узлом ***done***. Указанные ключевые узлы являются знаками множеств, состоящих из SC-дуг, и дуги из знаков указанных множеств проводятся в дуги, выходящие из знаков конкретных стратегий обучения.

Для отражения в модели обучаемого истории взаимодействия с системой имеются следующие ключевые узлы расширения языка SC. Для указания начала процесса обучения введен ключевой узел ***beginning***, являющийся знаком соответствующего унарного отношения. В составе модели конкретного обучаемого из данного узла должна выходить лишь одна дуга, что будет свидетельствовать о том, что для указанной в текущий момент цели обучения начало обучения было осуществлено с шага, знак которого является элементом данного отношения. Здесь под шагом обучения понимается либо применяемая по отношению к обучаемому некоторая стратегия или тактика, либо вопрос пользователя и т.п. В терминах языка SCL это знаки состояний ИОС. Для указания окончания процесса обучения с указанной в модели обучаемого целью введено отношение ***ending***, единственным элементом которого для конкретной модели обучаемого является знак последнего шага процесса обучения. Поскольку заданная цель обучения может быть достигнута за несколько сеансов обучения, то в текущий момент множество, обозначенное SC-узлом ***ending***, в составе конкретной модели обучаемого может быть пусто. В случае отсутствия цели обучения за один сеанс взаимодействия с системой могут быть сформированы оба множества ***beginning*** и ***ending***. Аналогами описанных отношений являются отношения, обозначенные SC-узлами ***first*** и ***last***, которые используются для указания начала и окончания отдельного сеанса обучения, в отличие от выше описанных отношений, которые используются для указания начала и окончания процесса обучения в целом (в соответствии с указанной целью обучения). Для того чтобы различать отдельные сеансы обучения, вводятся отношения ***date*** и ***time***, с помощью которых задаются дата и

время начала и окончания как процесса обучения, так и отдельного сеанса указанного процесса (см. рис.3.5). Кроме того, для указания текущей последовательности событий используется ключевой узел ***today***. Для задания последовательности применения каждого из шагов процесса обучения введен SC-узел ***next_***, выходящие из которого дуги входят в дуги, соединяющие знаки происходивших друг за другом шагов процесса обучения.

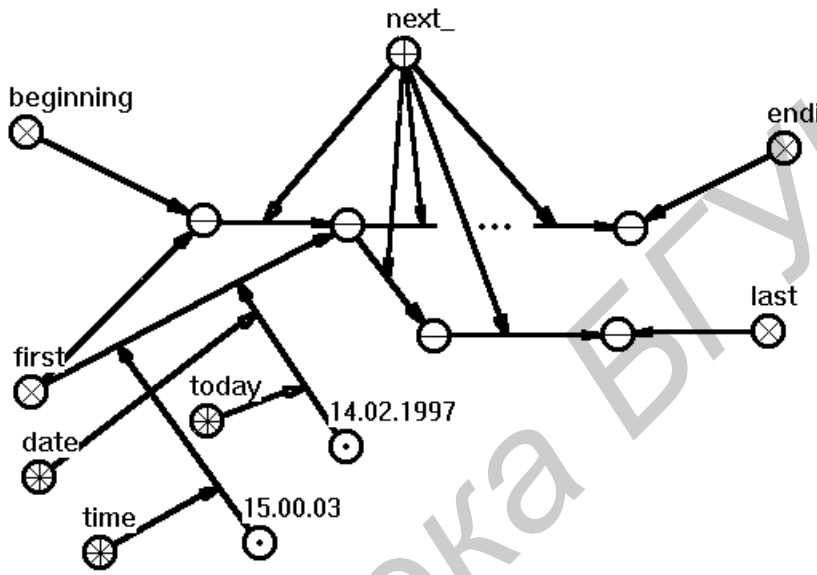

Рис.3.5. Представление на языке SC истории взаимодействия пользователя с системой

3.2. Операции переработки информации об обучаемом в графодинамической памяти

Помимо описанных выше средств представления знаний об обучаемом в состав языка SCT входят операции поддержки модели обучаемого в процессе функционирования ИОС. Совокупность этих операций разбивается на следующие основные группы:

1) операции модификации модели обучаемого;

2) операции анализа состояния обучаемого и выбора стратегий обучения;

3) операции ведения истории взаимодействия пользователя с ИОС.

Рассмотрим коротко каждую из указанных групп.

3.2.1. Операции модификации модели обучаемого

Данная группа операций осуществляет необходимые преобразования в SC-памяти по генерации и удалению SC-конструкций, описывающих модель обучаемого, т.е. задачами функционирования этих операций является поддержка текущего состояния описания модели обучаемого в БЗ ИОС. К ним относятся следующие операции.

OPInitStudent

Операция инициализации модели обучаемого. Эта операция инициируется в начале процесса взаимодействия пользователя с ИОС и предназначена для выбора или генерации в БЗ ИОС модели того обучаемого, с которым система начинает непосредственно взаимодействовать (далее такого пользователя будем называть активным). Целью данной операции является поиск в БЗ знака активного пользователя и включение его во множество, обозначенное ключевым узлом ***initStudent***. В случае же отсутствия в БЗ информации об указанном пользователе для него генерируется новая SC-конструкция, описывающая соответствующую ему модель обучаемого. Признаком, по которому производится поиск знака активного пользователя, является его полное имя (см. рис. 3.6, а). В случае, если модель обучаемого создается временно, т.е. на один сеанс работы, например, в составе некоторой другой (не обучающей) ИС, элементом запроса является некоторый переменный SC-узел stX (рис. 3.6, б), который в результате реализации данной операции становится знаком соответствующей временной модели обучаемого.

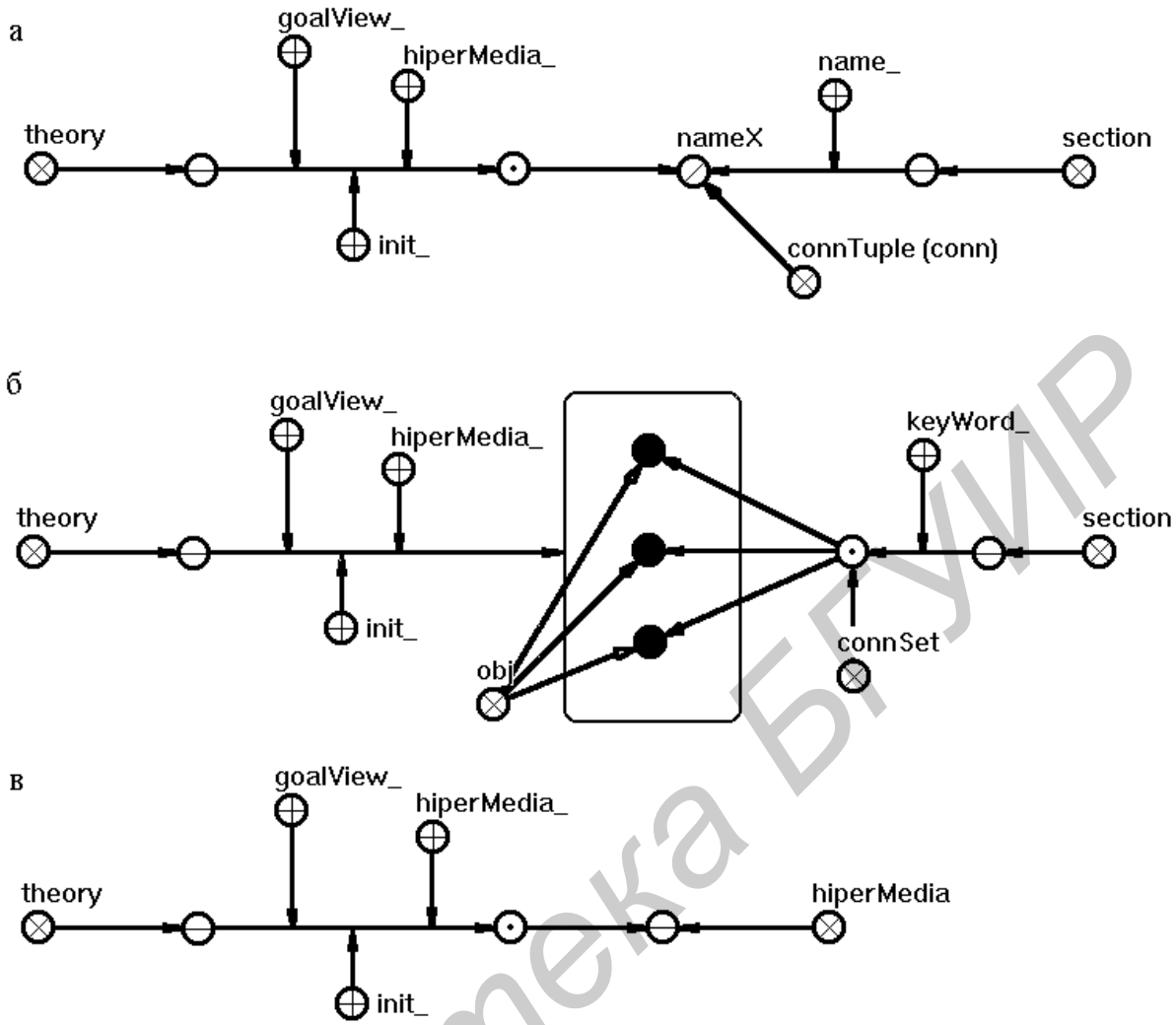

Рис. 3.6. Вид запроса на инициализацию модели обучаемого:
а - по полному имени; б - временно, т.е. без хранения имени

OPEndingStudy

Операция завершения сеанса обучения. Конечной целью данной операции является удаление знака активного обучаемого из множества, обозначенного ключевым узлом ***initStudent***. Данная операция инициируется при наличии в SC-памяти запроса (см. рис. 3.7), который формируется либо системой (в случае если цель обучения достигнута, о чем сообщается пользователю с целью получения подтверждения завершения работы), либо пользователем. В последнем случае также производится анализ текущего состояния знаний обучаемого для выяснения, достигнута ли цель обучения, после чего пользователю выводится соответствующее сообщение. Кроме того, производится форми-

рование соответствующих SC-конструкций в history-компоненте. К таким SC-конструкциям относятся конструкции типа ***last***, описывающие момент (дату, время) завершения сеанса работы с пользователем, и конструкции типа ***ending***, описывающие завершение процесса обучения в соответствии с указанной в модели обучаемого целью. В случае завершения процесса обучения пользователю отправляется запрос на удаление соответствующей модели обучаемого.

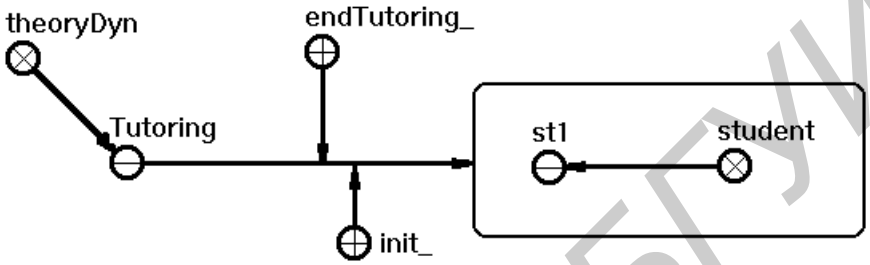

Рис. 3.7. Вид запроса на завершение сеанса обучения

OPDeleteStudent

Операция удаления модели обучаемого предназначена для удаления из БЗ ИОС SC-конструкции, описывающей обучаемого, имя или знак которого задано в соответствующем запросе (рис. 3.8). Этот запрос формируется либо системой после получения подтверждения от пользователя, завершающего обучение (см. предыдущую операцию), либо пользователем (как правило, учителем). При этом указанный в запросе обучаемый не обязан быть активным.

