

Федеральное государственное бюджетное образовательное учреждение
высшего образования
МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

На правах рукописи



Алексейчук Андрей Сергеевич

**МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ СИСТЕМЫ ДИСТАНЦИОННОГО
ОБУЧЕНИЯ НА БАЗЕ ВЕБ-КОНФЕРЕНЦИЙ**

Специальность 05.13.11

Математическое и программное обеспечение вычислительных машин,
комплексов и компьютерных сетей

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:
доктор физико-математических наук, профессор
Пантелеев Андрей Владимирович

Москва – 2017

Оглавление

Введение	5
Глава 1. Требования к программной системе и выбор технологий для её реализации	16
1.1. Обоснование необходимости разработки программной системы	16
1.2. Требования к программной системе и к ее компонентам.....	19
1.2.1. Функциональные требования к СДО	20
1.2.2. Нефункциональные требования к СДО	24
1.2.3. Общая архитектура программной системы.....	26
1.2.4. Требования к компонентам программной системы.....	28
1.2.4.1. Требования к клиентскому веб-приложению для проведения веб-конференций	28
1.2.4.2. Требования к приложению-редактору учебных заданий.....	31
1.2.4.3. Требования к мультимедийному серверу	32
1.2.4.4. Требования к серверному веб-приложению	34
1.2.4.5. Требования к СУБД	37
1.3. Выводы	38
Глава 2. Индивидуализация процесса обучения при помощи экспертной системы	40
2.1. Требования к процессу обучения в онлайн-системах СДО.....	40
2.2. Постановка задачи формирования индивидуальной траектории обучения.....	42
2.3. Экспертная система и её структура	46
2.3.1. Фаззификация входных данных	50
2.3.2. Блок фаззификации начального уровня подготовки студента	52
2.3.3. Определение параметров нейронной сети.....	54
2.3.3.1. Выбор количества итераций (эпох) обучения	55
2.3.3.2. Выбор архитектуры сети	58
2.3.3.3. Выбор крутизны функции активации нейронов скрытых слоёв	59
2.3.3.4. Выбор алгоритма обучения	60
2.3.4. Блоки нечеткого вывода экспертной системы	65
2.4. Этапы нечеткого вывода в иерархической нечеткой экспертной системе	71
2.5. Модель оценивания результатов решения учебных заданий	74
2.6. Моделирование индивидуализации процесса обучения с использованием экспертной системы	77
2.6.1. Условия моделирования	77
2.6.2. Результаты моделирования процесса обучения.....	78

Выводы	80
Глава 3. Технологии разработки, структура и принципы функционирования программного комплекса.....	82
3.1. Структура программного комплекса	82
3.2. Технологии реализации компонентов программного комплекса	83
3.2.1. Серверное веб-приложение.....	83
3.2.2. Клиентское приложение для проведения веб-конференций и редактор учебных заданий.....	86
3.2.3. Мультимедийный сервер.....	88
3.3. Структура базы данных.....	91
3.4. Протоколы и форматы обмена данными между компонентами программного комплекса.....	92
3.4.1. Протоколы передачи данных	92
3.4.2. Форматы хранения и передачи данных	93
3.5. Взаимодействие пользователей в режиме реального времени и его реализация в клиентском веб-приложении.....	97
3.6. Экспертная система и ее реализация	99
3.6.1. Структура классов экспертной системы.....	99
3.6.2. Подготовка исходных данных для экспертной системы.....	101
3.6.3. Осуществление вычислений экспертной системой	102
3.7. Выводы	103
Глава 4. Интерфейс и порядок работы с программным комплексом.....	104
4.1. Общее описание программного комплекса.....	104
4.2. Распределение ролей пользователей.....	104
4.2.1. Функции роли преподавателя	104
4.2.2. Функции роли студента	105
4.2.3. Функции роли администратора	105
4.3. Веб-интерфейс программного комплекса.....	106
4.3.1. Веб-интерфейс преподавателя.....	106
4.3.2. Веб-интерфейс администратора	113
4.3.3. Веб-интерфейс студента.....	113
4.4. Работа с клиентским приложением для проведения веб-конференций.....	114
4.4.1. Окно клиентского приложения.....	114
4.4.2. Демонстрирование презентаций.....	115
4.4.3. Работа с виртуальной доской.....	116

4.4.4. Демонстрирование рабочего стола.....	117
4.4.5. Обмен файлами с участниками.....	118
4.4.6. Функция пошагового решения учебных заданий	119
4.7. Работа с редактором учебных заданий	122
4.8. Практическое использование результатов исследования	124
Выводы	125
Заключение.....	126
Список литературы.....	128

Введение

Актуальность исследования. Тенденции развития современных систем высшего образования неразрывно связаны с широким внедрением в учебный процесс различных методов, форм и средств активного обучения. Использование мультимедийных технологий открывает новые возможности в организации учебного процесса и развитии творческих способностей обучающихся. В настоящее время мультимедийные технологии - одно из наиболее бурно развивающихся направлений в области информационных технологий в учебном процессе благодаря произошедшей в последние годы революции в области услуг по предоставлению высокоскоростного доступа в Интернет и быстро растущей мощности персональных компьютеров. Большинству студентов доступно недорогое и достаточно быстрое соединение с Интернетом. Это позволяет обеспечивать не только асинхронную форму взаимодействия в процессе дистанционного обучения (работу с электронной почтой, отправку файлов с работами, поиск информации в базах знаний и т.п.), но и передавать значительные объёмы мультимедийных данных в режиме реального времени. Рационально используя эту возможность, можно поднять качественное наполнение процесса дистанционного обучения на новый уровень. Применение мультимедийных технологий существенно обогащает образовательный процесс, позволяет легко визуализировать материал, задействовать различные каналы восприятия информации, делает обучение эффективным и увлекательным.

Одним из инновационных видов дистанционного обучения, активно использующим средства мультимедиа, является веб-конференция. Эта технология позволяет воссоздавать условия коллективной формы организации обучения посредством обмена аудио- и видеоданными в виртуальной аудитории в режиме реального времени. Кроме того, веб-конференции создают привлекательную, динамичную и доступную для всех среду обучения, что приносит пользу как студентам, так и преподавателям. Благодаря распространению высокоскоростного домашнего соединения с Интернетом открылась возможность создать новое применение веб-конференциям, заключающееся в проведении регулярных учебных занятий, воссоздающих модель реальных аудиторных занятий. Транслирование потоков данных через Интернет по стандартным протоколам позволяет участвовать в таких мероприятиях при наличии минимального количества недорогой аппаратуры – веб-камеры, аудио-гарнитуры и персонального компьютера.

Современные мультимедийные технологии позволяют расширить возможности веб-конференций путем использования дополнительных функций, к которым относятся показ презентаций, работа с виртуальной доской, обмен файлами, тестирование и т.д. Эти

функции, дополненные средствами работы с формулами, рисунками и т. п., расширяют возможности подачи учебного материала преподавателем при организации онлайн-обучения по различным дисциплинам.

Однако проведение веб-конференций с большими группами студентов связано с определенными сложностями. При проведении практических занятий в режиме веб-конференции возникает проблема, связанная с разными уровнями подготовки присутствующих на занятии студентов [15]. Более подготовленные студенты, которые справляются с решением задач быстрее, вынуждены ждать, пока все остальные выполнят задание или решат поставленную задачу. Разумное решение этой проблемы состоит в объединении студентов в группы в зависимости от их уровня подготовки и от сложности решаемых на занятии задач и проведении отдельных занятий для каждой группы.

Поскольку преподавателю затруднительно точно оценить способности и подготовленность каждого студента до проведения занятий, то для формирования групп можно использовать специализированную информационную систему, целями которой являются:

- анализ оценок, полученных студентами ранее (если таковые имеются);
- анализ дополнительной информации об успеваемости студентов (оценки за ЕГЭ и т.д.);
- разбиение студентов на группы по результатам этого анализа;
- составление расписания проведения занятий.

Тема использования веб-конференций в образовательном процессе в отечественной и зарубежной научной литературе остается слабо изученной. Существующие исследования в основном посвящены аспектам применения уже существующих, ранее созданных систем видео- и веб-конференций в учебном процессе. Недостаточно внимания уделено технологиям разработки новых образовательных систем, предназначенных для работы в режиме реального времени.

В целом, современные исследования в области онлайн-обучения показывают, что взаимодействие и коммуникация между его участниками – важнейшие факторы в достижении желаемых результатов обучения и удовлетворённости студентов процессом обучения, предоставляемые веб-конференциями [43, 108]. Отмечается важнейшее качество, отличающее онлайн (синхронные) технологии обучения от обычных (асинхронных) – обеспечение учебного диалога [56]. Но синхронные системы не могут целиком заменить асинхронные, поскольку им присущи недостатки, важнейший из которых – невозможность каждому студенту заниматься в удобное для себя время.

Поэтому предлагается сочетать оба режима коммуникации путём их интегрирования в единую среду [44].

В [43] приводится система принципов, отражающих основные требования к организации образовательного процесса при использовании информационно-коммуникационных технологий: принцип активности и самостоятельности обучающихся, принцип авторского участия в учебном процессе, принцип интерактивности, принцип мультимедийного представления учебной информации и др. Отмечается, что в этом случае занятиям становятся присущи такие свойства как мотивированность, проблемность, наглядность, эмоциональность, высокая активность.

В работе [72] отмечается, что веб-конференции при наличии расширенных средств совместной деятельности могут быть использованы для подготовки студентов наукоёмких специальностей, поскольку могут обеспечивать не только коммуникации вида «преподаватель – студент», но и «студент – профессионально-ориентированное сообщество», «студент – мировое информационное пространство» и «студент – инновационные инфраструктуры».

В настоящем исследовании для построения учебного процесса используется экспертная система, в основу принципа работы которой положена модель нечеткого вывода. Нечеткий логический вывод – это модель обработки информации, позволяющая использовать формулировки правил, задающих знания экспертов на естественном языке в форме «Если ..., то ...» и таким образом строить модели поведения сложных систем с высоким уровнем абстракции. В отечественных и зарубежных изданиях описано множество примеров работы нечетких систем, в том числе иерархических. Выпускаются специализированные журналы, посвященные нечетким системам: «IEEE Transactions on Fuzzy Systems», «Fuzzy Sets and Systems». Примеры успешного применения нечетких систем можно встретить во множестве различных прикладных областей: технических, экологических, медицинских и т.д. Примеры включают моделирование процессов, происходящих на сталелитейном заводе [116], моделирование процесса горения [112], моделирование потоков воды в реках [113], моделирование гашения колебаний пружинного подвеса [84]. Нечеткие модели используются для создания автоматических систем управления, например, системы управления беспилотным летательным аппаратом [46], химическим заводом [111]. Системы с нечеткой логикой могут быть использованы для задач финансового анализа [35], управления рисками [29]. Также они могут использоваться как системы поддержки принятия решений для операторов машин и механизмов [88].

В [93] и [122] подробно рассматриваются иерархические нечеткие системы. рассматривается вопрос чувствительности иерархических систем к малым возмущениям входных данных. Доказывается, что такие системы являются универсальными аппроксиматорами.

Таким образом, системы нечеткого вывода могут использоваться для моделирования объектов, которым присущи неточность и неформализованность, где точные методы не всегда дают адекватных методов исследования. Отсюда естественным образом следует обоснованность их применения в области дистанционного обучения, где учебный процесс является сложным явлением, зависящим от множества параметров и условий. В научной литературе описан ряд применений нечетких систем в сфере высшего образования. В статье [110] предлагается использовать иерархическую нечёткую базу знаний для вычисления общего рейтинга студента, учитывающего его академическую успеваемость, общественную деятельность, научно-исследовательскую деятельность, участие в конкурсах и т.д. В [89] рассмотрено проектирование иерархической нечеткой системы для управления процессом планирования набора студентов в вуз, приведена модель соответствующей системы поддержки принятия решений.

Применение информационных систем на основе методов нечеткой логики и нейронных сетей в дистанционном обучении рассматривается в ряде зарубежных исследований. Выпускаются специализированные издания, посвящённые этой теме, в частности, журнал «International Journal of Artificial Intelligence in Education» («Международный журнал об искусственном интеллекте в образовании»). Приведем краткий обзор зарубежных исследований по данной теме.

В работе [91] предлагается использовать систему иерархического нечеткого вывода для вычисления оценки на основе информации о фактической оценке за задание, затраченном времени и экспертных оценках сложности и важности данного задания. Предлагаемая система состоит из трех блоков нечеткого вывода, первый из которых вычисляет сложность на основе фактической оценки и затраченного времени, второй – вычисляет количество затраченных усилий, третий – определяет итоговую оценку с учетом важности задания. Для фаззификации используются треугольные функции принадлежности.

Усовершенствованная версия данной системы, использующая гауссовские функции принадлежности, предложена в работе [90]. Утверждается, что такой подход позволяет получить более точные значения за счет того, что функции принадлежности не обращаются в нуль, за счет чего при фаззификации учитываются все термы нечетких переменных, что позволяет более точно учитывать экспертную информацию.

В статье [110] предложена нейронечеткая модель определения индивидуальных характеристик каждого студента. Функции принадлежности в такой системе моделируются определенной архитектурой нейронной сети со специально подобранными функциями активации нейронов. Операции нечеткого вывода и дефаззификации также моделируются нейронными сетями. Такой комбинированный подход позволяет использовать готовые библиотеки для работы с нейросетями для моделирования нечеткого вывода.

В [89] проведено сравнение нескольких методов предсказания оценки за финальный экзамен на основе данных об успеваемости студентов в течение семестра: многослойный перцептрон, сеть радиально-базисных функций, метод опорных векторов, метод линейной регрессии. Показано, что многослойный перцептрон дает наиболее точные предсказания.

В [102] предложено использовать нейросеть для предсказания общей успеваемости студента, учитывая различные факторы: успеваемость по отдельным предметам, возраст, пол, уровень образования родителей, местоположение школы и т.д.

Работы [77, 92, 95, 98, 108] посвящены исследованию применения веб-конференций в контексте качества дистанционного обучения, его эффективности, заинтересованности студентов в участии в учебном процессе. В [98] предлагается гибридный подход к обучению, сочетающий очные занятия и занятия в форме веб-конференции. Однако и отечественные, и зарубежные исследования посвящены в основном техническим и методическим аспектам внедрения веб-конференций в вузах, недостаточно внимания уделено математическому моделированию учебного процесса, моделированию знаний студентов, формированию обратной связи в ходе обучения.

В статье [110] приводится множество примеров применения методов нечеткой логики для построения модели текущих знаний студента и индивидуализации процесса обучения в веб-ориентированных СДО, не относящихся к системам реального времени.

Описаний систем управления учебным процессом в СДО реального времени в отечественной и зарубежной научной литературе не приводилось, также нет примеров применения методов нечеткой логики и нейросетевых моделей в сфере обучения в режиме веб-конференций. Поэтому разработка программного комплекса поддержки дистанционного обучения в режиме веб-конференций, представляющей широкий набор возможностей изучения различных дисциплин в режиме реального времени и обладающей иерархической нечеткой экспертной системой, позволяющей выстроить учебный процесс с учетом индивидуальных способностей каждого студента является **актуальной задачей** сегодняшнего дня.

Область исследования соответствует пунктам паспорта специальности специальности 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»:

- модели, методы, алгоритмы, языки и программные инструменты для организации взаимодействия программ и программных систем;

- человеко-машинные интерфейсы; модели, методы, алгоритмы и программные средства машинной графики, визуализации, обработки изображений, систем виртуальной реальности, мультимедийного общения.

Научная новизна работы заключается в том, что в данной работе разработан программный комплекс поддержки дистанционного обучения в режиме веб-конференции, предоставляющий возможности для изучения различных дисциплин в режиме реального времени и обладающий экспертной системой, позволяющей выстроить учебный процесс с учётом индивидуальных способностей каждого студента.

Научная новизна состоит в следующих положениях:

1. Создан программный комплекс, реализующий общение в режиме веб-конференции и обладающий средствами и инструментами для проведения занятий и проверки знаний по различным дисциплинам. Разработан принцип взаимодействия участников веб-конференции при проведении занятия в режиме реального времени, включая одновременное пошаговое решение учебных заданий под контролем преподавателя, и созданы соответствующие элементы интерфейса.

2. Предложен метод организации учебного процесса путем создания различных уровней сложности заданий для одного и того же занятия, а также проведения отдельных занятий с группами студентов, сформированных в соответствии со сложностью наборов заданий, включенных в каждый уровень. Разработана структура и математическая основа принципа работы нечеткой экспертной системы, позволяющей индивидуализировать учебный процесс путем построения индивидуальной траектории обучения для каждого студента. Данная экспертная система анализирует исходные данные о каждом студенте и историю полученных им оценок и автоматически рассчитывает рекомендуемый уровень сложности каждого занятия для каждого студента. Данный алгоритм позволяет автоматически составлять расписание занятий, формируя группы студентов с близким уровнем подготовки.

3. Разработана структура и принципы взаимодействия компонентов информационной системы: клиентского веб-приложения для проведения веб-конференций, мультимедийного сервера, серверного веб-приложения с экспертной системой, редактора учебных заданий, базы данных, веб-сервера; созданы принципы хранения, обработки и

представления материалов учебных заданий, предназначенных для решения в режиме реального времени под контролем преподавателя.

Цель диссертационной работы состоит в создании средств дистанционного обучения на базе технологий веб-конференции, поиске и реализации индивидуальных подходов к организации такого обучения в соответствии со способностями студентов.

Для достижения указанной цели необходимо решить следующие **задачи**:

- разработать форму дистанционного обучения в режиме веб-конференции с применением модели индивидуализации обучения каждого студента, использующей доступную информацию о студенте и о сложности предлагаемых заданий;
- разработать математическое обеспечение информационной системы, поддерживающей предложенную модель индивидуализации обучения;
- разработать программный комплекс, реализующий систему для проведения учебных веб-конференций, имеющий средства проверки знаний студентов и включающий в себя указанную информационную систему;
- реализовать удобный пользовательский интерфейс программного комплекса, позволяющий проводить учебные веб-конференции, управлять учебным процессом, создавать и редактировать учебные задания.

Достоверность результатов работы обеспечивается корректным использованием математического аппарата и подтверждается экспериментальной проверкой работы разработанной информационной системы, а также использованием материалов диссертации и разработанного комплекса программ в учебных заведениях.

Практическая ценность результатов работы состоит в том, что разработанный программный комплекс позволяет воссоздавать условия обучения различным дисциплинам в режиме реального времени посредством обмена аудио- и видеоданными и совместной работы в виртуальной аудитории. Благодаря этому становится возможным проводить занятия в онлайн-режиме при непосредственном участии преподавателя. Внедрение разработанной системы позволяет:

- проводить занятия со студентами, не имеющими возможности очно присутствовать в институте (студенты филиалов, инвалиды, студенты заочной формы обучения);
- проводить различные виды занятий: лекции, практические занятия, а также дополнительные занятия и консультации без необходимости присутствия участников в учебном заведении;

- организовывать эффективный учебный процесс благодаря наличию в составе комплекса экспертной системы, позволяющей индивидуализировать процесс обучения каждого студента.

Апробация результатов исследования. Основные результаты диссертационного исследования докладывались на следующих конференциях и семинарах:

- XI научно-практическая межвузовская конференция молодых ученых и студентов «Молодые ученые – нашей новой школе» (Москва, МГППУ, 2012 г.);
- Научно-техническая международная молодежная конференция «Системы, методы, техника и технологии обработки медиаконтента» (Москва, МГУП им. И. Федорова, 2011 г.);
- XVII, XVIII, XIX Международные конференции по вычислительной механике и современным прикладным программным системам (ВМСППС'2011, ВМСППС'2013, ВМСППС'2015, г. Алушта);
- Международная научно-методическая конференция «Информатизация инженерного образования» (ИНФОРИНО) (Москва, МЭИ, 2012 г.);
- VIII, IX научно-практические конференции молодых ученых и студентов «Инновационный менеджмент в аэрокосмической промышленности» (Москва, МАИ, 2012, 2013 гг.);
- XI научно-практическая конференция молодых ученых и студентов «Инновации в экономике и менеджменте в авиации» (Москва, МАИ, 2015 г.);
- XII научно-практическая конференция молодых ученых и студентов «Инновации в экономике и менеджменте в аэрокосмической промышленности» (Москва, МАИ, 2016 г.);
- II международная научно-практическая конференция «Психологическая помощь социально незащищённым лицам с использованием дистанционных технологий (Интернет-консультирование и дистанционное обучение)» (Москва, МГППУ, 2012 г.);
- 11-я Международная конференция «Авиация и космонавтика – 2012». (Москва, МАИ, 2012 г.);
- Международная научно-техническая конференция «Актуальные проблемы прикладной математики, информатики и механики» (Воронеж, ВГУ, 2016 г.);
- Научные семинары Института проблем информатики РАН (2016 г), кафедры «Математическая кибернетика» МАИ (НИУ) (2013-2017 г).

Публикации. По теме диссертации опубликована 21 научная работа, из них 3 работы в журналах, входящих в перечень ВАК (1 статья из журнала, входящего в БД Scopus, 1 статья в соавторстве), 17 статей в других журналах и сборниках материалов конференций (1 статья в соавторстве), 2 свидетельства о государственной регистрации программ для ЭВМ.

Основные положения и результаты, выносимые на защиту:

1. Метод индивидуализации траектории обучения студента при обучении в режиме веб-конференции и принцип построения индивидуальной траектории обучения посредством выбора рекомендуемого уровня сложности заданий для каждого студента перед проведением занятия.

2. Структура и математическое обеспечение принципа работы иерархической нечеткой экспертной системы, реализующей предложенную математическую модель индивидуализации на основе методов нечеткой логики и нейронных сетей. Данная экспертная система обрабатывает экспертную информацию об уровне сложности каждого занятия и данные об успеваемости студентов и автоматически составляет расписание занятий, в котором каждому студенту назначен рекомендуемый уровень сложности занятий.

3. Программный комплекс системы дистанционного обучения, объединяющий систему поддержки веб-конференций и разработанную экспертную систему и обеспечивающий дистанционное обучение в режиме реального времени; принципы создания, архитектура и требования к компонентам программного комплекса,.

4. Принципы работы с компонентами пользовательского интерфейса разработанного программного комплекса, предназначенными для проведения занятий в форме веб-конференции, управления учебным процессом, просмотра результатов обучения, создания учебных заданий, администрирования системы.

Структура и объем работы. Работа состоит из введения, четырех глав, заключения и списка литературы. Общий объем диссертации составляет 137 машинописных страниц. Текст диссертации содержит 51 рисунок и 8 таблиц. Список литературы содержит 123 наименования.

Во **введении** обоснована актуальность темы диссертации, приведен обзор работ и публикаций в области программного и математического обеспечения систем дистанционного обучения, сформулированы цели и задачи исследования.

В **первой главе** приведен обзор существующих технических решений в области веб-конференций, обоснована целесообразность разработки программного комплекса, реализующего обучение различным дисциплинам в режиме реального времени.

Представлены функциональные и нефункциональные требования к проектируемой программной системе дистанционного обучения и ко всем ее компонентам. Функциональные требования определяют задачи или действия, которые система способна выполнять в ходе своего функционирования, также к ним относятся системные требования к используемому аппаратному и программному обеспечению. Нефункциональные требования определяют интеграционные свойства системы, не связанные непосредственно с ее функциональностью: доступность, производительность, удобство поддержки, масштабируемость, надежность и т. д. В главе выделены функциональные и нефункциональные требования к системе в целом и к каждому ее компоненту в отдельности. Для реализации функциональных возможностей предложена клиент-серверная архитектура со следующим составом компонентов программной системы: клиентское веб-приложение, серверное веб-приложение, мультимедийный сервер, HTTP-сервер, СУБД, редактор учебных заданий.

В соответствии с результатами анализа требований произведен выбор технологий разработки каждого компонента программного комплекса. Для реализации клиентского веб-приложения и редактора учебных заданий выбрана технология Adobe Flash, для разработки мультимедийного сервера - сервер Red5, для разработки серверного веб-приложения и реализации экспертной системы выбран фреймворк Ruby on Rails, в качестве СУБД выбрана PostgreSQL, в качестве веб-сервера – Nginx.

Вторая глава посвящена разработке математического обеспечения процесса индивидуализации обучения каждого студента в рамках разработанной СДО путем выбора рекомендуемого уровня сложности занятий. Предложена модель процесса обучения, где для каждого занятия составляются задания нескольких уровней сложности, а студенту предлагается прохождение определенного уровня, в наибольшей степени соответствующего его текущему уровню подготовки по данной дисциплине. Для определения рекомендуемого уровня сложности разработана иерархическая нечеткая экспертная система, автоматически обрабатывающая данные о начальной успеваемости студента (оценка за ЕГЭ по дисциплине, средний балл за ЕГЭ, балл в аттестате по дисциплине и т. д.), о его текущей успеваемости (экспоненциально взвешенная средняя сумма оценок, полученных в ходе предыдущих занятий в рамках рассматриваемой СДО) и об экспертных оценках сложности каждого уровня, задаваемых преподавателем при их составлении. Экспертная система включает блоки фаззификации входных величин, один из которых реализован на базе нейронной сети, три блока нечеткого вывода, образующих иерархический блок, блоки композиции и дефаззификации. Приведена процедура выбора параметров нейронной сети (архитектура сети, количество нейронов, крутизна функции

активации, алгоритм и длительность обучения сети). Приведена модель расчета оценки, выставляемой каждому студенту по результатам решения учебных заданий. Проведено моделирование работы экспертной системы с экспериментальным набором входных данных, проиллюстрирована реакция системы на получение студентом различных оценок. Как и ожидается, при получении высоких оценок траектория обучения студента в целом стремится к более сложным уровням, а после получения низких оценок – наоборот, к более простым.

В третьей главе приведено описание разработанной архитектуры программного комплекса, описаны технологии разработки, форматы хранения данных и способы обмена информацией между его компонентами.

Клиентское веб-приложение и редактор учебных заданий разработаны на основе технологии Flash, реализующей необходимые технологии мультимедиа (растровую и векторную графику, звук, видео) и позволяющей создавать интерактивные элементы веб-приложений.

Мультимедийный сервер служит для приёма и передачи аудио- и видеопотоков от клиентских компьютеров и синхронизации событий при помощи общих объектов (shared objects). Каждое событие, подлежащее синхронизации в режиме реального времени, обрабатывается соответствующим общим объектом на мультимедийном сервере и рассылается всем участникам конференции.

В главе приводится структура данных, предназначенная для хранения учебных заданий, отображаемая на иерархическую структуру данных в формате XML. Параметры всех элементов (координаты, цвет и т.д.) задаются атрибутами соответствующих узлов XML-документа. Описана модель взаимодействия пользователей (студентов и преподавателя) при решении учебных заданий в режиме веб-конференции. Приведена архитектура клиентского веб-приложения для проведения конференций и серверного приложения с экспертной системой, логическая структура базы данных, описана структура классов экспертной системы.

В четвертой главе приведено описание интерфейса системы. Описан порядок работы студентов и преподавателей с веб-интерфейсом, с клиентским приложением и с редактором заданий. Приведены примеры работы с дополнительными мультимедийными функциями, такими как показ презентаций, демонстрация рабочего стола, работа с виртуальной доской. Описан порядок работы пользователей с функцией решения учебных заданий в роли студента и преподавателя, а также приведены возможности редактора учебных заданий и порядок работы с ним. Приведены примеры практического использования разработанного программного комплекса.

Глава 1. Требования к программной системе и выбор технологий для её реализации

1.1. Обоснование необходимости разработки программной системы

В настоящее время в вузах нашло применение большое количество систем дистанционного обучения (СДО), основанных на гипертекстовых технологиях, например, Moodle [100], TrainingWare Class [119], Доцент [64], WebTutor [63], Прометей [54] и другие. В МАИ используется СДО собственной разработки CLASS.NET [47, 51], JQT [65]. Перечисленные системы имеют веб-интерфейсы, основанные на HTML-технологии с включением графических и видеоматериалов, интерактивных элементов. Такие системы ориентированы на самостоятельную работу студента и не предусматривают непосредственного взаимодействия студента с преподавателем или студентов друг с другом в режиме реального времени. Поэтому органичным дополнением учебного процесса при дистанционном обучении являются системы веб-конференций, позволяющие всем участникам учебного процесса видеть и слышать друг друга и совместно работать с наглядными материалами, представленными с помощью технологий мультимедиа.

Подобные онлайн-системы находят всё более широкое применение. Для проведения занятий в формате веб-конференции в вузах чаще всего используют готовые программные решения. Среди них можно отметить следующие: Skype [109], TrueConf [120], Adobe Connect [76], Apache Openmeetings [103], Comdi [82], WizIQ [123], Веда [28]. Также существуют веб-сервисы для проведения конференций с веб-интерфейсом, не требующие установки какого-либо программного обеспечения: www.webinar.ru, www.v-class.ru, www.elearningpro.ru и другие.

Основные возможности перечисленных систем представлены в табл. 1.

Таблица 1. Сравнение возможностей систем веб-конференций

	Плата за пользование	Многопольз. режим	Чат	Показ экрана	Доска	Презентации	Обмен файлами	Тесты, опросы	Запись	Планирование занятий	Управление раб. столом
Skype	+	+	+	+	-	-	+	-	-	+	+
TrueConf	+	+	+	+	+	+	-	-	+	+	+
Adobe Connect	+	+	+	+	+	+	+	+	+	-	+
Openmeetings	+	+	+	-	+	-	+	+	+	+	+
Comdi	+	+	+	+	-	+	-	+	+	-	+
WizIQ	+	+	+	+	+	-	+	+	+	-	+
Веда	-	+	+	+	+	+	+	+	-	+	+

В основном перечисленные системы предназначены для решения бизнес-задач (проведение тренингов, семинаров, обучения персонала) или для личного пользования, а доступ к режиму многопользовательской веб-конференции, как правило, требует расходов на оформление подписки.

Перечисленные программные продукты позволяют довольно легко организовывать занятия в режиме веб-конференции с базовыми возможностями, но они не являются оптимальными решениями для организации регулярных занятий, поскольку имеют закрытый исходный код и не позволяют изменить или расширить функциональность по желанию пользователей. У этих приложений могут отсутствовать некоторые из средств, воссоздающих атмосферу реальных занятий, таких как поддержание постоянного зрительного контакта между всеми без исключения участниками, функции «поднятия руки» и «вызова к доске», инструменты проверки знаний по различным дисциплинам, изучаемым в вузах. Также у перечисленных систем отсутствуют средства работы с формулами и другими математическими объектами, а проверка знаний реализована лишь в форме тестирования без возможности исправления ошибок под руководством преподавателя. Поэтому на практике их использование в вузах обычно ограничивается проведением единичных консультаций, совещаний, дискуссий и т.п.

Разумной альтернативой использования интернет-сервиса является разработка новой программной системы, располагающей соответствующими инструментами и позволяющей добавлять или изменять их состав и функциональность. Наличие собственной системы для проведения занятий в форме веб-конференций помогает учебному заведению избавиться от издержек, связанных с использованием сторонних разработок. Разработчик такой системы может непосредственно участвовать в её практическом применении и оперативно исправлять ошибки, реализовывать пожелания пользователей, добавлять или модифицировать функциональность её компонентов.

Большинство образовательных информационных систем включают следующие компоненты [37, 45]:

- модель текущих знаний студента, представляющую понимание студентом материала курса и составляемую на основе прошедших результатов студента и информации о его поведении при решении задач;
- систему индивидуального планирования процесса обучения, включающую в себя знания из определённой области и механизмы вывода, позволяющие программной системе применять эти знания для выбора или составления задач, проверки результатов решения, управления процессом обучения;
- механизмы и интерфейсы взаимодействия студента и преподавателя с системой, служащие для обеспечения процесса обучения.

Помимо проверки результатов самостоятельной работы студентов, преподаватель должен следить за успеваемостью каждого студента, чтобы уровень сложности предлагаемых задач соответствовал уровню подготовки студента. При большом количестве студентов это становится трудной задачей, особенно в случае, если преподаватель не знаком лично со студентами. Помочь преподавателю может информационная система, обрабатывающая данные о текущей успеваемости студента и дополнительную информацию о нем, имеющуюся на момент поступления в учебное заведение. Такая система может не только производить выбор уровня сложности, соответствующего способностям и потребностям студентов, но также может обеспечивать обратную связь в процессе обучения. Система может автоматически анализировать успеваемость каждого студента и на основе этой информации предлагать студенту наиболее подходящий ему уровень сложности решаемых заданий. Таким образом, при использовании подобной информационной системы учебный процесс может быть организован более эффективно, особенно с учетом того, что при обучении в режиме веб-конференции существенной проблемой является экономия временных и вычислительных ресурсов.

Для успешного решения поставленной задачи по проектированию программной системы вначале необходимо провести этап сбора и анализа требований. Цель этапа – точно определить функции, поддерживаемые программной системой, и выделить список требований различного уровня к системе [31, 39].

1.2. Требования к программной системе и к ее компонентам

Определение. Требования к программному обеспечению — это совокупность спецификаций относительно атрибутов, свойств или качеств разрабатываемой программной системы, описывающих характеристики, свойства и функциональные возможности программной системы, а также ограничения, накладываемые на нее [31].

Требования к ПО делятся на два больших уровня – функциональные и нефункциональные требования [53].

Функциональные требования идентифицируют задачи или действия, которые система способна выполнять в ходе своего функционирования.

К функциональным требованиям относятся пользовательские требования, определяющие набор пользовательских задач, решаемых при помощи программной системы, а также способы (сценарии) их решения. Одним из способов представления этого вида требований являются варианты использования («use case» в нотации языка моделирования UML) [27].

Также к функциональным требованиям относятся системные требования, представляющие собой характеристики аппаратного обеспечения (тип и частота процессора, объём оперативной памяти, объём жесткого диска и т.д.) и требования к программному окружению, необходимому для корректной работы программной системы (операционная система, наличие установленных компонентов и т. п.).

Нефункциональные требования – это группа требований к программному обеспечению, которые определяют критерии работы системы в целом и не связаны непосредственно с функциями, выполняемыми системой [48]. Нефункциональные требования определяют такие свойства как стабильность, производительность, удобство сопровождения, расширяемость, надежность. В работе [65] перечислены наиболее типичные нефункциональные требования:

T1. Доступность.

T2. Дружественный для пользователя интерфейс.

T3. Надежность (автоматический перезапуск, восстановление работы, дублирование и резервирование важных данных),

T4. Масштабируемость (требования к возможности горизонтального или вертикального масштабирования приложения),

T5. Удобство поддержки.

T6. Безопасность (разграничение доступа, снижение подверженности рискам от внешних атак).

T7. Производительность (количество одновременно работающих пользователей, обслуживаемых транзакций, времени реакции, продолжительности вычислений, скорости и пропускной способности каналов связи).

T8. Требования к платформонезависимости.

T9. Требования к протоколам и технологиям взаимодействия между компонентами решения и между внешними компонентами.

T10. Требования к скорости и стоимости разработки.

T11. Требования к возможности автоматического и ручного тестирования приложения.

T12. Гибкость (возможность внесения изменений в различные компоненты программной системы, находящейся в эксплуатации).

Для сложных программных систем и комплексов, состоящих из многих компонент, процесс анализа требований для каждого компонента выглядит так же, как и процесс формирования требований для всей системы. Поэтому для них также производится анализ требований и выделяются требования различных уровней.

Далее представлены функциональные и нефункциональные требования к проектируемой программной системе дистанционного обучения и ко всем ее компонентам, выбрана архитектура системы и ее компонентов, сформированы требования различных уровней для каждого компонента программной системы.

1.2.1. Функциональные требования к СДО

Функциональные требования к СДО в целом можно сформулировать следующим образом.

Разрабатываемая система должна быть ориентирована на дистанционное обучение различным дисциплинам с использованием мультимедийных технологий и синхронной работы в режиме реального времени. СДО должна быть инвариантной относительно предметной области, т.е. должна позволять проводить занятия по различным дисциплинам.

СДО должна поддерживать удаленное обучение с использованием выхода в Интернет для обеспечения возможности доступа из дома, других учреждений и городов.

Необходимо также наличие функции авторизации и разграничения доступа для пользователей с различными ролями (преподаватели, студенты, администраторы).

