

Московский Государственный институт электроники и математики
(Технический университет)

На правах рукописи

КАРПОВА ИРИНА ПЕТРОВНА

ИССЛЕДОВАНИЕ И РАЗРАБОТКА ПОДСИСТЕМЫ КОНТРОЛЯ ЗНАНИЙ
В РАСПРЕДЕЛЕННЫХ АВТОМАТИЗИРОВАННЫХ ОБУЧАЮЩИХ
СИСТЕМАХ

05.13.13 – "Телекоммуникационные системы и компьютерные сети"

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель
доктор технических наук,
профессор САКСОНОВ Е.А.

Москва 2002

Содержание

Введение	5
1. АВТОМАТИЗИРОВАННЫЕ ОБУЧАЮЩИЕ СИСТЕМЫ	9
1.1. История развития компьютерных средств учебного назначения.....	12
1.2. Распределенная автоматизированная обучающая система.....	18
1.2.1. Применение сетевых технологий в обучающих системах.....	19
1.2.2. Телекоммуникационная среда РАОС.....	21
1.2.3. Компьютерные сети РАОС	22
1.3. Классификация компьютерных средств учебного назначения	24
1.3.1. Существующие классификации.....	24
1.3.2. Виды классификации	28
1.3.3. Принцип многомерной классификации	29
1.4. Схема процесса обучения	31
1.5. Автоматизированные обучающие системы (АОС).....	34
1.5.1. Преимущества автоматизированных обучающих систем.....	34
1.5.2. Понятие автоматизированной обучающей системы.....	35
1.5.3. Требования, предъявляемые к РАОС.....	36
1.5.4. Инструментальные оболочки обучающих систем.....	39
1.5.5. Основные направления исследований	44
Выводы	49
2. ПРИНЦИПЫ СОЗДАНИЯ РАСПРЕДЕЛЕННЫХ АВТОМАТИЗИРОВАННЫХ ОБУЧАЮЩИХ СИСТЕМ.....	51
2.1. Организация РАОС	51
2.1.1. Создание сетевой среды РАОС.....	51
2.1.2. Особенности функционирования программного обеспечения РАОС...52	52
2.1.3. Описание протокола обмена данными между ядром АОС и вспомогательным программным обеспечением	56
2.2. Основные характеристики РАОС как программного продукта	57
2.3. Пользователи АОС.....	58
2.4. Логическая структура обучающей системы	61
2.4.1. Взаимодействие с обучаемым.....	62
2.4.2. Подсистема управления обучением	63
2.4.3. Модель предметной области (МПО).....	65
2.4.3.1. Требования, предъявляемые к МПО	66
2.4.3.2. Семиотическая сеть.....	68
2.4.3.3. Анализ модели предметной области	70
2.4.3.4. Организация модели предметной области (МПО)	71
2.4.4. Модель обучаемого	72
2.4.4.1. Методы построения модели обучаемого	75

2.4.4.2. Общие принципы построения модели обучаемого	76
2.4.4.3. Преобразование модели обучаемого.....	77
2.4.5. Дополнительные возможности	80
2.5. Контроль знаний.....	82
2.5.1. Схемы проведения контроля знаний.....	84
2.5.2. Параметры проведения контроля знаний	87
2.5.2.1. Встроенные параметры.....	87
2.5.2.2. Параметры, задаваемые пользователем	93
2.5.2.3. Общие замечания о параметризации.....	94
2.5.3. Способы формирования актуального множества вопросов	94
Выводы	96
3. ПОДСИСТЕМА КОНТРОЛЯ ЗНАНИЙ	98
3.1. Особенности работы подсистемы контроля знаний в РАОС	98
3.2. Характеристики систем контроля знаний.....	99
3.2.1. Анализ существующих систем контроля знаний.....	99
3.2.2. Требования к подсистеме контроля знаний.....	101
3.3. Анализ ответов обучаемого.....	102
3.3.1. Представление вопросов и ответов	102
3.3.2. Типы вопросов и ответов.....	104
3.3.3. Методы определения правильности выборочных ответов	105
3.3.3.1. Сравнение множеств элементов	106
3.3.3.2. Сравнение списков элементов	107
3.3.3.3. Множества списков.....	115
3.3.3.4. Списки множеств	115
3.3.4. Анализ ответа в виде арифметического выражения.....	116
3.3.5. Текстовые ответы	117
3.3.6. Графические формы задания ответов	119
3.3.6.1. Распознавание образов в обучающих системах.....	119
3.3.6.2. Задача распознавания графиков	122
3.4. Расчет вероятности случайного ввода правильного ответа.....	124
3.4.1. Ответ типа МНОЖЕСТВО	125
3.4.2. Ответ типа СПИСОК	126
3.4.3. Ответ типа СПИСОК МНОЖЕСТВ	127
3.4.4. Ответ типа МНОЖЕСТВО СПИСКОВ.....	129
3.5. Моделирование случайного ввода правильного ответа	131
3.5.1. Моделирование ответа типа МНОЖЕСТВО.....	131
3.5.2. Моделирование ответа типа СПИСОК	134
3.5.3. Моделирование ответа типа СПИСОК МНОЖЕСТВ	137
3.5.4. Моделирование ответа типа МНОЖЕСТВО СПИСКОВ	141
Выводы	143

4. РЕАЛИЗАЦИЯ РАСПРЕДЕЛЕННОЙ АВТОМАТИЗИРОВАННОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ	145
4.1. Описание программного комплекса	145
4.1.1. Функциональная структура программного комплекса	145
4.1.2. Структура распределенного программного комплекса.....	147
4.1.3. Протокол обмена данными с динамической реконфигурацией системы	150
4.1.4. Настройка АОС	156
4.2. Подсистема контроля знаний	157
4.2.1. Особенности системы контроля знаний	157
4.2.2. Язык описания тестов	159
4.2.2.1. Требования, предъявляемые к языку описания тестов	159
4.2.2.2. Формат языка описания тестов.....	159
4.2.3. Контроль знаний в режиме удаленного доступа.....	166
4.2.4. Управление тестированием	167
4.3. Эксперименты по применению подсистемы контроля знаний	167
Выводы	170
Заключение.....	173
СПИСОК ЛИТЕРАТУРЫ.....	175
Приложение 1. Наборы правил базы знаний для управления контролем знаний	187
Приложение 2. Вариант программной реализации КС-грамматики для разбора описания теста	195
Приложение 3. Параметрическая настройка системы контроля знаний	199
Приложение 4. Перечень сокращений, встречающихся в тексте	200

Введение

Программные средства учебного назначения по принципам использования можно условно разделить на обучающие системы, наполненные знаниями о конкретной предметной области, и инструментальные системы, предназначенные для наполнения их знаниями о произвольной предметной области с целью создания обучающей системы [103]. Наиболее перспективными с точки зрения соотношения конечного результата и трудозатрат на создание и поддержку являются инструментальные системы, которые принято называть автоматизированными обучающими системами (АОС) [17, 50, 55, 99, 134].

К основным достоинствам АОС относятся:

- возможность использования преимуществ индивидуального обучения [127];
- интенсификация обучения [8, 114];
- возможность индивидуальной адаптации курса обучения к потребностям обучаемых или условиям обучения [91];
- возможность использования и тиражирования передового опыта [90];
- повышение доступности образования [56, 76, 140];
- обучение навыкам самостоятельной работы [67, 121];
- разгрузка преподавателя от ряда рутинных, повторяющихся действий (чтение лекций, проверки контрольных работ и т.д.) [90];
- возможность использования в рамках дистанционного обучения, переобучения и повышения квалификации [3, 18, 44].

В настоящее время благодаря развитию вычислительных сетей и телекоммуникационных технологий АОС получили возможность выйти на новый уровень. При переходе от локальных обучающих систем к распределенным качественно изменяются функциональные возможности обучающей системы. Организация распределенных АОС (РАОС) требует проработки сетевых аспектов работы системы, связанных с предоставлением

удаленного доступа к системе, поддержкой распределенных данных и объединением сетевых ресурсов для решения стоящих перед системой задач.

Одной из важных задач при создании РАОС является организация контроля знаний. Большинство существующих АОС и систем контроля знаний имеют ограниченное количество форм представления ответов и двухбалльную систему оценки. Это обусловлено простотой анализа выборочных ответов и отсутствием формальных методов анализа и дифференцированной оценки ответов обучаемых на контрольные вопросы. Однако такой подход ограничивает возможности разработчика курса в отношении использования различных вариантов тестовых вопросов и анализа ответов обучаемых. В связи с этим тематика исследований, затрагивающих организацию контроля знаний в РАОС, является актуальной.

Цель работы. Целью диссертационной работы является исследование методов организации распределенных автоматизированных обучающих систем и разработка общих принципов построения систем контроля знаний на основе модели дифференцированной оценки ответов обучаемых.

Задачи исследования. В диссертационной работе решаются следующие задачи:

- анализ существующих автоматизированных обучающих систем и тенденций их развития;
- выявление требований, которые предъявляются к РАОС как к специализированному программному обеспечению, ориентированному на работу в компьютерной сети;
- разработка методов анализа и дифференцированной оценки ответов обучаемых;
- разработка математического, алгоритмического и программного обеспечения компонентов РАОС;
- экспериментальное подтверждение применимости предложенных методов.

Методы исследования основаны на использовании положений теории множеств, теории вероятности, комбинаторики и методов инженерии знаний. В разработке программного обеспечения использовалась технология объектно-ориентированного программирования.

Научная новизна. Предложен новый подход к организации контроля знаний в распределенных автоматизированных обучающих системах.

К новым результатам относятся:

- разработка протокола взаимодействия компонентов РАОС, которые позволяют выполнять распределенные вычисления и интегрировать в систему существующие программные средства учебного назначения;
- создание методов сравнения множеств и списков для определения правильности ответов;
- разработка моделей представления ответов различных типов, позволяющих использовать для их анализа и оценки методы сравнения списков и множеств.

Практическая ценность. Теоретические исследования завершены созданием на их основе математического, алгоритмического и программного обеспечения задачи создания подсистемы контроля знаний в РАОС. А именно:

- разработан протокол обмена данными между ядром РАОС и вспомогательным программным обеспечением;
- созданы базы знаний, реализующие различные методики управления контролем знаний обучаемого;
- разработаны алгоритмы проведения контроля знаний, методы и алгоритмы определения правильности различных типов ответов обучаемого на контрольные вопросы;
- создан и используется в учебном процессе комплекс программ, реализующий разработанные методы и алгоритмы.

Положения, выносимые на защиту. На защиту выносятся:

- методы анализа и дифференцированной оценки ответов обучаемого на контрольные вопросы;
- модели представления различных типов ответов обучаемого на основе списков и множеств;
- архитектура открытой РАОС и протоколы взаимодействия ее компонентов.

Апробация работы. Основные результаты диссертационной работы докладывались и обсуждались на 8-й Межвузовской научно-методической конференции "Информационные технологии и фундаментализация высшего образования" (РГУ нефти и газа им. Губкина, 21-22 февраля 2002 г.) и на научно-технической конференции, посвященной 40-летию МГИЭМ (19-28 февраля 2002 г.).

Публикации. По результатам выполненных исследований опубликовано 5 печатных работ, отражающих основные результаты работы.

Объем и структура диссертации. Диссертация состоит из введения, четырех глав, заключения, списка литературы (145 наименований) и приложений. Основное содержание диссертационной работы изложено на 204 страницах машинописного текста, иллюстрированного таблицами и рисунками.

1. АВТОМАТИЗИРОВАННЫЕ ОБУЧАЮЩИЕ СИСТЕМЫ

Под обучением понимается процесс передачи и усвоения знаний, умений и навыков деятельности [14]. В процессе обучения, вообще говоря, реализуются цели образования и воспитания. В дальнейшем мы будем иметь в виду только образовательные цели.

Традиционной формой получения образования является обучение с преподавателем. Но в процессе обучения издавна применялись различные вспомогательные средства. Например, в Китае еще несколько тысячелетий назад использовался тренажер для обучения искусству иглоукалывания – муляж человека с множеством скрытых отверстий, в которые безошибочно должен был попасть иглой обучаемый. Позже появились другие тренажеры, потом – механические и электронные системы тестирования. Далее, с появлением вычислительной техники, начал развиваться такой вид программного обеспечения, как *системы машинного обучения* или *автоматизированные обучающие системы* (АОС) [5].

Основу образовательного процесса при использовании обучающих систем составляет целенаправленная и контролируемая интенсивная самостоятельная работа обучаемого, который может учиться в удобное для себя время, по индивидуальному расписанию, имея при себе комплект специальных средств обучения и возможность контакта с преподавателем с помощью современных технических средств или очно [6].

Конечно, эффективность традиционного вида обучения с преподавателем в форме лекционных, практических и лабораторных занятий доказана всей историей развития человечества. Но с другой стороны, о сложностях, возникших в системе высшего образования в связи с техническим и информационным прогрессом, говорят уже давно, и не без основания [86–87].

Одна из основных проблем высшего образования – несоответствие между возможностями традиционных методов обучения и тем объемом

фактических знаний, которое современное общество требует от выпускников учебных заведений. Увеличение сроков обучения как средство решения его возросших задач исчерпано, поэтому необходимо полагаться на внутренние резервы учебного процесса. Речь идет, прежде всего, об интенсификации и оптимизации учебного процесса [9].

При современном уровне развития средств телекоммуникаций имеются возможности предоставления обучаемым доступа к образовательным ресурсам из-за пределов учебного заведения [64]. Это касается не только студентов, получающих образование по вечерней и заочной формам. (Хотя число людей, желающих получить высшее образование, но не имеющих возможности посещать традиционные дневные лекции из-за занятости на производстве, удаленности от вуза или физической инвалидности, непрерывно увеличивается [123]). Кроме этого, по оценкам специалистов, знания в технической сфере устаревают примерно за 5 лет [104], поэтому требуется постоянное обновление профессионального багажа специалиста.

Остроту стоящих перед высшим образованием проблем можно снять, применяя в учебном процессе компьютерные средства учебного назначения [19, 124].

Компьютерные средства учебного назначения (КСУН) [92] – это программные продукты, используемые в преподавании, обучении, самообразовании и повышении профессионального уровня специалиста.

Примечание: автор не является приверженцем употребления аббревиатур, поэтому в дальнейшем вместо аббревиатуры "КСУН" будет использоваться термин "обучающая система", под которым подразумевается любое компьютерное средство учебного назначения.

Появление обучающих систем поставило на повестку дня вопрос об автоматизации обучения. Задача автоматизации – повышение эффективности обучения, которое складывается из нескольких составляющих:

- более прочное усвоение материала;
- больший объем знаний;
- меньшее время на их усвоение.

Повышение эффективности обучения при использовании обучающих систем было подтверждено рядом исследований [9, 39, 111, 120]. Этому способствуют такие факторы, как:

1. Индивидуализация обучения [127].

Наиболее эффективно, но и наименее экономно индивидуальное обучение (один преподаватель и один ученик). Самая экономичная, но и наименее эффективная система – массовое обучение [104]. Внедрение обучающих систем позволит совместить достоинства индивидуального обучения (в смысле эффективности) и массового (в смысле экономичности).

2. Интенсификация обучения [8, 114].

Она достигается за счет индивидуальности обучения (толпа всегда идет медленнее одного человека), а также за счет того, что обучаемый не привязан ко времени занятия и к преподавателю, а может заниматься в удобное для себя время.

3. Использование выразительных средств вычислительной техники (ВТ), таких как наглядность, наличие средств моделирования объектов и процессов и т.п. [21, 102].

4. Возможность организации постоянного контроля степени усвоения знаний, способствующего более прочному закреплению материала [12].

Кроме повышения эффективности обучения внедрение обучающих систем имеет и другие положительные эффекты:

- Работа с обучающей системой развивает умение и навыки самостоятельной работы [67, 121].
- Обучающие системы разгружают преподавателя от ряда трудоемких и часто повторяющихся операций по представлению учебной информации и контролю знаний; способствуют разработке объективных методов контроля знаний; облегчают накопление передового учебно-методического опыта [90].
- Применение обучающих систем может упростить переход вузов к обучению по более широкому перечню специализаций, благодаря

которому каждый студент получает возможность получить подготовку с индивидуальным профессиональным и образовательным уклоном.

- Возможно применение обучающих систем в системе дополнительного профессионального образования [3, 18, 44], особенно в тех областях деятельности, в которых имеет место низкая эффективность традиционных способов передачи знаний посредством лекционных занятий [27].
- Применение обучающих систем позволяет предоставить образовательные услуги более широкому кругу обучаемых, в т.ч. в рамках дистанционного обучения [56, 76, 140].

Для того чтобы точнее определить место и роль обучающих систем в учебном процессе, проследим за историей развития этих систем.

1.1. История развития компьютерных средств учебного назначения

Возможность автоматизации любого вида деятельности появляется в том случае, когда выполняемые человеком функции могут быть в достаточной степени формализуемы и адекватно воспроизведены с помощью технических средств, при условии выполнения требований по качеству достигаемого результата. Для процесса передачи знаний эта возможность появилась вместе с появлением вычислительной техники – в середине прошлого века.

Первые эксперименты по применению компьютеров в образовании относятся к концу 50-х годов. Несмотря на то, что техническая база ЭВМ и программное обеспечение того времени явно не соответствовали успешному решению поставленной проблемы в целом, исследования в этой области начались во всех развитых странах. Выделим наиболее значимые этапы развития работ в этой области и проследим за изменением целей и задач, которые ставили перед собой исследователи и разработчики.

Первый этап исследования возможностей создания обучающих систем приходится на 50-е и 60-е годы двадцатого столетия. Профессор Б.Ф. Скиннер в 1954 году выдвинул идею, получившую название *программированного обучения* [142]. Она заключалась в призыве повысить эффективность управления учебным процессом путем построения его в полном соответствии с психологическими знаниями о нем, что фактически означает внедрение кибернетики в практику обучения [107]. Это направление начало активно развиваться в США, а потом и в других странах. И уже тогда одним из основных признаков программированного обучения считалась автоматизация процесса обучения [95].

Автоматизация программированного обучения началась с использования обучающих и контролирующих устройств различного типа. Они достаточно широко применялись в 60–70-е годы [43, 84, 88], хотя из-за ограниченных возможностей не обеспечивали достаточной эффективности и адекватности результатов контроля реальному уровню знаний обучаемого. Фактически применение таких устройств как в нашей стране [13, 49], так и за рубежом [130, 144] не вышло за рамки обучения разным навыкам, а также простейших методов контроля, в основном выборочного типа.

В это же время начали развиваться идеи искусственного интеллекта. Были разработаны основные модели представления знаний, появились первые системы, использующие методы искусственного интеллекта. Это было время эйфории. Казалось, еще немного, каких-то 10-20 лет, и будет создан искусственный разум, которому можно будет перепоручить многие обязанности человека, по крайней мере, те из них, которые не требуют творческого подхода.

Благодаря этой атмосфере всеобщей воодушевленности стоящие перед разработчиками обучающих систем цели были сформулированы следующим образом. Разработать такую обучающую систему, которая могла бы полностью имитировать преподавателя, т.е. обладала бы достаточным набором знаний не только в предметной области, но и в педагогике, и могла

бы в рамках предметной области общаться с обучаемым на естественном языке. Например, У. Аттель в статье [7] пишет: "Энтузиазм, вызванный возможностью применения вычислительной машины для обучения, связан с надеждой на то, что способность этих устройств перерабатывать естественный язык позволит, в конечном счете, ...моделировать естественное общение учителя и ученика".

Это была задача-максимум, но она определила цель, к которой следовало стремиться. В результате проводимых исследований была разработана структура обучающих систем и предложены некоторые методы решения этой проблемы [137, 143]. Но, как и в области исследований по искусственному интеллекту, реализация общих идей столкнулась с огромными практическими трудностями. В процессе создания первых прототипов АОС стало ясно, насколько сложными являются задачи представления предметных знаний, организации обратной связи с обучаемым (в том числе, полноценного диалога, для которого явно не хватало лингвистических знаний). Поэтому созданные в то время системы очень сильно отличались от идеала.

Тем не менее, в 60-е годы было разработано большое количество специализированных пакетов программ, ориентированных на создание и сопровождение прикладных обучающих программ – автоматизированных учебных курсов (АУК) на базе ЭВМ третьего поколения. Одними из самых известных в нашей стране проектов использования вычислительной техники и средств коммуникации в обучении является проект PLATO в наиболее развитой версии – PLATO-IV, а также отечественные автоматизированные обучающие системы АОС-ВУЗ, АОС-СПОК, АСТРА, САДКО и другие.

По сути дела эти и многие другие обучающие системы были системами селективного (выбирающего) типа. В таких системах определение методики обучения в целом и содержание обучающих воздействий в частности оставлялось педагогу, а их реализация и оценка результатов производилась средствами АОС. Связующим звеном между системой и педагогом была

специальная форма представления информации – обучающий курс, – в который человеком "закладывались" все обучающие воздействия и условия смены их последовательности по линейной или ветвящейся программе.

Кроме систем селективного типа были созданы продуцирующие обучающие системы, в которых диалог с обучаемым не программируется, а формируется по нескольким алгоритмам в соответствии с набором операций и фактов, заложенных в систему. Подобные обучающие системы предназначались для некоторых специфических предметных областей, которые по тем или иным причинам оказались исключительно подходящими для такого типа программирования. В качестве примеров можно привести систему Ликлайдера для обучения аналитической геометрии [137] и систему Битена и Лэйна, обучающую произношению слов иностранного языка [129].

Следующий этап в развитии автоматизированного обучения – с начала 70-х до середины 80-х. К этому времени идея создания интеллектуальных систем фактически потерпела временное фиаско, что нашло свое отражение в деградации понятия автоматизированного обучения. Автоматизированными обучающими системами начали называть любые программы, предназначенные для информационной или функциональной поддержки процесса обучения: тесты, электронные учебники, лабораторные практикумы и т.п., что нашло свое отражение в классификациях АОС, относящихся к тому времени (раздел 1.3.1).

Впрочем, несмотря на ослабление требований к обучающим системам, продолжались исследования возможности использования при создании АОС идей и методов представления знаний, разработанных к тому времени в области искусственного интеллекта. Но если для представления знаний о предметных областях эти разработки подходили в значительной степени, то для решения двух других задач – управление обучением и контроль знаний – требовались более сложные методы и средства. Именно эти проблемы находились в поле зрения разработчиков обучающих систем в конце данного

периода и все еще являются предметом современных исследований в области обучающих программ.

В начале этого периода основные усилия теоретиков автоматизированного обучения были направлены на поиск и проверку более глубоких моделей обучения на основе когнитивной психологии. Как следствие этих работ стали появляться экспериментальные обучающие системы продуцирующего типа, где обучающие воздействия выбираются не педагогом, а определяются алгоритмом функционирования системы и генерируются в зависимости от целей обучения и текущей ситуации. При этом предполагается, что в обучающей системе представлены знания о том, чему обучать, как обучать и знания о самом обучаемом.

Третий этап – вторая половина 80-х и 90-е годы. Этот период характеризуется двумя основными тенденциями. С одной стороны, широкое распространение персональных компьютеров (ПК) и развитие вычислительных сетей ориентирует обучающие системы на работу в сети с использованием общепринятых стандартов представления и передачи данных. С другой стороны, возросшие аппаратные возможности привели к тому, что одним из основных направлений развития обучающих систем стало применение в них новых компьютерных технологий (в первую очередь, гипертекста и мультимедиа). Повальное увлечение новомодными технологиями отодвинуло на второй план содержательную и методическую составляющие обучающих систем [30].

Вместе с тем, к середине 80-х стало ясно, что интеллектуализация обучающих систем в первую очередь связана с практическим использованием при их разработке и реализации методов и средств, созданных в рамках исследований по экспертным системам. Это, в свою очередь, вызвало к жизни серьезные исследования по моделям объяснения в обучающих системах [101], с одной стороны, и интеллектуальным технологиям формирования моделей предметной области, стратегий обучения и оценки знаний обучаемых на основе более сложных моделей

самих обучаемых, с другой стороны. Это позволило говорить об адаптирующихся обучающих системах, которые могли в зависимости от параметров обучаемого и результатов контроля знаний генерировать новые последовательности управляющих воздействий [26, 91].

С развитием вычислительных сетей и, в частности, сети Internet обучающие системы получили возможность выхода на новый уровень. При переходе от локальных обучающих систем к распределенным качественно изменяются функциональные возможности (прежде всего за счет объединения сетевых ресурсов для решения стоящих перед системой задач). Использование средств телекоммуникаций позволяет значительно расширить круг пользователей системы. Более того, при организации работы через вычислительную сеть общение между обучаемыми и преподавателем может быть даже более интенсивным, чем при традиционном обучении в высшей школе. Преподаватель получает возможность постоянного контроля состояния процесса обучения (в первую очередь, с использованием средств автоматического контроля), а обучаемый – возможность консультации в режиме on-line или по электронной почте.

Использование сетевых технологий и достижений в области искусственного интеллекта дает возможность создания перспективных обучающих систем, которые позволят адаптировать учебный процесс к конкретному обучаемому [32].

Итак, первые два этапа в разработке обучающих систем (60-е и, частично, 70-е годы) характеризовались активной работой по созданию специального программного обеспечения для обучающих систем, причем основное внимание уделялось авторским языкам "пакетного" описания обучающих программ. Для следующих этапов характерно возрастание роли инструментария общего назначения для разработки компонентов компьютерных обучающих программ, а также инструментария для формирования базы предметных знаний, реализации моделей обучения и обучаемого. Появляется понимание того, что будущее обучающих систем

связано с использованием возможностей вычислительных сетей и средств телекоммуникации.

Таким образом, отдельные задачи, из которых складывается проблема автоматизации обучения, уже имеют решение как в методологическом, так и в программном плане. Использование готовых решений не только упростит стоящую перед нами задачу, но и повысит качество ее решения. Современное развитие средств ВТ и программного обеспечения дает основания говорить о принципиальной возможности создания полнофункциональной РАОС.

1.2. Распределенная автоматизированная обучающая система

Распределенность обучающей системы имеет несколько аспектов:

1. Предоставление удаленного доступа к системе предполагает работу в режиме клиент–сервер.
2. Система должна обеспечивать поддержку распределенных данных.
3. Создание системы подобного уровня сложности в принципе возможно лишь при использовании вычислительных возможностей, предоставляемых сетью. Если говорить о полнофункциональной АОС, то решение задачи автоматизированного обучения в максимальном варианте включает:
 - предоставление учебных материалов в различных формах (текст, гипертекст, графика, аудио- и видеоматериалы и т.д.);
 - выполнение практических работ (моделирование, проектирование, решение задач и пр.);
 - организация диалога с обучаемым (т.е. ответы на его вопросы);
 - определение уровня знаний обучаемого;
 - адаптация системы к уровню знаний обучаемого в соответствии с целью обучения.

Размещение отдельных модулей, входящих в состав АОС, на разных узлах сети позволит повысить степень параллелизма работы системы с множеством пользователей.

Выделим основные принципы построения РАОС [28]:

1. Распределенность: функционирование на основе компьютерных сетей.
2. Полнофункциональность: предоставление возможности использования практически любых известных к настоящему времени технологий и методов компьютерного обучения.
3. Универсальность, т.е. пригодность базового программного обеспечения РАОС для создания произвольных курсов и изучения любых дисциплин (естественнонаучных, технических, гуманитарных).
4. Открытость, т.е. предоставление возможности использования готового программного и информационного обеспечения.
5. Стандартизация: использование стандартных сетевых и программных решений и построение системы на основе универсальной интегрированной базы данных, что позволит легко и практически неограниченно наращивать, переносить и масштабировать ее.

1.2.1. Применение сетевых технологий в обучающих системах

В последнее время в мире наблюдается повышенный интерес к использованию для образовательных целей ресурсов международных глобальных компьютерных сетей. Среди наиболее распространенных Internet-технологий, которые используются для информационного обеспечения образовательных услуг, можно выделить [51]:

- системы электронной почты (E-mail);
- средства организации файловых архивов и доступа к ним (FTP);
- сетевая файловая система (NFS);
- push-технология принудительной доставки информации;
- глобальная распределенная гипертекстовая информационная система (WWW).

Кроме этих стандартных для Internet технологий, в обучении применяются адаптивные обучающие системы, которые используют такие интересные подходы, как [15]:

- адаптивное планирование (curriculum sequencing),

- интеллектуальный анализ решений обучаемого,
- поддержка интерактивного решения задач,
- поддержка решения задач на примерах и поддержка совместной работы,
- поддержка адаптивного представления и адаптивной навигации.

Все технологии адаптации, применяемые в адаптивных обучающих системах в Web, взяты либо из области ИОС, либо из области адаптивной гипермедиа. В ближайшем будущем ожидается появление новых технологий адаптации, например: адаптивная поддержка совместной работы, специально спроектированная для обучения в Web.

В процессе обучения используются как информационные ресурсы общего назначения, уже существующие в глобальных сетях, так и специальные "образовательные" серверы. Но существующие образовательные серверы нельзя однозначно отнести ни к распределенным системам, ни к автоматизированным обучающим системам.

Как пример наиболее распространенного подхода к организации образовательных серверов можно привести инструментальную систему Distance Learning Studio, разработанную Санкт-Петербургским отделением института "Открытое общество" и компанией ГиперМетод [131]. Эта система предоставляет доступ в режиме клиент–сервер и обладает встроенной системой контроля знаний, основанной на дихотомических тестах, т.е. тестах, которые можно либо решить полностью, либо не решить вообще.

РАОС – это АОС, функционирующая в сети и объединяющая вычислительные возможности сети для реализации своих функций. Обучающая система не может быть названа распределенной, если она работает только в режиме удаленного доступа и не использует вычислительные возможности сети.

С другой стороны, АОС предполагает постоянное наблюдение за ходом обучения и адаптацию процесса обучения к индивидуальным характеристикам обучаемых. Поэтому отсутствие возможностей адаптации не позволяет считать образовательный сервер автоматизированной обучающей системой [15].

1.2.2. Телекоммуникационная среда РАОС

При построении распределенной автоматизированной системы обучения, предполагающей доступ как в рамках локальной вычислительной сети (ЛВС), так и через Internet, встает задача оптимального выбора среды передачи информации между элементами системы.

Оптимальность определяется временем реакции системы на запрос в интерактивном режиме работы и отношением скорости передачи данных к стоимости услуг связи.

Современные телекоммуникационные сети характеризуются большим разнообразием технологий и протоколов. Аналоговые системы связи в меньшей степени отвечают требованиям режима on-line из-за невысоких скоростей и ненадежности соединения. Они могут успешно применяться для низкоскоростной передачи данных, в частности по протоколу X.25, при режиме "толстый клиент – тонкий сервер".

Строительство и аренда выделенных цифровых каналов связи, построенных на основе медных кабелей, оптоволокну, беспроводных и спутниковых каналов связи, обходятся значительно дороже. Поэтому их использование для образовательных целей ограничено. Но в тех случаях, когда это возможно (например, в рамках образовательного учреждения), следует использовать выделенные каналы. Они отличаются надежностью и более высокими скоростями передачи данных, что позволяет расширить сферу услуг, предоставляемых системой.

Считается [36], что одной из наиболее перспективных технологий для организации РАОС является цифровая сеть с интеграцией служб ISDN (Integrated Services Digital Network). В основе ISDN лежит устоявшаяся технология и использование оборудования и каналов существующих телефонных сетей общего пользования (ТфОП).

По мере развития и удешевления сетевых технологий для организации распределенной обучающей системы будут применяться и новые сетевые технологии [37], среди которых можно отметить такие, как:

- сети с асинхронным режимом передачи данных (АТМ), которые рассчитаны на передачу любых видов трафика с высокой надежностью и эффективностью, а также масштабировать полосу пропускания;
- сети с ретрансляцией кадров (frame relay), поддерживающие многоточечные топологии и обычно базирующиеся на выделенных линиях; сети frame relay хорошо зарекомендовали себя при передаче различных видов трафика, в том числе для работы в режиме on-line, когда очень высоки временные требования к системе;
- многоточечная передача данных на основе коммутации ячеек (SMDS, Synchronous Multimegabit Digital Service);
- широкополосная ISDN (B-ISDN, Broadband ISDN);
- высокоскоростная передача интегрированных данных по сетям кабельного телевидения (КТВ) и телефонным проводам (xDSL).

Эти технологии очень перспективны, но пока малодоступны и дороги.

1.2.3. Компьютерные сети РАОС

В общем случае распределенная обучающая система состоит из нескольких подсистем, связанных вычислительной сетью. Каждая из этих подсистем является либо автономной полнофункциональной обучающей системой, либо выполняет отдельные операции, необходимые для функционирования РАОС.

В системе дистанционного обучения каналы связи, обеспечивающие взаимодействие удаленных элементов системы, должны быть не постоянными, а коммутируемыми при наличии информации для передачи. В этом случае одним из наиболее приемлемых решений – как по функциональным возможностям, так и по стоимости – может стать использование сетей ISDN. Они обеспечивают такие функции, как связь по требованию, пропускная способность по требованию (объединение нескольких В-каналов в один логический канал), компрессия данных в канале, защита информации, и позволяют реализовать разнообразные

решения проблем организации связи в системе дистанционного обучения. Сети ISDN способны решить также вопросы организации подключения ЛВС и рабочих мест пользователей к Internet при условиях достаточно высоких требований к пропускной способности сети и ограниченности финансовых ресурсов [36].

Общая схема РАОС приведена на рис.1.1.

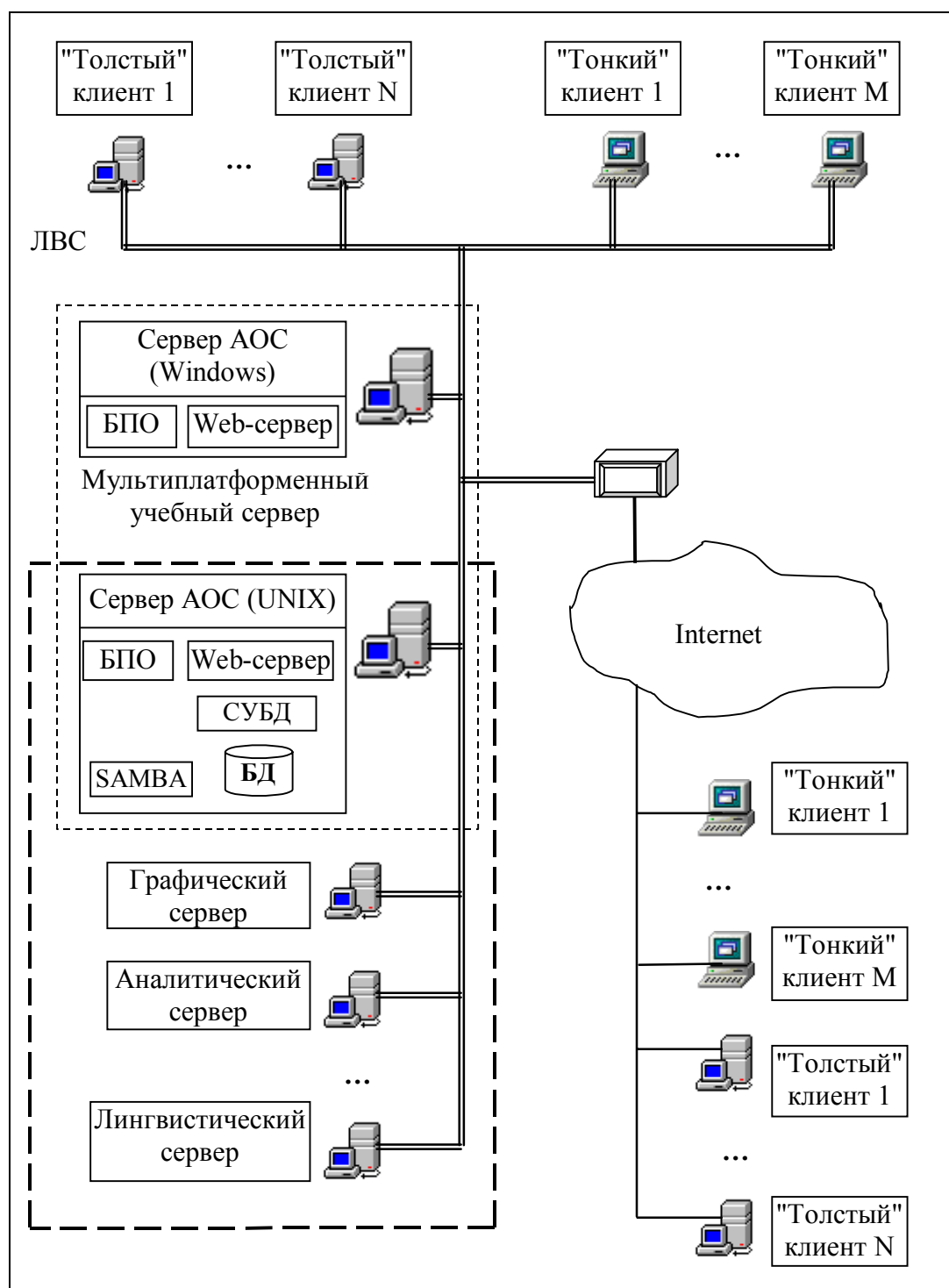


Рис. 1.1. Общая схема РАОС

В режиме "клиент-сервер" в зависимости от расположения программного обеспечения и разделения функций различают два типа организации работы:

1. "тонкий" клиент – "толстый" сервер;
2. "толстый" клиент – "тонкий" сервер.

В существующих обучающих системах в основном используется первый вариант (например, [46]).

Теперь рассмотрим положение дел в области создания и использования автоматизированных обучающих систем.

1.3. Классификация компьютерных средств учебного назначения

За время развития автоматизированного обучения было создано огромное количество обучающих систем разного уровня и назначения (см., например, сборник [75]). В литературе по компьютерным средствам обучения используются различные термины, характеризующие типы программ учебного назначения. К сожалению, разные исследователи часто вкладывают в один и тот же термин существенно разный смысл или наоборот, однотипные программы характеризуют разными терминами [50]. Рассмотрим различные классификации обучающих систем.

1.3.1. Существующие классификации

В основе всех видов классификации лежит отношение эквивалентности. При классификации некоторого множества в нем задают одно или несколько отношений эквивалентности и рассматривают классы эквивалентности, связанные с этими отношениями.

1. По принципам взаимодействия программных обучающих средств и обучаемого эти программные средства можно разделить на два больших класса [89]:

- учебные среды,
- обучающие программы.

При работе в учебной среде предполагается, что обучаемый имеет определенную цель (неважно, задана ли она обучающим или самим обучаемым), а система assiste ему в достижении этой цели. В системах данного класса отсутствует (по крайней мере явно) этап контроля обучаемого со стороны системы, так как система не знает цели, с которой обучаемый обратился к ней.

Жесткой границы между учебными средами и обучающими программами нет. Действительно, системы обеспечивающие демонстрацию учебного материала в своем развитии "идут" в направлении учебных сред. А учебные среды-тренажеры в определенной области в конечном счете приближаются к обучающим программам-тренажерам. Единственное различие, остающееся между обучающими системами этих классов – отсутствие контроля фискального типа в учебных средах и наличие его в обучающих программах.

2. Кривошеев А.О. (РосНИИИС) [52–53] предлагает делить компьютерные обучающие средства на следующие классы:

- Компьютерные (или электронные) учебники (КУ). Обеспечивают возможность самостоятельно освоить учебный курс или его раздел.
- Предметно-ориентированные среды (ПОС) [60, 122, 139, 141]. Это учебные пакеты программ, позволяющие оперировать с объектами определенного класса.
- Лабораторные практикумы (ЛП), позволяющие автоматизировать выполнение лабораторных работ по различным дисциплинам.
- Тренажеры (ТР). Служат для отработки и закрепления технических навыков решения задач.
- Контролирующие программы (КП), предназначенные для проверки (оценки) качества знаний обучаемых.
- Инструментальные системы (ИС) [17, 68, 93, 103, 112]. Это программные комплексы, предназначенные для создания различных программ учебного назначения.

- Справочники, базы данных учебного назначения (УБД), обеспечивающие хранение и предъявление учащемуся разнообразной учебной информации справочного характера.

Недостатки данной классификации очевидны. Приведем только один пример. В соответствии с этой классификацией контролирующие программы вынесены в отдельный класс, в то время как многие компьютерные учебники включают контроль знаний как составляющую часть. Таким образом, мы видим эффект пересечения классов, что говорит о неоднозначности данной классификации.

3. Савельев А.Я. выделяет два класса обучающих систем [98]:

- Автоматизированные обучающие системы (АОС). Ядром АОС являются т.н. авторские системы, позволяющие преподавателю вводить свой учебный материал в базу данных системы и программировать алгоритмы изучения с помощью специальных средств.
- Отдельные программы, пакеты программ, элементы автоматических систем (АСУ, САПР, АСНИ, АСУП и др.), предназначенные для автоматизации трудоемких расчетов, оптимизации, исследования свойств объектов и процессов на математических моделях и т.п. Особое место среди таких систем занимают экспертно-обучающие системы (ЭОС) [58, 80], которые обычно представляют собой промышленные экспертные системы (ЭС), адаптированные к целям обучения.

4. Соловов А.В., автор одной из известных обучающих систем (КАДИС – Комплекс Автоматизированных Дидактических Средств [103]) предлагает условно делить обучающие системы на два множества:

- Компьютерные обучающие программы (КОП), под которыми понимается любое программное средство, специально разработанное или адаптированное для применения в обучении. Назначение КОП – поддержка всех или отдельных составляющих учебного процесса.

- Инструментальные системы, предназначенные для разработки КОП и создания учебных курсов.

Две последние классификации близки друг другу, и обе объединяют в один класс такие разные программные средства как, например, лабораторный практикум по физике и тренажер по математическому анализу (МГТУ им. Баумана, [69]).

Как видно из приведенного перечня, ни одна из предложенных классификаций не является исчерпывающей. Все они страдают одним недостатком: это "плоские" классификации, в основе которых лежит **один** классифицирующий признак, например, функциональность или принцип взаимодействия с обучаемым. Все подобные классификации не являются однозначными, т.е. строго разграничивающими все пространство объектов.

В последнее время предпринимаются попытки систематизации учебных компьютерных средств. Например, Б.Х. Кривицкий [50] предлагает проводить классификацию ("разграничение") по двум признакам:

1. Вид учебной деятельности:

- средства, предназначенные для групповой работы;
- средства для самостоятельной работы.

2. Педагогическое (или дидактическое) назначение.

- информационные, обеспечивающие прямой канал передачи;
- контролирующие, обеспечивающие обратный канал передачи;
- обучающие, обеспечивающие замкнутый цикл управления.

Но и это предложение не решает задачи однозначной классификации, что признает и сам автор: "В настоящее время появляются такие новые... комплексы, которые предполагают использование как при самостоятельной, так и при групповой учебной работе... Это означает, что принятые подходы для таких комплексов как бы сливаются, и различия подходов для них становятся неактуальными" [там же].

Приведенных примеров достаточно для того, чтобы попробовать разобраться с тем, что такое классификация вообще и как должна выглядеть "хорошая" классификация в частности.

1.3.2. Виды классификации

Кроме "плоских" существуют *иерархические* и т.н. *фасетные* классификации [11].

В иерархической классификации все множество объектов разлагается на классы эквивалентности, после чего каждый класс может быть разбит на подклассы эквивалентности по другому отношению и т.д. Ограниченность этого подхода проистекает из самого принципа иерархии, в соответствии с которым каждый классифицируемый объект может быть приписан единственному классу. Поэтому все подобные классификации также являются не однозначными либо чрезвычайно избыточными.

Наиболее известная из иерархических классификаций – это УДК (универсальная десятичная классификация), которая является основой библиотечных каталогов. Также с помощью иерархических классификаторов организовано большинство современных поисковых серверов в Internet.

При использовании фасетных классификаций на одном множестве объектов строится несколько классификаций (возможно, иерархических) по разным признакам. Приведем пример такой классификации.

Принято считать [115–116], что экспертные системы по своим характеристикам делятся:

1. По назначению (цель создания, состав и характеристика пользователей; консультационные, исследовательские и управляющие).
2. По предметной области; в том числе различают статические и динамические предметные области (статическая или динамическая ЭС).
3. По степени сложности (по глубине анализа предметной области): поверхностные и глубинные. Поверхностные ЭС представляют знания об области экспертизы в виде правил (условие → действие). Глубинные ЭС,

помимо этого, обладают способностью при возникновении неизвестной ситуации определять с помощью некоторых общих принципов, справедливых для области экспертизы, какие действия следует выполнить.

4. По стадиям существования (демонстрационный прототип, исследовательский прототип, опытная эксплуатация, промышленный прототип, коммерческая система).
5. По типу используемых методов и знаний.
6. По используемым инструментальным средствам.

Фасетные классификации лучше иерархической. Но фактически они представляют собой совокупность нескольких классификаций, обычно иерархических, каждая из которых определяет принадлежность объекта к определенному классу по одному из признаков объекта. И если множество классифицирующих признаков достаточно велико, мы получим очень громоздкую, избыточную классификацию.

1.3.3. Принцип многомерной классификации

Основным критерием корректности построения классифицирующих систем является взаимно-однозначное соответствие между объектом и классифицирующей системой. То есть однозначное местоположение объекта в системе (идентификация) с одной стороны и возможность непротиворечивого определения множества свойств искомого объекта – с другой.

Для обеспечения этого требования представим все множество классифицируемых объектов в виде точек в некотором конечномерном пространстве признаков. Координатными осями этого пространства будут независимые классифицирующие признаки. При этом пространство признаков может быть как дискретным (чаще всего классифицирующих признаков немного и они "дискретны"), так и частично непрерывным. В

последнем случае речь может идти о некоторых "непрерывных" признаках (например, степень полноты, глубины, удобства и т.п. чего-либо).

Введя оси в соответствии с классифицирующими признаками и определив диапазон или перечень их значений мы получим универсальную классифицирующую систему – *многомерный классификатор*.

Принцип, лежащий в основе такой классификации, позволяет, во-первых, четко различать два объекта, если они существенно отличаются друг от друга. Во-вторых, эта классификация в любом случае не потребует внесения кардинальных изменений при появлении новых классов объектов. Достаточно будет добавления новых значений на существующие оси или добавления осей, соответствующих новым классификационным признакам.

Разумеется, определение набора осей – классифицирующих признаков – является непростой задачей, т.к. необходимо учитывать их "ортогональность", несоблюдение которой приведет к тому, что объект может быть представлен не одной точкой, а множеством. При неудачном выборе осей существует опасность чрезмерного возрастания размерности пространства.

Этот метод позволит строить разделяющие гиперплоскости в пространстве классифицирующих признаков. Возможность одновременно определять для конкретного объекта несколько значений одного признака делает классификатор более гибким и избавляет от необходимости учитывать в качестве значения признака различные варианты комбинаций элементарных значений.

Замечание: задача создания исчерпывающей классификации обучающих систем выходит за рамки данной работы, но предложенный принцип классификации, по мнению автора, позволяет сделать существенный шаг в направлении создания такой классификации. Эта работа требует участия различных специалистов в области компьютерных технологий обучения и может быть выполнена только в ходе открытой дискуссии в печати, началом которой можно считать статью [40]. Руководствуясь этим принципом, можно сформировать набор классифицирующих признаков, которые являются существенными с точки зрения исследователей, разработчиков и пользователей обучающих систем. Далее, для каждого устоявшегося термина определить, какими значениями этих признаков характеризуется тот или иной программный продукт, что внесет ясность в определения всех терминов и позволит специалистам говорить на одном языке.

Наличие классификации дает возможность определить принадлежность конкретного программного продукта к определенному классу, но не позволяет оценить качественные показатели обучающей системы.

По стандарту ISO 8402:1986 качество продукта определяется совокупностью его признаков и характеристик, которые позволяют рассматривать данный продукт с точки зрения его способности удовлетворять утвержденные или предполагаемые потребности. Для того чтобы получить возможность оценивать качество обучающей системы, необходимо рассмотреть схему процесса обучения и соответствие обучающей системы этой схеме.

1.4. Схема процесса обучения

Процесс обучения можно трактовать как процесс управления усвоением знаний, что не противоречит современной педагогике [12, 50]. Этот процесс реализуется в замкнутой системе и (как для любой замкнутой системы управления, [62]) характеризуется целью управления, имеет объект управления (обучаемых), устройство управления и канал обратной связи (рис. 1.2). Критерием качества управления могут служить результаты контроля знаний.