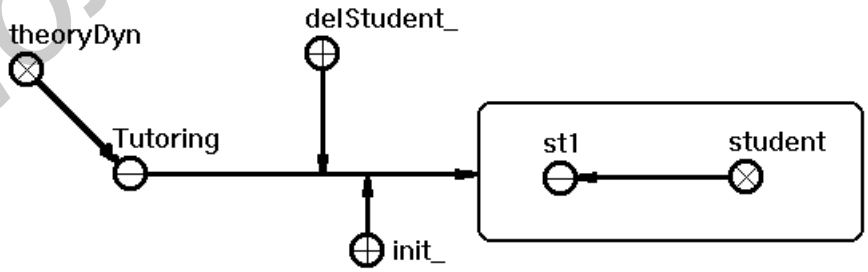

Рис. 3.8. Вид запроса на удаление модели обучаемого

OPBasisDataStudent

Операция формирования базовой информации об обучаемом. Данная операция осуществляет формирование SC-конструкций типа ***fullName***, ***pol***, ***character*** и т.п. в диалоге с активным пользователем. Заметим также, что выполнение данной операции не является обязательным для каждого обучаемого, так как базовая информация может быть введена учителем заранее в явном виде при формировании БЗ.

OPAimStudent

Операция постановки цели обучения формирует в модели обучаемого SC-конструкцию, описывающую цели обучения.

OPRefreshStudent

Операция обновления модели обучаемого осуществляет удаление всей информации из модели обучаемого, за исключением базовой. Данная операция инициируется с помощью запроса, общий вид которого приведен на рис. 3.9.

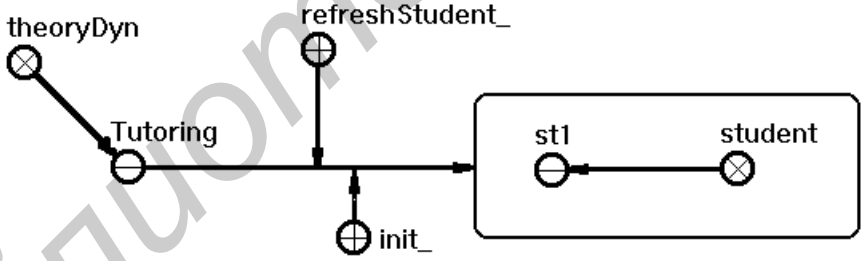

Рис. 3.9. Вид запроса на обновление модели обучаемого

3.2.2. Операции анализа состояния обучаемого и выбора стратегий обучения

Операции анализа состояния обучаемого и выбора стратегий обучения предназначены для осуществления адаптации к индивидуальным особенностям конкретного обучаемого. В соответствии с текущим состоянием обучае-

мого с помощью этих операций осуществляется выбор стратегий и тактик обучения для данного обучаемого по правилам, описываемым средствами языка SCL. Кроме того, к операциям анализа состояния обучаемого относятся также операции, осуществляющие вывод на экран пользователю (учителю или самому обучаемому) информации об обучаемом. Перед непосредственным выводом информации производится анализ состояния обучаемого. Для анализа состояния обучаемого могут быть использованы различные методы (например, логические, статистические). Особенности использования некоторого метода описываются разработчиком ИОС в правилах анализа состояния обучаемого на языке SCL и его расширении. Таким образом, для осуществления анализа состояния обучаемого и выбора стратегий обучения используется набор SCL-операций [8], а также дополнительный набор операций. Заметим, что одной из главных и наиболее часто используемых операций является операция сопоставления цели обучения с текущим состоянием знаний и умений обучаемого. Для корректировки знаний ИОС о состоянии обучаемого производится тестирование обучаемого в процессе выполнения операции активизации режима тестирования.

Таким образом, данный класс операций необходим для осуществления адаптации к индивидуальным особенностям конкретного обучаемого и включает следующие операции.

OPStrategyStudent

Операция выбора стратегий и тактик обучения для данного обучаемого. Данный выбор осуществляется по правилам, описанным на языке SCL и хранящимся в БЗ подсистемы управления обучением ИОС. Таким образом, выбор стратегий и тактик обучения осуществляется SCL-операциями реализации модели поведения системы.

OPOutputStudent

Операция вывода на экран пользователю (учителю или самому обучаемому) информации об обучаемом. Перед непосредственным выводом информации данной операцией производится анализ состояния обучаемого.

OPEGAimStudy

Операция сопоставления цели обучения с текущим состоянием знаний и умений обучаемого. Эта операция является одной из наиболее часто используемых и вызывается операциями модификации модели обучаемого.

OPInitTesting

Операция активизации режима тестирования необходима для выявления уровня знаний и умений обучаемого и , как следствие, корректировки знаний ИОС о состоянии обучаемого.

3.2.3. Операции ведения истории взаимодействия пользователя с ИОС

Операции ведения истории взаимодействия пользователя с ИОС осуществляют перенос информации из текущего состояния обучаемого (currentState-компонента модели обучаемого) в историю взаимодействия (history-компонент) с генерацией SC-конструкций, описывающих последовательность действий обучаемого и системы. К этому же типу операций относятся также операции, определяющие и сохраняющие дату и время сеанса обучения.

Краткие спецификации указанных операций указаны ниже.

OPNextStep

Операция генерации SC-конструкций, описывающих последовательность действий обучаемого и системы, т.е. конструкций типа ***next_***. Данная операция инициируется соответствующим запросом (рис. 3.10) на этапе завершения реализации некоторой стратегии или тактики обучения и прозводит перенос информации из currentState-компонента модели обучаемого в history-компонент.

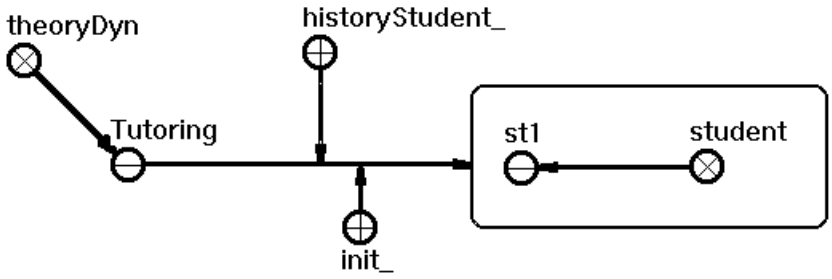

Рис. 3.10. Вид запроса на формирование истории взаимодействия ИОС с обучаемым

OPDataTime

Операции определения и сохранения даты и времени текущего сеанса обучения осуществляют запрос информации о текущей дате и времени у пользователя или операционной системы и заносят эту информация в history-компонент модели обучаемого с пометкой ***today***.

OPTimeStudy

Операции подсчета затраченного на обучение времени являются вспомогательными на этапе анализа модели обучаемого.

4. ГРАФОВЫЙ СЕМАНТИЧЕСКИЙ ЯЗЫК ЗАДАНИЙ И ВОПРОСОВ И СРЕДСТВА ОПИСАНИЯ СЕМАНТИЧЕСКОЙ СТРУКТУРЫ ДИАЛОГА

В [14] приведена типология вопросов, задаваемых пользователем системе. Рассмотрим средства языка SCQ (Semantic Code Questions) для представления и обработки этих вопросов с помощью однородных семантических сетей (примеры запросов приведены также выше при описании операций переработки знаний в подсистемах ИОС). В рамках данного языка формулировка каждого вопроса пользователя описывается в виде соответствующей SC-конструкции с использованием введенных в данном языке ключевых узлов, указывающих на тип вопроса. Таким образом, вопрос пользователя преобразуется в соответствующий запрос (задание), поступающий в память ИОС и об-

рабатываемый той подсистемой, которой он адресован.

Средства представления запросов на решение задачи описаны в [4]. В данной работе этот способ представления взят за основу, благодаря чему все запросы в ИОС представляются единым образом, что, заметим, значительно облегчает их обработку. Ниже описываются способы представления запросов ИОС на языке SCQ, дополняющие представленные в [4] средства.

Запрос на объяснение решения задачи имеет вид, аналогичный виду запроса на решение задачи, и формируется на основе выполненного (т.е. помеченного атрибутом ***goalEnd_***) запроса с помощью атрибута ***goalExplain_*** (рис. 4.1). Атрибут ***init_*** указывает на то, что данный запрос является активным, т.е. требует реализации. В примерах ниже этот атрибут опускается, так как активизация запроса производится подсистемой управления обучением автоматически в момент его поступления от пользователя.

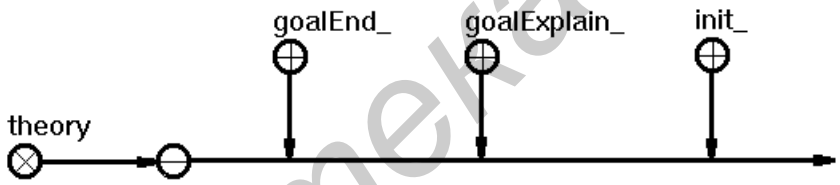

Рис. 4.1. Общий вид запроса на объяснение решения задачи

Для ***ознакомления с предметной областью*** формируется запрос с атрибутом ***goalView_***. Элементами запроса указанного типа являются элементы одного из отношений ***contents***, ***section***, ***hiperMedia*** или ***themePlan*** в зависимости от степени подробности поступившего от пользователя вопроса (рис. 4.2, 4.3). Этот запрос соответствует вопросу пользователя типа “покажи”, а также формируется в качестве подзапроса к вопросу типа “расскажи...”.