Основным функциональным требованием к СДО является поддержка функций, позволяющих обеспечивать всестороннее взаимодействие преподавателя и студентов в режиме реального времени и тем самым воссоздавать полноценный учебный процесс. К ним относятся следующие функции:

- мультимедийные функции режима реального времени (общение в режиме веб-конференции, показ презентаций, работа с виртуальной доской, обмен файлами, показ своего рабочего стола);
- функция решения учебных заданий в режиме реального времени;
- функция создания учебных заданий с мультимедийными элементами (графики, рисунки, формулы, произвольные фигуры и изображения, текст) и задание экспертных оценок их сложности;
- просмотр и редактирование структуры учебных материалов, текущих результатов студентов;
- составление расписания занятий с учетом текущего уровня подготовки каждого студента по данной дисциплине.

Поскольку разрабатываемая СДО является инвариантной относительно предметной области, то необходимо наличие удобного инструмента создания учебных заданий, предусматривающего как ввод формул, так и создание текстовых заданий либо заданий с рисунками, графиками, изображениями из графических файлов. Интерфейс для решения задач должен предусматривать возможность работы с формулами с использованием WYSIWYG-редактора, т.е. редактора, следующего принципу «What You See Is What You Get» и позволяющего формировать материалы непосредственно в таком же виде, в котором они будут представлены в ходе занятия [101].

Для учета оценок и успеваемости студентов необходимо наличие базы данных с удобным веб-интерфейсом, позволяющим визуализировать оценки студентов в различных разрезах.

СДО должна предусматривать возможность индивидуализации процесса обучения, т.е. разделения студентов на группы с близким уровнем соответствия успеваемости и сложности текущего занятия и отдельного проведения занятий для данных групп. Для этого должно быть предусмотрено математическое обеспечение программной системы, обеспечивающее автоматическую обработку данных об успеваемости каждого студента

(начальной и текущей) и экспертных данных о сложности каждого варианта учебного задания для текущего занятия.

Программная система, позволяющая проводить учебные занятия в режиме веб-конференции, должна предусматривать поддержку следующих пользовательских функций для трех ролей (актеров): преподаватель, студент, администратор. Ниже приведены спецификации в форме вариантов использования («use case») для каждого актора.

Актор: Преподаватель.

- Просмотр структуры учебных материалов
 - Создание дисциплины, темы или занятия
 - Изменение порядка следования тем или занятий
 - Перенос тем между дисциплинами и занятий между темами
- Просмотр результатов обучения студентов
 - Просмотр таблицы успеваемости студентов
 - Ручное добавление и изменение оценок
- Создание и редактирование учебных заданий
 - Добавление задания к выбранному занятию
 - Редактирование задания (добавление шагов задания, создание формул, добавление рисунков, чертежей, графиков, текстовых комментариев, выбор скрытых элементов, редактирование палитры)
 - Удаление задания
 - Задание экспертных оценок сложности заданий
- Проведение занятия в режиме веб-конференции
 - Устное общение с возможностью предоставления слова студенту
 - Отправка сообщений в текстовый чат
 - Показ презентации
 - Показ виртуальной доски с рисованием на ней
 - Рассылка произвольных файлов
 - Показ своего рабочего стола
 - Запуск пошагового решения учебного задания в режиме реального времени
 - Контроль правильности ответов студентов на каждый шаг учебного задания
 - Просмотр результатов студентов после решения учебного задания
- Составление расписания занятий
 - Выбор учебных групп, для которых составляется расписание

- Выбор дисциплины
- Ознакомление с рекомендациями системы и внесение коррективов
- Выбор даты и времени проведения занятия

Актор: Администратор системы.

Актор *Администратор системы* наследует все функции, доступные актору *Преподаватель*, а также имеет доступ к следующим функциям.

- Добавление пользователей
 - Назначение роли пользователю: преподаватель, студент, администратор
 - Ввод фамилии, имени, отчества, личных данных пользователя
 - Задание начального пароля пользователю
 - Ввод исходных данных об успеваемости студента
- Настройка экспертной системы
- Назначение дисциплин преподавателям
- Просмотр журнала расчетов системы

Актор: Студент.

- Просмотр списка изучаемых дисциплин
- Просмотр своих результатов по каждой дисциплине
- Просмотр расписания занятий
- Участие в занятиях в режиме веб-конференции
 - Использование функции «Поднять руку»
 - Устное общение при предоставлении слова
 - Отправка сообщений в текстовый чат
 - Прием файлов, отправленных преподавателем
 - Пошаговое решение учебных заданий
 - Просмотр результатов решения последнего задания

Системные требования

Для развертывания серверных компонент СДО необходима установка на сервере следующих программных средств.

- Операционная система семейства Linux (рекомендуется Ubuntu Server 14 или выше) или семейства Windows Server (рекомендуется Windows Server 2008 или выше).
- Веб-сервер Nginx версии 1.10 или выше.

- Интерпретатор Ruby версии 2.0.0 и система управления версиями Ruby RVM версии 1.25 или выше (для ОС Linux).
- СУБД PostgreSQL версии 9.1.3 или выше (возможно использование MySQL).
- Система контроля версий Git 1.8.3 или выше.
- Среда исполнения Java SE Runtime Environment 7.
- Дистрибутив Red5 версии 1.0.5 или выше.

Для полноценного использования разработанной информационной системы рекомендуются следующие аппаратные и программные средства для клиентских компьютеров.

- Персональный компьютер с процессором уровня не ниже Core 2 Duo и операционной системой Windows 7/8/10, OS X 10.8, Ubuntu 14 с графической оболочкой. Не рекомендуется использовать мобильные устройства (планшеты, смартфоны) ввиду ограниченной поддержки Flash-технологии на мобильных ОС.
- Графический планшет (желателен для работы с виртуальной доской и для работы с редактором задач), мышь.
- Постоянное соединение с Интернетом (желательно использовать широкополосный канал со скоростью 3-5 Мбит/с при наличии 10 участников в конференции; требования к скорости увеличиваются с ростом количества участников).
- Веб-камера, микрофон и наушники. Не рекомендуется использовать громкоговорящие устройства, поскольку они создают эхо и фоновый шум в эфире.
- Браузер Google Chrome (рекомендуется), Internet Explorer 11, Firefox, Opera или Safari (для OS X) с включенной поддержкой Flash.

Для работы с функцией показа своего рабочего стола преподавателю необходимо установить на свой компьютер бесплатно распространяемую программу Vn Screen Capture Driver, ссылка на скачивание которой появляется при первом запуске соответствующей функции.

1.2.2. Нефункциональные требования к СДО

Нефункциональные требования к разрабатываемой СДО можно сформулировать в виде следующего ряда характеристик, обозначенных в соответствии с перечнем требований, приведенных в разделе 1.2.

T1 (требование к доступности): программная система должна быть доступна через обычное домашнее соединение с Интернетом при отсутствии необходимости установки каких-либо программ на клиентские компьютеры.

T2 (наличие дружественного для пользователя интерфейса): интерфейс должен позволять начать работу с системой студентам и преподавателям без специальной подготовки.

T3 (требование к надежности): должна быть предусмотрена функция автоматической проверки работы серверных компонентов и их перезапуска в случае останова, а также возможность создания резервных копий базы данных.

T4 (масштабируемость): СДО должна позволять работу с использованием одного физического сервера, обеспечивающего работу всех серверных компонент, а также должна предусматривать возможность переноса данных и алгоритмов между серверами в случае замены сервера на более мощный. Должно быть предусмотрено развертывание серверных компонент на различных физических серверах.

T5 (удобство поддержки): система должна предусматривать доступ к серверу с использованием удаленной командной строки (SSH) для оперативного устранения проблем, а также наличие каналов связи с разработчиком для уведомления о проблемах.

T6 (безопасность): требуется наличие функции авторизации пользователей и современной, обновляемой защиты от внешних вторжений.

T7 (производительность): производительность программных и аппаратных средств должна обеспечивать надежную работу СДО при любом количестве участников в масштабах вуза, использующего инсталляцию программной системы (как правило, одновременно в системе могут присутствовать несколько десятков пользователей).

T8 (требования к платформенезависимости): СДО должна предусматривать работу на различных серверных платформах (Windows Server, различные дистрибутивы Linux) и на различных устройствах конечных пользователей (PC, Mac, Linux, за исключением мобильных устройств). СДО не должна требовать установки на клиентские компьютеры каких-либо программ (кроме браузера) или плагинов для браузеров, кроме повсеместно используемых (Java, Adobe Flash, Silverlight).

T9 (требования к протоколам и технологиям взаимодействия): клиентские устройства при работе с СДО должны обмениваться с сервером по стандартным протоколам, используемым в Интернете (TCP, HTTP).

T10 (требования к скорости и стоимости разработки): компоненты СДО должны быть доступны для разработки и модификации силами одного разработчика в срок порядка нескольких месяцев или лет.

T11 (требования к средствам тестирования): желательно наличие инструментария, позволяющего автоматически производить тестирование по заранее написанным сценариям тестов для выявления ошибок при разработке или после внесения изменений в программный код.

T12 (гибкость и модифицируемость): СДО должна позволять в короткие сроки вносить изменения в программный код любого компонента для устранения проблем или для реализации пожеланий пользователей, а также должна предусматривать возможность совершенствовать и дополнять функциональность СДО в соответствии с развитием методик учебного процесса.

1.2.3. Общая архитектура программной системы

Описываемая в данной работе система дистанционного обучения MathConference представляет собой набор программных компонентов, выполняющих взаимосвязанные функции, обменивающихся информацией при помощи выбранных протоколов и форматов передачи данных и составляющих единое средство для решения поставленной задачи. Поэтому данная система может быть названа программным комплексом.

Для реализации функциональных требований, предъявляемых к программному комплексу, необходимо обеспечить возможность обмена данных в режиме реального времени между клиентскими компьютерами. Это может быть реализовано с использованием клиент-серверной архитектуры, при которой клиентские устройства обмениваются мультимедийной информацией через сервер, а также запрашивают у сервера необходимые данные по протоколу HTTP (hypertext transfer protocol), повсеместно используемому в сети Интернет. Структура компонентов выбранной архитектуры СДО показана на рис. 1.

В качестве клиента выступает компьютер конечного пользователя – студента, преподавателя или администратора системы. Вся работа с системой производится через веб-браузер, в качестве которого может выступать Google Chrome, Mozilla Firefox, Microsoft Internet Explorer и другие. Работа в режиме веб-конференции реализуется при помощи мультимедийного клиентского веб-приложения, исполняемого в браузере. Создание учебных заданий производится при помощи редактора, который также представляет собой веб-приложение.

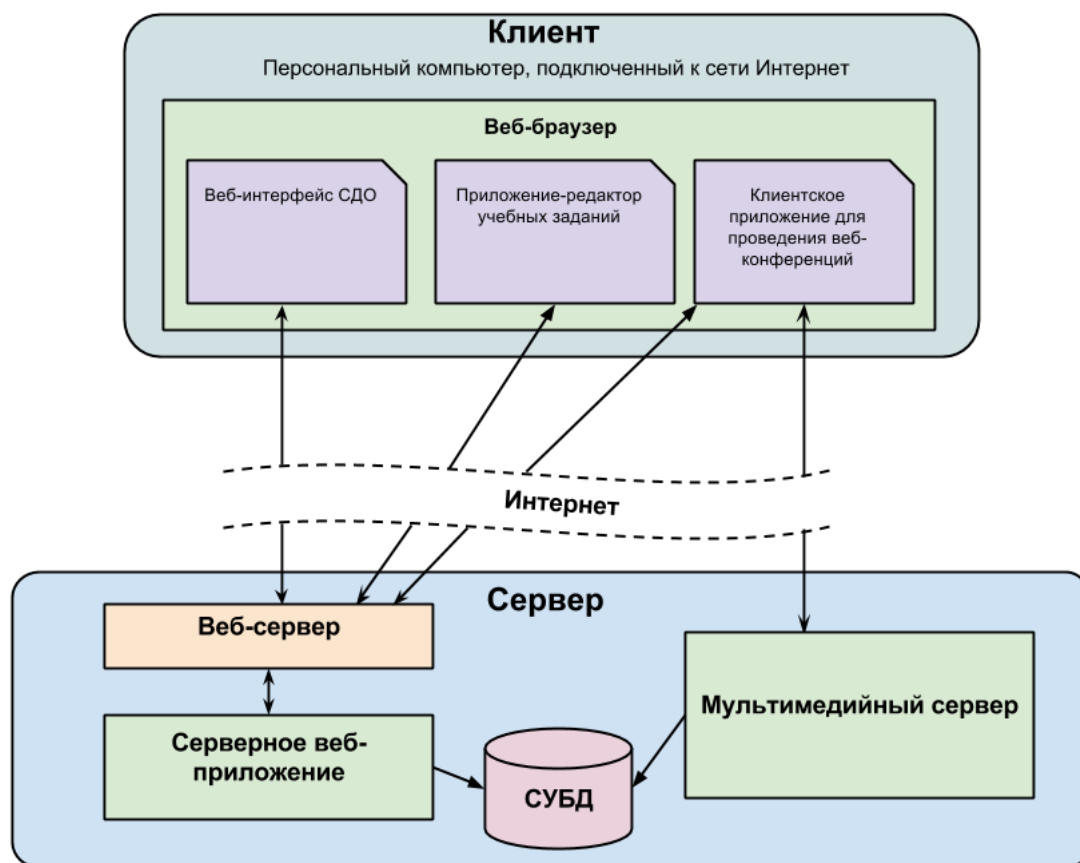


Рис. 1. Структура компонентов программной системы

Запросы, формируемые браузером или клиентскими веб-приложениями, обрабатываются веб-сервером, который перенаправляет эти запросы к серверному веб-приложению, реализующему логику работы СДО. Для работы веб-конференции необходим специальный мультимедийный сервер, задача которого – обработка потоков аудио- и видеоинформации и синхронизация событий, происходящих в виртуальной аудитории, у всех участников в режиме реального времени. Для долговременного хранения структурированной информации необходима система управления базами данных (СУБД), обрабатывающая запросы, поступающие к реляционной базе данных от серверного приложения и мультимедийного сервера.

Ниже приведен список функциональных и нефункциональных требований к каждому компоненту программной системы в соответствии с пунктами списка, приведенного в разделе 2.1. Приведен список доступных технологий, позволяющих выполнить указанные требования, и обоснован выбор наиболее подходящей технологии.

1.2.4. Требования к компонентам программной системы

1.2.4.1. Требования к клиентскому веб-приложению для проведения веб-конференций

Функциональные требования

Клиентское веб-приложение непосредственно реализует среду для проведения веб-конференций, включая дополнительные функции (чат, показ презентаций, решение учебных заданий, работу с виртуальной доской, показ рабочего стола, обмен файлами) и реализует все средства для работы с визуальными объектами – формулами, графиками – посредством WYSIWYG-редактора, то есть визуального редактора, следующего принципу «What You See Is What You Get». Идея этого принципа состоит в том, что все визуальные материалы создаются сразу в таком же виде, в котором они будут представлены в ходе занятия, при этом не требуется использование каких-либо языков разметки и форматирования.

Клиентское веб-приложение должно реализовывать функцию синхронизации всех происходящих событий у всех участников в режиме реального времени, обеспечивать потоковую передачу мультимедийных данных и обеспечивать следующие функции:

- устное общение с возможностью предоставления слова студенту;
- отправка сообщений в текстовый чат;
- показ презентации;
- показ виртуальной доски с рисованием на ней;
- рассылка файлов;
- показ своего рабочего стола;
- запуск пошагового решения учебного задания в режиме реального времени;
- контроль правильности ответов студентов на каждый шаг учебного задания;
- просмотр результатов студентов после решения учебного задания.

Из этих средств наиболее важной функцией является функция пошагового решения учебных заданий, позволяющая осуществлять обратную связь в процессе обучения. Средства проверки знаний могут быть реализованы в виде специального интерфейса, позволяющего осуществлять решение учебных заданий в режиме реального времени и предусматривающего:

1. Возможность работы с формулами с использованием WYSIWYG-редактора.
2. Возможность отображения учебных заданий для студентов различного уровня подготовки, созданных при помощи редактора заданий.

3. Одновременное пошаговое решение учебных заданий всеми студентами, присутствующими на занятии.
4. Автоматическое оценивание результатов работы каждого студента на основании количества попыток и ошибок при выполнении учебных заданий.

Использование подобного интерфейса позволит всем присутствующим на занятии решать задачи, непосредственно работая с формулами и рисунками. При этом нет необходимости реализовывать сложные алгоритмы проверки правильности ответов, набранных в виде формул, поскольку эта задача остаётся за преподавателем, как при традиционном обучении.

Нефункциональные требования

Список наиболее существенных нефункциональных требований для клиентского веб-приложения формируется следующим образом.

T1 (требование к доступности): приложение должно работать напрямую через браузер без необходимости установки дополнительных программ на клиентские компьютеры.

T2 (наличие дружелюбного для пользователя интерфейса): интерфейс должен быть интуитивно понятен и легок для освоения пользователями без каких-либо предварительных навыков. Особенно важно разработать интерфейс, не требующий знания каких-либо форматов данных для ввода математических формул, поскольку работа с формулами зачастую подразумевает предварительное овладение навыками форматирования формул (TeX, MathType и подобные технологии), а не все преподаватели и студенты имеют соответствующую подготовку. Интерфейс должен предусматривать возможность быстро и достаточно просто вводить ответ каждым студентом при выполнении учебных заданий.

T8 (требования к платформонезависимости): СДО должна предусматривать работу на различных платформах компьютеров конечных пользователей (за исключением, возможно, мобильных устройств).

T9 (требования к протоколам и технологиям взаимодействия): веб-приложение должно обмениваться с сервером по стандартным протоколам, используемым в Интернете (TCP, HTTP) с использованием форматов, поддерживаемых серверным веб-приложением (XML, JSON) и мультимедийным сервером (протокол RTMP, формат кодирования видеоданных FLV).

Выбор технологии для реализации

Клиентское веб-приложение для проведения веб-конференций представляет собой «богатое» интернет-приложение (Rich Internet Application, RIA). Доступны следующие современные технологии создания RIA:

- Adobe Flash – мультимедийная платформа компании Adobe для создания «богатых» интернет-приложений. Технология основана на использовании векторной графики и позволяет реализовать элементы мультимедиа: передачу потокового видео и звука, анимацию, интерактивность элементов интерфейса.
- Javascript – объектно-ориентированный язык разработки встраиваемых приложений, выполняющихся непосредственно в браузере клиента.
- Silverlight – мультимедийная платформа, разработанная Microsoft, по функциональности в целом аналогичная Adobe Flash.
- Java-апплеты – специальные Java-приложения, встроенные в Web-страницу. Эти приложения включаются в состав Web-страниц с целью добавления функциональности, которую сложно или невозможно реализовать с помощью скриптовых языков. Апплеты могут выполняться на всех платформах, для которых доступна виртуальная Java-машина.
- JavaFX – платформа для создания RIA, позволяет строить унифицированные приложения с насыщенным графическим интерфейсом. JavaFX расширяет возможности Java, позволяя использовать библиотеки Java в приложениях JavaFX при помощи специального языка программирования JavaFX Script.

Платформа Adobe Flash наиболее распространена в сфере создания «богатых» веб-приложений. Она обладает следующими достоинствами [107]:

- Высокая скорость и простота разработки благодаря наличию визуального редактора, позволяющего создавать визуальные элементы интерфейса приложения с использованием векторной графики.
- Большие возможности по работе с графикой, которые позволяют использовать различные встроенные визуальные эффекты и таким образом создавать веб-приложения с богатыми возможностями.
- Поддержка всех видов мультимедиа (изображений, звука, видео) в различных форматах и встроенная поддержка векторной графики.
- Развитые возможности языка ActionScript, используемого для программирования логики работы приложения.

- Возможность воспроизведения Flash-роликов внутри другого Flash-ролика, что актуально для демонстрирования презентаций, конвертированных в формат Flash.
- Поддержка средства синхронизации событий между всеми клиентами, подключёнными к серверу, называемого общими объектами («shared objects»).

Функциональность, обеспечиваемая технологией Flash, позволяет выполнить все функциональные и нефункциональные требования к веб-приложению, что обусловило ее выбор для разработки этого компонента.

1.2.4.2. Требования к приложению-редактору учебных заданий

Функциональные требования

Редактор учебных заданий предоставляет удобный набор инструментов для создания и редактирования учебных заданий, представляющих собой задачи для самостоятельного пошагового решения под контролем преподавателя в ходе веб-конференции. Инвариантность относительно предметной области должна быть обеспечена широким набором изобразительных средств, включающих:

- создание формул из символьных элементов (букв, цифр, математических символов), масштабирование и перемещение элементов, добавление фрагментов текста и полей для ввода значений с клавиатуры;
- создание рисунков, состоящих из геометрических фигур – окружностей, эллипсов, прямоугольников, прямых и произвольных линий, стрелок и т.д.;
- добавление изображений из внешних графических файлов.

Интерфейс редактора должен позволять создавать несколько задач для одного занятия с возможностью быстрого переключения между задачами.

Нефункциональные требования к редактору учебных заданий в целом аналогичны нефункциональным требованиям к клиентскому приложению.

T1 (требование к доступности): редактор должен работать напрямую через браузер клиентского компьютера.

T2 (требование наличия дружелюбного интерфейса): для комфортной работы требуется наличие достаточно простого интерфейса, доступного для освоения неподготовленным пользователем.

T8 (требования к платформонезависимости): редактор должен предусматривать работу на различных платформах и операционных системах компьютеров конечных пользователей.

T9 (требования к протоколам и технологиям взаимодействия): редактор должен использовать стандартные протоколы обмена данными с сервером – TCP, HTTP.

Технология должна поддерживать сохранение формул и графических элементов в компактном текстовом формате (XML или JSON) и обратное преобразование сохраненных материалов в наглядный графический вид и, соответственно, иметь встроенную поддержку обработки указанных форматов и средства векторного рисования графических примитивов с параметрами, указанными в XML- или JSON-файле.

Ввиду аналогичности требований к редактору и клиентскому веб-приложению для разработки редактора была также выбрана технология Adobe Flash.

1.2.4.3. Требования к мультимедийному серверу

Функциональные требования

Мультимедийный сервер – это компонент программной системы, обеспечивающий синхронную работу всех пользователей в режиме реального времени и обмен аудио- и видеoinформацией. К этому компоненту предъявляется следующий ряд функциональных требований.

- Наличие инструмента синхронизации событий, происходящих в ходе веб-конференции, и рассылка уведомлений об этих событиях всем клиентским приложениям, подключенным к серверу.
- Наличие средства синхронной передачи аудио- и видеопотоков от каждого клиента всем остальным клиентам, подключенным к серверу путем использования принципа «публикация-подписка».
- Возможность непосредственной работы с базой данных, общей с базой данных, используемой серверным приложением.
- Автоматическое подключение клиентских приложений в момент открытия пользователем веб-страницы, содержащей клиентское приложение для проведения веб-конференций, и отключение пользователей в момент закрытия веб-страницы или окна браузера.
- Обработка событий, происходящих в момент входа и выхода пользователей: рассылка события о появлении нового пользователя, определение места для добавления окна с видеопотоком пользователя, запись посещений в базу данных, удаление окон пользователей, покинувших конференцию.
- Запуск сервера перед началом занятия и его остановка после окончания занятия во избежание несанкционированного использования вычислительных ресурсов.

Нефункциональные требования

Нефункциональные требования, предъявляемые к мультимедийному серверу:

T3 (требование к надежности): технология должна обеспечивать стабильность и бесперебойность работы мультимедийного сервера с типичной нагрузкой (как правило, около 30-40 участников).

T7 (требование к производительности): производительность должна быть достаточной для одновременной работы студентов учебной группы, состоящей из нескольких десятков участников.

T8 (требования к платформонезависимости): для обеспечения переносимости необходима независимость реализации мультимедийного сервера от платформы, гарантирующая его работу на различных серверных платформах (Windows Server, различные дистрибутивы Linux).

T10 (требования к скорости и стоимости разработки): мультимедийный сервер должен быть основан на свободно распространяемых или недорогих технологиях, предусматривающих их применение для некоммерческих и образовательных проектов. Технология должна предусматривать быструю разработку собственного мультимедийного сервера на основе расширения базовой функциональности, поставляемой разработчиком технологии.

Выбор технологии для реализации

Для быстрой разработки мультимедийного сервера доступны следующие технологии:

- Adobe Flash Media Server – коммерческий медиасервер компании Adobe.
- Red5 – свободно распространяемый, расширяемый медиасервер с открытым исходным кодом, написанный на языке Java.
- Wowza – дорогостоящий коммерческий медиасервер для крупных проектов, принадлежащий Wowza Media Systems.

Из перечисленных решений лишь медиасервер Red5 удовлетворяет всем указанным требованиям к разрабатываемой системе, поскольку он относится к категории opensource-проектов и распространяется бесплатно. Red5 после установки обладает базовой функциональностью и предоставляет набор методов и предопределенных коллбэков (callback-функций, вызываемых автоматически при наступлении определенного события), позволяющих реализовать логику работы разрабатываемого сервера: вход и выход пользователей, учёт их посещаемости, синхронизация различных событий в режиме реального времени, оперативное и долговременное хранение данных.

По результатам тестирования, мультимедийный сервер на основе Red5 гарантированно обеспечивает стабильную работу веб-конференции при не менее чем 80 подключенных клиентах, что с избытком покрывает потребности разрабатываемой

системы. Это и обусловило выбор данного мультимедийного сервера для построения СДО.

1.2.4.4. Требования к серверному веб-приложению

Функциональные требования

Серверное веб-приложение – это компонент, который обеспечивает следующие функции программного комплекса:

- построение и отображение веб-интерфейса системы дистанционного обучения;
- авторизация пользователей и вход в систему;
- обработка HTTP-запросов на чтение и запись данных, поступающих от клиентского приложения и редактора учебных заданий;
- учёт успеваемости студентов;
- индивидуализация процесса обучения каждого студента;
- планирование занятий и составление расписания.

Одной из функций проектируемой программной системы является индивидуализация процесса обучения для каждого студента. Индивидуализация требует от системы принятия решений на основе знаний специалистов (преподавателей) о методах выбора подходящей сложности заданий. Таким образом, серверное веб-приложение должно включать алгоритмы вывода рекомендаций, опирающиеся на знания специалистов, что требует наличия следующие компонентов:

- база знаний, формализующая знания экспертов;
- механизмы логического вывода;
- интерфейс для работы с пользователем, включающий редактирование базы знаний, настройку алгоритмов расчета, просмотр журнала расчетов.

Таким образом, серверное приложение включает в себя программный компонент, обладающий всеми признаками экспертной системы. Серверное веб-приложение должно поддерживать реализацию алгоритмов, используемых экспертными системами, а также сохранять экспертную информацию и текущие данные в базе данных.

Нефункциональные требования

Нефункциональные требования к серверному приложению включают следующие пункты:

T6 (требование к безопасности): необходимо наличие функции авторизации пользователей и современной, обновляемой защиты от внешних вторжений.

T7 (требование к производительности): производительность должна быть достаточной для обеспечения учебного процесса в рамках одного вуза (несколько десятков или сотен пользователей).

T8 (требования к платформонезависимости): серверное веб-приложение должно предусматривать работу на различных серверных платформах (Windows Server, Linux).

T9 (требования к протоколам и технологиям взаимодействия): веб-интерфейс, клиентское приложение и редактор при работе с серверным приложением должны использовать только протокол HTTP, посредством которого передаются данные в форматах HTML, JSON, XML.

T12 (гибкость и модифицируемость): веб-приложение должно предусматривать оперативное внесение изменений в код для добавления новых функций, устранения ошибок и реализации пожеланий пользователей.

Выбор технологии для реализации

Для создания серверного веб-приложения, работающего в рамках программной системы указанных масштабов и удовлетворяющего перечисленным требованиям, доступны следующие широко распространённые технологии:

- язык PHP и фреймворки Yii, Zend, CakePHP, Laravel;
- язык Ruby и фреймворк Ruby on Rails;
- язык Python и фреймворк Django;
- язык Java и технология Java Server Pages (JSP);
- технология Active Server Pages (ASP) и др.

Для разработки серверного веб-приложения был выбран язык Ruby и фреймворк Ruby on Rails благодаря следующим преимуществам [87, 118]:

1. Фреймворк построен на основе архитектуры MVC (Model – View – Controller, Модель – Представление – Контроллер), что позволяет упорядочить программный код и полностью отделить реализацию серверных алгоритмов от процесса формирования HTML-страниц.
2. Использование Ruby on Rails позволяет достичь высокой скорости разработки приложений за счет сжатого, лаконичного синтаксиса языка Ruby и следования принципам приоритета соглашений над конфигурацией. Согласно этому принципу, если какой-либо компонент (класс, функция, таблица, файл и т.п.) соответствует соглашению наименования, то он наследует конфигурацию по умолчанию и не нуждается в дополнительной конфигурации.
3. Открытость и платформонезависимость языка Ruby, а следовательно, и всех технологий, созданных на его основе.
4. Развитая инфраструктура фреймворка Ruby on Rails, состоящая в том, что многие функции, такие как авторизация и разграничение полномочий пользователей, функции работы с различными структурами данных и т.д. либо встроены в

фреймворк, либо распространяются в виде бесплатных, легко подключаемых библиотек с открытым исходным кодом. Недостающие или не отвечающие требованиям алгоритмы могут быть дополнены или переработаны разработчиком самостоятельно.

5. Высокая надежность фреймворка, простота развертывания и поддержки готового веб-приложения.
6. Наличие развитого инструментария работы с базами данных, такого как Active Record, представляющего собой объектно-реляционную модель (ORM) и позволяющего работать с записями базы данных как с объектами языка Ruby, а с полями записей – как с атрибутами объектов [106].
7. Активное развитие и поддержка фреймворка сообществом разработчиков, гарантирующее его актуальность и соответствие требованиям времени на годы и десятилетия вперед.
8. Фреймворк Ruby on Rails предоставляет множество дополнительных функций, обеспечивающих удобную разработку, стабильную и безопасную работу всего серверного приложения: компоненты для авторизации, управления правами доступа, управления сессиями, cookies, генерацию SQL-запросов, в том числе для таблиц с отношениями «один-ко-многим» (1:N) и «многие-ко-многим» (N:N), средства тестирования приложения и многое другое.

Технологии Ruby on Rails присущи также некоторые недостатки, к которым можно отнести ограниченную поддержку многопоточных вычислений в языке Ruby версий 2.0-2.2 и не очень высокую производительность интерпретатора этих версий языка. Однако перечисленные недостатки существенны только для категории очень высоконагруженных проектов, к которым рассматриваемая система не относится.

Для построения веб-интерфейса используется язык разметки гипертекста HTML. Страницы могут включать в себя активное содержимое, управляемое сценариями, написанными на языке JavaScript, которые выполняются в браузере клиента. Для выполнения асинхронных запросов используется технология AJAX, позволяющая браузеру обмениваться информацией с сервером в фоновом режиме без перезагрузки страницы. Фреймворк Ruby on Rails позволяет разрабатывать шаблоны HTML-страниц не непосредственно, а с помощью шаблонизатора HAML, в котором вложенность тэгов HTML моделируется отступами строк. Это позволяет упростить и ускорить разработку сложных веб-интерфейсов. Для упрощения разработки сценариев JavaScript используется библиотека jQuery, позволяющая за счёт модификации синтаксиса JavaScript сделать работу с объектной моделью HTML-документа более эффективной. Также для упрощения

и ускорения разработки веб-интерфейса вместо JavaScript частично используется язык CoffeeScript, позволяющий писать более компактный код по сравнению с JavaScript. CoffeeScript транслируется специальным компилятором в JavaScript перед загрузкой веб-страницы в браузер клиента.

В фреймворке Ruby on Rails предусмотрена работа с двумя HTTP-серверами: Apache и Nginx. В качестве веб-сервера для программного комплекса выбран HTTP-сервер Nginx, поскольку он обладает значительно большей производительностью при работе с Rails-приложениями и более прост в установке и настройке.

1.2.4.5. Требования к СУБД

СУБД (система управления базой данных) – это компонент, отвечающий за долговременное хранение структурированных данных, а также за обработку запросов к базе данных от серверного приложения и от мультимедийного сервера.

Функциональные требования

- поддержка выполнения запросов на языке SQL (создание базы данных и таблиц, выборка, запись, удаление данных);
- поддержка функций резервного копирования и восстановления данных.

Нефункциональные требования

T3 (требование к надежности): СУБД должна надежно обеспечивать целостность данных путем поддержки транзакций, ограничений ссылочной целостности. При этом СУБД должна быть совместима с технологиями, использованными при создании других компонентов СДО, использующих для своей работы базу данных (мультимедийного сервера и серверного веб-приложения).

T5 (удобство поддержки): СУБД должна поддерживать интерфейс командной строки для ручного выполнения SQL-запросов администратором.

T7 (требование к производительности): СДО должна обеспечивать надежную работу при, как правило, небольшой нагрузке, соответствующей работе в масштабах одного вуза (в среднем один запрос в несколько секунд; во время работы экспертной системы может происходить несколько десятков запросов в секунду).

T8 (требования к платформонезависимости): СУБД должна иметь реализации для различных серверных платформ (Windows Server, Linux).

T10 (требования к скорости и стоимости разработки): СУБД должна быть бесплатной или недорогой, с лицензией, позволяющей ее применение для некоммерческих и образовательных проектов.

Выбор технологии

В качестве СУБД, удовлетворяющей указанным требованиям, разумно использовать одну из двух распространенных систем: MySQL или PostgreSQL. Обе СУБД распространяются бесплатно, не требуют сложной настройки и совместимы с фреймворком Ruby on Rails и мультимедийным сервером на основе Red5.

Преимущества СУБД MySQL:

- простая установка;
- легкое администрирование и использование;
- высокая скорость выборки данных благодаря упрощению стандартов SQL.

Преимущества СУБД PostgreSQL:

- высокое быстродействие при выполнении сложных запросов благодаря наличию оптимизатора запросов;
- большой запас масштабируемости;
- полное соответствие стандартам SQL;
- высокая компрессия данных, позволяющая экономить место на диске.

Фреймворк Ruby on Rails располагает набором адаптеров, отображающих объектно-реляционную модель на различные СУБД без каких-либо изменений кода разрабатываемой системы, поэтому возможно использование любой из перечисленных СУБД. В целом, PostgreSQL является более мощной и перспективной СУБД, что и обусловило ее выбор.

1.3. Выводы

В главе 1 проанализирована текущая практика использования веб-конференций в сфере образования, обоснована необходимость разработки новой системы дистанционного обучения с поддержкой режима обучения в форме веб-конференций, предоставляющей функцию решения учебных заданий в режиме реального времени и обладающей экспертной системой, позволяющей выстроить индивидуальный учебный процесс для каждого студента. Выделены функции, доступные различным ролям пользователей системы. Сформулированы требования различных уровней (функциональные, нефункциональные) к программной системе в целом и к каждому ее компоненту. Показана клиент-серверная модель организации компонентов системы. Обоснован выбор технологий для разработки каждого компонента.

В результате анализа требований и обзора существующих технологий были выбраны следующие технологии разработки программного комплекса СДО MathConference:

- для реализации серверного веб-приложения выбран фреймворк Ruby on Rails, созданный на основе языка Ruby;
- для разработки веб-интерфейса используется язык разметки гипертекста HTML с использованием сценариев на языках JavaScript и CoffeeScript и шаблонизатора HAML;
- в качестве СУБД выбрана свободно распространяемая СУБД PostgreSQL;
- для разработки клиентского веб-приложения и редактора учебных заданий выбрана технология Adobe Flash;
- для разработки мультимедийного сервера выбран свободно распространяемый сервер Red5, основанный на технологии Java.

Глава 2. Индивидуализация процесса обучения при помощи экспертной системы

2.1. Требования к процессу обучения в онлайн-системах СДО

Одним из преимуществ современных систем дистанционного обучения является их способность составлять модель знаний каждого пользователя и исходя из этого выбирать задачи. Модель знаний постоянно обновляется в ходе учебного процесса, позволяя адаптировать учебный процесс таким образом, чтобы сложность задач, решаемых студентом, соответствовала текущему уровню его подготовки.

В традиционных СДО время работы обычно не регламентируется, и студенту может быть предложено множество задач для самостоятельного решения. Специфика же систем реального времени состоит в том, что время проведения занятий строго регламентировано, и дополнительные попытки решения заданий приводят к трате времени преподавателя и расходу вычислительных ресурсов. Поэтому на первый план выходит соображение экономии времени и количества попыток решения каждого задания.