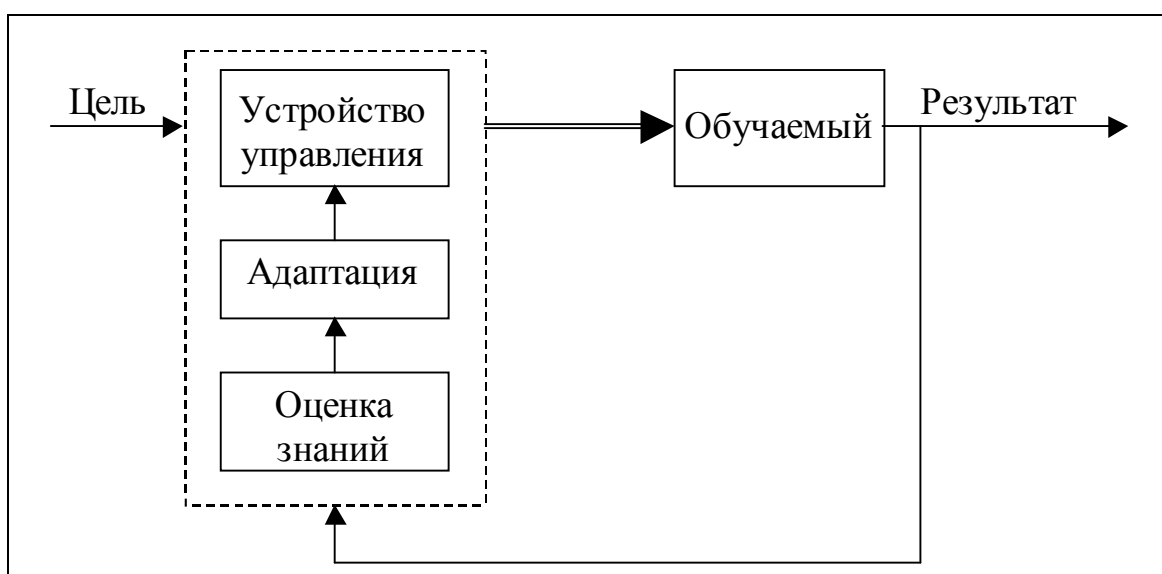


Рис. 1.2. Обобщенная схема процесса обучения

Схема, приведенная на рис. 1.2, является упрощенной и формальной. Она не учитывает (и не может учитывать) те особенности, которые накладывает на реальный процесс обучения участие в нем человека. Но эта схема дает общее представление о предмете обсуждения. В соответствии с целью обучения устройство управления вырабатывает набор управляющих воздействий на обучаемого (например, предъявление учебного материала или контрольного задания). Ответная реакция обучаемого (уточняющий вопрос, ответ и т.п.) по каналу обратной связи передается устройству управления и позволяет ему корректировать управляющие воздействия для достижения желаемого результата [110].

Очевидно, что традиционное обучение является адаптивным процессом. Следовательно, автоматизированная система обучения также должна быть адаптивной. Адаптация обучающей системы может заключаться в изменении параметров управления и изменении набора правил, вырабатывающих управляющее воздействие.

Адаптивная система должна уметь оценивать результат управляющих воздействий системы на обучаемого. Таким образом, особое значение в обучающей системе приобретает оценка знаний. Именно она в первую очередь обеспечивает обратную связь системы с объектом управления (обучаемым) и позволяет системе оценивать качество управления и адаптироваться к обучаемому. Для обеспечения адекватной оценки знаний методы анализа и оценки ответов обучаемых на контрольные вопросы должны базироваться на монотонном функционале, чтобы система могла отличать абсолютно неправильные ответы от частично неправильных.

Сложность задачи по организации обучающей системы заключается в том, что процесс обучения не формализован (возможно, в принципе не может быть формализован [50]). Не существует готового набора формальных параметров, с помощью которых можно было бы сформулировать цель обучения и критерии достижения этой цели. Следовательно, в отсутствие формальных моделей управления обучением, нужно использовать другие

подходы, например, методы инженерии знаний или нечеткую логику, которые предназначены для решения слабо формализованных задач.

Для организации работы обучающая система должна включать знания [30, 145]:

- о предметной области (о предмете изучения).

Если обучающая система является инструментальным средством, то она должна настраиваться на произвольную предметную область.

- о методике обучения (правила формирования управляющих воздействий).
Набор этих правил должен базироваться на теории обучения. К сожалению, единой общепризнанной теории обучения не существует. Разные специалисты в области обучения имеют различные представления об эффективных методах преподавания и целесообразности их использования.

Отсюда можно сделать вывод: в отсутствие единой теории обучения знания о стратегии обучения должны быть такими же вариативными, как и знания о предметной области, чтобы преподаватель мог вложить в систему свои представления об эффективной методике обучения.

- об обучаемом (об объекте управления).

Информация о цели обучения и о текущем состоянии процесса обучения нужна системе для обеспечения возможности адаптации к обучаемому и определения степени достижения цели.

Общепризнанных формальных методов описания модели обучаемого также не существует. Каждый специалист в области обучения имеет право предложить и обосновать свой набор параметров модели обучаемого и критериев достижения цели обучения. Таким образом, обучающая система должна предложить преподавателю механизм для формирования модели обучаемого, а не просто готовую модель.

Теперь рассмотрим современное состояние автоматизированных обучающих систем и оценим их соответствие принятой схеме обучения.

1.5. Автоматизированные обучающие системы (АОС)

Многие исследователи (например, [17, 23, 50, 55, 68, 99, 134]) считают АОС наиболее перспективным направлением в области создания компьютерных средств учебного назначения по целому ряду причин.

1.5.1. Преимущества автоматизированных обучающих систем

Преимущества АОС перед другими видами обучающих систем в первую очередь определяются тем, что фактически АОС – это инструментальный комплекс для создания компьютерных средств учебного назначения. Различие между произвольной обучающей системой и инструментальным комплексом для создания обучающих систем заключается в том, что инструментальный комплекс не содержит знаний по конкретной предметной области.

Создание любой обучающей программы требует больших трудозатрат [8]. По данным Дж. Морриса [138] один час курса стоит около 10 000 долларов. А. Борк [128] на основе экспертных данных оценил затраты на разработку 6–семестрового автоматизированного курса по высшей математике в 3–5 млн. долларов. Применение АОС позволяет сократить трудоемкость и сроки разработки за счет наличия готовых механизмов для выполнения функций обучения [53].

Кроме удешевления и ускорения разработки обучающей системы применение АОС имеет следующие преимущества:

- в систему можно вложить знания по произвольной предметной области;
- АОС предоставляет возможность в той или иной мере настроить систему на требования преподавателя;
- применение одной и той же АОС для разных курсов приводит к унификации интерфейса и, как следствие, к уменьшению времени на изучение правил работы с системой, затрачиваемое обучаемым.

Определим, что понимают под термином *автоматизированная обучающая система*, и какими признаками она должна характеризоваться.

1.5.2. Понятие автоматизированной обучающей системы

По причине отсутствия однозначной классификации и сложившейся терминологии к 2002 г. не сложилось единого мнения относительно определения автоматизированных обучающих систем. Существует множество определений АОС. Проследим за эволюцией этих определений с 60-х годов XX века и до наших дней.

В начале работ по созданию обучающих систем в соответствии с глобальной целью их создания под АОС понимался комплекс программных и аппаратных средств, предназначенных для автоматизации процесса обучения [87] (1963).

Прошло десятилетие, и понимание сути АОС изменилось. В качестве подтверждения можно привести работу [22] (1976), в которой АОС рассматривается как взаимосвязанный комплекс средств информационного, математического и программного обеспечения, организованный на базе ЭВМ и предназначенный для управления процессом обучения.

Еще десятью годами позже (1985) АОС определялись уже как функционально взаимосвязанный набор подсистем учебно-методического, информационного, математического и инженерно-технического обеспечения на базе средств вычислительной техники, предназначенный для оптимизации процесса обучения в различных его формах и работающий в диалоговом режиме коллективного пользования [78].

К настоящему времени сформировалось отношение к АОС как к "организованному на базе ЭВМ комплексу средств технического, лингвистического, учебно-методического и программного обеспечения, предназначенному для диалогового учебного взаимодействия и образующему программную оболочку, приспособленную для заполнения... учебным материалом пользователем-непрограммистом" [24] (1998).

Как видим, в разное время давались похожие определения АОС, но расставлялись разные акценты. Смещение этих акцентов – от автоматизации процесса обучения к управлению обучением, потом к оптимизации обучения,

и, в конце концов, к созданию оболочек, поддерживающих диалог с обучаемым – отражает изменение понимания места и роли обучающих систем в образовательном процессе.

Суммируя все вышесказанное и учитывая наметившуюся тенденцию возврата к ранее поставленной цели автоматизации обучения, можно дать еще одно определение: АОС – это инструментальный комплекс, включающий математическое, методологическое и программное обеспечения и предназначенный для заполнения знаниями пользователем-непрограммистом в целях создания обучающей системы и, в конечном итоге, реализации автоматизированного обучения.

С учетом распределенности и на основании рассмотренной схемы обучения систематизируем требования, которым должна отвечать РАОС.

1.5.3. Требования, предъявляемые к РАОС

Основные требования, предъявляемые к РАОС, можно условно разбить на две группы:

1. требования, обусловленные распределенным характером системы:
 - организация прозрачного взаимодействия пользователей и системы через вычислительную сеть;
 - возможность объединения разнородных программных средств, распределенных в сети, для решения поставленных задач;
 - поддержка распределенных данных.
2. требования, обусловленные функциональным назначением АОС:
 - учет цели обучения и задание критерия достижения этой цели;
 - возможность заполнения АОС знаниями по предметной области;
 - обеспечение канала обратной связи с обучаемым;
 - адаптация системы к обучаемому.

Для выполнения этих требований РАОС должна:

- иметь средства организации удаленного доступа к системе;
- включать средства межпроцессного взаимодействия;

- содержать инструментарий, позволяющий настроить РАОС на предметную область путем внесения в нее прикладных знаний;
- иметь средства предварительного определения уровня подготовки обучаемого и настройки системы на определенный уровень квалификации;
- содержать средства создания и использования различных стратегий обучения и разнообразных моделей обучаемого;
- включать средства контроля процесса обучения.
- поддерживать различные формы и виды обучения (лекции, лабораторные и контрольные работы, практические занятия и т.п.);

Реализация этих базовых требований представляет сложный и трудоемкий процесс, поэтому даже наиболее передовые АОС соответствуют лишь некоторым из указанных требований [133].

Существующие в настоящее время реально действующие АОС не могут характеризоваться как распределенные, потому что самое большее, что они обеспечивают, – это возможность удаленного доступа к учебному серверу (например, [20, 23]). (Поэтому при анализе существующих АОС имеет смысл учитывать только требования, определяемые функциональным назначением обучающих систем.)

Те же системы, которые можно зачислить в разряд распределенных, не могут считаться автоматизированными обучающими системами. Например, Центр дистанционного образования МГИЭМ функционирует на основе системы Lotus Notes. Фактически это система документооборота с возможностью обмена информацией по электронной почте, снабженная средствами автоматизированного контроля знаний. Но результаты контроля знаний обучаемого не влияют на поведение системы. Очевидно, что этих функций недостаточно для того, чтобы называться АОС.

Более того, подавляющее большинство систем дистанционного образования, работающих в Internet, вообще нельзя отнести к разряду

обучающих систем, т.к. они "являются ничем иным, как просто сетью статичных гипертекстовых страниц" [15].

Применим к обучающим системам ранее предложенный принцип многомерной классификации и рассмотрим возможную классификацию обучающих систем по качественным показателям. В табл. 1.1 выделены подчеркиванием значения признаков, которые должны характеризовать АОС, соответствующую вышеуказанным требованиям.

Таблица 1.1. Классифицирующие признаки обучающих систем по качественным показателям

Признак	Возможные значения признака
Учет цели обучения	- отсутствует - критерий задан, можно изменять его значение - <u>критерий задается преподавателем (составителем курса)</u>
Возможность заполнения АОС знаниями по предметной области	- отсутствует (система настроена на конкретную предметную область) - <u>существует</u>
Канал обратной связи с обучаемым	- отсутствует - <u>контекстная помощь</u> - <u>контроль знаний</u> - <u>учет реакции обучаемого на действия системы</u> - диалог на искусственном языке - диалог на естественном языке
Адаптация системы к обучаемому	- отсутствует - <u>параметрическая адаптация</u> - <u>алгоритмическая адаптация</u>

Детализируем эти требования и рассмотрим обучающие системы как специализированные программные средства, которые должны предоставлять определенный набор функциональных возможностей (табл. 1.2).

Таблица 1.2. Вариант классификации компьютерных средств учебного назначения

Признак	Возможные значения признака
1	2
Открытость	- <u>позволяет подключать внешние модули</u> - <u>настраивается на конкретную предметную область (заполнение знаниями по предметной области)</u> - <u>настраивается на конкретного преподавателя (изменение моделей и методов обучения и контроля знаний)</u> - <u>настраивается на конкретного обучаемого (адаптация к уровню подготовки и цели обучения)</u>

Таблица 1.2. Вариант классификации компьютерных средств учебного назначения
(продолжение)

1	2
Тип настройки	<ul style="list-style-type: none"> - параметрическая настройка - <u>алгоритмическая настройка</u>
Обратная связь с обучаемым	<ul style="list-style-type: none"> - <u>контекстная помощь</u> - <u>контроль знаний</u> - <u>учет реакции обучаемого на действия системы</u> - диалог на искусственном языке - диалог на естественном языке
Распределенность	<ul style="list-style-type: none"> - <u>возможность удаленного доступа</u> - <u>поддержка распределенной информации</u> - <u>поддержка распределенных вычислений</u>
Встроенные возможности	<ul style="list-style-type: none"> - используемые форматы файлов с учебными материалами - тип организации управления обучением: <u>либеральный</u> (по инициативе обучаемого) <u>консервативный</u> (жесткое программное управление) <u>адаптивный</u> (адаптивное программное управление) - модель предметной области: в виде сценария, списка тем, графа, семантической сети и т.п. - используемые встроенные параметры - схемы проведения контроля знаний (линейная, пороговая, адаптивная и т.д.) - используемые типы ответов - поддерживаемые типы тестов - методы определения правильности ответов

Набор классифицирующих признаков, используемых в табл. 1.2, введен по результатам анализа различных программных продуктов и работ [23, 24, 31, 47, 53, 98, 103, 133, 136].

1.5.4. Инструментальные оболочки обучающих систем

В настоящее время имеется много отечественных инструментальных систем общего назначения, предназначенных для создания обучающих программ: КАДИС, АДОНИС, КОБРА, УРОК, АОСМИКРО, СЦЕНАРИЙ, "Наставник" и другие, сравнимых по функциональным возможностям с такими зарубежными ИС как: Costoc, LinkWay, Quest, TenCore [52].

Проанализируем и сравним некоторые используемые в настоящее время инструментальные оболочки обучающих систем (то, что принято называть автоматизированными обучающими системами).

В качестве объектов сравнения были выбраны следующие системы: КАДИС [103]; АОСМИКРО [23]; "ЭКСТЕРН" [78]; КОБРА [20]; "Фея" [55]. Эти системы, с одной стороны, получили определенное распространение за рамки круга разработчиков, и, с другой стороны, демонстрируют разные подходы к решению основных задач, стоящих перед обучающими системами.

Результаты сравнения этих систем приведены в табл. 1.3. Анализ показал такие достоинства этих систем, как:

- развитые возможностями представления информации;
- наличие средств настройки системы;
- возможность удаленного доступа (по сети);
- разнообразные формы представления и анализа ответов обучаемого;
- возможность сбора статистики обучения.

Но эти системы не лишены недостатков:

1. Используемая методология обучения либо программируется (ЭКСТЕРН, "Фея"), либо уже заложена в алгоритм функционирования системы (КАДИС, КОБРА, АОСМИКРО), причем обоснование этой методологии отсутствует во всех рассмотренных системах, за исключением системы КАДИС.
2. Возможности организации адаптивного управления обучением отсутствуют (КАДИС, КОБРА, АОСМИКРО) или ограничены необходимостью программирования (ЭКСТЕРН, "Фея").
3. Отсутствие модели обучаемого сужает возможности системы по адаптации к обучаемому. Модель обучаемого есть только в системе ЭКСТЕРН, но алгоритм ее формирования запрограммирован и основан на статистическом наборе встроенных параметров.

Таблица 1.3. Инструментальные оболочки обучающих систем

Свойства	КАДИС	ЭКСТЕРН	КОБРА	АОСМИКРО	Фея
Методология управления обучением	Труды В.П. Беспалько [12]	Основана на ПО ¹ , программируется на авторском языке	Отсутствует	Основана на ПО ¹ , программируется на авторском языке	Программируется визуально или на языке Pascal
1. Удаленный доступ	Нет	Да	Да, развитый	Да	Да
Общие качественные показатели					
2. Управление обучением					
учет цели обучения	Да	Да	Нет	Нет	Программируется ²
критерий достижения цели	Задан, можно менять значение	Задан, можно менять значение	Нет	Нет	Программируется на языке Pascal
сценарий курса	Да	Да	Да	Да	Да
адаптация к обучаемому	Нет	Программируется на авторском языке	Нет	Нет	Программируется на языке Pascal
3. Обратная связь с обучаемым					
учет реакции обучаемого на действия системы	Да, программируется визуально	Да, запрограммирован	Нет	Нет	Программируется визуально
контекстная помощь	Программируется визуально	Программируется на авторском языке	Включена в файлы данных	Программируется на авторском языке	Программируется визуально
контроль знаний	Да	Да	Да	Да	Да
4. Возможности настройки					
на предметную область	Да	Да	Да	Да	Да
на преподавателя (изменение моделей и методов обучения)	Ограниченная модификация методов обучения	Программируется на авторском языке	Нет	Нет, но может быть запрограммирована на языке Pascal	Нет, но может быть запрограммирована на языке Pascal
на обучаемого (адаптация к уровню подготовки и цели обучения)	Нет	Программируется на авторском языке	Нет	Нет, но может быть запрограммирована на языке Pascal	Нет, но может быть запрограммирована на языке Pascal

Таблица 1.3. Инструментальные оболочки обучающих систем (продолжение)

Свойства	КАДИС	ЭКСТЕРН	КОБРА	АОСМИКРО	Фея
Дополнительные характеристики					
5. Типы организации обучения:					
либеральный	Да	Да	Да	Да	Программируется
консервативный	Да	Да	Да	Да	Программируется
адаптивный	Нет	Программируется на авторском языке	Нет	Нет	Программируется на языке Pascal
6. Модель обучаемого	Нет	Да (фиксир. алгоритм и набор параметров)	Нет	Нет	Нет
7. Предметная область					
модель предметной области	В виде графа	Нет	Нет	Нет	Нет
принцип организации учебного материала	Гипертекст и гипермедиа	Набор информац. и контрольн. кадров	Файлы форматов doc,txt,htm,gif...	Набор информац. и командных кадров	Система меню, окон и кадров
место хранения учебного материала	База данных (БД) системы	БД системы	БД системы	БД системы и внешние файлы	БД системы
11. Контроль знаний					
тестирование	Да	Да	Да	Да	Да
адаптивный контроль	Нет	Программируется на авторском языке	Нет	Программируется (на языке Pascal)	Программируется на языке Pascal
контроль с определением причин ошибок	Нет	Нет	Нет	Программируется (на языке Pascal)	Программируется на языке Pascal
реализованные схемы контроля знаний	ФЧВ ³ , билеты и пороговая схема	Программируются на авторском языке	Фиксированный тест	Программируется (на авт. Языке)	ФЧВ
алгоритм проведения контроля	Фиксированный	Фиксирован или программируется на авторском языке	Фиксированный	Фиксированный или программируется	Фиксированный или программируется

Таблица 1.3. Инструментальные оболочки обучающих систем (продолжение)

Свойства	КАДИС	ЭКСТЕРН	КОБРА	АОСМИКРО	Фея
Типы ответов	Выборочные, конструируемые и числовые ответы	Выборочные, конструируемые и выборочно-конструируемые ответы	Выборочные и числовые ответы	Мат. и хим. формулы, текст, графики, структурные схемы, вектора и др.	Строка символов, выборочные (выбор номера ответа или области на экране)
Анализ ответов	Требует перечисления всех возможных эталонов	Перечисление всех возможных эталонов или шаблон ответа	Примитивный	Развитый (можно использовать шаблоны ответов)	Примитивный
Подсистема ввода ответов	Простая	Простая	Простая	Сложная	Простая
8. Статистика	Да	Да	Да	Да	Да
Дополнительные характеристики АОС как программного средства					
9. Параметры					
встроенные	Да	Да	Да	Да	Да
определяемые пользователем	Нет	Нет	Нет	Да	Нет
10. Необходимость программирования	Визуальное программирование	Программирование на авторском языке	Нет	Визуальное программирование	Программирование на языке Pascal
11. Подключение внешних модулей	Да, через стандартный интерфейс	Нет	Да	Да	Да

¹ – программированное обучение.

² – если не указан язык программирования, а система допускает различные возможности (например, авторский язык и Pascal), значит можно использовать и тот, и другой язык.

³ – фиксированное число вопросов, произвольная выборка вопросов из теста.

4. Возможности организации адаптивного контроля отсутствуют (КАДИС, КОБРА) или ограничены необходимостью программирования (ЭКСТЕРН, "Фея", АОСМИКРО).
5. Только система КАДИС поддерживает модель предметной области (в виде графа), но при изложении учебного материала все равно сводит эту модель к списку тем (учебных элементов).
6. Все рассмотренные системы являются "вещью в себе", т.е. требуют хранения учебных и контролирующих материалов (тестов) в базе данных системы, не обеспечивая переносимость материалов в другую оболочку.
7. Формы представления ответов обучаемого в каждой системе ограничены, методы определения правильности ответов запрограммированы и не подлежат изменению.

Нельзя не отметить, что все составные части АОС в том или ином виде реализованы хотя бы в одной из рассмотренных систем. Но нет системы, которая соответствовала бы всем требованиям, предъявляемым к АОС.

1.5.5. Основные направления исследований

В настоящее время во всем мире активно ведутся работы по исследованию и разработке АОС [54, 132, 135]. Основные направления этих исследований:

- создание новых форм представления и способов хранения знаний (учебного материала), стратегий активации и использования этих знаний;
- разработка формальных и когнитивных моделей приобретения знаний;
- формирование моделей поведения обучаемых (student modelling);
- создание новых стратегий обучения студентов и изучения учебного материала (teaching and learning strategies).

Анализ существующих обучающих систем показывает, что подавляющее большинство их является электронными учебниками, дополненными, в лучшем случае, системами тестового контроля знаний [47, 70, 108–109]. Помимо этого, есть небольшое количество обучающих

систем (например, [29, 80]), которые направляют процесс обучения не по заранее написанному сценарию, а используют в своей работе знания преподавателя (обычно, в виде экспертной системы) и обратную связь с обучаемым. Но эти системы написаны под конкретную предметную область.

К сожалению, новые научно-методические разработки слабо отражаются на реальных программных продуктах. Об этом можно судить по проходившей в Москве 6-9 апреля 2000 г. 2-й Международной выставке-конференции "Информационные технологии и телекоммуникации в образовании" [69]. В конференции приняли участие 16 институтов и университетов из разных регионов России, а также компании, производящие программное обеспечение, ориентированное на использование в учебном процессе. Анализ предложенных там продуктов и технологий (табл. 1.4) выявил следующую ситуацию:

- среди множества программных продуктов не было *ни одной* системы, которая осуществляла бы полную подготовку обучаемого хотя бы по одной дисциплине;
- все программные продукты либо освещают какой-то фрагмент ПО, либо выполняют одну (в лучшем случае – несколько) функций обучающей системы, либо предназначены для формирования отдельных умений и навыков решения задач;
- в большинстве программных продуктов (за исключением инструментальных средств КОНСПЕКТ, АССОЛЬ, ПА9) *не предусмотрена настройка системы на произвольную предметную область и требования преподавателя*, т.е. учебный материал включен в состав системы и не подлежит изменению, а последовательность его подачи запрограммирована;
- *ни одна из представленных обучающих программ не содержит модели обучаемого*, т.е. индивидуализация процесса обучения осуществляется только за счет индивидуальной работы обучаемого с программой;

Таблица 1.4. Компьютерные средства учебного назначения, представленные на 2-й Международной выставке-конференции "Информационные технологии и телекоммуникации в образовании"

Производитель	Тип, название	Модель предметной области	Модель учебного	Настройка на ПО	Демонстрация учебных материалов	Контроль знаний	Тип контроля знаний	Статистика
1	2	3	4	5	6	7	8	9
МГТУ им. Баумана	Лабораторный практикум по физике	нет	нет	нет	да	да	эталон ¹	нет
	Инструментальная система для создания тестов, DIACOM	нет	нет	нет	нет	да	тест	да
	Инструментальная система для создания обучающих программ, КОНСПЕКТ	нет	нет	да	да	нет	нет	нет
	Инструментальная система для мат. моделирования в САПР, ПА9	аппарат мат. моделирования	нет	да	нет	нет	нет	да
	Тренажер по математическому анализу	нет	нет	нет	да	нет	нет	нет
Министерство путей сообщения	Компьютерные обучающие программы по жел.-дор. делу	нет	нет	нет	да	да	тест	нет
Петрозаводский колледж жел.-дор. транспорта	Компьютерная обучающая программа по фотометрии	нет	нет	нет	да	да	тест	да

Таблица 1.4. Компьютерные средства учебного назначения, представленные на 2-й Международной выставке-конференции "Информационные технологии и телекоммуникации в образовании" (продолжение)

1	2	3	4	5	6	7	8	9
РГОТУПС	Электронный учебник по высшей математике	нет	нет	нет	да	нет	нет	нет
ОАО "Техно-парк Центр"	Система контроля знаний, АССОЛЬ	нет	нет	да	нет	да	да	да
Дагестанский государственный университет	Инструментальная система для создания тестов	нет	нет	да	нет	да	тест	да
	Электронные учебники	нет	нет	нет	да	да	тест	да
Тамбовский ГТУ	Экспертно-обучающая система "Управление динамическими объектами"	да	Нет	нет	да, с объяснениями	нет	нет	нет
Озерский технологический институт, г.Озерск Челябинской обл.	Лабораторный практикум по численному моделированию	нет	Нет	нет	да	да	тест	нет
Всероссийский заочный финансово-экономический институт	Компьютерные обучающие программы (7 дисциплин)	нет	Нет	нет	да	да	тест, самоконтроль	да

¹ – определение правильности ответа основано на сравнении с эталоном

- большинство из представленных обучающих программ *не имеет модели знаний по предметной области*, структура информации в них линейна или, в лучшем случае, определяется гиперссылками.

Современные обучающие системы, включая инструментальные системы для создания обучающих программ, страдают отсутствием абстрактности и носят явный отпечаток ориентации на определенные предметные области. Скорее всего, это обусловлено двумя основными факторами. Во-первых, отсутствует единая общепризнанная теория создания компьютерных обучающих систем. Во-вторых, сама задача обучения в самом широком смысле этого слова настолько сложна, что слабо поддается формализации и автоматизации. Поэтому каждая группа разработчиков, обычно обладающая знаниями в какой-либо определенной предметной области, пытаясь создать нечто универсальное, неизбежно приходит к необходимости сузить функциональные возможности системы и ограничить их решением наиболее проработанных и/или близких им задач.

Отчасти это может служить объяснением сложившейся в настоящее время ситуации, когда существующие обучающие системы не находят широкого применения и часто не выходят за рамки круга разработчиков. При создании АОС разработчики идут по пути выбора какой-либо определенной теории обучения и закладывают в систему алгоритм обучения (или правила формирования этого алгоритма). При этом они даже не всегда обосновывают выбор той или иной теории, а просто ставят пользователей перед фактом. И получается, что, несмотря на декларацию широких возможностей для настройки, существующие АОС не предоставляют возможности существенно изменить самое главное – правила поведения системы (стратегию обучения).

Таким образом, в настоящее время роль ЭВМ в обучающих системах ограничивается оптимизацией дозирования и последовательности подачи изучаемого материала, а также использованием различных форм его представления. С помощью обучающих программ изучается в основном тот материал, который в традиционных некомпьютерных технологиях обучения

осваивается в процессе лекционных занятий и семинаров. Но идея "приблизить работу обучающих программ к работе учителя" [31] вновь возрождается, причем на более высоком, чем прежде, уровне.

Выводы

В первой главе был проведен анализ тенденций развития и проблематики одной из актуальных областей современных информационных технологий – автоматизированных обучающих систем.

Показано, что современный этап развития АОС связан, прежде всего, с использованием достижений в области инженерии знаний и тех возможностей, которые предоставляются компьютерными сетями. Развитие Internet и телекоммуникационных технологий не только открывает новые возможности для получения образования, но позволяет обучающим системам выйти на качественно новый уровень предоставления образовательных услуг за счет объединения вычислительных возможностей компьютерных сетей.

Рассмотрены различные аппаратные и программные решения и современные сетевые технологии, применяемые для построения распределенных автоматизированных обучающих систем.

Приведены некоторые существующие классификации обучающих систем и показана их неоднозначность. Предложен принцип многомерной классификации систем по набору классифицирующих признаков, а также набор этих признаков и их значений для классификации обучающих систем.

Проведенный анализ некоторых существующих обучающих систем и инструментальных средств для их создания показал, что многие обучающие системы либо вообще не имеют канала обратной связи, либо не обладают возможностями адаптации процесса обучения к уровню знаний и умений обучаемых. Не в последнюю очередь это связано с перекосом в сторону использования в обучающих системах гипертекста и мультимедиа в ущерб содержанию и методологии обучения.

Кроме того, большинство существующих обучающих систем с удаленным доступом не являются распределенными, т.к. не позволяют распределять функции обучающей системы между узлами сети.

Было отмечено, что все составные части АОС в той или иной форме реализованы хотя бы в одной из рассмотренных систем. Но нет системы, которая соответствовала бы всем требованиям, предъявляемым к АОС.

По результатам анализа можно сделать вывод о необходимости разработки методов взаимодействия компонентов РАОС, позволяющих выполнять распределенные вычисления, настраивать систему на различные методики обучения и интегрировать в систему существующие программные средства учебного назначения. Обоснована необходимость разработки способов и алгоритмов организации контроля знаний, основанных на формальных методах анализа и дифференцированной оценки ответов обучаемых.

2. ПРИНЦИПЫ СОЗДАНИЯ РАСПРЕДЕЛЕННЫХ АВТОМАТИЗИРОВАННЫХ ОБУЧАЮЩИХ СИСТЕМ

Распределенная автоматизированная обучающая система должна обладать следующими свойствами.

С одной стороны, быть как можно ближе к традиционному индивидуальному обучению с преподавателем. Для этого система должна поддерживать обратную связь с обучаемым, иметь способность адаптироваться к его уровню и потребностям, объективно оценивать его знания. С другой стороны, такая система должна включать те возможности, которые предоставляет вычислительная техника: различные формы представления информации, моделирование процессов и явлений, использование электронных справочников, доступ к большим объемам информации.

2.1. Организация РАОС

2.1.1. Создание сетевой среды РАОС

Сетевая среда распределенной обучающей системы включает в себя локальную вычислительную сеть и глобальную (Internet).

Рассмотренные в Главе 1 сети ISDN способны во многом решить проблемы доступа в сеть Internet [36]. Выделяют три варианта подключения отдельных компьютеров и ЛВС к Internet. Для доступа в Internet отдельных пользователей можно применять ISDN BRI-адаптеры, которые устанавливаются в стандартное гнездо шины персонального компьютера (ISA, PCI или PC-Card). Для связи с провайдером обычно используется Point-to-Point Protocol (PPP), а для аутентификации входящих в сеть пользователей – протоколы PAP и CHAP.

Кроме внутренних адаптеров, существуют внешние терминальные адаптеры или внешний ISDN-модем, которые предназначены для

конвертации последовательного интерфейса ПК, обычного моста/маршрутизатора или другого не ISDN-устройства, в формат ISDN BRI.

Для сетей ISDN возможно применение активных или пассивных адаптеров. Активный адаптер построен на основе процессора со своей оперативной памятью и ориентирован на выполнение коммуникационного программного обеспечения. Он позволяет значительно меньше использовать ресурсы ЦП файлового сервера. Пассивный адаптер ISDN аналогичен обычному сетевому адаптеру и использует ресурсы ЦП сервера. Обычно активные адаптеры дороже пассивных, но и более производительны.

Для подключения ЛВС к Internet обычно используется маршрутизатор, позволяющий разделять внутреннюю и внешнюю IP-сети и осуществляющий функции брандмауэра. Сами маршрутизаторы могут быть реализованы программным путем на серверах NetWare (IntranetWare), Windows NT или UNIX. Необходимо также программное обеспечение типа NetWare Multiprotocol Router for ISDN, либо аналогичное для Windows NT или UNIX.

Другим вариантом является использование аппаратного маршрутизатора, который выполнен в виде отдельного устройства и имеет один или несколько портов для подключения ЛВС и один или несколько WAN-портов. Производительность и широта функциональных возможностей (в частности, поддержка разнообразных протоколов) зависят от стоимости маршрутизатора.

Теперь перейдем к рассмотрению особенностей функционирования программного обеспечения РАОС.

2.1.2. Особенности функционирования программного обеспечения РАОС

Функциональная схема РАОС приведена на рис. 2.1.

Здесь используются следующие сокращения:

БПО – базовое программное обеспечение;

ППО – прикладное программное обеспечение;

ЛППО – локальное прикладное программное обеспечение;

ЛКПО – локальное коммуникационное программное обеспечение;

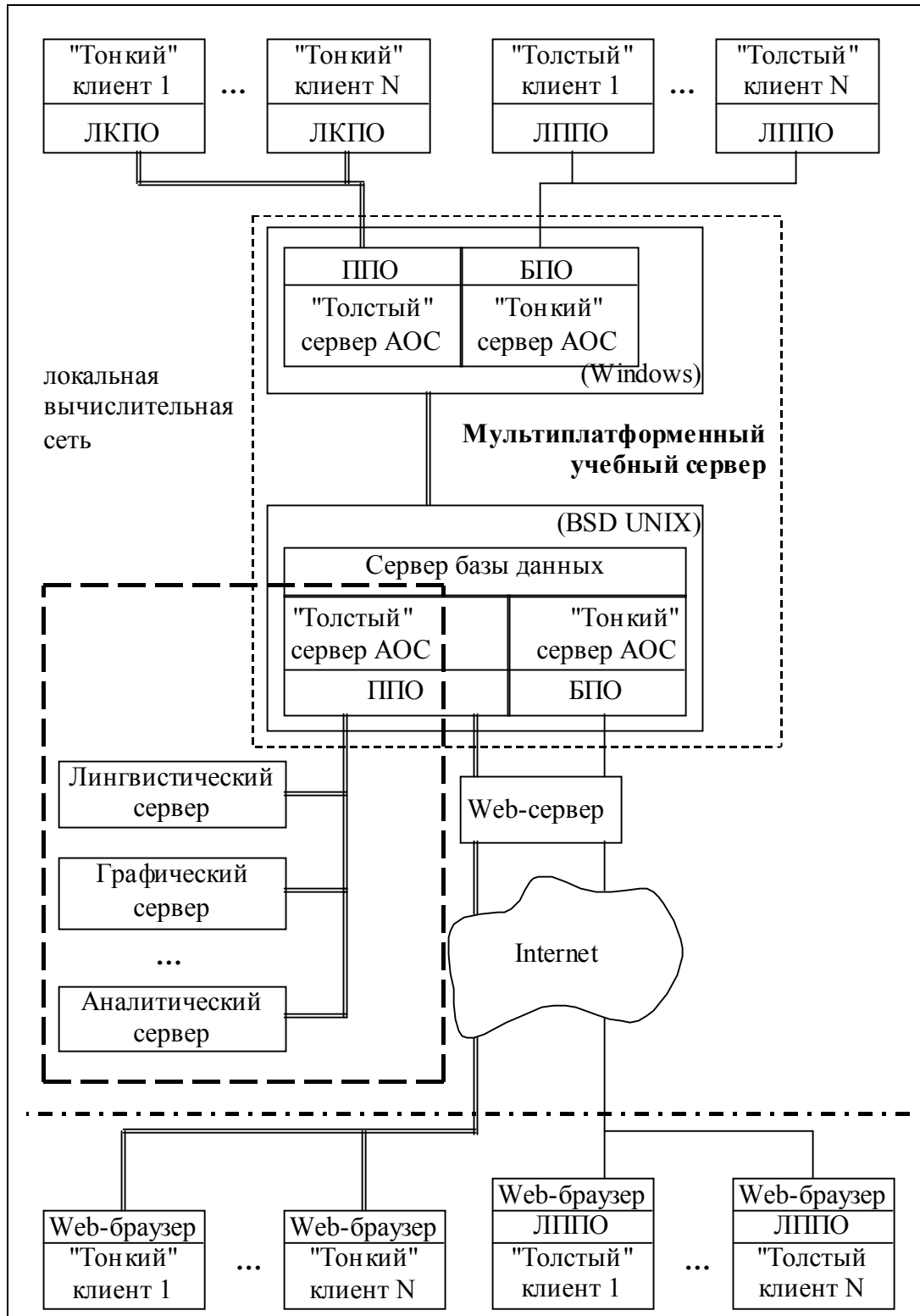


Рис. 2.1. Функциональная схема РАОС

Центральное место в этой структуре занимает учебный сервер, который управляет работой всей системы в целом. В состав учебного сервера

входит ядро АОС и сервер базы данных (БД). Состав и функции ядра зависят от технологии организации работы режима "клиент–сервер".

Возможны различные подходы к распределению функций между клиентами и сервером. Размещение прикладного программного обеспечения на клиенте позволяет сократить до минимума функциональность сервера ("толстый" клиент – "тонкий" сервер). При этом на машине клиента располагается локальное ППО, в т.ч. учебные материалы и вспомогательное программное обеспечение: всевозможные моделирующие пакеты, лабораторные практикумы, тренажеры и т.п. А на сервере располагается базовое программное обеспечение.

Основное преимущество подхода "толстый" клиент – "тонкий" сервер заключается в минимизации сетевого трафика: обмен по сети ограничивается передачей сообщений и обращениями к БД. Кроме того, снижаются требования к пропускной способности канала связи.

При переносе на сервер прикладного программного обеспечения (и учебных материалов) реализуется технология "тонкий" клиент – "толстый" сервер. На клиенте размещается только локальное коммуникационное программное обеспечение. Это значительно увеличивает сетевой трафик, зато избавляет пользователя от необходимости хранить на своем компьютере прикладные программы и большие объемы данных, а также упрощает администрирование программного обеспечения.

И та, и другая технология могут быть реализованы как в локальной сети, так и в сети Internet. Для ЛВС наиболее распространенным решением является использование сервера под управлением ОС Windows. В пользу этого подхода можно привести такие соображения, как простота организации сервера; стандартный интерфейс; естественная интеграция приложений; высокая скорость обмена данными; возможность распределения программного обеспечения между узлами сети; прозрачная файловая система.

Для работы через Internet более целесообразным с точки зрения надежности является использование сервера под управлением ОС UNIX. Сервер БД также следует разместить на UNIX-машине. В БД хранится информация о пользователях системы, поэтому она должна быть хорошо защищена от несанкционированного доступа. Информация включает в себя паспорт пользователя (его имя, пароль, статус, права, которыми он обладает в системе), данные о текущем состоянии процесса обучения и некоторые другие сведения.

Кроме рассмотренных выше, в состав обучающей системы могут входить различные вспомогательные модули – дополнительные программы (программные комплексы), которые предназначены или адаптированы для выполнения отдельных функций, поддерживающих процесс обучения. Их набор может меняться в зависимости от требований к обучающей системе. В совокупности с ядром АОС они обеспечивают выполнение всех внутренних функций системы.

В качестве примеров вспомогательных средств можно привести:

- лингвистический сервер – программный комплекс, предназначенный для распознавания текстов на естественном языке. В АОС его можно использовать для анализа вопросов и ответов обучаемого, для синтеза ответов системы с целью поддержки диалога между системой и пользователем.
- графический сервер – программный комплекс, предназначенный для распознавания графических изображений при анализе ответов обучаемого.

Для работы этих (и подобных им) комплексов требуется много системных ресурсов – и процессорного времени, и дисковой памяти. Кроме того, обычно такое программное обеспечение, в отличие от АОС, ориентировано на определенную предметную область. Поэтому эти программные комплексы нецелесообразно включать в состав ядра АОС и размещать на том же узле сети.

С другой стороны, эти вспомогательные программные средства берут на себя часть функций АОС, поэтому доступ к этим средствам (к узлам сети) критичен для системы в целом. Повышения надежности и работоспособности системы можно достигнуть, используя специальный протокол обмена данными, который обеспечивает автоматическую реконфигурацию системы вспомогательного программного обеспечения.

Задача динамической реконфигурации и маршрутизации в неоднородных вычислительных средах, актуальная, прежде всего, в САУ, рассмотрена, например, в [38]. Но применительно к данному случаю можно рассмотреть более простой вариант протокола обмена данными с реконфигурацией системы. (На рис. 2.1. составляющие элементы этой системы обведены пунктирной линией — — —).

2.1.3. Описание протокола обмена данными между ядром АОС и вспомогательным программным обеспечением

Перед протоколом обмена данными стоят задачи организации распределенных вычислений и автоматической реконфигурации системы с целью обеспечения надежности ее функционирования. Это протокол уровня приложения, основанный на использовании ТСР/ІР.

Конфигурация сети определяется следующей информацией:

- топология сети;
- процессы, которые могут быть запущены на каждом узле сети.

Реализация протокола должна обеспечиваться минимально возможными средствами. Для этого требуются процессы-демоны: главный демон на сервере АОС и вспомогательные демоны на тех узлах, где расположено вспомогательное программное обеспечение (ВПО). Демоны запускаются оператором вручную.

При запуске главный демон опрашивает узлы сети для установления их доступности и определяет те узлы, на которых будут запущены процессы ВПО. Критерием распределения процессов является максимальное

распараллеливание работы при условии, что каждый процесс запускается на одном узле. Потом главный демон направляет вспомогательным демонам выбранных узлов команды на запуск соответствующих процессов. При обращении ядра АОС к вспомогательному процессу главный демон направляет запрос соответствующему вспомогательному демону и ожидает от него ответа, который возвращает ядру АОС. Если в течение определенного периода (time-out) ответ не получен или получено сообщение об ошибке, то главный демон производит реконфигурацию системы и повторные попытки обработать запрос. Кроме того, главный демон может периодически опрашивать узлы сети для подтверждения их доступности.

Вспомогательные демоны сразу после запуска переходят в состоянии ожидания запроса от главного демона. При запросе на подтверждение готовности вспомогательный демон отправляет отклик главному демону. Обработка запроса на инициацию процесса заключается в том, что вспомогательный демон запускает процесс и отправляет главному демону сообщение об успехе (или неуспехе) запуска. При запросе на передачу данных вспомогательный демон передает данные запущенному процессу и ожидает результата, который отправляет обратно главному демону.

При такой организации работы система будет работоспособна, пока для каждого вспомогательного процесса доступен хотя бы один узел, на котором он может быть запущен.

Подробнее протокол обмена данными между ядром РАОС и вспомогательным программным обеспечением рассмотрен в Главе 4.

2.2. Основные характеристики РАОС как программного продукта

Итак, РАОС – это функционирующий в вычислительной сети программный комплекс, предназначенный для формирования и эксплуатации обучающей системы и позволяющий преподавателю заложить в систему *свои* представления о методологии преподавания и *свои* предметные знания.

Примечание: предметом данного исследования являются только самые общие свойства, которыми система должна обладать, и возможности, которые система должна предоставлять преподавателю при создании АОС. Теоретическое или эмпирическое обоснование правомерности того или иного педагогического или методологического решения должно выполняться создателем конкретного курса.

К основным свойствам, которые должны характеризовать РАОС как программный продукт, можно отнести следующие:

1. Полнофункциональность системы. Инструментарий РАОС должен обеспечивать возможность создания системы, которая поддерживает различные формы организации обучения в течение всего цикла обучения (от определения начального уровня знаний обучаемого и его потребностей до итогового контроля знаний).
2. Открытость системы, т.е. предоставление пользователю возможности настраивать систему и расширять ее путем подключения дополнительных исполняемых модулей или замены существующих. Настройка системы должна быть параметрической и алгоритмической.
3. Работа в сети. Подразумевает не только предоставление возможности удаленного доступа к системе, например, в рамках дистанционного образования, но и активное использование вычислительных возможностей, предоставляемых сетью. С учетом требования открытости это означает возможность создания на основе РАОС гетерогенной обучающей системы.

2.3. Пользователи АОС

АОС предназначена для организации автоматизированного обучения по произвольной дисциплине (курсу). Все множество пользователей АОС можно разбить на три группы:

1. Составитель курса (преподаватель).

Он осуществляет подготовку системы к использованию, в его задачи входит:

- заполнение системы знаниями по предметной области, учебными материалами, вспомогательными компьютерными программами и т.д.;
- настройка системы (в рамках предоставленных возможностей) в соответствии с целями использования системы и представлениями преподавателя об эффективных методах управления обучением.

2. Консультант (преподаватель).

Он контролирует работу обучаемых с системой; в его задачи входит:

- регистрация обучаемого, настройка системы на данного обучаемого, определение цели его работы с АОС;
- периодический контроль за ходом процесса обучения;
- принятие решения об окончании обучения с учетом рекомендаций системы.

3. Обучаемый.

Работа системы с обучаемым делится на сеансы. Во время каждого сеанса система:

- определяет текущее состояние процесса обучения;
- выполняет действия, предусмотренные текущим состоянием (например, демонстрацию учебных материалов, запуск внешних модулей, контроль знаний);
- по результатам ответных действий обучаемого изменяет текущее состояние процесса обучения;
- осуществляет проверку достижения заданной цели. Если цель достигнута, система сообщает об этом обучаемому и консультанту; если не достигнута – просто завершает сеанс.

Последовательность работы АОС с обучаемым отражена на структурной диаграмме (рис. 2.2) и состоит из следующих шагов:

1. Идентификация обучаемого преследует две цели:

- Предотвращение несанкционированного доступа к системе.

Предлагается использовать парольную идентификацию как наиболее распространенную и простую в реализации.

- Настройка системы на конкретного обучаемого.

Во время первого сеанса работы системы с обучаемым настройка заключается в определении исходного состояния объекта управления, а в дальнейшем – в использовании модели обучаемого (объекта управления), сформированной в результате предыдущих сеансов.

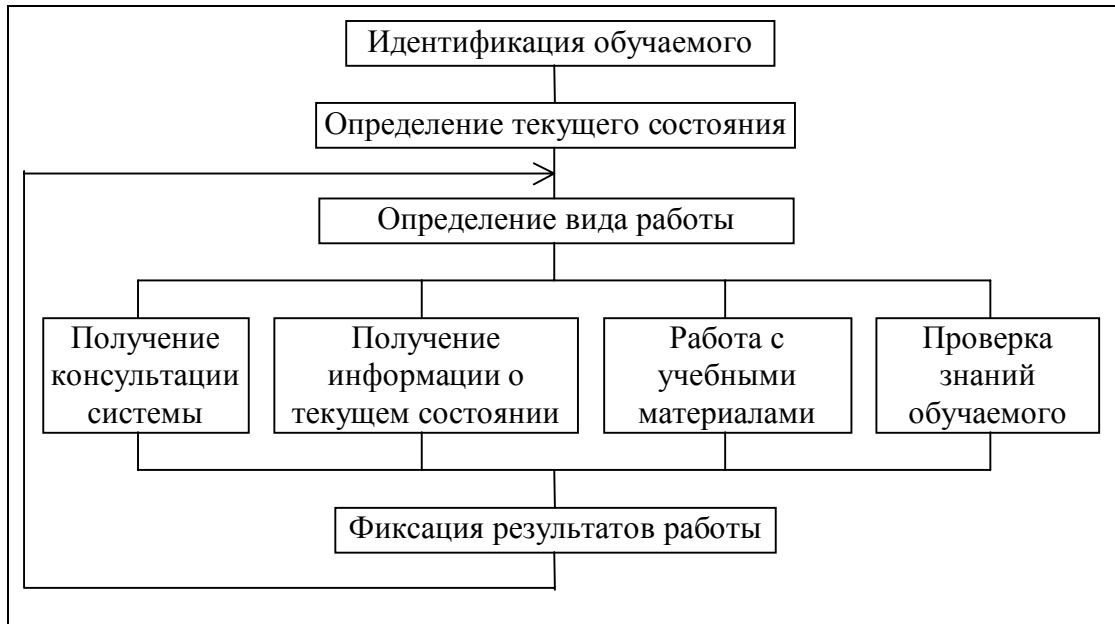


Рис. 2.2. Структурная диаграмма работы обучающей системы

2. Определение текущего состояния процесса обучения заключается в анализе статистических данных, полученных при работе с обучаемым, определении степени достижения цели и выборе стратегии поведения.
3. Определение вида работы происходит в диалоге с обучаемым. В общем случае у обучаемого может быть несколько вариантов начала (продолжения) обучения, и обучаемый выбирает из них самостоятельно. Если он хочет воспользоваться помощью системы, он может получить консультацию. В ходе консультации система выдает рекомендации относительно дальнейших действий обучаемого и/или последовательности изучения курса в целом. Альтернативой консультации является получение информации о текущем состоянии процесса обучения.

Для работы с учебными материалами обучаемый выбирает тему, которую он будет изучать, после чего ему выдаются соответствующие материалы. При этом возможна временная передача управления другим программам. При проверке знаний обучаемого система сначала определяет стратегию проведения контроля знаний. Выбор стратегии определяется преподавателем или происходит на основе модели обучаемого. После тестирования выполняется анализ результатов и изменение модели обучаемого.