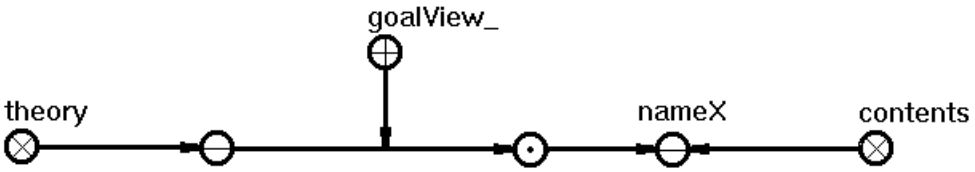

Рис. 4.2. Вид запроса на просмотр структуры учебного материала

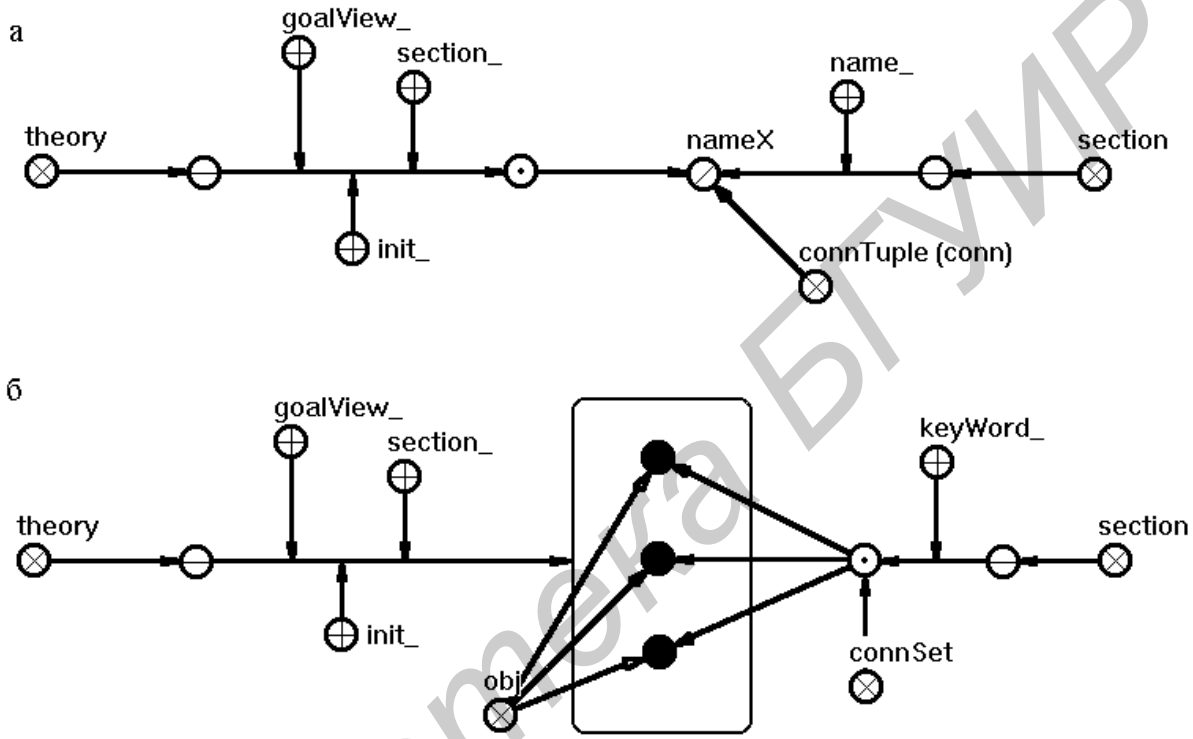

Рис. 4.3. Вид запроса на вывод текста раздела:
а - по его наименованию, б - по набору ключевых слов

Запросы, соответствующие вопросам обучаемого типа “расскажи все” и “расскажи что-нибудь”, формируются с помощью атрибутов ***goalTellAll_*** и ***goalTellAnything_*** соответственно. Элементами запросов указанного типа являются знаки элементов учебного материала, о которых спрашивает пользователь. Запросы данного типа формируются в составе подсистемы управления обучением, т.е. в составе нестационарной SCL-метатеории ***Tutoring***. Обработка этих запросов производится по правилам, описанным в SCL-метатеории ***st_learn***, и, как правило, приводит к формированию подзапросов вида ***goalView_*** в составе SCL-теории, описывающей предметную область, по ко-

торой ведется обучение. Например, на рис. 4.4 приведен пример формирования подзапроса вида ***goalView_*** на запрос типа ***goalTellAll_***, означающий, что в УБЗ необходимо найти все разделы, в которых указанное в структуре запроса понятие **termX** является ключевым словом. На рис. 4.5 приведен пример подзапроса к запросу вида ***goalTellAnything_***, согласно которому в УБЗ будут найдены и выведены пользователю все определения понятия **termX**.

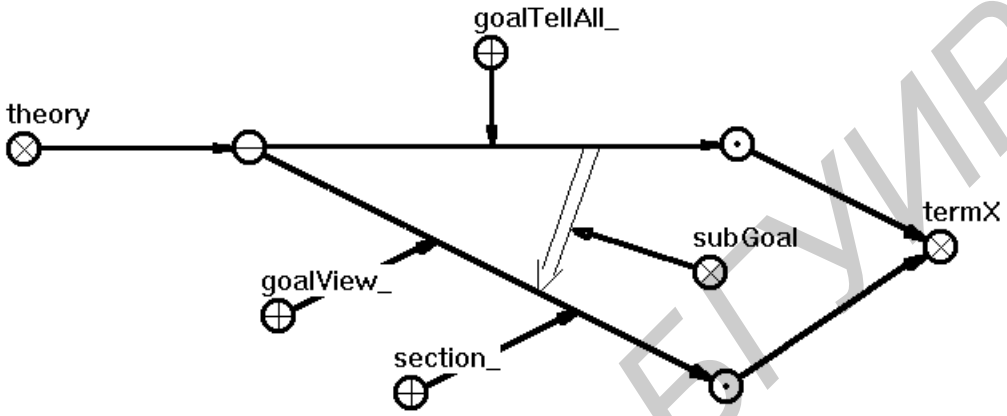

Рис. 4.4. Формирование подзапроса к запросу типа "расскажи все"

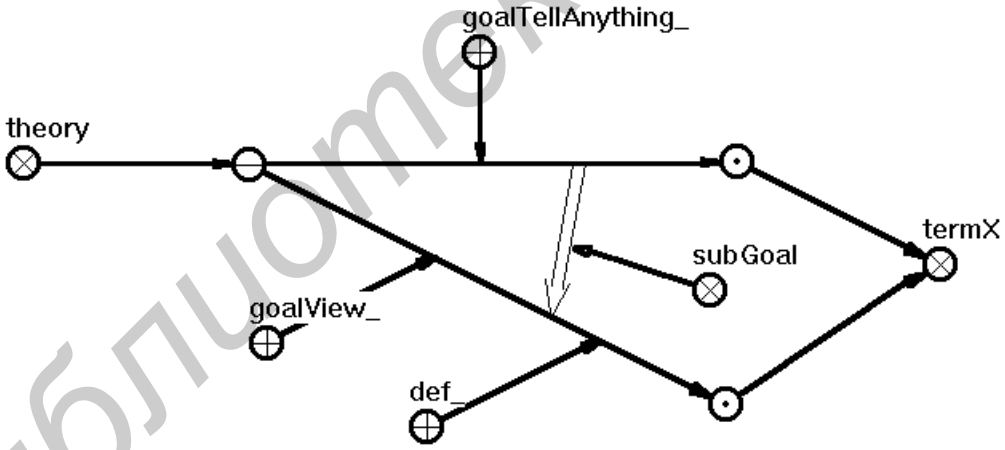

Рис. 4.5. Формирование подзапроса к запросу типа "расскажи что-нибудь"

Описанные выше запросы могут быть сформированы как пользователем, так и подсистемой управления обучением в процессе реализации некоторой стратегии обучения. Обработка и реализация указанных запросов производится набором соответствующих SCQ-операций, некоторые из которых были описаны.

5. ОПИСАНИЕ И РЕАЛИЗАЦИЯ ПРОЦЕССА УПРАВЛЕНИЯ ОБУЧЕНИЕМ С ИСПОЛЬЗОВАНИЕМ ОДНОРОДНЫХ СЕМАНТИЧЕСКИХ СЕТЕЙ

Осуществление управления обучением фактически представляет собой анализ информационных процессов, происходящих между пользователем и системой. Описание модели управления обучением сводится к формализации этих информационных процессов в виде метаинформации над информацией о пользователе и о процессе обучения. Это обстоятельство приводит к значительному усложнению структур БЗ в ИОС, и использование подхода на основе графодинамической парадигмы параллельной асинхронной переработки информации позволяет решить многие проблемы, возникающие при реализации механизмов функционирования ИОС.

Для реализации процесса управления обучением в ИОС требуется описание:

- правил функционирования каждой из подсистем в составе ИОС;
- правил и способов взаимодействия подсистем в составе ИОС между собой;
- правил и способов взаимодействия пользователя с системой в целом.

Описание правил функционирования подсистем в составе ИОС представляет собой описание соответствующей модели обучения. Поскольку архитектура ИОС нового поколения представляет собой иерархическую структуру, состоящую из ряда подсистем, то описание правил и способов их взаимодействия представляет собой описание процессов управления взаимодействием указанной иерархии систем. Благодаря избранноиу подходу единообразного описания и реализации процессов взаимодействия как подсистем между собой, так и системы с пользователем, описание правил и способов взаимодействия с пользователем производится теми же языковыми средствами. В связи с этим ниже более подробно будут рассмотрены средства описания сложноструктурированных моделей обучения и процессов управления взаимодействием иерархии подсистем.

5.1. Описание сложноструктурированных моделей обучения

В данной работе не реализуется некоторая конкретная модель управления обучением. Напротив, разработчику предоставляются средства описания любых моделей поведения ИОС. Выше были описаны средства описания статической модели обучение с помощью отношений ***hiperMedia*** и ***themePlan***. Реализация модели данного типа производится с помощью операций обработки указанных отношений.

Рассмотрим способы описания динамической модели обучения на языке SCL и его расширении.

Описание поведения ИОС соответственно динамической модели производится согласно принципам ситуационного управления [18], что позволяет с помощью предлагаемых средств описывать различные модели поведения [1]. Модель управления обучением представляет собой динамическую (нестационарную, семиотическую, изменяющуюся во времени) модель. В связи с этим для представления состояний модели управления обучением на языке SCL используется набор ключевых узлов языка SCL для описания семиотических моделей [4].

5.1.1. Описание семиотических моделей средствами языка SCL

Коротко рассмотрим основные принципы описания семиотических моделей (нестационарных предметных областей) средствами языка SCL, изложенные в [4].

"Знание о нестационарной предметной области на языке SCL представляется путем его расчленения на множество знаний о квазистационарных предметных областях, каждое из которых описывает некоторое состояние описываемой предметной области, трактуя нестационарную предметную область как стационарную на некотором промежутке времени по отношению к указываемым свойствам. Описание каждого такого состояния оформляется как SCL-теория, являющаяся state-компонентом SCL-метатеории, описывающей

нестационарную предметную область в целом. Для задания таких SCL-метатеорий вводится специальное отношение ***theoryDyn***.

Итак, SC-конструкция вида ***theoryDyn*** -> **ki** -> ***state_ :*** **bj** ; семантически означает, что **bj** есть высказывание, описывающее некоторое состояние (некоторую ситуацию) либо класс состояний (если высказывание **bj** имеет свободные переменные) той нестационарной предметной области, которая описывается высказыванием **ki** . Высказывание **bj** может быть либо атомарным (atomExpr-высказыванием), либо конъюнктивным (conj-высказыванием). Элементы атомарного высказывания **bj** и элементы const-компонента конъюнктивного высказывания **bj** могут иметь как тип ***const*** , так и тип ***var*** . При этом, если такой SC-элемент типа ***const*** является константой SCL-метатеории (элементом const-компонента высказывания **ki** , если **ki** <- ***theoryDyn***), то он называется стационарной константой. Если же указанный SC-элемент типа ***const*** (как узел, так и дуга) не является константой SCL-метатеории, то он называется нестационарной (ситуативной) константой.