Поскольку решение задач происходит по шагам синхронно всеми участниками, то наиболее подготовленные студенты, которые справляются с задачами быстрее, вынуждены ждать, пока все остальные выполнят поставленную задачу. По той же причине наличие на занятии слабых студентов замедляет темп совместной работы, особенно при высокой сложности решаемой задачи. С другой стороны, проведение занятий в формате веб-конференции предъявляет высокие требования к квалификации преподавателя и требует от него определенных навыков, причем трудоемкость проведения таких занятий быстро возрастает с ростом количества участников конференции. Обе эти трудности могут быть преодолены путём разделения студентов на группы и отдельного проведения практических занятий с использованием задач различных уровней сложности. Это позволяет, с одной стороны, обеспечить присутствие на каждом практическом занятии студентов с сопоставимым уровнем успеваемости и, с другой стороны, индивидуализировать процесс обучения каждого студента. Соответственно, возникает задача создания заданий различного уровня сложности и формирования расписания занятий с разбиением студентов на группы, соответствующие этим уровням.

Для выбора уровней сложности решаемых задач разработана специальная экспертная система, интегрированная в единую среду с рассматриваемой системой дистанционного обучения MathConference. Работа экспертной системы заключается в подборе наиболее подходящего уровня сложности каждого занятия для каждого студента. По результатам расчётов автоматически составляется расписание занятий, где занятия различных уровней сложности разведены по времени. Пример расписания для одной группы студентов приведен в табл. 2.

Таблица 2. Пример расписания занятий для учебной группы

Дата	Время	Занятие	Уровень сложности	Студенты
03.02.2017	10:00	Дисциплина: Математический анализ Тема: Последовательности	1	Варфоломеев И. Гаврилов С. Зайцев А. Барынин А. Николаев Е. Яценко А. Кочетков В.
03.02.2017	13:00	Занятие: Вычисление пределов	2	Ярошенко М. Завальнюк М. Горячева О. Лазарева Т. Костина Е. Фролов Р.
01.02.2017	10:00	Дисциплина: Линейная алгебра	1	Ярошенко М. Яценко А. Варфоломеев И. Гаврилов С. Завальнюк М.
01.02.2017	13:00	Тема: Матрицы Занятие: Определители	2	Зайцев А. Кочетков В. Барынин А. Николаев Е. Горячева О.
02.02.2017	10:00		3	Лазарева Т. Костина Е. Фролов Р. Ануфриев Б.

Количество уровней сложности занятий задается преподавателем при их составлении.

Определение. *Индивидуальная траектория обучения* – это последовательность уровней сложности решаемых студентом задач в рамках заданной дисциплины.

Экспертная система позволяет автоматизировать процесс построения индивидуальной траектории обучения каждого студента и таким образом обеспечить соответствие сложности заданий потребностям и способностям каждого студента.

2.2. Постановка задачи формирования индивидуальной траектории обучения

Рассматриваемая модель учебного процесса опубликована в работе [15].

Пусть учебным планом по дисциплине предусмотрено некоторое количество занятий N . Сценарий обучения включает встречу участников в назначенное время в виртуальной аудитории, объяснение материала преподавателем с использованием доступных мультимедийных средств, а также решение всеми участниками одного или нескольких типовых заданий (упражнений) в реальном времени под контролем преподавателя. После каждого занятия по его результатам с учетом предыстории и данных о начальном уровне подготовки студенты делятся на три нечетких класса: «сильные», «средние», «слабые». Преподаватель составляет несколько вариантов (уровней) учебных заданий для одного и того же занятия, различающихся по сложности, и дает оценки сложности каждого созданного уровня для каждого класса студентов. Перед следующим занятием происходит процедура составления расписания, при котором каждому студенту из числа изучающих данную дисциплину назначается один определённый уровень сложности заданий предстоящего занятия. Соответственно, множество студентов S разбивается на подмножества, соответствующие уровням сложности этого занятия. Количество и состав этих подмножеств зависит от количества уровней сложности занятий и меняется от занятия к занятию.

Обозначим через $i \in [1, N]$ номер текущего занятия, где N – количество занятий по данной дисциплине, предусмотренное учебным планом. Пусть создано M_i вариантов уровней сложности задач в текущем занятии, а S – множество студентов в данной учебной группе, изучающих данную дисциплину.

Модель занятия в рассматриваемой СДО представляет собой кортеж

$$\left(M_i, (X_{i1}, \bar{C}_i(1)), (X_{i2}, \bar{C}_i(2)), \dots, (X_{iM_i}, \bar{C}_i(M_i)) \right), i \in [1, N], \quad (2.1)$$

где M_i – количество уровней сложности i -го занятия, X_{i1}, \dots, X_{iM_i} – текстовые, графические и формульные материалы для каждого уровня сложности, хранящиеся в базе данных, $\bar{C}_i(1), \dots, \bar{C}_i(M_i)$ – оценки сложности каждого уровня. Обозначим через $j \in [1; M_i]$ номер рассматриваемого уровня сложности текущего занятия. Оценка

сложности для j -го уровня представляется вектором $\bar{C}_i(j) = (C_{i1}(j), C_{i2}(j), C_{i3}(j))$, $j = 1, \dots, M_i$, где $C_{i1}(j) \in [0, 1]$ – оценка сложности j -го уровня i -го занятия для класса «сильные», $C_{i2}(j)$ – оценка сложности для класса «средние» и $C_{i3}(j)$ – оценка для класса «слабые». На рис. 2 проиллюстрирован пример оценок сложности для занятия, включающего четыре уровня сложности.

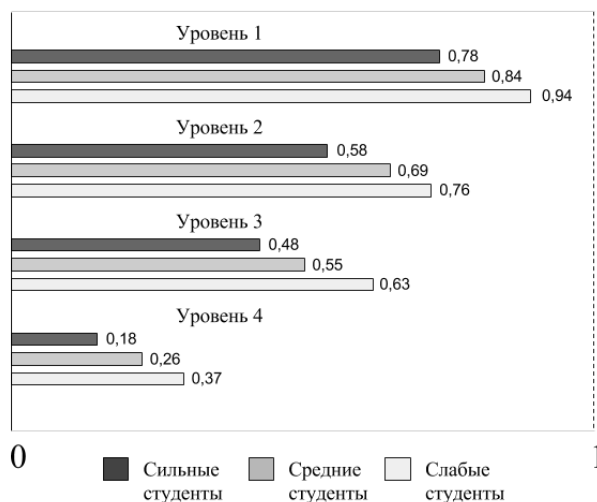


Рис. 2. Пример оценок сложности задания

Модель студента формируется исходя из доступных данных о начальной успеваемости студента (средний балл в аттестате, оценка в аттестате по данной дисциплине и др.) и предыдущих результатов его обучения в рассматриваемой СДО. Если в традиционных СДО могут использоваться сложные модели текущих и требуемых знаний студента, учитывающие взаимосвязи между темами заданной дисциплины, то рассматриваемая система инвариантна относительно предметной области, а формирование последовательности тем, занятий, учёт взаимосвязей между ними остаются за преподавателем, как при очном обучении. Таким образом, в качестве модели студента используются тройки величин

$$\langle \langle w \rangle_1^{i-1}, w_{i-1}, \bar{D} \rangle, \quad (2.2)$$

где:

- $\langle w \rangle_1^{i-1} \in [0; 1]$ – средневзвешенная сумма оценок, полученных студентом по данной дисциплине с первого до $(i-1)$ -го занятия включительно (при их наличии), вычисляемая по формуле

$$\langle w \rangle_1^{i-1} = \frac{\sum_{k=1}^{i-1} e^{-\beta(i-k-1)} w_k}{\sum_{k=1}^{i-1} e^{-\beta(i-k-1)}} \quad \text{при } i \geq 2, \quad (2.3)$$

где $w_k \in [0; 1]$ – оценка за k -е занятие, $k=1, \dots, N$; $\beta \in [0, \beta_{\max}]$ – задаваемый преподавателем параметр, характеризующий скорость уменьшения влияния предыдущих оценок на средневзвешенную сумму и позволяющий учесть изменение подготовленности студента по данной дисциплине со временем;

- $w_{i-1} \in [0; 1]$ – оценка, полученная студентом за предыдущее занятие (при наличии);
- \bar{D} – вектор, содержащий данные о начальной успеваемости студента: средний балл в аттестате, оценка в аттестате по данной дисциплине, оценка за ЕГЭ по дисциплине, средний балл за ЕГЭ по всем предметам, оценка за вступительный тест и первую контрольную работу.

Значение параметра β может быть задано администратором системы. Значение $\beta = 0$ соответствует простому среднему арифметическому всей истории оценок, т.е. отсутствию «забывания». При выборе максимально возможного значения $\beta = \beta_{\max} = 3$ происходит очень быстрое «забывание», и оценки, полученные более чем 2 занятия назад, практически не влияют на результаты расчетов экспертной системы (рис. 3). На практике рекомендуется использовать значение $\beta \approx 0,1 \dots 1$.

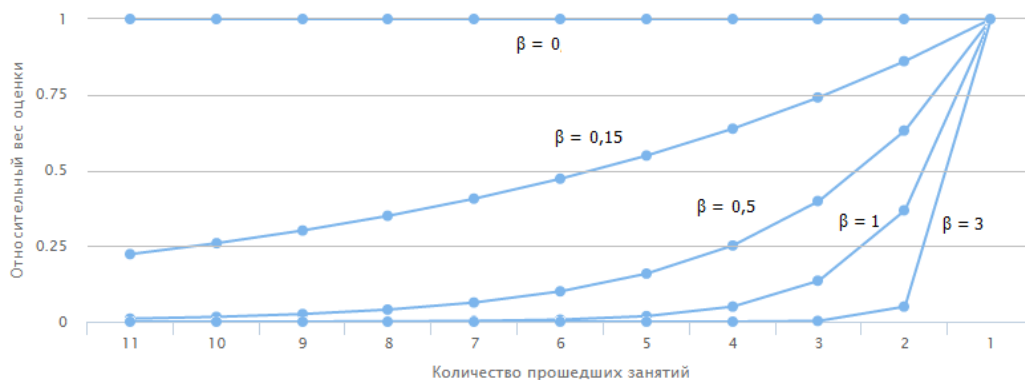


Рис. 3. Зависимость относительных весов оценок от параметра β

Наиболее важное значение имеет оценка предыдущего (последнего пройденного) занятия, поскольку она имеет наибольшее значение при выборе уровня следующего занятия или текущего занятия при его повторном прохождении. Например, если студент получил неудовлетворительную оценку за предыдущее занятие, то требуется повторное проведение этого же занятия, и неудовлетворительное значение оценки является

решающим при определении уровня сложности повторного занятия. Включение предыдущей оценки только в средневзвешенную сумму $\langle w \rangle_1^{i-1}$ повлияло бы на результат вычислений гораздо меньше, особенно при большой длине истории оценок и при малом значении параметра β . Поэтому обработка оценки за предыдущее занятие w_{i-1} требует подхода, отличного от обработки всей истории оценок.

Требуется создать модель работы информационной системы, автоматически составляющей расписание проведения текущего занятия для различных уровней, т. е. формирующей разбиение множества студентов S на подмножества S_1, S_2, \dots, S_{M_i} , соответствующие уровням сложности занятия.

Для разбиения множества S требуется вычислить рекомендуемый номер уровня J_i текущего занятия для каждого студента с использованием всей доступной информации, содержащейся в модели студента и модели занятия, т.е.

$$J_i = J_i \left(\langle w \rangle_1^{i-1}, w_{i-1}, \bar{D}, \bar{C}_{i-1}(J_{i-1}), \{ \bar{C}_i(j) \}_{j=1}^{M_i} \right), i \geq 2. \quad (2.4)$$

Для $i=1$ (первого занятия) номер рекомендуемого уровня должен определяться без учёта оценки за предыдущее занятие и истории оценок, т. е.

$$J_1 = J_1 \left(\bar{D}, \{ \bar{C}_1(j) \}_{j=1}^{M_1} \right). \quad (2.5)$$

Чтобы построить математическую модель задачи выбора рекомендуемого уровня сложности занятия для данного студента, введем понятие степени рекомендуемости.

Определение. *Степень рекомендуемости* – это числовой параметр $Q_i(j) \in [0; 1]$, соответствующий данному студенту и j -му уровню сложности i -го занятия и отражающий степень соответствия сложности заданий j -го уровня уровню подготовки студента.

Вычислив $Q_i(j)$ для всех $j=1, \dots, M_i$, можно определить рекомендуемый уровень, для которого достигается максимум степени рекомендуемости:

$$J_i = \arg \max_j Q_i(j). \quad (2.6)$$

Для автоматизации вычисления степени рекомендуемости необходимо построить информационную систему, автоматически обрабатывающую информацию о каждом студенте и вырабатывающую рекомендации о рекомендуемом уровне сложности следующего задания. Для этого предлагается использовать экспертную систему, работающую с использованием формализованных знаний экспертов (преподавателей). Для вычисления степени рекомендуемости $Q_i(j)$ используется нечёткий вывод,

представляющий собой процесс получения заключения о значении $Q_i(j)$ на основе нечётких правил, формализующих знания экспертов в данной области.

2.3. Экспертная система и её структура

Определение. *Экспертная система* – это вычислительная система, в которую включены знания специалистов о некоторой узкой предметной области в форме базы знаний, предназначенная для решения задач в заданной проблемной области путем моделирования механизма мышления человека.

В текущей работе экспертная система применяется для управления учебным процессом каждого студента путем моделирования мышления преподавателя, составляющего учебные материалы и назначающего каждому студенту задания, соответствующие его способностям и уровню текущей подготовки.

Задача экспертной системы состоит в формировании определенных выводов на основе заложенных в ней знаний экспертов. В данной системе знания экспертов (преподавателей) представлены в следующих видах:

1. Обучение нейронной сети модуля классификации произведено при помощи экспертных данных, предоставленных преподавателем, знакомым с группой студентов, данные о которых составляют обучающую выборку. Таким образом, его знания представлены в виде параметров и весов связей обученной нейронной сети.

2. Правила нечеткого вывода, содержащиеся в трех блоках нечеткого вывода, отражают мнения экспертов о значении выходных переменных при заданных входных значениях.

Полная структура экспертной системы приведена на рис. 4. В состав экспертной системы входят следующие компоненты: блок нечёткой классификации, четыре блока фаззификации, три блока нечеткого вывода и блок дефаззификации. Структура системы и принцип ее работы опубликованы в работах [8, 15, 19].

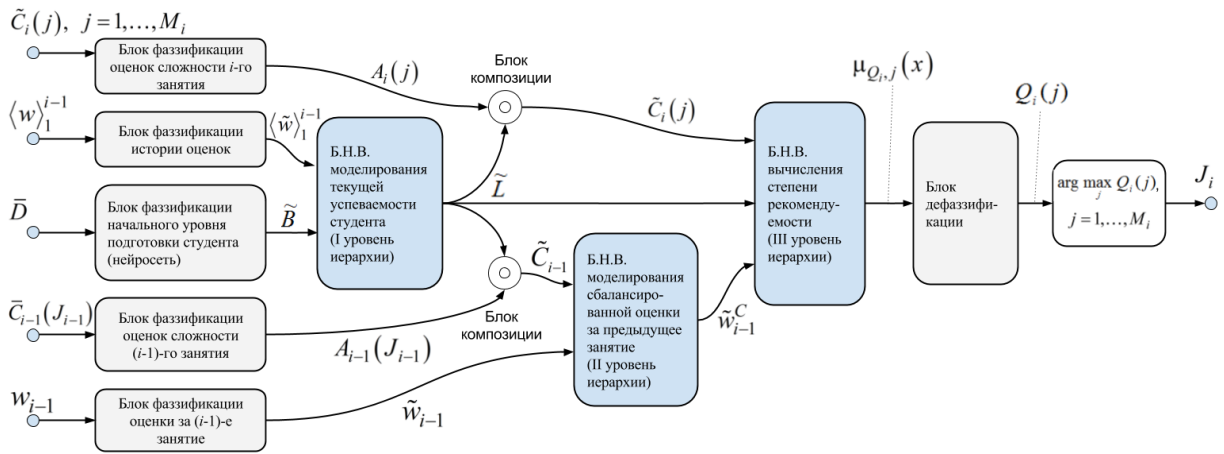


Рис. 4. Структурная схема иерархической нечеткой экспертной системы

Для моделирования величин, описываемых при помощи нечетких множеств, в экспертной системе используются *лингвистические переменные*. Понятие лингвистической переменной введено создателем теории нечеткой логики и нечетких множеств Л. Заде для построения моделей приближенных рассуждений человека для их использования в вычислительных системах [34].

Определение. Лингвистическая переменная – это пятерка объектов $\{X, T(X), U, G, M\}$, где:

X – название переменной;

$T(X)$ – терм-множество переменной X , т. е. множество значений лингвистической переменной, каждое из которых является нечетким подмножеством универсального множества U ;

U – универсальное множество, на котором определены термы, с базовой переменной x ;

G – синтаксическое правило для образования имен значений переменной X из множества термов $T(X)$ (например, правила использования связок «и», «или» и модификаторов «не», «очень», «не очень» и т.д.);

M – семантическая процедура, позволяющая преобразовать новое имя, образованное процедурой G , в нечеткую переменную, т.е. задать вид функции принадлежности для каждого значения переменной X при помощи операций над нечеткими множествами.

Для моделирования переменных в рассматриваемой системе используются лингвистические переменные с конечным набором термов, выбранных таким образом, чтобы экспертам было удобно составлять правила нечеткого вывода на естественном языке. Предполагается, что в правила вывода входят только атомарные термы лингвистических переменных, заданные при проектировании системы, поэтому

синтаксические правила G и семантические процедуры M для используемых лингвистических переменных не рассматриваются. Это позволяет упростить процедуру фаззификации и представлять значение лингвистической переменной в виде вектора, координаты которого представляют собой степени принадлежности базовой переменной упорядоченному множеству атомарных термов лингвистической переменной.

В экспертной системе используются четыре лингвистических переменных: сложность занятия, успеваемость студента, оценка за занятие, степень рекомендуемости.

Лингвистическая переменная C «сложность занятия» моделирует сложность задач, предлагаемых на занятии, для всех студентов какого-либо нечеткого класса – «сильные», «средние», «слабые» либо для конкретного студента. Терм-множество переменной имеет вид $T(C) = \{\text{«высокая»}, \text{«средняя»}, \text{«низкая»}\}$. Универсальное множество имеет вид $U = [0, 1]$, где $x = 0$ соответствует самой легкой задаче, а $x = 1$ – самой сложной. Графики функций принадлежности термов $\mu_C^1(x)$, $\mu_C^2(x)$, $\mu_C^3(x)$ приведены на рис. 5. В качестве функций принадлежности термов использованы треугольные и трапециевидные функции, поскольку они наиболее распространены и просты в программной реализации.

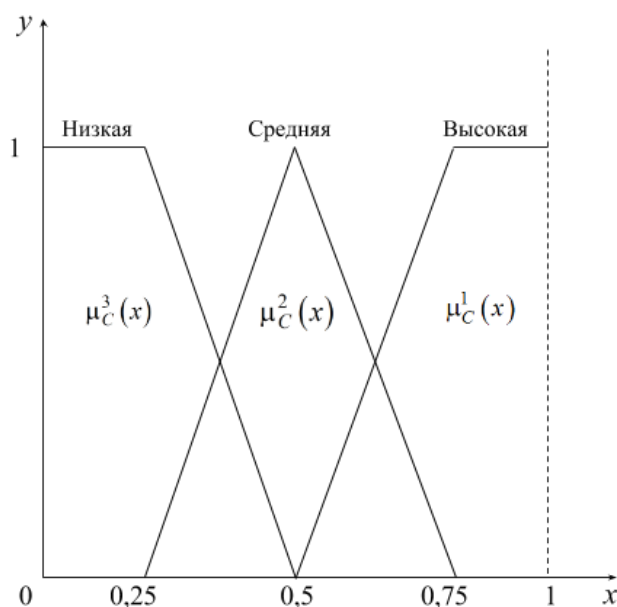


Рис. 5. Функции принадлежности термов переменной C «сложность занятия»

Лингвистическая переменная L «успеваемость студента» используется для моделирования начального или текущего уровня подготовки студента. Данная переменная

используется как промежуточная переменная в процедуре иерархического нечеткого вывода без проведения фаззификации и дефаззификации, поэтому для ее описания достаточно задать только терм-множество $T(L) = \{\text{«высокая»}, \text{«средняя»}, \text{«низкая»}\}$ без определения функций принадлежности.

Лингвистическая переменная W «оценка за занятие» моделирует оценку, полученную студентом за занятие. Поскольку для составления правил вывода преподавателю удобно оперировать привычными оценками из четырехбалльной системы, то в качестве терм-множества переменной выбрано множество $T(W) = \{\text{«отлично»}, \text{«хорошо»}, \text{«удовлетворительно»}, \text{«неудовлетворительно»}\}$. Универсальное множество имеет вид $U = [0, 1]$, где $x=0$ соответствует самой низкой оценке (полностью нерешенное задание), а $x=1$ – самой высокой оценке (полностью правильно решенное задание). Функции принадлежности термов приведены на рис. 6.

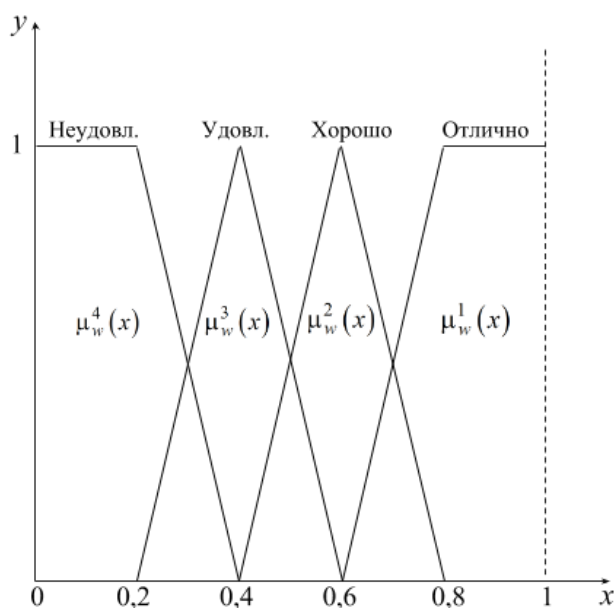


Рис. 6. Функции принадлежности термов лингвистической переменной W «оценка за занятие»

Замечание. Для удобства оценки в системе MathConference представляются в виде процентов. Самая низкая оценка ($x=0$) представляется значением 0%, а самая высокая ($x=1$) – значением 100%.

Лингвистическая переменная Q «степень рекомендуемости» моделирует значение степени рекомендуемости данного уровня сложности для определенного студента. Термножество переменной $T(Q) = \{\text{«очень высокая»}, \text{«высокая»}, \text{«средняя»}, \text{«ниже среднего»}, \text{«низкая»}\}$. Все термы переменной являются атомарными, несмотря на наличие

в названиях квантификатора «очень», т. е. функции принадлежности термов заданы при проектировании системы и не изменяются никакими синтаксическими правилами. Универсальное множество имеет вид $U = [0, 1]$, где $x = 0$ соответствует самому низкому значению рекомендуемости (уровень совершенно не рекомендуется), а $x = 1$ – самому высокому значению (уровень настоятельно рекомендуется).

Функции принадлежности μ_{Q_l} , $l = \overline{1, 5}$ термов переменной Q приведены на рис. 7.

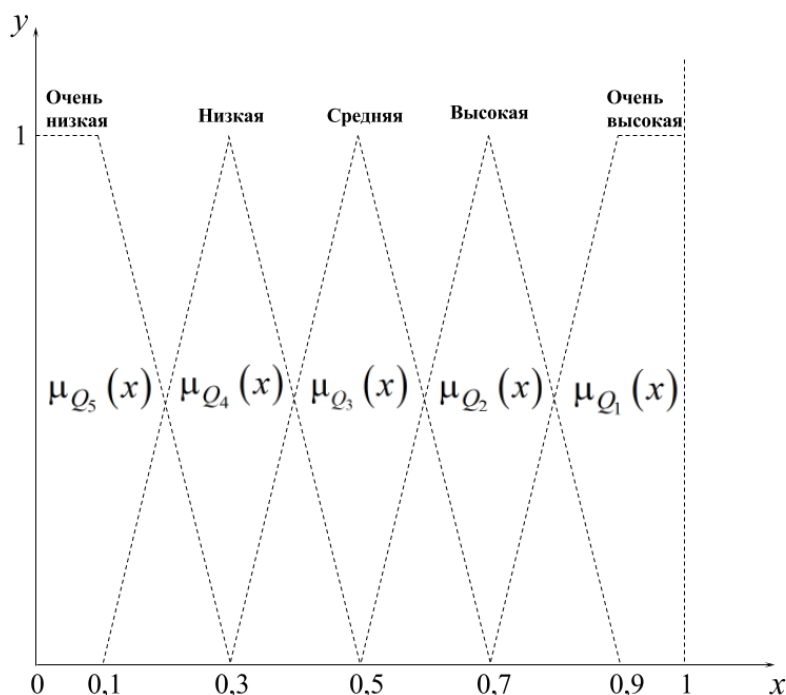


Рис. 7. Функции принадлежности термов переменной Q «степень рекомендуемости»

2.3.1. Фаззификация входных данных

Определение. *Фаззификация* – это установка соответствия между численным значением входной переменной системы нечеткого вывода и значениями функций принадлежности термов соответствующей лингвистической переменной.

Значения лингвистических переменных в рассматриваемой системе моделируются при помощи векторов, каждая i -я координата которых является значением функции принадлежности i -го атомарного термина соответствующей лингвистической переменной. Следовательно, выход блока фаззификации скалярной входной величины представляет собой вектор, координаты которого есть совокупность значений функций принадлежности термов лингвистической переменной для заданного значения входной

переменной. Блок фаззификации векторной входной величины производит фаззификацию каждой компоненты входного вектора по такому же принципу, поэтому выход блока фаззификации векторной величины представляет собой набор векторов одинаковой размерности, представленный единой матричной переменной.

Блок фаззификации оценок сложности i -го занятия осуществляет фаззификацию оценок сложности j -го уровня i -го (текущего) занятия. Сложность занятий задается преподавателем при подготовке материалов к занятию в виде вектора $\bar{C}_i(j) = (C_{i1}(j), C_{i2}(j), C_{i3}(j))$, где компонента $c_j \in [0; 1]$ задает экспертную оценку сложности заданий для студентов j -го нечеткого класса. Например, оценка сложности может иметь вид $\bar{C} = (0,4; 0,6; 0,8)$, что означает, что сложность этого занятия составляет 0,4 для сильных студентов, 0,6 для средних и 0,8 для слабых.

Блок фаззификации производит фаззификацию каждой компоненты вектора \bar{C} в соответствии с функциями принадлежности термов лингвистической переменной C . Блок порождает бинарное нечеткое отношение $R_C : X \times Y \rightarrow [0, 1]$, где базовое множество X представляет собой терм-множество лингвистической переменной C «сложность занятия», а базовое множество Y – множество нечетких классов студентов – {«сильные», «средние», «слабые»} [30, 61]. Матрица отношения имеет вид

$$A_i(j) = \|a_{kp,i}(j)\|, \text{ где } a_{kp,i}(j) = \mu_C^k(C_{ip}(j)), k = \overline{1, 3}, p = \overline{1, 3}, j = \overline{1, M_i}. \quad (2.7)$$

Следующий за блоком блок композиции осуществляет вычисление вектора $\tilde{C}_i(j)$, соответствующего степеням принадлежности сложности текущего задания для данного студента термам лингвистической переменной «сложность задания» C . Для расчета используется модель текущей успеваемости студента, представляемая лингвистической переменной L «текущая успеваемость». Вычисление производится при помощи операции композиции нечетких отношений по формуле:

$$\tilde{C}_i(j) = A_i(j) \circ \bar{L}, \quad (2.8)$$

где $A_i(j)$ – матрица нечеткого отношения R_C , \bar{L} – вектор, представляющий значение лингвистической переменной L «текущая успеваемость». Компоненты вектора $\tilde{C}_i(j)$ вычисляются по формуле максиминной композиции, где в качестве t -нормы используется операция вычисления минимума [62]:

$$\tilde{C}_{k,i}(j) = \max_{p=1,3} \min \{a_{kp,i}(j), L_p\}, k = \overline{1, 3}. \quad (2.9)$$

Блок фаззификации средневзвешенной суммы оценок осуществляет вычисление вектора, соответствующего степеням принадлежности средневзвешенной суммы оценок

$\langle w \rangle_1^{i-1}$ термам лингвистической переменной W «оценка за занятие». В случае отсутствия истории оценок у данного студента блок исключается из работы системы. Входной переменной является средневзвешенная сумма оценок $\langle w \rangle_1^{i-1}$, вычисленная по формуле (2.3). Выходом блока является вектор, каждая компонента которого является значением соответствующей функции принадлежности для заданного значения входной переменной:

$$\langle \tilde{w} \rangle_1^{i-1} = \left(\mu_w^1(\langle w \rangle_1^{i-1}) \quad \mu_w^2(\langle w \rangle_1^{i-1}) \quad \mu_w^3(\langle w \rangle_1^{i-1}) \quad \mu_w^4(\langle w \rangle_1^{i-1}) \right)^T. \quad (2.10)$$

Блок фаззификации оценок сложности $(i-1)$ -го занятия осуществляет фаззификацию оценок сложности предыдущего пройденного студентом занятия $\bar{C}_{i-1}(J_{i-1})$. Номер предыдущего уровня сложности J_{i-1} известен и хранится в базе данных. Выходом блока является матрица бинарного нечеткого отношения $A_{i-1}(J_{i-1}) = \|a_{kp}(J_{i-1})\|_{i-1}$, где $a_{kp}(J_{i-1}) = \mu_C^k(C_{i-1,p}(J_{i-1}))$. Принцип работы этого блока и следующего за ним блока композиции аналогичны принципам работы блока фаззификации оценок сложности i -го занятия и следующего за ним блока композиции.

Блок фаззификации оценки за $(i-1)$ -е занятие осуществляет вычисление вектора, соответствующего степеням принадлежности оценки за предыдущее пройденное занятие термам лингвистической переменной W «оценка за занятие». Входной переменной блока является значение оценки w_{i-1} , принимающее значения из отрезка $[0, 1]$ с шагом 0,01, а выход – вектор $\tilde{w}_{i-1} = \left(\mu_w^1(w_{i-1}) \quad \mu_w^2(w_{i-1}) \quad \mu_w^3(w_{i-1}) \quad \mu_w^4(w_{i-1}) \right)^T$.

2.3.2. Блок фаззификации начального уровня подготовки студента

Блок фаззификации начального уровня подготовки студента предназначен для оценивания уровня успеваемости каждого студента на основе информации о студентах, хранящейся в базе данных. Экспертная информация о начальном уровне подготовки каждого студента представлена вектором \tilde{B} , представляющим степени принадлежности начального уровня подготовки студента термам лингвистической переменной L «успеваемость студента».

Задачу вычисления степеней принадлежности начального уровня подготовки данного студента термам лингвистической переменной L можно рассматривать как задачу нечёткой классификации при наличии обучающей выборки. Нечёткая классификация

представляет собой разбиение множества студентов на априорно заданное число классов, при которой каждый студент может принадлежать одновременно нескольким классам с различной степенью принадлежности [30]. Исходная информация для классификации представлена в виде числовых данных, отражающих успеваемость каждого студента (оценки за ЕГЭ, вступительный тест и контрольные работы, средний балл в аттестате).

Пример вектора обучающей выборки приведён в табл. 3.

Таблица 3. Пример вектора обучающей выборки

Входной вектор \bar{D}						Выходной вектор \tilde{B}		
Средний балл в аттестате	Балл в аттестате по дисциплине	Средний балл за ЕГЭ	Балл за ЕГЭ по дисциплине	Оценка за вступительный тест	Оценка за контрольную работу	Экспертные оценки принадлежности студента нечетким классам		
						Сильный	Средний	Слабый
4,6	5	75	80	70	100	65	30	5

Блок фаззификации реализован на основе искусственной нейронной сети.

Определение. *Искусственная нейронная сеть* – это математическая модель, представляющая собой совокупность элементарных преобразователей входного сигнала (нейронов), соединённых между собой связями с определёнными весовыми коэффициентами [40, 78].

Искусственная нейронная сеть представляет собой упрощённую модель нервной системы человека и может использоваться для решения различных задач, в которых требуется моделирование мышления человека. Достоинством применения искусственных нейронных сетей является возможность обучения этой модели решению требуемой задачи по обучающей выборке [38]. В дальнейшем для обозначения искусственной нейронной сети будет использоваться более короткий термин «нейронная сеть» или «нейросеть».

Процесс работы нейронной сети в блоке фаззификации заключается в следующем. При разработке экспертной системы производится обучение нейросети на обучающей выборке, состоящей из данных о некотором количестве студентов, в результате чего сеть приобретает способность классифицировать произвольные входные данные. На этапе обучения на вход подаются координаты вектора \bar{D} , приведенные линейным преобразованием к интервалу $[-1, 1]$. Одновременно на выход сети представляется вектор

$\tilde{B} = \|b_i\|$, представляющий собой экспертную информацию о том, к какому классу и с какой степенью принадлежности относится данный студент. Экспертная информация основана на данных об успеваемости студентов и личном опыте преподавателей, работавших с этими студентами. Например, студент, по мнению преподавателя, может принадлежать классу «сильные» со степенью принадлежности $\mu = 0,6$, «средние» с $\mu = 0,3$ и «слабые» с $\mu = 0,1$. В этом случае экспертная информация будет представлена вектором $\tilde{B} = (0,6 \ 0,3 \ 0,1)^T$.

При реальной эксплуатации на вход нейросети подается предварительно обработанный вектор \bar{D} (также приведенный линейным преобразованием к интервалу $[-1, 1]$). При отсутствии какой-либо координаты вектора \bar{D} на соответствующий вход сети подается среднее (нулевое) значение. Значения сигнала на выходах сети трактуются как степени принадлежности предъявленного объекта соответствующим классам. Результат работы блока классификации можно записать в виде вектора-столбца $\tilde{B} = \|b_i\|$, $i = 1, \dots, K_L$, где b_i – степень принадлежности студента i -му классу.

2.3.3. Определение параметров нейронной сети

При использовании нейронной сети встает вопрос выбора архитектуры сети, параметров ее элементов, алгоритма и длительности обучения сети.

Для экспериментального определения подходящих параметров и для обучения сети была использована выборка из 220 записей о студентах. Проведены эксперименты с многократным обучением сети с различными комбинациями параметров. Для построения и обучения нейронной сети использована библиотека FANN (Fast Artificial Neural Network). Набор параметров, определяемых в ходе экспериментов, следующий:

- архитектура сети (количество скрытых слоев и количество нейронов в каждом слое);
- количество эпох обучения сети;
- крутизна функции активации нейронов скрытых слоев;
- алгоритм обучения из числа предоставляемых библиотекой FANN.

Выбор оптимального значения параметра заключается в переборе значений выбранного параметра из некоторого диапазона значений при фиксировании остальных параметров и измерении среднеквадратических ошибок обучения и тестирования на выходе нейросети. Каждая конфигурация тестируется 50 раз, и в качестве выходного значения ошибок обучения и тестирования выбирается среднее арифметическое значений

этих величин, зафиксированных в конце каждого цикла обучения. При каждом цикле обучающая выборка случайным образом делится на 170 обучающих и 50 тестовых записей, все значения входных параметров нормируются на диапазон $[-1, 1]$. В качестве функции активации нейронов скрытых слоев выбрана симметричная сигмоидная функция, описываемая формулой

$$f_z(x) = \frac{2}{1 + e^{-zx}} - 1, \quad (2.11)$$

где z – крутизна функции активации (рис. 8).

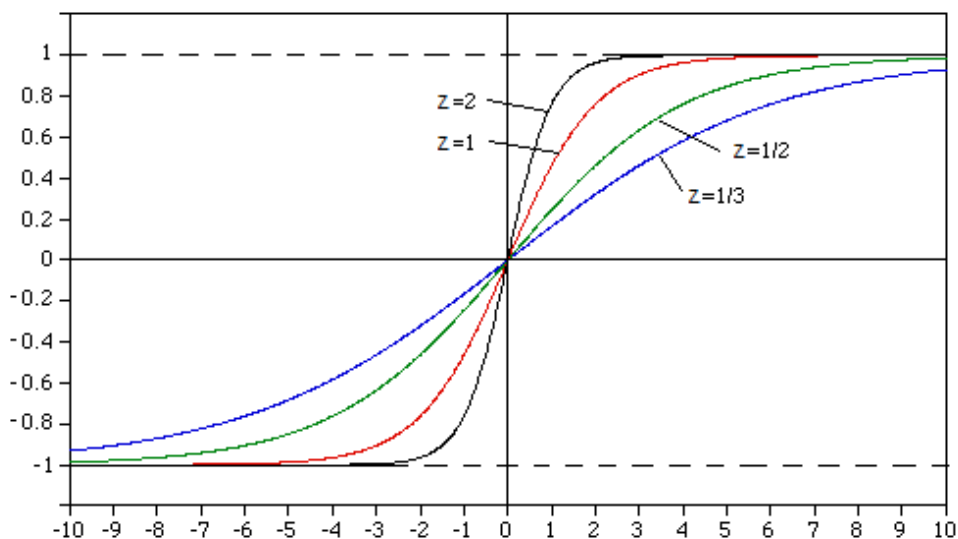


Рис. 8. Общий вид функции активации нейронов скрытых слоев

Оптимальные значения параметров выбраны из условия минимальных среднеквадратических ошибок тестирования сети. Ниже приведены примеры результатов расчетов, обусловивших окончательный выбор параметров сети.