4. По окончании выполнения каждого вида работы система фиксирует результаты в базе данных обучаемых. После этого обучаемый может приступить к другой работе или завершить сеанс.

2.4. Логическая структура обучающей системы

В Главе 1 было определено, что для реализации всех функций обучающая система должна содержать общие и специальные знания трех видов [30, 145]:

1. о предметной области (модель предметной области, a model of the subject).
Модель предметной области должна отражать структуру ПО. Она может быть использована при определении последовательности изучения тем и проведения контрольных занятий [25].
2. об учащемся (модель обучаемого, a model of the specialist).
Модель обучаемого включает динамически обновляемый набор параметров, отражающих общие характеристики обучаемого, и проекцию его знаний на знания системы (на модель предметной области).
3. о стратегии обучения.
Стратегия обучения определяет последовательность изучения тем (на основе модели предметной области) и формирует задания для контроля знаний обучаемого (на основе модели обучаемого).

На основании вышеизложенного можно предложить логическую структуру обучающей системы (рис. 2.3).

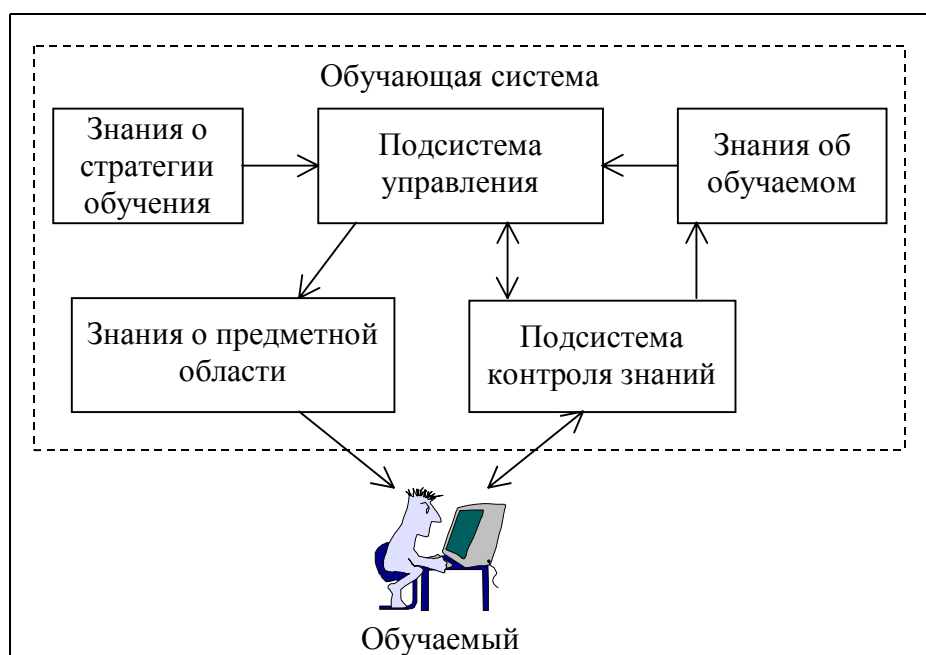


Рис.2.3. Логическая структура обучающей системы

К сожалению, в настоящее время отсутствуют реально используемые в учебном процессе системы, содержащие все упомянутые выше модели (предметной области, стратегии обучения и обучаемого) [30], а без них невозможна полноценная адаптация системы к обучаемому.

Рассмотрим подробнее принципы организации и взаимодействия этих моделей.

2.4.1. Взаимодействие с обучаемым

Выделяют три подхода к организации управления взаимодействием с обучаемым в АОС [74]:

1. либеральный;
2. консервативный;
3. адаптивный.

В первом случае управление со стороны системы отсутствует, обучаемый самостоятельно выбирает последовательность изучения материала. Единственной подсказкой здесь может служить оглавление курса, задающее целесообразный порядок изучения предмета. Такая организация подобна выдаче обучаемому списка рекомендованной литературы.

При консервативном подходе последовательность подачи учебных материалов жестко запрограммирована (и, обычно, дозирована). Эта последовательность описывается в *сценарии курса*, выполнение которого управляется программно. При программированном обучении [4] обучающая система получается негибкой: она **одинаково** управляет **разными** пользователями, и теряется одно из преимуществ использования АОС – индивидуальность обучения. Введение же альтернативных сценариев, во-первых, сопряжено с большими дополнительными расходами времени и сил разработчика, а, во-вторых, не решает проблему полностью, т.к. по отдельности каждый альтернативный сценарий все равно остается запрограммирован.

Наиболее эффективным способом организации управления обучением является адаптивный подход. Система подстраивается под обучаемого, устанавливая очередность и интенсивность изучения материалов на основании достигнутых обучаемым результатов.

При разработке обучающих программ нельзя идти по пути применения какого-либо одного метода управления взаимодействием с обучаемым. Необходимо предусмотреть различные методы управления, как по инициативе системы, так и по инициативе обучаемого [98].

Система, претендующая на то, чтобы называться автоматизированной системой обучения, должна предоставлять преподавателю возможность использовать любой из этих трех вариантов. Очевидно, что наибольшие трудности для реализации представляет адаптивный вариант, а либеральный и консервативный способы организации управления обучением могут считаться частным случаем адаптивного.

2.4.2. Подсистема управления обучением

Формирование управляющих воздействий на обучаемого (стратегия управления обучением) обычно определяется алгоритмом функционирования системы. Но принцип открытости АОС подразумевает возможность

алгоритмической настройки системы (без необходимости программирования), поэтому алгоритмы управления обучением нельзя программировать.

Для того чтобы обеспечить алгоритмическую настройку системы, можно воспользоваться механизмом вывода, реализованным в системах, основанных на знаниях, и вынести за рамки программного кода правила управления обучением. Эти правила (база знаний) должны определять последовательность работы системы на основе модели предметной области и модели обучаемого.

При разработке подсистемы управления очень важно правильно выбрать форму представления знаний. Это связано с тем, что представление знаний в конечном итоге определяет характеристики системы [83]. Выбор оптимального способа представления знаний зависит от характера и сложности решаемой задачи.

В подсистему управления знания закладываются преподавателем – разработчиком конкретной АОС. Эта подсистема должна соответствовать следующим требованиям:

1. легкость пополнения и модификации базы знаний (БЗ);
2. естественность представления знаний (правила должны отражать способ оформления экспертами собственной эвристики для решения проблемы);
3. простота создания и понимания отдельных правил БЗ;
4. возможность получения четких пояснений действий системы.

Из четырех основных моделей представления знаний этим требованиям наиболее полно удовлетворяет продукционная модель [66].

На основании правил этой базы знаний обучающая система будет определять последовательность изучения материала и степень свободы обучаемого в управлении этой последовательностью.

При моделировании консервативного подхода к организации обучения система сама определяет последовательность прохождения тем курса в соответствии со связями, отраженными в модели предметной области,

"разрешая" переходить к очередной теме только после успешного изучения предшествующих тем.

При моделировании либерального подхода обучаемый сам определяет последовательность изучения материала, а система лишь может выдавать ему рекомендации и консультировать относительно возможных причин пробелов в знания, которые выявляются в ходе контроля знаний.

При моделировании адаптивного управления обучением могут использоваться разные подходы. Например, устанавливаются некоторые контрольные точки, и обучаемый может выбирать: начинать ему сразу сдавать эти контрольные точки или сначала "почитать теорию". Пока он сдает эти контрольные точки успешно, система не вмешивается; но если контрольная точка не сдана, система берет управление на себя и не позволяет обучаемому двигаться дальше, пока он не изучит соответствующий этой контрольной точке учебный материал, и не позволяет сдавать контрольные точки в этот же день.

2.4.3. Модель предметной области (МПО)

В автоматизированном обучении модель предметной области приобретает особую роль, т.к. качество обучения практически определяется алгоритмом управления процессом обучения, который базируется на МПО. Модель предметной области (учебного курса) можно использовать как для управления процессом обучения, так и для обоснованного решения вопроса о включении тех или иных фрагментов знаний в программу учебного курса. Она упрощает разработку и понимание учебных программ, особенно типовых и базовых, а также проведение экспертизы их качества [72].

Предметная область (ПО) характеризуется сущностями и связями между ними. В качестве сущностей ПО в обучающих системах можно рассматривать понятия или темы, каждой из которых соответствует единица учебного материала, не требующая (с точки зрения преподавателя) деления

на подтемы. Каждая тема описывается набором параметров (атрибутов), существенных для управления обучением.

Связь между темами i и j подразумевает зависимость между ними, которую можно интерпретировать так: для понимания темы j нужно знать тему i . Таким образом, связи могут определять последовательность изучения тем. Связи между темами могут быть факультативными и обязательными. В случае факультативной связи последовательность изучения тем носит рекомендательный характер.

Связи могут иметь также различную семантику и взаимозависимости. В частности, структура ПО не может содержать циклов. В остальном никаких ограничений на структуру ПО не накладывается.

Назовем *курсом* совокупность предметных знаний, внесенных в АОС на этапе заполнения и предназначенных для обучения по определенному предмету (специальности).

В большинстве существующих обучающих программ учебный материал имеет линейную структуру, соответствующую последовательности изложения материала. Такой подход не может считаться оптимальным, т.к. в общем случае учебный материал "...может не иметь заранее определенного наилучшего пути следования, поэтому, подчиняя процесс научения некоторой однонаправленной форме, мы тем самым можем дать неверное представление об истинности тех или иных явлений или фактов" [103].

Рассмотрим возможный вариант организации модели предметной области произвольной структуры. Для этого сначала определим требования, которым она должна удовлетворять.

2.4.3.1. Требования, предъявляемые к МПО

На основании вышеизложенного понимания структуры предметной области ее модель должна удовлетворять следующим требованиям:

1. возможность отражать различные типы связей между элементами;
2. возможность получения целостного образа знаний;
3. возможность объединения процедурных и декларативных знаний.

Этим требованиям удовлетворяют фреймовая и семантическая модели представления знаний [85]. Семантические сети и фреймовые модели близки друг другу, но механизм вывода в семантических сетях более прозрачен. Отчасти поэтому семантические сети можно назвать традиционным способом организации модели предметной области [16, 63].

Преставление этой модели в виде семантической сети позволяет проводить анализ предметной области по таким параметрам, как:

- связность (достижимость любой вершины);
- наличие циклов;
- степень важности определенной темы (понятия), определяемая количеством тем, зависящих от данной темы.

Семантическая сеть в работах, затрагивающих вопрос организации модели предметной области, рассматривается в классическом понимании, т.е. как направленный граф с помеченными вершинами и дугами, в котором вершинам соответствуют объекты, а дугам – семантические отношения между ними [61]. В некоторых работах для этой модели предлагаются наборы семантических взаимоотношений. Например, в работе [63] выделяются связи типа "является частью", "следует из" и другие.

Не умаляя практическую ценность работы по формированию полного набора взаимоотношений объектов (понятий) предметной области, отметим чрезвычайную трудоемкость этой задачи и невысокую эффективность применения такого подхода. Как и в большинстве реальных случаев, в обучающих системах используется очень небольшое количество типов отношений, поэтому существенная часть полного набора типов отношений никогда не найдет своего применения. Но, предоставляя пользователю ограниченный набор типов отношений, нельзя говорить об универсальности системы. Всегда найдется такая связь между объектами, которую нельзя будет адекватно отразить ограниченным набором.

Поэтому в данной работе предлагается подход, при котором типы семантических связей не выделяются, а интерпретации связей выполняется

процессами, относящимися к соответствующим вершинам. Кроме возможности не ограничивать пользователя определенными типами связей, такой метод позволяет ввести в модель предметной области прагматику, т.е. учет цели использования модели.

2.4.3.2. Семиотическая сеть

Семантическая модель базируется на семантической сети [59]. Это система знаний, имеющая определенный смысл в виде целостного образа сети. Механизм вывода заключается в распространении по сети возбуждения в зависимости от топологии сети и входных данных.

Определим семантическую сеть Ω как двойку вида

$$\Omega = \{V, D\},$$

где $V = \{v_i\}$ – множество вершин (узлов сети), а $D = \{d_j\}$ – множество дуг. Вершина v_i семантической сети может быть определена как

$$v_i = \{S, c\},$$

где $S = \{s_k\}$ – множество точек входа в вершину (синапсов, рис.2.4), а c – функция, определяющая состояние вершины:

$$c = \vee f_n,$$

т.е. дизъюнкция состояний множества синапсов S , относящихся к вершине v_i .

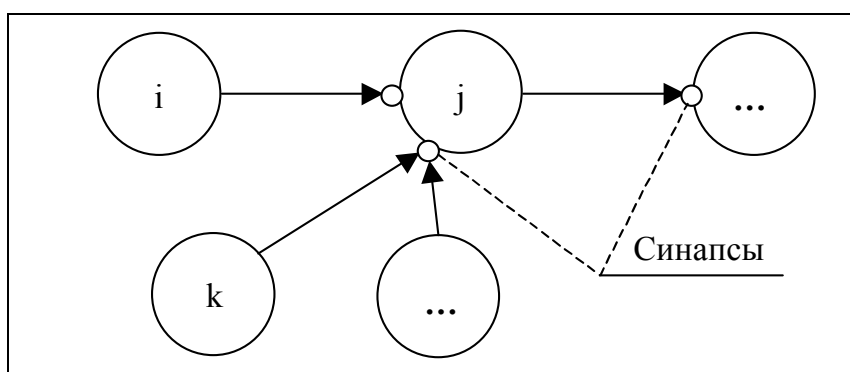


Рис.2.4. Фрагмент семантической сети

Синапс s_k вершины v_i есть пара вида:

$$s_k = (p_k, f_k),$$

где p_k – процесс, связанный с данным синапсом, а функция f_k определяется как произвольная логическая функция от состояний вершин v_m , дуги от которых входят в синапс s_k , например, конъюнкция:

$$f_k = \bigwedge v_m.$$

Узлами сети могут быть любые элементы процесса обучения: теоретический материал по определенной теме курса, лабораторные и контрольные работы, задания для самоконтроля и т.п. С программной точки зрения каждая вершина представляет собой некоторый процесс. Дуги между элементами определяют взаимосвязи между вершинами и задают последовательность изучения курса.

Каждый синапс может находиться в двух состояниях: возбужденном и невозбужденном. Если какой-либо из синапсов возбужден, то возбуждается соответствующая вершина и это возбуждение передается по всем дугам, выходящим из нее к другим синапсам. Синапс же переходит в возбужденное состояние только тогда, когда возбуждены все вершины, дуги от которых идут к данному синапсу. Изначально в семантической сети считаются возбужденными все вершины, соответствующие не входящим в курс темам.

С синапсом может быть связана произвольная процедура, и тогда возбуждение синапса будет вызывать выполнение этой процедуры. Она определяет зависимость использования вершины от пути, по которому пришло возбуждение в эту вершину. Это позволяет ввести в модель ПО прагматику, т.е. учет цели, с которой обучаемый работает с системой, например, ознакомление, изучение всего курса, проверка знаний по выбранным темам и т.д. Таким образом, мы перешли к *семиотической сети* – модели, обладающей синтаксисом (определенным способом выражения), семантикой (содержанием) и прагматикой (способом использования) [82].

На основании такой структуры каждому обучаемому можно задавать свое подмножество изучаемых тем (выполняемых работ), которое будет определяться, например, списками начальных и конечных вершин сети. В качестве начальной вершины может выступать вершина, которая имеет хотя

бы одну исходящую дугу, в качестве конечной – вершина, в которую можно попасть из заданных начальных вершин. Изучение курса заключается в прохождении по всем вершинам, входящим в маршруты от начальных до конечных вершин.

Естественно, изучаемая предметная область неразрывно связана с другими областями знаний [94]. Изучение некоторых из этих областей по необходимости должно предшествовать изучению данной ПО (например, владение математическим аппаратом и знания по физике требуется при изучении большинства технических дисциплин, т.к. составляет методы описания и исследования объектов ПО, а также их естественнонаучную базу). Таким образом, для определения места изучаемой ПО в системе знаний и тех областей (методов, понятий), которые непосредственно предшествуют и подводят к изучению данной ПО, в модель предметной области необходимо включать темы из других курсов, знание которых требуется для понимания изучаемой ПО. Наличие таких связей позволит системе определять возможные пробелы в знаниях обучаемого и выдавать ему рекомендации по изучению смежных дисциплин.

2.4.3.3. Анализ модели предметной области

Организация МПО в виде семиотической сети, в основе которой лежит оргграф, позволяет использовать для анализа МПО аппарат теории графов [57].

Пусть G – оргграф, описывающий МПО, $VG = \{v_i\}$ – множество вершин и $VE = \{e_{ij}\}$ – множество дуг этого оргграфа. Оргграф G не должен содержать:

1. петель, т.е. дуг (v_i, v_i) .
2. циклов, т.е. таких маршрутов $v_1, e_1, v_2, e_2, \dots, e_k, v_{k+1}$, в которых $v_1 = v_{k+1}$ (где v_i, e_i – соответственно номера вершин и дуг, входящих в маршрут).
3. несвязных вершин или подграфов. Для определения наличия несвязных вершин (подграфов) оргграф рассматривается как неориентированный граф. Неориентированный граф является связным тогда и только тогда,

когда для произвольной фиксированной вершины v существует маршрут (v, \dots, u) , где u – любая другая вершина графа.

Кроме анализа МПО использование графов дает возможность решить оптимизационную задачу, связанную с определением целесообразной последовательности изучения тем как в рамках курса, так и нескольких взаимосвязанных курсов (дисциплин). МПО отражает логические связи между темами (понятиями), которые относятся, возможно, к разным дисциплинам. Соответственно, последовательность освоения тем должна быть такова, чтобы к началу изучения темы i все предшествующие ей темы (понятия) были уже изучены.

Эта задача сводится к задаче раскраски вершин графа. Напомним, что раскраской графа G называется произвольная функция вида

$$f: VG \rightarrow \{1, 2, \dots, k\},$$

где k – количество различных красок. В данном случае решением задачи определения последовательности изучения тем является такая раскраска орграфа G , при которой для любого маршрута $v_1, e_1, v_2, e_2, \dots, e_k, v_{k+1}$, вершины которого раскрашены цветами l_1, l_2, \dots, l_k , верно утверждение $l_i < l_j$, если $i < j$.

Внесение в МПО вершин, относящихся к смежным курсам, позволит определить не только последовательность изучения одного курса, но и порядок изучения разных (взаимосвязанных) курсов. (Задача построения учебного плана по модели предметной области подробно рассмотрена многими исследователями, в частности, можно упомянуть работы [73, 84]).

2.4.3.4. Организация модели предметной области (МПО)

Модель предметной области предназначена для решения двух взаимосвязанных задач. Эта модель отражает связи между элементами учебного процесса и, следовательно, определяет последовательность изучения материала (административная задача), а также возможные причины ошибок обучаемого, возникших из-за незнания предшествующего материала (задача консультации и адаптивного управления).

Возможна реализация работы с семиотической сетью через программный модуль на языке высокого уровня. Описание этой сети может задаваться в текстовом файле определенного формата, что соответствовало бы принципам открытости и переносимости. Но зададимся вопросом: нужна ли переносимость в данном случае? Неизменное содержание обучающей системы составляют учебные материалы и тесты. Рассчитывать на то, что какая-либо другая оболочка обучающей системы позволит использовать сложно организованное описание МПО в его изначальном виде, было бы наивно. Значит, от принципа переносимости для описания МПО можно безболезненно отказаться.

С другой стороны, знания о структуре ПО применяются в правилах базы знаний подсистемы управления, организованной с помощью продукционной модели (набора правил). Если организовать МПО также с использованием продукционной модели, можно легко решить задачу сопряжения этих двух подсистем. Кроме того, представление МПО в виде набора правил имеет еще одно преимущество: возможность модификации этого набора позволяет изменять саму процедуру работы с сетью (распространение возбуждения по сети), т.е. осуществлять алгоритмическую настройку МПО.

Но для организации управления процессом обучения недостаточно иметь модель предметной области. Система должна иметь информацию о том, в каком состоянии находится процесс обучения конкретного обучаемого. Для учета этих сведений вводится *модель обучаемого*.

2.4.4. Модель обучаемого

В настоящее время не существует общепринятого определения понятия "модель обучаемого". Но можно выделить два основных подхода к построению такой модели.

1. В экспертно-обучающих системах (ЭОС) под моделью пользователя понимают набор характеристик (параметров) и совокупность правил,

которые на основании значений этих характеристик управляют процессом общения системы с пользователем [85].

2. В других классах обучающих систем под моделью пользователя обычно понимают набор параметров, измеряемых во время работы системы с обучаемым и определяющей степень усвоения им знаний по изучаемому предмету.

На самом деле, понимание модели обучаемого как набора параметров – это слишком узкое понимание. Без учета методов, которые работают с данным набором характеристик, этот набор теряет смысл. Следовательно, модель обучаемого (МО) можно определить как совокупность набора характеристик обучаемого и методов (правил) обработки этого набора.

В ЭОС эти правила привязаны к конкретной ПО, что позволяет, естественно, более качественно имитировать общение обучаемого со специалистом в данной предметной области. Но АОС не может быть ориентирована на конкретную ПО, поэтому и назначение правил, входящих в модель обучаемого, должно быть другим. В первую очередь эти правила должны проводить изменения самой модели обучаемого по результатам его работы с системой. Это позволит преподавателю управлять формированием этой модели, т.е. осуществлять алгоритмическую настройку без программирования.

Модель обучаемого должна включать в себя информацию:

- a) о цели обучения;
- b) о знаниях обучаемого в рамках изучаемого курса (текущее состояние процесса обучения);
- c) об особенностях подачи учебных материалов и выбора контрольных заданий и вопросов;
- d) о правилах изменения модели обучаемого по результатам работы с обучаемым.

(Пункт c) тесно соотносится со знаниями о стратегии обучения).

Для каждого обучаемого может быть задана своя цель работы с системой и свое подмножество изучаемого материала, которое определяет начальную настройку системы и является базой для дальнейшей работы с обучаемым.

Текущее состояние процесса обучения фактически представляет собой проекцию знаний обучаемого на модель предметной области. Проекция ограничена рамками заданных обучаемому маршрутов и включает сведения о результатах изучения отдельных тем курса (прохождения вершин сети). Результаты изучения отдельных тем могут быть представлены как совокупность результатов контроля знаний по данной теме (если он предусмотрен) и набор значений параметров, которые могут быть измерены во время работы с обучаемым (количество обращений к вершине (теме), время работы с материалом, среднее время ответа на вопросы и т.д.).

Максимальную способность системы настраиваться на требования конкретного преподавателя к модели обучаемого можно обеспечить, предоставив пользователю возможность самостоятельно формировать модель обучаемого, т.е. включать в эту модель произвольные параметры и задавать способы их подсчета и правила изменения модели.

Наличие модели обучаемого позволит организовать гибкое (адаптивное) управление процессом обучения. Адаптивная обучающая система может содержать несколько вариантов изложения одного и того же материала. Решение о продолжении обучения по одному из вариантов должно приниматься на основании значений параметров модели обучаемого. Значения параметров могут также учитываться системой при выборе контрольных заданий, лабораторных работ и т.п.

Итак, модель обучаемого выполняет следующие функции:

- адаптация к обучаемому управляющих воздействий системы;
- определение уровня знаний студента по изучаемому курсу и степени достижения заданной цели обучения.

Рассмотрим некоторые существующие методы построения моделей обучаемого.

2.4.4.1. Методы построения модели обучаемого

Методы построения модели пользователя (обучаемого) активно исследуются в области систем компьютерного обучения и систем для диалога на естественном языке [77].

Наиболее простой для реализации является *оверлейная модель*. Она строится в предположении, что знания обучаемого и знания системы имеют аналогичную структуру, при этом знания обучаемого являются подмножеством знаний системы. Каждой теме ПО добавляется числовой атрибут, показывающий степень понимания обучаемым материала по этой теме. Значение этого атрибута определяется в ходе опроса обучаемого.

Более сложная стратегия предложена для *разностной модели*. При построении этой модели система анализирует ответы обучаемого и сравнивает их с теми знаниями, которые заложены в системе и которыми пользуется эксперт при решении подобных задач. Различия между этими знаниями и ложатся в основу модели пользователя. Эта модель позволяет учитывать не только отсутствие знаний у обучаемого, но и неправильное их использование.

В качестве особенности разностной модели можно указать, что она, так же как и оверлейная модель, предполагает идентичность структуры знаний системы и обучаемого. (Впрочем, для обучающей системы это не является препятствием к использованию этих моделей, т.к. задача системы как раз и состоит в том, чтобы передать свои знания обучаемому в той форме, в какой они в ней заложены).

Пертурбационная модель строится в предположении, что знания пользователя и знания системы могут частично не совпадать. В этом случае важной предпосылкой построения такой модели является идентификация причин расхождения, т.к. без определения расхождений модель пользователя будет слишком неопределенной.

Различают следующие причины расхождений:

- недостаток знаний;
- наличие ошибочных знаний;
- неправильное применение знаний или неумение их применить;
- ошибка, порожденная невнимательностью;
- умышленно допущенная ошибка (дается первый попавшийся ответ).

Последняя причина легко обнаружится, если задать несколько простых вопросов. Выявление других причин можно осуществить путем повторного опроса обучаемого (т.н. *уточнение*).

Как видно, все эти методы предполагают наличие не только некоторых количественных параметров (описательной части), но и наличие способов (правил) определения степени соответствия знаний системы и знаний обучаемого, т.е. процедурной части.

2.4.4.2. Общие принципы построения модели обучаемого

Анализ систем обучения позволяет сделать следующий вывод: большинство из существующих обучающих систем не содержит модели обучаемого [47, 75]. Те же системы, в состав которых входит эта модель, построена на основании жестко запрограммированных алгоритмов и наборов параметров (ЭКСТЕРН, РЕПЕТИТОР, КАДИС и др.).

Разработчики закладывают в систему определенный метод (в лучшем случае – несколько методов) построения модели обучаемого. Модель строится по совокупности характеристик (параметров), которые разработчики сочли существенными для определения МО и которые программа умеет рассчитывать. Этот подход является общепринятым, но он обладает очевидными недостатками:

- жестко заданный набор параметров и алгоритмов снижает гибкость системы;
- ограничения в возможности настройки системы сужают круг пользователей системы.

Этих недостатков можно избежать, если включить в обучающую систему возможность формирования *произвольного* набора параметров и задания алгоритма поведения системы в зависимости от значений этих параметров. Кроме этого в систему необходимо включить некоторые готовые модели обучаемого, которые можно будет использовать без предварительной настройки, например, автоматически. Это облегчит работу с системой неподготовленным пользователям (преподавателям) и сократит время на подготовку системы к использованию.

2.4.4.3. Преобразование модели обучаемого

Модель обучаемого формируется (и изменяется) в процессе работы с ним. Система может начинать это формирование "с чистого листа", т.е. при полном отсутствии знаний об обучаемом. Единственное неудобство такого подхода состоит в том, что при этом может увеличиться время адаптации системы к обучаемому.

Другой подход заключается в определении *исходной модели* обучаемого перед началом работы. Здесь можно предложить два метода:

1. Использовать собеседование, а именно: во время регистрации пользователя в системе задать ему несколько вопросов, чтобы определить уровень его знаний и/или его потребности.
2. Предложить несколько усредненных моделей обучаемых, например:
 - по уровню знаний – "новичок", "студент", "практик" и т.д.;
 - по целям общения с обучающей системой – "обучение", "ознакомление", "повышение квалификации" и т.д.

В этом случае каждый из обучаемых будет сам (или с помощью преподавателя) относить себя к определенной группе пользователей.

Можно комбинировать эти методы, оставляя за преподавателем право выбора.

Замечание: нельзя не отметить, что неправильное соотнесение обучаемого одной из усредненных моделей может принести больше вреда, чем пользы. Поэтому этой возможностью следует пользоваться с осторожностью.

Для правильного выбора управляющих воздействий обучающая система должна знать цель, которой требуется достичь, а также критерии, по которым оценивается степень и эффективность достижения этой цели [65].

Цель АОС – передать обучаемому знания, которые в нее заложены. Определить степень достижения этой цели система может с помощью модели обучаемого. Для этого в модель обучаемого (МО) следует включить *целевую модель* обучаемого, при достижении которой задача системы считается выполненной.

Предоставляя преподавателю возможность определять для модели обучаемого произвольный набор параметров, логично определить целевую модель через эти же параметры (или их подмножество). Представим описательную часть МО как набор параметров $M\{p_1, p_2, \dots, p_n\}$. Каждый из этих параметров задается произвольным арифметическим или логическим выражением, которое может включать операции (арифметические или логические), константы и параметры, входящие в МО. Тогда для целевой модели обучаемого:

$$M^c\{p_1^c, p_2^c, \dots, p_k^c\},$$

где $k \leq n$.

Представим декларативную часть модели обучаемого в виде комбинации параметров. Каждый из параметров является произвольной характеристикой обучаемого, отдельной темы или отдельного вопроса. Совокупность характеристик определяется преподавателем на этапе формирования описания модели обучаемого и предметной области. Для каждого параметра P_i устанавливаются возможные значения в виде диапазона или списка значений:

$$P_i = \begin{cases} \langle \text{список значений} \rangle \\ \min : \max \\ \text{NULL} \end{cases}, \quad (2.1)$$

где $\langle \text{список значений} \rangle$ – перечисление возможных значений параметра; \min , \max – минимальное и максимальное значения параметра, а значение $NULL$ в

каждом конкретном случае означает, что для данного обучаемого (темы, вопроса) этот параметр не установлен (не известен).

Декларативная часть модели может представлять собой совокупность средних значений установленных параметров с поправкой на оценку знаний обучаемого.

Изначально об обучаемом ничего не известно, и значения всех параметров принимаются равными некоторым усредненным значениям, которые определяются преподавателем, а по умолчанию приравниваются нулю. При прохождении начального тестирования (и/или очередной темы) значения параметров модели, установленных для данной темы (не равных NULL), пересчитываются следующим образом. По результатам опроса обучаемого ему выставляется оценка b : $b \in [\min, \max]$. Эта оценка нормализуется:

$$b' = 0.5 - (\max - b) / (\max - \min). \quad (2.2)$$

Таким образом, оценка $b' \in [-0.5, 0.5]$.

Новое среднее значение параметра P_i на текущий момент t дискретного локального времени рассчитывается исходя из предыдущего значения этого параметра

$$P_i(t + 1) = P_i(t) + \frac{b' \cdot p_i}{(t + 1)},$$

где b' – нормализованное значение последней оценки (2.2), полученной обучаемым, p_i – значение параметра P_i , установленное для данной темы, а $P_i(t)$ – среднее значение параметра P_i на предыдущий момент времени, которое рассчитывается как среднее арифметическое значение

$$P_i(t) = \frac{\sum_{i=1}^t p_i}{t}.$$

Под локальностью времени подразумевается, что для каждого параметра P_i ведется свой отсчет времени t , равный количеству тем (вопросов), для которых этот параметр установлен.

Значения параметров модели обучаемого используются в процессе работы с обучаемым для определения стратегии и тактики поведения системы. Управление моделью обучаемого также возлагается на базу знаний.

2.4.5. Дополнительные возможности

Рассмотрим подробнее возможности такой организации управления обучением, при которой АОС включает модель предметной области (МПО) и модель обучаемого (МО).

Семиотическая сеть позволяет анализировать заданную структуру курса. Анализ может включать нахождение несвязанных или, наоборот, ключевых вершин (например, вершин с большим количеством выходов или вершин, принадлежащих любому пути из начальной вершины в конечную).

Зачастую обучаемые имеют слабое представление о том, какое место конкретная дисциплина занимает в курсе обучения, где и когда обучаемому могут пригодиться знания, полученные при изучении данной дисциплины. Поэтому наличие модели предметной области имеет огромное значение для обучаемого. Графическое представление этой модели может быть выведено на экран, что позволит обучаемому наглядно увидеть содержимое курса обучения, структуру курса и связи данного курса с другими дисциплинами.

Говоря о роли наглядного изображения информации в более широком плане, следует отметить, что графические иллюстрации занимают все большее место в печатных и электронных изданиях, существенно облегчая восприятие и запоминание текстовой информации [33]. Общая тенденция развития научного познания состоит в том, что по мере возрастания удельного веса абстрактности в науках возникает и противоположный процесс поиска наглядных интерпретаций самых сложных теоретических положений. Это особенно важно на этапе передачи знаний – в процессе обучения [21].

Графическая визуализация взаимосвязей дисциплин и тем в рамках дисциплины позволит обучаемому получить целостное представление об

изучаемом курсе и о его практической ценности. Здесь можно говорить уже о когнитивной графике, т.е. "графике, способствующей познанию" [34]. Проиллюстрируем вышесказанное фрагментом семантической сети, описывающей курс "Базы данных" (рис. 2.5). (Пунктиром указаны понятия, относящиеся к смежным дисциплинам).

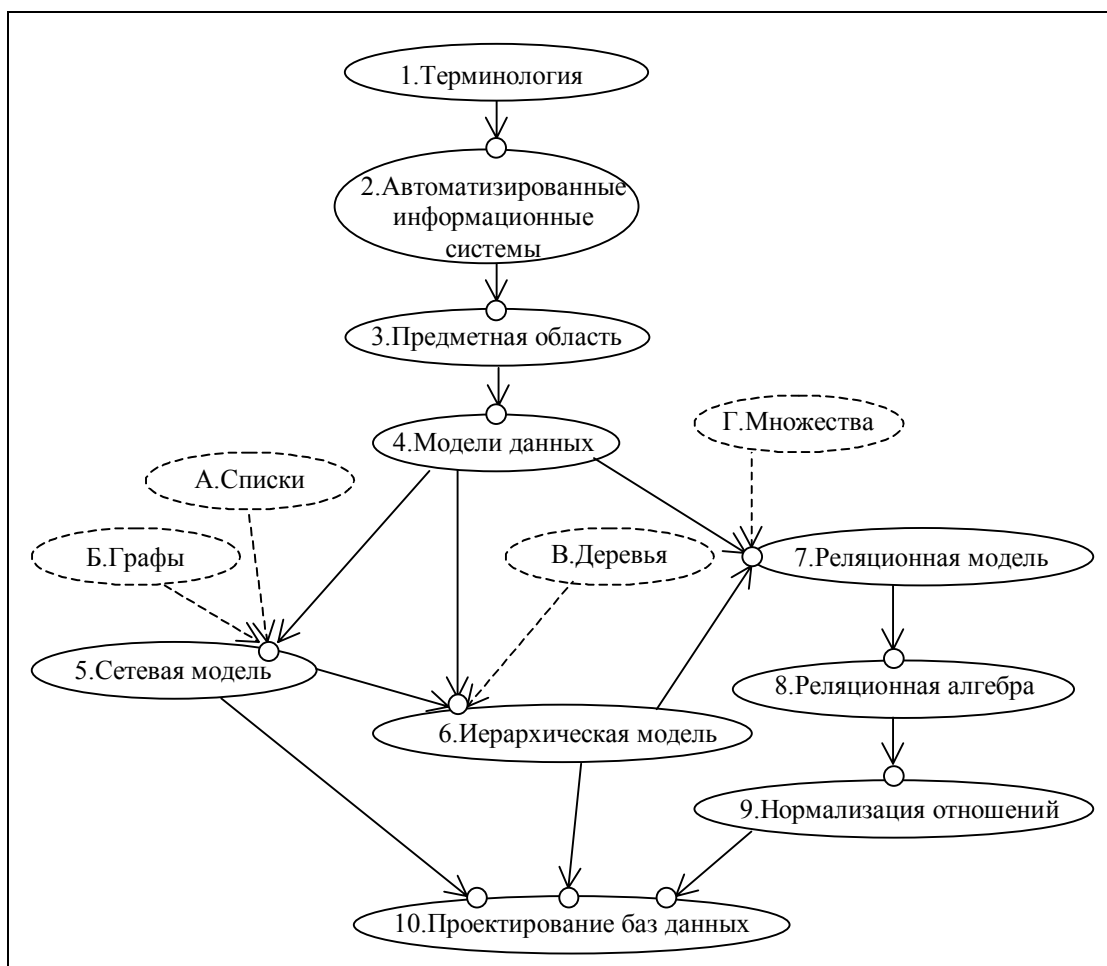


Рис.2.5. Фрагмент сети, описывающей дисциплину "Базы данных"

Текстовое описание всех взаимосвязей этого фрагмента было бы громоздким и неинформативным. Почти каждая вершина этой сети участвует более чем в одной связи, поэтому последовательная форма записи потребовала бы введения повторов, а значит, стала бы избыточной.

Трактовка модели обучаемого как проекции его знаний на модель предметной области имеет и еще одно важное свойство. В исследованиях по управлению обучением (в частности, проведенных в педагогическом институте Чехословацкой Академии наук [111]) было показано, что

обучаемые испытывают "потребность иметь адекватный образ собственных действий и их изменений", но им мешает "неспособность создать такой образ быстро и собственными силами". В указанном эксперименте было установлено, что только часть обучаемых (около 60 процентов) изначально стремится научиться решать задачу наилучшим образом, а остальные действуют методом проб и ошибок. Однако наличие наглядной содержательной информации о динамике результатов своих действий способствует тому, что у всех обучаемых начинает вырабатываться оптимальная последовательность действий или увеличивается скорость ее формирования.

Графическое представление МО позволит обучаемому получить наглядное динамическое представление об своих успехах (и неудачах) при изучении предмета. Достаточно отразить на модели предметной области информацию о том, какие темы обучаемый изучил и насколько успешно, а какие – нет. Эта информация может быть выдана на экран в виде раскрашенного графа, отражающего структуру предметной области и проекцию на нее знаний обучаемого. Такая иллюстрация поможет обучаемому сориентироваться в предмете и определить для себя план изучения предмета (если он не задан жестко и не управляется программно).

2.5. Контроль знаний

Контроль знаний обучаемого является важной частью работы с пользователем. Он обеспечивает обратную связь с обучаемым и предназначен в первую очередь для определения уровня знаний обучаемого с целью организации адаптивного управления обучением.

В обучающих системах обычно используются два основных подхода к организации контроля знаний:

1. Оценка действий обучаемого [2, 84]. Этот метод применяется в экспертно-обучающих системах, т.е. обучающих системах, основанных на знаниях. Знания о предметной области и правила оценки действий обучаемого

позволяют системе в ходе диалога с обучаемым определять уровень его знаний без задания контрольных вопросов. С помощью этого метода можно на высоком уровне моделировать взаимодействие преподавателя с обучаемым, но, как уже говорилось выше, обсуждение подобных систем выходит за рамки данной работы.

2. Стандартизованный контроль знаний [100]. Сущность их состоит в том, обучаемому предлагается выборка специальных заданий и по ответам на нее выносится суждение о его знаниях. Стандартизованные методы контроля знаний имеют следующие положительные свойства, определяющие целесообразность их применения:

- кратковременность проверки;
- стандартность проведения проверки и анализа результатов;
- возможность представления результатов проверки в числовой форме;
- возможность математической обработки результатов.

Для измерения способностей стандартизованными методами необходимо проводить периодические, а не одноразовые проверки.

Стандартизованный контроль знаний в АОС выполняется с помощью тестов [1]. В общем случае, когда проверку результатов теста осуществляет человек, тест – это набор вопросов и правильных ответов к ним (эталонов). Если же определение правильности ответа возлагается на ЭВМ, необходимо включить в тест набор параметров, управляющих алгоритмами проверки выполнения теста.

Педагогические тесты принято делить на нормативно–ориентированные и критериально–ориентированные [10]. В первом случае результатом теста является количественная оценка знаний тестируемых, а во втором – качественная (типа "сдал" – "не сдал"). Для описания нормативно–ориентированного тестирования широкое распространение получили логистические модели и исследующий их IRT–анализ (Item Response Theory). Эта теория хорошо изучена в части, относящейся к дихотомическим тестам,

для которых разработаны одно-, двух- и трехпараметрические модели для анализа результатов тестирования (модели Раша–Бирнбаума).

К сожалению, для дихотомических тестов вводится сильное ограничение – двухбалльная система оценки ответа: ответ считается либо абсолютно правильным, либо совсем неправильным. Это ограничение не соответствует реальному положению, когда правильность ответа определяет преподаватель, который подходит к оценке ответа дифференцированно. Поэтому целесообразно включить в систему контроля знаний возможность анализа ответа тестируемого с определением степени правильности ответа, т.е. степени его соответствия эталону.

2.5.1. Схемы проведения контроля знаний

Рассмотрим общую схему проведения контроля знаний. В общем случае контроль осуществляется за несколько сеансов C_i , каждый из которых строится на основе модели обучаемого. Результаты сеансов в свою очередь вызывают (могут вызвать) изменение модели (рис.2.6).

Если рассматривать отдельный сеанс, то он состоит из трех этапов:

- подготовка задания для контроля (с учетом модели обучаемого),
- опрос обучаемого,
- оценка результатов опроса и внесение изменений в модель обучаемого.

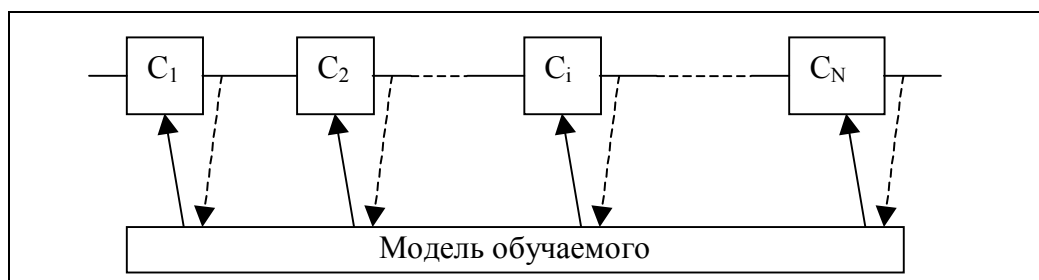


Рис.2.6. Взаимовлияние сеансов контроля знаний и модели обучаемого

Общая схема проведения сеанса контроля знаний представлена на рис. 2.7.

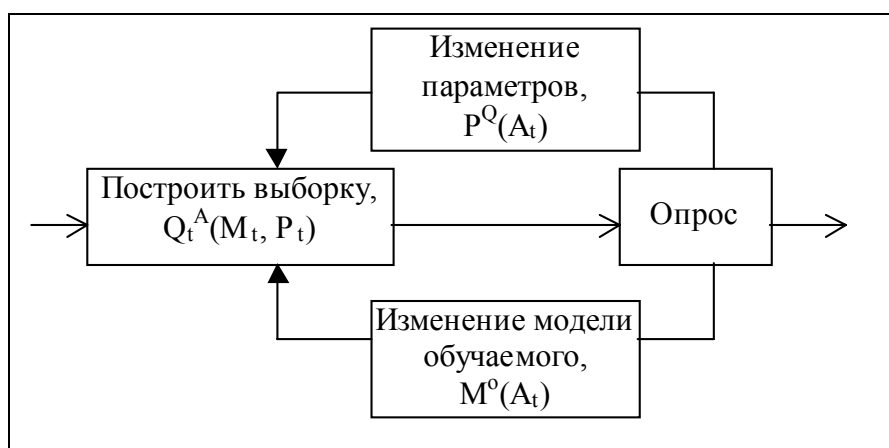


Рис.2.7. Общая схема проведения контроля знаний

Для формирования контрольного задания из множества вопросов Q выбирается подмножество вопросов (выборка), которые будут заданы обучаемому. Назовем это *актуальным множеством* Q^A , $Q^A \supset Q$. Изначально это множество зависит от модели обучаемого M^o и от параметров вопросов P^Q . В процессе опроса это множество может претерпевать изменения. Изменение актуального множества на основании ответа A_t обучаемого на очередной вопрос может осуществляться через модификацию модели обучаемого или задание других параметров вопросов.

Наличие обратной связи актуального множества и ответа обучаемого обеспечивает адаптацию Q^A к обучаемому во время проведения контроля. При реализации более примитивных форм контроля знаний одна или обе обратных связи могут отсутствовать.

Нельзя не отметить, что в существующих системах обучения чаще используются либо готовые тесты в виде фиксированного набора вопросов, либо простейшая схема "построить выборку" – "провести опрос" – "оценить ответы". Адаптивный контроль знаний или САТ-подход (Computer Adaptive Testing) применяется редко; но даже там, где он используется (например, [126]), алгоритм адаптации запрограммирован.

Блок-схема проведения контроля знаний выглядит так (рис.2.8):

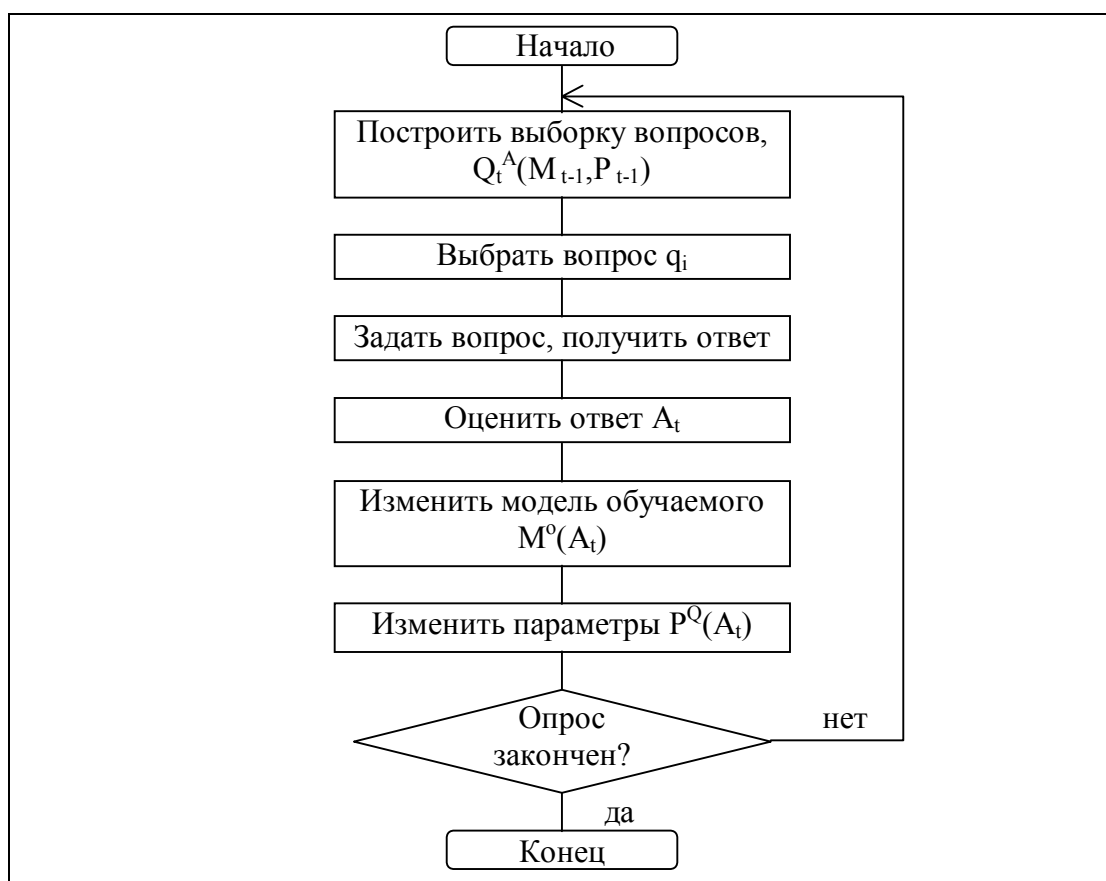


Рис.2.8. Блок-схема проведения адаптивного контроля знаний

Адаптивный тест предполагает, что для каждого задания известны параметры трудности и дифференцирующей способности. Информация об этом может быть получена эмпирическим путем. Задания такого теста должны быть упорядочены в соответствии с интересующими характеристиками заданий [1].

Одновременно обучаемому может быть задан только один вопрос. С учетом этого обстоятельства и для упрощения алгоритмической реализации более приемлем другой вариант (рис.2.9). Сначала формируется достаточно большое актуальное множество, а каждый следующий вопрос из него выбирается на основании ответа на предыдущий вопрос. Актуальное множество формируется с учетом возможных изменений параметров выборки в процессе контроля. Изменения вносятся в модель обучаемого по результатам контроля знаний в целом за сеанс.