Каждое высказывание, являющееся state-компонентом некоторого другого высказывания, в свою очередь само может также иметь несколько state-компонентов, которые осуществляют разбиение некоторого состояния (процесса) на несколько более "мелких" процессов. Если некоторое высказывание является state-компонентом и имеет свободные переменные, то оно представляет собой высказывание о существовании соответствующего состояния.

Конструкция **bi** -/> ***implDn_ : (pos :*** **be** **,** ***neg :*** **bt** ***)*** семантически означает, что для каждого состояния (state-компонента) процесса **bi** , которое удовлетворяет условию **be** , имеет место также и **bt** , т.е. implDn-компонент описывает общее свойство некоторого класса состояний соответствующего процесса.

Конструкция ***bi*** -> ***transfDn_ : (if_ :*** **be** **,** ***then_ :*** **bt** ***)*** семантически означает, что каждое состояние процесса **bi** , удовлетворяющее условию **be** , обязательно (независимо ни от чего) преобразуется в следующее за ним состояние, описываемое высказыванием **bt** . Нетрудно заметить, что transfDn-высказывания есть способ формального описания всевозможных причинно-следственных связей.

Конструкция

bi -> ***transfDnW_: (if_ :*** **be** **,** ***worker_ :*** **w** **,** ***then_ :*** **bt** ***)*** описывает преобразование состояния (ситуации) **be** в непосредственно следующее за ним состояние **bt** , к которому может привести некий исполнитель **w** . Описание условий, необходимых исполнителю **w** для выполнения указанного преобразования, также входит в состав высказывания **be** ."

Помимо перечисленных выше способов описания состояний семиотической модели, в языке SCL имеется также ряд отношений, заданных на множестве процессов и описывающих различные соотношения процессов во времени. К ним относятся следующие знаки отношений и атрибутов [4]: ***localBeginStateDn***, ***localEndStateDn***, ***localSpaceStateDn***, ***beginStateDn_***, ***endStateDn_***, ***eqBeginStateDn***, ***eqEndStateDn***, ***eqDurationStateDn***, ***comprBeginStateDn***, ***grt_***, ***comprEndStateDn***, ***comprDurationStateDn***, ***nextTimeStateDn***, ***next_***.

5.1.2. Расширение языка SCL для описания и реализации динамических моделей управления обучением

При подробном рассмотрении особенностей описания процессов управления обучением выяснилось, что приведенных в работе [4] средств не достаточно. Для более адекватного описания модели управления обучением в дополнение к описанным выше средствам разработаны расширенные средства языка SCL для описания динамических моделей поведения системы.

Поскольку моделирование и реализация динамических процессов сводится к анализу в каждый конкретный момент времени некоторой создавшейся к этому времени ситуации, которая определяет текущее состояние системы, необходимым оказалось более подробное рассмотрение типологии состояний динамической модели. В связи с этим еобходимо рассмотреть следующую типологию состояний ИОС:

1) ***фиксированное состояние*** - представляет собой некоторое промежуточное состояние динамической модели, фиксируемое в некоторый момент времени и описываемое некоторой конкретной SC-конструкцией, состоящей из стационарных и ситуационных констант соответствующей SCL-метатеории.

При описании правил перехода из одного состояния в другое совокупность подобных фиксированных состояний описывается изоморфными SC-конструкциями, в которых ситуационные константы обозначаются переменными SC-элементами (рис. 5.1). Здесь и далее под ***подобными состояниями*** будем понимать состояния, которые влекут за собой одинаковые действия системы над семантически однородными объектами;

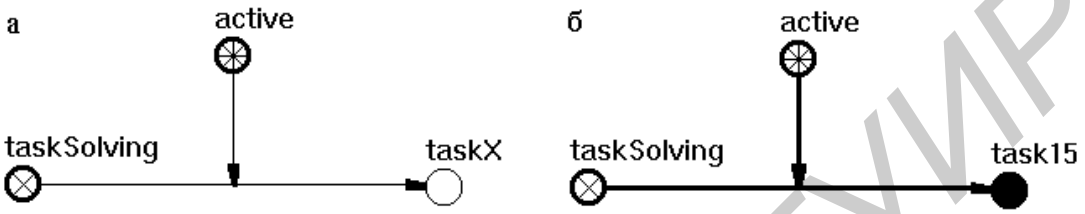

Рис. 5.1. Описание на языке SCL: а - совокупности подобных фиксированных состояний и б - конкретного фиксированного состояния

2) ***состояние перехода*** (переходное состояние). Описание переходного состояния представляет собой правило перехода из одного фиксированного состояния в другое. На языке SCL для описания переходных состояний используются ключевые узлы ***implDn_***, ***transfDn_***, ***transfDnW***, ***if_***, ***then_***, ***worker_***. При формальном описании на языке SCL различаются также ***общее*** описание переходного состояния и ***частное***. Общее описание представляет собой SCL-высказывание, которое постоянно хранится в SC-памяти и может быть применено многократно. Частное описание генерируется в SC-памяти в конкретный момент времени в результате каких-либо действий системы и после однократного применения удаляется. Для выделения множества частных описаний переходных состояний введено соответсвующее унарное отношение ***realizeAndDelete***. Пример общего и частного описания переходных состояний см. на рис. 5.5 и 5.6 соответственно;

3) ***состояние ожидания*** - при описании на языке SCL помечается в составе нестационарной SCL-метатеории атрибутом ***waitState_***. SC-узел, обозначающий состояние ожидания, является знаком SC-конструкции, описы-

вающей ожидаемое состояние фрагмента SC-памяти. Примером состояния ожидания в ИОС является ожидание завершения реализации некоторой стратегии обучения (рис. 5.2);

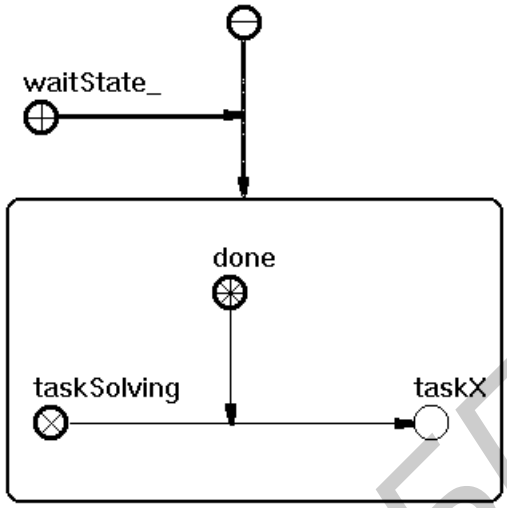

Рис. 5.2. Пример представления на языке SCL состояния ожидания

4) ***состояние преобразования текущего состояния SC-памяти*** - помечается в составе нестационарной SCL-метатеории атрибутом ***performState_***. Дуги, выходящие из SC-узла, обозначающего состояние преобразования SC-памяти, могут дополнительно помечаться атрибутами ***generate_*** - указывает на элемент, который необходимо сгенерировать в SC-памяти, ***delete_*** - указывает на элемент, подлежащий удалению, ***input_*** - указывает на SC-узел, содержимое которого необходимо загрузить в SC-память. На рис. 5.3 приведен пример описания состояния преобразования SC-памяти, где указывается, что необходимо удалить SC-дугу, выходящую из узла ***active***, и сгенерировать SC-дугу, выходящую из узла ***done***. Семантически описание данной операции означает преобразование стратегии решения задач, обозначенной SC-узлом ***taskSolving***, из активной (***active***) в выполненную (***done***).

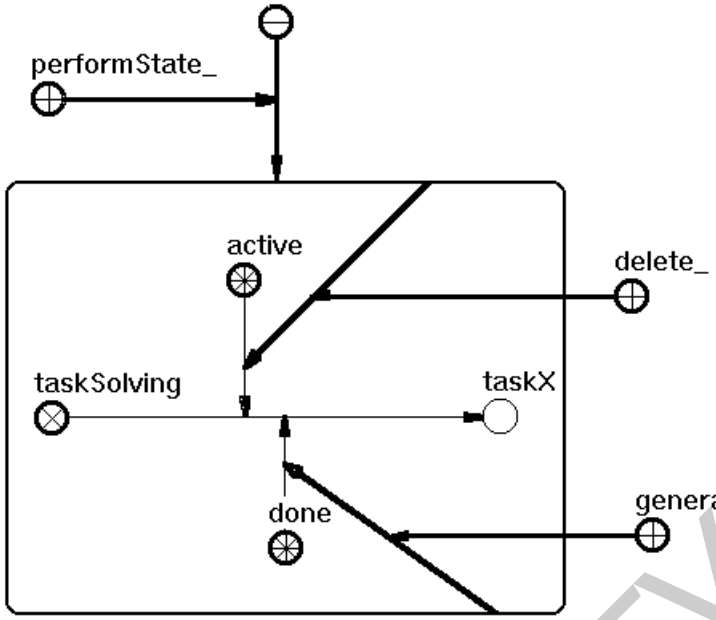

Рис. 5.3. Пример представления на языке SCL состояния преобразования SC-памяти

Согласно описанным выше средствам языка SCL модель управления обучением обозначим узлом ***Tutoring***, который является знаком соответствующей нестационарной SCL-метатеории (рис. 5.4), т.е. обозначает динамическую модель управления обучением.

В [14] перечислены некоторые возможные состояния ИОС и указано, что подсистема управления обучением является центральным блоком в составе ИОС и выполняет роль посредника во взаимодействии подсистем. В связи с этим все состояния разбиваются на несколько групп (основных состояний), соответствующих основным типам задач, решаемых каждой из подсистем (режимам их функционирования). Заметим, что типология этих задач, как правило, коррелирует с применяемыми по отношению к обучаемому стратегиями обучения. Итак, основными состояниями ИОС являются следующие (см. рис. 5.4):

- решение задач и объяснение их решения. Соответсвует режиму решения задач подсистемы решения задач и ознакомления с предметной областью. На рис. 5.4 обозначается SC-узлом с идентификатором **st_tasks** (далее при более подробном рассмотрении ситуаций для каждого из перечисляемых здесь состояний будем использовать идентификаторы, указанные на рис. 5.4);

- ознакомление с понятиями предметной области, по которой ведется обучение. Соответствует режиму ознакомления с предметной областью подсистемы решения задач и ознакомления с предметной областью. Обозначается SC-узлом с идентификатором **st_learn**;

- тестирование обучаемого с целью выявления уровня знаний и умений. Включает все возможные состояния подсистемы тестирования. Обозначается SC-узлом с идентификатором **st_testing**;

- обработка модели обучаемого, выбор стратегий обучения, собственно управление обучением. Обозначается SC-узлом с идентификатором **st_student**;

- анализ психологического и физического состояния пользователя. Описывает совокупности состояний подсистемы анализа психофизического состояния в соответствующих режимах функционирования. Обозначается SC-узлами с идентификаторами **st_psihology** и **st_phisiology** соответственно.

Заметим, что приведенная типология основных состояний ИОС может быть беспрепятственно расширена разработчиком конкретной ИОС. При этом ему не придется создавать каких-либо новых процедур обработки вводимых им конструкций.