2.3.3.1. Выбор количества итераций (эпох) обучения

Для выбора количества эпох обучения проведено многократное обучение сети с различными вариантами фиксированных параметров расчета. В разделе представлены результаты при некоторых комбинациях параметров.

Ниже приведены результаты расчета для однослойного перцептрона с 7 нейронами скрытого слоя.

Фиксированные параметры расчета:

- архитектура сети: один скрытый слой, количество нейронов: 7;
- крутизна функции активации: $z = 0,7$;
- алгоритм обучения: RPROP.

Переменный параметр: количество эпох обучения N , диапазон изменения: $N \in [10, 150]$ с интервалом в 10 итераций.

На рис. 9 приведен график зависимости ошибок обучения и тестирования от количества эпох обучения.

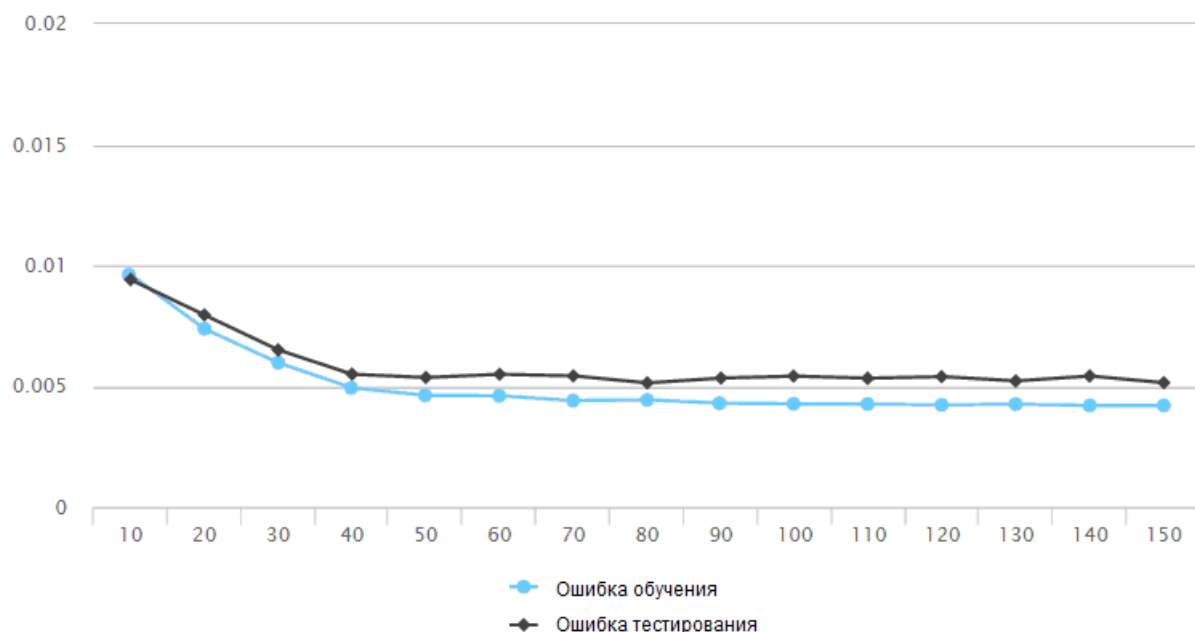


Рис. 9. Зависимость ошибок обучения и тестирования от количества эпох обучения (для однослойного перцептрона с 7 нейронами скрытого слоя)

Результаты расчета для сети с 10 нейронами скрытого слоя приведены на рис. 10.

Фиксированные параметры расчета:

- архитектура сети: один скрытый слой, количество нейронов: 10;
- крутизна функции активации: $z = 0,7$;
- алгоритм обучения: RPROP.

Переменный параметр: количество эпох обучения N , диапазон изменения: $N \in [10, 150]$ с интервалом в 10 итераций.

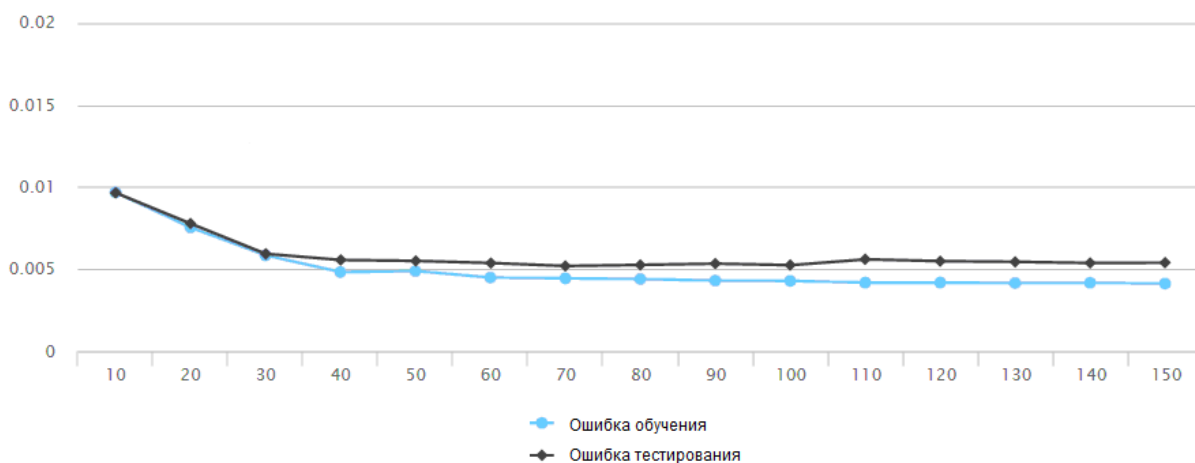


Рис. 10. Зависимость ошибок обучения и тестирования от количества эпох обучения (для однослойного перцептрона с 10 нейронами скрытого слоя)

Результаты расчета для сети с двумя скрытыми слоями приведены на рис. 11.

Фиксированные параметры расчета:

- архитектура сети: два скрытых слоя, количество нейронов: 4 в первом слое и 5 во втором слое;
- крутизна функции активации: $z = 0,7$;
- алгоритм обучения: RPROP.

Переменный параметр: количество эпох обучения N , диапазон изменения: $N \in [10, 150]$ с интервалом в 10 итераций.

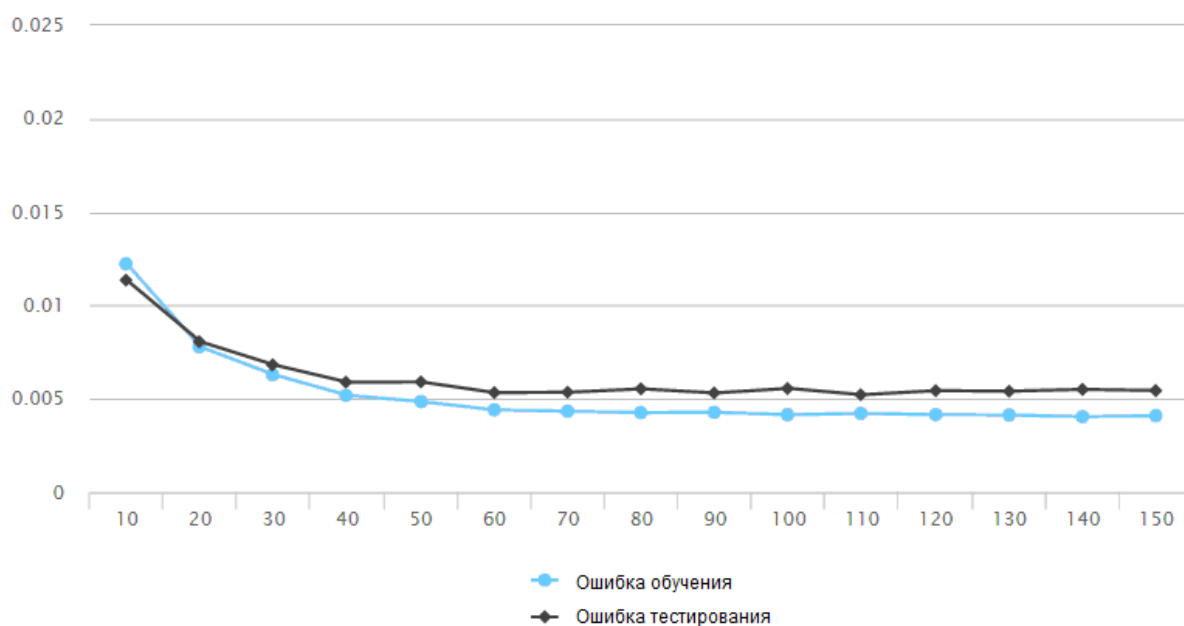


Рис. 11. Зависимость ошибок обучения и тестирования от количества эпох обучения (двухслойный перцептрон с 4 и 5 нейронами скрытых слоев)

При любой конфигурации сети прекращение уменьшения ошибки тестирования наступает после 60-70 итераций. Такой же результат получается при выборе других значений крутизны z , а также при других алгоритмах обучения. Поэтому в качестве итогового выбрано значение 70 итераций.

2.3.3.2. Выбор архитектуры сети

Для выбора архитектуры сети (количества слоев и нейронов в каждом слое) проведено многократное обучение сети со следующими параметрами.

Фиксированные параметры расчета:

- Крутизна функции активации: $z = 0,7$.
- Количество эпох обучения: $N = 70$.
- Алгоритм обучения: RPROP.

Переменный параметр: количество слоев и нейронов сети. Расчеты производятся для одно-, двух- и трехслойной архитектуры сети, количество нейронов представлено на оси абсцисс графика через запятую (1-й, 2-й и 3-й слой).

На рис. 12 приведен график зависимости ошибок обучения и тестирования от конфигурации сети.

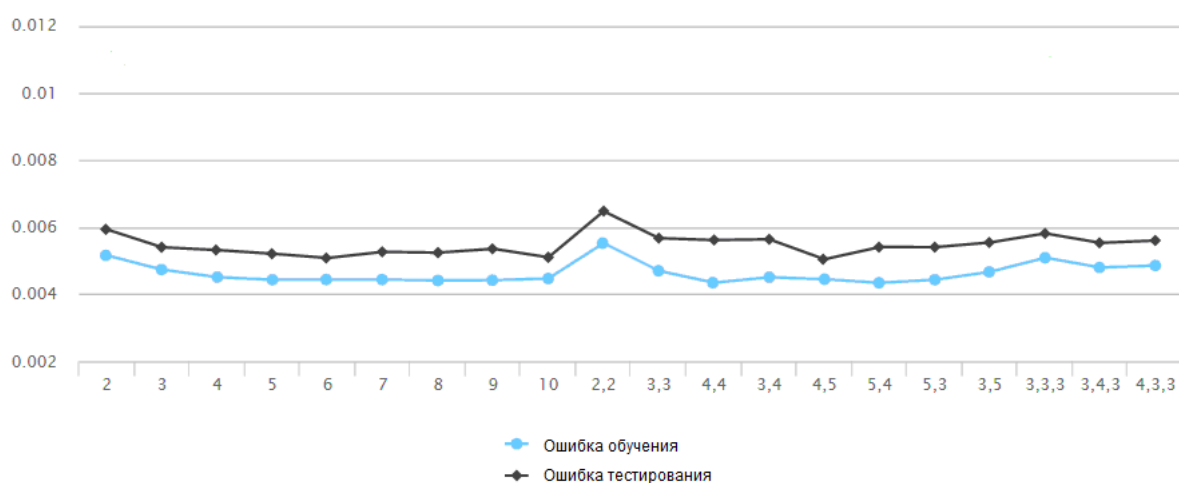


Рис. 12. Зависимость ошибок обучения и тестирования от количества слоев и нейронов сети ($z = 0,7$)

На рис. 13 приведены результаты аналогичных расчетов при крутизне $z = 0,5$. Фиксированные параметры расчета:

- крутизна функции активации: $z = 0,5$;
- количество эпох обучения: $N = 70$;
- алгоритм обучения: RPROP.

Переменный параметр: количество слоев и нейронов сети. Набор конфигураций аналогичен предыдущему расчету.

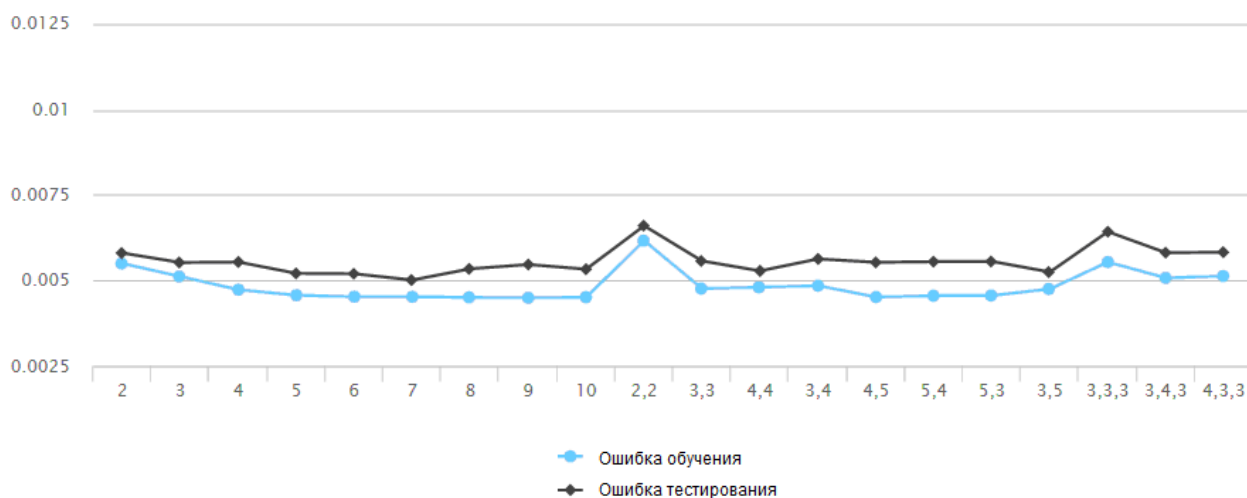


Рис. 13. Зависимость величины ошибок обучения и тестирования от количества слоев и нейронов сети ($z = 0,5$)

В результате наименьшая среднеквадратическая ошибка тестирования получена при конфигурации сети, имеющей два скрытых слоя с 4 и 5 нейронами в первом и втором слоях и при крутизне $z = 0,7$. Архитектуру такой сети можно обозначить как «6-4-5-3» [71].

2.3.3.3. Выбор крутизны функции активации нейронов скрытых слоёв

Для выбора крутизны функции активации проведено многократное обучение сети со следующими фиксированными параметрами (рис. 14):

- архитектура сети: два скрытых слоя, количество нейронов: 4 в первом слое и 5 во втором слое;
- количество эпох обучения $N = 70$;
- алгоритм обучения: RPROP.

Переменный параметр: крутизна функции активации нейронов скрытых слоев z , диапазон изменения: $z \in [0,1; 0,9]$ с интервалом изменения 0,1.

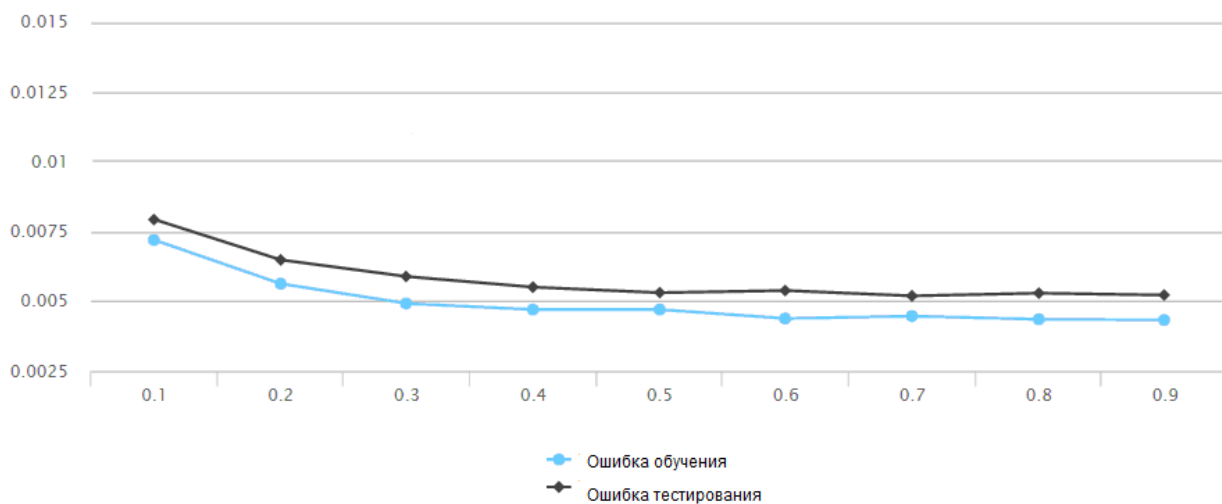


Рис. 14. Зависимость ошибок обучения и тестирования от крутизны функции активации нейронов

Наименьшие значения ошибок обучения и тестирования получены при значениях $z \in [0,5, 0,9]$, причем увеличение крутизны не приводит к уменьшению ошибки тестирования. Такие же результаты получены и при других значениях фиксированных параметров. Поэтому в качестве итогового значения выбрано среднее значение $z = 0,7$.

2.3.3.4. Выбор алгоритма обучения

Библиотека FANN предоставляет четыре алгоритма обучения нейросетей: `incremental` (стандартный алгоритм обратного распространения ошибки), `batch` (стандартный алгоритм с обучением по эпохам), `RPROP` (эластичный алгоритм обратного распространения с обучением по эпохам) [42, 97], `quickprop` (усовершенствованный алгоритм обратного распространения ошибки, использующий дополнительные параметры для обучения [42]). На рис. 15 приведено сравнение результатов обучения с использованием этих алгоритмов.

Фиксированные параметры обучения:

- архитектура сети: два скрытых слоя, количество нейронов: 4 в первом слое и 5 во втором слое;
- крутизна функции активации $z = 0,7$;
- количество эпох обучения: $N = 70$.

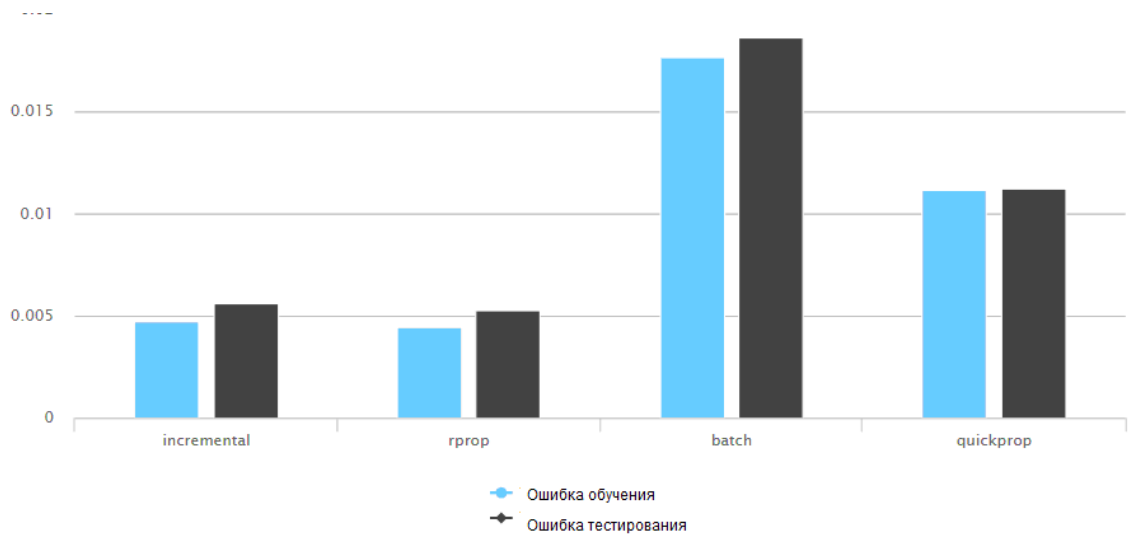


Рис. 15. Величины ошибок обучения и тестирования для различных алгоритмов обучения сети

Наименьшее значение среднеквадратических ошибок обучения и тестирования дает алгоритм RPROP. Название алгоритма означает «resilient propagation», или эластичный алгоритм обратного распространения ошибки. Это одна из модификаций стандартного алгоритма обратного распространения ошибки. Суть эластичного алгоритма обратного распространения ошибки состоит в следующем [81, 97]. Алгоритм использует обучение по эпохам, т. е. коррекция весов происходит после предъявления сети всех примеров из обучающей выборки. Обозначим через E значение среднеквадратической ошибки обучения сети, определяемой по формуле:

$$E = \frac{1}{2} \sum_{i=1}^n \sum_{p=1}^P (\varphi_{ip} - \hat{\varphi}_{ip})^2, \quad (2.12)$$

где n – количество нейронов в выходном слое, P – количество элементов обучающей выборки, φ_{ip} – значение на выходе i -го нейрона при подаче на вход p -го элемента обучающей выборки, $\hat{\varphi}_{ip}$ – требуемое значение на выходе i -го нейрона при том же входном элементе. Обозначим через t номер итерации (эпохи) обучения, а все значения величин, относящихся к этой итерации, будем обозначать верхним индексом (t) . Для каждого весового коэффициента w_{ij} связи между j -м и i -м нейронами вводится индивидуальная величина коррекции Δ_{ij} , определяемая по формуле:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ \Delta_{ij}^{(t-1)}, & \text{если } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t)}}{\partial w_{ij}} > 0, \\ \eta^- \Delta_{ij}^{(t-1)}, & \text{если } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t)}}{\partial w_{ij}} < 0, \\ \Delta_{ij}^{(t-1)}, & \text{иначе,} \end{cases} \quad (2.13)$$

где η^+ , η^- – параметры, выбираемые из условия $0 < \eta^- < 1 < \eta^+$. Данное правило применяется следующим образом: когда частная производная $\frac{\partial E^{(t)}}{\partial w_{ij}}$ ошибки по коэффициенту w_{ij} меняет знак, что означает, что на последнем шаге алгоритм проскочил локальный минимум, то величина коррекции Δ_{ij} уменьшается на коэффициент η^- . Если производная $\frac{\partial E^{(t)}}{\partial w_{ij}}$ сохраняет знак, то величина коррекции увеличивается, чтобы ускорить нахождение минимума.

После нахождения величины коррекции приращения весов связей вычисляются в соответствии со следующим правилом:

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, & \text{если } \frac{\partial E^{(t)}}{\partial w_{ij}} > 0, \\ \Delta_{ij}^{(t)}, & \text{если } \frac{\partial E^{(t)}}{\partial w_{ij}} < 0, \\ 0, & \text{иначе,} \end{cases} \quad (2.14)$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}.$$

Однако, если частная производная меняет знак, т.е. предыдущий шаг был слишком большим и алгоритм проскочил локальный минимум, то возвращается предыдущее значение величины коррекции, что выражается следующей формулой:

$$\Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)}, \text{ если } \frac{\partial E^{(t-1)}}{\partial w_{ij}} \cdot \frac{\partial E^{(t)}}{\partial w_{ij}} < 0. \quad (2.15)$$

Кроме того, при практической реализации алгоритма значение $\frac{\partial E^{(t)}}{\partial w_{ij}}$ в данном случае полагается равным нулю, чтобы избежать изменения величины коррекции Δ_{ij} на следующем шаге. В этом случае активизируется третье условие в формуле (2.13).

Параметрами алгоритма являются коэффициенты изменения величин коррекции η^+ , η^- , начальные значения величин коррекции Δ_0 и пределы изменения этих величин Δ_{\min} и Δ_{\max} . Значения этих параметров выбираются экспериментально.

На рис. 16 приведен график изменения ошибки в зависимости от параметра η^- при следующих фиксированных параметрах:

- Крутизна функции активации $z = 0,7$.
- Количество эпох $N = 70$.
- Количество скрытых слоев – 2, архитектура сети – «6-4-5-3».
- Алгоритм: RPROP (эластичный алгоритм обратного распространения ошибки).

Переменный параметр: величина коррекции η^- , диапазон изменения: $\eta^- \in [0,2; 0,9]$ с интервалом изменения 0,05.

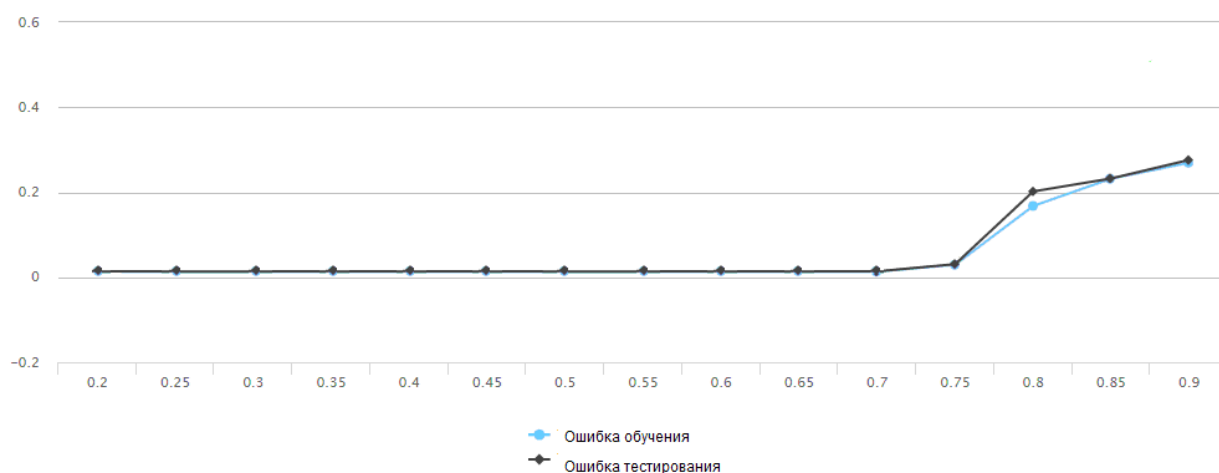


Рис. 16. Зависимость ошибок обучения и тестирования от величины параметра η^-

Видно, что изменение параметра η^- в довольно широких пределах от $\eta^- = 0,2$ до $\eta^- = 0,7$ никак не влияет на окончательный результат обучения сети. Дальнейшие расчеты со всевозможными комбинациями параметров показали, что то же самое справедливо по отношению к остальным параметрам алгоритма, то есть к достоинствам алгоритма можно отнести устойчивость к изменению начальных значений параметров. Помимо этого, алгоритм RPROP обладает рядом других достоинств [81]:

- быстрота и хорошее качество обучения по сравнению с другими распространенными алгоритмами;

- использование только производной первого порядка, что даёт линейную сложность алгоритма по количеству оптимизируемых параметров;
- независимость от модуля значения производной, что позволяет предотвратить «паралич» процесса обучения, когда значение производной близко к нулю.

В результате экспериментов получена следующая окончательная конфигурация нейронной сети. Архитектура сети и представляет собой многослойный перцептрон с 6 входными нейронами (в соответствии с размерностью вектора \bar{D}), двумя скрытыми слоями (4 нейрона в первом скрытом слое, 5 нейронов во втором) и 3 нейронами выходного слоя (архитектура «6-4-5-3», рис. 17).

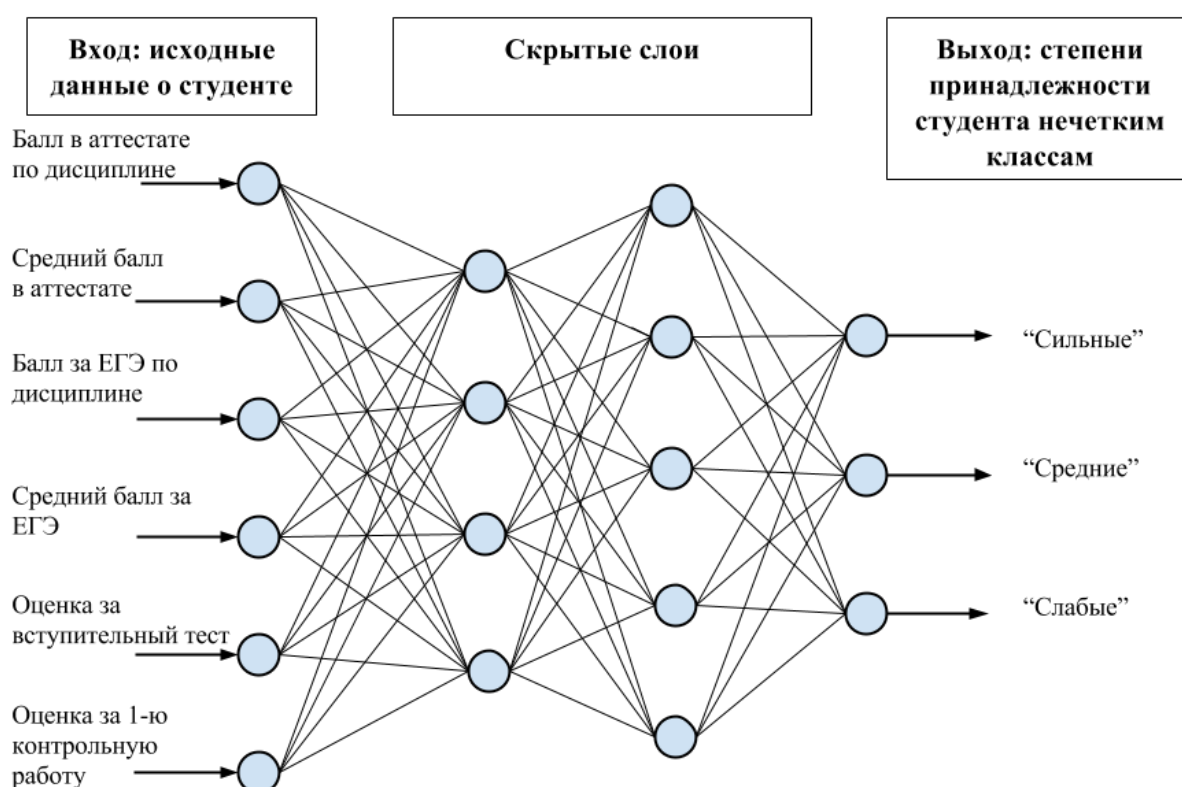


Рис. 17. Архитектура нейронной сети блока фазификации начального уровня подготовки студента

Функция активации нейронов скрытых слоев – симметричная сигмоидная с крутизной функции активации $z = 0,7$. Алгоритм обучения – эластичный алгоритм обратного распространения ошибки RPROP с параметрами $\eta^+ = 1,2$; $\eta^- = 0,5$; $\Delta_0 = 0,1$; $\Delta_{\min} = 1 \cdot 10^{-6}$; $\Delta_{\max} = 50$, количество эпох обучения – 70.

2.3.4. Блоки нечеткого вывода экспертной системы

Определение. *Нечетким логическим выводом* называется получение заключения в виде нечеткого множества, соответствующего текущим значениям входов, с использованием нечеткой базы знаний и нечетких операций [79, 104].

Входящие в структуру экспертной системы три блока нечеткого вывода образуют иерархический блок, где результаты работы первого и второго блоков не подвергаются дефаззификации, а непосредственно передаются на вход третьего блока. Применение схемы иерархического нечеткого вывода позволяет преодолеть «комбинаторный взрыв», заключающийся в том, что при увеличении количества входов база правил очень быстро разрастается, а эксперту трудно описать причинно-следственные связи в виде нечетких правил. Преимущество иерархических систем заключается в том, что они позволяют относительно небольшим количеством правил адекватно описать сложные зависимости между входными и выходными данными [75]. Допустим, что есть n входных переменных и m функций принадлежности термов для каждой переменной. Тогда для построения системы нечеткого вывода с одним блоком потребуется m^n правил. С увеличением n количество правил растет экспоненциально. Однако при организации в виде иерархической структуры количество правил растет лишь линейно, чем достигается большое удобство составления правил и экономия вычислительных ресурсов.

Рассмотрим общие принципы работы всех блоков нечеткого вывода. Каждый из блоков использует схему нечеткого вывода Мамдани. Нечеткий логический вывод по алгоритму Мамдани выполняется по нечеткой базе знаний, отражающей мнения экспертов [57, 58]. Эту базу знаний можно представить в виде совокупности предложений на естественном языке, например, одно из правил блока нечеткого вывода может выглядеть как «Если сбалансированная оценка за предыдущее занятие отличная, текущая задача сложная и уровень студента высокий, то значение степени рекомендуемости текущей задачи – высокое». В данной работе предполагается, что каждое правило вывода содержит только связки «И». Правила вывода всех блоков составлены путем опроса экспертных мнений 10 преподавателей разных дисциплин МГППУ и выбором наиболее часто встречающегося варианта заключения для каждой комбинации термов входных переменных.

Определение. *Нечеткое продукционное правило* – это упорядоченное множество объектов вида [58]:

$$(i): Q, P, A \Rightarrow B, S, F, N, \quad (2.16)$$

где i – имя нечеткой продукции, Q – сфера применения нечеткой продукции, P – условие применимости ядра нечеткой продукции, $A \Rightarrow B$ – ядро нечеткой продукции, в котором A – условие ядра (антецедент), B – заключение ядра (консеквент), $A \Rightarrow B$ – знак следования (логической секвенции), S – метод или способ определения количественного значения степени истинности заключения ядра, F – коэффициент достоверности нечеткой продукции, N – постусловия продукции.

Каждое нечеткое правило можно представить в виде нечеткого отношения. Для правила с двумя входными переменными соответствующее нечеткое отношение имеет вид [79]:

$$R_j : A_{1j} \cap A_{2j} \rightarrow B_j, \quad (2.17)$$

где A_{1j} , A_{2j} , B_j – нечеткие множества, определенные соответственно на X и Y с помощью функций принадлежности $\mu_{A_{1j}}(x)$, $\mu_{A_{2j}}(x)$, $\mu_{B_j}(x)$.

Для правила с тремя входными переменными отношение выглядит аналогично:

$$R_j : A_{1j} \cap A_{2j} \cap A_{3j} \rightarrow B_j. \quad (2.18)$$

В общем случае, для правила с n входными переменными получаем:

$$R_j : \bigcap_{i=1}^n A_{ij} \rightarrow B_j. \quad (2.19)$$

Объединив все правила и обозначив полученное отношение через $R : \bigcup_{j=1}^m R_j$, получим нечеткую базу знаний (продукционную систему) блока нечеткого вывода в виде следующего нечеткого отношения, определённого на $T_1 \times T_2 \times \dots \times T_n \times T_y$, где T_1, T_2, \dots, T_n – терм-множества входных переменных блока, T_y – терм-множество выходной переменной:

$$R : \bigcup_{j=1}^m \left[\bigcap_{i=1}^n A_{ij} \rightarrow B_j \right]. \quad (2.20)$$

Структуру базы знаний, соответствующей отношению R , можно записать в следующем виде:

$$\bigcup_{j=1}^m \left(\bigcap_{i=1}^n (x_i = a_{i,j}) \rightarrow y = b_j \right), \quad (2.21)$$

где n – количество входных переменных блока, x_i – i -я входная переменная, $a_{i,j}$ – значение i -й входной переменной, j – порядковый номер правила, m – количество правил, где $1 \leq m \leq K_1 \cdot \dots \cdot K_n \cdot K_y$ [57]. Подобные правила составляются для всевозможных комбинаций термов входных и выходной переменных для каждого блока нечеткого вывода. Например, для третьего блока нечеткого вывода, имеющего три входные

переменные, j -е правило может быть записано в виде: «Если L есть a_{1j} , C есть a_{2j} и W есть a_{3j} , то Q есть b_j ».

Основу нечеткого логического вывода составляет *композиционное правило вывода Заде*, которое формулируется следующим образом: если известно нечеткое отношение R между входной переменной x и выходной переменной y , то при нечетком значении входной переменной $x = \tilde{A}$ нечеткое значение выходной переменной определяется как $y = R \circ \tilde{A}$, где \circ – оператор максиминной композиции [75, 104].