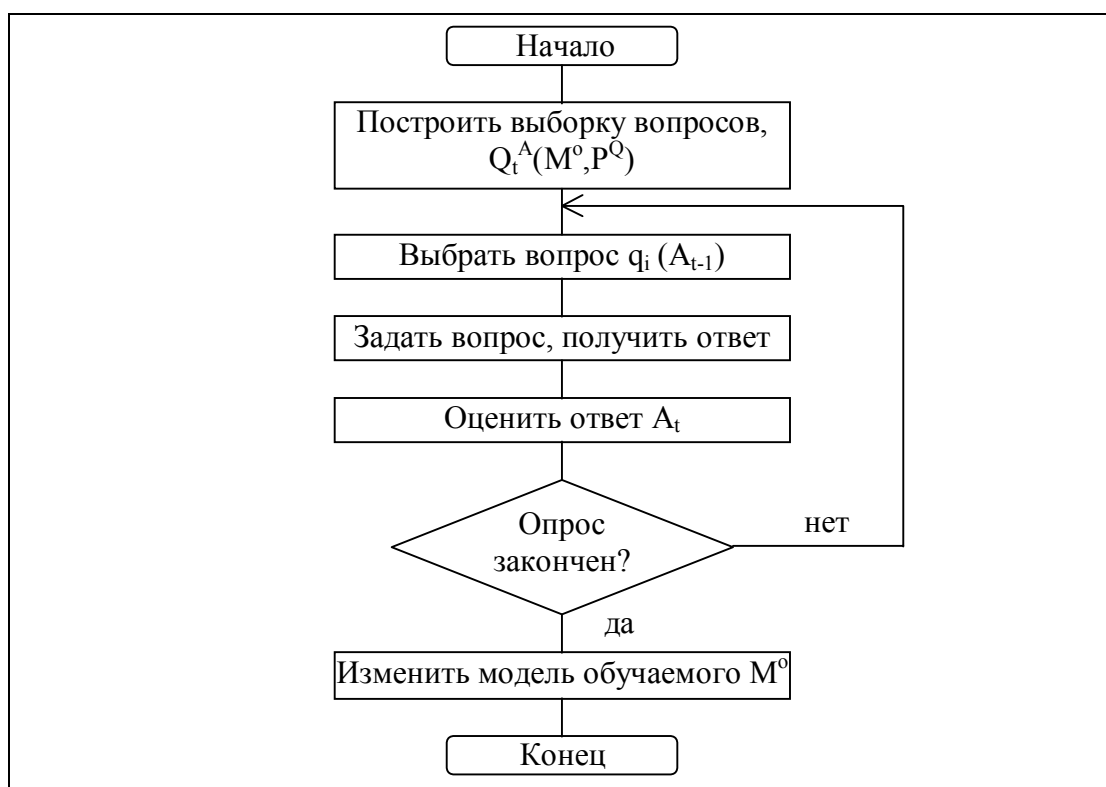


Рис.2.9. Модифицированная блок-схема адаптивного контроля знаний

Для управления процессом контроля знаний и моделью обучаемого система должна предоставлять возможность введения в тест параметров.

2.5.2. Параметры проведения контроля знаний

По способу задания и обработки параметры, относящиеся к описанию способов контроля знаний, можно разделить на две группы:

- Встроенные параметры. Позволяют настраивать внутренние алгоритмы управления контролем знаний.
- Пользовательские параметры. Задаются преподавателем, предназначены для управления тестированием и формирования модели обучаемого.

2.5.2.1. Встроенные параметры

С помощью встроенных параметров можно организовать различные способы проведения контроля знаний. По результатам анализа работ, в которых освящается этот вопрос [71, 78, 100], можно выделить такие группы параметров:

- *Тип контроля.* Определяет влияние полученной оценки на дальнейшие действия системы.
- *Минимальная оценка.* Предназначена для подсчета оценки.
- *Количество вопросов.* Определяет, сколько вопросов следует задать обучаемому во время опроса.
- *Количество ответов.* Определяет, сколько ответов должен дать обучаемый, чтобы опрос был признан состоявшимся.
- *Способ подсчета оценки.* Определяет способ получения итоговой оценки на основании оценок за отдельные вопросы.
- *Схема проведения опроса.* Задаёт условие окончания опроса.
- *Способ формирования выборки вопросов.* Определяет принцип включения вопросов в задание.
- *Способ подачи вопросов.* Задаёт порядок выдачи вопросов обучаемому.
- *Время проведения опроса.* Ограничивает время, которое дается обучаемому на выполнение задания.

Рассмотрим возможные значения этих параметров:

1. Тип контроля (обязательный параметр). Возможны следующие значения:
 - Завершающий – предназначен для опроса по одной теме. Результаты этого контроля определяют, сдана тема или нет.
 - Промежуточный – предназначен для самоконтроля по одной теме. Полученная оценка фиксируется (например, в модели обучаемого), но не влияет на сдачу темы.
 - Выборочный – служит для итогового контроля по нескольким темам. Фиксируется общая оценка и оценки по отдельным темам, входящим в опрос.
2. Минимальная оценка (обязательный параметр): оценка M_p , при которой тема считается сданной.
3. Количество вопросов N_q . Может быть указано как:
 - конкретное натуральное число;

- доля от общего количества вопросов по данной теме – число в интервале]0,1].
4. Количество ответов N_a . Значение определяется аналогично параметру "количество вопросов" и имеет следующий смысл: если обучаемый ответил меньше чем на N_a вопросов, то опрос считается несостоявшимся (но попытка и оценка могут фиксироваться системой). Для N_a должно соблюдаться условие: $N_a \leq N_q$.

5. Способ подсчета итоговой оценки:

- СУММА – итоговая оценка O равна сумме баллов O_i , полученных за отдельные ответы:

$$O = \sum_{i=1}^{N_q} O_i. \quad (2.3)$$

- ШКАЛА – итоговая оценка O подсчитывается в соответствии со шкалой, заданной минимальным и максимальным баллами:

$$O = \frac{\sum_{i=1}^{N_q} O_i * (\max - \min) + \min}{N_q}, \quad (2.4)$$

где O_i – оценки, полученных за отдельные ответы, \min и \max – минимальный и максимальным баллы соответственно.

- ПРОЦЕНТ – итоговая оценка O подсчитывается как процент правильных ответов:

$$O = \frac{m}{N_q} * 100\%, \quad (2.5.)$$

где N_q – общее число заданных вопросов, m – число правильных ответов.

6. Схема проведения опроса:

- **Линейная схема.** Задаются все N_q вопросов, затем подсчитывается оценка.
- **Пороговая схема.** Вопросы из выборки длиной N_q задаются до тех пор, пока не перейден порог "тема сдана" или "тема не сдана". Пороги считаются перейденными, если оставшиеся вопросы не могут изменить результата контроля. Порог "тема не сдана" считается перейденным, если

получение за оставшиеся вопросы максимальных оценок не позволит обучаемому набрать нужной оценки M_p . А порог "тема сдана" – если получение за оставшиеся ответы минимальной оценки не приведет к выставлению обучаемому оценки меньше M_p .

- **Схема с дополнительными вопросами.** Задается минимальное число вопросов N_q , если по результатам подсчета оценки M выясняется, что велик разброс оценок или оценка близка к критической (M_p), задаются дополнительные вопросы, пока не будет:

- исчерпан весь список вопросов по данной теме или
- задано максимально допустимое количество вопросов или
- уменьшен до приемлемого уровень разброса оценок или
- превышен порог положительной оценки M_p .

Эта схема требует введения дополнительных параметров:

- N_{max} – максимально допустимое количество вопросов.
- ε_p – уровень близости оценки к критической: полученная оценка M считается близкой к критической, если она удовлетворяет условию:

$$|M - M_p| \leq \varepsilon_p.$$

- ε_r – уровень допустимого разброса оценок, например, как среднее квадратичное отклонение:

$$\varepsilon_r \geq \sqrt{\sum_i (m_i - m_{cp})^2},$$

где m_{cp} – средняя оценка за все полученные ответы, m_i – оценка за i -й ответ.

- **Схема с уточнением.** Задается по одному вопросу; выбор следующего вопроса зависит от ответа на предыдущий и может быть с ним связан. Всего задается минимум N_q вопросов. Условия окончания опроса те же, что и в предыдущем случае. Для этой схемы выборка вопросов должна быть сформирована по адаптивному методу (см. ниже). Набор правил, реализующих эту схему, приведен в Приложении 1.

Замечание: естественно, схема с уточнением является самой общей из всех рассмотренных. Но она требует много подготовительной работы при составлении

заданий для контроля, что неизбежно увеличит срок подготовки и ввода в эксплуатацию обучающих курсов. Поэтому приводятся и другие (упрощенные) схемы, в том числе, и не требующие задания каких-либо параметров для вопросов теста.

7. Способ формирования выборки вопросов. Выборка вопросов формируется сразу, и потом обучаемому задается по одному вопросу. Можно выделить следующие способы формирования выборки:
- a) Случайно – все вопросы выбираются случайным образом (вторично один и тот же вопрос не включается в выборку).
 - b) По условию – в выборку включаются только те вопросы, для которых истинна заданная пользователем функция $F(p_1, p_2, \dots, p_n)$, где p_i – произвольный параметр вопроса.
 - c) Пропорционально значениям определенного параметра P . Если в полном списке вопросов по данной теме содержится n_1 вопросов со значением параметра p_1 , n_2 вопросов со значением параметра p_2 и т.д., то в выборку вопросов длиной N_q войдет k_i вопросов типа p_i :

$$k_i = \frac{N_q * n_i}{\sum_{i=1}^L n_i}, \quad (2.6)$$

где L – количество различных значений параметра P .

- d) По значению группирующего параметра – вопросы выбираются на основании определенного (основного или группирующего) параметра p_z , по значению которого они объединяются в группы, относящиеся к одному понятию (термину, объекту, методу и т.п.) и имеющие, например, разные степени сложности (общности). Выборка формируется таким образом, чтобы каждый следующий вопрос в группе являлся уточнением предыдущего. Порядок вопросов в группе определяется значением другого (уточняющего) параметра p_y , который, вообще говоря, может иметь произвольную семантику.

Для реализации данной схемы можно использовать пороговую схему проведения опроса для каждого значения группирующего параметра.

Примечание: подробнее способы формирования выборки вопросов описаны в разделе 2.5.3. "Способы формирования актуального множества вопросов".

8. Способ подачи вопросов:

- в порядке формирования списка;
- случайно;
- в порядке увеличения (уменьшения) значения определенного функционала.

9. Время проведения контроля знаний T . Задается в минутах и определяет максимальный временной интервал, в течение которого обучаемый должен дать все ответы. Если обучаемый не уложился в отведенное время, для оценки результатов опроса используются те ответы, которые уже даны.

Для увеличения гибкости управления контролем знаний целесообразно позволить пользователю комбинировать эти способы, т.е. указывать для одного типа контроля по одной и той же теме список значений одного и того же параметра.

Очевидно, что список параметров, через которые пользователи могут влиять на алгоритмы проведения контроля знаний, ограничен. Но можно не ограничивать количество возможных значений этих параметров, если предоставить пользователю возможность самому указывать те способы и методы, которые будут управлять контролем знаний.

Для этого необходимо предусмотреть в системе задание значений параметров с помощью формул – арифметических и логических выражений, состоящих из знаков арифметических и логических операций, названий встроенных функций, имен параметров и значений характеристик, которые могут быть подсчитаны в процессе работы системы.

Замечание: к сожалению, этот способ не подходит для параметров, значения которых не являются вычислимыми, например, "тип контроля".

Если значения каких-либо необязательных параметров для проведения контроля знаний не указаны, необходимо предусмотреть значения, устанавливаемые по умолчанию, например:

1. Количество вопросов N_q – задаются все вопросы по данной теме.
2. Количество ответов N_a – равно количеству вопросов.
3. Способ подсчета оценки – сумма.
4. Схема проведения опроса – простая.
5. Способ формирования выборки вопросов – случайно.
6. Способ подачи вопросов – в порядке формирования списка.
7. Время проведения опроса – не ограничено.

2.5.2.2. Параметры, задаваемые пользователем

Для того чтобы в процессе контроля знаний можно было моделировать разные формы проведения опроса, задание должно сопровождаться соответствующим набором параметров. Несмотря на декларацию необходимости "широкой параметризации алгоритмов управления обучением" [78], практически все существующие системы ограничивают набор параметров, вводя только те из них, которые, вероятно, представляются им наиболее значимыми. Например, в той же системе ЭКСТЕРН вопрос может оцениваться одним параметром – "сложность", – имеющим пять степеней градации.

Действительно широкую параметризацию можно обеспечить, позволив пользователю вводить *любые* параметры для единиц учебного и контрольного материала (тем и вопросов). Параметр определяется именем (строкой символов) и значением числового или строкового типа. Если какой-либо из введенных параметров не определен для конкретного вопроса (темы), то считается, что его значение для этого вопроса (темы) неизвестно (NULL). Преподаватель может ввести любое количество параметров для любого вопроса (темы). Значения параметров, относящихся к вопросу, входят в набор значений, по которым собирается статистика. Эти параметры предназначены:

- для построения модели обучаемого,
- для организации управления обучением (в правилах базы знаний),

- для отражения в статистических отчетах с целью получения общей картины успехов обучаемых и для совершенствования самих заданий.

Наличие подобной возможности позволит преподавателю проводить опрос на основе собственных представлений о методах эффективного контроля знаний.

2.5.2.3. Общие замечания о параметризации

Наличие широких возможностей по настройке программного обеспечения имеет свои плюсы и минусы. С одной стороны, наличие большого количества параметров настройки увеличивает гибкость системы. С другой стороны, такая система требует предварительного изучения и много подготовительной работы.

Для того чтобы избавить пользователя от необходимости изучать какой-либо внутренний язык системы и свести подготовительную работу к минимуму, в АОС можно включить готовые наборы параметров и правил, с помощью которых реализуются те или иные схемы управления работой с обучаемым. Каждый набор должен сопровождаться кратким описанием, объясняющим суть данной схемы.

Такой подход предоставит преподавателю возможность, используя готовые схемы проведения контроля знаний, быстрее ввести обучающую систему в эксплуатацию. В дальнейшем же, по мере изучения возможностей АОС, задания могут быть модифицированы путем использования других схем контроля знаний.

2.5.3. Способы формирования актуального множества вопросов

Представим процесс формирования актуального множества как последовательность преобразований, проводимых над множеством вопросов Q . Выборка формируется последовательно в несколько шагов. На каждом шаге из множества Q выбирается по одному или несколько элементов (вопросу или группе вопросов), выбранные элементы удаляются из множества Q и помещаются в множество Q^A . Каждое преобразование

описывает один шаг построения актуального множества Q^A . Количество шагов определяется количеством вопросов, которое задано для данной выборки параметром N_q . Изначально актуальное множество пусто:

$$Q^A(0) = \emptyset.$$

Запишем эти преобразования для разных методов формирования выборки:

a) Случайно:

$$Q^A(t+1) = Q^A(t) \cup \psi_R(Q(t)),$$

где $Q^A(t+1)$ – актуальное множество в момент времени $(t+1)$, $(0 \leq t < N_q)$; ψ_R – случайная функция выбора очередного вопроса из множества.

b) По условию:

$$Q^A(t+1) = Q^A(t) \cup \psi_F(Q(t)),$$

где ψ_F – функция, выбирающая из множества $Q(t)$ вопрос, удовлетворяющий условию F .

c) Пропорционально значениям определенного параметра P :

$$Q^A(t+1) = Q^A(t) \cup \psi_{F(t)}(Q(t)),$$

где $\psi_{F(t)}$ – функция, выбирающая из множества $Q(t)$ вопрос, удовлетворяющий условию $F(t)$. Функция $F(t)$ задается на основании анализа множества Q и формулы (2.6). Она может представлять собой, например, вектор V длиной N_q , в котором элемент v_i являются значением параметра P для i -го вопроса выборки (при этом условие $F(t)$ заключается в равенстве элемента v_i и значения параметра P для очередного выбираемого вопроса).

d) По значению группирующего параметра:

$$Q^A(t+1) = Q^A(t) \cup \psi_R(Q(t)) \cup \phi_{F(t)}(Q(t)),$$

где ψ_R – случайная функция выбора очередного вопроса q_n из множества $Q(t)$; $\phi_{F(t)}$ – функция, выбирающая из множества $Q(t)$ все вопросы, удовлетворяющие условию $F(t)$:

$$F(t) = \begin{cases} q_j, & \text{если } p_y^{q_j} = p_y^{q_n} \\ \emptyset, & \end{cases},$$

где q_j – это очередной элемент (вопрос) множества $Q(t)$, $p_y^{q_n}$ – значение уточняющего параметра p_y для вопроса q_n . Отметим, что в процессе работы последнего преобразования исходное множество $Q(t)$ меняется.

Таким образом, множество способов формирования актуального множества сводится к комбинации 3-х типов преобразований: выбор вопроса случайным образом (ψ_R), на основании взвешенной функции ($\psi_{F(t)}$) и по условию (ψ_F).

Выводы

Рассмотрены способы организации сетевой среды распределенной обучающей системы, которая включает в себя локальную вычислительную сеть (ЛВС) и предоставляет доступ через Internet. Определена функциональная структура РАОС.

В работе приведены наиболее распространенные решения для организации работы внутри ЛВС и через Internet. Предложена конфигурация мультиплатформенного учебного сервера, включающего в себя сервер РАОС, ориентированный на работу с пользователя в рамках ЛВС, и сервер РАОС, предназначенный для управления базой данных, для организации распределенных вычислений и доступа через Internet.

Определены основные требования к распределенным автоматизированным обучающим системам:

1. работа в сети, включая удаленный доступ, удаленные вызовы процедур, использование распределенных данных и распределенных вычислений;
2. открытость (использование стандартных форматов данных, подключение внешних модулей, параметрическая и алгоритмическая настройка);
3. полнофункциональность;
4. адаптация системы к обучаемому;
5. наличие готовых шаблонов различных моделей и стратегий обучения, облегчающих (ускоряющих) ввод системы в эксплуатацию.

Показано, что для выполнения функций обучения в состав РАОС должны входить знания о стратегии обучения (методиках обучения), модель

предметной области и модель обучаемого. Для обеспечения открытости РАОС предложен подход, при котором стратегия управления обучением, модель предметной области и правила обработки модели обучаемого оформлены как внешние базы знаний, замена которых позволяет менять поведение системы. Такая организация РАОС дает преподавателю возможность вложить в обучающую систему экспертные знания и представления об обучении.

Предложена семиотическая модель для представления структуры предметной области, объединяющая декларативные и процедурные знания. Графическая визуализация взаимосвязей дисциплин и тем в рамках дисциплины дает обучаемому возможность получить целостное представление об изучаемом курсе и о его практической ценности. Это позволяет говорить об использовании когнитивной графики.

Рассмотрены различные алгоритмы и схемы проведения контроля знаний, определен набор базисных характеристик, предназначенных для параметрической настройки подсистемы контроля знаний.

Построение программного комплекса на вышеприведенных принципах позволит создавать на его основе обучающие системы, ориентированные на работу в сети и имеющие доступ к большим объемам распределенных данных. Введение в состав программного комплекса экспертной системы даст преподавателю возможность с помощью правил базы знаний управлять процессом обучения в зависимости от характеристик конкретного процесса обучения, в частности, от результатов контроля знаний обучаемого, обеспечивая тем самым дифференцированный подход.

3. ПОДСИСТЕМА КОНТРОЛЯ ЗНАНИЙ

Организация подсистемы контроля знаний является одной из наиболее сложных задач при создании РАОС.

Примечание: вопросы составления тестов и проверки их качества в данной работе не рассматриваются. Эти проблемы являются предметом многих исследований, наиболее известными из которых являются работы В.С. Аванесова, А.П. Свиридова и некоторые другие [1–2, 100].

3.1. Особенности работы подсистемы контроля знаний в РАОС

Особенности работы подсистемы контроля знаний в РАОС определяются архитектурой режима клиент-сервер. Мультиархитектурная среда предполагает тестирование как в режиме удаленного доступа, так и с помощью локального программного обеспечения.

1. При использовании архитектуры **"тонкий" клиент – "толстый" сервер** тестирование проходит в режиме удаленного доступа. В рамках ЛВС это не влечет за собой особых проблем. Тестирование проходит в штатном режиме (по одному вопросу), запуск внешних процессов осуществляется через RPC (Remote Process Call), возможно применение схем адаптивного тестирования.

При наличии высокоскоростного канала возможен аналогичный доступ через Internet. Основная проблема при этом связана, естественно, с внешними процессами, которые приходится оформлять как Internet-приложения.

2. При использовании архитектуры **"толстый" клиент – "тонкий" сервер** в задачи сервера входит: хранение информации о результатах тестирования; организация взаимодействия подсистемы контроля знаний, расположенной на машине клиента, и вспомогательного программного обеспечения (ВПО).

Централизованное хранение базы данных с результатами тестирования имеет следующие преимущества:

- надежная защита от несанкционированного доступа;
- простота организации удаленного администрирования;
- широкие возможности статистической обработки данных.

Для "тонкого" клиента этот подход вообще является единственно возможным, а для "толстого" клиента можно предусмотреть временную буферизацию результатов тестирования в случае отсутствия доступа к серверу.

Во время контроля знаний ВПО может быть задействовано при оценке ответов (прежде всего, текстовых и графических) и при анализе результатов теста (например, для определения причин неправильных ответов). Использование ВПО позволяет расширить функциональные возможности системы. Но оценка ответов с помощью ВПО неизбежно приведет к увеличению времени реакции системы.

Таки образом, для тестирования через Internet, а также для ситуации использования ВПО для оценки ответов, необходимо (наряду с интерактивным режимом) предусмотреть пакетный режим тестирования. Пакетный режим подразумевает использование линейной схемы тестирования: обучаемый сначала отвечает на все вопросы, а потом система оценивает ответы. Пороговая, адаптивная и другие схемы предполагают учет оценки текущих ответов в процессе тестирования, поэтому не подходят для пакетного режима.

3.2. Характеристики систем контроля знаний

3.2.1. Анализ существующих систем контроля знаний

В настоящее время существует большое количество систем контроля знаний (СКЗ), как выполненных в виде отдельных программных продуктов (например, ITEMAN, RASCAL, RSP, The Examiner testing system, FastTEST professional, C-Quest, CONTEST, ГРАММАТЕЙ-КЛАСС, ПОЛСТАР, "Контроль знаний", "Экзаменатор", "Аттестация"), так и встроенных в

обучающие системы (например, во всех АОС предусмотрен контроль знаний). Отсутствие стандарта или хотя бы общепринятых норм и правил написания таких систем привело к тому, что практически каждая из них является "вещью в себе" и используется, за редким исключением, только ее собственными разработчиками.

Анализ существующих СКЗ выявил следующие недостатки этих систем:

1. Ограниченное количество типов ответов. Не во всех системах реализованы даже основные формы (по Аванесову, [1]) представления ответов.
2. Отсутствие формальных методов дифференцированной оценки ответов. Существуют тестовые системы, в которых оценка может быть дифференцированной, но это реализуется неформальными способами:
 - задается несколько эталонов, один из которых является абсолютно правильным, а другие – частично правильными (для них указывается более низкий балл, чем для абсолютно правильного эталона);
 - обучаемому предоставляется N попыток ответа: если он сразу дал правильный ответ, он получает максимальный балл, если с i -й попытки ($i \leq N$), то оценка уменьшается пропорционально значению i .
3. Слабые возможности параметрической настройки теста.
4. Отсутствие открытой архитектуры. Закрытость систем обуславливается:
 - хранением тестов в базе данных системы;
 - отсутствием возможности подключения внешних модулей для представления вопросов, получения ответов и оценки ответов;
 - ограниченным количеством алгоритмов формирования теста и управления проведением тестирования.
5. Многие СКЗ предоставляют возможности удаленного тестирования (через Internet), но возможности таких систем еще более ограничены по сравнению с локальными СКЗ.

3.2.2. Требования к подсистеме контроля знаний

При создании подсистемы контроля знаний необходимо:

- сделать систему максимально открытой;
- обеспечить переносимость тестов на уровне их исходных текстов;
- по возможности учесть все типы вопросов и ответов;
- реализовать методы оценки ответов различных типов, которые позволят дифференцированно оценивать ответы обучаемого.

Для решения этой задачи требуется:

- проанализировать различные типы тестовых вопросов и ответов;
- ввести метрику для определения степени сходства ответов обучаемого и эталонных ответов;
- разработать формат описания тестов, учитывающий различные варианты ответов и вопросов и возможность подключения внешних модулей.

Создание языка описания тестов решает задачу унификации СКЗ. Разработка развитого средства описания позволяет, с одной стороны, унифицировать интерфейс, используя одну и ту же оболочку для различных тестов. Это приведет к снижению трудозатрат пользователя на изучение системы и позволит ему сосредоточиться на ответах на вопросы. С другой стороны, открытость формата описания теста дает возможность создавать разные оболочки для одного и того же набора контрольных заданий, адаптируя систему для различных операционных систем, аппаратных платформ и требований по организации интерфейса.

Для того чтобы отделить описание теста от его алгоритмической реализации и сделать его переносимым, язык описания теста должен быть декларативным.

Любая система контроля знаний включает в себя множество предлагаемых вопросов, правильные ответы на них и правила определения корректности ответов, полученных в ходе опроса. Поэтому описание теста состоит из следующих частей:

1. описание правил формирования теста из списка контрольных заданий и методов оценки результатов теста в целом;
2. описание вопросов;
3. описание эталонов ответов и методов оценок ответов, полученных в ходе опроса.

Начнем с рассмотрения различных типов тестовых вопросов и ответов.

3.3. Анализ ответов обучаемого

Одним из основных блоков специализированного математического обеспечения АОС является блок анализа ответов обучаемого, который функционирует следующим образом. При использовании любого способа ввода ответа выделяются некоторые признаки, в соответствии с которыми ответ относится к категории правильных или неправильных.

3.3.1. Представление вопросов и ответов

Вопрос – это передача информации по прямому каналу связи (от системы к обучаемому). Обычно он оформляется в виде текста, возможно, сопровождающегося графической иллюстрацией или звуковым сообщением.

В реальных системах наиболее распространенная в СКЗ форма ответов – выборочная: вопрос сопровождается несколькими готовыми вариантами ответов, из которых нужно выбрать один, реже – нескольких правильных ответов. Вторым по популярности идет числовой ответ, обычно, как результат решения предложенной задачи. Кроме этого, иногда используется текстовый ответ, но (за редким исключением) без анализа, т.е. правильным считается ответ, полностью совпадающий с эталоном. Некоторые системы контроля знаний поддерживают также отдельные специфические формы ответов, например, позволяют создавать структурные схемы на заданной элементной базе или выбирать определенную область на графическом изображении. Но подобные формы ответов не имеют широкого применения, т.к. не обладают достаточным уровнем абстракции.

На основании анализа существующих тестовых систем можно выделить:

- Типы вопросов (по форме представления):
 1. Текст.
 2. Изображение.
 3. Процесс.
 4. Речевое сообщение (сводимо к типам "текст" и/или "процесс").
- Типы ответов (по форме ввода и представления):
 1. Множество элементов (неупорядоченное).
 2. Список элементов (упорядоченный).
(В качестве элементов в пп.1,2 могут выступать списки и множества.)
 3. Выражение (арифметическое).
 4. Фраза (текст).
 5. Рисунок.
 6. Речевое сообщение (сводимо к фразе).

Один из основоположников научного подхода к проблеме тестирования В.С. Аванесов показал, что все ответы могут быть сведены к следующим типам [1]:

1. Закрытые: выбор одного или нескольких вариантов ответов в произвольном порядке. Такие ответы называют выборочными.
2. Определение порядка вариантов ответов.
3. Открытые: конструируемые ответы (числовые и текстовые).
4. Определение соответствия элементов двух множеств.

Фактически, первый вариант – это множество элементов, второй – это список, определение соответствия элементов двух множеств может быть представлено как множество множеств, а конструируемые ответы – это текст или арифметическое выражение (как более общее представление числового ответа). Таким образом, без аналога остается только рисунок. Поэтому в дальнейшем мы попробуем свести его представление к одному из четырех указанных типов.

3.3.2. Типы вопросов и ответов

С программной точки зрения наиболее общей формой представления вопросов и ответов является процесс: для задания вопросов и для получения ответов можно запускать соответствующий процесс, передавая ему информацию о вопросе и эталон ответа и получая взамен оценку ответа обучаемого. Но не для всех ситуаций этот вариант является приемлемым с точки зрения целесообразности и быстродействия. Например, известно, что большинство вопросов и ответов представляется в текстовой форме, поэтому, как минимум, необходимо выделить тип вопроса и ответа ТЕКСТ.

С практической точки зрения множество вариантов вопросов и ответов сводимо к следующим типам:

- **Типы вопросов:**

1. ТЕКСТ – вопрос, представленный в виде строки символов.
2. КОМАНДА – командная строка – вопрос, для задания которого необходим запуск внешнего процесса (вывод изображения, звуковое сообщение и т.д.).

- **Типы ответов:**

1. ТЕКСТ – ответ, представленный в виде строки символов.
2. МНОЖЕСТВО – ответ, представленный как неупорядоченное множество элементов.
3. СПИСОК – ответ, представленный как упорядоченное множество элементов.
4. ВЫРАЖЕНИЕ – ответ, представленный как арифметическое выражение.

Все ответы, представление которых выходит за рамки указанных типов (рисунок, речевое сообщение и др.), могут быть получены в рамках процесса – командной строки, указанной для соответствующего вопроса. Процесс осуществляет вывод вопроса и получение ответа, который передается системе контроля знаний через буфер обмена данными (например, файл с определенным именем). Таким образом, ответ обучаемого может быть

преобразован в строку (например, речевое сообщение) или оценен внутри процесса (например, рисунок). В последнем случае тип ответа – ВЫРАЖЕНИЕ, а в качестве эталона выступает оценка.

3.3.3. Методы определения правильности выборочных ответов

Популярность выборочных ответов отчасти объясняется общей практикой проведения тестирования, отчасти – простотой анализа выборочного ответа.

В общем случае выборочный ответ представляет собой множество элементов (неупорядоченное) или список элементов (упорядоченный). В качестве элементов могут выступать списки и множества, тогда получаются двухуровневые схемы МНОЖЕСТВО СПИСКОВ и СПИСОК МНОЖЕСТВ.

Примечание: Теоретически можно рассматривать схемы произвольного уровня вложенности. Но на практике излишнее усложнение формы ответа может привести к дополнительным ошибкам при вводе и, как следствие, к неверной оценке знаний обучаемого.

Правильность выборочного ответа оценивается путем сравнения ответа и эталона и определения их сходства. В существующих системах правильность выборочного ответа обычно определяется полным совпадением с эталоном. Если учесть, что чаще всего применяется схема "N вариантов – из них один правильный", то такой подход является оправданным. Но в том случае, если применяется схема "N вариантов – из них k правильных ($0 \leq k < N$)", было бы целесообразнее ввести дифференцированную оценку. Например, ответ (2,3,5) гораздо ближе к эталону (3,4,5), чем ответ (1,2), поэтому он должен быть оценен выше. Ведь преподаватель при опросе поступает именно так. Для дифференцирования оценки необходимо использовать какой-либо монотонный функционал, который позволял бы определять степень сходства ответа и эталона. Далее предложен такой подход.

Способ определения правильности ответа зависит от типа ответа. Мы выделили четыре типа выборочного ответа – МНОЖЕСТВО, СПИСОК, МНОЖЕСТВО СПИСКОВ и СПИСОК МНОЖЕСТВ.

3.3.3.1. Сравнение множеств элементов

В тех случаях, когда ответ представляет собой множество, т.е. произвольную последовательность элементов, оценка ответа заключается в определении расстояния между множествами.

Пусть даны два множества M_a и M_e , состоящие из элементов базового множества R . Требуется определить расстояние между множествами.

Расстояние должно удовлетворять следующим условиям:

1. $|aa| = 0, \forall a$;
2. $|ab| = |ba|, \forall a, b$;
3. $|ab| + |bc| \leq |ac|, \forall a, b, c$.

Примем $[0, 1]$ за интервал изменения расстояния r таким образом, что:

- $r = 0$, если $M_a \equiv M_e$;
- $r = 1$, если $M_a \cap M_e = \emptyset$;
- $0 < r < 1$, если $M_a \cap M_e \neq \emptyset$.

Пусть дано произвольное конечное множество R . Расстояние между подмножествами $A = \{a_i\}$ и $B = \{b_i\}$ множества R может быть определено как:

$$r = 1 - \frac{K}{L}, \quad (3.1)$$

где K – количество совпадающих элементов в подмножествах A и B , $L = |A| = |B|$ – мощность подмножеств. Если $|A| \neq |B|$, т.е. количество элементов в подмножествах различно, это нужно учитывать.

Тогда формула (3.1) изменится следующим образом:

$$r = 1 - \frac{K}{L + K_b}, \quad (3.2)$$

где $K = |A \cap B|$ – количество совпадающих элементов в подмножествах A и B , $L = |A|$ – мощность подмножества A , $K_b = |A| - K$ – количество элементов в подмножестве B , которых нет в A .

В соответствии с таким определением расстояния введем понятие степени сходства множеств. **Степенью сходства множеств** назовем величину, обратную к расстоянию:

$$\delta = 1 - r. \quad (3.3)$$

Объединив формулы (3.2) и (3.3), можно оценить ответ обучаемого по степени сходства δ_l между ответом обучаемого (множество $S_a = \{a_1, a_2, \dots, a_i, b_1, b_2, \dots, b_j\}$) и эталонным ответом (множество $S_e = \{a_1, a_2, \dots, a_i\}$) по формуле:

$$\delta_l = \frac{K_a}{L + K_b}, \quad (3.4)$$

где $L = |S_e|$, $K_a = |S_a \cap S_e|$, $K_b = |S_a| - K_a$.

3.3.3.2. Сравнение списков элементов

Для списка элементов правильность ответа проверяется путем определения расстояния между списками. Понятие расстояния между списками базируется на работе Кендала (подробно описанной в [117]), где введена мера сравнения порядка списков.

Пусть имеется два списка $X_i^T = \{x_{i1}, x_{i2}, \dots, x_{ik}, \dots, x_{in}\}$ и $X_j^T = \{x_{j1}, x_{j2}, \dots, x_{jk}, \dots, x_{jn}\}$, состоящие из элементов одного и того же базового множества R . Требуется определить расстояние между списками.

Коэффициент сравнения определяют следующим образом:

$$\Delta_{lk}^i = \begin{cases} +1 & \text{при } x_{il} > x_{ik} \\ -1 & \text{при } x_{il} < x_{ik} \\ 0 & \text{при } x_{il} = x_{ik} \end{cases},$$

где $l < k$.

Расстояние по Кендалу вычисляется так:

$$d(X_i, X_j) = 1 - \frac{2}{n(n-1)} \sum_{l < k} \Delta_{lk}^j \Delta_{lk}^i.$$

Если компоненты обоих списков упорядочены однотипно, то имеет место следующее равенство:

$$\Delta_{lk}^i = \Delta_{lk}^j$$

для $\forall l, k$ и результат суммирования равен половине числа размещений из n по два. Число размещений из n по два равно:

$$A_n^2 = n(n-1),$$

поэтому

$$d(X_i, X_j) = 1 - \frac{2}{n(n-1)} \frac{n(n-1)}{2} = 0.$$

При этом максимальное расстояние между списками равно 2. Оно получается в том случае, когда элементы списков упорядочены в противоположном порядке.

Недостаток этого метода заключается в том, что мера порядка должна быть определена на всех элементах базового множества R . Если же сравнивать эталон и ответ обучаемого, то мера порядка может быть определена только на элементах, входящих в эталон, а ответ может содержать "лишние" элементы.

В той же работе [117] А.Фор предлагает другой метод сравнения списков. Каждый список состоит из символов, входящих в один и тот же алфавит. Пусть заданы два списка:

$$X_i^T = \{x_{i1}, x_{i2}, \dots, x_{in}\} \quad \text{и} \quad X_j^T = \{x_{j1}, x_{j2}, \dots, x_{jm}\}.$$

В общем случае $n \neq m$. Задача заключается в том, чтобы определить такую функцию, которая определяла бы расстояние между этими двумя списками.

Обе последовательности определены на одном и том же алфавите, поэтому одна из них может быть преобразована в другую. Введя пустой символ λ можно записать все три варианта преобразований в виде трех операций:

подстановка:	SUBstitution	$x_i \rightarrow x_j$	SUB (x_i, x_j),
уничтожение:	DEstruction	$x \rightarrow \lambda$	DES (x, λ),
создание:	CREation	$\lambda \rightarrow x$	CRE (λ, x).

Каждому преобразованию соответствует своя цена $c()$:

$$c(x_i, x_j) \quad \text{для SUB}, \quad c(x, \lambda) \quad \text{для DES}, \quad c(\lambda, x) \quad \text{для CRE}.$$

Для обратимости проводимых преобразований требуется выполнение условия:

$$c(x, \lambda) = c(\lambda, x). \quad (3.5)$$

Для оценки расстояния между двумя списками вводится понятие полной цены последовательности преобразований как наименьшей из всех возможных цен, которые следует "уплатить" за переход от исходного списка к конечному. Искомое расстояние $\delta(X_i, X_j)$, соответствующее полной цене, является минимальным, если переход от X_1 к X_2 происходит без получения промежуточных последовательностей.

Пусть $X_i(l) = \{x_{i1}, x_{i2}, \dots, x_{il}\}$ и $X_j(k) = \{x_{j1}, x_{j2}, \dots, x_{jk}\}$, тогда, например, $X_i(5) = \{x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}\}$ и $X_j(3) = \{x_{j1}, x_{j2}, x_{j3}\}$.

Положим расстояние $D(l, k) = \delta[X_i(l), X_j(k)]$. Для перехода от $D(l-1, k-1)$ к $D(l, k)$ имеются три возможности:

- путем подстановки $X_i(l)$ и $X_j(k)$, цена которой соответствует цене подстановки $SUB(x_{il}, x_{jk})$ последних элементов в каждом списке;
- путем создания последнего элемента в списке j , за что придется уплатить цену $CRE(\lambda, x_{jk})$;
- путем уничтожения последнего элемента в списке i , цена которого будет $DES(x_{il}, \lambda)$.

Кратко эту процедуру можно описать так:

$$D(l, k) = \min \begin{cases} D(l-1, k-1) + c(x_{il}, x_{jk}) \\ D(l, k-1) + c(\lambda, x_{jk}) \\ D(l-1, k) + c(x_{il}, \lambda) \end{cases}.$$

Минимальная цена соответствует оптимальному пути.

Расстояние между списками, вычисляемое по методу Фора, удовлетворяет условиям, которые накладываются на определение расстояния:

1. $\delta(X_i, X_i) = 0$, т.к. не требуется никаких изменений;

2. $\delta(X_i, X_j) = \delta(X_j, X_i)$, т.к. последовательное проведение обратных изменений приведет нас от списка b к списку a и цена проведенных преобразований будет прежней при условии (3.5);
3. $\delta(X_i, X_j) + \delta(X_j, X_k) \leq \delta(X_i, X_k)$, т.к. список X_j лежит на пути преобразований списка X_i в список X_k или ближе.

Недостаток вышеизложенного метода заключается в том, что нет однозначно определенной верхней границы расстояния. Расстояние в 2.5 – это много или мало? Какие списки считать максимально удаленными – упорядоченные в обратном порядке или состоящие из непересекающихся подмножеств элементов? В зависимости от этого выбора максимальная цена перехода будет различной, а значит, и степень сходства списков – разная.

Для сравнения списков в данной работе предлагается метод (Д-метод), свободный от указанных недостатков. Примем $[0,1]$ за интервал изменения расстояния r таким образом, что:

- $r = 0$, если списки совпадают,
- $r = 1$, если подмножества элементов списков не пересекаются или списки упорядочены в противоположном порядке,
- $0 < r < 1$, если списки частично совпадают.

И воспользуемся сортировкой списков для определения расстояния между списками [45].

Отношение порядка на множестве элементов вводится таким образом, чтобы для любых трех значений a, b, c выполнялись следующие условия:

- справедливо одно и только одно из соотношений $a < b$, $a = b$, $a > b$ (закон трихотомии);
- если $a < b$ и $b < c$, то $a < c$ (закон транзитивности).

Эти свойства определяют математическое понятие линейного упорядочения. Упорядочение элементов выполняется с помощью сортировки. Задача сортировки – найти такую перестановку элементов исходного множества, после которой они расположились бы в неубывающем порядке.

Напомним, что *перестановкой* конечного множества называется некоторое расположение его элементов в ряд.

Пусть $a_1 a_2 \dots a_n$ – перестановка множества $\{1, 2, \dots, n\}$. Если $i < j$ и $a_i > a_j$, то пара (a_i, a_j) называется *инверсией перестановки*. Например, перестановка 1 3 4 2 имеет три инверсии: (3, 1), (3, 2) и (4, 2). Каждая инверсия – это пара элементов, "нарушающих порядок"; следовательно, единственная перестановка, не нарушающая порядок – это отсортированная перестановка 1, 2, ..., n.

Таблицей инверсий перестановки $a_1 a_2 \dots a_n$ называется последовательность чисел $i_1 i_2 \dots i_n$, где i_j – число элементов, *больших* j и расположенных *левее* j . (Другими словами, i_j – число инверсий, у которых второй элемент равен j). По определению:

$$0 \leq i_1 \leq n-1, 0 \leq i_2 \leq n-2, \dots, i_n = 0. \quad (3.6)$$

Таблица инверсий единственным образом определяет соответствующую перестановку. Из любой таблицы инверсий, удовлетворяющей условию (3.6), можно однозначно восстановить исходную перестановку.

Исходя из свойства (3.6), для любой перестановки $a_1 a_2 \dots a_n$ количество шагов, за которые происходит ее упорядочение, равно сумме значений в таблице инверсий или числу инверсий (K_i).

Для перестановки $a_1 a_2 \dots a_n$ максимальное число шагов, требующихся для упорядочения входящих в нее элементов, придется проделать в том случае, когда элементы данной перестановки расположены в обратном порядке. Если проводить упорядочение путем попарных перестановок элементов, то максимальное число перестановок n элементов определяется по формуле:

$$K_n = \frac{n(n-1)}{2}. \quad (3.7)$$

На основании формулы (3.7) расстояние r между списками можно определить следующим образом:

$$r = \frac{K_i}{K_n}. \quad (3.8)$$

Определение расстояния между списками a и b путем сортировки одного из списков в соответствии с отношением порядка, введенного на основе другого списка, удовлетворяет понятию расстояния:

1. $r(a, a) = 0$, т.к. $K_i = 0$.
2. $r(a, b) = r(b, a)$, т.к. проведение обратных преобразований приводит к исходному списку, причем кратчайшим путем. Цена каждой попарной перестановки – 1. Переход от списка a к списку b осуществляется кратчайшим путем исходя из определения таблицы инверсий. Проведя эти перестановки в обратном порядке, мы придем от списка b к списку a , и цена проведенных перестановок будет равна количеству перестановок, что и требовалось доказать.
3. $r(a, b) + r(b, c) \leq r(a, c)$. Расстояние от списка a до списка c не может быть больше суммы расстояний $r(a, b)$ и $r(b, c)$, т.к. список b лежит на пути перехода от списка a к списку c и расстояние $r(a, c)$ складывается из этой суммы расстояний.

В соответствии с таким определением расстояния введем понятие степени сходства списков. **Степенью сходства списков** назовем величину, обратную к расстоянию между списками, и с учетом (3.3) и (3.8) получим:

$$\delta = 1 - \frac{K_i}{K_n}, \quad (3.9)$$

где K_i – количество попарных перестановок (инверсий) анализируемого списка, K_n – максимальное количество инверсий для списка длиной n .

Величина степени сходства списков δ_2 , определяемая по формуле (3.9), также удовлетворяет заданным ограничениям:

- если $S_a \equiv S_e$, то $K_i = 0$ и $\delta_2 = 1$;
- если $S_a \cap S_e = \emptyset$, то $K_i = K_n$ и $\delta_2 = 0$;
- иначе $0 < \delta_2 < 1$.

В ситуации, когда исходный и эталонный списки состоят из одних и тех же элементов, формулы (3.9) достаточно для сравнения списков. Отношение порядка вводится на основании упорядочения элементов эталонного списка.

В общем ответ S_a и эталон S_e определены на исходном множестве элементов R :

$$R = \{a_1, a_2, \dots, a_k, b_1, b_2, \dots, b_m\},$$

$$S_e = \{a_1, a_2, \dots, a_k\} \subset R,$$

$$S_a \subset R.$$

Если анализируемый список S_a содержит элементы $\{b_i\}$, то для этих элементов не может быть определено отношение порядка. Поэтому в общем случае определение степени сходства списков производится в два этапа.

На первом этапе списки рассматриваются как множества и определяется величина сходства δ_1 между множествами S_a и S_e по формуле (3.4).

Далее происходит замена элементов b_i , входящих в список S_a , на специальный элемент λ , который равен любому другому элементу. Это необходимо для того, чтобы иметь возможность производить операции сравнения элементов списков, т.к. отношение порядка вводится только на множестве элементов $\{a_i\}$, а для элементов b_i оно не определено. В том случае, если длина списков разная, более короткий список дополняется справа элементами λ . В результате получаем списки S'_a и S'_e одинаковой длины, содержащие только те элементы, для которых определено отношение порядка.

На втором этапе определяется сходство списков S'_a и S'_e . Это производится путем сортировки списка S'_a в соответствии с отношением порядка, введенным на основании списка S'_e . Степень сходства списков δ_2 определяется по формуле (3.9).

Получение окончательной величины степени сходства списков может быть выполнено одним из следующих способов:

$$\delta = (\delta_1 + \delta_2)/2 \quad (3.10)$$

$$\delta = \max(\delta_1, \delta_2) \quad (3.11)$$

$$\delta = \min(\delta_1, \delta_2) \quad (3.12)$$

$$\delta = \delta_1 * \delta_2 \quad (3.13)$$

Графики этих функций представлены на рис. 3.1.

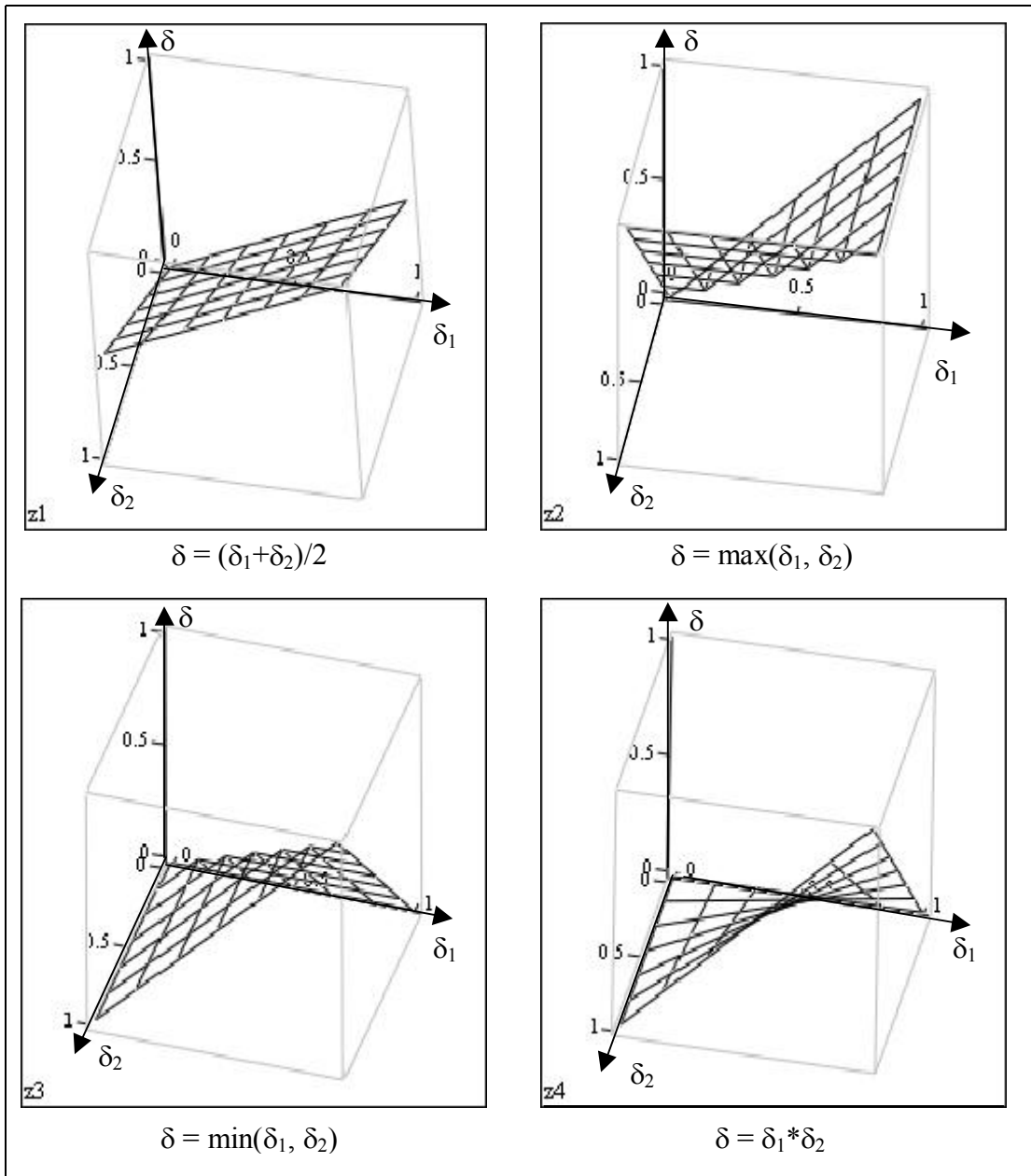


Рис. 3.1. Графики степени сходства списков δ при различных методах подсчета

Анализ этих графиков показывает, что первые две функции (3.10) и (3.11) являются более либеральными, чем две последние (3.12) и (3.13). Использование функций подсчета (3.10) – среднее арифметическое – и (3.13)

– произведение – являются более предпочтительным, чем (3.11) – максимум – и (3.12) – минимум, – т.к. учитывает обе оценки: и степень сходства множеств, и степень сходства списков. В случае же применения функции "максимум" могут возникнуть такие ситуации, при которых неверный ответ будет оценен как правильный. Например, если $S_a = \{1, 2, 3, 4\}$, а $S_e = \{4, 3, 2, 1\}$, то $\delta_1 = 1$, $\delta_2 = 0$, $\delta = \max\{1, 0\} = 1$. Поэтому использование этого способа не рекомендуется.