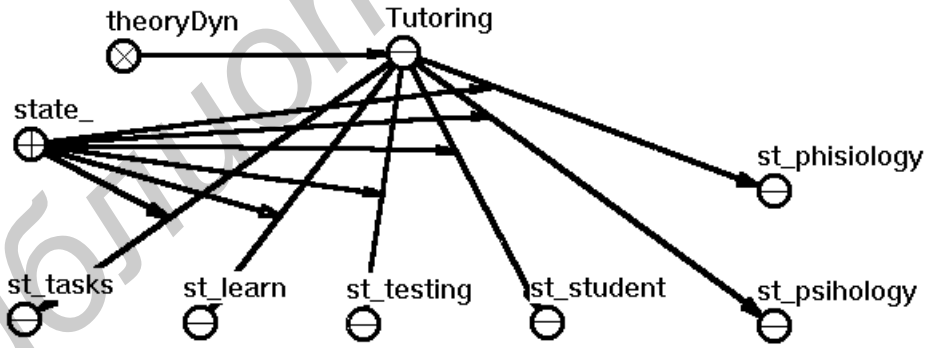

Рис. 5.4. Основные состояния ИОС

Во множество констант представленной на рис. 5.4 SCL-метатеории включаются знаки стратегий обучения и знаки отношений и атрибутов, с помощью которых описываются свойства стратегий, зависимости между ними, состояния стратегий (активная, выполненная) и т.д.: ***strategy***, ***taskSolving***, ***taskExplanation***, ***theoryView***, ***testing***, ***object_***, ***plan_***, ***active***, ***done***,

student, ***startData_***, ***aim_***, ***currentState_***, ***history_***, ***initStudent***.

Далее описание поведения ИОС производится путем добавления в указанные множества состояний описаний переходных состояний. На рис. 5.5, например, приведено правило реализации стратегии решения задач: “Если активизирована стратегия решения задач, то загрузить в SC-память условие и запрос на решение соответствующей задачи и перейти в состояние ожидания завершения реализации указанной стратегии”. Аналогично описывается правило реализации стратегии объяснения. На рис. 5.6 - фрагмент возможной SC-конструкции, хранящейся в содержимом константного SC-узла **task1**, соответствующего переменной **taskX** на рис. 5.5: условие задачи, запрос на решение задачи, правило преобразования активной стратегии в выполненную: “Как только задача будет решена, преобразовать активную стратегию решения задач в выполненную”. Правила преобразования активной стратегии решения задач в выполненную для каждой конкретной задачи описываются аналогичным образом с привязкой к знаку соответствующего запроса на решение задачи. На рис. 5.7 приведено представление на языке SCL одного из правил реализации стратегии объяснения: “Если выбрана стратегия объяснения решения задачи, но эта задача не решена, то сначала решить ее, а затем сразу вывести ее объяснение”.

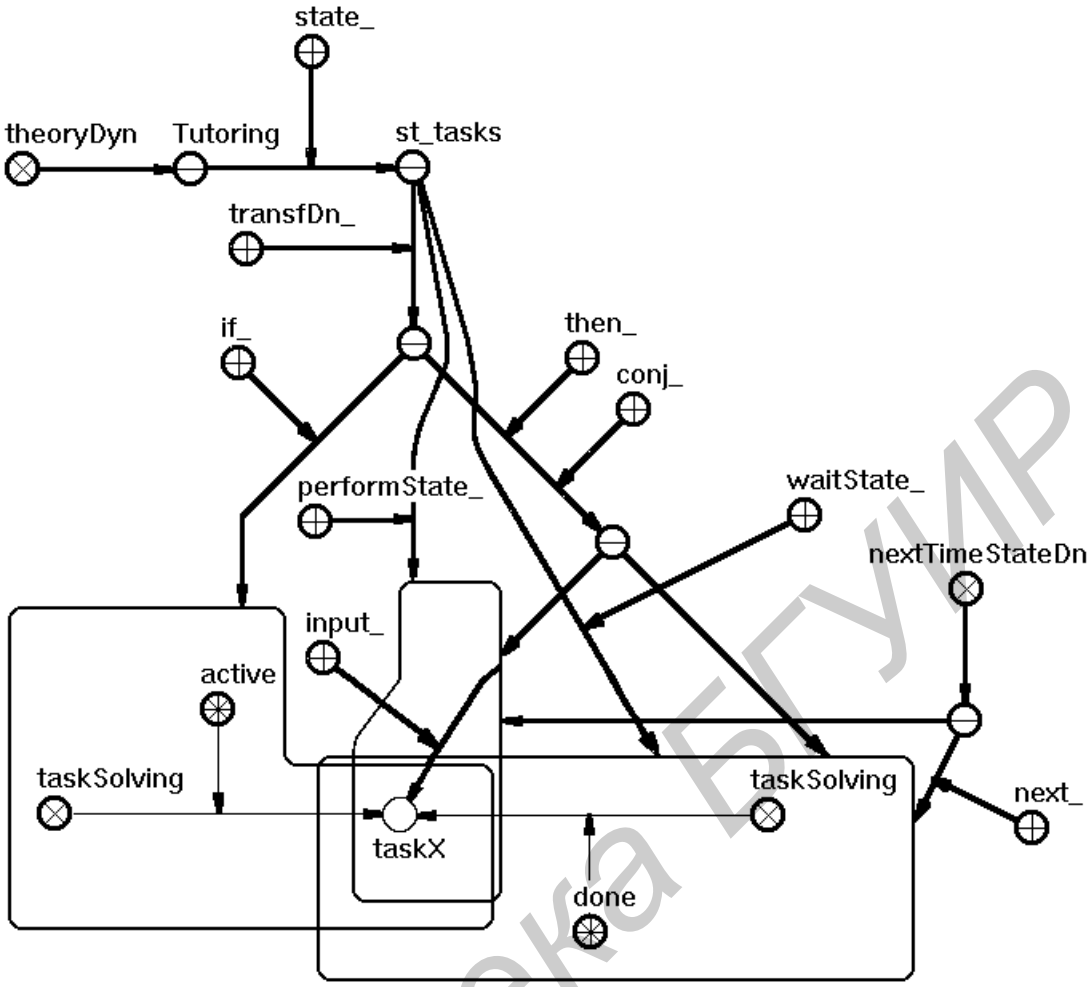

Рис. 5.5. Правило реализации стратегии решения задач

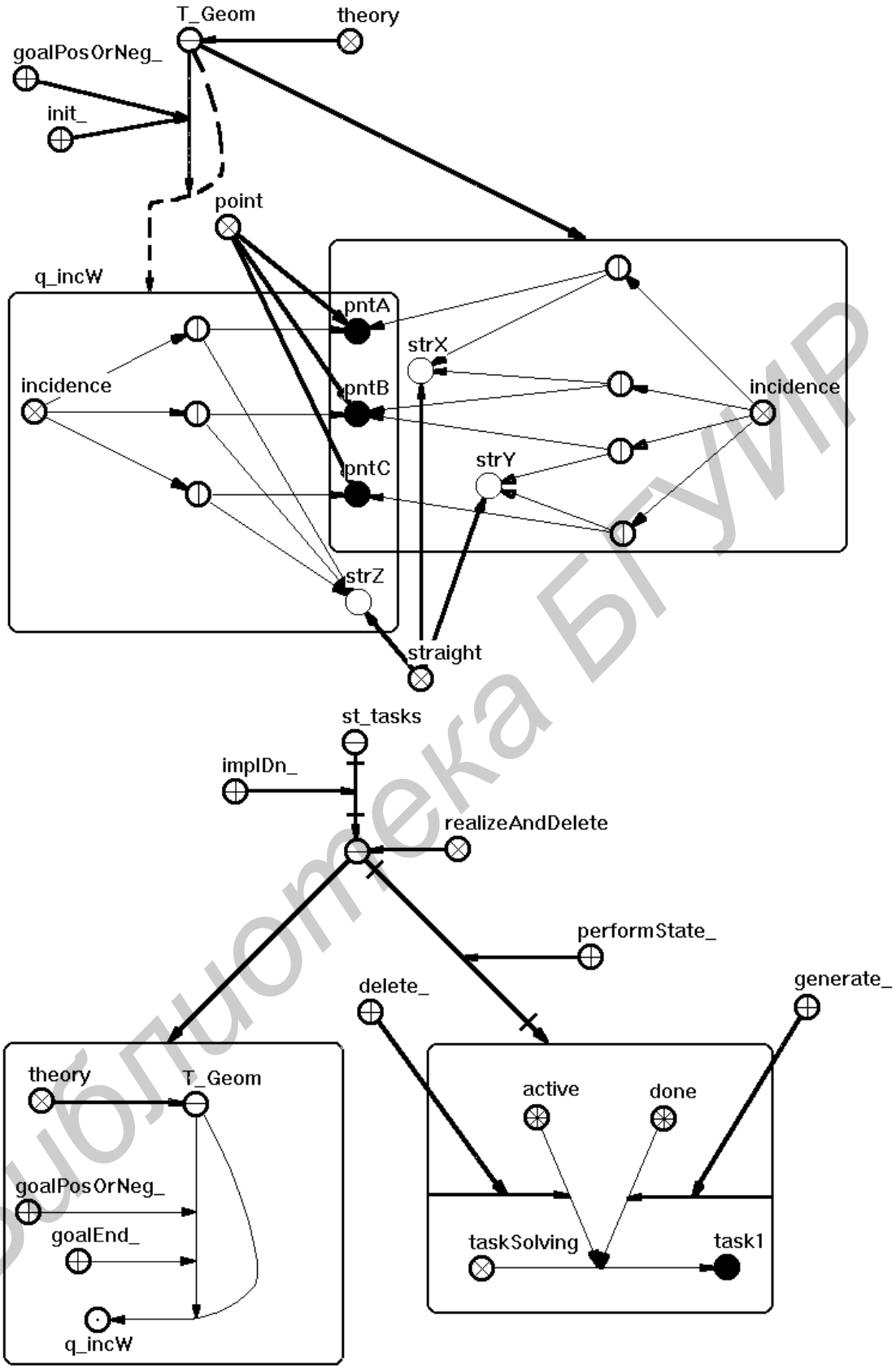

Рис. 5.6. Описание условия задачи, запроса на ее решение и правила преобразования активной стратегии в выполненную