Рассмотрим осуществление нечеткого вывода блоками экспертной системы.

Блок нечеткого вывода I уровня иерархии (блок моделирования текущей успеваемости студента) определяет общую успеваемость студента, исходя из начального уровня подготовки студента и его текущей успеваемости. Блок принимает на вход средневзвешенную сумму оценок $\langle \tilde{w} \rangle_1^{i-1}$, полученную блоком фаззификации средневзвешенной суммы оценок, и начальный уровень подготовки студента \tilde{B} , полученный на выходе блока фаззификации начального уровня подготовки. Выход блока – вектор \tilde{L} , представляющий значение лингвистической переменной L «текущая успеваемость студента». Правила вывода блока приведены в табл. 4.

Таблица 4. Правила вывода блока моделирования текущей успеваемости студента

Значения переменной «Текущая успеваемость студента»					
Начальный уровень студента	Взвешенная средняя оценка				
	Отлично	Хорошо	Удовл.	Неудовл.	Отсутствует
Сильный	Высокая	Средняя	Средняя	Низкая	Высокая
Средний	Высокая	Средняя	Низкая	Низкая	Средняя
Слабый	Средняя	Средняя	Низкая	Низкая	Низкая

Приведем формальное описание правил первого блока в соответствии с общим видом (2.16). Здесь:

i (имя нечеткой продукции) – порядковый номер нечеткой продукции;

Q (сфера применения нечеткой продукции) – заключение об успеваемости студента на основе информации об оценках, полученных им на вступительных экзаменах, дополнительной информации и (при наличии) о его текущей успеваемости по дисциплине;

R (условие применимости ядра нечеткой продукции) – определяется следующим образом. Часть правил, имеющих в antecedенте по два элементарных логических высказывания, применима, если обе входные переменные \tilde{B} и $\langle \tilde{w} \rangle$ имеют ненулевые

значения. Если же $\langle \tilde{w} \rangle$ пустое, что соответствует последнему столбцу в табл. 1, то применима часть правил, имеющих одно элементарное высказывание;

$A \Rightarrow B$ (ядро нечеткой продукции) – выражения вида «Если текущая успеваемость студента высокая и начальный уровень средний, то успеваемость студента по данной дисциплине – высокая»;

S (метод определения степени истинности заключения ядра) – операция нахождения минимума;

F (коэффициент достоверности нечеткой продукции) – принят равным единице для каждой продукции;

N (постусловия продукции) – отсутствуют.

Блок нечеткого вывода II уровня иерархии (блок моделирования сбалансированной оценки за предыдущее занятие) вычисляет сбалансированную оценку предыдущего занятия \tilde{w}_{i-1}^C , т.е. оценку, скорректированную с учетом уровня сложности этого занятия для данного студента. Сбалансированная оценка позволяет компенсировать неоднородность шкалы оценок, заключающуюся в том, что фактическая величина оценки на практике зависит не только от уровня подготовки студента, но и от сложности занятия для данного студента [19]. Например, оценка в 60% за сложное занятие для слабого студента может быть очень хорошим результатом, тогда как та же оценка за простое занятие для того же студента – результат довольно средний.

Блок принимает на вход переменные: \tilde{C}_{i-1} – сложность пройденного уровня $(i-1)$ -го занятия для данного студента; \tilde{w}_{i-1} – вектор, соответствующего степеням принадлежности оценки за предыдущее пройденное занятие термам лингвистической переменной W «оценка за занятие». Выход блока – вектор \tilde{w}_{i-1}^C , представляющий степени принадлежности сбалансированной оценки термам лингвистической переменной W «оценка за занятие». Правила вывода блока II-го уровня приведены в табл. 5.

Таблица 5. Правила вывода блока моделирования сбалансированной оценки за предыдущее занятие

Значения переменной «Сбалансированная оценка»			
Оценка за занятие	Сложность занятия		
	Высокая	Средняя	Низкая
Отлично	Отлично	Отлично	Хорошо
Хорошо	Хорошо	Хорошо	Удовл.
Удовлетворительно	Хорошо	Удовл.	Удовл.
Неудовлетворительно	Удовл.	Неудовл.	Неудовл.

В случае отсутствия оценки за предыдущее занятие блок исключается из работы экспертной системы.

Приведем формальное описание правил второго блока в соответствии с (2.16):

Q (сфера применения нечеткой продукции) – вычисление оценки студента, скорректированной с учетом его успеваемости по данной дисциплине;

P (условие применимости ядра нечеткой продукции): все правила применимы, если обе входные переменные \tilde{w}_{i-1} , \tilde{C}_{i-1} непустые. Если текущее занятие является первым в дисциплине, то обе входные переменные становятся пустыми, и второй блок нечеткого вывода полностью исключается из работы, так как все его правила становятся неприменимы;

$A \Rightarrow B$ (ядро нечеткой продукции) – выражения вида «Если оценка за предыдущее занятие удовлетворительная и уровень студента низкий, то сбалансированная оценка – хорошая»;

остальные значения – i, S, F, N – аналогичны правилам первого блока.

Блок нечеткого вывода III уровня иерархии (блок вычисления степени рекомендуемости) вычисляет функцию принадлежности выходной нечеткой переменной «рекомендуемость занятия» $\mu_{Q,i,j}(x)$. Блок принимает на вход переменные: \tilde{C}_i – оценка сложности j -го уровня i -го (текущего) занятия для данного студента; \tilde{L} – текущий уровень подготовки студента; \tilde{w}_{i-1}^C – сбалансированная оценка за предыдущее занятие (при наличии). Выход блока представляет собой агрегированную функцию принадлежности $\mu_{Q,i,j}(x)$.

Формальное описание правил третьего блока в соответствии с (2.16):

Q (сфера применения нечеткой продукции) – вычисление степени рекомендуемости данного уровня данного занятия для данного студента;

P (условие применимости ядра нечеткой продукции): правила, имеющие в antecedенте по три элементарных логических высказывания, применимы, если все входные переменные \tilde{w}_{i-1}^C , \tilde{L} и \tilde{C}_i имеют ненулевые значения. Если \tilde{w}_{i-1}^C пустое, то применима часть правил, имеющих два элементарных высказывания;

$A \Rightarrow B$ (ядро нечеткой продукции) – выражения вида «Если сложность занятия для данного студента высокая, уровень студента средний, сбалансированная оценка за предыдущее занятие низкая, то рекомендуемость занятия – низкая»;

остальные значения – i, S, F, N – аналогичны соответствующим значениям продукций первого блока.

Правила вывода блока III уровня иерархии представлены в таблице 6.

Таблица 6. Правила вывода блока вычисления степени рекомендуемости

Значения переменной «Степень рекомендуемости»			
Сбалансированная оценка за предыдущее занятие – «Отлично»			
Успеваемость	Сложность		
	Высокая	Средняя	Низкая
Сильный	Очень высокая	Высокая	Очень низкая
Средний	Высокая	Высокая	Низкая
Слабый	Средняя	Очень высокая	Высокая
Сбалансированная оценка за предыдущее занятие – «Хорошо»			
Сильный	Высокая	Средняя	Низкая
Средний	Средняя	Очень высокая	Средняя
Слабый	Очень низкая	Средняя	Высокая
Сбалансированная оценка за предыдущее занятие – «Удовлетворительно»			
Сильный	Низкая	Средняя	Высокая
Средний	Очень низкая	Средняя	Высокая
Слабый	Очень низкая	Низкая	Очень высокая
Сбалансированная оценка за предыдущее занятие – «Неудовлетворительно»			
Сильный	Очень низкая	Средняя	Высокая
Средний	Низкая	Средняя	Высокая
Слабый	Очень низкая	Низкая	Очень высокая

Сбалансированная оценка за предыдущее занятие отсутствует			
Сильный	Высокая	Средняя	Низкая
Средний	Средняя	Высокая	Средняя
Слабый	Низкая	Средняя	Высокая

2.4. Этапы нечеткого вывода в иерархической нечеткой экспертной системе

Рассмотрим процедуру нечеткого вывода в соответствии со структурой базы знаний экспертной системы. База знаний каждого блока нечеткого вывода представляется в соответствии с (2.21) в виде набора продукций вида

$$R_l: \text{Если } x_1 \text{ есть } A_{l1} \text{ и } x_2 \text{ есть } A_{l2} \text{ и } \dots \text{ и } x_p \text{ есть } A_{lp}, \text{ то } y \text{ есть } B_l^m, l=1, \dots, L,$$

где L – количество правил вывода блока, x_1, \dots, x_p – входные лингвистические переменные, A_{l1}, \dots, A_{lp} – термы входных переменных, y – выходная лингвистическая переменная, B_l^m ($m=1, \dots, K_y$) – m -й терм выходной переменной, находящийся в заключении l -го правила, K_y – количество термов выходной переменной, p – количество входных переменных блока.

Нечёткий вывод с использованием указанной структуры базы знаний включает следующие этапы [58, 79].

Первый этап – конъюнкция – нахождение степени истинности антецедента ядра каждого правила при заданном значении входной переменной путем объединения посылок в антецеденте каждого правила с использованием t -нормы, в качестве которой используется операция вычисления минимума:

$$\alpha_l = \min_{k=1, \dots, p} A_{lk}, l=1, \dots, L. \quad (2.22)$$

Второй этап – объединение – это процедура нахождения степени истинности каждого консеквента ядер всех нечетких продукционных правил. Обозначим через $\{l_m\}$ множество номеров правил вывода, содержащих в заключении m -й терм выходной переменной B_l^n . Тогда в блоках I и II уровня иерархии m -я компонента \tilde{y}_m вектора выходной переменной \tilde{y} , соответствующая степени принадлежности выходной переменной m -му терму, будет определяться по формуле

$$\tilde{y}_m = \max_{l \in \{l_m\}} \{\alpha_l\} = \max_{l \in \{l_m\}} \left\{ \min_{k=1, \dots, p} A_{lk} \right\}, \quad m = 1, \dots, K_y. \quad (2.23)$$

Замечание. Если m -й терм выходной переменной отсутствует в правилах вывода, то полагается $\tilde{y}_m = 0$.

Для I и II блоков на этом этапе алгоритм работы заканчивается, и вектор выходной переменной \tilde{y} передается на следующий уровень иерархии вывода. На выходе III блока требуется получить единую функцию принадлежности выходной лингвистической переменной, поэтому в нем предусмотрены еще два этапа – активизация подзаключений и агрегация.

Пусть $\mu_{Q_m}(x)$ – функция принадлежности m -го термина выходной переменной Q . Активизация подзаключений представляет собой построение усеченных функций принадлежности $\mu_{Q_m, i, j}^*(x)$ для j -го уровня i -го занятия с уровнем отсечения \tilde{y}_m :

$$\mu_{Q_m, i, j}^*(x) = \min \left\{ \mu_{Q_m}(x), \tilde{y}_m \right\}, \quad m = 1, \dots, K_y. \quad (2.24)$$

Агрегация – это этап получения нечеткого множества, представляющего значение выходной переменной. Все нечеткие подмножества, назначенные к терминам выходной переменной во всех правилах, объединяются, чтобы формировать одно нечеткое подмножество:

$$\mu_{Q, i, j}(x) = \max_{m=1, \dots, K_y} \mu_{Q_m, i, j}^*(x). \quad (2.25)$$

Функция $\mu_{Q, i, j}(x)$ и является результатом работы III блока иерархического нечеткого вывода.

Последним этапом нечеткого вывода является дефаззификация [79]. Блок дефаззификации вычисляет окончательное числовое значение степени рекомендуемости путем поиска центра тяжести фигуры, образуемой графиком функции принадлежности $\mu_{Q, i, j}(x)$ [57]:

$$Q_i(j) = \frac{\int_0^1 x \mu_{Q, i, j}(x) dx}{\int_0^1 \mu_{Q, i, j}(x) dx}. \quad (2.26)$$

На рис. 18 приведена иллюстрация выбранного метода дефаззификации.

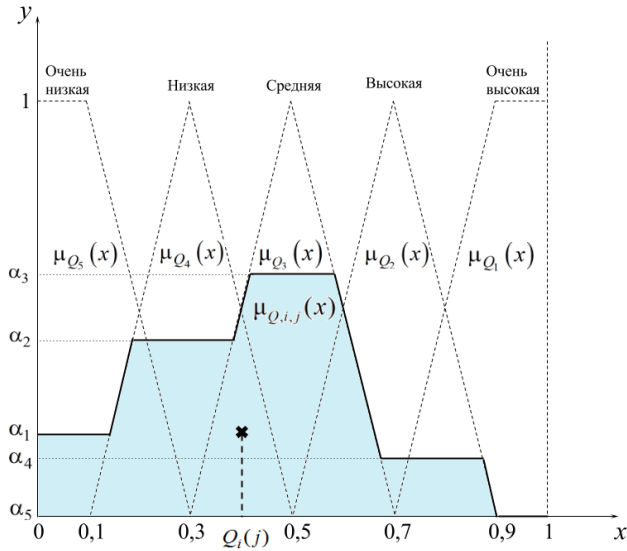


Рис. 18. Дефаззификация выходной переменной

Интегралы в формуле (2.26) находятся численно с использованием метода трапеций, который состоит в следующем [73]. Рассмотрим определенный интеграл $I = \int_a^b f(x)dx$, где $f(x)$ – функция, непрерывная на отрезке $[a, b]$. Построим разбиение отрезка $[a, b]$ на n равных отрезков: $[x_0, x_1]$, $[x_1, x_2]$, $[x_2, x_3]$, ..., $[x_{n-1}, x_n]$, где $x_0 = a$, $x_n = b$. Величина $h = \frac{b-a}{n}$ представляет собой длину каждого отрезка разбиения. Тогда определенный интеграл можно приближенно вычислить по формуле:

$$I = \int_a^b f(x)dx \approx h \cdot \left(\frac{f(x_0) + f(x_n)}{2} + f(x_1) + f(x_2) + \dots + f(x_{n-1}) \right). \quad (2.27)$$

Количество интервалов n определяется при помощи правила Рунге, которое состоит в следующем. Обозначим через I_n и I_{2n} вычисленное значение интеграла для разбиения отрезка $[a, b]$ на n и $2n$ отрезков соответственно. Тогда погрешность R_{2n} вычисления I_{2n} можно оценить по формуле: $R_{2n} \approx \frac{I_{2n} - I_n}{2^p - 1}$, где p – порядок точности выбранного метода вычисления интеграла (для метода трапеций $p = 2$). Для практического вычисления интеграла I с заданной точностью ε выбираются разбиения отрезка $[a, b]$ на n и $2n$ частей и вычисляются величины I_n и I_{2n} . Если $|R_{2n}| \leq \varepsilon$, то с точностью ε полагают $I \approx I_{2n}$. В противном случае строится разбиение на $4n$ отрезков, вычисляется значение I_{4n} и

проверяется условие $|R_{4n}| \leq \varepsilon$, и т.д. Для вычисления интегралов в формуле (2.26) на отрезке $[0, 1]$ вполне достаточно положить $\varepsilon = 0,0001$. Исходя из этого значения, программа автоматически рассчитывает необходимое количество отрезков разбиения и вычисляет приближенное значение Q по формуле (2.27).

После нахождения искомой степени рекомендуемости $Q_i(j)$ для всех $j = 1, \dots, M_i$ происходит последний этап работы экспертной системы – выбор номера уровня сложности с максимальным значением степени рекомендуемости в соответствии с (2.6): $J_i = \arg \max_j Q_i(j)$. Расчет по приведенной выше методике производится для каждого студента, и в соответствии с результатами расчетов составляется расписание, в котором каждому студенту назначен наиболее рекомендуемый ему уровень J_i . Это назначение не является окончательным, а выступает в роли рекомендации преподавателю, который может либо согласиться с ней, либо назначить студенту другой уровень по своему усмотрению (см. рис. 39).

2.5. Модель оценивания результатов решения учебных заданий

Для упрощения работы преподавателя предусмотрено автоматическое оценивание результатов студентов, заключающееся в расчете оценки w_i исходя из количества ошибок, допущенных студентами при решении задачи. Пошаговое обучение решению задач, применяемое в рассматриваемой СДО, предусматривает возможность проверки каждого ответа преподавателем и исправление студентами своих ошибок. Также возможен вариант, когда студент после нескольких попыток так и не смог ввести правильный ответ на текущий шаг задачи. В этом случае во избежание траты большого количества времени преподаватель переходит к следующему шагу, а студенту показывается правильный ответ. В таком случае шаг задания считается нерешенным. Модель вычисления оценки за задание, адекватно отражающая уровень усвоения студентом материала, должна быть основана на предположении, что при отсутствии ошибок в решении значение оценки максимально, а с увеличением количества ошибок или нерешенных шагов оценка монотонно убывает.

Пусть n – количество шагов учебного задания, а $t \in [0, n]$ – количество нерешенных шагов, то есть шагов, по которым не был дан правильный ответ и которые были пропущены данным студентом. Обозначим через p суммарное количество ошибок во всех

решенных шагах для всех $k=1, \dots, n$. Оценка за задание представляется величиной $w(p, t) \in [0, 100\%]$, причем положительной считается оценка не менее 60%.

Рассмотрим случай, когда нерешенные шаги отсутствуют ($t=0$). В этом случае задание считается решенным, и оценка должна быть положительной (60% или более). Величина оценки в этом случае определяется исходя только из числа ошибочных элементов: с ростом p оценка монотонно убывает, но остается выше 60% при любом p . Эту зависимость можно представить экспоненциальной функцией (рис. 19):

$$w(p, t)|_{t=0} = (0,6 + 0,4e^{-\gamma p}) \cdot 100\%, \quad (2.28)$$

где p принимает неотрицательные целочисленные значения, а $\gamma > 0$ – параметр, задающий скорость убывания оценки при росте p . На практике можно принять значение $\gamma = 0,05 \dots 0,3$. В рассматриваемой СДО используется значение $\gamma = 0,1$.

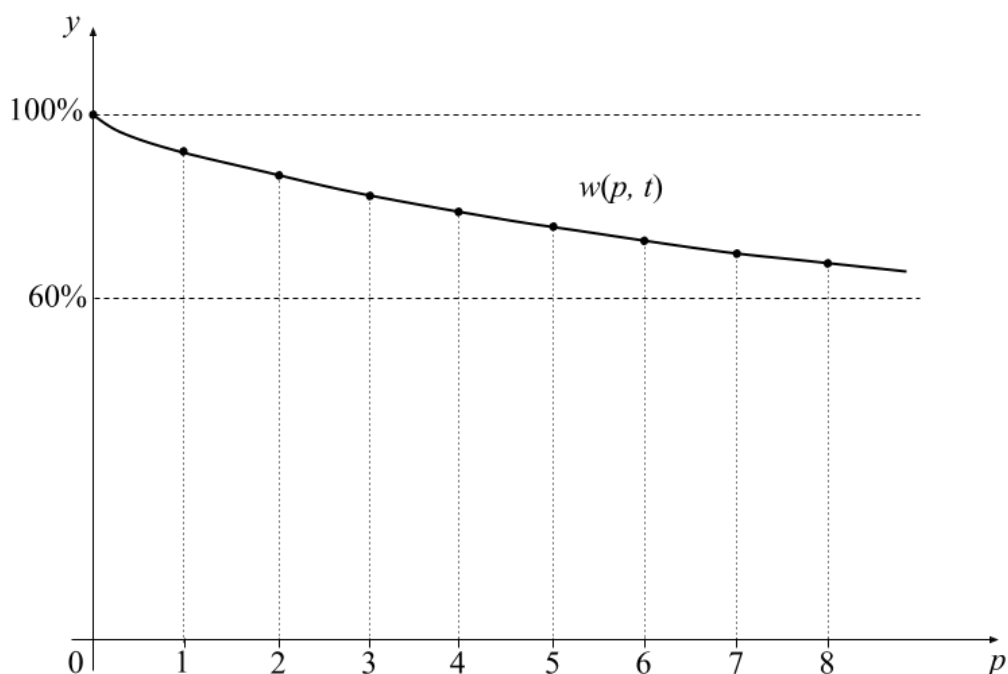


Рис. 19. Зависимость значения оценки от количества ошибочных элементов при отсутствии нерешенных шагов

В случае наличия нерешенных шагов оценка определяется исходя как из количества нерешенных шагов t , так и из количества ошибок p , причем в этом случае оценка может принимать любые значения из отрезка $[0, 1]$. Рассмотрим сначала случай, когда $p=0$. Тогда при $t=0$ (все шаги решены без ошибок) оценка должна быть равна $w(0, t)=100\%$, а при $t=n$ (ни один шаг не решен) оценка должна составлять $w(0, n)=0$. Следовательно, оценку $w = w(p, t)$ при $p=0$ можно представить следующей линейной зависимостью:

$$w(p, t)|_{p=0} = \frac{n-t}{n} \cdot 100\% \quad (2.29)$$

(рис. 20, верхний график).

Рассмотрим общий случай, когда $t \neq 0$ и $p \neq 0$. В этом случае с ростом количества ошибок p оценка будет снижаться, поэтому график зависимости $w(p, t)$ будет проходить ниже графика $w(p, t)|_{p=0}$ таким образом, чтобы при $t=0$ значение оценки совпадало с полученной ранее зависимостью для $w(p, t)|_{t=0}$ (рис. 19). При $p \rightarrow \infty$ график асимптотически приближается к предельному положению, соответствующему отрезку, проходящему через точки $(0, 60\%)$ и $(n, 0)$ (рис. 20, нижний график).

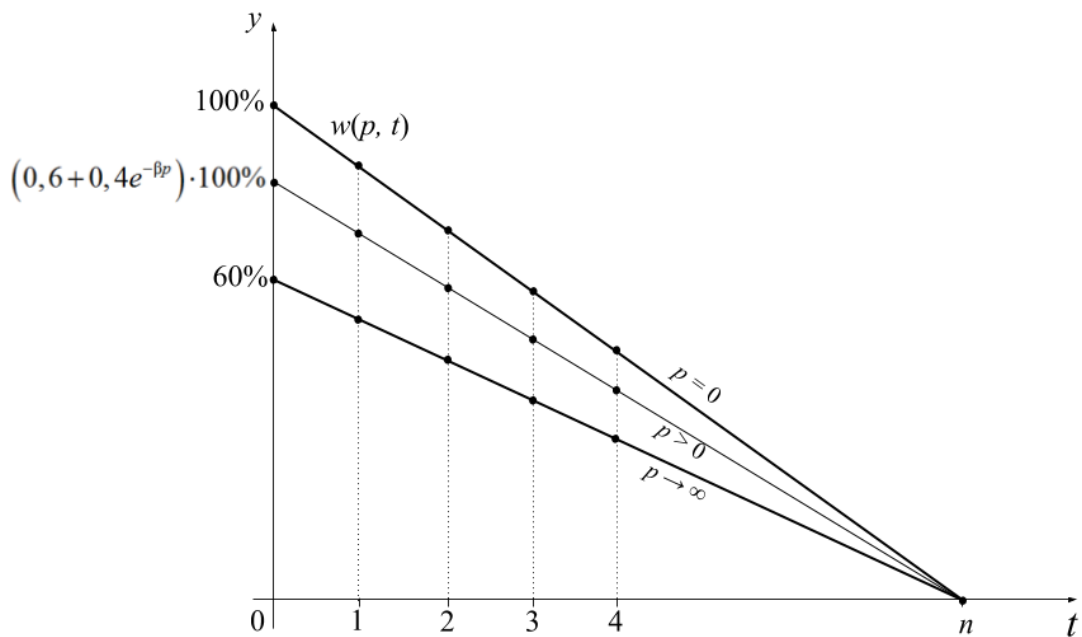


Рис. 20. Зависимость значения оценки от количества нерешенных шагов

Следовательно, из (2.28) и (2.29) получаем окончательную формулу для расчета оценки за задание:

$$w(p, t) = \frac{n-t}{n} (0,6 + 0,4e^{-6p}) \cdot 100\% . \quad (2.30)$$

Полученная оценка записывается в базу данных и в дальнейшем используется для анализа успеваемости и индивидуализации дальнейшего процесса обучения каждого студента. После сохранения оценки происходит перерасчет текущей оценки студента за занятие, тему и дисциплину.

Оценка за занятие вычисляется как среднее арифметическое оценок, полученных студентом за все задания, предусмотренные в занятии, с округлением до целого числа процентов:

$$w_i = \frac{1}{m_i} \sum_{k=1}^{m_i} w_{ik}, \quad (2.31)$$

где w_{ik} – оценка за k -е задание i -го занятия, m_i – количество учебных заданий i -го занятия.

Аналогично вычисляются текущие оценки студента за тему и дисциплину как средние оценки за пройденные занятия, входящие в тему, и пройденные темы, входящие в дисциплину.

2.6. Моделирование индивидуализации процесса обучения с использованием экспертной системы

2.6.1. Условия моделирования

Рассмотрим модель цикла изучения дисциплины для одного студента с формированием траектории при помощи указанного программного комплекса. Результаты моделирования опубликованы в работе [15].

Процесс обучения можно представить в виде блок-схемы (рис. 21). Предполагается, что преподаватель следует рекомендациям экспертной системы, каждый раз выбирая рекомендуемый уровень сложности, поэтому назначаемый студенту уровень выбирается автоматически из условия (2.6): $J_i = \arg \max_j Q_i(j)$.

После прохождения каждого занятия студенту ставится оценка w_i , которая записывается в базу данных, пополняя историю оценок и выступая в роли оценки за предыдущее занятие при расчёте рекомендуемого уровня сложности следующего занятия. Неудовлетворительной считается оценка ниже заданного порога, в качестве которого выбрано значение 60%. В случае получения неудовлетворительной оценки по какому-либо занятию студенту назначается повторное прохождение занятия, а требуемый уровень сложности вновь определяется экспертной системой с учётом последней полученной оценки.

В случае успешного прохождения занятия, если занятие не было последним ($i < N$), происходит переход к расчёту рекомендаций для $(i+1)$ -го занятия. Обучение считается законченным, когда выполняется условие $i = N$, то есть студент получил положительную оценку за все занятия курса.

2.6.2. Результаты моделирования процесса обучения

В табл. 7 приведены результаты моделирования описанного цикла обучения для одного студента. Моделируемый курс состоит из 7 занятий, для каждого из которых задано 4 уровня сложности заданий. Уровни пронумерованы в порядке уменьшения сложности. Пример оценок сложности для одного из занятий приведён на рис. 2, для остальных занятий оценки сложности выглядят аналогично.

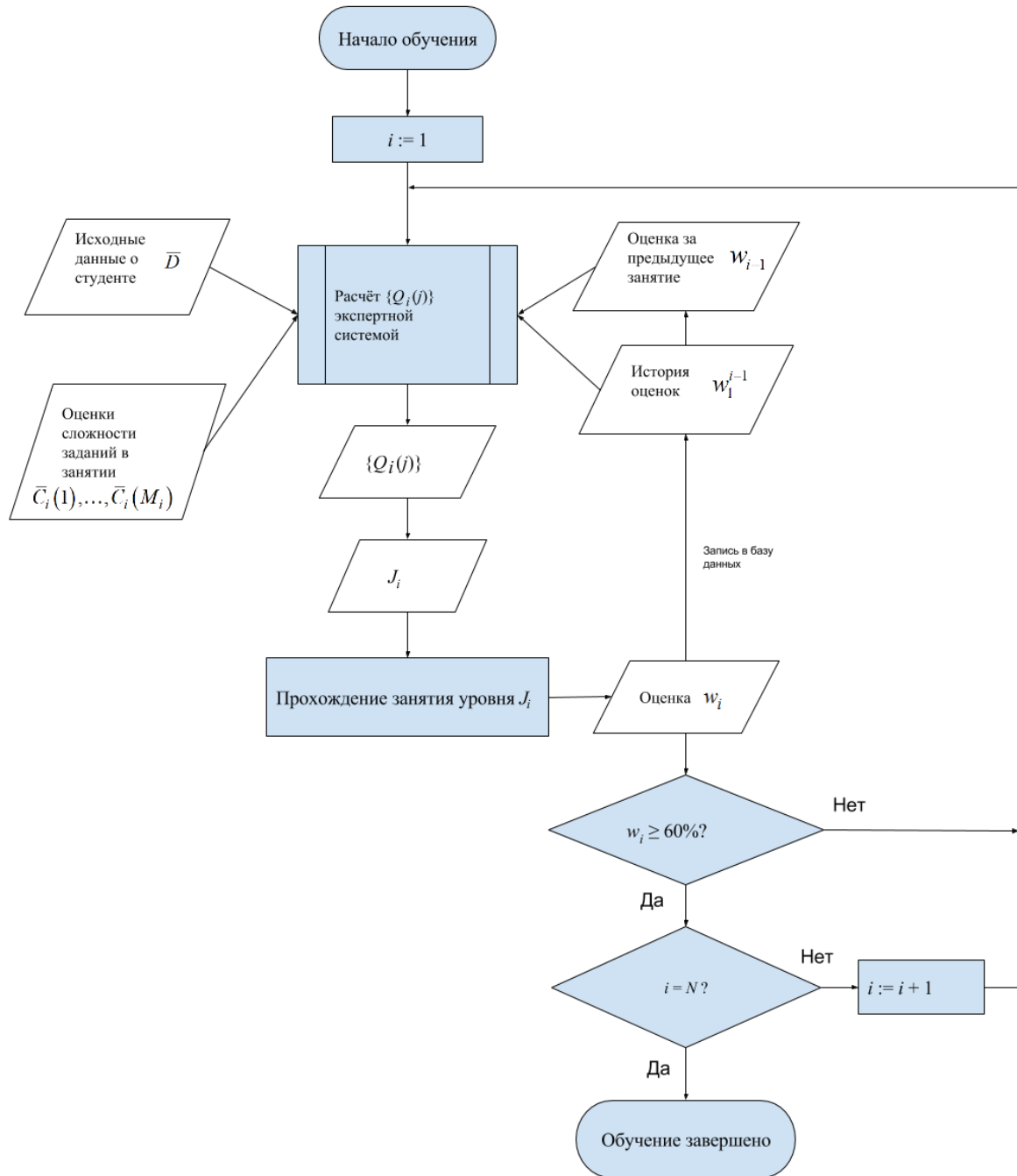


Рис. 21. Схема цикла изучения дисциплины при моделировании процесса обучения

В таблице приведены степени рекомендуемости для каждого уровня сложности каждого занятия и выделен уровень с наибольшей степенью рекомендуемости. При совпадении степени рекомендуемости для нескольких уровней выбирается уровень с наибольшей сложностью (наименьшим номером). В последнем столбце приведена последовательность оценок w_i , использованных для моделирования процесса обучения.

Таблица 7. Результаты моделирования цикла обучения для одного студента

Номер попытки	Номер занятия	Степень рекомендуемости уровней сложности				Рекомендуемый уровень	Оценка за занятие
		1	2	3	4		
1	1	0,525	0,575	0,492	0,474	2	100
2	2	0.788	0.747	0.593	0.470	1	100
3	3	0.842	0.749	0.397	0.158	1	45
4	3 (повтор)	0.580	0.609	0.503	0.419	2	90
5	4	0.595	0.614	0.571	0.417	2	95
6	5	0.832	0.701	0.627	0.167	1	40
7	5(повтор)	0.474	0.713	0.761	0.512	3	30
8	5(повтор)	0.405	0.580	0.590	0.601	4	70
9	6	0.434	0.573	0.571	0.546	2	80
10	7	0.614	0.622	0.488	0.446	2	100

На рис. 22 приведена визуализация полученной траектории. Прямоугольниками выделено повторное прохождение одного и того же занятия после получения неудовлетворительной оценки.

Как видно из графика, вначале экспертная система рекомендовала начать обучение со второго уровня сложности ввиду хороших, но не самых высоких входных результатов студента. После получения высокой оценки (100 из 100) система предложила решать задачи первого, самого сложного уровня.

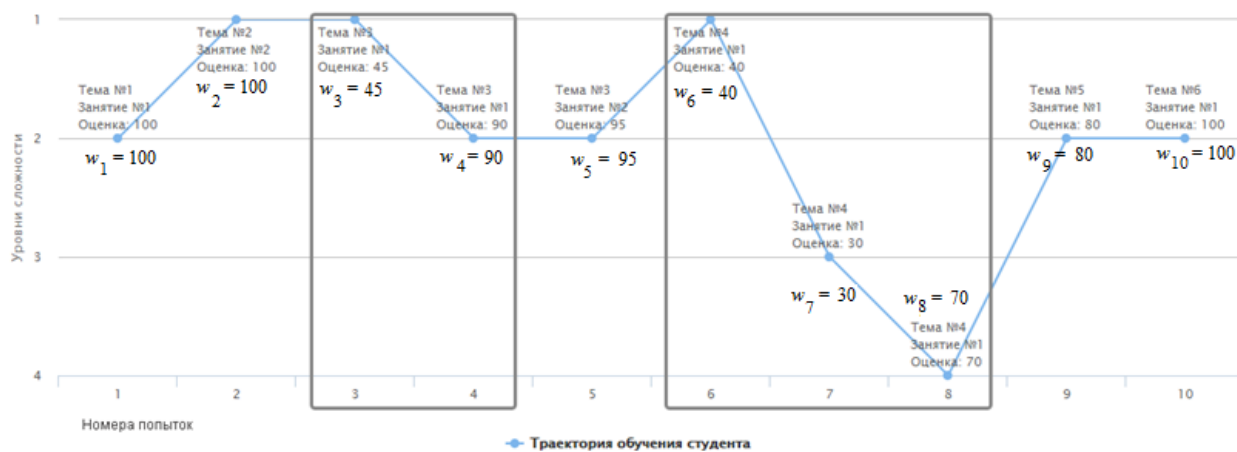


Рис. 22. Пример траектории обучения студента

После получения довольно низкой оценки (45/100) за третье занятие студенту предложено пройти второй, более простой уровень того же занятия. Далее после получения высоких оценок (90% и 95%) траектория возвращается к первому уровню. После получения очень низких оценок за пятое занятие (40% и 30%) траектория спускается до самого простого, четвертого уровня. Далее, после получения положительных оценок (70%, 80%) траектория вновь поднимается до второго уровня, но не до первого, так как сказывается недавняя история низких оценок за 5-е занятие.

Выводы

В третьей главе описано математическое обеспечение информационной системы, осуществляющей поддержку учебного процесса в форме веб-конференции. Эта система представляет собой экспертную систему, опирающуюся на экспертные знания преподавателей. Описанная система использует метод иерархического нечеткого вывода. Также в третьей главе описано построение системы вывода на основе нейронной сети, используемой для решения задачи нечеткой классификации, заключающейся в вычислении степени принадлежности входного вектора к априорно определенным нечетким множествам. Приведен процесс экспериментального выбора параметров нейронной сети. Представлены результаты моделирования работы экспертной системы. Результаты моделирования согласуются с ожидаемой реакцией экспертной системы на изменения в текущей успеваемости студента.

Описанная в главе экспертная система позволяет выстроить учебный процесс для каждого студента с учётом его начальных данных об успеваемости на момент поступления и текущей успеваемости, определяемой из истории оценок, полученных студентом по данной дисциплине. Использование данной системы помогает рационально

использовать временные ресурсы при планировании занятий, что существенно для онлайн-систем дистанционного обучения.

Глава 3. Технологии разработки, структура и принципы функционирования программного комплекса

3.1. Структура программного комплекса

Программный комплекс системы дистанционного обучения в режиме веб-конференций MathConference построен в соответствии с клиент-серверной архитектурой, где в качестве клиента выступает компьютер конечного пользователя (преподавателя, студента или администратора), подключенный к сети Интернет, а в качестве сервера – серверная ЭВМ с установленными на ней серверными компонентами. Структурная схема комплекса показана на рис. 23 в виде диаграммы UML.

В структуре комплекса можно выделить шесть основных компонентов: клиентское веб-приложение, редактор учебных заданий, серверное веб-приложение, мультимедийный сервер, HTTP-сервер и система управления базой данных (СУБД). Серверное приложение в свою очередь делится на два компонента: серверная часть (back-end) и веб-интерфейс (front-end). Клиентское приложение и редактор учебных заданий представляют собой «богатые» Интернет-приложения (Rich Internet application, RIA), встроенные в веб-интерфейс [85]. Данная структура опубликована в работах [14, 15].