3.3.3.3. Множества списков

Рассмотрим случай, когда элементами множества S_e являются множества r_i или списки s_i ($i = 1, \dots, N$). Тогда оценка ответа S_a может быть выполнена в два этапа:

1. каждый элемент (множество r_i или список s_i) сравнивается со всеми элементами эталонного множества S_e (по правилам сравнения множества или списка соответственно). В качестве степени сходства этого элемента с элементами эталонного множества выбирается максимальная степень сходства между элементами двух множеств. (Элемент эталонного множества, наиболее близкий к рассматриваемому элементу ответа, исключается из дальнейшего рассмотрения.) Из этих степеней сходства формируется числовой вектор L . Длина этого вектора равна мощности N . Если элементу r_i (s_i) множества S_a не соответствует никакой элемент эталонного множества S_e , то i -й элемент вектора L равен 0.
2. степень сходства ответа и эталонного множества вычисляется по вектору L , например, как среднее арифметическое элементов вектора L .

3.3.3.4. Списки множеств

Оценка списка множеств (списков) также может выполняться в два этапа:

1. На первом этапе (аналогично сравнению множества списков) происходит сопоставление элементов списка S_a элементам списка S_e . Формируется вектор соответствия K . Элемент k_i вектора соответствия K равен номеру

наиболее близкого элемента списка S_e или числу, превышающему количество элементов в эталонном списке. Степень сходства δ_1 множества элементов ответа вычисляется, например, как среднее арифметическое степеней соответствия сопоставляемых списков.

2. Полученный список соответствия K упорядочивается путем попарных перестановок. Степень сходства δ_2 вычисляется по правилам определения степени сходства списков (3.9). Окончательно степень сходства ответа и эталонного множества вычисляется по одной из формул (3.10)–(3.13).

Существует и более простой с точки зрения реализации способ. Список S_a считается упорядоченным, и каждый i -й элемент списка S_a сопоставляется i -му элементу списка S_e . Степень сходства списков δ равна, например, среднему арифметическому или произведению степеней сходства элементов списков.

3.3.4. Анализ ответа в виде арифметического выражения

Арифметическое выражение является обобщенным представлением числового ответа. Арифметическое выражение может быть вычислимым значением и формулой. Для вычислимого значения должны быть определены эталон (правильное значение) и допустимая погрешность ε . Один из вариантов использования погрешности заключается в том, что если ответ находится в ε -окрестности эталона:

$$(E - \varepsilon) \leq A \leq (E + \varepsilon),$$

где E – эталон, A – полученный ответ, то ответ считается правильным, иначе ответ неверен.

Если выражение требуется рассматривать как формулу, то можно воспользоваться алгоритмом унификации, разработанным в 1966 г. Ж Питра и независимо от него Дж. Робинсоном [61]. Этот алгоритм позволяет определить идентичность любых двух выражений и не зависит от формальной системы, к которой применяется.

3.3.5. Текстовые ответы

Использование ответов, вводимых в свободной текстовой форме, является самой естественной и наиболее сложной задачей при организации системы контроля знаний.

Задача интеллектуальной обработки текстов на естественном языке впервые была поставлена на рубеже 60х–70х гг. С тех пор было предпринято множество различных попыток ее решения (см., например, [35, 48, 58, 81]), созданы десятки экспериментальных программ, способных вести диалог с пользователем на естественном языке. Однако широкого распространения такие системы пока не получили – как правило, из-за невысокого качества распознавания фраз, жестких требований к синтаксису “естественного языка”, а также больших затрат машинного времени и ресурсов, необходимых для их работы [125]. Практически во всех системах машинного понимания текста используется ограниченный естественный язык, поскольку полной формальной модели ни для одного естественного языка пока не создано.

Тем не менее, естественно-языковые средства общения человека с ЭВМ постоянно развиваются, оставаясь одним из наиболее оптимальных способов построения пользовательского интерфейса к сложным информационным системам. Одна из последних чрезвычайно перспективных работ в этой области – проект семантического анализатора, выполненный в Российском Фонде Фундаментальных Исследований под руководством Д.А. Поспелова. В рамках этого проекта впервые разработана программная система, выполняющая весь цикл задач анализа текста, начиная от морфологического анализа и заканчивая построением базы знаний, содержащей понятия предметной области и связи между ними. К сожалению, пока это только исследовательский прототип. Поэтому, если мы говорим об оболочках АОС, то здесь приходится пользоваться более простыми, но формализованными вариантами.

Один из методов организации работы с текстовыми ответами – шаблоны. Под шаблоном понимается анкета, содержащая произвольное количество полей и пояснения к ним. Приведем пример вопроса с шаблоном ответа:

<u>Вопрос:</u>	<i>Какие виды систем управления базами данных Вы знаете?</i>	
<u>Шаблон:</u>	<u><i>по степени универсальности</i></u>	общего назначения и специализированные
	<u><i>по методам размещения данных</i></u>	централизованные и распределенные
	<u><i>по модели данных</i></u>	реляционные, сетевые, иерархические

Пояснения к полям ввода определяют, какая информация должна быть введена в данное поле. Обучаемый может заполнить не все поля, но при этом его оценка, естественно, уменьшится. Для того чтобы избежать ввода слов, неизвестных системе, целесообразно предоставить обучаемому словарь, содержащий все слова, входящие в эталонные ответы для данного теста (или группы тестов).

Существуют и другие способы анализа текстовых ответов, основанные на использовании некоторых правил унифицированного описания текстового ответа. Один из них реализован в системе AOSMICRO [23]. При задании эталона текстового ответа преподаватель указывает ключевые слова, которые должны присутствовать в ответе обучаемого, и слова, которых там быть не должно. Можно также использовать шаблоны "один любой символ" и "произвольное количество любых символов". На основании этой информации система анализирует ответ обучаемого и определяет его правильность. Это наиболее мощная (по сведениям автора) процедура анализа текстовых ответов в отечественных АОС. Но использование этой процедуры предполагает некоторую предварительную обработку эталонного ответа – разбиение его на словосочетания. Эта работа выполняется, естественно, преподавателем.

Таким образом, подготовленный для анализа эталонный ответ представляется набором элементов, каждый из которых может быть словом или словосочетанием. Если рассмотреть этот набор как двухуровневую структуру, то можно свести текстовый ответ к типу СПИСОК МНОЖЕСТВ или СПИСОК СПИСКОВ. При использовании Д-метода для определения правильности ответов типа СПИСОК мы получим формальную процедуру определения правильности текстового ответа, которая по своим характеристикам не уступает процедуре, реализованной в AOSMICRO. Таким образом, анализ текстового ответа можно свести к формальному анализу списка или множества.

Замечание: для улучшения качества анализа текстового ответа, естественно, может быть предусмотрена некоторая предобработка. Она может включать грамматический анализ (для устранения ошибок ввода), удаление вспомогательных слов и словосочетаний (типа "видимо", "я думаю", "конечно" и пр.), анализ слов ответа и эталона на синонимичность и др. Эту предобработку целесообразно выполнять с помощью внешней процедуры, чтобы не перегружать систему контроля знаний вспомогательными функциями и иметь возможность использовать произвольную программу для анализа.

3.3.6. Графические формы задания ответов

Использование графических вопросов и ответов значительно расширяет возможности системы контроля знаний обучаемых, поэтому вопрос о распознавании графических образов мы рассмотрим отдельно.

3.3.6.1. Распознавание образов в обучающих системах

Задача распознавания образов заключается в том, чтобы классифицировать объект, т.е. определить, что он относится к данному классу и не относится к другому [119]. Существуют различные методы распознавания графических образов, которые условно можно разбить на два класса:

- Моделирующие методы (к ним относятся ассоциативные методы, бионические методы, перцептроны, методы моделирования физических аналогов, например, метод потенциалов, спиновые стекла и др.).
- Формальные методы (например, структурные и синтаксические методы).

Для решения задачи распознавания образов в обучающих системах больше подходят формальные методы, потому что классы распознаваемых образов определены и могут быть описаны формальными категориями.

Выделяют три основных способа описания и разделения классов образов [113]:

- принцип перечисления членов класса.

Задание класса перечислением входящих в него элементов предполагает реализацию автоматического распознавания образов путем сравнения с эталоном.

- принцип общности свойств.

Задание класса с помощью свойств, общих для его членов, предусматривает реализацию процесса автоматического распознавания образов путем выделения подобных признаков и работы с ними. Принцип общности свойств лежит в основе процессов распознавания, реализуемых методами теории формальных языков.

- принцип кластеризации.

Если образы некоторого класса представляют собой векторы, компонентами которых являются действительные числа, этот класс можно рассматривать как кластер и выделять только его свойства в пространстве кластеров.

Существуют три основные методологии для реализации этих способов:

- эвристические методы.

В основе этих методов лежат опыт и интуиция человека. Система включает набор специфических процедур, разработанных применительно к конкретным задачам распознавания.

- математические методы.

Они основаны на правилах классификации, которые формулируются и выводятся в рамках определенного математического формализма с помощью принципов общности свойств и кластеризации. Различают

детерминистические и статистические методы (последние используют методы математической статистики).

- лингвистические (синтаксические) методы.

В том случае, если описание образов осуществляется с помощью непроеизводных элементов (подобразов) и их отношений, то применяется лингвистический или синтаксический подход с использованием принципа общности свойств [118]. Образ можно описать с помощью иерархической структуры подобразов, подобно синтаксической структуре языка. Это позволяет применять теорию формальных языков. Предполагается, что грамматика образов содержит конечные множества элементов, называемых переменными, непроеизводными элементами и правилами подстановки.

В распознавании графических образов синтаксическими методами можно выделить три основных этапа: предобработка (кодирование, аппроксимация, фильтрация, восстановление и улучшение объекта), представление объекта (сегментация и выделение непроеизводных элементов) и анализ. Содержимое этих этапов зависит от того, какой метод распознавания используется – дискриминантный или структурный.

В обучающих системах постановка задачи распознавания графических образов отличается от классической.

Когда речь идет об обучающих системах, распознаваемым объектом является ответ обучаемого на поставленный вопрос. Для того чтобы обучаемый мог дать ответ в графическом виде, необходимо предоставить инструментарий (например, графический редактор). Таким образом, задача сильно упрощается. В качестве элементов графического изображения выступают непроеизводные элементы, а операции, производимые над ними, становятся операциями соединения этих непроеизводных элементов в изображение (объект). Поэтому этап предобработки упраздняется, а этап представления становится чисто техническим. Этап анализа представляет собой простейшую форму распознавания образов – сравнение с эталоном.

Многообразие задач, решение которых может быть представлено в графическом виде, не позволяет рассмотреть все возможные варианты анализа графических изображений или предложить какой-либо универсальный подход к анализу без усложнения инструментария и самих методов анализа изображений. Например, если для качественного анализа графиков отмечаются точки экстремумов и характер линий (прямые или дуги), то проверять правильность электронной схемы целесообразнее с помощью функции, которую она должна реализовать. Поэтому мы рассмотрим подробнее только задачу качественного анализа графиков как одну из наиболее распространенных.

3.3.6.2. Задача распознавания графиков

Когда преподаватель предлагает обучаемому изобразить какой-либо график, обучаемый должен правильно отразить на графике точки экстремума и тенденцию графика. Это означает, что анализ графиков должен быть качественным и основываться на тех же контрольных элементах: привязка к координатам, точки экстремума, тенденция.

Сформируем множество производных элементов для задачи анализа графиков. Определение производных элементов сильно зависит от характера образов, от предметной области и от набора доступных технических средств. Общее решение проблемы выбора производных элементов пока не найдено, но можно предложить несколько общих рекомендаций:

1. Производные элементы должны служить основными элементами образов и обеспечивать сжатое и адекватное описание образов.
2. Выделение и распознавание производных элементов должно быть простым и осуществляться несинтаксическими методами.

Очевидно, что эти два требования противоречат друг другу, поэтому в реальных задачах приходится искать компромисс между компактностью описания и простотой выделения производных элементов.

В некоторых случаях (и рассматриваемый относится к их числу) требуется, чтобы производные элементы содержали семантическую информацию, важную для конкретного приложения. В нашем примере такой информацией являются координаты. Выделим производные элементы для нашей задачи:

- точка экстремума, в которой производная 1-го порядка равна 0;
- линии с разными углами наклона (рис. 3.2).

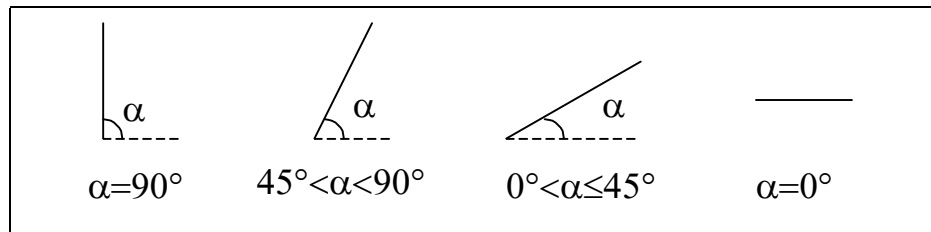


Рис. 3.2. Производные элементы для задачи качественного анализа графиков

Инструментарий для такой системы должен позволять рисовать прямые линии, позиционировать курсор в точке на линии и отмечать ее координаты. Для формирования эталонного изображения преподавателю предоставляется тот же инструментарий. Если оба изображения (эталонное и ответ обучаемого) одинаковым образом перевести во внутреннее представление, где каждому производному элементу ставится в соответствие некоторое мнемоническое и/или цифровое обозначение, то в результате объект (график) будет представлен цепочкой (списком) элементов. Таким образом, можно свести графический ответ к ответу типа СПИСОК. Но при этом возникают некоторые сложности.

График – это двумерное изображение, поэтому его нельзя описать цепочкой символов, используя только операцию конкатенации. Например, если ограничиться указанными на рис. 3.2. производными элементами, то представление образов, приведенных на рис. 3.3, будет одинаковым.

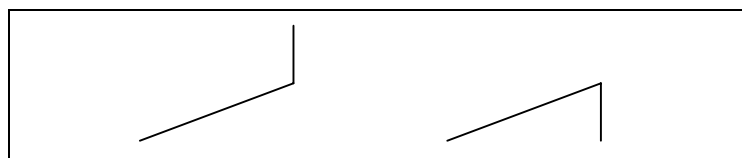


Рис. 3.3. Примеры графиков

Путей решения возникшей проблемы два: усложнить набор производных элементов или ввести дополнительные операции, указывающие местоположение каждого следующего элемента относительно предыдущего, например, "над предыдущим", "слева от предыдущего" и т.д. Выберем первый путь, чтобы сохранить представление описания образа в виде цепочки мнемонических обозначений, и дополним набор производных элементов (рис. 3.4). Таким образом, в производные элементы добавлена синтаксическая информация, а описание образа не усложнилось.

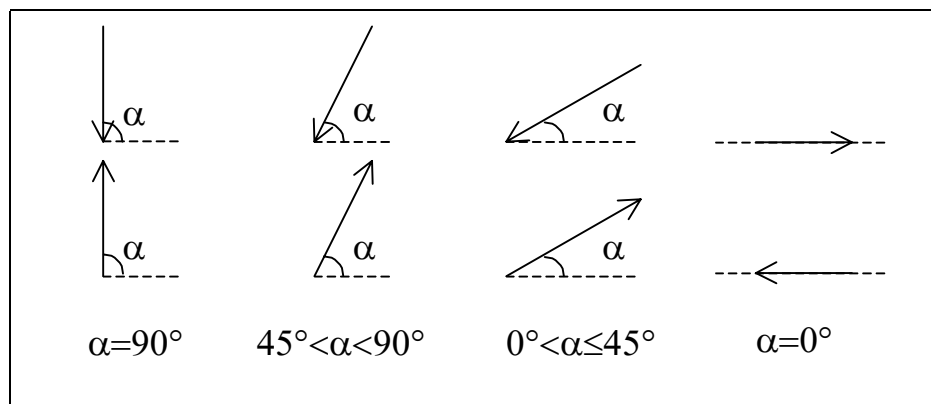


Рис. 3.4. Дополненное множество производных элементов

Сведение графической формы представления ответа к списку позволяет нам использовать для качественного анализа графика предложенный Д-метод и получать дифференцированные оценки ответов обучаемого.

Несмотря на чрезвычайную простоту, почти примитивность подобного варианта представления графиков, этот подход дает качественно иной результат по сравнению с выбором одного из нескольких готовых вариантов ответов.

3.4. Расчет вероятности случайного ввода правильного ответа

Использование выборочных ответов сопряжено с возможностью неадекватной оценки знаний обучаемого в случае случайного ввода правильного ответа. Оценим эту вероятность для предложенных ответов типа

МНОЖЕСТВО, СПИСОК, МНОЖЕСТВО СПИСКОВ, СПИСОК МНОЖЕСТВ.

Расчет вероятности случайного ввода правильного ответа проведем при условиях, что все ответы считаются равновероятными и оцениваются по двухбалльной системе. Все вопросы в тесте будем считать независимыми.

3.4.1. Ответ типа МНОЖЕСТВО

Для ответа типа МНОЖЕСТВО рассмотрим две схемы:

1. " N вариантов ответов – из них один правильный". Эта вероятность равна:

$$q_1^s = \frac{1}{N}. \quad (3.14)$$

2. " N вариантов ответов – из них k правильных" ($0 \leq k \leq N$) вероятность ввода каждого ответа для выборки длиной k равна:

$$q_k^s = \frac{1}{M}, \quad (3.15)$$

где M – общее количество возможных ответов, которое рассчитывается на основании формулы для выборки из N по k :

$$C_N^k = \frac{N!}{k!(N-k)!}.$$

Общее число возможных ответов M является суммой всех C_N^k :

$$M = \sum_{k=0}^N \frac{N!}{k!(N-k)!}.$$

Из перечислительной комбинаторики известно [105], что:

$$\sum_{k=0}^N C_N^k = 2^N.$$

Значение k неизвестно отвечающему, поэтому будем считать все значения k равновероятными. Только один из возможных ответов является правильным, и окончательно вероятность случайного ввода правильного ответа равна:

$$q_k^s = \frac{1}{2^N}. \quad (3.16)$$

3.4.2. Ответ типа СПИСОК

Для ответа типа СПИСОК также рассмотрим две схемы:

1. "Все N вариантов входят в ответ". Для этой схемы при прежней вероятности ввода каждого ответа (3.15) количество M возможных ответов равно:

$$M = N!,$$

отсюда вероятность случайного ввода правильного ответа для списка:

$$q_N^L = \frac{1}{N!}. \quad (3.17)$$

2. Для схемы " N вариантов ответов – из них k правильных" минимальное значение k равно 2, т.к. список меньшей длины не имеет смысла. Количество M различных вариантов равно:

$$M = \sum_{k=2}^N \frac{N!}{(N-k)!}.$$

Правильным является единственный вариант, поэтому вероятность случайного ввода правильного ответа равна:

$$q_k^L = \frac{1}{\sum_{k=2}^N \frac{N!}{(N-k)!}}. \quad (3.18)$$

Изменение вероятности случайного ввода правильного ответа в зависимости от количества вариантов для разных схем показано в табл. 3.1 (формулы (3.14), (3.16)-(3.18)), соответствующие графики приведены на рис. 3.5.

Таблица 3.1. Вероятности случайного ввода правильного ответа для различных одноуровневых схем

Количество ответов, N	Тип и схема ответа			
	МНОЖЕСТВО, "N вариантов – 1 правильный"	МНОЖЕСТВО, "N вариантов – k правильных"	СПИСОК, "N вариантов – нет лишних"	СПИСОК, "N вариантов – есть лишние"
3	0.33	0.125	0.167	0.08
4	0.25	0.063	0.042	0.017
5	0.20	0.032	0.008	0.003
6	0.17	0.016	0.0014	0.0005
7	0.14	0.008	0.0002	0.00007
8	0.13	0.004	0.000025	0.000009

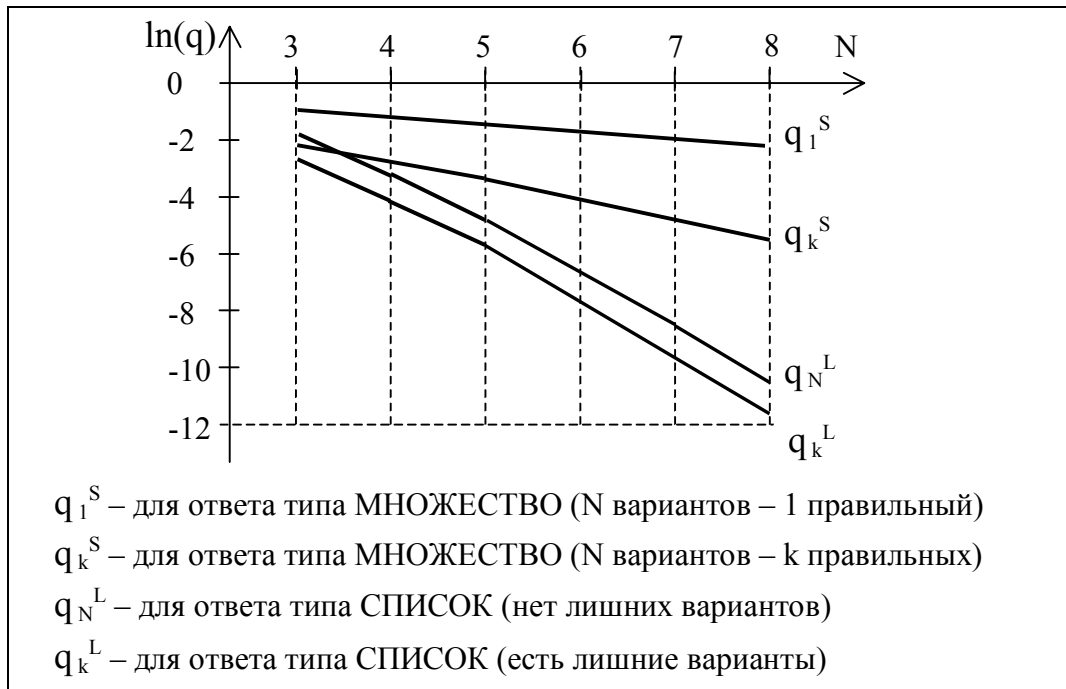


Рис. 3.5. Графики зависимости логарифмов вероятностей случайного ввода правильного ответа от количества элементов в ответе

На основании анализа графиков можно сделать следующие выводы:

- ответ, построенный по схеме " N вариантов – k правильных", значительно предпочтительнее ответа, построенного по схеме " N вариантов – 1 правильный";
- ответ типа СПИСОК, естественно, имеет меньшую вероятность случайного ввода правильного ответа, чем ответ типа МНОЖЕСТВО;
- при наличии "лишних" вариантов в ответе типа СПИСОК его оценка улучшается еще приблизительно в 2-3 раза.

3.4.3. Ответ типа СПИСОК МНОЖЕСТВ

Двухуровневую схему СПИСОК МНОЖЕСТВ рассмотрим при дополнительном ограничении: ответ должен содержать все N вариантов ответа, соответствующих вопросу. Количество множеств обозначим J : $2 \leq J \leq (N-1)$. Вероятность случайного ввода правильного ответа равна:

$$q_N^{LS} = \frac{m}{M}, \quad (3.19)$$

где M – общее количество вариантов ответа, m – количество правильных ответов. Разбиение на множества проводится для упорядоченного списка. Количество различных разбиений для одного варианта упорядочения равно:

$$A(N) = \sum_J \Psi_J(N),$$

где функция Ψ определяется рекурсивно:

$$\Psi_J(N) = \sum_{i=1}^{N-1} \Psi_{J-1}(N-i) \quad (3.20)$$

при следующих начальных условиях:

1. $\Psi_1(N) = 1, \forall N$;
2. $\Psi_k(N) = 1, k = N$;
3. $\Psi_k(N) = 0, k > N$.

Разбиение множества на подмножества можно рассматривать аналогично разложению целого числа n в виде упорядоченной суммы положительных целых чисел. Используя формулы пересчетной комбинаторики для разложений целого числа [105], можно перейти от рекурсивной функции к считающей функции. Известно, что общее количество разложений $I(n)$ целого числа n равно:

$$I(n) = 2^{n-1},$$

а количество k -разложений, в которых разложение содержит в точности k слагаемых, определяется как выборка из $(n-1)$ по $(k-1)$:

$$I(n) = C_{n-1}^{k-1}.$$

С учетом того, что количество способов упорядочения исходного списка равно $N!$, общее количество вариантов ответов M_0 :

$$M_0 = N!(2^{N-1} - 2).$$

Поправка -2 введена для того, чтобы исключить из рассмотрения разбиения исходного множества на одно и на N множеств, т.к. такие разбиения идентичны списку элементов.

Теперь исключим из общего набора списков повторяющиеся варианты (например, $\{(1, 2), (3, 4)\}$ и $\{(2, 1), (3, 4)\}$, построенные для разных способов

упорядочения исходного множества вариантов ответов), и получим M – количество вариантов ответов без повторов:

$$M = \frac{N!(2^{N-1} - 2)}{N - 1},$$

и окончательно формула (3.19) примет такой вид:

$$q_N^{LS} = \frac{N - 1}{N!(2^{N-1} - 2)}. \quad (3.21)$$

Вероятности случайного ввода правильного ответа в зависимости от количества вариантов для СПИСКА МНОЖЕСТВ показаны в табл. 3.2.

Таблица 3.2. Вероятности случайного ввода правильного ответа для схемы СПИСОК МНОЖЕСТВ

Количество ответов, N	Количество вариантов ответов	Вероятность q_N^{LS}
3	12	0.17
4	144	0.02
5	1680	0.002
6	21600	0.00023
7	312480	0.000019
8	5080320	0.0000014

3.4.4. Ответ типа МНОЖЕСТВО СПИСКОВ

Вероятность случайного ввода правильного ответа для варианта типа МНОЖЕСТВО СПИСКОВ рассмотрим при том же ограничении, что ответ должен содержать все N вариантов, соответствующих вопросу. Количество списков обозначим L ($2 \leq L \leq N/2$). Все списки должны иметь длину не менее двух, количество списков – также не менее двух, отсюда получаем, что $N > 3$. Разбиение на группы (списки) проводится для упорядоченного списка, количество перестановок – $N!$. Количество разбиений на группы можно определить так:

$$A(N) = \sum_J \Psi_J(N - J),$$

где функция определена рекурсивно аналогично (3.20) при тех же начальных условиях.

Для того чтобы убрать лишние варианты (типа $\{1,2\}$, $\{3,4,5\}$ и $\{3,4,5\}$, $\{1,2\}$), полученные для разных перестановок, необходимо учесть повторение.

Оно определяется как количество перестановок групп и равно $J!$. Отсюда получаем общее количество вариантов ответа (без повтора):

$$M = N! \sum_J \frac{\Psi_J(N-J)}{J!}.$$

Правильным является единственный вариант, поэтому окончательно эту вероятность можно определить как:

$$q_N^{SL} = \frac{1}{N! \sum_{J=2}^{N/2} \frac{\Psi_J(N-J)}{J!}}. \quad (3.22)$$

Вероятности (3.22) случайного ввода правильного ответа в зависимости от количества вариантов для МНОЖЕСТВА СПИСКОВ показаны в табл. 3.3. На рис. 3.6 приведены графики вероятности случайного ввода правильного ответа для СПИСКА МНОЖЕСТВ и МНОЖЕСТВА СПИСКОВ.

Таблица 3.3. Вероятности случайного ввода правильного ответа для схемы МНОЖЕСТВО СПИСКОВ

Количество ответов, N	Количество вариантов ответов	Вероятность q_N^{SL}
4	12	0.083
5	120	0.0083
6	1200	0.00167
7	12600	0.000159
8	142800	0.000021

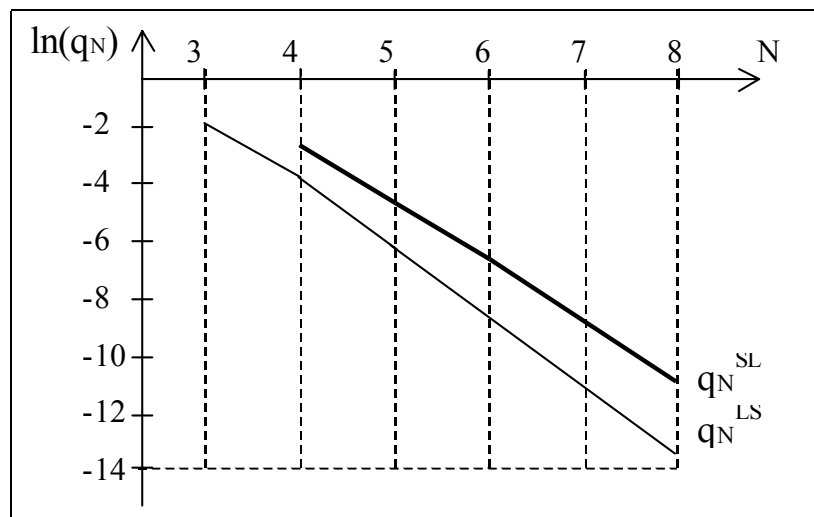


Рис. 3.6. Графики зависимости логарифмов вероятностей q_N^{SL} (МНОЖЕСТВО СПИСКОВ) и q_N^{LS} (СПИСОК МНОЖЕСТВ) от количества ответов N

Сравнение двухуровневых схем показало, что с точки зрения вероятности случайного ввода правильного ответа схема СПИСОК МНОЖЕСТВ значительно лучше схемы МНОЖЕСТВО СПИСКОВ.

3.5. Моделирование случайного ввода правильного ответа

Разработанные методы оценки ответов типа СПИСОК и МНОЖЕСТВО позволяют определять правильность ответов не по двухбалльной шкале (0, 1), а на интервале [0,1]. При использовании этих методов вероятность случайного ввода абсолютно правильного ответа соответствует формулам (3.16, 3.18, 3.21–3.22), но появляются "частично правильные ответы", оценка которых отлична от нуля. Распределение вероятностей появления всех возможных оценок в интервале [0,1] можно определить по формуле:

$$q = \frac{\sum_{i=1}^M \varphi(a_i)}{M},$$

где M – общее количество ответов, $\varphi(a_i)$ – функция определения оценки ответа a_i . Оценка ответа зависит от самого ответа (от его сходства с эталоном), и посчитать по этой формуле распределение вероятностей нельзя. Поэтому было проведено программное моделирование различных схем построения ответа для множеств вариантов разных мощностей.

Моделирование проводилось при следующих условиях:

- 1) Если $S_a = \{a_i\}$ – ответ, а $S_e = \{e_i\}$ – эталон, то $\{a_i\} \subset \{e_i\}$.
- 2) $\forall a_i, a_j: a_i \in S_a, a_j \in S_a, i \neq j \Rightarrow a_i \neq a_j$.

3.5.1. Моделирование ответа типа МНОЖЕСТВО

Моделирование ответа типа МНОЖЕСТВО проводилось для $N = 3 \div 7$ и схем " N вариантов ответа – из них 1 правильный" и " N вариантов ответа – из них k правильных".

Результаты моделирования ответа типа МНОЖЕСТВО представлены на рис. 3.7–3.10.

Примечание: на рисунках, отражающих результаты моделирования, для каждого графика указана погрешность (число на оси δ , выделенное курсивом), при использовании которой высший балл (единица по двухбалльной шкале) будет поставлен только за правильный ответ.

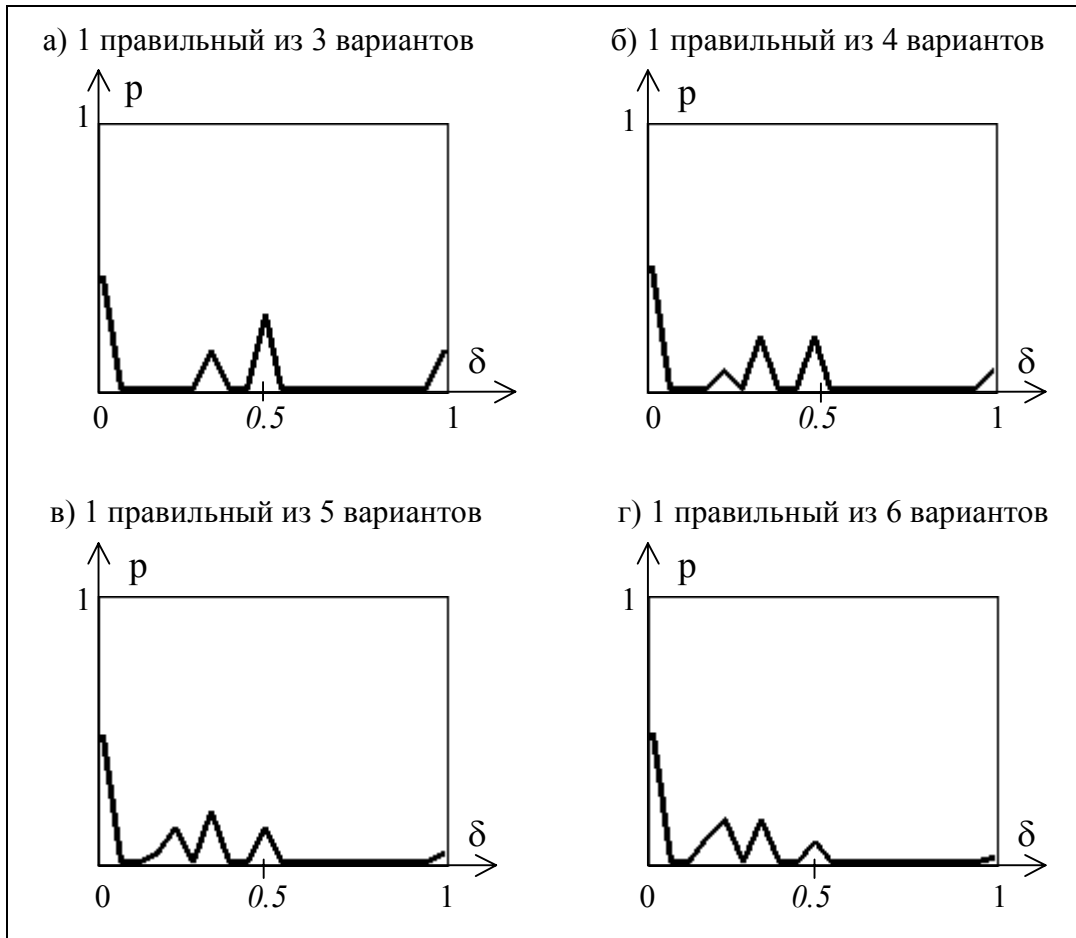


Рис. 3.7. Распределение оценок для ответов типа МНОЖЕСТВО, схема "N вариантов ответа – из них 1 правильный"

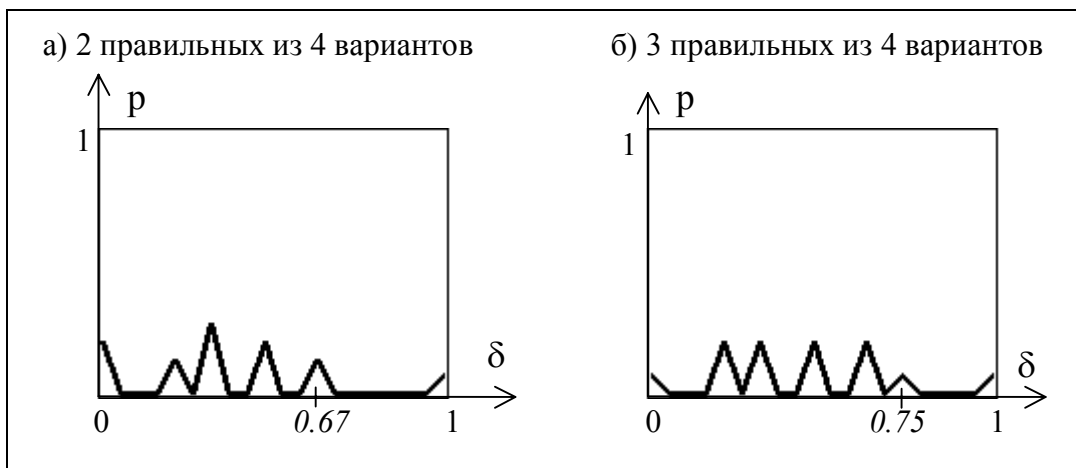


Рис. 3.8. Распределение оценок для ответов типа МНОЖЕСТВО, схема "N вариантов ответа – из них k правильных" (N=4)

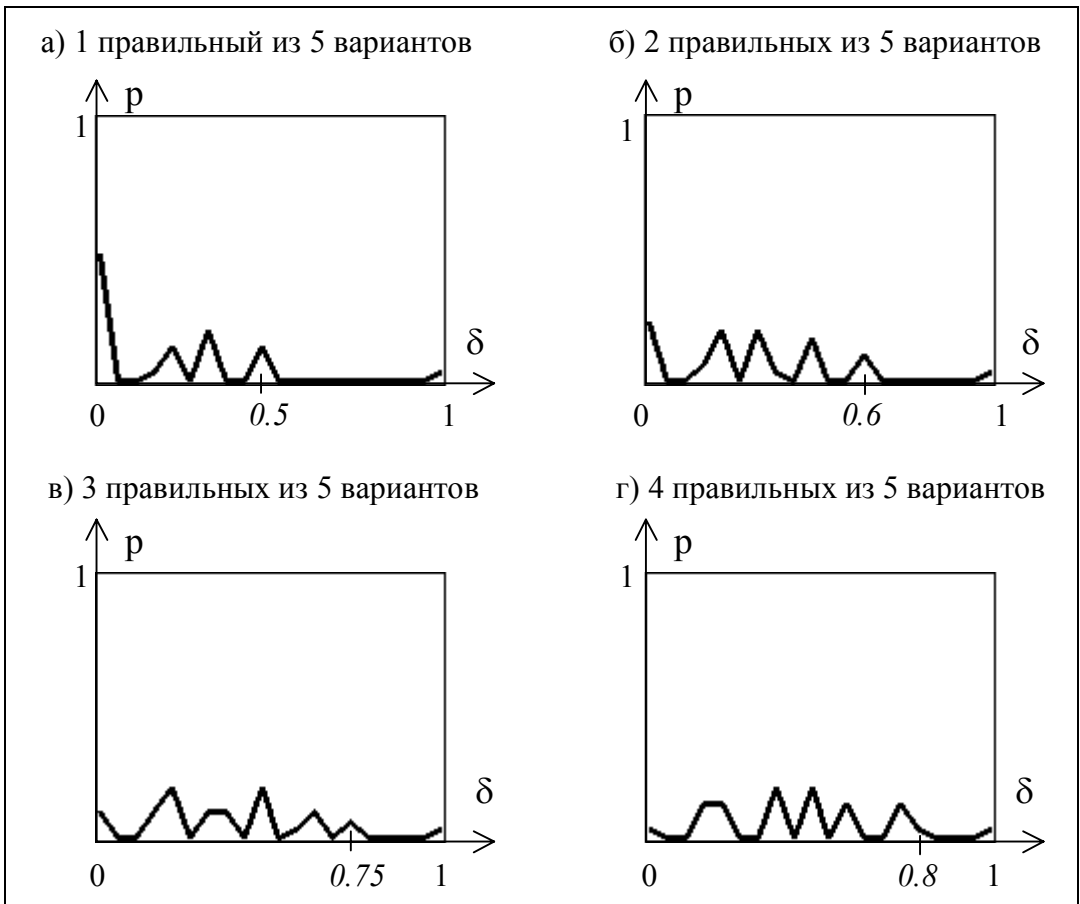


Рис. 3.9. Распределение оценок для ответов типа МНОЖЕСТВО (N=5)

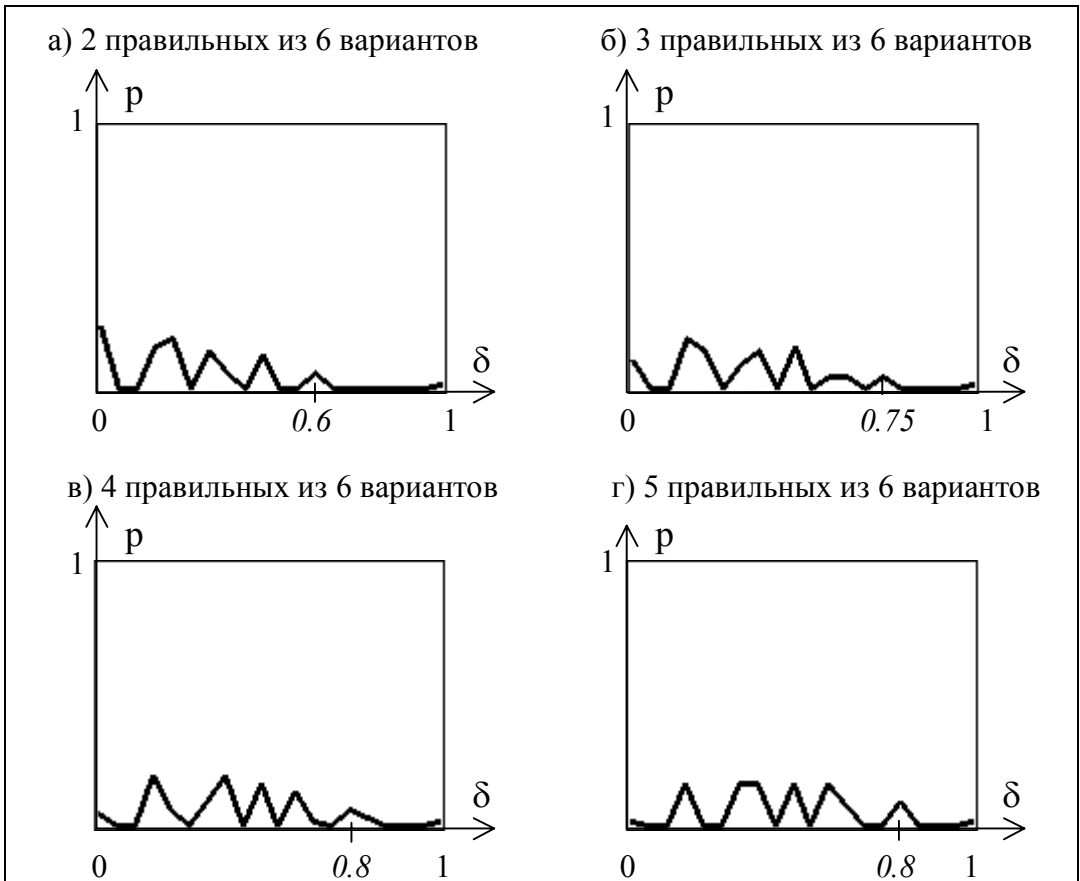


Рис. 3.10. Распределение оценок для ответов типа МНОЖЕСТВО (N=6)

Математическое ожидание MX для вероятности случайного ввода правильного ответа равно:

$$MX = \sum_k \delta_k p_k .$$

Значения математического ожидания вероятности случайного ввода правильного ответа для ответа типа МНОЖЕСТВО приведены в табл. 3.4.

Таблица 3.4. Математическое ожидание для ответа типа МНОЖЕСТВО

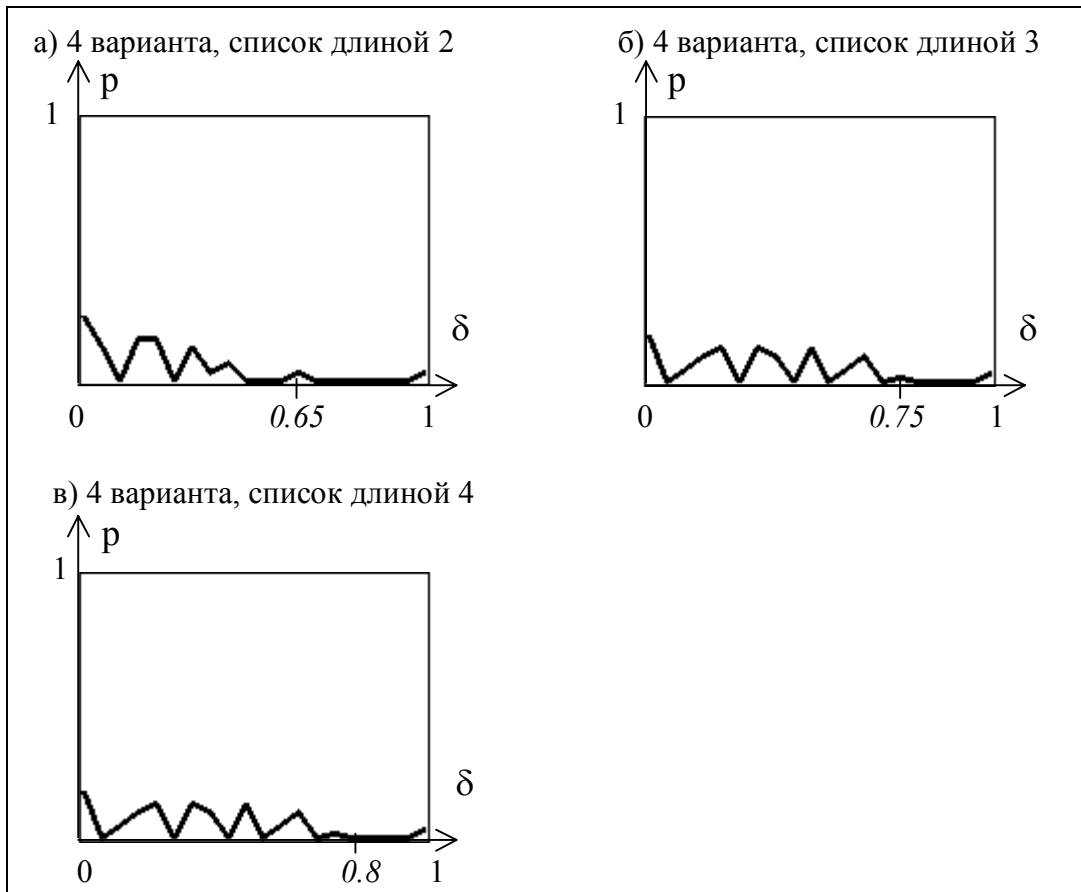
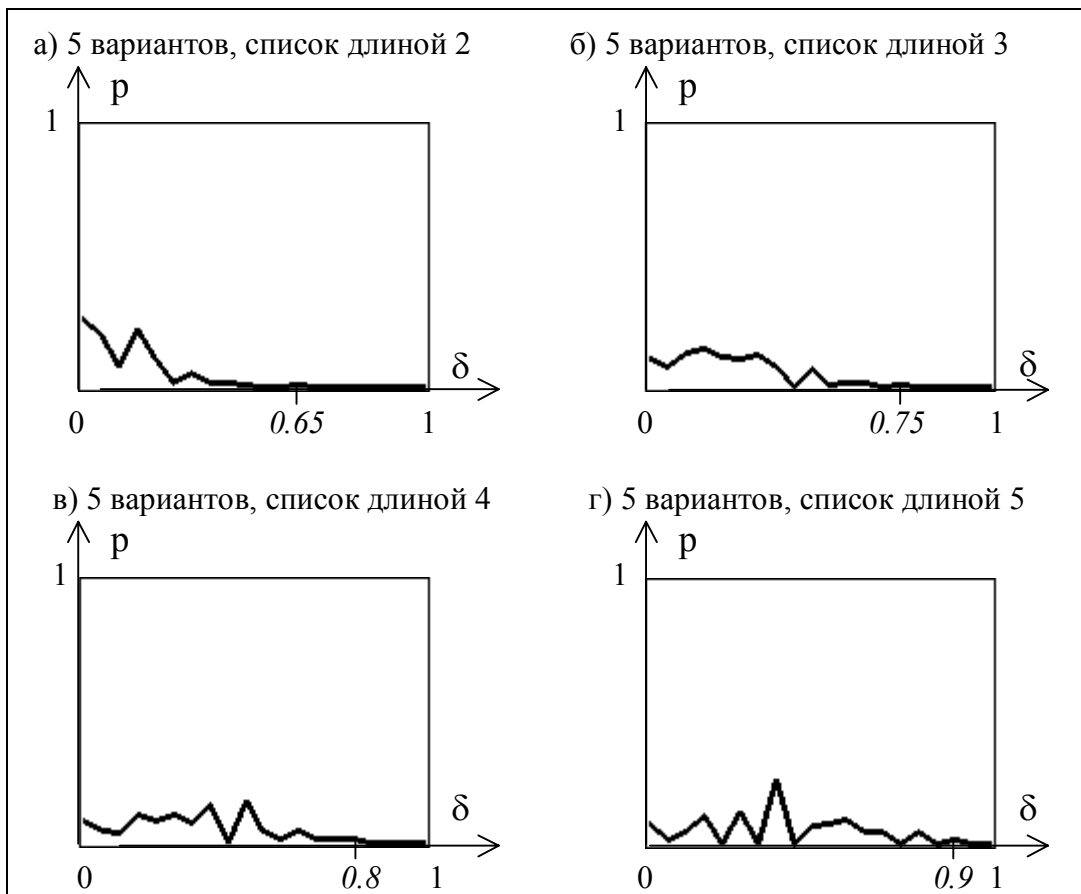
Количество правильных ответов	Количество вариантов ответов				
	3	4	5	6	7
1	0.33	0.25	0.20	0.17	0.14
2	0.48	0.38	0.32	0.27	0.24
3	—	0.47	0.40	0.35	0.32
4	—	—	0.46	0.41	0.38
5	—	—	—	0.46	0.43
6	—	—	—	—	0.47

Анализируя графики распределения оценок ответов типа МНОЖЕСТВО (рис. 3.7–3.10) и табл. 3.4 можно дать следующие рекомендации для применения таких ответов в тестах:

1. При использовании стандартной шкалы оценок (2, 3, 4, 5) и схемы " N вариантов ответа – из них 1 правильный" количество вариантов ответов должно быть не менее четырех (для $N=3$ математическое ожидание равно 0.36, что при приведении к стандартной шкале дает 3.1 балла, т.е. оценку "удовлетворительно").
2. Увеличение количества правильных вариантов ответа ведет к увеличению вероятности случайного получения более высокой оценки. Поэтому не следует использовать дифференцированную оценку ответа в тех случаях, когда необходим точный ответ и наличие "лишних" вариантов (или отсутствие требуемых) должно расцениваться как неверный ответ.