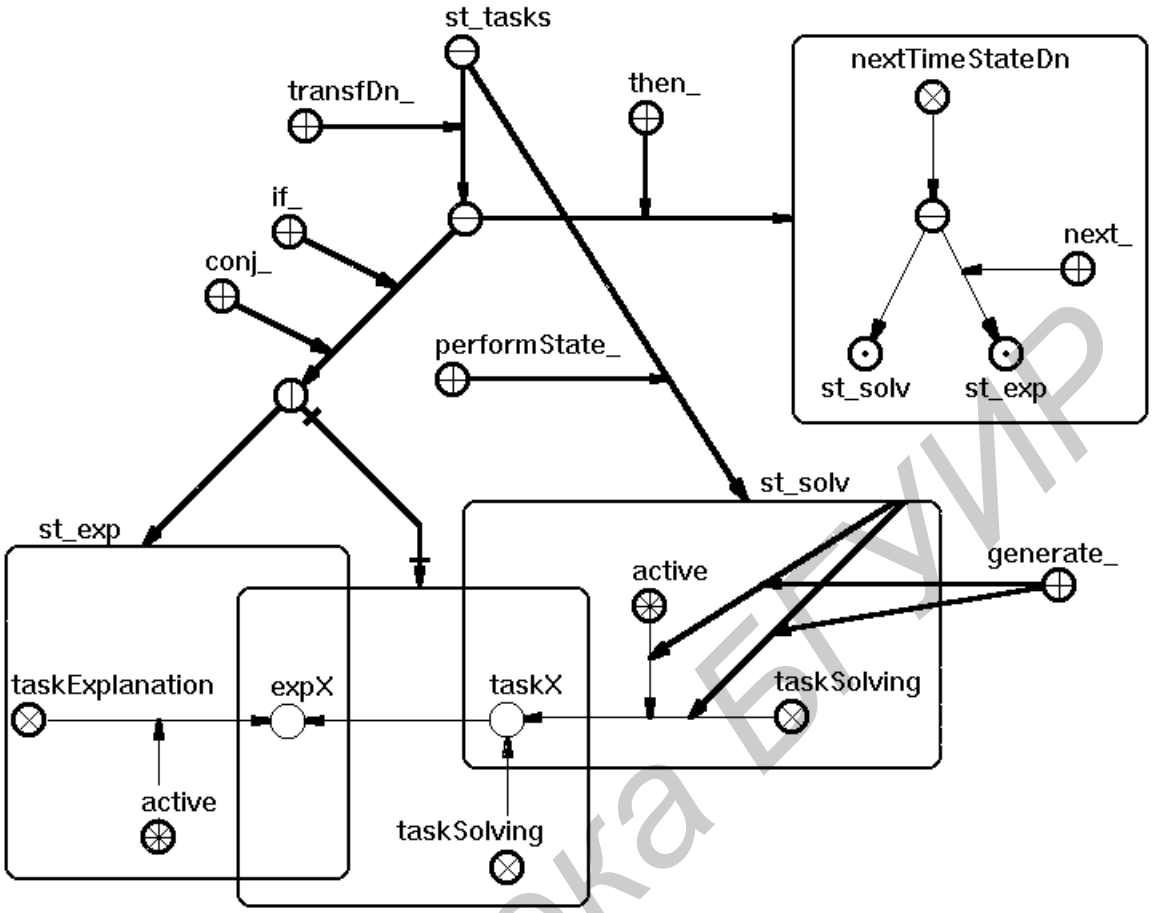

Рис. 5.7. Представление на языке SCL одного из правил реализации стратегии объяснения

Для реализации правил, представляемых с помощью описанных выше средств языка SCL, имеется набор соответствующих SCL-операций поддержки динамического режима функционирования ИС. К этим операциям относятся следующие:

- операции обработки состояний, имеющих некоторое общее свойство;
- операции реализации причинно-следственных связей;
- операции реализации состояний, вызванных некоторым исполнителем;
- операция поддержки состояний ожидания;
- операция реализации состояний преобразования SC-памяти;
- операции поддержки различных соотношений состояний во времени.

Каждая операция из некоторого конкретного указанного выше класса реализует некоторый частный вид соответствующего высказывания.

Ниже коротко описаны спецификации некоторых классов операций.

OPImplDn

Операции обработки состояний, имеющих некоторое общее свойство. Иначе говоря, это операции реализации implDn-высказываний. Данный класс операций аналогичен операциям реализации продукции в прямом и обратном направлении.

OPTransfDn

Операции реализации причинно-следственных связей, т.е. состояний перехода (transfDn-высказываний). С помощью данной операции осуществляется переход из одного фиксированного состояния системы в другое.

OPTransfDnW

Операции реализации состояний, вызванных некоторым исполнителем, т.е. transfDnW-высказываний. Данный класс операций разбивается на три вида. К первому виду относится операция, которая инициируется в случае появления в БЗ некоторого активного исполнителя, ранее описанного в составе transfDnW-высказывания. В процессе реализации данной операции производится поиск в БЗ всех высказываний, содержащих описание активного исполнителя, а затем из них выбирается то, в одном из компонентов которого описано состояние, в котором находится в текущий момент система. Далее производится переход из текущего состояния в следующее согласно найденному transfDnW-высказыванию. Ко второму виду операций данного класса относится операция, которая инициируется в случае наличия в SC-памяти ситуации, описанной в if-компоненте обрабатываемого ею transfDnW-высказывания. В процессе реализации данной операции осуществляется поиск соответствующего активного исполнителя путем формирования запроса. После получения подтверждения от исполнителя осуществляется переход в состояние, описанное в then-компоненте обрабатываемого высказывания. К третьему виду операций реализации состояний, вызванных некоторым исполнителем, относится операция, инициируемая в случае перехода системы в состояние, описанное в then-компоненте обрабатываемого transfDnW-высказывания. Результатом вы-

полнения данной операции является формирование сообщения исполнителю о том, что система перешла в соответствующее состояние.

OPWaitState

Операция поддержки состояний ожидания, т.е. waitState-состояний. Целью данной операции является поиск в SC-памяти SC-конструкций, изоморфных конструкции, описанной в обрабатываемом состоянии ожидания. Данная операция является выполненной успешно в случае успешного поиска указанной конструкции. В обратном случае поиск возобновляется.

OPPerformState

Операции реализации состояний преобразования SC-памяти, т.е. performState-состояний. Данная группа операций на этапе выполнения разбивается на виды, соответствующие тому, какие действия нужно произвести над SC-памятью. Каждый из видов является аналогом SCL-оператора в обобщенном виде. Первая из операций данного класса выполняет генерацию в SC-памяти SC-элементов, помеченных в описании performState-состояния атрибутом ***generate_***. Вторая осуществляет удаление SC-элементов, помеченных атрибутом ***delete_***. Результатом выполнения третьей операции данного класса является загрузка в SC-память содержимого SC-узла, помеченного атрибутом ***input_***.

OPlocalBeginStateDn, ***OPlocalEndStateDn***, ***OPlocalSpaceStateDn***, ***OPeqBeginStateDn***, ***OPeqEndStateDn***, ***OPeqDurationStateDn***, ***OPcomprBeginStateDn***, ***OPcomprEndStateDn***, ***OPcomprDurationStateDn***, ***OPnextTimeStateDn***

Операции поддержки различных соотношений состояний во времени. Данный класс SCL-операций разбивается на виды, каждый из которых осуществляет обработку отношений, задаваемых с помощью следующих ключевых узлов: ***localBeginStateDn***, ***localEndStateDn***, ***localSpaceStateDn***, ***beginStateDn_***, ***endStateDn_***, ***eqBeginStateDn***, ***eqEndStateDn***, ***eqDurationStateDn***, ***comprBeginStateDn***, ***grt_***, ***comprEndStateDn***, ***comprDurationStateDn***, ***nextTimeStateDn***, ***next_***.

6. РЕАЛИЗАЦИЯ УПРАВЛЕНИЯ ВЗАИМОДЕЙСТВИЕМ ИЕРАРХИИ ПОДСИСТЕМ

Управление взаимодействием подсистем в составе ИОС производится согласно выбранной разработчиком и описанной рассмотренными выше средствами модели поведения. Процесс взаимодействия подсистем осуществляется посредством передачи соответствующих запросов между ними. Посредником в этом взаимодействии является подсистема управления обучением (ПсУО). Именно ПсУО осуществляет активизацию и непосредственную передачу запросов между подсистемами. Некоторые из этих запросов, в частности запросы, адресованные подсистеме решения задач и ознакомления с предметной областью (ПсРЗиПОб), формируются средствами языка SCQ. ПсУО, получая любой из указанных запросов, инициализирует его, и таким образом (так как все это происходит в общей для всех подсистем графодинамической памяти), он становится активным в ПсРЗиПОб, где происходит его непосредственная реализация. Помимо данного типа запросов, опишем также другие, адресованные ПсУО, подсистеме тестирования обучаемого и подсистеме диагностики психофизического состояния.

Выбор стратегий и тактик обучения в ПсУО производится по запросу типа ***goalStrategy_***. Элементом запроса является знак активного в текущий момент обучаемого. Обработка данного запроса влечет за собой применение правил выбора стратегий и тактик обучения, описанных в составе SCL-метатеории **st_student**. Аналогичным образом формируется и обрабатывается запрос на обработку модели обучаемого, который задается атрибутом ***goalStudent_***. В зависимости от состояния модели обучаемого в процессе обработки данного запроса могут быть сформированы подзапросы следующего типа:

- инициализации модели обучаемого – с помощью атрибута ***initStudent_***;

- ведения истории взаимодействия обучаемого с ИОС – с помощью атрибута ***historyStudent_***;

- завершения сеанса обучения – с помощью атрибута ***endTutoring_***.

По завершении реализации данного запроса может быть сформирован запрос на удаление модели обучаемого – с помощью атрибута ***delStudent_***. Перед началом нового сеанса обучения в зависимости от выбранной модели поведения возможно обновление модели обучаемого. Это осуществляется в случае наличия активного запроса типа ***refreshStudent_*** . Пример описания запросов указанного типа приведен на рис. 6.1.

В составе подсистемы тестирования знаний обучаемого, которой соответствует состояние **st_testing** SCL-метатеории ***Tutoring***, формируются и обрабатываются запросы:

- на реализацию конкретного теста - с помощью атрибута ***goalTest_***;

- на анализ и обобщение результатов теста - с помощью атрибута ***goalTestAnalysis_***;

- на активизацию режима работы над ошибками - с помощью атрибута ***goalMistake_***.

Аналогичным образом и с использованием тех же атрибутов формируются и обрабатываются запросы подсистемы диагностики психофизического состояния, которой соответствует состояние **st_psihology** SCL-метатеории ***Tutoring***:

- на реализацию конкретного теста - с помощью атрибута ***goalTest_***;

- на обобщение результатов теста - с помощью атрибута ***goalTestAnalysis_***.