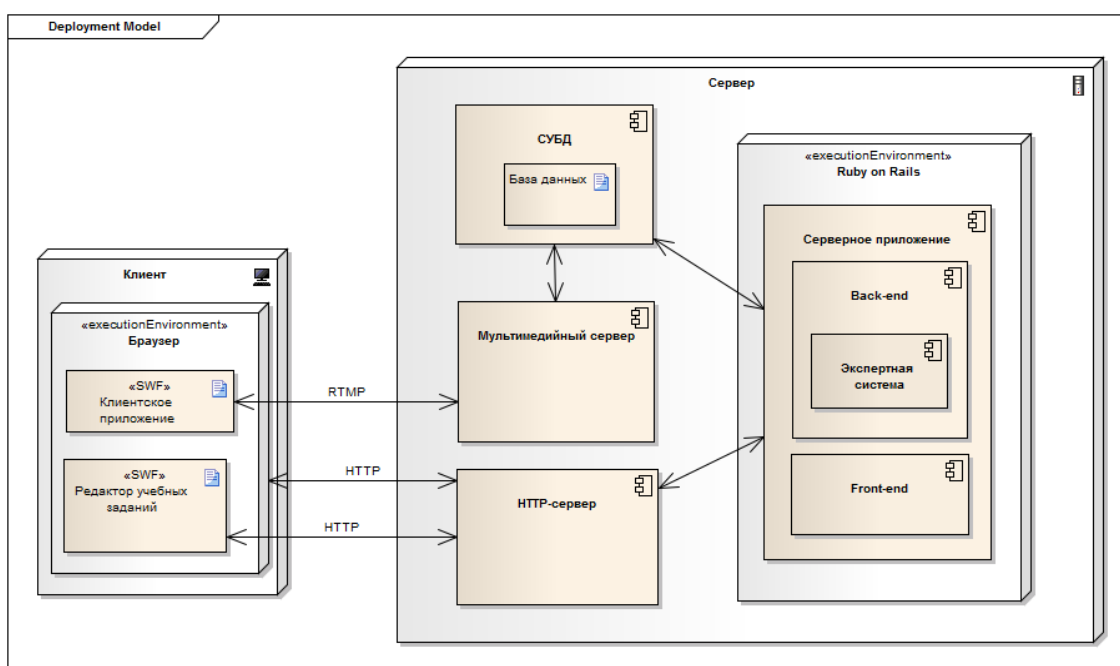


Рис. 23. Структурная схема программного комплекса

Пользователь системы дистанционного обучения (клиент) с помощью браузера входит в веб-интерфейс. Браузер отправляет HTTP-запросы к серверу, которые принимаются HTTP-сервером и перенаправляются серверному приложению. Компонент *Back-end* серверного приложения обрабатывает запрос, производит необходимые вычисления и запросы к базе данных, после чего компонент *Front-end* формирует ответ в виде HTML-страницы. Ответ возвращается пользователю через HTTP-сервер и выводится в браузере. Редактор учебных заданий и клиентское приложение также формируют запросы к серверу, в этом случае ответ серверного приложения формируется в виде XML-документа. Для проведения веб-конференций клиентское приложение связывается с мультимедийным сервером при помощи отдельного соединения по протоколу RTMP, работающего независимо от HTTP-сервера и серверного приложения. Мультимедийный сервер, как и веб-приложение, может формировать SQL-запросы к базе данных.

В главе приведено описание технологий, использованных при разработке компонентов программной системы, и основные принципы их работы.

3.2. Технологии реализации компонентов программного комплекса

3.2.1. Серверное веб-приложение

Серверное веб-приложение отвечает за построение веб-интерфейса программного комплекса, работу с базой данных, авторизацию пользователей, управление процессом обучения при помощи встроенной экспертной системы. Приложение состоит из двух компонентов: серверная часть (back-end), работающая на сервере, и клиентская часть (front-end), загружаемая в браузер клиентского компьютера. Структура серверного приложения приведена на рис. 24.

Серверная часть включает в себя модели, контроллеры, представления, роутер, библиотеку, конфигурационные файлы.

Клиентская часть представляет собой веб-интерфейс, включающий в себя HTML-страницы, генерируемые представлениями, скрипты на языке JavaScript и файлы стилей в формате CSS. Клиентское приложение для проведения веб-конференций и редактор учебных заданий встроены в веб-интерфейс при помощи HTML-тэгов.

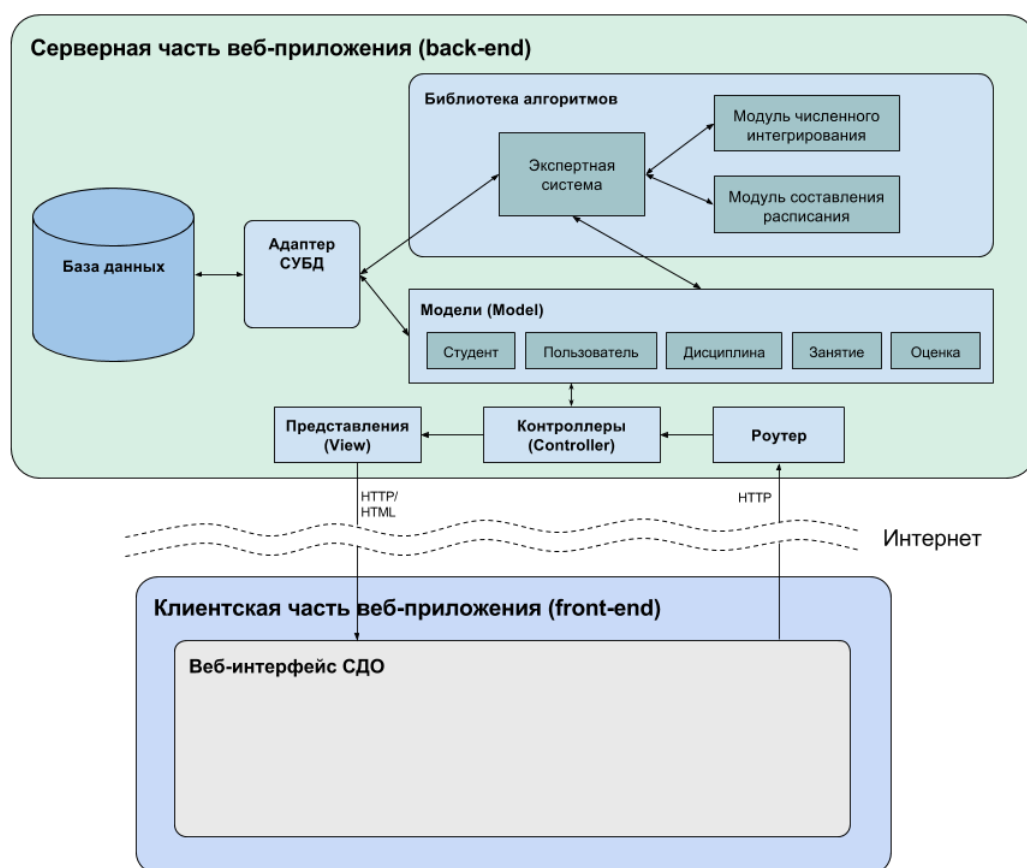


Рис. 24. Структурная схема компонентов серверного приложения

Серверная часть веб-приложения разработана на основе языка Ruby и построенного на его основе фреймворка Ruby on Rails. Ruby – это динамический объектно-ориентированный язык программирования с открытым исходным кодом. Язык обладает независимой от операционной системы реализацией многопоточности и динамической типизацией. Ruby — полностью объектно-ориентированный язык, и любые данные в нём являются объектами. Ruby поставляется с большой стандартной библиотекой для работы с различными сетевыми протоколами, форматами представления данных (в том числе XML), средствами отладки, кодировками, матрицами и т. д.

Ruby on Rails, или просто Rails – это многоуровневый фреймворк для построения веб-приложений, использующих базы данных, основанный на архитектуре Модель-Представление-Контроллер (Model-View-Controller, MVC) [87, 94, 117].

Модель – это класс, предоставляющий всем компонентам приложения объектно-ориентированное отображение данных из таблиц БД (таких как студент, дисциплина, занятие и т.д.) на объекты языка Ruby. Объекты модели могут осуществлять загрузку и сохранение данных в БД, а также реализуют логику обработки информации, относящейся к моделям (например, сохранение оценки за занятие с одновременным пересчетом

текущих результатов по теме и дисциплине). Например, моделями в рассматриваемой СДО являются следующие классы: *Студент*, *Пользователь*, *Дисциплина*, *Занятие*, *Оценка* и другие.

Контроллер — это класс, предоставляющий методы, предназначенные для обработки HTTP-запросов, полученных сервером. Контроллер отвечает за вызов методов моделей и запускает формирование соответствующего *представления*. Например, контроллер, отвечающий за работу со списком студентов, имеет имя класса *StudentController* и включает методы *create*, *show*, *update*, *destroy*, которые производят соответственно создание новой записи, просмотр страницы студента, изменение данных студента и удаление записи о студенте из БД и запускают формирование соответствующих веб-страниц, возвращаемых пользователю и содержащих информацию о результатах выполнения метода. При поступлении запроса в формате XML (например, от приложения-редактора учебных заданий) контроллер не вызывает формирование представления, а непосредственно формирует ответ в формате XML и возвращает его клиенту.

Представление обрабатывает полученные от контроллера данные и возвращает пользователю HTML-страницу с результатами обработки. В Ruby on Rails представление описывается при помощи *шаблонизаторов*. В текущей работе использован шаблонизатор *HAML*. Шаблоны *HAML* представляют собой файлы специального формата, описывающие структуру возвращаемой HTML-страницы и содержащие собственную логику формирования, описанную фрагментами кода на языке Ruby. Вывод, сгенерированный представлением, возвращается клиенту в виде HTML-страницы. Вложенность тэгов HTML моделируется отступами в шаблоне *HAML*.

Пример. Простейший шаблон *HAML*, формирующий список студентов, может выглядеть следующим образом.

```
%div.list
  %ul#students
    %li Петров Владимир
    %li Яковлев Николай
    %li Семенов Сергей
```

Результат преобразования этого шаблона в HTML:

```
<div class="list">
  <ul id="students">
```

```
<li> Петров Владимир  
<li> Яковлев Николай  
<li> Семенов Сергей  
</ul>  
</div>
```

Роутер – это специальный конфигурационный файл, в котором записаны все возможные HTTP-адреса запросов к приложению. При поступлении запроса от клиента приложение с помощью роутера находит контроллер, соответствующий URL-адресу и использованному HTTP-методу запроса (GET, POST, PUT и т.д.) и перенаправляет запрос определенному действию найденного контроллера.

Также серверное приложение содержит библиотеку алгоритмов, в которую входят экспертная система и вспомогательные алгоритмы, используемые ей – модуль численного интегрирования и модуль составления расписания. Как модели, так и алгоритмы из библиотеки алгоритмов используют адаптер объектно-реляционной модели для взаимодействия с базой данных, что позволяет работать с атрибутами таблиц как с атрибутами классов-моделей без явного использования языка запросов SQL.

3.2.2. Клиентское приложение для проведения веб-конференций и редактор учебных заданий

Клиентское приложение разработано на основе технологии Flash, реализующей все элементы мультимедиа, необходимые для проведения учебных занятий: растровую и векторную графику, звук, видео, интерактивные графические объекты. Основным достоинством Flash-технологии является её кроссплатформенность, то есть этот формат может использоваться на большинстве распространённых аппаратно-программных платформ (в частности, на компьютерах Macintosh под управлением MacOS, на компьютерах с ОС Windows и Linux). Ещё одним важным достоинством является то, что для использования приложений, разработанных при помощи данной технологии, не требуется устанавливать каких-либо программ – достаточно лишь зайти на веб-страницу, содержащую встроенное с помощью специальных тэгов HTML приложение. Для его работы необходим специальный плагин, встроенный в большинство современных браузеров и доступный для бесплатного скачивания в случае его отсутствия.

В момент входа в конференцию SWF-файл копируется с сервера на компьютер пользователя и начинает работу, устанавливая соединение с мультимедийным сервером.

Архитектура клиентского приложения показана на рис. 25.

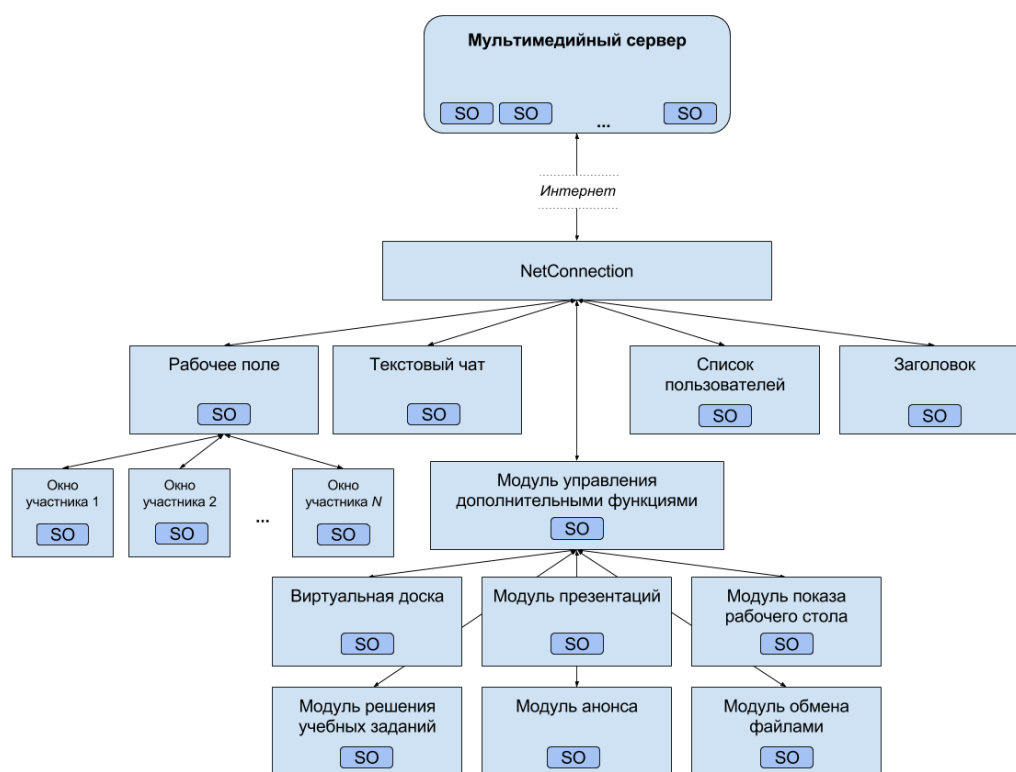


Рис. 25. Архитектура клиентского приложения для проведения веб-конференций

Приложение состоит из набора модулей, функционирующих независимо друг от друга, но использующих для связи с мультимедийным сервером один и тот же объект класса *NetConnection*, предоставляющий функции для установления и поддержания соединения. Каждый модуль отвечает за свою часть функциональных возможностей приложения, включая построение своей части графического интерфейса, обработку связанных с ней событий и действий пользователей, синхронизацию событий. Например, модуль решения учебных заданий при запуске осуществляет загрузку материалов задания с сервера в формате XML, преобразует его в графический вид в пользовательском интерфейсе, синхронизирует события смены шага задания, отправляет ответы студента, принимает результаты проверки и т.д. Модули «Рабочее поле», «Текстовый чат», «Список пользователей», «Заголовок» и «Модуль управления дополнительными функциями» функционируют постоянно, и их графические интерфейсы составляют общий интерфейс приложения (см. рис. 43). Модуль «Рабочее поле» управляет окнами участников, в которых отображаются потоки с их веб-камер. Остальные модули могут быть запущены преподавателем через модуль управления дополнительными функциями, которому соответствует панель с кнопками дополнительных функций в нижней части окна, и их интерфейс открывается в дополнительных окнах поверх основных модулей.

Каждый модуль имеет в своем составе средство синхронизации, называемое Shared Object (SO), или общий объект. Каждый общий объект в составе клиентского приложения (локальный общий объект) имеет соответствующий общий объект в памяти мультимедийного сервера (удаленный общий объект). Таким образом, все клиенты, подключенные к серверу, используют одни и те же удаленные общие объекты, что позволяет реализовывать синхронизацию событий в режиме реального времени. Подробно процесс синхронизации при помощи общих объектов рассмотрен в гл. 3.5.

Структура приложения-редактора учебных заданий в целом аналогична структуре клиентского приложения, поскольку оба приложения используют общую библиотеку классов и реализованы с использованием технологии Flash. Редактор использует общий с клиентским приложением графический компонент, отвечающий за отображение материалов учебных заданий в интерфейсе пользователя. Дополнительными компонентами приложения-редактора являются селектор заданий текущего занятия (панель в левой части интерфейса) и набор инструментов для рисования и добавления изображений из внешних графических файлов (см. рис. 26).

3.2.3. Мультимедийный сервер

Мультимедийный сервер – это компонент программного комплекса, который служит для обработки мультимедийной информации в режиме реального времени. Функции мультимедийного сервера включают приём аудио- и видеопотоков от клиентских компьютеров, передачу потоков всем подключенным к серверу клиентам, синхронизацию событий, происходящих в ходе веб-конференции [32]. Мультимедийный сервер рассматриваемого программного комплекса создан на основе проекта Red5. Это свободно распространяемый медиасервер с открытым исходным кодом, написанный на языке Java и предоставляющий базовые возможности взаимодействия клиентов, включающие передачу и приём аудио- и видеопотоков и синхронизацию событий. Более сложная логика, требуемая для работы рассматриваемого приложения (распределение ролей пользователей, учёт посещаемости, сохранение информации в базу данных и т.п.), создаётся путём расширения базового класса Red5. Благодаря кросс-платформенности технологии Java разработанный медиасервер может работать на любой серверной платформе – Linux, Unix, Windows Server.

Структура классов мультимедийного сервера приведена на рис. 27. Базовый класс *ApplicationAdapter* предоставляет набор коллбэков (callback-функций), вызываемых автоматически при наступлении определенного события. Коллбэки обрабатывают следующие события:

appStart() – запуск мультимедийного сервера;

roomStart() – запуск виртуальной аудитории, происходящий при входе первого пользователя;

serverConnect() – соединение нового клиента с сервером;

roomConnect() – вход нового клиента в виртуальную аудиторию;

roomDisconnect() – выход клиента из виртуальной аудитории;

roomStop() – выход всех клиентов и закрытие виртуальной аудитории.

Класс *confAS3* – это основной класс, реализующий функциональность, требуемую для работы программного комплекса. Он наследует функциональность базового класса *ApplicationAdapter* и переопределяет коллбэки *roomStart()*, *roomConnect()* и *roomDisconnect()*, добавляя в эти функции регистрацию посещений конференции в базу данных. Также класс *confAS3* имеет собственный набор методов и атрибутов, используемых для работы веб-конференции.

Класс *confAS3* содержит вложенные классы *studentData*, *userData*, *windowData*, используемые как структуры данных для хранения о каждом студенте, пользователе и окне пользователя соответственно. Наиболее важные методы класса *confAS3* включают: *chatMessage()* – метод для передачи сообщений в текстовый чат; *recalculateSpaces()* – метод для расчета местоположения новых окон на рабочем поле во избежание их наложения; *getHistory()* – получение истории сообщений чата для новых пользователей, входящих в конференцию; *shareFile()* – метод, используемый модулем обмена файлами; *DBSelect()* и *DBUpdate()* – методы для создания запросов к базе данных.

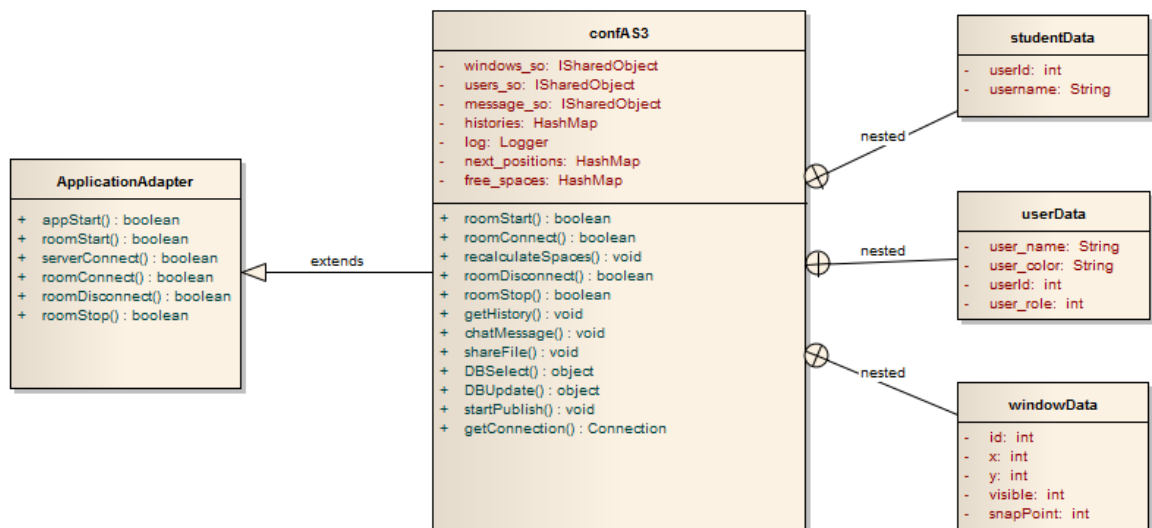


Рис. 27. Структура классов мультимедийного сервера

Мультимедийный сервер предоставляет средство событийного взаимодействия между клиентами и сервером в режиме реального времени, называемое общими объектами

(shared objects). Применение этой технологии позволяет синхронизировать события и таким образом разрабатывать разнообразные режимы взаимодействия клиентских приложений в режиме реального времени [107].

Каждый локальный общий объект на компьютере клиента имеет соответствующий удаленный общий объект в памяти мультимедийного сервера. Когда один клиент изменяет свойство локального общего объекта, то данное свойство меняется для всех подключенных клиентов посредством удаленного общего объекта. Каждый удаленный общий объект имеет атрибут *data*, в котором хранятся данные, подлежащие синхронизации у всех клиентов, использующих данный общий объект.

Синхронизация при помощи общих объектов происходит следующим образом (рис. 28).

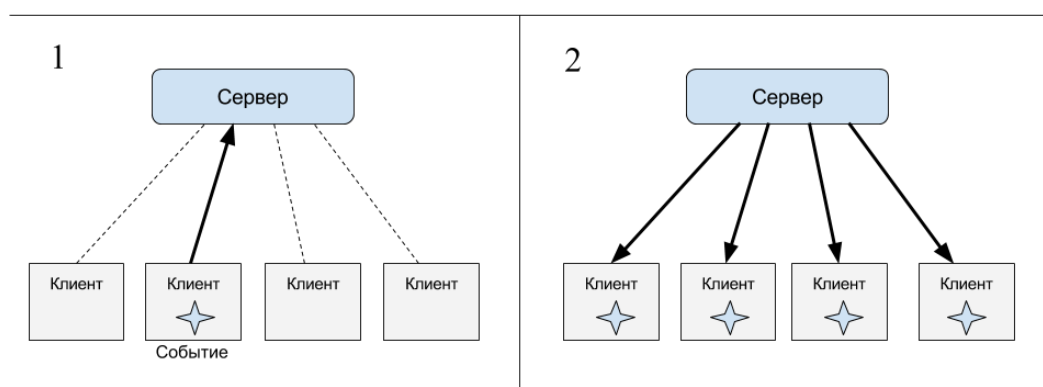


Рис. 28. Схема процесса синхронизации событий между клиентами

1. Как только у какого-либо клиента происходит событие, подлежащее синхронизации, у соответствующего локального общего объекта происходит изменение атрибута. Локальный объект посылает сообщение удаленному общему объекту об изменении атрибута.

2. Удаленный общий объект изменяет свое значение этого же атрибута и рассылает сообщение об изменении всем подключенным клиентам. Локальные общие объекты всех подключенных клиентов изменяют значение этого атрибута и запускают функцию-обработчик, производящую необходимое действие.

Пример. Рассмотрим процесс синхронной смены слайдов презентации.

Пусть текущий слайд презентации – первый. Тогда локальный общий объект на всех клиентских компьютерах хранит данные с атрибутом `{data: {slideNumber: 1}}`.

1. Преподаватель нажимает кнопку смены слайда, при этом обработчик нажатия кнопки инкрементирует атрибут локального общего объекта, отвечающий за номер слайда, после чего его новое значение будет иметь вид `{data: {slideNumber: 2}}`.

Локальный объект посылает сообщение удаленному общему объекту с информацией об измененном состоянии.

2. Удаленный общий объект изменяет свое значение атрибута *slideNumber* и рассылает клиентам сообщение следующей структуры:

```
Event: {
  type: change;
  name: slideNumber;
  value: 2;
  oldValue: 1
}
```

Локальные общие объекты на компьютерах всех студентов получают сообщение, модифицируют атрибут *data* и запускают обработчик *soChanged()*, который получает измененное значение атрибута *slideNumber* и открывает слайд презентации с указанным номером.

При необходимости сообщения об изменении состояния общего объекта может инициировать сам мультимедийный сервер, без участия клиента-инициатора. Примером такого сообщения может служить рассылаемое всем сообщение о появлении нового участника при его входе в конференцию. Приняв это сообщение, общие объекты на клиентских компьютерах запускают обработчик, который добавляет новое окно участника на рабочее поле.

3.3. Структура базы данных

Логическая структура базы данных программного комплекса приведена на рис. 29. Показаны наиболее важные сущности и атрибуты, используемые при создании структуры учебных материалов, составлении расписания при помощи экспертной системы, прохождении занятий.

Иерархическая структура учебных материалов отображена на сущности *Дисциплина* (*Discipline*), *Тема* (*Topic*), *Занятие* (*Lesson*) и *Задание* (*Task*) с отношением «один-ко-многим» между ними. Данные о студентах хранятся в таблице *Student*, а оценки студентов, полученные ими за занятия – в таблицах *DisciplineResult*, *TopicResult*, *LessonResult* и *TaskResult*. Сущность *LessonComplexity* представляет собой экспертные оценки сложности занятий. В таблице *StudentClassification* сохраняются результаты нечеткой классификации студентов, полученные фаззификатором на базе нейронной сети.

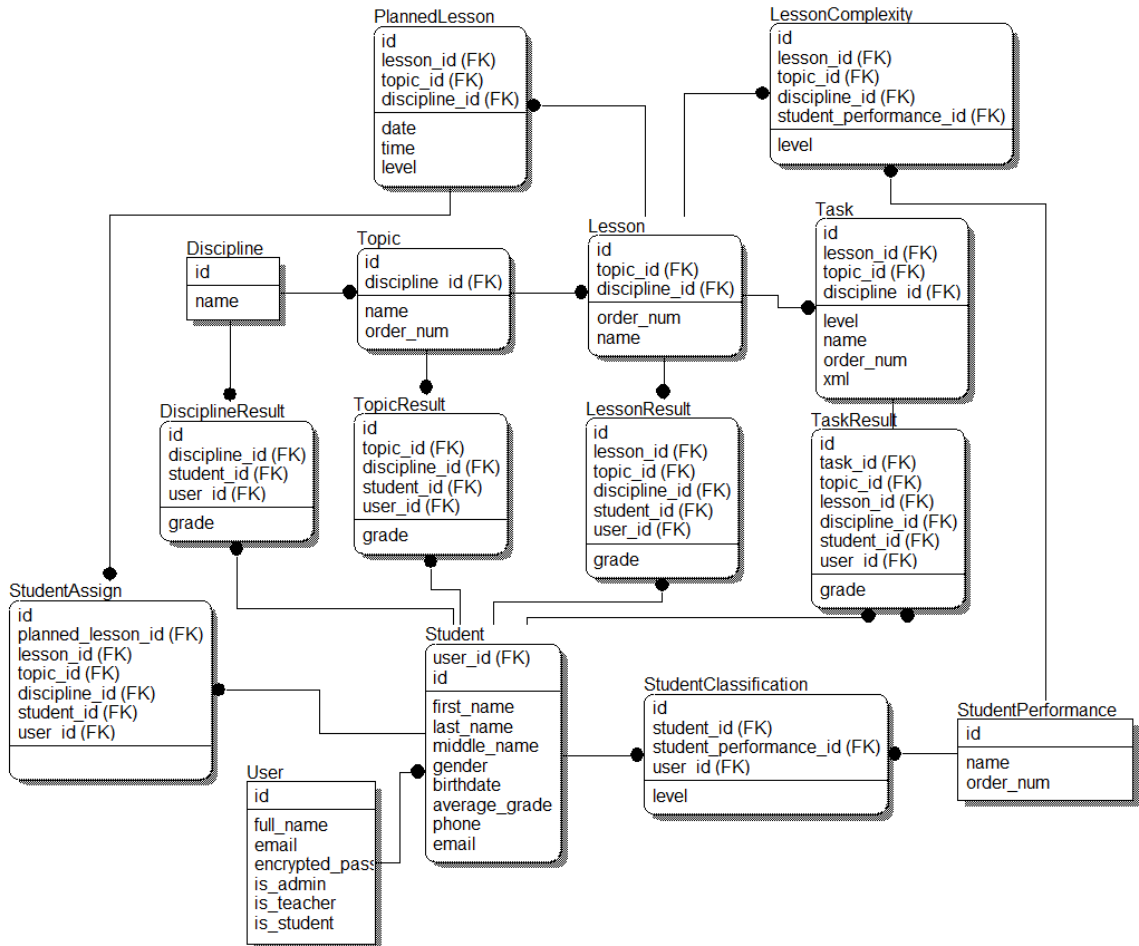


Рис. 29. Логическая структура базы данных программного комплекса

Таблицы *PlannedLesson* и *StudentAssign* представляют собой расписание занятий, составленное преподавателем при помощи экспертной системы. Результаты расчетов экспертной системы сохраняются в таблице *PlannedLessons*, содержащей дату запланированного занятия, время, название занятия и уровень сложности, а в таблице *StudentAssign* сохраняется окончательное назначение студентов по уровням сложности, скорректированное преподавателем.

3.4. Протоколы и форматы обмена данными между компонентами программного комплекса

3.4.1. Протоколы передачи данных

Аудио- и видеоданные передаются от клиентов к мультимедийному серверу по протоколу RTMP (Real Time Messaging Protocol) [32, 107]. Это протокол прикладного уровня, использующий TCP в качестве транспортного протокола. RTMP-соединение

поддерживается открытым на протяжении всего сеанса работы, что позволяет серверу инициировать передачу данных любому клиенту в любой момент времени.

Обмен служебной информацией между сервером и клиентским приложением происходит двумя способами: через RTMP-соединение либо путем создания HTTP-запросов к серверному приложению с использованием формата XML. В первом случае запросы, сформированные приложением на языке SQL, направляются по протоколу RTMP к мультимедийному серверу, который создает запросы к базе данных и возвращает клиенту полученные результаты. Во втором случае создаются фоновые HTTP-запросы с использованием API серверного приложения. Запросы обрабатываются контроллерами серверного приложения, которые создают запросы к БД с использованием объектно-реляционной модели и возвращают результат клиенту в формате XML. Такой принцип обмена данными аналогичен принципу работы технологии AJAX для динамических веб-страниц, позволяющей веб-странице получать данные с сервера без перезагрузки всей страницы.

3.4.2. Форматы хранения и передачи данных

При построении сложных информационных систем одной из задач является организация обмена данными между различными подсистемами. Задача разработки формата обмена данными существенно облегчается, если в подсистемах будет заложена технологически реализованная возможность передавать данные в едином, стандартизованном формате. В рассматриваемой СДО для представления учебных заданий с мультимедийными элементами (текст, математические формулы, различные текстовые символы, графические примитивы, изображения, добавленные из файлов) используется формат XML [66, 67]. Выбор этого формата обусловлен его универсальностью, наличием развитых средств манипулирования XML-документами в языках ActionScript (в клиентском приложении и редакторе задач) и Ruby (в серверном веб-приложении), а также тем, что структура учебных заданий и входящих в них материалов естественным образом отображается на иерархическую структуру XML-документа.

Каждое учебное задание представляет собой упорядоченное множество шагов $P = (S_1, S_2, \dots, S_n)$, где n – количество шагов задания. Каждый из этих шагов можно представить в виде множества $S_k = (D_k, E_k, Q_k, H_k)$, где $k = 1, \dots, n$; D_k – множество элементов формул, использованных при составлении k -го шага; E_k – палитра элементов, доступных студенту для составления ответа; Q_k – текстовые поля для свободного ввода;

H_k – совокупность графических примитивов (линий, точек, окружностей, эллипсов и т.д.), образующих графические материалы k -го шага задания. Обозначим через M множество символов библиотеки, предоставляемых преподавателю для составления формул и их частей. Оно состоит из заглавных и строчных латинских и греческих букв, цифр, знаков препинания и математических символов (интеграл, знак бесконечности и т.п.). Таким образом, $D_k \subset M$ и $E_k \subset M$. Часть элементов студентам не видна – они должны самостоятельно их добавить на поле из палитры в процессе решения шага задачи. Таким образом, множества D_k и Q_k состоят из двух подмножеств: $D_k = (D_{Fk}, D_{Rk})$, $Q_k = (Q_{Fk}, Q_{Rk})$ где D_{Fk} и Q_{Fk} – видимые элементы, D_{Rk} и Q_{Rk} – невидимые элементы. Очевидно, что должно выполняться условие $D_{Rk} \subset E_k$, чтобы студенты всегда могли составить правильный ответ. Это условие контролируется специальным алгоритмом, реализованным в редакторе учебных задач.

Пример отображения описанной теоретико-множественной структуры учебного задания на иерархическую структуру XML-документа приведено на рис. 30.

sequence

caption Задание

task_id 19

step S_k

id 0

palette

element (4)

type
1 symbol_cap_A
2 symbol_cap_B
3 symbol_cap_C
4 symbol_b

field

height 390

element (8)

layout	type	image_id	name	x	y	value	scale
1 fixed	symbol_a			86.75	44.9		
2 fixed	image	7	f62bed846794e17b0c c7276927e9ccc4.png	255	76		0.1472140762 463343
3 hidden	symbol_b			107.5	46.9		1
4 hidden	symbol_c			127.5	46.9		1
5 fixed	symbol_textfield_times			93.85	305	y	1
6 fixed	symbol_textfield_times			113.85	305	y	1
7 hidden	symbol_textfield_times			93.85	345	y	1
8 hidden	symbol_textfield_times			113.85	345	y	1

graphics

shape (11)

t	c	fc	th	tr	x0	y0	w	h
1 rect	0		1		129	111.03	121	61.01
2 circle	0		1					
3 ellipse_f	0	16711680	1	1	204	196.04	105	43.0
4 arrow	16763904	16711680	3		311	173.04		
5 region	16763904	16711680	3	1				
6 point	16711935		3					
7 lneto	16711935		3		50	108.03		
8 lneto	16711935		3		50	106		
9 lneto	16711935		3		51	106		
10 lneto	16711935		3		55	106		
11 line	16711935		3	1	79	16.01		

E_k

D_{Fk}

D_{Rk}

Q_{Fk}

Q_{Rk}

H_k

Рис. 30. Структура XML-документа, определяющего структуру материалов учебного задания

Корневой элемент XML-документа имеет имя *sequence* и включает атрибуты *caption* (заголовок учебного задания) и *task_id* (идентификатор задания в базе данных). Корневой элемент содержит множество дочерних элементов *step*, которые содержат информацию о каждом шаге задания.

Каждый элемент *step* включает атрибут *id* (порядковый номер шага) и элементы *palette* (палитра элементов шага) и *field* (содержимое рабочего поля шага).

Элемент *palette* включает в себя дочерние элементы *element*, образующие множество символов палитры E_k . Каждый элемент содержит атрибут *type*, определяющий имя символа в библиотеке. Например, значение *type* = “*symbol_cap_A*” соответствует заглавной латинской букве *A*.

Элемент *field*, определяющий содержимое рабочего поля, содержит атрибут *height*, задающий высоту рабочего поля, множество элементов *element*, образующих множества D_k и Q_k , и узел *graphics*, дочерние элементы которого образуют множество графических примитивов H_k .

Каждый узел множества *element* задает один символ из библиотеки символов, находящийся на рабочем поле. Атрибут *layout* задает роль символа в процессе решения задачи: видимый элемент формулы, невидимый элемент, элемент ответа студента, верно или неверно добавленный элемент ответа. Узлы со значением атрибута *layout* = «*fixed*» (кроме текстовых полей) – это видимые элементы формул, образующие множество D_{Fk} , а узлы со значением атрибута *layout* = «*hidden*» являются скрытыми и образуют множество D_{Rk} . Скрытые элементы не видны студентам и используются только для проверки их ответов преподавателем. Атрибут *type* содержит имя символа в библиотеке символов Flash-приложения. Атрибуты *x* и *y* задают координаты символа на рабочем поле, а атрибут *scale* – его коэффициент масштабирования.