3.5.2. Моделирование ответа типа СПИСОК

Моделирование ответа типа СПИСОК проводилось для $N = 4 \div 7$ и схем " N вариантов ответа – есть лишние" и " N вариантов ответа – нет лишних". Результаты приведены на рис. 3.11–3.13, значения математического ожидания вероятности случайного ввода правильного ответа – в табл. 3.5.

Рис. 3.11. Распределение оценок для ответов типа СПИСОК ($N=4$)Рис. 3.12. Распределение оценок для ответов типа СПИСОК ($N=5$)

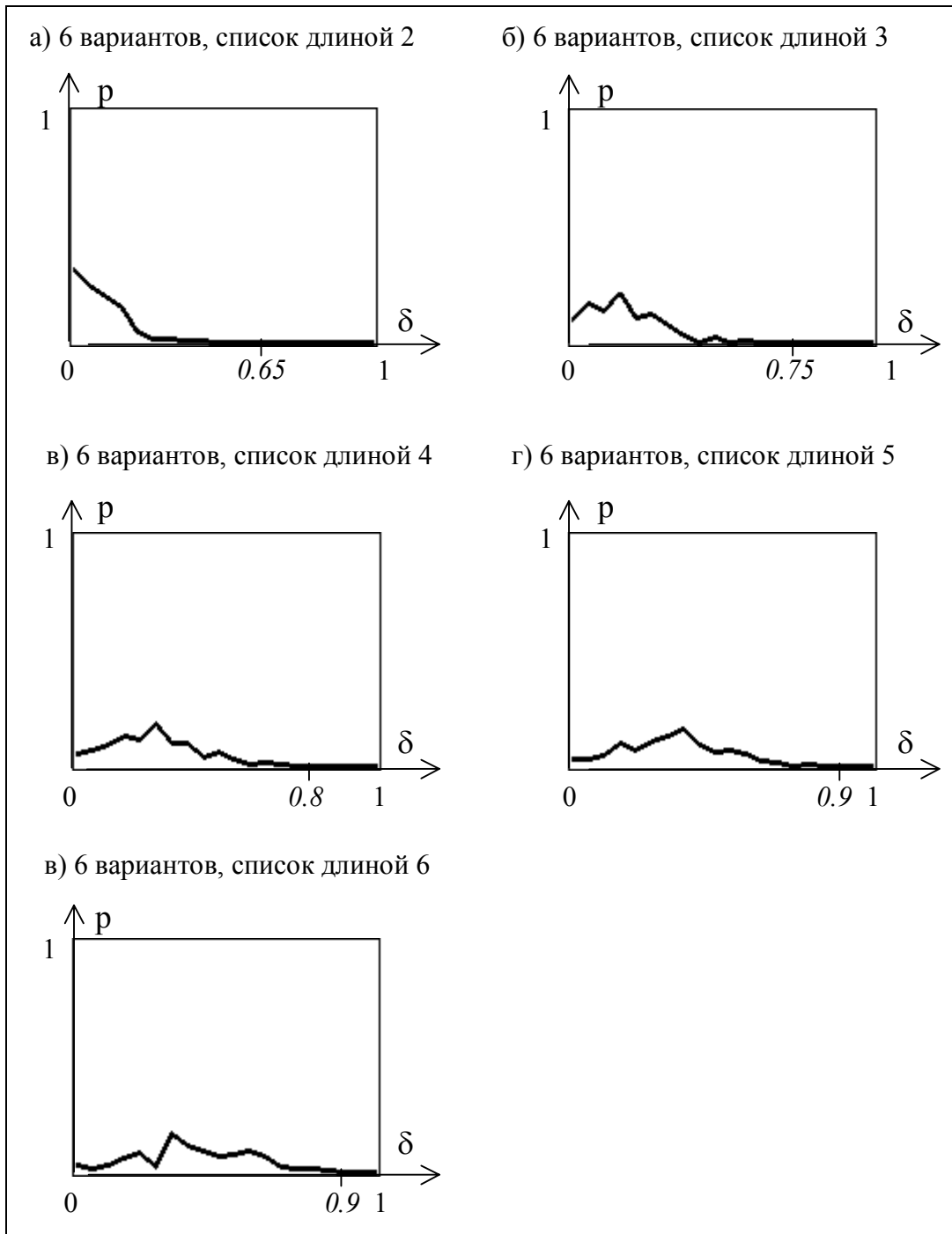
Рис. 3.13. Распределение оценок для ответов типа СПИСОК ($N=6$)

Таблица 3.5. Математическое ожидание для ответа типа СПИСОК

Количество ответов, входящих в список	Количество вариантов ответов			
	4	5	6	7
2	0.27	0.16	0.11	0.08
3	0.35	0.26	0.19	0.15
4	0.42	0.35	0.28	0.23
5	—	0.41	0.35	0.29
6	—	—	0.42	0.37
7	—	—	—	0.43

Анализ графиков распределения оценок ответов типа СПИСОК (рис. 3.11–3.13) и табл. 3.5 показывает: для того, чтобы исключить возможность случайно получить удовлетворительную оценку при использовании стандартной шкалы оценок (2, 3, 4, 5), нужно предлагать для вопроса не менее двух "лишних" вариантов.

3.5.3. Моделирование ответа типа СПИСОК МНОЖЕСТВ

Для сравнения списков множеств в п. 3.3.3.4. "Списки множеств" предлагались два очевидных алгоритма сравнения списков множеств:

- (A1) – множества, входящие в ответ, считаются упорядоченными, поэтому i -й элемент ответа сравнивается с i -м элементом эталона (более короткий список дополняется пустыми множествами).
- (A2) – ответ не считается упорядоченным, поэтому каждый элемент ответа сравнивается с каждым элементом эталона. Элементы ответа a_i и эталона e_j сопоставляются в соответствии с максимальной суммой степеней сходства δ элементов ответа и эталона, и оценивается упорядочивание элементов a_i на основе меры порядка, введенной эталоном $\{e_j\}$ (т.е. оценка происходит два этапа).

Второй алгоритм дает более точную оценку. Но первый алгоритм работает значительно быстрее за счет исключения определения максимальной суммы, которое является NP-полной задачей с временной сложностью $N!$.

Сравним результаты моделирования этими алгоритмами для того, чтобы определить возможность использования алгоритма (A1) для ускорения анализа ответа без существенной потери качества. Результаты приведены в табл.3.6 (математическое ожидание случайного ввода правильного ответа) и табл.3.7 (корреляция между оценками ответов, рассчитанная как среднее квадратичное отклонение).

Как видно из результатов, погрешность оценок достаточно высока (10% и более), поэтому для моделирования ответов типа СПИСОК МНОЖЕСТВ использовался алгоритм (A2).

Таблица 3.6. Математическое ожидание случайного ввода правильного ответа для СПИСКА МНОЖЕСТВ при использовании алгоритмов (A1) и (A2)

Количество множеств, N	5 вариантов		6 вариантов		7 вариантов	
	(A1)	(A2)	(A1)	(A2)	(A1)	(A2)
2	0.16	0.19	0.11	0.15	0.078	0.119
3	0.19	0.27	0.15	0.23	0.107	0.188
4	0.20	0.33	0.16	0.30	0.123	0.267
5	–	–	0.17	0.34	0.138	0.295
6	–	–	–	–	0.143	0.330

Таблица 3.7. Корреляция между оценками при использовании алгоритмов (A1) и (A2)

Количество множеств, N	5 вариантов	6 вариантов
2	0.12	0.10
3	0.19	0.15
4	0.23	0.21
5	–	0.25

Моделирование проводилось для $N = 4 \div 7$ и схемы " N вариантов ответа – нет лишних". Результаты приведены на рис. 3.14–3.17, значения математического ожидания вероятности случайного ввода правильного ответа – в табл. 3.8.

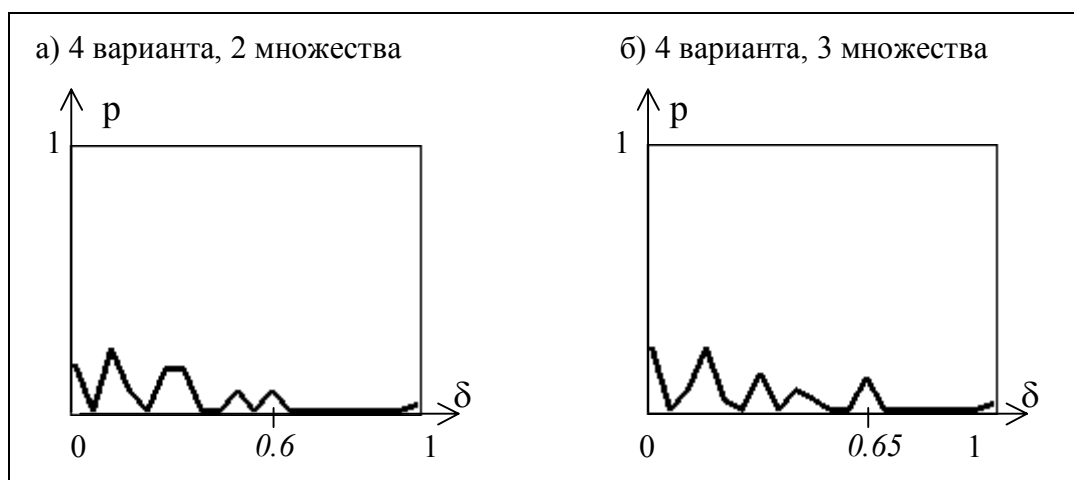
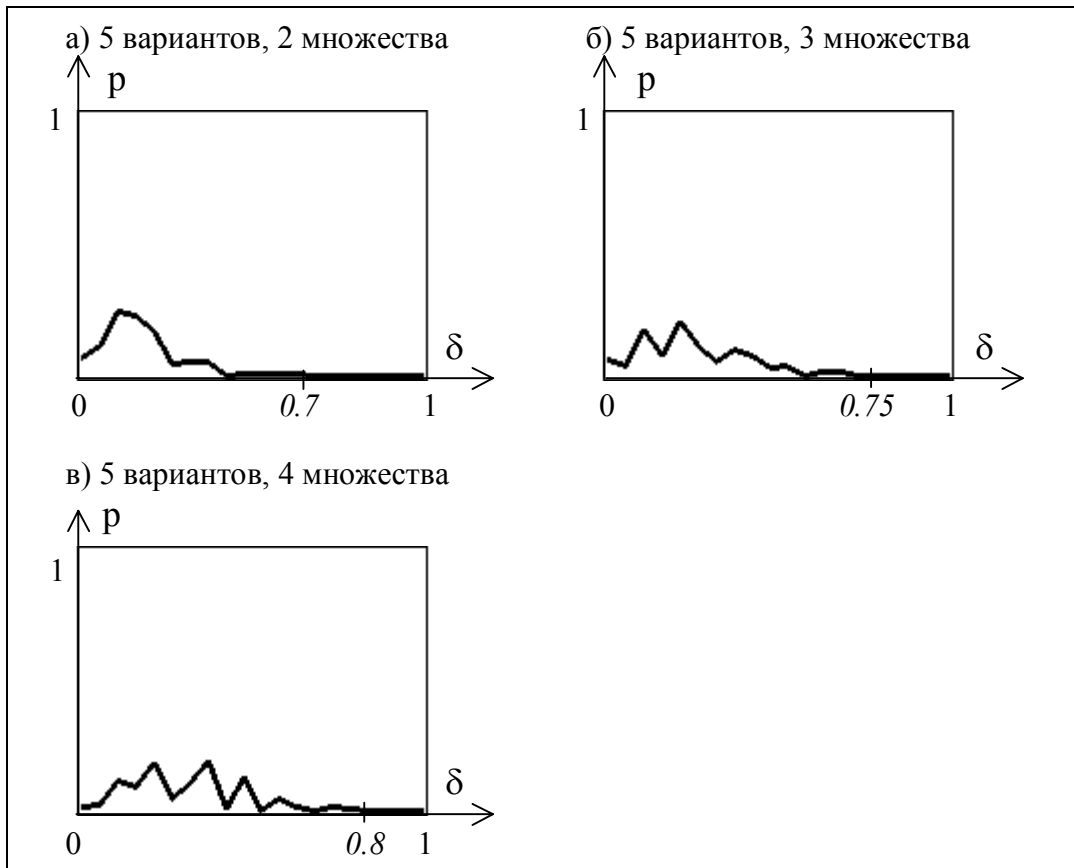
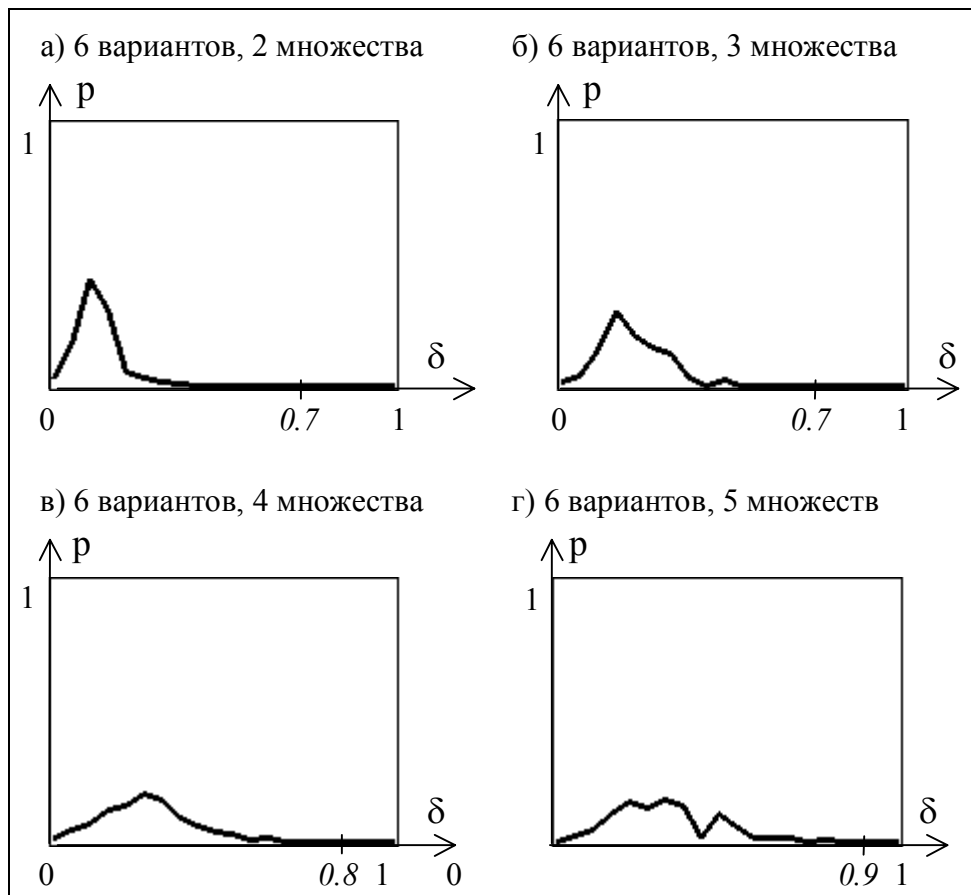


Рис. 3.14. Распределение оценок для ответов типа СПИСОК МНОЖЕСТВ (N=4)

Рис. 3.15. Распределение оценок для СПИСКА МНОЖЕСТВ ($N=5$)Рис. 3.16. Распределение оценок для СПИСКА МНОЖЕСТВ ($N=6$)

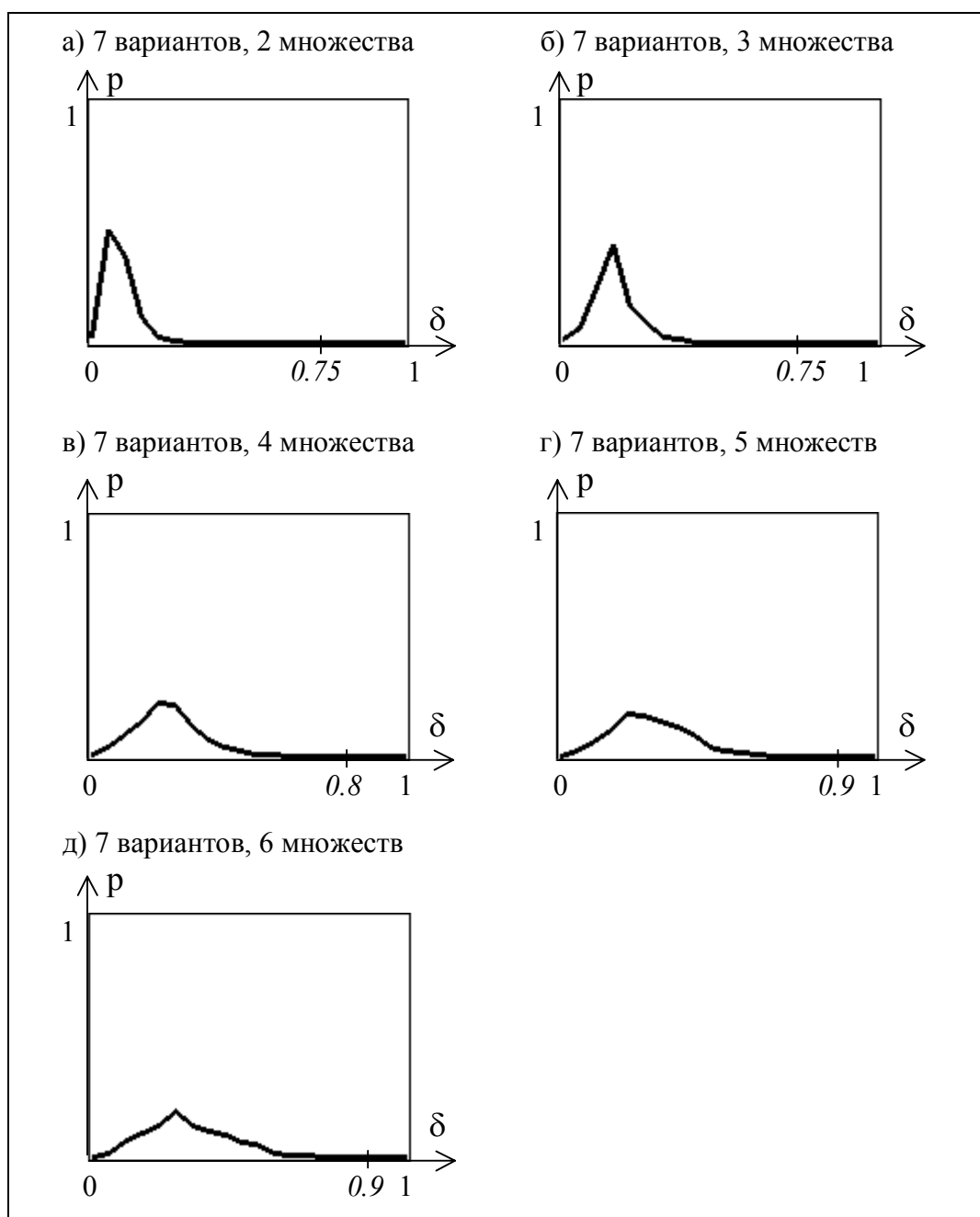


Рис. 3.17. Распределение оценок для СПИСКА МНОЖЕСТВ (N=7)

Таблица 3.8. Математическое ожидание для ответа типа СПИСОК МНОЖЕСТВ

Количество множеств	Количество вариантов ответов			
	4	5	6	7
2	0.26	0.19	0.15	0.12
3	0.31	0.27	0.23	0.19
4	—	0.33	0.30	0.27
5	—	—	0.34	0.30
6	—	—	—	0.33

Анализ результатов моделирования для схемы СПИСОК МНОЖЕСТВ показывает, что эта схема имеет хорошие характеристики и может применяться даже при N=5.

3.5.4. Моделирование ответа типа МНОЖЕСТВО СПИСКОВ

При моделировании ответа типа МНОЖЕСТВО СПИСКОВ для сравнения ответа с эталоном необходимо сопоставить каждому элементу (списку) ответа элемент (список) эталона. Вариантов такого сопоставления может быть $L!$, где L – количество списков. В качестве оценки варианта ответа выбирается максимальная из всех оценок, полученных в ходе сопоставления элементов.

Моделирование ответа типа МНОЖЕСТВО СПИСКОВ проводилось для $N=4\div 8$ и схемы " N вариантов ответа – нет лишних". Результаты приведены на рис. 3.18–3.19, значения математического ожидания – в табл. 3.9.

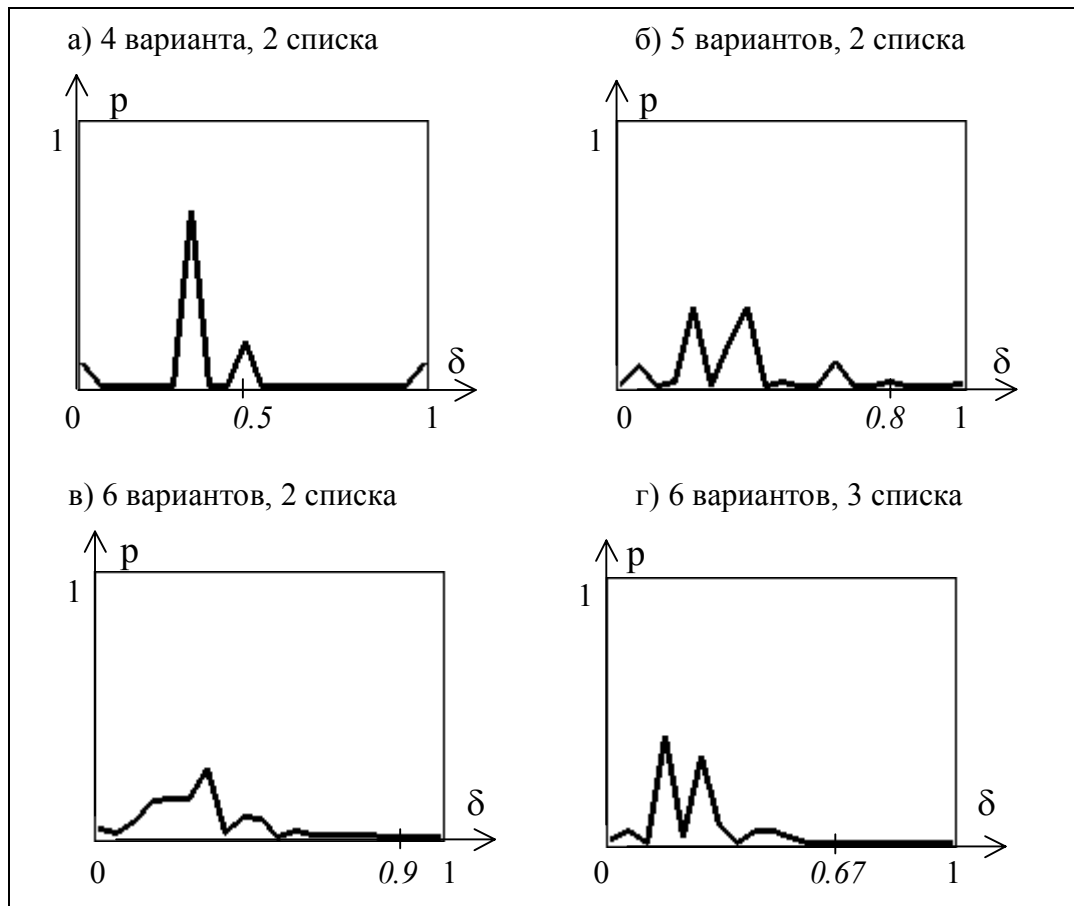


Рис. 3.18. Распределение оценок для МНОЖЕСТВА СПИСКОВ ($N=4,5,6$)

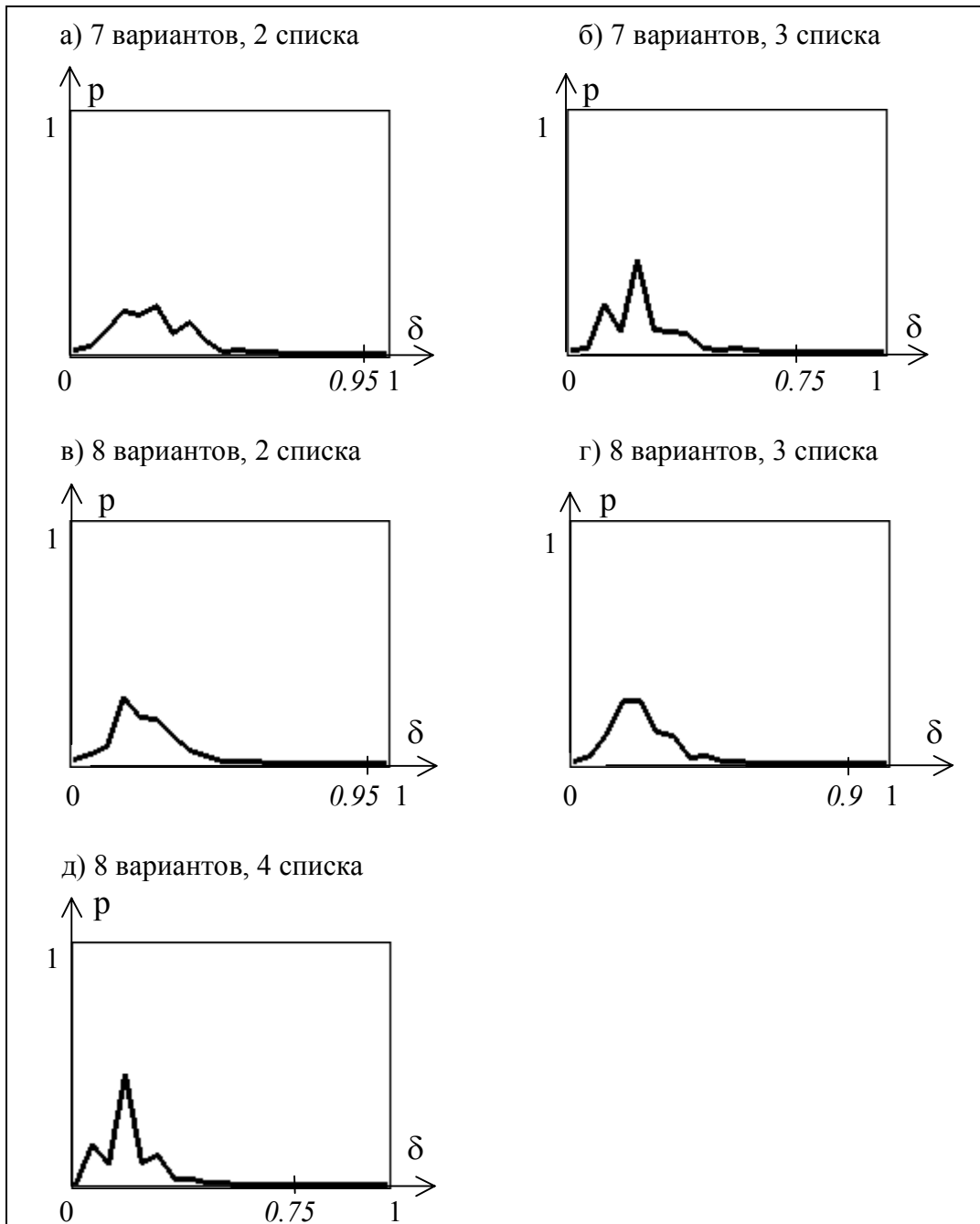


Рис. 3.19. Распределение оценок для МНОЖЕСТВА СПИСКОВ (N=7,8)

Таблица 3.9. Математическое ожидание для ответа типа МНОЖЕСТВО СПИСКОВ

Количество списков	Количество вариантов ответов				
	4	5	6	7	8
2	0.40	0.35	0.32	0.29	0.27
3	—	—	0.27	0.26	0.25
4	—	—	—	—	0.21

Анализ результатов моделирования для схемы МНОЖЕСТВО СПИСКОВ показывает, что эту схему следует использовать при $N > 5$.

Естественно, что для двухуровневых схем добавление в вопрос "лишних" вариантов, не входящих в эталон ответа, улучшит характеристики и снизит вероятность случайного получения положительной оценки.

Выводы

В главе были определены требования, которым должна соответствовать подсистема контроля знаний, и предложены пути решения задачи организации контроля знаний.

Анализ различных типов тестовых вопросов и ответов показал, что с практической точки зрения можно выделить следующие типы ответов: множество элементов (неупорядоченное); список элементов (упорядоченный); двухуровневые схемы, в которых в качестве элементов списка (множества) могут выступать множества и списки; выражение (арифметическое); фраза (текст); рисунок. Первые два варианта относятся к выборочному типу ответов, а с помощью двухуровневых схем можно формулировать вопросы на соответствие. Такой подход существенно расширяет возможности разработчика тестов по моделированию различных связей между элементами ответа и позволяет использовать для анализа и оценки ответов методы, основанные на положениях комбинаторного анализа.

Для оценки выборочных ответов был разработан Д-метод, который не зависит от семантики вопроса и позволяет проводить оценку дифференцировано, а не по наиболее часто используемой двухбалльной шкале.

Показано, что в качестве моделей свободно-конструируемых ответов могут использоваться списки и множества. Например, текстовые ответы, вводимые через шаблоны, можно рассматривать как список слов (словосочетаний). Для определения правильности текстовых ответов при таком подходе может применяться Д-метод. Также рассмотрена задача качественного анализа графиков. Показана сводимость синтаксического описания графиков к типу СПИСОК с последующим применением для определения его правильности Д-метода.

Поскольку при использовании выборочных ответов существует возможность неадекватной оценки знаний обучаемого в случае случайного ввода правильного ответа, была проведена оценка предложенных типов ответов (МНОЖЕСТВО, СПИСОК, МНОЖЕСТВО СПИСКОВ и СПИСОК МНОЖЕСТВ) с точки зрения величины такой вероятности.

Также была проведена оценка предложенных методов анализа ответов, при использовании которых появляются "частично правильные ответы" с оценкой, отличной от нуля. Оценка методов осуществлялась с помощью программного моделирования различных схем построения ответа для множеств вариантов разных мощностей.

Анализ полученных результатов моделирования позволил дать следующие рекомендации для составления тестов:

1. При использовании ответов типа МНОЖЕСТВО и схемы " N вариантов ответа – из них 1 правильный" количество вариантов ответов $N \geq 4$.
2. Не следует использовать дифференцированную оценку ответа типа МНОЖЕСТВО в случае, когда необходим точный ответ и наличие "лишних" вариантов (или отсутствие требуемых) должно расцениваться как ошибка.
3. Для ответов типа СПИСОК нужно предлагать не менее двух "лишних" вариантов ответов.
4. Схема типа СПИСОК МНОЖЕСТВ имеет хорошие характеристики и может применяться даже при $N=5$.
5. Схему типа МНОЖЕСТВО СПИСКОВ следует использовать при $N>5$.
6. Увеличение количества множеств для схемы типа СПИСОК МНОЖЕСТВ ухудшает ее характеристики, а увеличение количества списков для схемы типа МНОЖЕСТВО СПИСКОВ, наоборот, улучшает ее характеристики.

Соблюдение этих рекомендаций позволяет исключить возможность случайно получить удовлетворительную оценку при использовании стандартной шкалы оценок (2, 3, 4, 5).

4. РЕАЛИЗАЦИЯ РАСПРЕДЕЛЕННОЙ АВТОМАТИЗИРОВАННОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ

В целях экспериментальной проверки применимости предложенных методов создания РАОС был разработан макетный программный комплекс, в состав которого входят:

- подсистема управления (монитор), которая организует работу системы в целом;
- подсистема контроля знаний;
- набор баз знаний, реализующих управление процессом обучения и контролем знаний.

4.1. Описание программного комплекса

Программный комплекс предназначен для создания распределенных полнофункциональных обучающих систем по произвольной предметной области.

4.1.1. Функциональная структура программного комплекса

Функциональная структура программного комплекса приведена на рис. 4.1. В задачи монитора и системы контроля знаний, помимо указанных на рис. 4.1., входит формирование задания для процессора базы знаний (для внешнего управляющего модуля) и считывание результатов его работы из файла обмена данными. СКЗ также организует взаимодействие с обучаемым, оценивает его ответы и записывает результаты в базу данных.

Основные особенности этого комплекса:

- 1. Ориентация на работу в сети.** Система обеспечивает работу в режиме "клиент-сервер" в рамках ЛВС и через Internet. Включает специализированный протокол обмена данными в рамках ЛВС для поддержки распределенных вычислений.

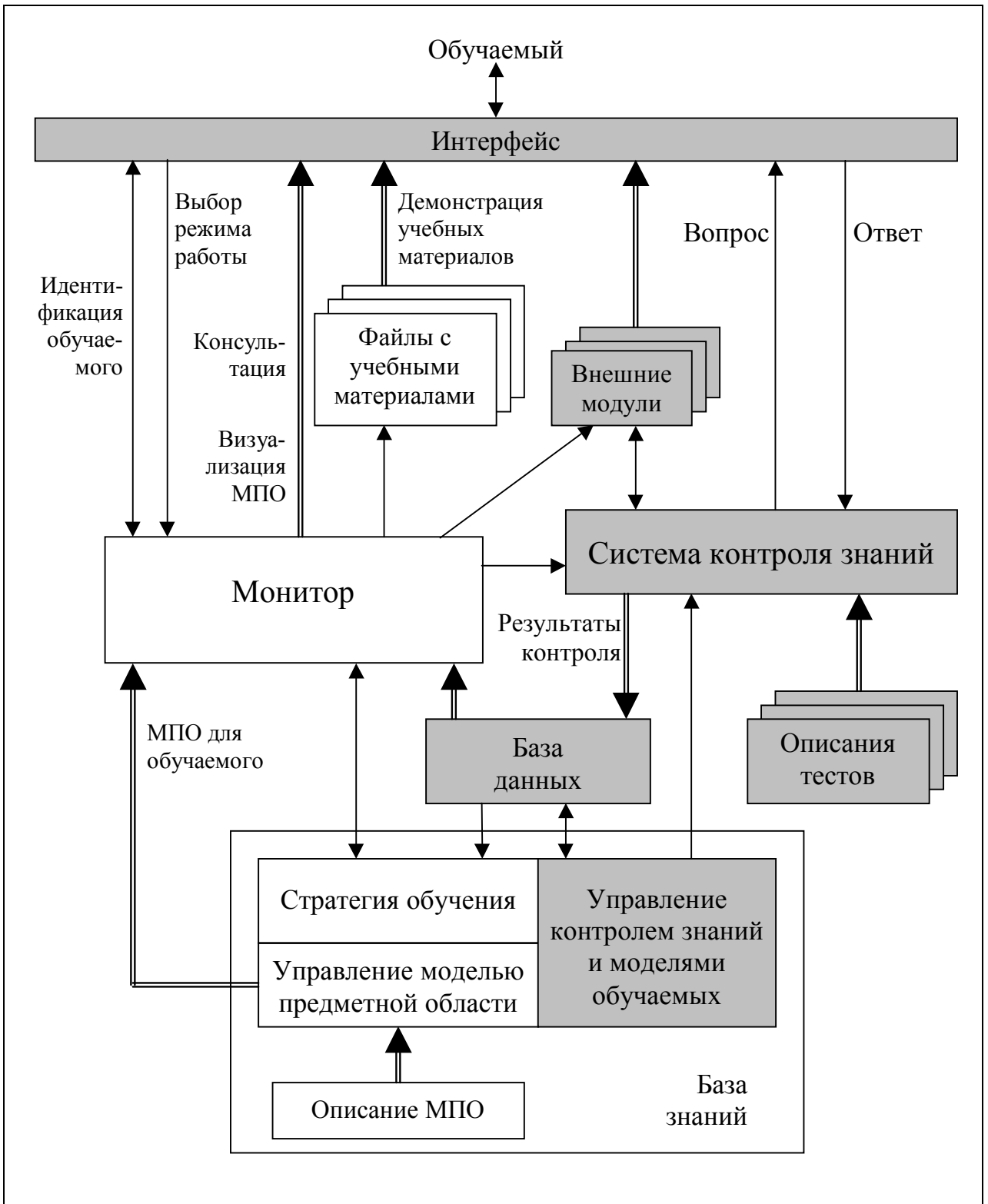


Рис. 4.1. Функциональная структура программного комплекса

2. **Открытость.** Принцип открытости подразумевает возможность настройки системы на требования конкретного пользователя. Здесь можно выделить три аспекта.

- Изменение режимов и параметров функционирования программного обеспечения, входящего в состав ядра системы. Это реализовано путем вынесения параметров в файл конфигурации (Приложение 3).
- Возможность использования любых учебных материалов на машинных носителях и методов подачи этих материалов: электронный справочник (от просмотра текстовых файлов до подключения гипертекста), запуск демонстрационных программ и т.д.
- Организация внешнего управления процессом обучения. Это достигается путем вынесения во внешние файлы (в базу данных и в базу знаний) правил организации управления обучением и всех возможных параметров и характеристик, которые можно измерить в процессе работы с обучаемым. Наборы правил можно модифицировать. Значения характеристик и параметров доступно для изменения и для использования в правилах БЗ.

3. **Переносимость** на различные платформы на уровне исходных текстов программ.

Переносимость обеспечивается за счет того, что монитор и ядро системы контроля знаний написаны на языке C и адаптированы к использованию под управлением ОС DOS (не ниже 3.3), UNIX (точнее, Free BSD 4.4 и Linux) и WIN9x/NT.

Система контроля знаний реализована в двух вариантах: как приложение под ОС Windows (система TS) и как консольное приложение (система IPDO).

4.1.2. Структура распределенного программного комплекса

Распределенность системы имеет два основных аспекта: наличие вспомогательного программного обеспечения (ВПО), которое выполняет отдельные внутренние функции АОС, и работа в режиме удаленного доступа.

Для обмена данными по сети Internet требуется дополнительное программное обеспечение, которое отвечает за создание HTML-страниц и

обмен информацией с ядром PDO. При создании СКЗ IPDO были рассмотрены различные варианты такого ПО, а именно:

1. Протокол Telnet, эмулирующий терминал пользователя. Для этого требуется минимум дополнительного программного обеспечения (собственно драйвер Telnet на машине клиента), этот подход обладает широкой функциональностью, но имеет небогатые выразительные возможности.
2. Использование CGI-интерфейса (автор скрипта – Силкин А.Г.). При этом на серверной машине необходимо наличие интерпретатора языка Perl (версия 5.0 и выше) и сервера, поддерживающего протокол CGI (например, Sambar). На машине клиента должен быть установлен браузер Internet Explorer 3.0 и выше (или любого другого, поддерживающего протокол CGI). Основное достоинство этого подхода – переносимость: написанный скрипт без каких бы то ни было изменений работает на разных платформах и позволяет просматривать HTML-страницы как с помощью Internet Explorer, так и с помощью Netscape Communicator. Недостатками такого решения являются скромные функциональные возможности (линейная схема тестирования и статическая информация) и ограниченное количество одновременно работающих с ним пользователей – не более 50.
3. Использование Java-апплета (автор – Потомский С.). Этот подход предоставляет наиболее развитые функциональные возможности вплоть до адаптивного тестирования.

С учетом функциональной структуры РАОС, распределенного характера системы и наличия дополнительного программного обеспечения общая структура программного комплекса будет выглядеть так (рис. 4.2):

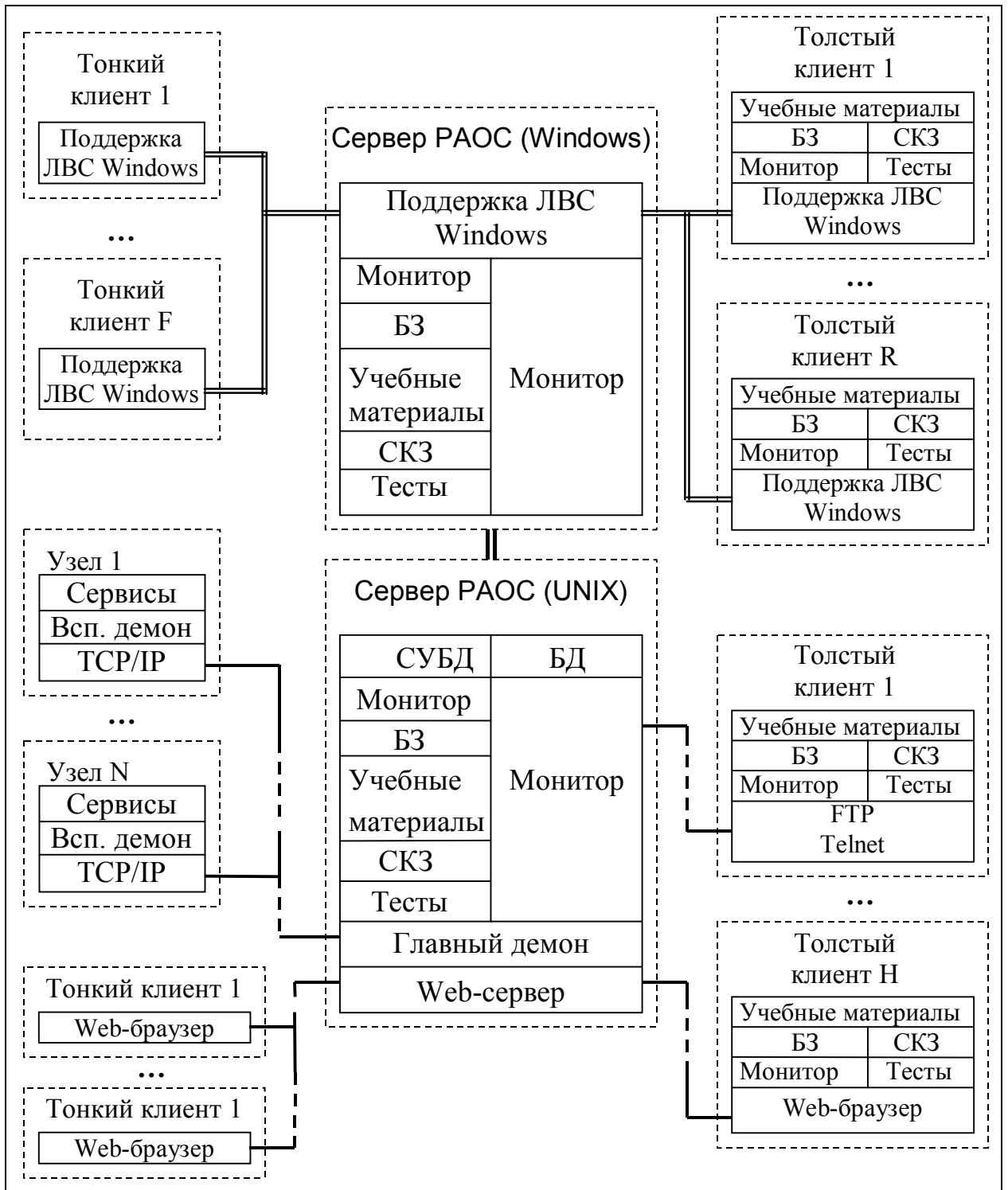


Рис. 4.2. Структура распределенного программного комплекса PAOC

Протокол обмена данными между ядром PAOC и ВПО осуществляет динамическую реконфигурацию системы при нарушении топологии сети с целью повышения надежности функционирования распределенной системы.

4.1.3. Протокол обмена данными с динамической реконфигурацией системы

Перед протоколом обмена данными стоят следующие задачи:

1. организация распределенных вычислений;
2. в случае обнаружения нарушения топологии сети – реконфигурация системы, которая заключается в запуске аналогичного процесса на одном из доступных узлов и перенаправлении потока данных вновь запущенному процессу.

Конфигурация системы описывается в конфигурационном файле, формат которого приведен ниже:

```

period = t
count = N
Имя_узла_1 = Адрес_1
Имя_узла_2 = Адрес_2
...
Имя_узла_N = Адрес_N
Services = M
Имя_процесса_1 = Процесс_1
timeout = t1
sites = имя_узла_1[, имя_узла_2,..., имя_узла_i]
...
Имя_процесса_M = Процесс_M
timeout = tm
sites = имя_узла_m1[, имя_узла_m2,..., имя_узла_mj]

```

(В квадратные скобки заключены необязательные параметры). Здесь $period=t$ – время отклика обслуживающего процесса; $count=N$ – количество узлов; M – количество процессов; $Имя_узла_i$ – идентификатор узла i ; $Имя_процесса_i$ – идентификатор процесса i ; $timeout$ – период времени ожидания ответа; $sites$ – список идентификаторов узлов, на которых может быть запущен данный процесс.

Для реализации протокола требуются процессы-демоны: главный демон на сервере АОС и вспомогательные демоны на тех узлах, на которых расположено вспомогательное программное обеспечение (ВПО). Фактически

процессы ВПО являются сервисами, для которых должен быть описан интерфейс.

Система распределенной обработки характеризуются небольшой длиной сообщений и существенным временем, которое требуется на их обработку. Сообщения, которыми обмениваются сервер и вспомогательные процессы, умещаются в одном пакете.

Формат пакета обмена данными следующий:

Адрес отправителя	Имя отправителя	Адрес получателя	Имя получателя	Код	Длина пакета данных	Данные
-------------------	-----------------	------------------	----------------	-----	---------------------	--------

Адрес и имя отправителя необходимы для того, чтобы процесс-получатель знал, кому отправлять ответ. Адрес и имя получателя определяют сам процесс-получатель. Код интерпретируется в зависимости от того, кому адресован пакет. Если адресат – вспомогательный демон, то это код команды, которую ему требуется выполнить. Если же пакет адресован главному демону, то это отклик (подтверждение готовности) или код завершения вспомогательного процесса.

Для исключения потери запросов главный демон должен поддерживать очередь запросов (в порядке поступления, FIFO). Очередь запросов хранит следующую информацию:

1. Идентификатор запроса.
2. Время поступления запроса.
3. Адрес и имя отправителя.
4. Имя адресуемого процесса.
5. Собственно запрос.
6. Идентификатор обслуживающего процесса.
7. Имя узла, на котором обслуживается запрос.
8. Таймаут, в течение которого ожидается ответ на запрос.
9. Ответ (результат обработки или сообщение об ошибке).

Запрос удаляется из очереди, если он обслужен или если он не может быть обслужен из-за отсутствия сервиса, которому был адресован запрос.

Структурная схема организации распределенных вычислений приведена на рис. 4.3. При получении запроса сервер с помощью списка сервисов определяет, что этот запрос адресован ВПО, и передает его главному демону. На рис. 4.4 приведен алгоритм работы главного демона. Для организации многопользовательской работы он порождает обслуживающий процесс (ОП) для каждого запроса.