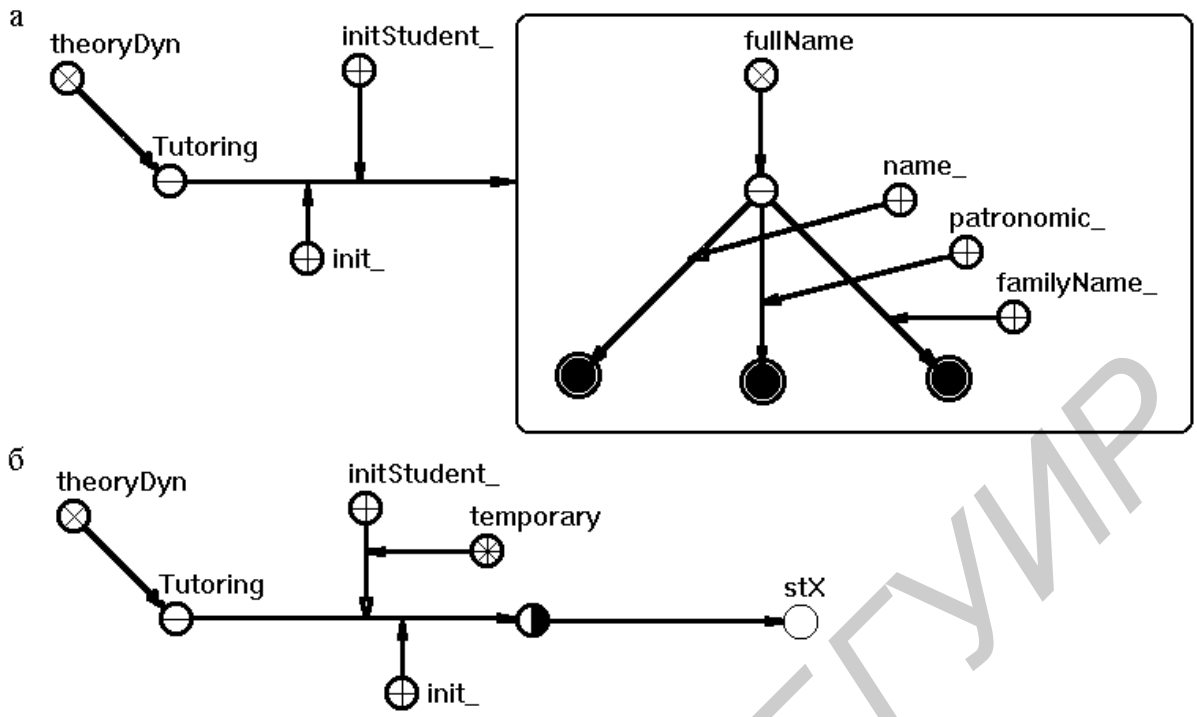

Рис. 6.1. Примеры запросов на инициализацию модели обучаемого:
а - по полному имени; б - временно, т.е. без хранения имени

7. СЕМАНТИЧЕСКОЕ ОПИСАНИЕ СТРУКТУРЫ ТЕСТОВ И ИХ РЕАЛИЗАЦИЯ В ГРАФОДИНАМИЧЕСКОЙ ПАМЯТИ

В БЗ подсистемы тестирования обучаемого ИОС описываются тесты и правила их реализации, а также модель ошибок. В данном разделе описываются средства языка SCT для представления знаний в подсистеме тестирования.

Согласно приведенной в [14] классификации тестов для их описания введены следующие знаки отношений:

externalTest - знак множества внешних тестов. SC-узел, в который входит дуга из данного узла, должен иметь содержимое, в котором задается командная строка для запуска внешнего теста, причем тип содержимого указанного узла задается с помощью ключевого узла ***file_exe*** (см. описание языка SC в [4]). Указанный SC-узел является знаком кортежа отношения

externalTest, количество элементов которого зависит от специфики внешнего теста и, как следствие, способа его описания. Однако один из элементов данного отношения должен обязательно указывать на элемент учебного материала, к которому относится описываемый тест. Этот элемент в составе данного отношения вводится с помощью атрибута ***object_***. Остальные атрибуты вводятся разработчиком конкретной ИОС;

internalTest - знак множества внутренних тестов, каждый из которых, в свою очередь, является знаком множества вопросов (заданий) пользователю, элементы которого упорядочены с помощью атрибутов ***1_***, ***2_***, ... ***n_***. Каждый из элементов указанного множества обозначает конкретный вопрос (задание) и является соответственно элементом отношения ***question*** (см. ниже). В содержимое знака конкретного вопроса (задания) помещается текстовая формулировка соответствующего вопроса (задания), которая будет выводиться пользователю. С помощью атрибута ***object_*** отношения ***internalTest*** указывается принадлежность теста к определенному разделу учебного материала (рис. 7.1);

examinationTest - знак множества экзаменационных тестов;

diagnosticTest - знак множества диагностических тестов.

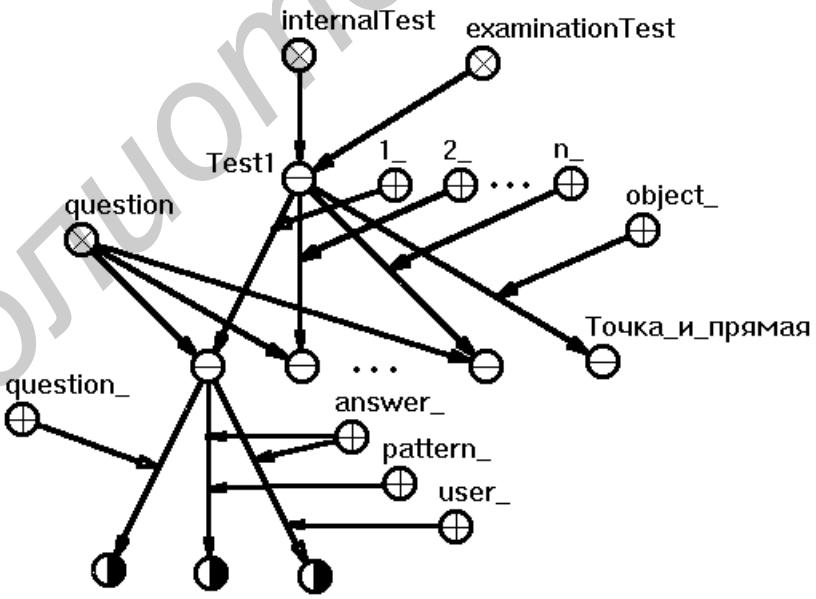

Рис. 7.1. Пример описания внутреннего теста на языке SCT

Ключевые узлы ***examinationTest*** и ***diagnosticTest*** вводятся для того,

чтобы дополнительно задать тип теста и предусмотреть особенности его реализации и обработки. Таким образом, знак каждого теста является элементом одного из множеств ***externalTest*** или ***internalTest*** и одновременно одного из множеств ***examinationTest*** или ***diagnosticTest***.

Заметим также, что описываемый тест может иметь некоторую степень сложности, которая задается с помощью отношения ***complication***. В случае если тест внутренний, для каждого его вопроса (задания) может быть указана степень сложности, и суммарная степень сложности всех вопросов рассматриваемого теста будет задавать степень сложности самого теста. При наличии в БЗ сведений о степени сложности теста оценивание обучаемого производится автоматически с помощью простого арифметического подсчета. Если же степени сложности тестов или вопросов не заданы, то оценивание будет производиться в процентном отношении правильных и неправильных ответов на предлагаемые обучаемому вопросы.

Для описания вопросов и ответов на них вводится отношение ***question*** с атрибутами ***question_***, ***answer_***, ***pattern_***, ***user_*** и ***addition_***. Кортеж указанного отношения обозначает некоторый конкретный вопрос (задание), предлагаемый пользователю. Как сказано выше, в содержимое этого узла вносится текстовая формулировка вопроса (задания). Если задание является достаточно сложным и для автоматического поиска ответа требуется использование SCL-машины, это задание описывается на языке SCL и помещается в содержимое SC-узла, который в составе данного отношения помечается атрибутом ***question_***. На этапе реализации теста происходит загрузка содержимого указанного SC-узла в SC-память и подсистема решения задач получает запрос на решение описанной в вопросе задачи. Таким образом, и пользователь, и система одновременно ищут ответ на вопрос, после чего их ответы будут сверяться. Для этого в составе кортежа отношения ***question*** формируются элементы, помеченные атрибутом ***answer_***, которые дополнительно помечаются также атрибутом ***pattern_*** - для указания эталонного ответа (этот ответ может быть получен системой или заранее введен разработчиком теста) или атрибутом ***user_***, который указывает на знак ответа пользователя. SC-конструкция, соответствующая ответу пользователя, генерируется в SC-памяти на этапе реализации теста, т.е. в процесе диалога с пользователем. По завершении реализа-

ции теста все пользовательские ответы удаляются либо переносятся в историю взаимодействия с системой. Это зависит от описания поведения ИОС в процессе тестирования. С помощью атрибута ***addition_*** указывается дополнительная информация к описываемому вопросу, например набор дополнительных вопросов. Примеры использования отношения ***question*** приведены на рис. 7.1 и 7.2.

После анализа ответа пользователя соответствующими операциями выводится заключение о его правильности. Если ответ на вопрос правильный, то знак этого ответа помещается во множество правильных ответов, обозначаемое SC-узлом ***correctly***, в противном случае - во множество неправильных ответов, задаваемое отношением ***notCorrectly***, с помощью атрибута которого ***mistake_*** указываются ошибки в SC-конструкции, описывающей данный ответ.

Помимо указания ошибок обучаемого, в структуре его ответа в рамках модели ошибок формируется набор типичных (возможных) ошибок с указанием их веса и рекомендуемые действия по их устранению (исправлению). Для представления типичных ошибок на языке SC введено отношение ***typicalMistake***. Знак кортежа этого отношения обозначает некоторую возможную ошибку, а выходящие из него дуги входят в элементы SC-конструкции, описывающей соответствующую ошибочную ситуацию, за исключением дуг, помеченных атрибутами ***object_*** и ***plan_***. Указанные атрибуты выделяют в составе описания ошибки раздел учебного материала, где может быть такая ошибка допущена, и последовательность действий ИОС, нацеленных на исправление допущенной ошибки. Для указания веса ошибки вводится отношение ***mistakeWeight***, которое используется аналогично отношению ***complication***. Заметим также, что SC-конструкции, описывающие ошибочную ситуацию, представляют собой ложные высказывания в составе SCL-теории, описывающей предметную область, по которой ведется обучение.

Набор типичных ошибок обучаемого в зависимости от конкрентной предметной области может быть определенным образом классифицирован. Однако это задача эксперта-предметника и эксперта-педагога. Для введения подобных классификаций нет никаких препятствий, так как предлагаемый и используемый в качестве базового язык представления и переработки знаний

является открытым, т.е. легко расширяемым.

Наличие набора типичных ошибок значительно облегчает обработку допускаемых пользователем в процессе тестирования ошибок. Однако в зависимости от типа теста процесс обработки ошибки различается. Так, в случае реализации экзаменационного теста нет необходимости сразу приступать к исправлению каждой допущенной ошибки; достаточно лишь информации о цене ошибки для выставления результирующей оценки. При необходимости может быть проведена над ошибками после завершения процесса тестирования.

В случае же диагностического теста работа над ошибками является неотъемлемой частью реализации самого теста. Для этого в процессе диагностического тестирования формируется план работы над ошибками. При этом в тесте должен быть предусмотрен набор дополнительных вопросов, вводимый с помощью атрибута ***addition_*** (рис. 7.2).