Узлы, задающие текстовые поля, имеют значение атрибута *type*, равное «*symbol_textfield*» или «*symbol_textfield_times*» (для шрифта с засечками). Эти узлы задают множество Q_k : узлы со значением атрибута *layout* = «*fixed*» – это статические текстовые поля, образующие множество Q_{Fk} , а узлы со значением атрибута *layout* = «*hidden*» являются скрытыми текстовыми полями и образуют множество Q_{Rk} . Статические текстовые поля используются для ввода произвольных символов, отсутствующих в библиотеке. Скрытые текстовые поля используются для свободного ввода ответа студентами. Для студентов они отображаются пустыми текстовыми полями, в которые можно установить фокус ввода и ввести нужное значение с клавиатуры.

Узлы, определяющие изображения, добавленные из внешних графических файлов, имеют атрибут *name*, соответствующий имени файла в файловом хранилище. Для отображения этих элементов клиентское приложение производит HTTP-запрос к серверному приложению с использованием указанного имени файла.

Рисунки представляются узлами *shape*, являющимися дочерними узлами элемента *graphics*. Рисунки создаются в процессе редактирования учебного задания из графических примитивов – точек, линий, окружностей, эллипсов, прямоугольников, произвольных замкнутых фигур. Замкнутые фигуры (окружности, эллипсы, прямоугольники, произвольные фигуры) могут иметь цветную заливку. Произвольные линии аппроксимируются ломаными линиями, вершины которых соответствуют последовательности положений курсора мыши при ее рисовании.

Пример. XML-документ, задающий учебное задание, состоящее из единственного рабочего поля высотой 450 пикселей с расположенным на нем символом «А» с координатами (256,75; 100,9) и прямой линией черного цвета толщиной 1 пиксель, соединяющей точки (115; 83) и (285; 88), будет иметь следующий вид.

```
<sequence caption="Модельный пример" task_id="1">
  <step id="0">
    <field height="450">
      <element layout="fixed" type="symbol_cap_A" x="256.75" y="100.9"/>
      <graphics>
        <shape t="line" c="0" th="1" x0="115" y0="83" x1="285" y1="88"/>
      </graphics>
    </field>
  </step>
</sequence>
```

Список используемых элементов XML, задающих графические примитивы, и атрибутов, задающих их размер, цвет и положение, приведен в табл. 8. Через *h* обозначена высота рабочего поля, задаваемая при составлении шага задания.

Таблица 8. Параметры элементов XML, задающих графические примитивы

Тип элемента	Параметры элемента							
	Координаты начала	Координаты конца	Координаты центра	Толщина линии	Цвет линии	Видимость линии	Цвет заливки	Радиус
Имя атрибута	$x0, y0$	$x1, y1$	x, y	th	c	tr	fc	r
Допустимые значения атрибута	$x0: [0; 480]$ $y0: [0; h]$	$x1: [0; 480]$ $y1: [0; h]$	$x: [0; 480]$ $y: [0; h]$	$[1; 15]$	$[0; 256^3]$	$\{0; 1\}$	$[0; 256^3]$	$[1; 15]$
Наличие атрибута в элементах XML, задающих графические примитивы								
Точка			+		+			+
Прямая	+	+		+	+			
Окружность			+	+	+	+	+	+
Прямоугольник	+	+		+	+	+	+	
Эллипс	+	+		+	+	+	+	
Элемент ломаной линии	+	+		+	+			
Стрелка	+	+		+	+			
Замкнутая область				+	+	+	+	
Элемент контура замкнутой области		+						

3.5. Взаимодействие пользователей в режиме реального времени и его реализация в клиентском веб-приложении

Рассмотрим процесс взаимодействия студента и преподавателя на k -м шаге решения задачи и реализацию этого взаимодействия при помощи обмена данными в формате XML. При переходе к k -му шагу клиентское приложение считывает k -й узел *step* корневого узла XML-документа, добавляет новое рабочее поле в конец прокручиваемой области и

отображает на нем элементы формул D_{Fk} , палитру E_k , рисунки H_k , текстовые поля Q_{Fk} и пустые поля свободного ввода Q_{Rk} в соответствии с атрибутами и значениями дочерних элементов узла *step*.

Студент составляет ответ G_k^0 , перемещая элементы D_k^0 из палитры на места скрытых элементов и вводя значения Q_k^0 в текстовые поля Q_{Rk} . Таким образом, ответ студента состоит из множества элементов $G_k^0 = D_k^0 \cup Q_k^0$. При этом происходит формирование XML-документа, задающего структуру ответа, и передача этого документа через мультимедийный сервер. Этот XML-документ содержит элементы $S_k^0 = (D_k^0, E_k, Q_k^0, H_k)$, представленные в виде таких же узлов, что и исходный XML-документ, представляющий текущий шаг, но узлы, представляющие введенный ответ студента G_k^0 , имеют значение атрибута *layout* = «*added*».

Веб-приложение на компьютере преподавателя принимает XML-документ с ответом студента и отображает его содержимое на экране рядом с правильным ответом, состоящим из множества символов D_k и правильных значений текстовых полей Q_k . Преподаватель производит проверку правильности ответа путем его визуального сравнения с правильным ответом.

В случае, если $G_k^0 = D_{Rk} \cup Q_{Rk}$ (элементы ответа совпадают с невидимыми элементами и значения, введенные в текстовые поля, верны) и все элементы поставлены на правильные места, то преподаватель засчитывает ответ как правильный, и студент ожидает перехода к следующему шагу. Таким образом, общее условие правильности ответа имеет вид $G_k^0 = D_{Rk} \cup Q_{Rk}$. Однако правильный ответ зачастую может быть введен различными способами, поэтому ответ может быть засчитан и при невыполнении данного условия. В этом заключается гибкость подхода к обучению в режиме реального времени с использованием мультимедийных средств, поскольку допускается свобода действий при вводе ответа и не требуется наличие алгоритмов анализа формул для проверки правильности ответов.

Если какой-либо элемент ответа неверен, поставлен на неправильное место или неправильно введено значение в текстовое поле, то преподаватель переходит в режим указания ошибок, отмечая неправильные элементы и делая устное разъяснение. Результат проверки можно записать в виде $F_k^0 = D_{Fk} \cup G_k^C \cup G_k^N$, где G_k^C и G_k^N – соответственно правильно и неправильно добавленные студентом элементы, включая текстовые поля. Эти

элементы отображаются у студента разными цветами: правильные – зеленым, неправильные – красным.

Проанализировав результат проверки преподавателем F_k^0 , студент удаляет с поля неправильные элементы G_k^N и набирает с помощью палитры исправленный вариант ответа, а также может ввести в текстовые поля новые значения. Затем студент отправляет новый вариант ответа $G_k^1 = G_k^C \cup G_k^{1L}$, где G_k^{1L} – заново введенные элементы. Преподаватель проверяет новое решение, и если $G_k^1 = D_{Rk} \cup Q_{Rk}$ и все элементы поставлены на правильные места, то ответ засчитывается, а если нет, то весь цикл исправления ошибок повторяется заново.

Возможен случай, когда студент после нескольких попыток так и не смог ввести правильный ответ. В этом случае преподаватель переходит к следующему шагу задачи, а студенту показывается правильный ответ, состоящий из всех видимых и невидимых элементов $S_k = (D_k, E_k, Q_k, H_k)$. Соответствующий шаг задачи в этом случае считается нерешенным, и это отражается на величине оценки, выставляемой студенту в конце занятия.

3.6. Экспертная система и ее реализация

3.6.1. Структура классов экспертной системы

Экспертная система реализована на основе языка программирования Ruby, являющемся высокоуровневым объектно-ориентированным языком общего назначения, позволяющим реализовать необходимые алгоритмы и структуры данных. В структуру экспертной системы входят следующие компоненты.

1. База знаний, состоящая из наборов правил нечеткого логического вывода для каждого блока нечеткого вывода.
2. Блоки фаззификации, содержащие модели функций принадлежности термов лингвистических переменных.
3. Блоки нечёткого вывода и дефаззификации, которые реализуют соответствующие процедуры и обрабатывают информацию, поступающую на вход системы.
4. Блок работы с нейронной сетью, реализующий модель нейросети и содержащий алгоритмы для ее загрузки в память, сохранения на диск, обучения и тестирования.

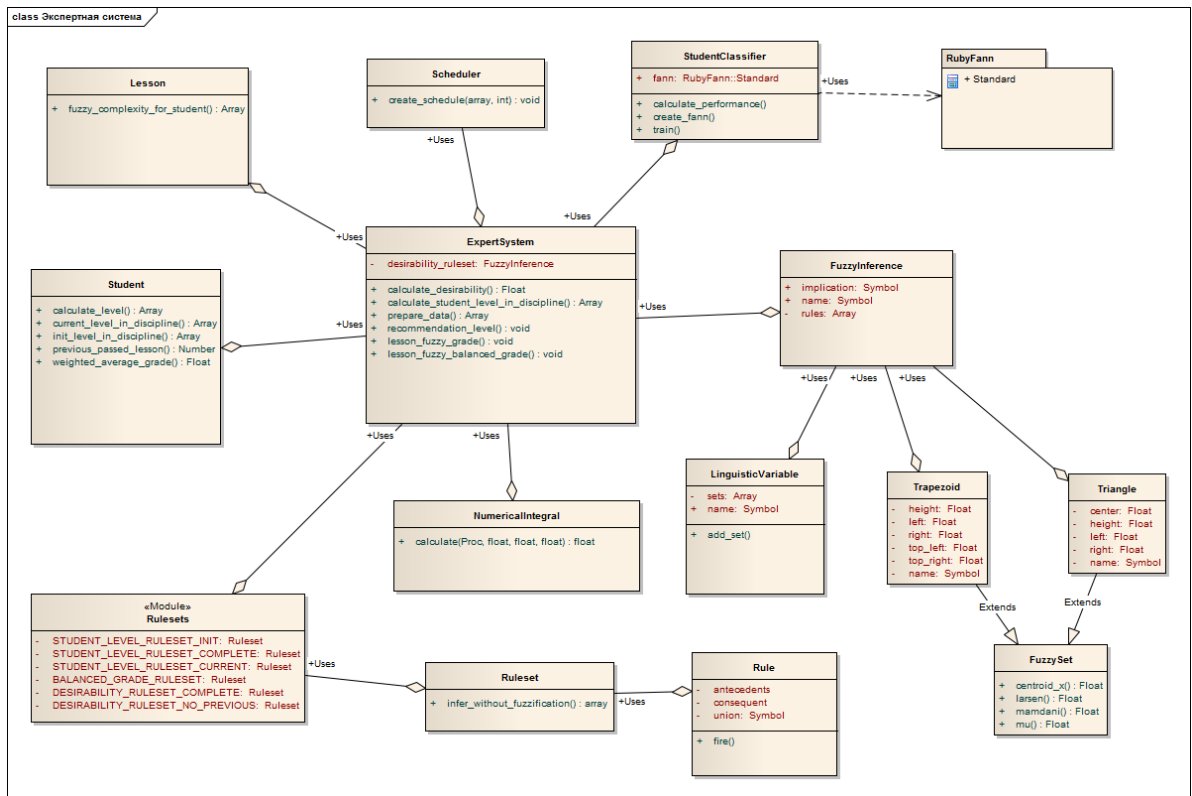


Рис. 31. Диаграмма классов экспертной системы

Каждой сущности (блоку, переменной, функции и т.д.) экспертной системы соответствует класс или модуль (класс, не порождающий объектов) в языке Ruby. UML-диаграмма классов и отношений между ними приведена на рисунке 31.

Класс *Scheduler* предназначен непосредственно для составления расписания занятий при помощи экспертной системы. Класс содержит функцию *create_schedule*, из которой и производится вызов функции расчета степени рекомендуемости для каждого студента и каждого уровня сложности текущего занятия и по результатам расчетов составляется назначение занятий студентам, сохраняемое в базе данных. Эта функция вызывается преподавателем или администратором из веб-интерфейса при составлении расписания (кнопка «Составить» на рис. 38).

Большая часть функций, относящихся к экспертной системе, реализована в классе *ExpertSystem*. Функции, специфические для студента и занятия, реализованы в их соответствующих классах *Student* и *Lesson*, одновременно являющихся моделями соответствующих сущностей в рамках архитектуры «Модель-Представление-Контроллер». Функции работы с нейронной сетью выделены в класс *StudentClassifier*. Класс *NumericalIntegral* реализует численное интегрирование, используемое для дефаззификации на заключительном этапе расчетов экспертной системы. Правила вывода, используемые для нечеткого вывода, собраны в модуль *Rulesets*. Класс *Ruleset* реализует процедуру нечеткого вывода, используемую блоками экспертной системы.

Классы *Trapezoid* и *Triangle* реализуют соответственно трапециевидные и треугольные функции принадлежности, используемые в блоках фаззификации. Оба класса унаследованы от класса *FuzzySet*, реализующего общие методы работы с нечеткими множествами. Алгоритм нечеткого вывода по схеме Мамдани реализован в классе *FuzzyInference*.

3.6.2. Подготовка исходных данных для экспертной системы

Рекомендуемый номер уровня сложности рассчитывается для каждого студента и каждого уровня сложности текущего занятия. Поэтому исходными данными для каждого цикла вычислений являются три числа: идентификатор студента *student_id*, идентификатор занятия *lesson_id* и порядковый номер уровня сложности *level_num*. Однако входными данными экспертной системы являются пять величин: w_{i-1} – оценка, полученная студентом за предыдущее занятие; \bar{C}_{i-1} – оценки сложности предыдущего решенного студентом задания; \bar{C}_i – оценки сложности текущего занятия рассматриваемого уровня сложности; \bar{D} – вектор, содержащий исходную информацию о студенте; $\langle w \rangle$ – средневзвешенная сумма оценок студента по данной дисциплине. Следовательно, требуется этап подготовки исходных данных для экспертной системы. Эту функцию выполняет функция *prepare_data* из класса *ExpertSystem*. Схема работы этой функции представлена на рис. 32.

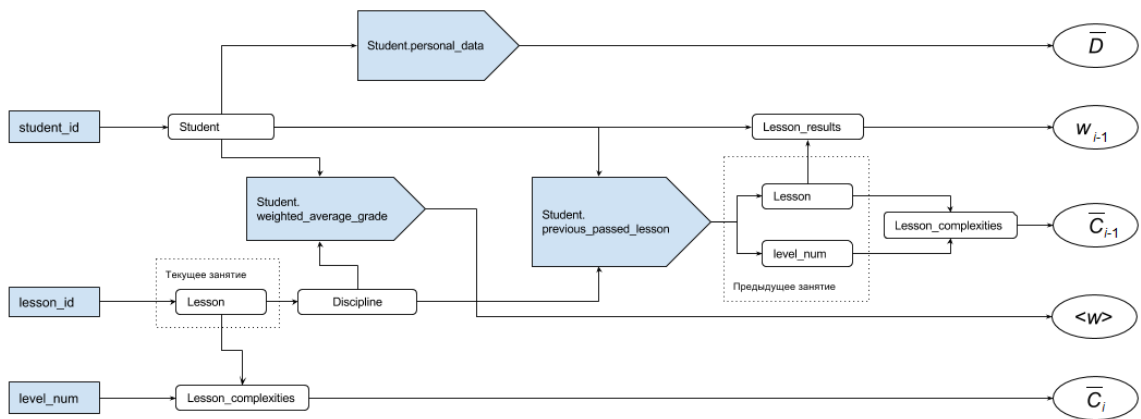


Рис. 32. Схема подготовки исходных данных для экспертной системы

Закрашенными прямоугольниками обозначены исходные данные; прозрачные прямоугольники – данные, получаемые из базы данных; прямоугольники с углом – функции; овалы – готовые данные, подаваемые на вход экспертной системы.

Вначале из таблицы *Students* базы данных по идентификатору *student_id* запрашивается запись с информацией о студенте, атрибуты которой помещаются в экземпляр класса *Student*. Функция *personal_data*, определенная в классе *Student*, извлекает из базы данных величины, составляющие координаты вектора \bar{D} исходной информации: оценка за вступительный тест, средний балл в аттестате, оценка за ЕГЭ по предмету и т. д.. Затем по идентификатору *lesson_id* из базы данных запрашивается запись с информацией о занятии (*Lesson*), а далее по внешнему ключу *discipline_id* из родительской таблицы *Discipline* извлекается информация о дисциплине, которой принадлежит это занятие. Функция *weighted_average_grade*, определенная в классе *Student*, вычисляет средневзвешенную оценку $\langle w \rangle_1^{i-1}$ для данного студента по данной дисциплине по формуле (2.3). Функция *previous_passed_lesson* находит идентификатор последнего пройденного с положительной оценкой занятия вместе с номером пройденного уровня (*level_num*). Если студент прошел несколько уровней одного и того же занятия, то выбирается уровень с наименьшим номером (как правило, он является самым сложным). Далее из таблицы результатов *Lesson_results* извлекается оценка за предыдущее занятие w_{i-1} , а из таблицы сложностей *Lesson_complexities* – информация о сложности текущего (\bar{C}_i) и предыдущего (\bar{C}_{i-1}) пройденного студентом занятий. Если какая-либо величина отсутствует в базе данных (например, отсутствует предыдущее занятие, если текущее занятие является первым), то вместо нее возвращается пустое значение, которому в Ruby соответствует специальное значение *nil*, и экспертная система корректно обрабатывает такие случаи.

3.6.3. Осуществление вычислений экспертной системой

Основной функцией, производящей вычисление степени рекомендуемости для заданного уровня сложности, является функция *calculate_desirability*, определенная в классе *ExpertSystem* и производящая вычисление в соответствии с формулой (2.4) или (2.5). Функция принимает на вход массив исходных данных, подготовленный функцией *prepare_data*, обрабатывает данные путем вызова функций фаззификации, композиции, нечеткого вывода и дефаззификации (определенные в классах *FuzzyInference*, *FuzzySet* и т.д.) и возвращает искомую степень рекомендуемости в виде числового значения.

Полное вычисление степени рекомендуемости для всех уровней сложности какого-либо занятия является процедурой довольно высокой вычислительной сложности, поэтому при вычислениях применяется оптимизация путем кэширования значений, не изменяющихся в ходе расчета. При составлении расписания происходит перебор всех

имеющихся номеров уровней сложности для текущего занятия, при этом в каждом цикле вычислений данные о студенте (начальный и текущий уровень успеваемости, история оценок, предыдущее пройденное занятие, сбалансированная оценка за него) остаются неизменными. Поэтому величины, представленные переменными $\langle \tilde{w} \rangle_1^{i-1}$ (средневзвешенная сумма оценок), \bar{L} (текущая успеваемость), w_{i-1}^C (сбалансированная оценка за предыдущее пройденное занятие), сохраняются в специальную статическую переменную после первого их вычисления и повторно используются на протяжении всего цикла вычислений.

3.7. Выводы

В главе 3 приведено подробное описание технологий, использованных для построения программного комплекса, форматов хранения и передачи данных.

Разработана структура программного комплекса, отражающая клиент-серверную архитектуру его построения. Разработаны компоненты программной системы: серверное приложение, клиентское веб-приложение с дополнительными мультимедийными функциями, редактор учебных заданий, мультимедийный сервер. Произведена интеграция разработанных компонентов и готовых программных решений (PostgreSQL, Nginx) в единый программный комплекс СДО MathConference. На клиентское веб-приложение и мультимедийный сервер получены свидетельства о государственной регистрации программ для ЭВМ.

Разработана структура данных для хранения и передачи мультимедийных материалов (формул, текста, рисунков и т.п.), используемых для решения учебных заданий в режиме реального времени, на основе формата XML. Реализовано отображение данной структуры в графический вид и обратное преобразование из интерфейса визуального редактора в XML. Разработана модель обмена данными между преподавателем и студентами, позволяющая производить учебный процесс в режиме реального времени. Создана структура базы данных программной системы.

Реализована экспертная система, интегрированная в серверное веб-приложение. Разработана структура классов и функций экспертной системы, осуществлена программная реализация процедуры иерархического нечеткого вывода, описанной в главе 2.

Глава 4. Интерфейс и порядок работы с программным комплексом

4.1. Общее описание программного комплекса

Описываемая в настоящей работе СДО MathConference представляет собой программный комплекс, позволяющий адаптировать процесс обучения каждого студента под его индивидуальные нужды. Он включает систему управления обучением, позволяющую визуализировать текущие результаты, рассчитывать требуемый уровень сложности решаемых задач, планировать индивидуальную траекторию обучения каждого студента. Веб-интерфейс системы содержит встроенное приложение для проведения занятий в форме веб-конференции. Программный комплекс может быть использован для проведения занятий, проведения итоговой аттестации в форме экзамена или зачёта для студентов, обучающихся в филиалах институтов или университетов, а также для обучающихся по заочной или дистанционной форме.

Рассматриваемая СДО предназначена прежде всего для изучения математических дисциплин студентами начальных курсов, однако благодаря универсальности мультимедийных средств она может быть использована и для других дисциплин. С помощью этой СДО возможно реализовать все основные виды учебного процесса: лекции, практические занятия, консультации, защита курсовых работ с презентациями, зачеты.

4.2. Распределение ролей пользователей

Каждый пользователь, зарегистрированный в системе, относится к одному из трех классов пользователей: «преподаватель», «администратор» или «студент». Преподаватель может одновременно являться и администратором.

В зависимости от роли меняется список доступных функций системы. Также меняется внешний вид и список доступных функций веб-приложения конференции.

4.2.1. Функции роли преподавателя

Преподавателю, работающему с СДО, доступны следующие функции:

- просмотр структуры учебных материалов: списка дисциплин, тем дисциплин, тем занятий;
- добавление, изменение, удаление дисциплин;

- добавление, изменение, удаление тем в каждой дисциплине;
- добавление, изменение, удаление занятий;
- создание заданий для решения студентами при помощи специального интерфейса, назначение уровней сложности и оценок сложности заданий для каждого уровня;
- просмотр списка студентов из назначенных групп;
- составление расписания занятий при помощи экспертной системы;
- проведение занятий в режиме веб-конференции с использованием дополнительных функций: показа презентаций, работы с виртуальной доской, показа рабочего стола, обмена файлами (стр. 0);
- решение математических задач в режиме веб-конференции со студентами в роли преподавателя (стр. 0);
- просмотр результатов и оценок всех студентов из групп, назначенных данному преподавателю, в разрезе дисциплин, тем, занятий;
- просмотр траекторий обучения каждого студента из назначенных групп, журнала расчетов экспертной системы.

4.2.2. Функции роли студента

Пользователю в роли студента доступны следующие функции:

- просмотр своих результатов обучения по всем назначенным дисциплинам;
- участие в занятиях в режиме веб-конференции в роли студента;
- решение математических задач в режиме веб-конференции в роли студента;
- просмотр расписания занятий.

4.2.3. Функции роли администратора

Администраторы обладают полным доступом к системе и могут использовать все функции, доступные студентам и преподавателям, а также ряд дополнительных функций:

- создание учетных записей пользователей, задание первоначального пароля для каждого пользователя;
- создание учебных групп и назначение групп преподавателям;
- назначение дисциплин учебным группам;
- настройка экспертной системы и просмотр журнала расчетов.

4.3. Веб-интерфейс программного комплекса

Интерфейс программного комплекса MathConference представляет собой веб-сайт, доступный через Интернет по адресу www.mathconference.ru [12]. При переходе по указанному адресу пользователю будет предложено войти на сайт под своими учётными данными (рис. 33). В качестве имени пользователя используется адрес электронной почты. В системе закрытая регистрация, т.е. отсутствует функция регистрации для произвольного посетителя сайта. Учетные записи пользователей создаются администраторами либо преподавателями, имеющими доступ к функциям администратора.

Рис. 33. Форма входа в систему

4.3.1. Веб-интерфейс преподавателя

Внешний вид главной страницы интерфейса СДО в роли преподавателя показан на рис. 34.

В верхней части страницы находится логотип системы, одновременно являющийся ссылкой на главную страницу. Под логотипом расположено главное меню, состав пунктов которого различен для разных ролей. Преподавателю доступны пункты «Учебные материалы», «Студенты», «Оценки» и «Занятия». Справа в строке меню расположена ссылка на страницу личных данных текущего пользователя и ссылка для выхода из системы.

Под меню находится область уведомлений, в которой отображается сообщение об успешности последнего действия пользователя либо сообщение об ошибке.

Перечисленные элементы присутствуют на каждой странице интерфейса. Под областью уведомлений находится основная область, в которой отображается информация

текущей страницы. В качестве примера на рис. 34 приведено содержание страницы результатов студентов в разрезе дисциплин, доступной через пункт меню «Оценки», подменю «По дисциплинам». Интерфейс отображает в удобном табличном виде текущие результаты обучения каждого студента в виде круговых диаграмм. Этот интерфейс доступен и преподавателям, и студентам, но преподавателям доступна информация обо всех студентах, а студенты могут просматривать только свои результаты.

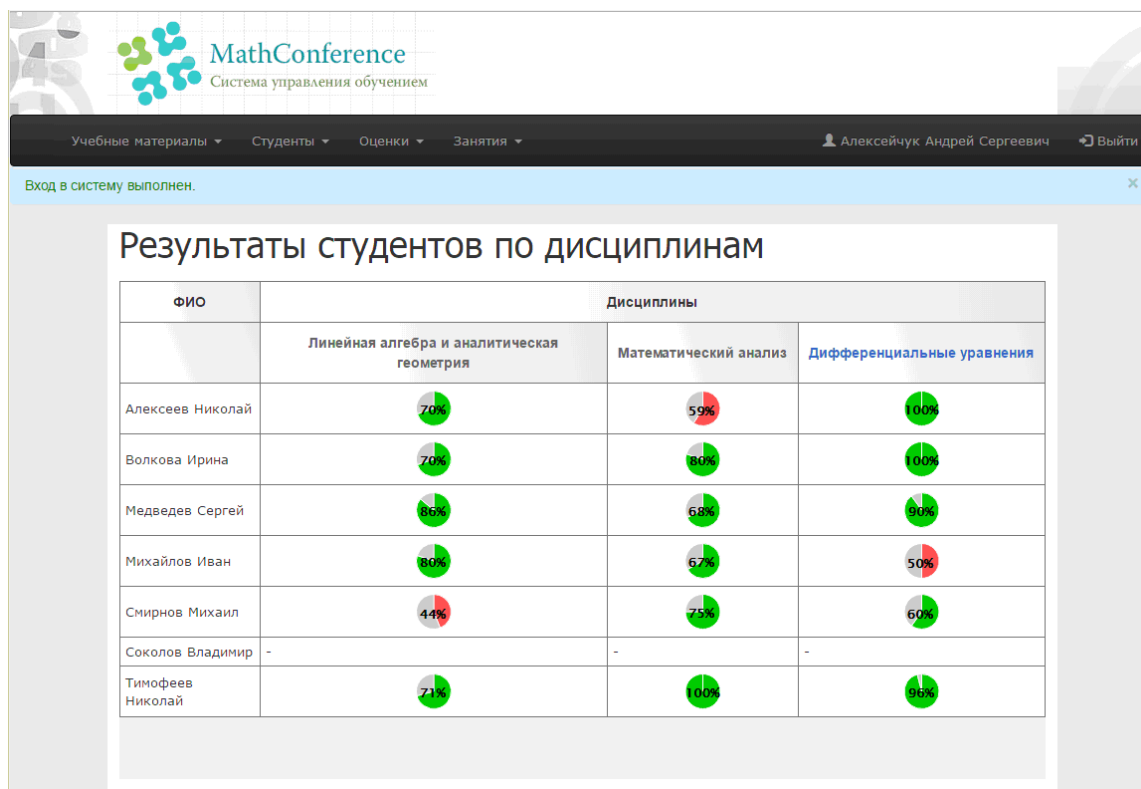


Рис. 34. Веб-интерфейс в роли преподавателя

Преподаватель может просматривать текущие оценки студентов в разрезе дисциплин, тем и занятий. Оценки отображаются в виде круговых диаграмм с процентами (от 0 до 100%). Зеленый цвет означает, что студент успешно справляется со всеми задачами данной дисциплины (любого уровня сложности), красный – что по указанной теме получена оценка ниже 60%.

Таблицы успеваемости в разрезе тем (рис. 35) и занятий (рис. 36) выводятся в виде линейного списка.

Результаты студентов по темам

Тема	Студент	Оценка	Действия
Линейная алгебра > Матрицы	Петров Михаил	 80%	  
Линейная алгебра > Матрицы	Смирнов Николай	 86%	  
Линейная алгебра > Матрицы	Иванов Владимир	 42%	  

Рис. 35. Таблица успеваемости студентов по темам

Результаты студентов по занятиям

Занятие	Студент	Оценка	Действия
1. Линейная алгебра > Матрицы > Определение, свойства	Иванов Владимир	 42%	  
2. Линейная алгебра > Матрицы > Умножение	Иванов Владимир	 32%	  
1. Линейная алгебра > Матрицы > Определение, свойства	Петров Михаил	 80%	  
1. Линейная алгебра > Матрицы > Определение, свойства	Смирнов Николай	 86%	  

Рис. 36. Таблица успеваемости студентов по занятиям

Полная структура меню в роли преподавателя выглядит следующим образом:

- Учебные материалы
 - Структура материалов
 - Дисциплины
 - Темы
 - Занятия
 - Задания
- Студенты
 - Список студентов
 - Учебные группы
- Оценки
 - По дисциплинам
 - По темам
 - По занятиям
 - По заданиям
- Занятия
 - Расписание занятий на неделю

- Расписание занятий на месяц
- Составление расписания
- Демо-конференция

Учебные материалы имеют иерархическую структуру. Каждая дисциплина, изучаемая при помощи данной системы (например, «Линейная алгебра»), разделяется на темы (например, «Матрицы», «Определители», «СЛАУ» и т.д.). Каждой теме соответствует определенный набор занятий (например, теме «Определители» соответствуют практические занятия, на которых изучаются разные методы вычисления определителей).

На рис. 37 приведен вид страницы, отображающей иерархическую структуру учебных материалов. Страница доступна через пункт меню «Учебные материалы», подпункт «Структура материалов».

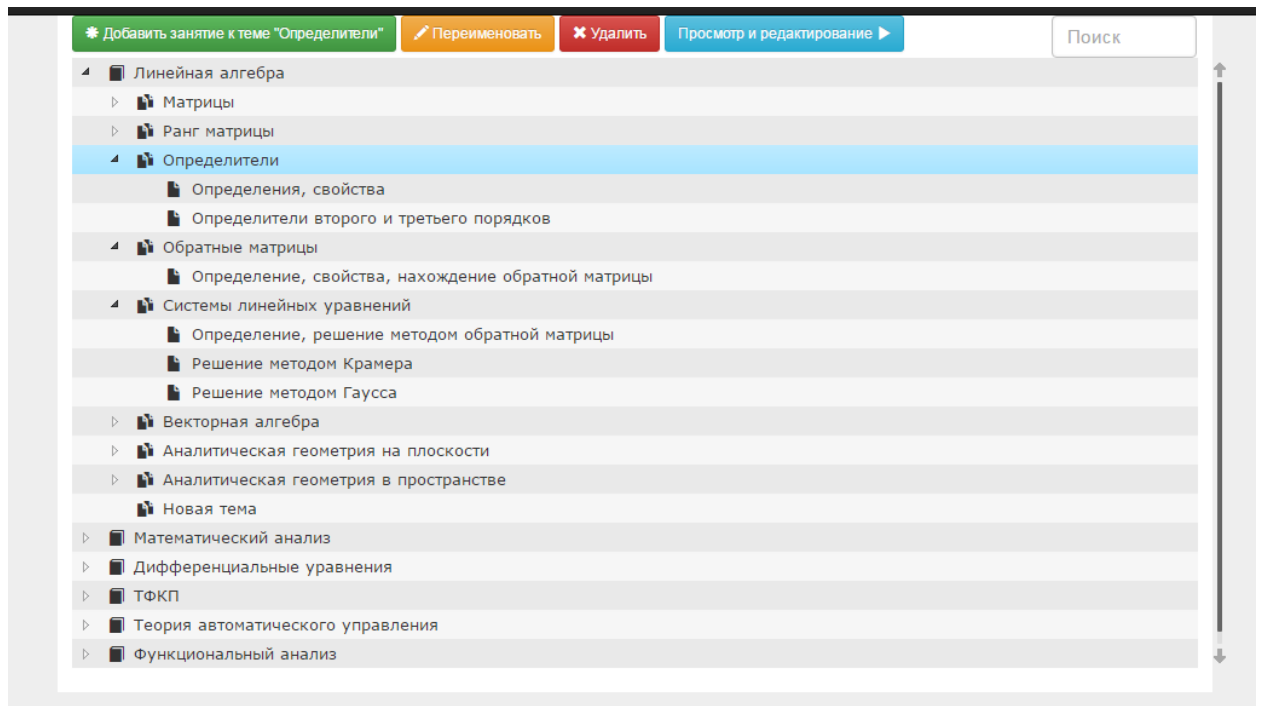


Рис. 37. Структура учебных материалов

Каждая строка, соответствующая дисциплине, содержит раскрывающийся список тем, а каждая тема, в свою очередь, содержит список занятий, относящихся к этой теме, в порядке их проведения. Над таблицей находится набор кнопок, с помощью которых можно добавлять темы и занятия, переименовывать или удалять пункты и переходить к подробному просмотру и редактированию данных о дисциплине, теме или занятии. Выделенный пункт, к которому относятся действия кнопок, подсвечен синим цветом. Также можно изменять порядок тем и занятий, а также перемещать занятия между темами

при помощи перетаскивания пунктов списка мышью. Предусмотрена функция поиска для быстрого нахождения темы или занятия по названию или части названия.

Составление расписания – ключевая функция, посредством которой реализуется индивидуализация процесса обучения для каждого студента при помощи экспертной системы. Функция доступна через пункт меню «Занятия», подпункт «Составление расписания».

Составление расписания происходит в три шага. Первый шаг – выбор занятия в иерархическом списке учебных материалов и выбор учебных групп, для которых составляется расписание (рис. 38). Допускается участие студентов из разных групп в одном занятии.

Шаг 1: выбор занятия и учебной группы.

Выберите занятие:

Линейная алгебра и аналитическая геометрия	> Матрицы	> Определение, типы
Математический анализ	> Определители	> Действия над матрицами (умножение)
Дифференциальные уравнения	> СЛАУ	> Действия над матрицами (сложение, транспонирование)
	> Ранг матрицы	

Выберите учебные группы, для которых будет составлено расписание:

08-105

08-101

08-102

Занятие: Действия над матрицами (сложение, транспонирование)
Группы: 08-105, 08-101

Рис. 38. Составление расписания занятий (первый шаг)

После нажатия кнопки «Составить» происходит расчет рекомендуемого уровня сложности экспертной системой для каждого студента из выбранных групп. Результаты расчетов представляются на втором шаге в виде предварительного распределения студентов по уровням (рис. 39). На этом этапе преподаватель знакомится с рекомендациями системы и при необходимости вносит коррективы при помощи перетаскивания пунктов мышью между уровнями. Студентов, которым не требуется прохождение занятия, можно убрать из расписания при помощи кнопки «Убрать студента».

Шаг 2: назначение уровней сложности.

Ниже приведены рекомендации экспертной системы.

✖ Отменить составление расписания ✔ Далее ➡

- ▲ 📅 Уровень 1
 - 👤 Медведев Сергей
 - 👤 Алексеев Николай
- ▲ 📅 Уровень 2
 - 👤 Волкова Ирина
 - 👤 Михайлов Иван
 - 👤 Соколов Владимир
- ▲ 📅 Уровень 3
 - 👤 Смирнов Михаил
 - 👤 Тимофеев Николай

Рис. 39. Составление расписания занятий (второй шаг)

Нажав «Далее», преподаватель переходит к заключительному шагу составления расписания, где необходимо выбрать дату и время проведения занятия для каждого уровня (рис. 40).

Шаг 3: назначение даты и времени занятий.

Занятие для уровня 1:	03.11.2016	10:30
Занятие для уровня 2:	03.11.2016	14:00
Занятие для уровня 3:	04.11.2016	10:30

Рис. 40. Составление расписания занятий (третий шаг)

Выбор даты и времени производится при помощи всплывающего календарика, появляющегося при помещении фокуса ввода в текстовое поле. Также дату и время занятия можно ввести вручную. Закончив ввод, преподаватель нажимает кнопку «Сохранить». После этого открывается страница расписания занятий на текущий месяц с добавленными занятиями (рис. 41).

Расписание занятий на ноябрь 2016 г.