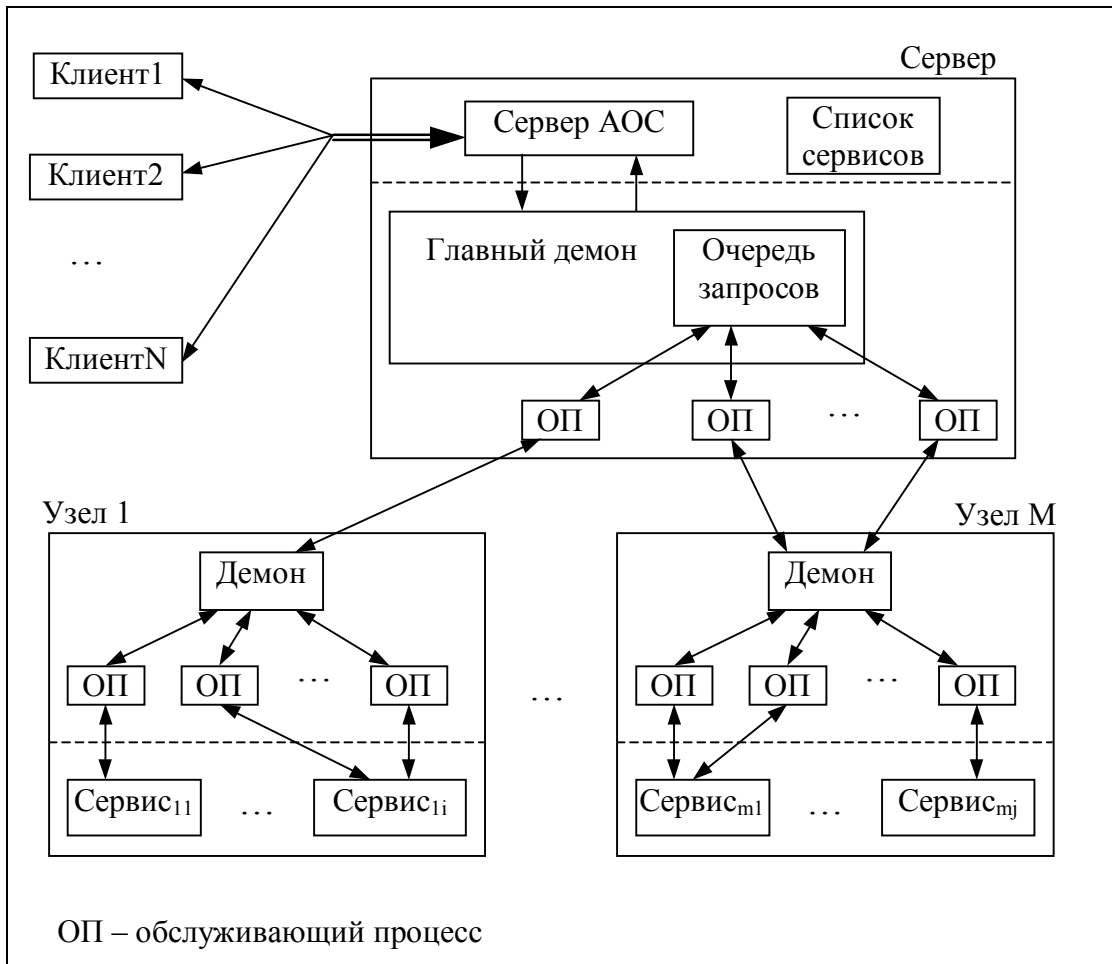


Рис. 4.3. Схема организации обмена данными между ядром АОС и ВПО

Через определенные промежутки времени (period), ОП должен присылать демону отклик в подтверждение своей работоспособности. Если демон во время не получает этот отклик, он считает, что ОП "завис" и начинает процедуру реконфигурации (рис. 4.5). При невозможности обработки запроса главный демон сохраняет запрос в файле трассировки, а сервер выдает сообщение тому клиенту, от которого пришел запрос.

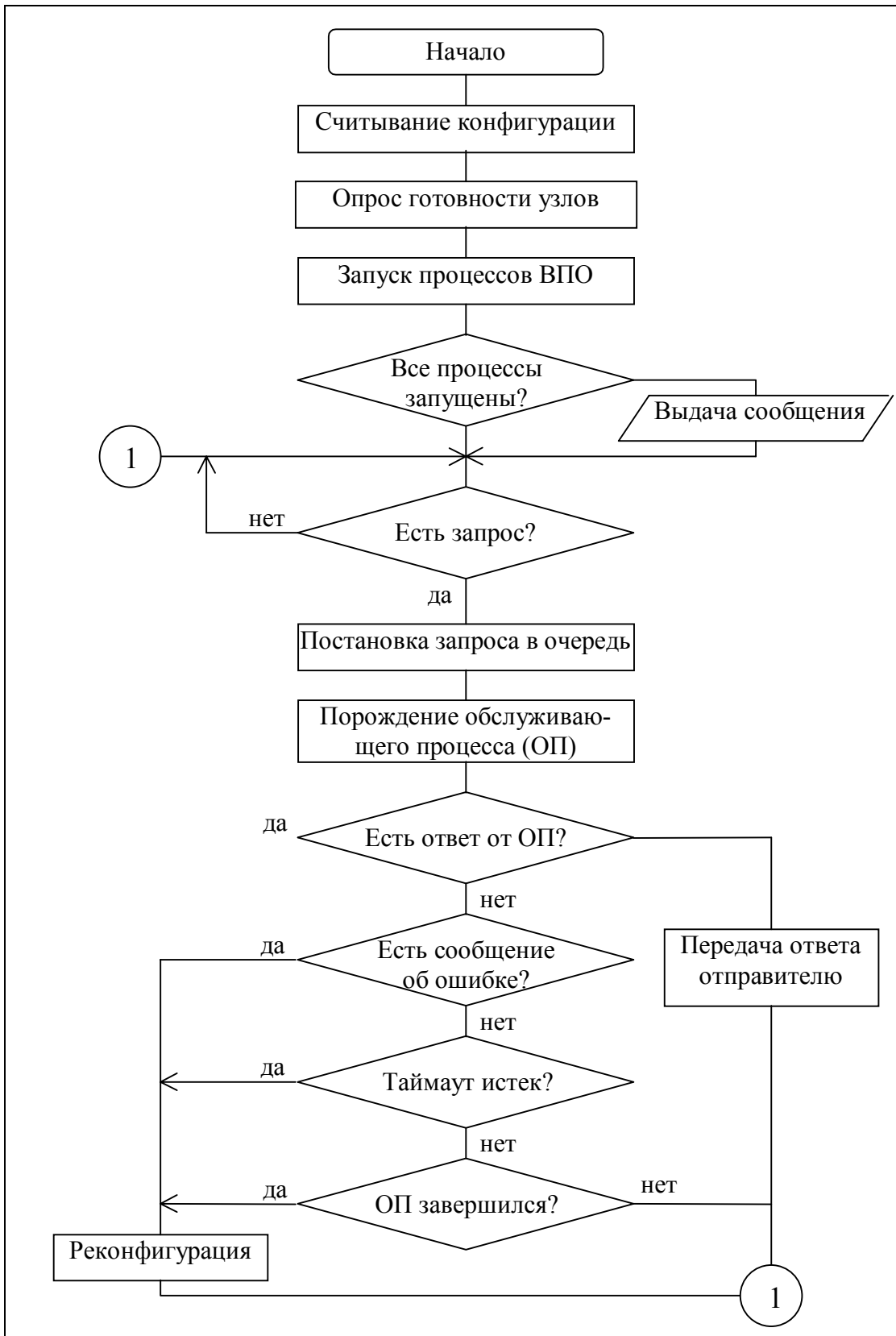


Рис. 4.4. Алгоритм работы главного демона

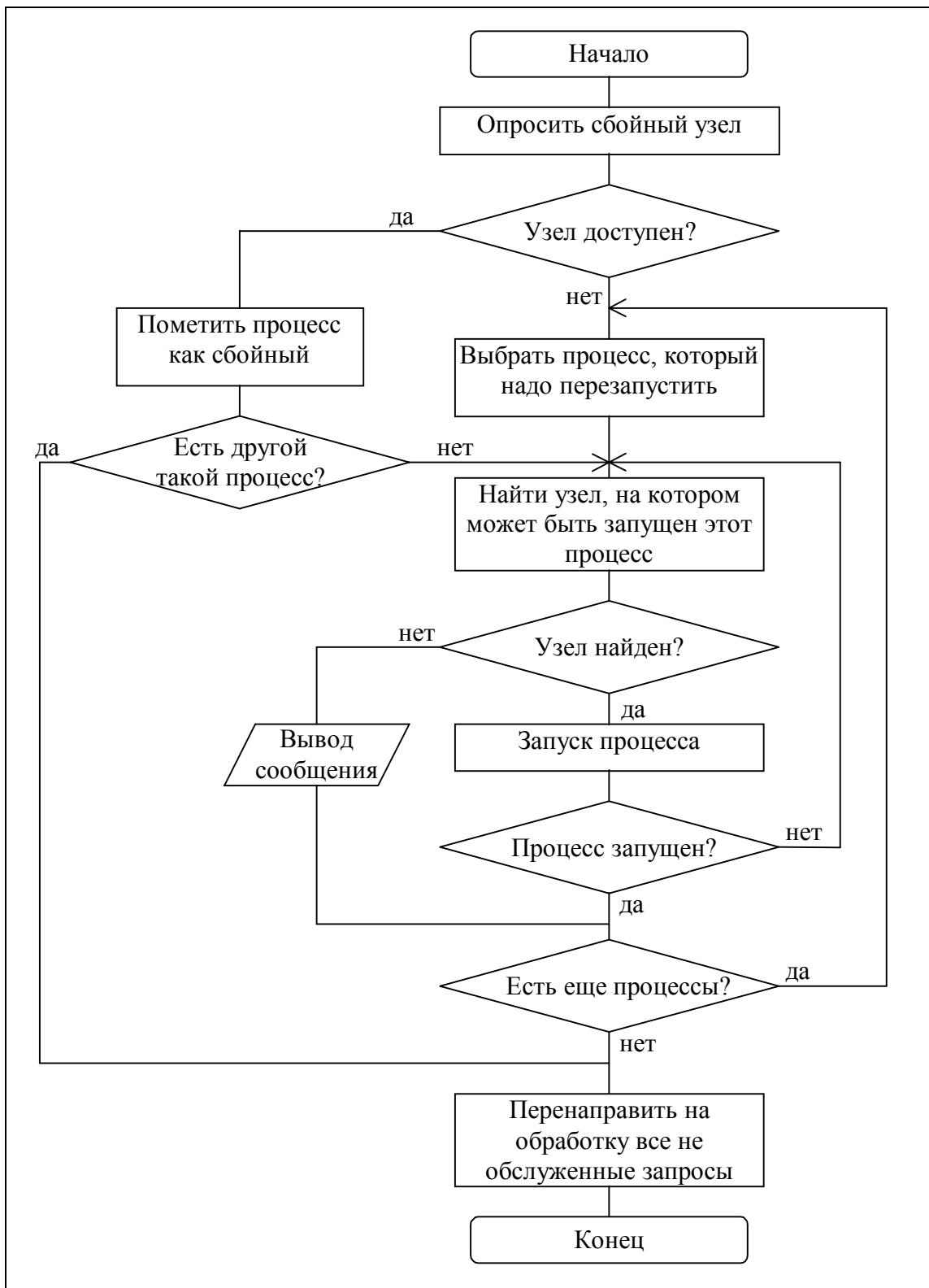


Рис. 4.5. Алгоритм реконфигурации

Вспомогательные демоны также порождают обслуживающие процессы для выполнения запросов. Алгоритм работы вспомогательного демона приведен на рис. 4.6.

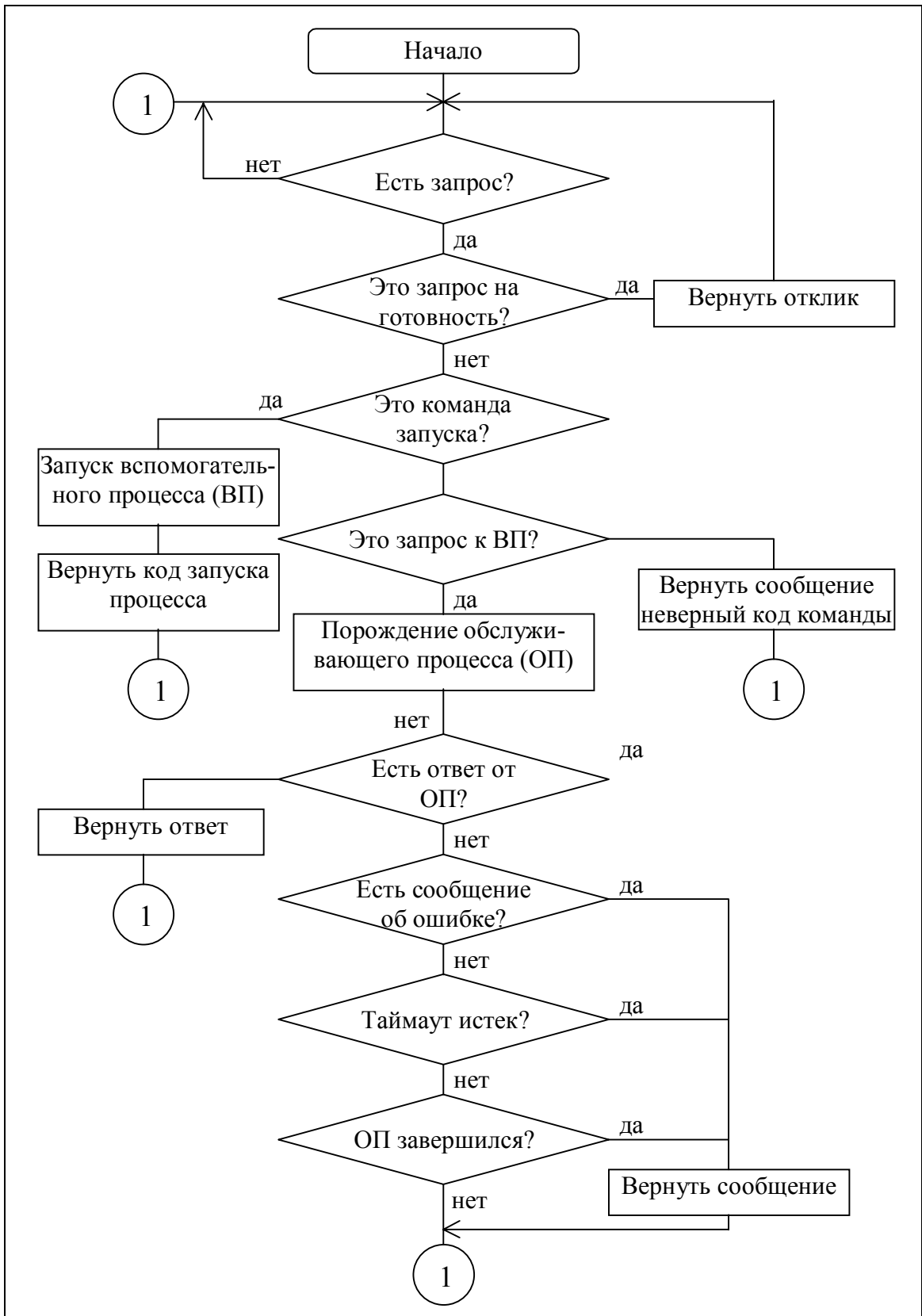


Рис. 4.6. Алгоритм работы вспомогательного демона

4.1.4. Настройка АОС

Настройка АОС происходит путем заполнения системы предметными данными, создания модели предметной области, изменения значений параметров в файлах конфигурации. Система поддерживает встроенные и пользовательские параметры.

Для встроенных параметров определены названия, алгоритмы обработки и, в большинстве случаев, множество допустимых значений. Эти параметры можно разделить на группы по принципам их обработки:

1. Параметры с определенными значениями.

Эти параметры могут принимать значения из фиксированного списка значений, для каждого из которых предусмотрен свой алгоритм обработки. Например, параметр СПОСОБ_ВЫДАЧИ для системы контроля знаний со значениями ОДИН и ВСЕ.

Если такой параметр не задан пользователем в явном виде, ему присваивается определенное значение по умолчанию.

2. Параметры с определенным алгоритмом.

Значения таких параметров задаются пользователем произвольно, и предусмотрен единственный алгоритм обработки, не зависящий от значения. Например, параметр PROCESS определяет команду строку, которая будет запущена при возбуждении синапса.

3. Параметры с произвольными значениями.

Значения таких параметров задают ограничения соответствующих характеристик работы системы. Например, параметр TIMEOUT задает максимальное время ответа на вопрос.

Перечень встроенных параметров приведен в Приложении 3.

Пользовательские параметры определяются преподавателем при заполнении системы, для них задается имя и выражение для расчета значения. Выражение может быть арифметическим или логическим, в качестве операндов могут выступать названия встроенных и пользовательских параметров и имена функций, определенных в системе.

Эти параметры описывают произвольные характеристики составляющих предметной области и модели обучаемого и предназначены, в частности, для использования в правилах базы знаний.

4.2. Подсистема контроля знаний

Ядро СКЗ оформлено в виде библиотеки функций. На основе этой библиотеки реализованы два варианта СКЗ:

- система TS, для которой написан оконный интерфейс под ОС Windows на Builder C++;
- система IPDO, которая является консольным приложением и предназначена для удаленного тестирования обучаемого в интерактивном режиме с использованием сети Internet.

4.2.1. Особенности системы контроля знаний

СКЗ характеризуется следующими особенностями:

1. Реализация основных форм (по Аванесову, [1]) представления ответов, а именно: закрытый (выборочный); открытый (числовой и текстовый); на упорядочение элементов; на соответствие (обычно, элементов двух множеств). Ответ на закрытый тест реализован как множество, ответ на упорядочение элементов – как список.
2. Множество типов ответов расширено за счет того, что:
 - в качестве числового ответа рассматривается арифметическое выражение;
 - используются двухуровневые схемы, в которых элементами множеств и списков являются списки или множества, что дает широкие возможности по представлению связей между элементами. Эти схемы:
 - позволяют моделировать ответы на соответствие, не ограничивая количество исходных множеств;
 - могут использоваться, например, для анализа синтаксического описания графических образов.

- для реализации тех форм ответов, которые не включены в систему, предусмотрена возможность вызова внешних процессов.
3. Использование дифференцированных методов для оценки ответов.
 4. Возможность параметрической настройки, которая заключается в выборе:
 - способа формирования теста;
 - способа оценки теста;
 - способа оценки ответа.
 5. Открытость архитектуры системы:
 - а) внешняя:
 - использование декларативного языка описания тестов;
 - хранение описаний тестов и заданий на тестирование во внешних файлах;
 - возможность подключения внешних модулей:
 - для представления вопросов;
 - для получения ответов обучаемого.
 - б) внутренняя, которая обеспечивается возможностью подключения внешних модулей:
 - для получения оценок ответов;
 - для управления формированием теста;
 - для управления проведением тестирования.

Представление текстовых вопросов в виде шаблона (см. п. 3.3.5.) осуществляется с помощью специальной программы (автор – Карпов В.Э.). Для получения ответов, представление которых выходит за рамки внутренних возможностей системы, были написаны графический редактор (авторы – Карпов В.Э. и Щеповалин Д.) и редактор формул (автор – Щеповалин Д.).

Внешнее управление работой СКЗ дает возможность выбора произвольных схем проведения тестирования, в том числе, адаптивных. Для организации внешнего управления подсистемой контроля знаний созданы базы знаний (Приложение 1), реализующие:

- формирование теста на основе параметров вопросов;
- тестирование в соответствие с пороговой и адаптивной схемами.

Для обеспечения переносимости тестов создан язык описания тестов [41].

4.2.2. Язык описания тестов

4.2.2.1. Требования, предъявляемые к языку описания тестов

Основные свойства, которыми должен обладать язык описания контрольных заданий, следующие:

- простота описания теста;
- учет максимально большого количества возможных вариантов ответов и вопросов;
- переносимость создаваемых тестов на различные платформы (на уровне их текстов);
- возможность подключения и использования внешних модулей (для демонстрации вопросов, получения ответов и для определения правильности полученных ответов).

4.2.2.2. Формат языка описания тестов

Для того чтобы отделить описание теста от его алгоритмической реализации и сделать его переносимым, язык должен быть декларативным.

Любая система контроля знаний включает в себя множество предлагаемых вопросов, правильные ответы на них и правила определения корректности ответов, полученных в ходе опроса. Поэтому описание теста состоит из следующих частей:

1. Описание вопросов.
2. Описание эталонов, методов и оценок ответов обучаемых.
3. Описание правил формирования теста из списка контрольных заданий и методов оценки теста в целом.

По результатам анализа систем контроля знаний в третьей главе были выделены следующие типы вопросов и ответов:

- **Типы вопросов:** ТЕКСТ, КОМАНДА.
- **Типы ответов:** ТЕКСТ, МНОЖЕСТВО, СПИСОК, ВЫРАЖЕНИЕ.

Все ответы, представление которых выходит за рамки указанных типов (рисунок, речевое сообщение и др.), могут быть получены в рамках процесса – командной строки, указанной в соответствующем вопросе.

Описание теста может содержать данные по нескольким темам. Оно хранится в текстовом файле и представляет собой последовательность блоков описаний вопросов, ответов и типов контроля. Каждая фраза (блок) начинается с одного из ключевых слов КОНТРОЛЬ_ПО_ТЕМЕ, СПИСОК_ТЕМ, ВОПРОС и ОТВЕТ и заканчивается ключевым словом КОНЕЦ и символом ';'. Внутри каждой фразы в произвольном порядке располагаются ключевые слова, определяемые типом фразы, и данные, относящиеся к этим ключевым словам. Необязательные параметры описания теста могут быть опущены (в данном ниже описании они заключаются в квадратные скобки).

Комментарии. Тип комментария – строчный. В качестве символа начала комментария используется символ ‘;’, стоящий в начале строки.

Разделители. Разделителями лексем являются пробел, запятая, табуляция и символ перевода строки. Кроме того, символы ‘=’ и ‘:’ (знак равенства и двоеточие) являются специальными разделителями; они рассматриваются как разделители наряду с пробелами и табуляциями, но должны стоять в местах, определенных синтаксисом описания.

Идентификатор является произвольной последовательностью символов длиной не более 255. Если идентификатор содержит пробелы, он должен быть заключен в двойные кавычки.

Система нечувствительна к регистру: идентификаторы и ключевые слова могут быть написаны как прописными, так и строчными буквами. Текст, заключенный в двойные кавычки, воспринимается так, как он записан, т.е. с учетом регистра.

В качестве данных система воспринимает:

- идентификаторы – последовательности символов, не содержащие пробелов;
- строки – произвольные последовательности символов, заключенные в двойные кавычки;
- выражения – вычисляемые последовательности символов (констант и идентификаторов, имеющих значение), разделенные знаками арифметических операций.

Блок описания вопроса

```

ВОПРОС = <имя вопроса>
    [ТЕМА = <тема>]
    ТИП = <тип вопроса>
    СОДЕРЖАНИЕ = <текст вопроса> | <командная строка>
    [СВОЙСТВО <идентификатор свойства 1> = "<значение 1>"
    СВОЙСТВО <идентификатор свойства 2> = "<значение 2>"
    . . . . .
    СВОЙСТВО <идентификатор свойства n> = "<значение n>"]
КОНЕЦ;

```

<имя вопроса> – идентификатор вопроса (название).

<тема> – идентификатор темы (название).

<тип вопроса> – одно из ключевых слов: *ТЕКСТ*, *КОМАНДА*. Если тип вопроса *ТЕКСТ*, то на экран выводится строка <текст вопроса> (параметр СОДЕРЖАНИЕ). Если тип вопроса *КОМАНДА*, то запускается указанная командная строка.

СВОЙСТВО – характеристика вопроса, вводимая и определяемая пользователем: <идентификатор свойства i> – название i-го свойства, <значение i> – выражение, порождающее значение (произвольно типа) этой характеристики для данного вопроса.

Свойства могут быть использованы:

- при формировании списков вопросов для тестирования;
- при создании модели обучаемого;

- для управления вопросом типа КОМАНДА (см. Приложение 3).

ТЕМА – объявление темы может встречаться в любом месте текста. В зависимости от местоположения различаются глобальное и локальное объявление темы. *Глобальным* считается объявление, расположенное непосредственно перед началом фразы, его действие распространяется на все вопросы до следующего объявления (если в вопросе не введено локальное объявление). *Локальным* считается объявление, расположенное внутри блока описания вопроса, и оно действует в пределах этого блока.

Блок описания ответа

ОТВЕТ = <идентификатор вопроса>

[ТИП = <тип ответа>]

ЭТАЛОН = <список> | <множество> | <текст> | <выражение>

[ЭТАЛОН = <список> | <множество> | <текст> | <выражение>

...]

БАЛЛ = min : max

ОЦЕНКА = <тип оценки>

[СВОЙСТВО <идентификатор свойства 1> = "<значение 1>"

СВОЙСТВО <идентификатор свойства 2> = "<значение 2>"

.

СВОЙСТВО <идентификатор свойства n> = "<значение n>"]

КОНЕЦ;

<тип ответа> – одно из ключевых слов: *СПИСОК*, *МНОЖЕСТВО*, *ТЕКСТ*, *ВЫРАЖЕНИЕ*:

- <список> – список разделенных запятыми вариантов, заключенный в квадратные скобки, например [a, b, c].
- <множество> – множество разделенных запятыми вариантов, заключенных в фигурные скобки, например: {a, b, c}.
- <текст> – произвольная последовательность символов, заключенная в двойные кавычки, например: “Физика высоких температур”.
- <выражение> – арифметическое выражение, например: (a+b)/2.

Примечание: тип комбинированного ответа (множество списков или список множеств) определяется внешними скобками. Например:

[{a, b}, {c}, {d, e}] – СПИСОК
 {[a], [b, c, d], [e]} – МНОЖЕСТВО

ЭТАЛОН – правильный ответ (эталон). Каждому вопросу могут соответствовать несколько эталонов. Ответ обучаемого сравнивается со всеми эталонами и максимальная степень сходства ответа и эталонов берется в качестве окончательного результата.

БАЛЛ – максимальный и минимальный балл, который можно получить за правильный или неправильный ответ на данный вопрос (разделяются двоеточием).

<тип оценки> – описывает способ определения правильности ответа:

- **РАВНО**(<величина погрешности>): действует по-разному в зависимости от типа ответа. Для ответа типа **ВЫРАЖЕНИЕ** величина погрешности определяет разброс значения ответа относительно эталона, а для остальных типов она изменяется в интервале [0,1]. Если ответ укладывается в интервал (эталон) \pm (величина погрешности), то ответ считается правильным (оценка равна max), в противном случае ответ неверен (оценка равна min).
- **СХОДСТВО**(<величина погрешности>): не используется для ответа типа **ВЫРАЖЕНИЕ**.

Если степень сходства ответа и эталона не больше указанной погрешности, то ответ считается правильным (оценка равна max), иначе ответ неверен (не совсем верен) и оценка O вычисляется так:

$$O = \max - (1 - \delta)(\max - \min),$$

где δ – степень сходства ответа и эталона, $\delta \in [0,1]$; max, min – максимальная и минимальная оценки за ответ.

Величина погрешности может задаваться арифметическим выражением.

Блок выбора тем

Этот блок дает возможность ограничить список тем, участвующих в контроле, и определить порядок выдачи тем для контроля.

СПИСОК_ТЕМ = <список тем> | ВСЕ
 [СПОСОБ_ВЫДАЧИ = ОДИН | ВСЕ]
 КОНЕЦ;

СПИСОК_ТЕМ – список тем через запятую или все темы (ключевое слово *ВСЕ*). Темы для опроса выбираются в том порядке, в котором они указаны.

СПОСОБ_ВЫДАЧИ – выдается по одной теме (*ОДИН*) или все (*ВСЕ*) темы сразу в виде списка тем. В последнем случае пользователь сам может выбирать последовательность тем для ответов. (По умолчанию – *ОДИН*).

Блок описания контроля по теме

Этот блок позволяет указывать способы формирования списка вопросов по теме, выдачи этих вопросов обучаемому и подсчета итоговой оценки. Можно указывать разные способы для разных типов контроля.

КОНТРОЛЬ_ПО_ТЕМЕ = <название темы>
 [СПИСОК = {<список вопросов>} | ВСЕ]
 ТИП = <тип контроля 1>
 [СПОСОБ_ВЫБОРА = СЛУЧАЙНО_БЕЗ_ПОВТОРОВ | СЛУЧАЙНО
 ПОДРЯД | ПОЛЬЗОВАТЕЛЬ]
 [СПОСОБ_ВЫДАЧИ = ОДИН | ВСЕ]
 [КОЛИЧЕСТВО_ВОПРОСОВ = <N_queryes >]
 [КОЛИЧЕСТВО_ОТВЕТОВ = <N_answers>]
 ОЦЕНКА = <число> | <процент>
 [ТИП_ОЦЕНКИ=СУММА | ПРОЦЕНТ | ШКАЛА <низший балл>:<высший балл>]
 [ТИП = <тип контроля 2>
 ...]
 [ТИП = <тип контроля 3>
 ...]
 КОНЕЦ;

СПИСОК – для каждой темы может быть указан конкретный список идентификаторов вопросов (через запятую) или ключевое слово *ВСЕ*.

Если данный параметр опущен, то это означает, что вопросы будут выбираться из всего списка вопросов по данной теме.

ТИП – могут быть указаны разные типы контроля. Тип контроля – одно из ключевых слов:

ЗАВЕРШАЮЩИЙ – итоговый контроль, по результатам которого определяется, сдана ли тема. Например, если тема сдана (полученная оценка не меньше той, что указана в предложении *ОЦЕНКА* (см. ниже)), то вопросы по этой теме обучаемому больше задаваться не будут.

ПРОМЕЖУТОЧНЫЙ – контроль знаний с выставлением оценки по теме. Например, если промежуточный контроль проводится несколько раз по одной и той же теме, то программа может хранить последнюю оценку и среднее арифметическое всех полученных оценок по данной теме.

ВЫБОРОЧНЫЙ – по результатам контроля выставляется одна общая оценка за все темы, участвующие в выборочном контроле. Может использоваться при итоговом контроле знаний по курсу.

СПОСОБ_ВЫБОРА – способ выбора вопроса из списка:

СЛУЧАЙНО_БЕЗ_ПОВТОРОВ – вопрос выбирается из списка случайным образом, но вопросы не повторяются (значение по умолчанию).

СЛУЧАЙНО – вопрос выбирается случайным образом из указанного (или общего) списка.

ПОЛЬЗОВАТЕЛЬ – пользователь сам выбирает вопрос из полного списка вопросов. Может использоваться для самоконтроля.

ПОДРЯД – выбирается N_queries вопросов подряд из полного списка.

СПОСОБ_ВЫДАЧИ – способ выдачи вопросов из списка:

ОДИН – пользователю выдается по одному вопросу из списка выбранных вопросов. По умолчанию выдается по одному вопросу.

ВСЕ – сразу выдается весь список вопросов по теме, пользователь сам выбирает очередной вопрос из списка.

КОЛИЧЕСТВО_ВОПРОСОВ – количество задаваемых вопросов. Если параметр опущен, задаются все вопросы из списка; если при этом отсутствует список вопросов, то задаются все вопросы по теме.

КОЛИЧЕСТВО_ОТВЕТОВ – количество полученных ответов, при котором опрос считается состоявшимся. Если параметр опущен, необходимо ответить на все заданные вопросы (опрос нельзя прервать).

ОЦЕНКА – оценка (число или процент правильных ответов), при которой тема считается сданной.

ТИП_ОЦЕНКИ – способ определения общей оценки O за тему на основании оценок O_i , полученных за каждый ответ:

- *СУММА* (используется по умолчанию) (формула (2.3)).
- *ШКАЛА* $\langle \min \rangle : \langle \max \rangle$ – оценка нормализуется в соответствии с указанной размерностью балльной шкалы (формула (2.4)):
- *ПРОЦЕНТ* (формула (2.5)).

Вариант программной реализации КС-грамматики [59], осуществляющей разбор описания теста, приведен в Приложении 2.

4.2.3. Контроль знаний в режиме удаленного доступа

Последовательность проведения контроля знаний в режиме удаленного доступа определяется используемым программным обеспечением. Система TS позволяет проводить тестирование в рамках ЛВС, при этом ее возможности полностью аналогичны возможностям локального режима. Для системы IPDO было рассмотрено три варианта организации удаленного тестирования: применение Telnet, CGI и Java.

Telnet подразумевает эмуляцию терминала, поэтому вопросы и ответы могут быть только текстовые и числовые. Этот подход позволяет использовать базы знаний для организации тестирования по любой схеме, в том числе, интерактивной (пороговая и адаптивная схемы).

CGI-скрипт имеет более широкие возможности представления вопросов (текст и графическое изображение формата GIF) и ответов. Для

ввода ответов в виде формул и графиков были созданы Internet-аналоги графического редактора и редактора формул (автор – Силкин А.Г.). Но формирование HTML-страницы при таком подходе происходит один раз, после чего она отправляется обучаемому. Поэтому с его помощью возможно тестирование только по линейной схеме.

Java-апплет, реализующий еще один вариант интерфейса к СКЗ, позволяет системе не только проводить удаленное тестирование по любой схеме (в том числе, интерактивной), но имеет также возможности для удаленного администрирования пользователей и использования вышеупомянутых Internet-приложений.

4.2.4. Управление тестированием

Для организации управления тестированием были разработаны базы данных, реализующие две схемы: пороговую и адаптивную. Описание этих схем приведено в разделе 2.5.2.1. "Встроенные параметры", а созданные базы знаний – в Приложении 1.

4.3. Эксперименты по применению подсистемы контроля знаний

Была проведена серия экспериментов по тестированию студентов по двум техническим дисциплинам, существенно различающимся по типам ответов:

- "Электрические машины" (характеризуются большим количеством графиков и формул);
- "Базы данных" (включают большое количество терминов).

В ней участвовали: студенты IV-го курса кафедры "Вычислительные системы и сети" и студенты IV-го курса кафедры "Управление и информатика в технических системах".

Специфика исследуемой предметной области такова, что проведение экспериментов с целью определения некоторых количественных характеристик, например, количественное сравнение результатов

автоматического контроля и устного зачета, чрезвычайно затруднено. Эти результаты зависят не только от методов оценки ответов студентов, но и от качества самих тестов. Поэтому цель была сформулирована иначе: определить качественные изменения, возникающие при применении СКЗ.

Результаты экспериментов приведены в табл. 4.1 – 4.3. Тест считался пройденным при получении оценки "хорошо" (4 балла).

Таблица 4.1. Сравнение результатов тестирования и устного зачета

Характеристики	Основная группа	Контрольная группа
Количество студентов	58	49
Среднее количество попыток сдать тест	2.4	—
Среднее количество попыток сдать зачет	1.2	1.8

Таблица 4.2. Корреляция результатов теста и устного зачета

№ п/п	Попытки сдать и полученные оценки					Устный зачет	Разность последних оценок
	Тест						
	1	2	3	4	5		
1	2.7	3.2	3.4	4.8	—	4	-0,8
2	4.6	—	—	—	—	5	0,4
3	4.0	—	—	—	—	5	1,0
4	3.9	4.0	—	—	—	5	1,0
5	4.6	—	—	—	—	5	0,4
6	3.5	4.0	—	—	—	4	0
7	3.5	4.7	—	—	—	5	0,3
8	3.1	2.7	3.1	3.6	—	3	-0,6
9	4.0	4.2	—	—	—	5	0,8
10	3.5	3.3	4.3	—	—	5	0,7
11	3.9	4.0	—	—	—	5	1,0
12	3.7	3.9	4.1	—	—	5	0,9
13	3.9	4.0	—	—	—	5	1,0
14	2.9	4.1	—	—	—	5	0,9
15	3.9	4.8	—	—	—	5	0,2
16	2.8	2.9	3.3	3.5	4.2	4	-0,2
17	3.1	2.9	3.5	4.4	—	5	0,6
18	3.9	4.5	—	—	—	5	0,5
19	4.1	—	—	—	—	3	-1,1
20	3.9	4.3	—	—	—	5	0,7
21	3.6	3.8	4.2	—	—	5	0,8
22	3.4	3.8	—	—	—	4	0,2
23	3.9	4.1	—	—	—	5	0,9
24	4.4	—	—	—	—	5	0,6

Таблица 4.2. Корреляция результатов теста и устного зачета (продолжение)

№ п/п	Попытки сдать и полученные оценки					Устный зачет	Разность последних оценок
	Тест						
	1	2	3	4	5		
25	3.6	4.0	4.1	–	–	5	0,9
26	4.2	–	–	–	–	4	–0,2
27	3.0	3.6	3.8	4.8	–	4	–0,8
28	4.0	4.4	–	–	–	5	0,6
29	3.4	3.9	4.2	–	–	5	0,8
30	4.3	–	–	–	–	5	0,7
31	3.6	3.7	4.5	–	–	5	0,5
32	3.9	4.2	–	–	–	4	–0,2
33	3.1	2.9	3.7	4.0	–	3	–1,0
34	3.8	4.0	–	–	–	4	0
35	3.6	3.9	4.4	–	–	5	0,6
36	4.6	5.0	–	–	–	5	0
37	4.0	–	–	–	–	5	1,0
38	4.0	4.4	–	–	–	5	0,6
39	4.2	–	–	–	–	5	0,8
40	3.0	2.7	2.6	3.3	4.1	5	0,9
41	3.8	4.8	–	–	–	5	0,2
42	2.8	2.7	3.3	3.7	3.8	3	–0,8
43	3.3	3.2	3.5	4.0	–	4	0
44	3.2	2.9	3.9	4.3	–	4	–0,3
45	4.3	–	–	–	–	5	0,7
46	3.8	4.0	4.2	–	–	4	–0,2
47	3.8	5.0	–	–	–	5	0
48	3.6	4.1	3.9	4.6	4.8	5	0,2
49	3.9	–	–	–	–	3	–0,9
50	3.9	4.0	–	–	–	4	0
51	3.7	4.1	–	–	–	5	0,9
52	4.8	–	–	–	–	5	0,2
53	3.3	3.1	3.8	4.3	–	5	0,7
54	3.6	3.3	3.4	3.5	3.6	3	–0,6
55	2.6	2.2	2.9	3.8	3.8	3	–0,8
56	3.7	4.0	4.8	–	–	4	–0,8
57	3.6	3.1	4.0	4.2	4.6	5	0,4
58	2.9	2.5	3.9	4.2	–	4	–0,2
Корреляция результатов (в баллах)							0.08

Корреляция результатов рассчитывалась как среднее квадратичное отклонение оценок, полученных студентами за тест и на зачете.

Таблица 4.3. Распределение типов вопросов в тестах

Типы ответов		Тесты			
		ЭМ	ЭМУ	БД	МД
Свободно-конструируемые ответы	Текст	2	3	13	6
	Выражение	2	9	1	1
	Графики	3	1	0	0
	Формула	5	5	0	0
	Список терминов	0	0	1	1
	Множество терминов	1	0	4	4
Итого:		13	18	19	12
Двухуровневые схемы	Список множеств	0	0	1	1
	Множество списков	0	0	3	3
Итого:		0	0	4	4
Выборочные ответы	Множество, 1 из N	7	9	5	2
	Множество, k из N	3	5	4	6
	Список	2	0	5	3
Итого:		13	14	14	11

Примечание. В табл. 4.3 используются следующие сокращения:

ЭМ – электрические машины; ЭМУ – электромеханические устройства; БД – базы данных; МД – модели данных.

Результаты экспериментов подтверждают существенное сокращение времени, затрачиваемого преподавателем на проведение контрольных мероприятий. Устный зачет, проведенный после тестирования, показал высокую степень корреляции результатов обеих форм проверки знаний.

Анализ используемых в тестах типов ответов показал, что составители тестов отдают предпочтение расширенным типам ответов (двухуровневые схемы и свободно-конструируемые ответы: текст, графики, формулы, арифметические выражения).

Выводы

В четвертой главе описана реализация полнофункциональной РАОС, основанной на принципе открытой архитектуры. Определены требования к программному комплексу и его структура.

Основные особенности разработанного комплекса:

1. **Ориентация на работу в сети.** Система обеспечивает работу в режиме "клиент-сервер" по ЛВС и через Internet с использованием различных технологий. Она включает специализированный протокол обмена

данными для поддержки распределенных вычислений. Этот протокол позволяет определить работоспособность узлов сети и вспомогательного программного обеспечения и выполняет реконфигурацию системы в случае возникновения сбоев.

2. **Открытость**, включающая в себя:

- Изменение режимов и параметров функционирования программного обеспечения, входящего в состав ядра системы, путем настройки параметров в файлах конфигурации.
- Возможность использования различных учебных материалов на машинных носителях и методов подачи этих материалов.
- Организация внешнего управления процессом обучения. Это достигается путем вынесения во внешние файлы (в базу данных и в базу знаний) правил организации управления обучением и всех возможных параметров и характеристик, которые можно измерить в процессе работы с обучаемым.

3. **Переносимость на различные платформы** на уровне исходных текстов программ.

Для системы разработан декларативный язык описания тестов, позволяющий хранить описание тестов во внешних по отношению к системе файлах, что делает тест независимым от программного обеспечения.

На основе спроектированного языка описания тестов и разработанных дифференцированных методов создана система контроля знаний, в которой реализованы рассмотренные ранее формы представления ответов.

Открытость архитектуры СКЗ также обеспечивается возможностью подключения внешних модулей для выполнения следующих функций:

- реализации тех форм ответов, которые не включены в систему;
- получения оценок ответов;
- управления формированием теста;
- управления проведением тестирования.

В рамках СКЗ созданы базы знаний, реализующие:

- формирование теста на основе параметров вопросов для различных схем тестирования;
- проведение тестирования в соответствии с линейной, пороговой и адаптивной схемами.

Система контроля знаний реализована в двух вариантах: как приложение под ОС Windows (система TS) и как консольное приложение (система IPDO).

С использованием системы IPDO в режиме удаленного доступа через Internet проведено тестирование студентов. Результаты экспериментов подтверждают сокращение времени, затрачиваемого преподавателем на проведение контрольных мероприятий.

Анализ используемых в тестах типов ответов показал, что составители тестов отдают предпочтение расширенным типам ответов (двухуровневые схемы и свободно-конструируемые ответы: текст, графики, формулы, арифметические выражения).

Заключение

К основным результатам работы можно отнести следующие:

1. Проведен анализ существующих автоматизированных обучающих систем. Предложен метод многомерной классификации АОС, использование которого позволяет проводить сравнительный анализ обучающих систем. Показана необходимость ориентации АОС на исследование сетевых технологий для создания распределенных систем (РАОС), обеспечивающих удаленный доступ к системе и объединение сетевых ресурсов для решения стоящих перед системой задач.
2. Предложен способ построения полнофункциональных РАОС с открытой архитектурой, применение которого обеспечивает:
 - использование вычислительных ресурсов сети для выполнения функций обучающей системы;
 - осуществление адаптивного управления процессом обучения на основе знаний о предметной области и об обучаемом;
 - использование в обучающих системах произвольных методик преподавания и контроля знаний.
3. Сформулированы и решены следующие задачи, возникающие при создании РАОС:
 - организация распределенных вычислений и повышение надежности функционирования системы путем ее динамической реконфигурации;
 - организация внешнего управления взаимодействием с обучаемым путем вынесения правил управления во внешний модуль для обеспечения возможности алгоритмической настройки системы без необходимости программирования;
 - поддержка пользовательских параметров для организации параметрической настройки системы.
4. Для оценки выборочных ответов разработан Д-метод, который не зависит от семантики вопроса и позволяет проводить оценку дифференцировано, а

не по двухбалльной шкале. Показано, что в качестве моделей свободно-конструируемых ответов (в том числе, для качественного анализа графиков) могут использоваться списки и множества. Для определения правильности свободно-конструируемых ответов при таком подходе может применяться Д-метод.

5. Разработана подсистема контроля знаний, обеспечивающая более полные возможности представления и анализа ответов обучаемых и включающая:
 - средства поддержки удаленного доступа для тестирования в рамках ЛВС и через Internet;
 - базы знаний, управляющие формированием теста на основе набора контрольных заданий и проведением тестирования по различным методикам;
 - методы сравнения множеств и списков.

Получены расчетные и эмпирические оценки вероятности случайного ввода правильного ответа или получения положительной оценки для различных форм выборочных ответов, что позволило сформулировать рекомендации по составлению контрольных заданий.

6. На основе разработанных методов и алгоритмов создан макетный программный комплекс РАОС и подсистема контроля знаний, которая в настоящее время используется в учебном процессе. Результаты анализа работы с этим комплексом подтвердили основные положения диссертационного исследования.

СПИСОК ЛИТЕРАТУРЫ

1. Аванесов В.С. Научные проблемы тестового контроля знаний / Монография. – М.: Исследовательский центр проблем качества подготовки специалистов, 1994. – 135 с.
2. Агеев В.Н. Электронные учебники и автоматизированные обучающие системы. – М.: 2001. – 79 с.
3. Аджемов А.С. Единое образовательное пространство на основе инфотелекоммуникационных технологий // Сети и системы связи, 2001, №11. – с. 20-23.
4. Александров Г.Н. Программированное обучение и новые информационные технологии обучения. // Информатика и образование, 1993, №5. – с. 7-19.
5. Андерсон Дж. Р., Рейзер Б. Дж. Учитель Лиспа. / В сб. "Реальность и прогнозы искусственного интеллекта" под ред. Стефанюка В.Л. / Пер. с англ. – М.: Мир, 1987. – с. 27-47.
6. Андреев А. Определимся в понятиях. // Высшее образование, 1998, № 4. – с. 53-58.
7. Аттель У. Обучающая вычислительная машина: моделирование в истинном масштабе времени обучающего диалога / В сб. "Кибернетика и проблемы обучения" / Ред. и предисл. А.И. Берга. – М.: Прогресс, 1970. – с. 206-228.
8. Афанасьев В.В., Афанасьева И.В., Тыщенко О.Б. Основные компоненты компьютерных технологий обучения // НИИВО 23.04.98, № 86-98, деп. Муром. ин-т, фил. Владим. гос. Ун-та. – Муром: 1998.
9. Барина С.Н. Автоматизированные учебные курсы и их влияние на качество процесса обучения / Материалы конференции "Информационные технологии в образовании", 1999. – <http://ito.bitpro.ru/>
10. Березин Н.В. Перспективы создания системы адаптивного тестирования как элемента централизованного тестирования / Научный вестник МГТУ ГА, серия "Информатика", 2001, № 38. – с. 26-30.
11. Беркгаут В.В., Чардин И.С. Интернет: первые шаги. – М.: Наука, 1999. – с.66-69.
12. Беспалько В.П. Педагогика и прогрессивные технологии обучения. – М.: 1995.

13. Бирюков В.В. Программированное обучение автокоду "Инженер" с использованием многопультной системы. / В кн.: Теория и применение математических машин / Под ред. А.М. Оранского, Н.Н. Поснова. – Мн.: Изд-во БГУ, 1972. – с. 213-216.
14. Большая Советская Энциклопедия, 3-е издание, 1970, т. 18. – с. 236.
15. Брусиловский П.Л. Адаптивные обучающие системы в Word Wide Web: обзор имеющихся в распоряжении технологий. –
<http://ifets.ieee.org/russian/depository/WWWITS.html>
16. Брусиловский П.Л., Зырянов М.И. Интеллектуальная учебная среда "Остров". // 3-я Конференция по искусственному интеллекту. – Тверь: Ассоциация искусственного интеллекта, 1992. – с.33-35.
17. Булгаков М.В., Якивчук Е.Е. Инструментальные системы для разработки обучающих программ / В кн. "Компьютерные технологии в высшем образовании". / Ред. кол.: А.Н. Тихонов, В.А. Садовничий и др. – М.: Изд-во Моск. ун-та., 1994. – с. 153-162.
18. Валеева Н. Дополнительное образование студентов // Высшее образование в России, № 3, 1998. – с.27-29.
19. Васильев В.Н., Сигалов А.В. Информационные технологии в учебном процессе СПбГИТМО // Материалы Международной конференции-выставки "Информационные технологии в непрерывном образовании" // Петрозаводск, 5-9 июня 1995 г. – <http://petsu.karelia.ru/psu/general/conferences/data/19950605>
20. Васин Б.И., Галаев Д.А., Лаптев В.С. Инструментальные программные средства для организации дистанционного обучения, учитывающие требования международных стандартов // Конференция RELARN-2001. Тезисы докладов. – <http://www.relarn.ru>
21. Владимирский Б. М. Роль и место когнитивной машинной графики в обучении // Тезисы докладов уч.-мет. конференции "Современные информационные технологии в учебном процессе" – Ростов: РГУ, 25-26 апреля 2000.
22. Вопросы создания автоматизированных обучающих систем на базе ЭВМ. – М., 1976.
23. Воронин А.Т., Чернышев Ю.А. Интеллектуальная инструментальная система для WINDOWS // Материалы Международной конференции-выставки

- "Информационные технологии в непрерывном образовании". – Петрозаводск, 5-9 июня 1995 г. // <http://petrsu.karelia.ru/psu/general/Conferences/Data/19950605>
24. Гиркин И.В. Новые подходы к организации учебного процесса с использованием современных компьютерных технологий // Информационные технологии, 1998, № 6. – с. 44-47.
25. Горюнов Ю.П. Логическая структура курса и обучающий алгоритм курса. / В сб.: Программированное обучение и кибернетические обучающие машины. / Под ред. Шестакова А.И. – М.: Сов. Радио, 1963. – с. 24-31.
26. Грибкова В.А., Зайцева Л.В., Новицкий Л.П. Управление адаптивным диалогом в автоматизированных обучающих системах. Методические указания. – Рига: РПИ, 1988. – 52 с.
27. Джалалуддин А.К. Применение компьютеров для целей непрерывного образования // Перспективы, 1991, № 2. – с. 72-89.
28. Джалиашвили З.О., Николаев Д.Г. Сетевые технологии как эффективное средство поддержки дистанционного обучения // Материалы конференции "Информационные технологии в образовании", 2001. – <http://www.bitpro.ru/>
29. Джонсон У.Л., Солоуэй Э. PROUST (автоматический отладчик программ на языке Паскаль) / В сб. "Реальность и прогнозы искусственного интеллекта" под ред. Стефанюка В.Л. / Пер. с англ. – М.: Мир, 1987. – с. 48-70.
30. Домрачев В.Г., Ретинская И.В. О классификации образовательных информационных технологий // Информационные технологии, 1996, № 2. – с. 10-13.
31. Дубровский В.Н. От компьютера-книги к компьютеру-учителю: принципы разработки комплекса "1С:РЕПЕТИТОР. Математика" // Материалы конференции "Информационные технологии в образовании", 1999. – <http://www.bitpro.ru/>
32. Журавлева И.И. Интеллектуальные обучающие системы и дистанционном образовании // Материалы конференции "Информационные технологии в образовании", 2001. – <http://www.bitpro.ru/>
33. Зайцева Ж.Н., Солдаткин В.И. Генезис виртуальной образовательной среды на основе интенсификации информационных процессов современного общества // Информационные технологии, 2000, № 3. – с. 44-48.