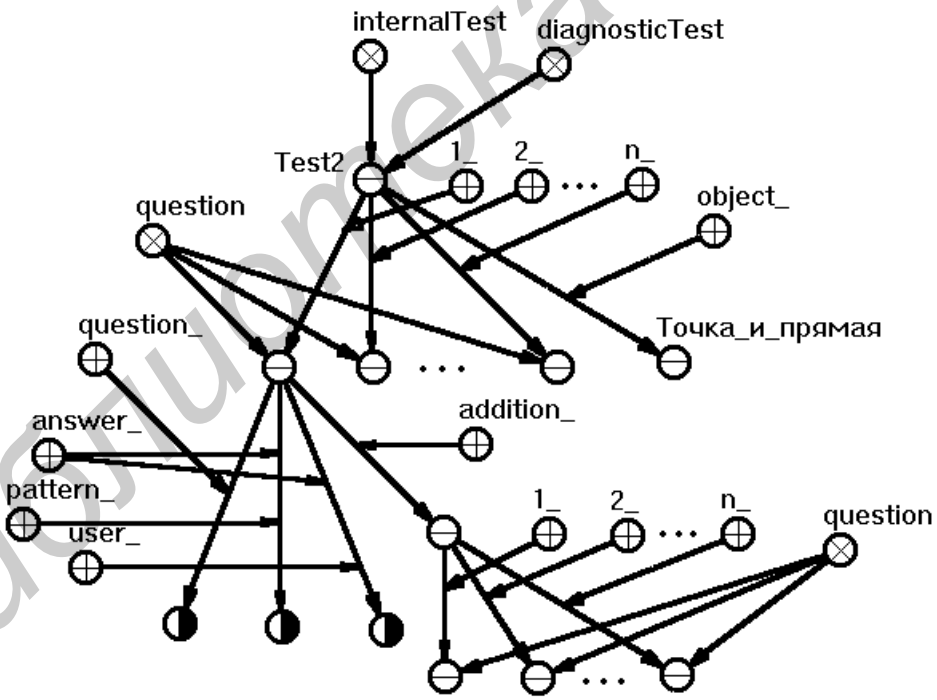

Рис. 7.2. Пример описания диагностического теста на языке SC

Заметим также, что набор тестов в ИОС - это не просто некоторый фиксированный набор вопросов и задач, предлагаемых пользователю. Набор та-

ких задач может отсутствовать, а сами тесты, их структура, отношение к конкретной предметной области и т.п. описываться. Затем с помощью генератора тестовых задач тот или иной тест “наполняется” набором задач. Это позволяет, во-первых, значительно сэкономить объем памяти, необходимый для хранения тестов, а, во-вторых, обеспечить случайность (неповторяемость) задач, что также является крайне важным для любой обучающей системы.

Как было сказано выше, в ИОС все входящие в ее состав подсистемы активизируются из подсистемы управления обучением. Подсистема тестирования не является исключением. Она начинает функционировать после выбора для конкретного обучаемого стратегии тестирования. Описание стратегии тестирования производится с помощью соответствующего отношения ***testing***, атрибуты которого ***test_***, ***mistake_***, ***estimation_*** указывают соответственно на реализуемый в рамках данной стратегии тест, допущенные при этом ошибки и результат (оценку) теста. По результатам теста в процессе обработки модели обучаемого формируются SC-конструкции, отражающие текущий уровень знаний и умений обучаемого.

Процесс функционирования подсистемы тестирования описывается и реализуется средствами языка SCL.

ЗАКЛЮЧЕНИЕ

В заключение сделаем следующие основные выводы:

1) инструментальные средства проектирования ИОС нового поколения, которые разработаны на основе графодинамических параллельных ассоциативных моделей переработки сложноструктурированных знаний, являются одним из перспективных подходов к проектированию систем указанного класса. В рамках указанного подхода знания каждой из подсистем ИОС представляются в виде специальным образом организованных семантических сетей, называемых SC-конструкциями или графовыми конструкциями;

2) графическое представление связей между понятиями позволяет пользователям эффективнее усваивать и запоминать информационную структуру предметной области. Таким образом, организация семантической сети способ-

ствует обучению, поскольку заставляет обучаемого анализировать базовую структуру изучаемых понятий [15]. Наличие средств визуализации (явного отображения на экране компьютера) графовых конструкций позволяет обучаемому изучать семантическую структуру ПОб в динамике;

3) основным отличием и преимуществом рассмотренных средств является то, что они построены на теоретико-множественной основе и ориентированы на семантическое описание и обработку информации в БЗ ИОС. При этом в основу реализации механизмов переработки знаний изначально заложена возможность параллельной обработки информации, что является крайне актуальным для систем рассматриваемого класса. В рамках указанных средств выделим следующие:

- семантический графовый язык представления учебного материала, являющийся расширением базового графового языка SC и позволяющий описывать знания для модели предметной области, которые хранятся и обрабатываются в учебной базе знаний подсистемы решения задач и ознакомления с предметной областью. В соответствии с моделью предметной области указанные языковые средства являются предметно-независимыми и рассчитаны на семантическую интеграцию со средствами мультимедиа. Кроме того, разработанные языковые средства ориентированы на семантическую структуризацию и систематизацию учебной информации. В составе указанного языка разработаны также операции обработки сложноструктурированной информации из УБЗ. Набор указанных операций является открытым и легко модифицируемым, что является необходимым для поддержки гибкости ИОС и удовлетворяет одному из основных требований, предъявляемых к ИОС [16; 2];

- семантический графовый язык описания состояния обучаемого SCT (Semantic Code Tutoring), позволяющий описывать и обрабатывать информацию об обучаемом в рамках модели обучения. Данный язык построен на базе языка SC и является его подъязыком, включает средства описания и переработки информации о пользователях ИОС, а также о семантической структуре тестов, ответов и ошибок пользователя;

- семантический графовый язык заданий и вопросов SCQ (Semantic Code Questions), включающий средства представления и обработки вопросов и заданий, а также средства описания семантической структуры диалога с пользо-

вателем. Особенностью данного языка является то, что как вопросы пользователя, так и запросы, передаваемые из одной подсистемы в другую описываются и обрабатываются единым образом, с использованием одних и тех же средств, что сокращает и упрощает как набор средств описания вопросов и заданий, так и набор механизмов их обработки;

- расширенные средства семантического графового языка SCL для описания и реализации динамических моделей управления обучением для подсистемы управления обучением. Спецификой указанных средств является то, что они позволяют описывать самые различные модели поведения ИОС. Таким образом, разработчику не навязывается какая-то определенная модель, и он имеет возможность реализовать любую, на его взгляд наиболее подходящую модель управления обучением. Аналогично и процесс тестирования обучаемого организуется в соответствии с требованиями эксперта-преподавателя, участвующего в разработке конкретной ИОС.

ЛИТЕРАТУРА

1. Гаазе-Рапопорт М.Г., Поспелов Д.А. От амебы до робота, модели поведения. - М.: Наука, 1987. - 138 с.
2. Гаврилова Т.А., Червинская К.Р. Извлечение и структурирование знаний для экспертных систем. - М.: Радио и связь, 1992. - 200 с.
3. Голенков В.В. Графодинамические методы и средства параллельной асинхронной переработки информации в интеллектуальных системах. - Мн., БГУИР, 1996. - 295 с.
4. Голенков В.В. Графодинамические модели и методы параллельной асинхронной переработки информации в интеллектуальных системах: Диссертация д-ра техн. наук: 05.13.17; 05.13.11. - Мн., 1996. - 341 с.
5. Голенков В.В., Гулякина Н.А. Модели представления и переработки знаний: Мет. указания. - Мн.: БГУИР, 1998. - 34 с.
6. Голенков В.В., Гулякина Н.А. Язык параллельного программирования, ориентированный на переработку сложноструктурированных знаний в структурно перестраиваемой ассоциативной памяти. - Мн.: БГУИР, 1998. - 83 с.
7. Голенков В.В., Гулякина Н.А., Елисеева О.Е. Описание языка SCP. - Мн., 1995. - 152 с. - (Материалы по математическому обеспечению ЭВМ / Ин-т техн. Кибернетики АН Беларуси).
8. Голенков В.В., Королев В.Г., Елисеева О.Е. Операции языка SCL для обработки простых запросов. - Мн., 1995. - 140 с. - (Материалы по математическому обеспечению ЭВМ / Ин-т техн. кибернетики АН Беларуси).
9. Голенков В.В., Королев В.Г., Малевич И.Е. Представление знаний различного вида на языке SCL - Semantic Code Logic. - Мн., 1995. - 58 с. - (Препринт / Ин-т техн. кибернетики АН Беларуси; № 4).
10. Голенков В.В., Королев В.Г., Малевич И.Е. Представление логических высказываний на языке SCL - Semantic Code Logic. - Мн., 1994. - 46 с. - (Препринт / Ин-т техн. кибернетики АН Беларуси; № 19).
11. Голенков В.В. Описание графового языка SC. - Мн., 1994. - 86 с. - (Материалы по математическому обеспечению ЭВМ / Ин-т техн. Кибернетики

АН Беларуси).

12. Голенков В.В. Параллельный графовый компьютер (PGC), ориентированный на решение задач искусственного интеллекта, и его применение. - Мн., 1994. - 60с. - (Препринт / Ин-т техн. кибернетики АН Беларуси; № 2).
13. Голенков В.В., Королев В.Г., Елисеева О.Е. Решение на языке SCL задач из области геометрии. - Мн., 1995. - 135 с. - (Материалы по математическому обеспечению ЭВМ / Ин-т техн. кибернетики АН Беларуси).
14. В.В.Голенков, Н.А.Гулякина, О.Е.Елисеева. Инструментальные средства проектирования интеллектуальных обучающих систем: Метод. пособие по курсу "Интеллектуальные обучающие и тренажерные системы" для студентов специальности "Искусственный интеллект". -Мн.: БГУИР, 1999. - 102 с.
15. Джонассен Д.Х. Компьютеры как инструменты познания: изучение с помощью технологии, а не из технологии // Информатика и образование. - 1996. - № 4. - С. 117 - 131.
16. Дякин М.В., Подорожный Д.А., Сапир М.В. Подход к созданию гибких обучающих систем // Известия Академии Наук. Теория и системы управления. - 1996. - № 5. - С. 77-84.
17. Елисеева О.Е. Инструментальные средства проектирования интеллектуальных обучающих систем // Вышэйшая школа. - 1998. - № 1. - С.63-69.
18. Поспелов Д.А. Ситуационное управление: теория и практика. - М.: Наука. - Гл. ред. физ.-мат.лит., 1986. - 288 с.
19. Представление теоретико-множественых и арифметических отношений на языке SCL - Semantic Code Logic / В.В.Голенков, В.Г.Королев, О.Е.Елисеева, И.Е.Малевич - Мн., 1995. - 20с. - (Препринт / Ин-т техн. кибернетики АН Беларуси; № 3).

Св. план 1999, поз. 14 (вед.)

Учебное издание

Голенков Владимир Васильевич
Гулякина Наталья Анатольевна
Елисеева Ольга Евгеньевна

ГРАФОДИНАМИЧЕСКИЕ МОДЕЛИ И ЯЗЫКИ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ ОБУЧАЮЩИХ СИСТЕМАХ

УЧЕБНОЕ ПОСОБИЕ
по курсу "Интеллектуальные тренажерные и обучающие системы" для студентов специальности “Искусственный интеллект”

Редактор Т.Н.Крюкова, Н.Б.Гриневич
Корректор Е.Н.Батурчик

Подписано в печать Формат 60 × 84 1/16.
Бумага Печать офсетная.
Усл.печ.л. Уч.-изд.л. 4,5 Тираж 150 экз.
Заказ

Белорусский государственный университет информатики и радиоэлектроники
Отпечатано в БГУИР. Лицензия ЛП № 156. 220027, Минск, П.Бровки, 6