34. Зенкин А.А. Когнитивная компьютерная графика / Под ред. Д.А. Пospelова. – М.: Наука, Гл. ред. физ.-мат. лит., 1991. – 192 с.
35. Искусственный интеллект : В 3-х кн. Кн. 1. Системы общения и экспертные системы: Справочник / Под ред. Э.В. Попова. – М.: Радио и связь, 1990. – 464 с.
36. Использование сети ISDN в системе дистанционного образования // НПЦ Инфосфера – <http://www.infosfera.ru/>
37. Камер Дуглас Э. Компьютерные сети и Internet. Разработка приложений для Internet : Пер. с англ. – М.: Изд. дом "Вильямс", 2002. – 640 с.
38. Каперко А.Ф., Карпов В.Э., Королев А.В. Система управления сложным техническом объектом на основе распределенной операционной системы реального времени // Датчики и системы, 2001, №2. – с.18-21.
39. Карлащук В.И. Обучающие программы. М.: "СОЛОН-Р", 2001. – 528 с.
40. Карпов В.Э., Карпова И.П. К вопросу о классификации систем // Информационные технологии, 2002, №2. – с. 35-38.
41. Карпов В.Э., Карпова И.П. Язык описания системы контроля знаний // Компьютеры в учебном процессе, 2000, № 4. – с. 147-155.
42. Карташева О.В. Использование адаптивной системы тестирования АСТ-Тест для контроля знаний при дистанционном изучении темы "Базы данных" // Материалы конференции "Информационные технологии в образовании", 2001. – <http://www.bitpro.ru/>
43. Кибернетика и проблемы обучения: Сборник переводов / Ред. и предисл. А.И. Берга. – М.: Прогресс, 1970. – 389 с.
44. Клаудио де Мора Кастро, Торкель Альфтан. Компьютеры во внешкольном образовании // Перспективы, 1991, № 2. – с. 59-71.
45. Кнут Д. Искусство программирования для ЭВМ / т.3. Сортировка и поиск / Пер. с англ. / Под ред. Баяковского и Штаркмана. – М.: Мир, 1978. – 848 с.
46. Колесников А.О. Реализация авторской системы в сетевой среде // <http://www.ulstu.ru/conf/NMK99/2-14.htm>
47. Компьютерные технологии в высшем образовании. / Ред. кол.: А.Н. Тихонов, В.А. Садовничий и др.- М.: Изд-во Моск. ун-та., 1994. – 272 с.
48. Коутс Р., Влейминк И. Интерфейс "человек-компьютер": Пер. с англ. – М.: Мир, 1990. – 501 с.

49. Кривец В.А. Автоматизированный класс на базе ОЭМ-2. / В кн.: Теория и применение математических машин / Под ред. А.М. Оранского, Н.Н. Поснова. – Мн.: Изд-во БГУ, 1972 – с. 209-213.
50. Кривицкий Б.Х. О систематизации учебных компьютерных средств // Кафедра педагогики, психологии и методики преподавания в высшей школе МГУ. – http://ifets.ieee.org/russian/depository/v3_i3/html/3.html
51. Кривошеев А.О., Голомидов Г.С., Таран А.Н. Перспективные Internet-технологии информационного обеспечения образовательных услуг // Российский НИИ информационных систем, 2000.
52. Кривошеев А.О. Компьютерные обучающие программы. Состояние и перспективы развития // Мат-лы научно-технич. конференции "Перспективные информационные технологии в высшей школе". Самара, 1993. – с. 18-20.
53. Кривошеев А.О. Проблемы оценки качества программных средств учебного назначения // Сборник докладов 1-го научно-практического семинара "Оценка качества программных средств учебного назначения". – М.: Гуманитарий, 1995. – с. 5-12.
54. Кривошеев А.О., Фомин С.С. Конкурс "Электронный учебник" / В кн. "Компьютерные технологии в высшем образовании". / Ред. кол.: А.Н. Тихонов, В.А. Садовничий и др. – М.: Изд-во Моск. ун-та., 1994. – с. 264-268.
55. Кручинин В.В., Ямпольский С.З. Инструментальные системы для создания электронных учебников, практикумов, экзаменаторов // Материалы конференции "Новые информац. технологии в университетском образовании" – Новосибирск: 1995. – <http://src.nsu.ru/conf/nit/95/>
56. Кулешов В. О дистанционном обучении студентов специальности. "Радиотехника" в МЭИ. // Радиотехнические тетради, 1994, № 6. – с. 65-66.
57. Лекции по теории графов / Емеличев В.А., Мельников О.И. Сарванов В.И., Тышкевич Р.И. – М.: Наука, Гл. ред. физ.-мат. лит., 1990. – 384 с.
58. Лехто Г.Ф., Тарасов В.А. Экспертные технологии контроля и диагностики знаний обучаемых // Материалы Международной конференции-выставки "Информационные технологии в непрерывном образовании" // Петрозаводск, 5-9 июня 1995 г. – <http://petrsu.karelia.ru/psu/general/conferences/data/19950605/>

59. Логический подход к искусственному интеллекту: от классической логики к логическому программированию / Пер. с франц. – М.: Мир, 1990. – 432 с.
60. Лозинский Л.Д. Математические пакеты в высшей школе // Мир ПК, 1992, № 9. – с. 89-97.
61. Лорьер Ж.-Л. Системы искусственного интеллекта. / Пер. с франц. – М.: Мир, 1991. – 568 с. (Алгоритм унификации – с. 116-161.)
62. Лукас В.А. Теория автоматического управления: Учебник для вузов. – 2-е изд. – М.: Недра, 1990. – 416 с.
63. Мазурина С.М. Разработка моделей представления и обработки знаний в продукционных экспертно-обучающих системах // Диссертация на соискание звания канд. техн. наук по специальности 05.13.11. – М.: МГИЭМ, 1995.
64. Малышев Ю.А., Нежурина М.И., Шатровский В.А. Технологии представления учебных курсов для дистанционной формы обучения в среде WWW. // Информационные технологии, 1997, № 6. – с. 39-42.
65. Мамиконов А.Г. Принятие решений и информация. – М.: Наука, 1983. – 184 с.
66. Марселлус Д. Программирование экспертных систем на Турбо Прологе: Пер. с англ. / Предисл. С.В. Трубицына. – М.: Финансы и статистика, 1994. – 256 с.
67. Мартынов Д.В., Смольникова И.А. Искусственный интеллект и образование. // Тезисы научно-мет. конференции "Информационные технологии в образовании", Москва, 1999. – <http://ito.bitpro.ru/>
68. Мартыанов С.В., Недоумов А.Н., Щербаков Н.В. HYPER-PC и HM-CARD – инструменты для создания обучающих и hypermedia систем. / В кн. "Компьютерные технологии в высшем образовании". /Ред. кол.: А.Н. Тихонов, В.А. Садовничий и др. – М.: Изд-во Моск. ун-та., 1994, с. 32-41.
69. Материалы 2-й Международной выставки-конференции "Информационные технологии и телекоммуникации в образовании" // Каталог и тезисы докладов // Москва, ВВЦ, 6-9 апреля 2000 г.
70. Материалы конференции "Новые информационные технологии в университетском образовании" – Новосибирск: 1997. – <http://www.nsu.ru>
71. Микрокомпьютерная система обучения "Наставник": Брусенцов и др. – М.: Наука, 1990. – 224 с.

72. Морозевич А.И., Комличенко В.Н., Гедранович В.В. Стратегия автоматизации управления познавательной деятельностью на основе информационной модели образовательного процесса // Информационные технологии, 2000, № 5. – с. 47-52.
73. Нетушил А.В., Никитин А.В. О методе синтеза учебных программ // Проблемы нейрокибернетики. – Ростов-на-Дону: Из-во Ростов. ун-та, 1969. – с. 236-243.
74. Норенков Ю.И. Исследование и разработка принципов построения адаптивных обучающих систем. / Автореферат. М.: 1993. – 20 с.
75. Обучающие машины, системы и комплексы: Справочник / Под ред. А.Я. Савельева. – Киев: Вища шк., 1986. – 303 с.
76. Орехов В.Д. Дистанционная технология переподготовки руководителей для работы в рыночных условиях // Машиностроитель, 1995, № 4-5.
77. Оцуки С. Приобретение знаний и обучение в диалоге // В кн. "Приобретение знаний": Пер. с япон. / Под ред. С. Осуги, Ю. Саэки. – М.: Мир, 1990. – 304 с.
78. Пасхин Е.Н., Митин А.И. Автоматизированная система обучения ЭКСТЕРН. – М.: Изд-во Моск. ун-та, 1985. – 144 с.
79. Петрик Ю.С., Афанасьев А.С., Лица Д.В., Заугольников Н.С. Возможности дифференцированной оценки качества знаний студентов в процессе их адаптивного тестирования // Тезисы докладов конференции в Туле. – <http://cam.psn.ru/conferences/tula99/theses/petric2.html>
80. Петрушин В.А. Экспертно-обучающие системы. – Киев: Наукова думка, 1991. – 196 с.
81. Попов Э.В. Общение с ЭВМ на естественном языке. М.: Наука, 1982. – 360 с.
82. Поспелов Д.А. Семиотические модели в управлении. / В кн. "Кибернетика. Дела практические". – М.: Наука, 1984. – с. 70-87.
83. Представление и использование знаний: Пер. с япон. / Под ред. Х. Уэно, М. Исидзука. – М.: Мир, 1989. – 220 с.
84. Применение ЭВМ в учебном процессе / Сборник докладов научно-технич. семинара под ред. А.И. Берга. – М.: Сов. радио, 1969. – 248 с.
85. Приобретение знаний: Пер. с япон. / Под ред. С. Осуги, Ю. Саэки. – М.: Мир, 1990. – 304 с.

86. Проблемы создания автоматизированных обучающих и тестирующих систем: Сборник науч. трудов / Редколл. Иванченко А.И. и др. – Новочеркасск, 2001. – 199 с.
87. Программированное обучение и кибернетические обучающие машины: Сборник статей под ред. Шестакова А.И. – М.: Сов. радио, 1963. – 247 с.
88. Программированное обучение и обучающие машины / Труды научно-технич. семинара. – Киев: Вып. 2, 1967.
89. Разработка и исследование системных средств и прикладных программ для автоматизации обучения и научных исследований на базе ЭВМ: Отчет по НИР / № гос. рег. 01860022813 – Мн.: БГУ, НИИ ПФП, 1990.
90. Растригин Л.А. Вычислительные машины, системы, сети... – М.: Наука. Гл. ред. физ.-мат. лит., 1982. – 224 с.
91. Растригин Л.А., Эренштейн М.Х. Адаптивное обучение с моделью обучаемого. – Рига: Зинатне, 1988. – 160 с.
92. Ретинская И.В. Системы и методы поддержки принятия решений по оценке качества и выбору компьютерных средств учебного назначения (Обзор). // Информационные технологии, 1997, № 6. – с. 42-44.
93. Ретинская И.В., Шугрина М.В. Отечественные системы для создания компьютерных учебных курсов. // Мир ПК, 1993, № 7. – с. 55-62.
94. Романов В.П. Эволюция образовательных технологий в свете интеллектуализации информационных систем (об одном подходе к формированию информационного образовательного ресурса) // Материалы конф. "Информационные технологии в образовании", 1999. – <http://ito.bitpro.ru/>
95. Ростунов Т.И. Сущность программированного метода обучения. / В сб.: Программированное обучение и кибернетические обучающие машины. / Под ред. Шестакова А.И. – М.: Сов. радио, 1963. – с. 10-23.
96. Рубин Ю.Б., Самойлов В.А., Шевченко К.К. Технологические системы в открытом образовании // Материалы конференции "Информационные технологии в образовании", 2001. – <http://www.bitpro.ru/>
97. Рудинский И.Д. Принципы интеллектуального автоматизированного тестирования знаний // Материалы конференции "Информационные технологии в образовании", 2001. – <http://www.bitpro.ru/>

98. Савельев А.Я. Автоматизированные обучающие системы на базе ЭВМ / вып.1./ М.: Знание, 1977. – 36 с.
99. Савельев А.Я., Новиков В.А., Лобанов Ю.И. Подготовка информации для автоматизированных обучающих систем: Метод. пособие для преподавателей и студентов / Под ред. А.Я. Савельева. – М.: Высшая школа, 1986. – 176 с.
100. Свиридов А.П. Основы статистической теории обучения и контроля знаний: Метод. пособие. – М.: Высшая школа, 1981. – 262 с.
101. Сивохин А.В. Представление знаний в интеллектуальных системах обучения / Уч. пос. – Пенза: ППИ, 1990. – 86 с.
102. Сливина Н.А., Чубров Е.В. Приобретение знаний по математике с использованием учебных и научных пакетов / В кн. "Компьютерные технологии в высшем образовании". / Ред. кол.: А.Н. Тихонов, В.А. Садовничий и др. – М.: Изд-во Моск. ун-та., 1994.
103. Соловов А.В. Проектирование компьютерных систем учебного назначения: Учебное пособие. – Самара: СГАУ, 1995. – 137 с.
104. Соломатин Н.М., Сонин А.И., Соколов Н.К. Серебрякова И.Л., Семенов Д.В. Особенности дистанционного обучения в системе высшего образования // Вестник МГТУ им. Баумана, сер. Приборостроение, 1998, № 2. – с. 101-108.
105. Стенли Р. Перечислительная комбинаторика: Пер. с англ. – М.: Мир, 1990. – 440 с.
106. Стоунс Е. Стратегия и тактика программированного обучения. / В сб. "Кибернетика и проблемы обучения" / Редакция и предисловие А.И. Берга. – М.: Прогресс, 1970. – с. 255-265.
107. Талызина Н.Ф. Теоретические проблемы программированного обучения. – М.: Изд-во МГУ, 1969. – 133 с.
108. Тезисы докладов Международной конференции-выставки "Информационные технологии в непрерывном образовании" – Петрозаводск: 5-9 июня 1995г. -<http://petsu.karelia.ru/psu/general/conferences/data/19950605/>
109. Тезисы докладов уч.-мет. конференции "Современные информационные технологии в учебном процессе" – Ростов: РГУ, 25-26 апреля 2000 г. – <http://www.uic.rsu.ru/~nprohoro/DO/>

110. Терещенко Л.Я., Панов В.П., Майоркин С.Г. Управление обучением с помощью ЭВМ. – Л.: Изд-во ЛГУ, 1981. – 143 с.
111. Толингерова Д. Программирование и управление обучением / В сб.: "Кибернетика и проблемы обучения" / Редакция и предисловие А.И. Берга. – М.: Прогресс, 1970. – с. 236-254.
112. Трапезников С.Н. УРОК – универсальный редактор обучающих курсов / В кн. "Компьютерные технологии в высшем образовании" / Ред. колл.: А.Н. Тихонов, В.А. Садовничий и др. – М.: Изд-во Моск. ун-та., 1994. – с. 23-32.
113. Ту Дж., Гонсалес Р. Принципы распознавания образов. / Пер с англ. Под ред. Ю.И. Журавлева. – М.: Мир, 1978. – 414 с.
114. Тюрина Л. Вузовский учебник сегодня и завтра. // Высшее образование, 1998, № 1. – с. 11-20.
115. Убейко В.М., Убейко В.В. Экспертные системы в технике и экономике. – М.: Изд-во МАИ, 1992. – 240 с.
116. Уотермен Д. Руководство по экспертным системам: Пер. с англ. – М.: Мир, 1989. – 388 с.
117. Фор А. Восприятие и распознавание образов / Пер. с фр. / Под ред. Г.П. Катыса. – М.: Машиностроение, 1989. – 272 с.
118. Фу К. Последовательные методы в распознавании образов и обучении машин: Пер. с англ. / М.: Наука, Гл. ред. физ.-мат. лит., 1971. – 256 с.
119. Фу К. Структурные методы в распознавании образов: Пер. с англ. / М.: Мир, 1977. – 320 с.
120. Хартли Д. К вопросу об оценке обучающих программ. / В сб. "Кибернетика и проблемы обучения" / Редакция и предисловие А.И. Берга. – М.: Прогресс, 1970. – с. 350-387.
121. Чиликин М.Г. Основные задачи программированного обучения. / В сб.: Программированное обучение и кибернетические обучающие машины. / Под ред. Шестакова А.И. – М.: Сов. радио, 1963. – с. 5-9.
122. Чубров Е.В., Сливина Н.А., Демушкин А.С. Компьютер и изучение математики. // ИНФО, 1992, №3-4. – с. 96-97.
123. Шаммазов А., Беленкова О. Технические университеты в информационно-индустриальном обществе // Высшее образование, 1998, №1. – с. 20-25.

124. Шампанер Г., Шайдук А. Обучающие компьютерные системы // Высшее образование в России, 1998, № 3. – с. 97-99.
125. Шемакин Ю.И. Начала компьютерной лингвистики: Учеб. пособие – М.: Изд-во МГОУ, А/О "Росвузнаука", 1992. – 115 с.
126. Шмелев А.Г., Бельцер А.И., Ларионов А.Г., Серебряков А.Г. Адаптивное тестирование знаний в системе "ТЕЛЕТЕСТИНГ" // IX конференция-выставка "Информационные технологии в образовании", 1999. – <http://ito.bitpro.ru/>
127. Bloom B.S. The sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring // Education Researcher, № 13, 1984. – p. 3.
128. Bork A. Computer and Information Technology as a learning Aid // Education and Computing, 1985, v.1, № 1. – p. 29-34.
129. Buiten R., Lane H.S. Experimental system gives language student instant error feedback. / Digital Equipment Corporation Computer Application Note, 1965.
130. Coulson J.E. Computers in research and development on automated instruction. // "Proceedings of the IV-th international congress of cybernetic medicine", Nice, 1966. – p. 241-257.
131. Distance Learning Studio. Система создания мультимедийных дистанционных курсов: Документация // Составители: Пушков А.И., Ченосова Е.С., Юрков А.В. – С.-Петербург, 2000. – 272 с.
132. Etienne Wenger. Artificial Intelligence and Tutoring Systems (Computational and Cognitive Approaches to the Communication of Knowledge) // Morgan Kaufmann Publishers. – Los Altos, California, USA, 1987. – 487 p.
133. Haan B, Kahn P. IRIS Hypermedia Services // Communications of ACM, 1992, v.35, № 1. – p. 36-51.
134. Hebenstreit J. Computers in education – The next step // Education and Computing, v.1, 1995. – p. 37-43.
135. Intelligent Tutoring Systems: Proceedings of Second International Conference // University de Montreal. – Montreal, Canada, 1988. – 578 p.
136. Intelligent Tutoring Systems: Proceedings of Second International Conference // University de Montreal. – Montreal, Canada, 1992. – 422 p.
137. Licklider J. Preliminary experiments in computer-aided teaching. // "Programmed Learning and Computer Based Instruction". – New York, Wiley, 1962. – p. 217-239.

138. Morris J. The case for CAI. – SIGCUE bulletin, 1984, v.18. – p. 11-14.
139. Patrick W. Thompson. Mathematical Microworlds and Intelligent Computer-assisted Instruction. In: "Artificial Intelligence and Instruction". Ed: Kearsly, 1987. – p. 83-109.
140. Providing computing for distance learners: a strategy for home use. // Computers Education, 1992, vol.18, № 1.
141. Ronald G. Ragsdale. Effective computing in education: tools and training // Education and computing, 1991, v.7. – p. 157-166.
142. Skinner B.F. The science of learning and art of teaching. // Harvard Education Review, Spring, 24, 1954. – p. 86-97.
143. Uhr L. The compilation of natural language text into teaching machine programs. // American Federation of Information Processing Societies Conference Proceedings, 1964. – p. 26-35.
144. Uttal W.R. On conversational interaction // "Programmed Learning and Computer Based Instruction". – New York, Wiley, 1962.
145. Zaitseva L., John D. Zakis. Course Development for Tutoring and Training Systems in Engineering Education / Global J. of Engng. Educ., 1991, vol. 1, № 3. Printed in Australia.

Приложение 1. Наборы правил базы знаний для управления контролем знаний

База знаний реализована на языке PDC Prolog.

Описание теста

Описание теста из соответствующих файлов (с расширением TDL) автоматически конвертируется в файл формата базы данных языка Пролог (database). Этот файл содержит следующие факты:

```
% для блоков КОНТРОЛЬ_ПО_ТЕМЕ
% control(название_темы,
%   тип_контроля: ПРОМЕЖУТОЧНЫЙ - 0
%   ЗАВЕРШАЮЩИЙ - 1
%   ВЫБОРОЧНЫЙ - 2
%   способ_выбора: СЛУЧАЙНО_БЕЗ_ПОВТОРОВ - 0
%   СЛУЧАЙНО - 1
%   ПОЛЬЗОВАТЕЛЬ - 2
%   ПОДРЯД - 3
%   способ_выдачи: ОДИН - 0, ВСЕ - 1
%   количество_вопросов, количество_ответов, оценка,
%   тип_оценки : СУММА - 0
%   ПРОЦЕНТ - 1
%   ШКАЛА - 2
%   минимальный_балл, максимальный_балл)
% для блоков ВОПРОС
% question(название_темы, название_вопроса, название_свойства, значение_свойства)
% для блоков ОТВЕТ
% answer(название_темы, название_вопроса,
%   тип_ответа : МНОЖЕСТВО - 0
%   СПИСОК - 1
%   ВЫРАЖЕНИЕ - 2
%   ТЕКСТ - 3
%   эталон,
%   тип_оценки : СХОДСТВО - 0, РАВНО - 1
%   погрешность, минимальный_балл, максимальный_балл)
% для управления порядком выдачи вопросов
% order(свойство, ASC | DESC )
```

Набор правил, реализующих адаптивную схему тестирования

domains

```
str = string
slist = string *
file = datafile
```

database

```
% предикаты описания теста
control(str,str,str,str,str,str,str,str,str,str)
question(str,str,str,str)
qout(string)
answer(str,str,str,str,str,str,str,str)
```

```

order(str,str) % сортировка списка вопросов по значению свойства
list(slist)
% предикаты задания (файл с параметрами командной строки)
command(symbol) % исполняемый предикат
dbname(string) % название файла базы данных
cntype(string) % тип контроля
thema(string) % название темы
prevquery(string) % название предыдущего вопроса
mark(real) % оценка за предыдущий вопрос
% название группирующего свойства для адаптивного опроса
group(str)
predicates
nondeterm prlist(slist) % печать списка вопросов
nondeterm usage(string) % если ком. строка пуста, выдать сообщение
execute(symbol) % выполнить предикат symbol
% вспомогательные предикаты
ltrim(str, str) % удаление пробелов слева в строке
rtrim(str, str) % удаление пробелов справа в строке
alltrim(str, str) % удаление пробелов слева и справа в строке
% формирование списка вопросов по группам
formgroup
% посчитать количество групп и вопросов, вернуть список вопросов (без повторов)
nondeterm compute(str,integer,integer,slist)
% вернуть минимум из 1-го и 2-го аргументов
nondeterm min(integer,integer,integer)
% добавить к БД все вопросы, принадлежащие % к группе вопроса str
move(str,slist)
% вывод списка вопросов в БД (list(вопросы))
nondeterm callprint(slist)
% анализ списка вопросов
nondeterm qlist(slist,str,integer,integer)
% выбор вопросов из списка (i,i,i)
% (список, способ выбора, кол-во выбир.вопросов, длина списка)
% результат - в БД - list(список выбранных вопросов)
nondeterm choice(slist,str,integer)
% удаление повторяющихся элементов из списка
uniq(slist,slist,slist)
% вхождение элемента в список
nondeterm elem(string,slist)
% добавить в список элемент str, если он уже есть - не добавлять
nondeterm append(slist,str,slist)
% выбор вопросов случайно без повторов
nondeterm ql_random(integer)
% выбор вопросов случайно
nondeterm ql_onrand(integer)
% добавить в БД integer вопросов из списка
nondeterm asslist(slist,integer)
% добавить в БД все вопросы из списка
nondeterm printlist(slist)
% вызов сортировки по условиям ORDER
callsort(slist,slist,slist)
% сортировка списка

```

```

nondeterm sort(str,slist,slist)
% поиск в списке элемента номер integer
nondeterm memnum(integer,slist,str)
% удаление из списка элемента str
nondeterm delete(str,slist,slist)
% удаление из 1-го списка 2-го списка
nondeterm delfromlist(slist,slist,slist,slist)
% определение длины списка
lenlist(slist,integer)
% вспомогательная функция, возвращает i1 или i2, если i2 < 0;
nondeterm checknum(integer,integer,integer)
% принимает тему, тип контроля, длину списка вопросов по этой теме;
% возвращает способ выбора, количество вопросов и ответов (i,i,i,o,o,o)
nondeterm getinfo(str,str,integer,str,integer,integer)
% следующий вопрос адаптивного опроса (i,o), (предыдущий, следующий)
anextquery(str,str)
% следующий вопрос из группы (если он есть
% и максимальная оценка за него выше, чем предыдущая)
nondeterm nextqout(slist,str,real,str)
% первый вопрос из следующей группы (i,i,o)
nondeterm nextgroup(slist,str,str)
% удалить начало списка до элемента str (i,i,o)
nondeterm head(slist,str,slist)
% сортировка списка по значению свойства
insert(string,string,slist,slist,integer)
% свойство, элемент, вх.список, вых.список, способ упорядочения
insert_sort(string,slist,slist,integer)
acs(string,string,string) % упорядочение по возрастанию (0)
decs(string,string,string) % упорядочение по убыванию (1)

```

goal

```

envsymbol("TS",Var),          % получить значение переменной TS
concat(Var,"\\swap\\kbp.out",SWP), % определить путь к файлу обмена
comline(C),                   % получить командную строку
usage(C), C <> "",            % если она пуста, выдать сообщение
consult(C),                   % подключить файл с заданием
command(Cmd),                 % считать из него предикат задания
openwrite(datafile,SWP),      % открыть swap-файл
writedevice(datafile),        % перенаправить в него выходной поток
write(" "),
execute(Cmd),                  % выполнить предикат
closefile(datafile),          % закрыть swap-файл
save("cnt.db").               % сохранить БД

```

clauses

```

usage("") :- write("Usage: task_filename"),nl.
usage(_).

execute("formgroup") :-
  dbname(DBname),
  consult(DBname),
  formgroup,!.
```

```

execute("anextquery") :-
    consult("cnt.db"), % для выдачи следующего вопроса
    prevquery(Q),
    anextquery(Q,Next),
    write(0),nl,
    write(Next),nl,
    !.
execute(_) :- !.

ltrim("", "") :- !.
ltrim(A,B) :-
    frontchar(A,' ',T1), !,
    ltrim(T1, B).
ltrim(X,X) :- !.
rtrim("", "") :- !.
rtrim(A,B) :-
    concat(T1," ",A), !,
    rtrim(T1, B).
rtrim(X,X) :- !.

alltrim(S, Res) :-
    rtrim(S,S1),
    ltrim(S1, Res).

min(I1,I2,I1) :- I1<I2.
min(_,I2,I2).

formgroup :-
    cntype(Ctype),
    thema(S),
    alltrim(S,Thema),
    compute(Thema,NG,Len,F),
    getinfo(Thema,Ctype,Len,Choice,N1,N2),
    min(NG,N1,NQ),
    min(NQ,N2,NA),
    qlist(F,Choice,NQ,NG),
    write(0),nl,
    write(NQ),nl,
    write(NA),nl,!

formgroup :-
    write(1),nl,
    write("Can't find control_block for this thema"),nl.

compute("",0,0,[]).
compute(Thema,NG,Len,F) :-
    findall(TName,question(Thema,TName,_,_),Q),
    uniq(Q,[],F),
    lenlist(F,Len),
    group(Pname),
    findall(Val,question(Thema,_,Pname,Val),P),
    uniq(P,[],G),
    lenlist(G,NG).

```

```

elem(_,[]) :- fail. % поиск элемента X в списке
elem(X,[X|_]).
elem(X,[_|H]) :- elem(X,H).

```

```

append([],H,[H]). % добавление нового элемента в список
append(L,H,L) :- elem(H,L).
append(L,H,[H|L]).

```

```

uniq([],L,L). % удаление из списка повторяющихся элементов
uniq([H|T],L,R) :-
    append(L,H,S),
    uniq(T,S,R).

```

```

lenlist([],0). % длина списка
lenlist([_|T],L) :- lenlist(T,N), L=N+1.

```

```

callprint([]).
callprint(Q) :-
    findall(Prop,order(Prop,_),F),
    callsort(F,Q,S),
    printlist(S),!.

```

```

qlist(_,_,0,_).
qlist([],_,_,_).
qlist(Q,Ctype,NQ,NG) :-
    NQ<NG,
    choice(Q,Ctype,NQ).
qlist(Q,_,NQ,NG) :-
    NQ=NG,
    callprint(Q).

```

```

choice(Q,"0",N) :- % способ_выбора: СЛУЧАЙНО_БЕЗ_ПОВТОРОВ - 0
    time(_,_,S,_),
    I=S/10,
    randominit(I),
    assert(list(Q)),
    ql_random(N),
    retractall(list(_)),
    findall(TName,qout(TName),ST),
    retractall(qout(_)),
    callprint(ST).

```

```

choice(Q,"1",N) :- % СЛУЧАЙНО - 1
    time(_,_,S,_),
    I=S/10,
    randominit(I),
    assert(list(Q)),
    ql_onrand(N),
    retractall(list(_)),
    findall(TName,qout(TName),ST),
    retractall(qout(_)),
    callprint(ST).

```

```
choice(Q,"2",_) :- % ПОЛЬЗОВАТЕЛЬ - 2
    callprint(Q).
```

```
choice(Q,"3",N) :- % ПОДРЯД - 3
    findall(Prop,order(Prop,_),F),
    callsort(F,Q,S),
    asslist(S,N),!.
```

```
choice(____) :-
    not(qout(_)),
    write(1),nl,
    write("Unknown choice method"),nl.
```

```
delete(____).
delete(X,[X|Y],Y):- !.
delete(X,[A|Y],[A|Z]):- delete(X,Y,Z).
```

```
delfromlist(____).
delfromlist(L,____,L).
delfromlist(S,[H|T],L,K) :-
    delete(H,S,L),
    delfromlist(L,T,____,K).
```

```
memnum(____).
memnum(0,[H|____],H).
memnum(Num,____T,R) :-
    N=Num-1,
    memnum(N,T,R).
```

```
move(QName,Res) :-
    group(PName),
    thema(TName),
    question(TName,QName,PName,Gr),
    findall(Qn,question(TName,Qn,Pname,Gr),Res),
    printlist(Res).
```

```
ql_random(0).
ql_random(N) :-
    list(L),
    lenlist(L,Len),
    random(Len,Num),
    memnum(Num,L,Name),
    move(Name,Res),
    delfromlist(L,Res,____,Q),
    retractall(list(____)),
    assert(list(Q)),
    M=N-1,
    ql_random(M),!.
```

```
ql_onrand(0).
ql_onrand(N) :-
    list(L),
```



```

lenlist(L,Len),
random(Len,Num),
memnum(Num,L,Name),
move(Name,_),
M=N-1,
ql_onrand(M),!.

printlist([]).
printlist([H|T]) :-
    assert(qout(H)),
    printlist(T).

asslist([],_).
asslist([H|T],N) :-
    N>0,
    assert(qout(H)),
    M=N-1,
    asslist(T,M).
asslist(_,_) :- !.

% сортировка списка вопросов
sort(_,[],[]).
sort(P,F,S) :- % по возрастанию
    order(P,"ASC"),
    insert_sort(P,F,S,0).
sort(P,F,S) :- % по убыванию
    order(P,"DESC"),
    insert_sort(P,F,S,1).

insert_sort(_,[],[],_).
insert_sort(P,[X|T],Slist,R) :-
    insert_sort(P,T,Stail,R),
    insert(P,X,Stail,Slist,R).
insert(P,X,[Y|Slist],[Y|Slist1],0) :-
    acs(P,X,Y),!,
    insert(P,X,Slist,Slist1,0).
insert(P,X,[Y|Slist],[Y|Slist1],1) :-
    decs(P,X,Y),!,
    insert(P,X,Slist,Slist1,1).
insert(_,X,Slist,[X|Slist],_).

acs(P,X,Y) :-
    question(_,X,P,VX),
    question(_,Y,P,VY),
    VX>VY.

decs(P,X,Y) :-
    question(_,X,P,VX),
    question(_,Y,P,VY),
    VX<VY.

callsort([],Q,Q).
callsort([H|T],Q,L) :-

```

```

sort(H,Q,V),
callsort(T,V,L).

checknum(Len,0,Len).
checknum(_,N,N).

getinfo(Thema,Ctype,Len,Choice,NumQ,NumA) :-
    control(Thema,Ctype,Choice,_,NQ,NA,_,_,_),
    str_int(NQ,N1),
    checknum(Len,N1,NumQ),
    str_int(NA,N2),
    checknum(NumQ,N2,NumA).

head([],_,[""]).
head([H|T],H,T).
head([_T],N,S) :- head(T,N,S).

prlist([]).
prlist([H|T]) :- write(H), nl, prlist(T).

% адаптивный опрос
anextquery("",Next) :-
    qout(Next),!.
anextquery(Qname,Next) :-
    mark(Ball),
    findall(Name,qout(Name),F),
    head(F,Qname,S),
    group(Pname),
    question(_,Qname,Pname,V),
    nextqout(S,V,Ball,Next).
nextqout([],_,_, "").
nextqout([H|_],P,Mark,H) :-
    answer(_,H,_,_,_,Max),
    str_real(Max,Ball),
    Mark<Ball,
    group(Pname),
    question(_,H,Pname,P).
nextqout(L,P,_,N) :- nextgroup(L,P,N).
nextgroup([],_, "").
nextgroup([H|_],P,H) :-
    group(Pname),
    question(_,H,Pname,Val),
    Val<>P.
nextgroup([H|T],P,N) :-
    group(Pname),
    question(_,H,Pname,Val),
    Val=P,
    nextgroup(T,P,N).

```

Приложение 2. Вариант программной реализации КС-грамматики для разбора описания теста

В качестве варианта реализации рассмотрим систему, которая воспринимает описание грамматики в текстовом файле и на основании этой грамматики проводит синтаксически управляемый разбор описания системы контроля знаний.

Грамматика включает в себя правила подстановки (редукции), условия применимости этих правил и названия функций, которые должны быть выполнены при срабатывании правил. Правила состоят из категорий, ключевых слов и специальных символов (разделителей и ограничителей, требующихся в соответствии с синтаксисом). Например, строка

<КОМАНДА> ::= ТЕМА <EXPR> ?';!<not> :push.P10

означает, что последовательность ТЕМА <EXPR> заменяется на категорию <КОМАНДА>, если следующий за выражением <EXPR> символ не является точкой с запятой ('!<not>), и при этом выполняется функция P10. Условия записываются в польской форме и накладываются на следующую за текущей лексическую единицу. Если функция должна быть выполнена независимо от истинности условия, то вместо двоеточия (:) ставится восклицательный знак (!).

Функции предназначены для выполнения действий по формированию структур данных на основании анализируемого описания теста или носят вспомогательный характер и выполняют действия по локализации редукции.

Ниже приведено содержимое текстового файла с описанием грамматики. Все символы и идентификаторы (кроме ключевых слов) должны быть перечислены в правой части правил подстановки. Для этого вводится произвольная категория (в данном случае – <QWERTY>). Строки, начинающиеся с символа ';', являются комментариями.

```

; .....
<QWERTY> ::= ; , # p10 p13 p14 p21 p23 p25 p26 p30
<QWERTY> ::= P32 P36 p37 p38 p40 p41 p100 p201
<QWERTY> ::= .._..... .._..... .._..... .._.....
<QWERTY> ::= .._..... .._..... .._..... .._.....
<QWERTY> ::= .._..... .._..... .._..... .._.....
<QWERTY> ::= .._..... .._..... .._..... .._.....
<QWERTY> ::= .._..... .._..... .._..... .._.....
<QWERTY> ::= .._..... .._..... .._..... .._.....
<QWERTY> ::= .._..... .._..... .._..... .._.....
<QWERTY> ::= .._..... .._..... .._..... .._.....
;-----
<.....> ::= <.....> <.....>
<.....> ::= <.....>

<.....> ::= <.....> ; :push.._.....
<.....> ::= <.....> ; :push.._.....
<.....> ::= <.....> ; :push.._.....
<.....> ::= <.....> ; :push.._.....

<.....> ::= ..... <EXPR> ; :push.._.....
<.....> ::= <....._.....> # ; :push.._.....
<....._.....> ::= <EXPR> ?'#' :push.....
<....._.....> ::= <.....> ?'#' :push.....

<....._.....> ::= <....._.....> <....._.....> .....
<....._.....> ::= <....._.....> ..... :push.P41
<....._.....> ::= ..... :push.....
<....._.....> ::= ..... <.....> :push.....
;-----<.....>:
<.....> ::= <.....QQQ> <EXPR> <....._.....> ..... :push.P14
<.....QQQ> ::= ..... :push.....

```

```

;-----<.....>:
<.....> ::= <.....QQQ> <EXPR> <....._.....> ..... :push.P25
<.....QQQ> ::= ..... :push....._.....

;-----<.....>
<.....> ::= <...._.....> <....._.....> ..... :push.P32
<...._.....> ::= ....._.._..... <EXPR> :push....._.....
<....._.....> ::= <....._.....> <.....>
<....._.....> ::= <.....>

<.....> ::= ..... <EXPR> ?';'.<not> :push.P10

<.....> ::= ..... :push.3.push....._.....
<.....> ::= ..... :push.1.push....._.....

<.....> ::= ..... :push.0.push....._.....
<.....> ::= ..... :push.1.push....._.....
<.....> ::= ..... :push.2.push....._.....
<.....> ::= ..... :push.0.push....._.....
<.....> ::= ..... :push.1.push....._.....
<.....> ::= ..... :push.2.push....._.....
<.....> ::= ..... <EXPR> :push.0.push....._.....
<.....> ::= ..... <EXPR> :push.1.push....._.....
<.....> ::= ..... <EXPR> : <EXPR> :push.P13
<.....> ::= ..... <EXPR> : <EXPR> :push.P23
<.....> ::= ..... <EXPR> :push.P21
<.....> ::= ..... <.....> :push.P26
<.....> ::= ..... <EXPR> :push.0.push.....
<.....> ::= ..... <EXPR> :push.1.push.....
<.....> ::= ....._....._....._.....
:push.0.push....._.....
<.....> ::= ....._..... :push.1.push....._.....
<.....> ::= ....._..... :push.2.push....._.....
<.....> ::= ....._..... :push.0.push....._.....
<.....> ::= ....._..... :push.1.push....._.....
<.....> ::= ....._..... <EXPR> :push.P36
<.....> ::= ....._..... <EXPR> :push.P37
<.....> ::= ..... <EXPR> :push.P38
<.....> ::= ....._..... :push.0.push....._.....
<.....> ::= ....._..... :push.1.push....._.....
<.....> ::= ....._..... <EXPR> : <EXPR> :push.2.push....._.....
<.....> ::= ..... <.....> :push.P30
<.....> ::= .....

;-----<.....>:
<.....> ::= <...._.....> <....._.....> <...._.....> :push.P201
<.....> ::= <...._.....> <....._.....> <...._.....> :push.P201
;.....
<.....> ::= <...._.....> <....._.....>
<.....> ::= <...._.....> <....._.....>
<....._.....> ::= <....._.....> , <....._.....>
<....._.....> ::= <....._.....>
<....._.....> ::= <.....>
<....._.....> ::= <EXPR> ?','.}'.'.'].<or>.<or>

<...._.....> ::= { :push....._.....
<...._.....> ::= [ :push....._.....
<...._.....> ::= } :push....._.....
<...._.....> ::= ] :push....._.....
;-----
<EXPR> ::= <E> ?'+'.<not>.'-'.<not>.'*'.<not>.'/'.'<not>.')'
.<not>.<and>.<and>.<and>.<and> :push.P100
;.....
<E> ::= <E> - <T> ?'*'.<not>.'/'.'<not>.<and> :push.-
<E> ::= <E> + <T> ?'*'.<not>.'/'.'<not>.<and> :push.+
<E> ::= <T> ?'*'.<not>.'/'.'<not>.<and>
<T> ::= <T> / <F> :push./
<T> ::= <T> * <F> :push.*
<T> ::= <F>
<F> ::= <i> ( <E> ) :push.CALL
<F> ::= <i> ?'='.'<not>.'('.'<not>.<and> !push.I

```

<F> ::= (<E>)

Процедуры, используемые в данной грамматике, перечислены в табл. 1.

Таблица 1. Процедуры, используемые при разборе описания теста

Название процедуры	Назначение процедуры
НАЧАТЬ_МНОЖЕСТВО	Добавление в список нового множества, подсчет уровней вложенности списков и множеств.
НАЧАТЬ_СПИСОК	Добавление в список нового списка, подсчет уровней вложенности списков и множеств.
СПИСОК_ГОТОВ	Уменьшение количества уровней вложенности списков и множеств на 1.
ФР_ВОПРОС	Установка флага окончания разбора блока (фразы) и завершение формирования структуры данных ВОПРОС.
ФР_ОТВЕТ	Установка флага окончания разбора блока (фразы) и завершение формирования структуры данных ОТВЕТ.
ФР_КОНТРОЛЬ	Установка флага окончания разбора блока (фразы) и завершение формирования структуры данных БЛОК КОНТРОЛЯ.
ФР_СПИСТЕМ	Установка флага окончания разбора блока (фразы) и завершение формирования структуры данных СПИСОК ТЕМ.
ФР ТЕМА	Установка названия текущей темы.
P10	Установка названия темы для текущего вопроса или для разбора в целом.
ТИП_ТЕКСТ_КОМАНДА	Установка типа текущего вопроса (ТЕКСТ, КОМАНДА).
СОДЕРЖАНИЕ_ВОПРОСА	Запоминание содержимого текущего вопроса.
P13	Добавление свойства для текущего вопроса.
P14	Запоминание имени (названия) текущего вопроса.
ТИП_ОТВЕТА	Установка типа текущего ответа (МНОЖЕСТВО, СПИСОК, ВЫРАЖЕНИЕ).
P21	Запоминание эталонного варианта для текущего ответа (эталон не является списком или множеством).
ОЦЕНКА	Запоминание типа определения оценки и степени погрешности для текущего ответа.
P23	Запоминание минимального и максимального баллов за текущий ответ.
P25	Связывание соответствующего вопроса с текущим ответом.
P26	Запоминание эталонного варианта для текущего ответа, если эталон является списком или множеством.
P30	Формирование списка вопросов для блока КОНТРОЛЬ.

Таблица 1. Процедуры, используемые при разборе описания теста (продолжение)

Название процедуры	Назначение процедуры
P32	Завершение формирования блока КОНТРОЛЬ.
ТИП_КОНТРОЛЯ	Запись типа контроля для блока КОНТРОЛЬ.
СПОСОБ_ВЫБОРА	Установка способа выбора вопроса для текущего блока КОНТРОЛЬ.
СПОСОБ_ВЫДАЧИ	Установка способа выдачи вопроса для текущего блока КОНТРОЛЬ или для текущего списка тем.
ТИП_ОЦЕНКИ	Установка типа подсчета оценки (СУММА,

	ПРОЦЕНТ) для текущего блока КОНТРОЛЬ.
P36	Установка количества вопросов для текущего блока КОНТРОЛЬ.
P37	Установка количества ответов для текущего блока КОНТРОЛЬ.
P38	Установка минимальной оценки, при которой тема считается сданной, для текущего блока КОНТРОЛЬ.
ТИП_ОЦЕНКИ_ШКАЛА	Установка типа подсчета оценки (ШКАЛА) для текущего блока КОНТРОЛЬ.
P40	Формирование списка тем для блока СПИСОК_ТЕМ
P41	Установка способа выдачи вопросов по темам (ОДИН).
P100	Переход к следующему выражению с занесением полученного выражения в список.
+	Запоминание знака '+' в массиве выражения.
-	Запоминание знака '-' в массиве выражения.
*	Запоминание знака '*' в массиве выражения.
/	Запоминание знака '/' в массиве выражения.
CALL	Запоминание имени в массиве выражения.
P201	Завершение формирования списка (множества).
ФР_ОТВЕТ_ПОЛЬЗОВАТЕЛЯ	Пустая функция.
ОППВЫРАЖ	Возвращение ответа пользователя как выражения.
ОППСПИСО	Возвращение ответа пользователя как списка.

На основании описания грамматики и файла-шаблона утилита IGEN создает файл, в который записываются фрагменты программы на языке C, реализующие данную грамматику. Все встреченные в описании ключевые слова транскрибируются в латинский алфавит, описываются как DEFINE и получают числовые идентификаторы. Имена функций, указанных в грамматике, также транскрибируются в латинский алфавит. После добавления в этот файл данных функций получается программа, реализующая данную грамматику.

Вызов утилиты IGEN осуществляется так:

IGEN <имя файла грамматики> <имя файла шаблона> <имя выходного файла>

Приложение 3. Параметрическая настройка системы контроля знаний

Параметрическая настройка может осуществляться через установки в файле конфигурации (TS.CFG) или путем использования встроенных параметров при описании тестов.

Файл конфигурации

Ниже приведен пример файла конфигурации:

```
; Здесь указывается путь к базе данных (каталог, где лежат dbf-файлы)
dbpath = d:\do\database
; Здесь указывается имя программы, которая управляет тестированием
qform = kbp.bat
; Имя файла с результатами работы программы, указанной в параметре qform
qres = %TS%\swap\kbp.out
```

Перечень встроенных параметров и переменных для СКЗ

Для вопросов и ответов можно использовать в качестве СВОЙСТВ следующие встроенные параметры:

1. qtip – способ выполнения процесса (system, process).
2. cmd – команда (имя внешнего процесса).
3. exportfile – имя файла экспорта, формируемого системой для внешнего процесса.
4. exportfile1 – имя 2-го файла, формируемого системой.
5. exportfile2 – имя 3-го файла, формируемого системой.
6. export – содержимое файла exportfile.
7. export1 – содержимое файла exportfile.
8. export2 – содержимое файла exportfile.
9. file – имя файла, содержащего ответ от внешнего процесса.
10. importfile – имя файла, содержащего ответ от внешнего процесса (см. file).

При описании содержимого СВОЙСТВ можно использовать следующие встроенные переменные:

1. %TS% – родительский каталог системы TS (tsDir).
2. %uname%" – имя пользователя (username).
3. %uid% – идентификатор пользователя (uid).
4. %sid% – текущее время (current time).
5. %exportfile% – имя файла, формируемого системой для внешнего процесса.
6. %exportfile1% – имя 2-го файла, формируемого системой для внешнего процесса.
7. %exportfile2% – имя 3-го файла, формируемого системой для внешнего процесса.
8. %importfile% – имя файла, содержащего ответ от внешнего процесса (см. file).
9. %userans% – ответ пользователя.
10. %tmp% – имя временного файла для обмена данными.
11. %tdl% – имя tdl-файла с описанием теста.
12. %tsk% – имя tsk-файла с описанием задания на тестирование.

Приложение 4. Перечень сокращений, встречающихся в тексте

АОС	– автоматизированная обучающая система;
АУК	– автоматизированный учебный курс;
БД	– база данных;
БЗ	– база знаний;
БПО	– базовое программное обеспечение;
ВТ	– вычислительная техника;
ИС	– инструментальная система;
КОП	– компьютерная обучающая программа;
КСУН	– компьютерные средства учебного назначения;
ЛВС	– локальная вычислительная сеть;
ЛКПО	– локальное коммуникационное программное обеспечение;
ЛППО	– локальное прикладное программное обеспечение;
МО	– модель обучаемого;
МПО	– модель предметной области;
ПО	– предметная область;
ППО	– прикладное программное обеспечение;
РАОС	– распределенная автоматизированная обучающая система;
СКЗ	– система контроля знаний;
СУБД	– система управления базой данных.