Пн	Вт	Ср	Чт	Пт	Сб	Вс
31.10.2016	01.11.2016	02.11.2016	03.11.2016	04.11.2016	05.11.2016	06.11.2016
					15:00 Линейная алгебра и аналитическая геометрия 08-102, 08-105	
07.11.2016	08.11.2016	09.11.2016	10.11.2016	11.11.2016	12.11.2016	13.11.2016
14:00 Линейная алгебра и аналитическая геометрия 08-102, 08-105		15:15 Линейная алгебра и аналитическая геометрия 08-105	10:00 Линейная алгебра и аналитическая геометрия 08-102, 08-105		09:25 Линейная алгебра и аналитическая геометрия 08-102	17:00 Линейная алгебра и аналитическая геометрия 08-105
16:00 Линейная алгебра и аналитическая геометрия 08-105			14:00 Линейная алгебра и аналитическая геометрия 08-102, 08-105		15:35 Линейная алгебра и аналитическая геометрия 08-102, 08-105	
14.11.2016	15.11.2016	16.11.2016	17.11.2016	18.11.2016	19.11.2016	20.11.2016

Рис. 41. Расписание занятий на месяц

Также доступно расписание занятий на текущую неделю (рис. 42) через пункт меню «Занятия», подпункт «Расписание занятий на неделю».

Расписание занятий на неделю

Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
07.11.2016 14:00 Линейная алгебра и аналитическая геометрия (тема: Матрицы) (занятие: Определение, типы) Группы: 08-105, 08-102 (4 студента)		09.11.2016 15:15 Линейная алгебра и аналитическая геометрия (тема: Матрицы) (занятие: Действия над матрицами (умножение)) Группа 08-105 (2 студента)	10.11.2016 10:00 Линейная алгебра и аналитическая геометрия (тема: Матрицы) (занятие: Действия над матрицами (сложение, транспонирование)) Группы: 08-105, 08-102 (2 студента)		12.11.2016 15:00 Линейная алгебра и аналитическая геометрия (тема: Матрицы) (занятие: Определение, типы) Группы: 08-102, 08-105 (2 студента)	13.11.2016 17:00 Линейная алгебра и аналитическая геометрия (тема: Матрицы) (занятие: Действия над матрицами (умножение)) Группа 08-105 (1 студент)
07.11.2016 16:00 Линейная алгебра и аналитическая геометрия (тема: Матрицы) (занятие: Определение, типы) Группа 08-105 (1 студент)			10.11.2016 14:00 Линейная алгебра и аналитическая геометрия (тема: Матрицы) (занятие: Действия над матрицами (сложение, транспонирование)) Группы: 08-105, 08-102 (5 студентов)			

Рис. 42. Расписание занятий на неделю

Расписание содержит ссылки, по которым можно перейти к странице с веб-приложением для проведения конференций. Веб-приложение открывает виртуальную аудиторию, где проводится соответствующее занятие.

4.3.2. Веб-интерфейс администратора

Структура меню в роли администратора повторяет структуру меню преподавателя, но в нем присутствует еще один пункт – «Администрирование». Структура этого пункта следующая:

- Администрирование
 - Настройка экспертной системы
 - Журнал расчетов
 - Список преподавателей
 - Список пользователей
 - Назначение дисциплин преподавателям

Настройка экспертной системы включает редактирование правил нечеткого вывода для каждого блока нечеткого вывода и настройку параметра «забывания» β , используемого при расчете средневзвешенной оценки. Пункт «Журнал расчетов» открывает результаты расчетов экспертной системы, из которых можно получить сведения о значениях входных, промежуточных и выходных переменных экспертной системы в процессе последнего цикла расчетов. Пункт меню «Назначение дисциплин преподавателям» позволяет установить соответствие между преподавателями и дисциплинами, по которым они проводят занятия.

4.3.3. Веб-интерфейс студента

Веб-интерфейс в роли студента в целом выглядит аналогично интерфейсу в роли преподавателя. Студенту доступны страницы с индивидуальными результатами занятий и расписанием занятий, на которые назначен данный студент, на неделю и на месяц. Расписание состоит только из занятий, назначенных данному студенту каким-либо преподавателем при помощи функции составления расписания.

Из каждой строки расписания по ссылке можно перейти в виртуальную аудиторию, где будет проводиться соответствующее занятие.

4.4. Работа с клиентским приложением для проведения веб-конференций

Клиентское приложение для проведения учебных веб-конференций представляет собой многопользовательское веб-приложение, встроенное в веб-интерфейс программного комплекса.

4.4.1. Окно клиентского приложения

Внешний вид окна клиентского приложения в роли преподавателя, когда не запущены никакие дополнительные функции, показан на рис. 43.



Рис. 43. Внешний вид окна клиентского приложения

Большую часть окна приложения занимает рабочее поле, разделённое горизонтальной линией на две части. У каждого участника занятия есть своё окно на рабочем поле, в котором отображается видеопоток с его веб-камеры. Каждый участник может в любой момент включать или выключать своё видео. Окна преподавателей появляются над горизонтальной линией, а студентов – под линией. Преподаватель может перемещать окна участников по рабочему полю, при этом расположение окон синхронизируется у всех участников. Предусмотрено место для 3 окон преподавателей и 30 окон студентов. Возможно присутствие и большего количества участников, однако в любом случае будут показаны только окна первых 30 участников.

Вверху окна, над рабочим полем, находится поле заголовка. Оно представляет собой текстовое поле, доступное для редактирования преподавателю. В это поле можно вводить заголовок текущего занятия (например, название предмета и тему занятия). Содержимое заголовка видно всем участникам.

Для участия в веб-конференции требуется наличие веб-камеры и микрофона. При отсутствии у какого-либо участника веб-камеры его окно будет отображаться пустым. Каждый участник может в любой момент включать или выключать своё видео. В начале занятия микрофон включен только у преподавателя. Студенты не могут сами включать свой микрофон во избежание появления эха и фонового шума. Если студент хочет что-либо сказать, он нажимает кнопку «Поднять руку». Его окно окрашивается в красный цвет, и это видят все участники. Увидев этот сигнал, преподаватель может предоставить слово студенту, включив его микрофон. Остальные участники могут слушать диалог этого студента и преподавателя. Можно включать микрофон одновременно у нескольких студентов, инициируя таким образом многостороннее общение.

Все участники могут в любой момент обмениваться текстовыми сообщениями в чате. Появление новых сообщений сопровождается звуком. История сообщений чата сохраняется даже после выхода всех участников, до тех пор, пока её не очистит кто-либо из преподавателей.

Внизу окна приложения находится список участников конференции. Фамилии преподавателей постоянно находятся вверху списка, а ниже располагаются фамилии студентов в алфавитном порядке. Список обновляется автоматически, когда какой-нибудь участник входит в конференцию или покидает её.

4.4.2. Демонстрирование презентаций

Функция демонстрирования презентаций предназначена для показа дополнительного визуального материала. Презентации дают возможность чётко выделить структуру изучаемого материала, показывать схемы, выведение формул, уравнений и отдельных их частей, а преподаватель может сопровождать показ презентации устными пояснениями (рис. 44).

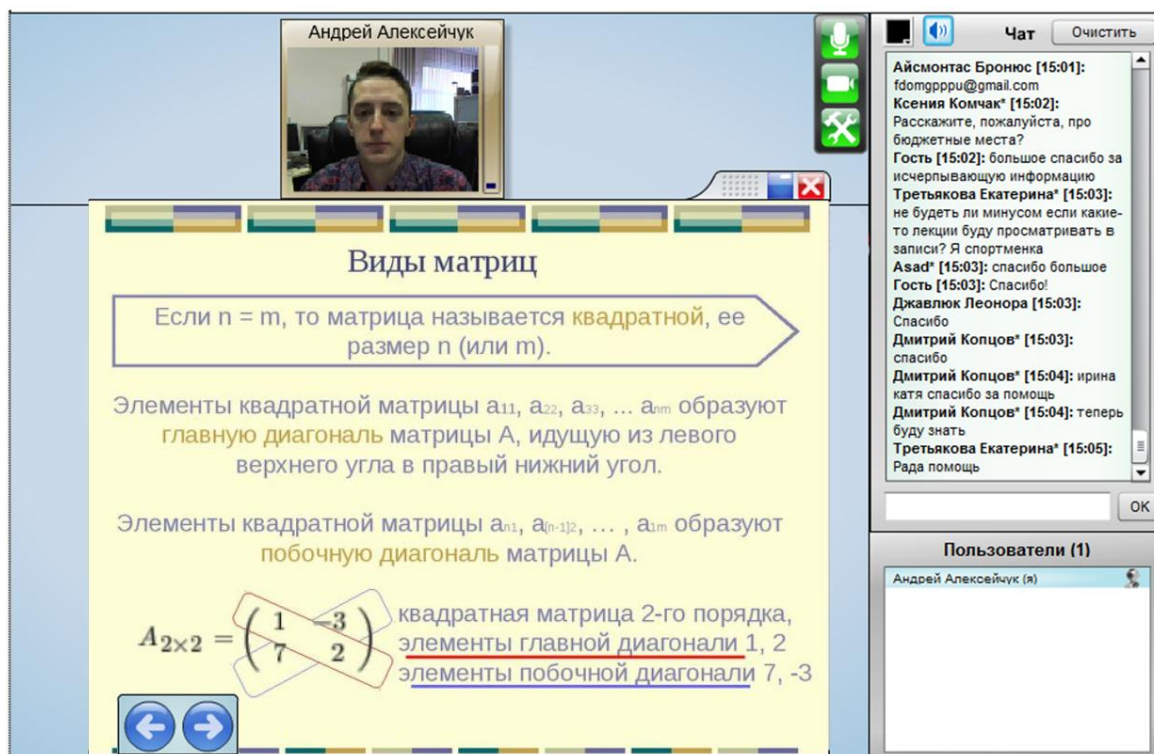


Рис. 44. Функция демонстрации презентаций

Презентации готовятся преподавателем в формате PowerPoint (.ppt) или OpenDocument Presentation (.odp) и загружаются на сервер. Загрузив нужные файлы презентаций, преподаватель может вернуться в веб-конференцию и начинать занятие. Преподаватель может запустить любую из загруженных им презентаций, при этом он сменяет слайды и может комментировать или обсуждать с другими участниками их содержимое. Слайды отображаются в специальном дополнительном окне на рабочем поле и сменяются синхронно у всех участников. Доступна функция рисования маркером поверх презентации, благодаря чему у преподавателя есть возможность наносить поверх слайдов пометки, которые видны всем участникам.

4.4.3. Работа с виртуальной доской

Виртуальная доска – ещё одно мультимедийное средство преподавателя, которое выполняет те же функции, что и обычная доска в аудитории – доведение до учащихся графической информации (рис. 45).

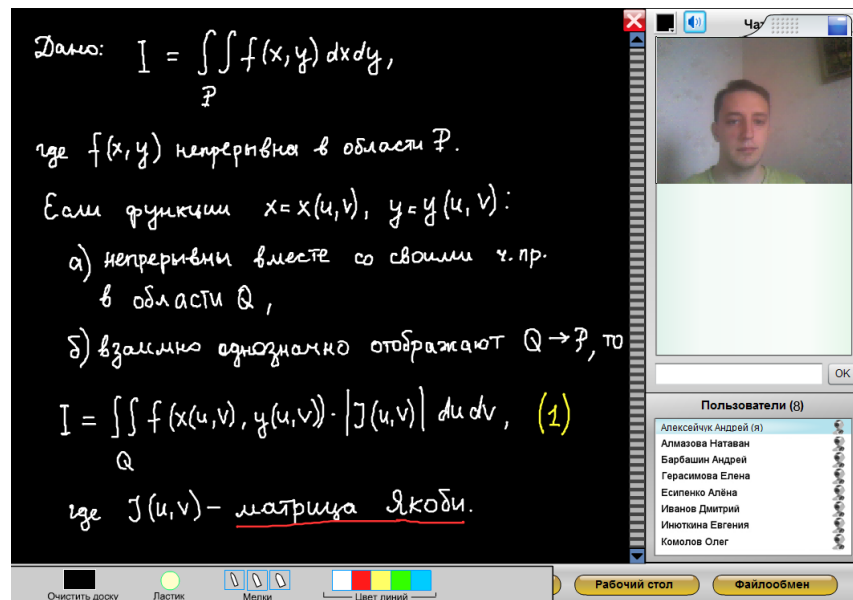


Рис. 45. Виртуальная доска

С помощью виртуальной доски можно проводить занятия и консультации по математике и другим предметам, требующим отображения большого количества формул и рисунков. На доске можно писать и рисовать, как на обычной доске. Доступна возможность изменения цвета и толщины линий, ластик, а также функция вертикальной прокрутки доски. Содержимое виртуальной доски синхронизируется в режиме реального времени у всех участников конференции. Виртуальная доска наиболее полезна при использовании графического планшета либо интерактивной доски – устройства, позволяющего рисовать специальными маркерами на проецируемом с компьютера изображении.

4.4.4. Демонстрирование рабочего стола

Функция демонстрирования рабочего стола позволяет преподавателю показывать всем участникам содержимое своего экрана или любой его части в плавающем окне на рабочем поле. Эта функция может быть полезна для показа веб-сайтов, документов, рисунков, обучения работе с различными программами непосредственно с компьютера преподавателя. Пример показа веб-сайта с помощью данной функции показан на рис. 46. Доступны функции выбора скорости видеопотока и разрешения захватываемого изображения. Большие значения этих параметров позволяют получить лучшее качество изображения, однако требуют большей скорости Интернет-соединения у участников конференции для комфортного просмотра.

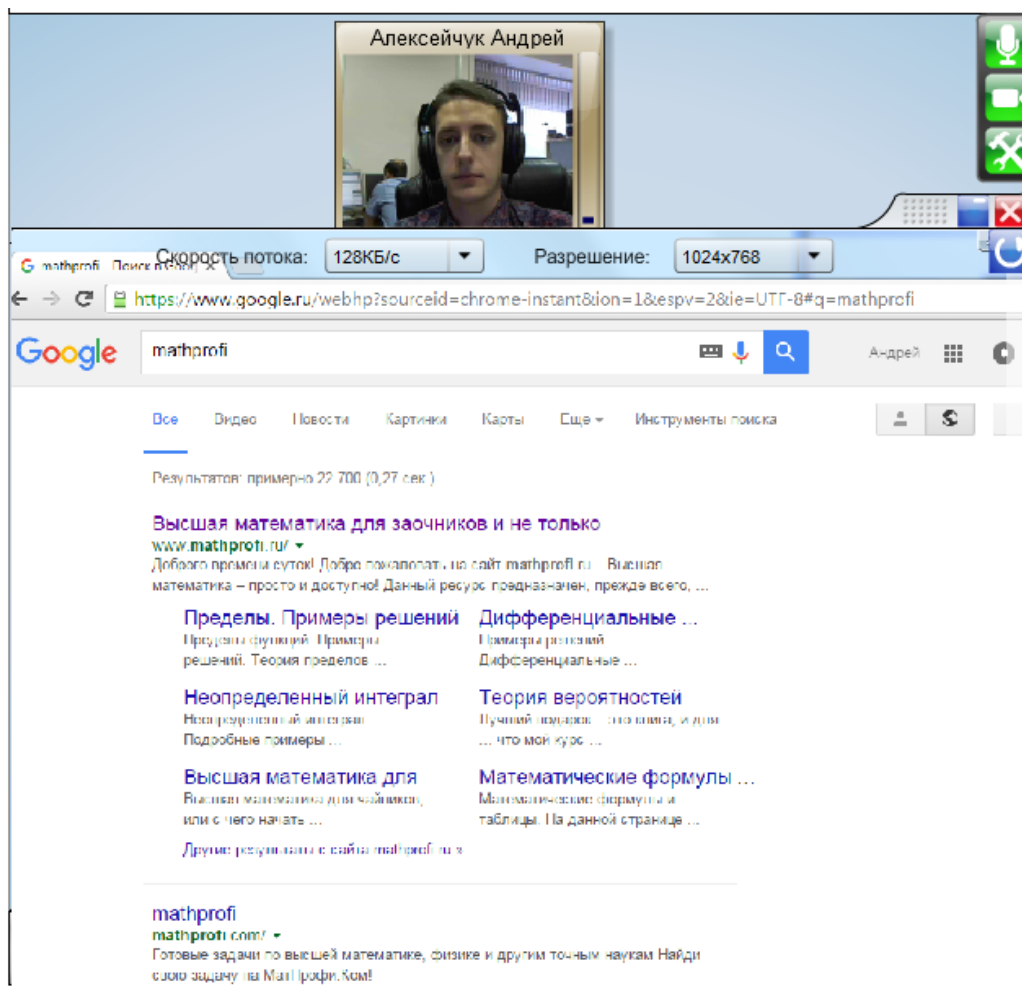


Рис. 46. Функция демонстрации рабочего стола

4.4.5. Функция обмена файлами

Функция обмена файлами позволяет преподавателю делиться файлами произвольного типа (документы, таблицы, презентации, изображения и т.д.) со всеми участниками. Для отправки файлов служит специальное окно (рис. 47). Нажатие кнопки «Обзор...» открывает диалог, с помощью которого преподаватель может выбрать файл для отправки со своего компьютера.

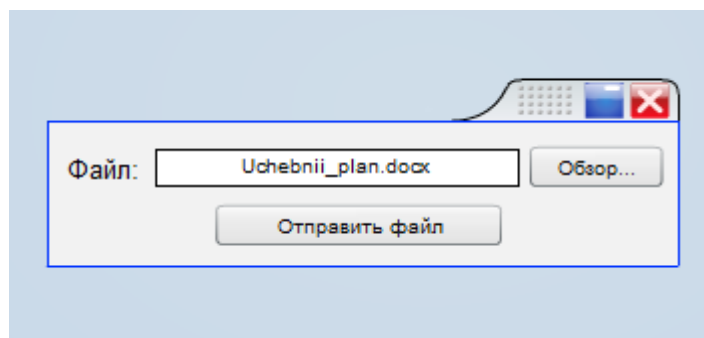


Рис. 47. Окно обмена файлами

После того как преподаватель отправил файл, у всех студентов появляется всплывающее уведомление (рис. 48), позволяющее сохранить копию этого файла себе на компьютер для дальнейшей работы с ним.

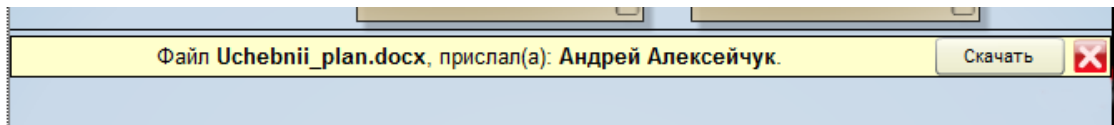


Рис. 48. Уведомление о присланном файле

Возможен одновременный запуск нескольких указанных функций. В этом случае возможно реализовывать сценарии обучения:

- объяснение материала по заранее подготовленным презентациям с последующей проверкой усвоения путём совместного решения задачи;
- одновременный показ презентации и решение задач по шагам после каждого слайда;
- приведение примера решения задачи на виртуальной доске с одновременным пошаговым решением задачи;
- демонстрацию презентации с последующей рассылкой соответствующего файла для самостоятельного изучения и т.д.

4.4.6. Функция пошагового решения учебных заданий

Функция пошагового решения учебных заданий – важнейшее средство, отличающее рассматриваемую систему дистанционного обучения от множества других систем веб-конференций. Эта функция позволяет всем присутствующим в веб-конференции студентам одновременно решать задачи в режиме реального времени под контролем преподавателя с использованием мультимедийных элементов: формул, графиков, рисунков, текста. Решение задач происходит по шагам, т.е. каждая задача делится на небольшие подзадачи, материалы каждого из которых могут быть уместены на рабочем поле шага задания. Функция пошагового решения учебных заданий является инвариантной относительно предметной области, поскольку широкие возможности создания и редактирования мультимедийных материалов и пошаговый подход к решению заданий позволяют создавать учебные задания по широкому кругу дисциплин: математическим, гуманитарным, естественнонаучным. Решение каждой подзадачи происходит одновременно всеми студентами, при этом преподаватель проверяет результаты решения каждого студента и указывает ошибки и неточности. При этом сохраняется визуальный контакт и возможность устного общения между преподавателем и студентами [19].

Внешний вид окна клиентского веб-приложения в режиме решения учебных заданий в роли преподавателя приведен на рис. 49 [12, 17].

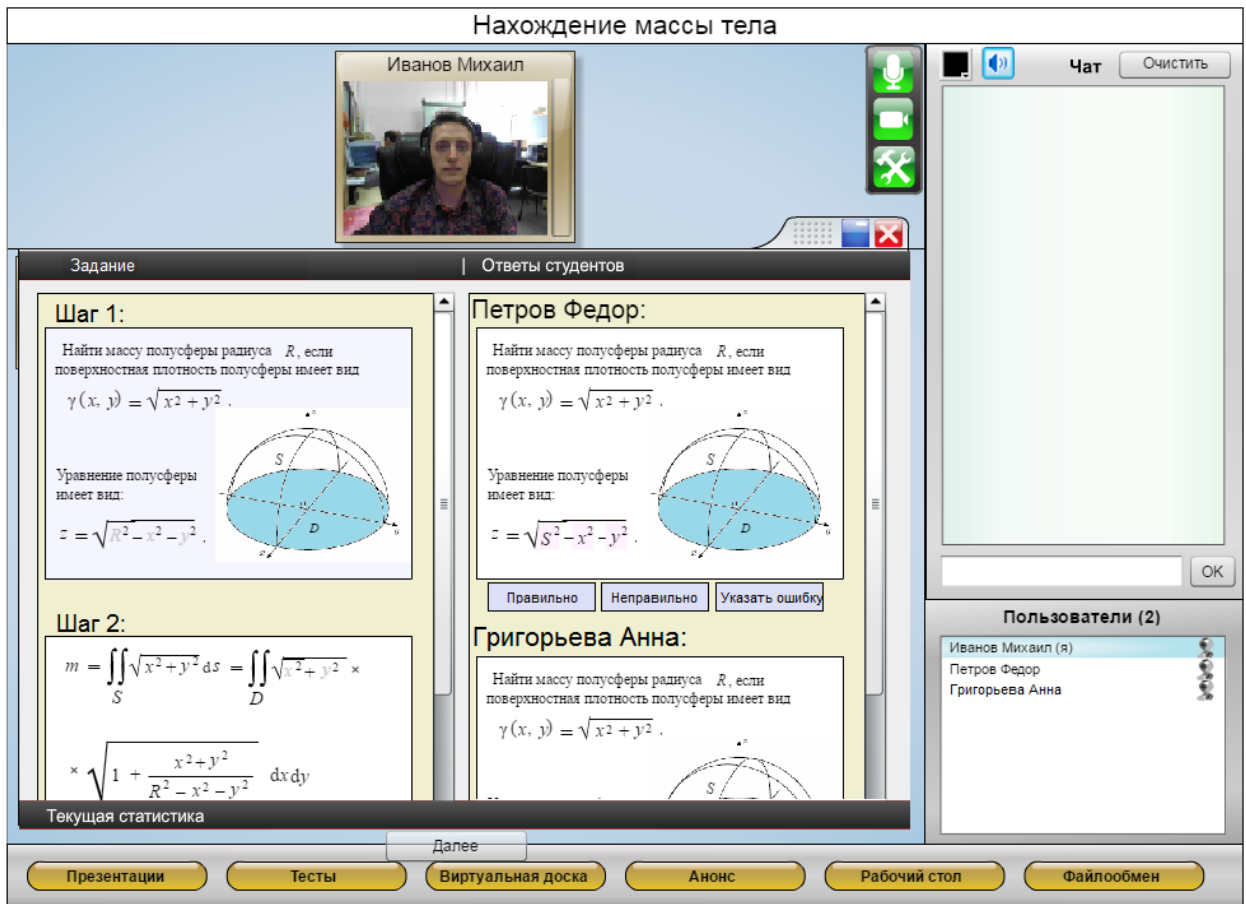


Рис. 49. Интерфейс средства решения математических задач (вид в роли преподавателя)

Интерфейс состоит из двух полей с вертикальной прокруткой. В левом поле отображается полное решение выбранного учебного задания. Поле, соответствующее текущему шагу, выделено голубым цветом. Серым цветом обозначены скрытые элементы, невидимые для студентов. В правом прокручиваемом поле отображаются ответы студентов на текущий шаг задания. Задача преподавателя состоит в том, чтобы сравнить каждый полученный ответ в правом поле с образцом текущего шага на левом поле и принять решение о правильности этого ответа. Для этого преподавателю доступны три опции: «Правильно», «Неправильно», «Указать ошибку». В случае неправильного ответа преподаватель нажимает кнопку «Неправильно» либо, нажав «Указать ошибку», отмечает ошибочные элементы в ответе студента, а также может дать устное разъяснение. После этого студент получает результат проверки и, в случае наличия ошибок, может ввести исправленный ответ.

Когда все студенты выполнили текущий шаг задания (или вышло отведённое время), преподаватель переходит к следующему шагу, и процесс решения повторяется. Если кто-

либо из студентов так и не дал правильного ответа, то ответ будет показан, но отредактировать его будет уже невозможно.

Преподавателю доступна открывающаяся в нижней части интерфейса таблица статистики, где отображаются текущие результаты каждого студента в разрезе шагов задания.

Интерфейс студента состоит из поля с прокруткой, содержащего поля текущего учебного задания (рис. 50). На этом поле отображаются все шаги решаемого задания от первого до текущего. Благодаря наличию функции вертикальной прокрутки в линейке студент может просмотреть все предыдущие шаги решаемого задания.

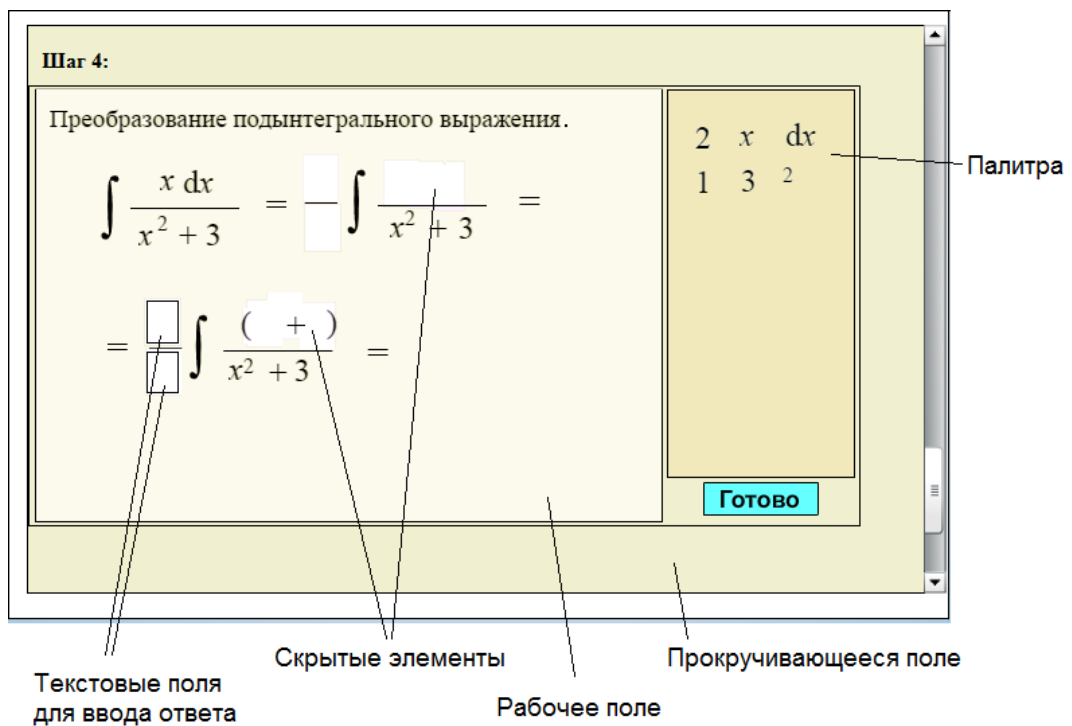


Рис. 50. Интерфейс средства решения математических задач (вид в роли студента)

На рабочем поле, соответствующем текущему шагу задания, отображаются подготовленные преподавателем графические материалы – формулы, рисунки, графики, текст и т.д., при этом элементы, назначенные скрытыми, не отображаются. Справа от рабочего поля находится палитра. Для составления варианта ответа студенту требуется перетащить элементы палитры на рабочее поле в места, соответствующие положению скрытых элементов. Студент может добавлять произвольное количество элементов из палитры на рабочее поле. Также шаг задания может предусматривать ввод текстовых или числовых значений в текстовые поля для ввода ответа (рис. 50, слева).

4.7. Работа с редактором учебных заданий

Для создания учебных заданий предусмотрен специальный редактор, представляющий собой приложение, встроенное в веб-интерфейс программной системы и выполненное с использованием Flash-технологии (рис. 26). Он позволяет создавать задачи, состоящие из любого числа шагов, с использованием большого набора изобразительных средств [12].

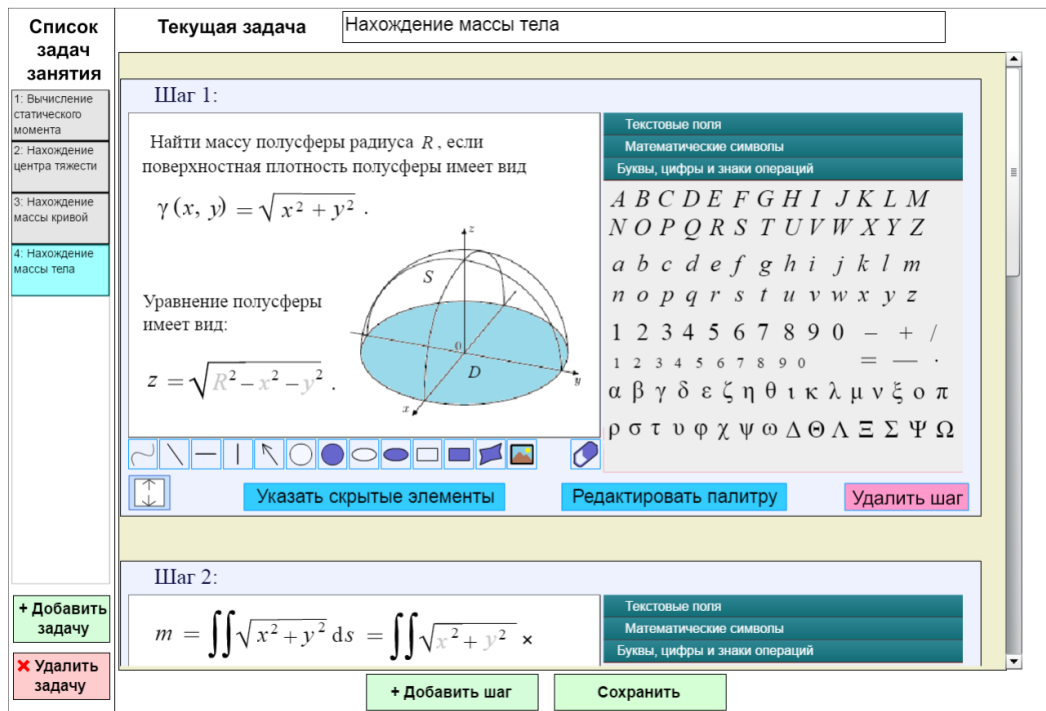


Рис. 51. Редактор учебных заданий

Каждое учебное задание представляет собой последовательность небольших подзадач, или шагов, материалы каждого из которых изображены на соответствующем рабочем поле. В левой части окна редактора находится список задач, подготовленных для текущего занятия. В центре окна редактора расположено прокручивающееся поле с рабочими полями шагов и библиотекой символов для каждого шага.

Для создания и редактирования учебных заданий доступны следующие функции и изобразительные средства:

- добавление и удаление шагов для каждой задачи,
- увеличение и уменьшение высоты рабочего поля для каждого шага,
- перенос произвольного количества символов из библиотеки на рабочее поле,
- рисование произвольных линий,
- рисование прямых линий (с отдельными режимами рисования горизонтальных и вертикальных линий) и стрелок,

- рисование прямоугольников, окружностей, эллипсов с заливкой цветом или без неё;
- рисование замкнутой области произвольной формы с заливкой цветом, с контуром или без него;
- изменение толщины линий, цвета линий и заливки во всех режимах рисования;
- многоуровневая отмена и возврат действий;
- удаление элементов ластиком и клавиатурой (клавишей Delete),
- одиночное и групповое перемещение и масштабирование символов, добавленных из библиотеки,
- точное позиционирование символов стрелками клавиатуры,
- добавление символов из библиотеки на палитру для каждого шага,
- назначение скрытых элементов для каждого шага,
- добавление любых изображений из внешних графических файлов формата JPEG, GIF, PNG, BMP;
- добавление текстовых полей, позволяющих вводить произвольные символы и части формул, а также играющих роль полей для ввода ответа студентами;
- добавление текстовых комментариев на рабочее поле.

Перечисленные инструменты позволяют создавать учебные задания с использованием широкого набора графических средств, благодаря чему рассматриваемая система является инвариантной относительно предметной области.

Добавление символов на рабочее поле производится путем их перетаскивания из библиотеки на нужное место рабочего поля. После добавления символа становится доступна функция изменения масштаба символа, позволяющая увеличить или уменьшить размеры символа. Для одновременного перемещения или изменения масштаба нескольких символов доступна функция группового выбора символов. Для группового выбора необходимо нажать левую клавишу мыши и протянуть курсор таким образом, чтобы прозрачный прямоугольник выделения пересекался со всеми выделяемыми элементами. После отпускания клавиши мыши выделенные элементы оказываются подсвеченными синим цветом, и появляется подвижный уголок справа от выделенных элементов, перемещением которого достигается их масштабирование. Перемещение выделенных элементов производится их перетаскиванием мышью. Для точного позиционирования выделенных элементов можно воспользоваться клавишами со стрелками на клавиатуре. Клавишей Delete производится удаление выделенных элементов с рабочего поля.

Интерфейс редактора включает библиотеку элементов со всевозможными символами (буквами, цифрами, математическими символами, знаками операций и т.д.) и текстовыми полями. Текстовые поля могут использоваться для создания комментариев, для ввода частей формул или как поля для ввода ответа студентами. Инструменты рисования позволяют изображать как стандартные фигуры – окружности, эллипсы, линии, прямоугольники – так и рисовать произвольные фигуры и надписи. Доступна функция выбора цвета, а также ластик и кнопки отмены и возврата последнего нарисованного элемента. С помощью графических средств преподаватель полностью изображает решение каждого шага. Затем он отмечает на поле скрытые элементы, т.е. элементы, которые студентам не будут видны и которые они должны будут добавить самостоятельно. После этого он из всех доступных элементов (цифры, буквы, знаки арифметических операций и т.д.) выбирает элементы для палитры студента в текущей подзадаче. Завершив редактирование, преподаватель нажимает кнопку «Сохранить», после чего материалы всех задач, составленных для текущего занятия, сохраняются в базу данных.

4.8. Практическое использование результатов исследования

На кафедре «Математическая кибернетика» МАИ было проведено тестирование созданной системы для дистанционного обучения студентов факультета, расположенного в г. Луховицы. Получены положительные отзывы студентов и администрации об удобстве и функциональности разработанной СДО, а также значительно упрощена работа преподавателя благодаря разделению проведения занятий разного уровня сложности со студентами сопоставимого уровня подготовки.

Также программный комплекс активно используется на факультете дистанционного обучения МГППУ. В формате веб-конференции регулярно проводятся лекции и семинарские занятия по широкому кругу дисциплин – математическим, гуманитарным, естественнонаучным. В основном эти занятия проводятся для студентов с ограниченными возможностями здоровья (ОВЗ), не имеющими возможности приезжать в университет, но зачастую они заменяют обычные занятия для всех студентов. Присутствие и активность на занятиях каждого студента отмечается преподавателем, и эта информация учитывается на зачётах и экзаменах. В целом внедрение новой формы занятий в МГППУ прошло весьма успешно и нашло поддержку как преподавателей, так и студентов, поскольку онлайн-общение между всеми без исключения участниками конференции играет

важнейшую роль в социализации и адаптации студентов-инвалидов. Также преимущество использования веб-конференций состоит в отсутствии необходимости тратить время и силы на дорогу до университета и обратно, что особенно актуально для студентов с ОВЗ и иногородних студентов. Программная система используется не только для учебных занятий. С его помощью организуется дистанционная защита курсовых работ, сдача зачётов, проводятся консультации перед экзаменами. Общение в виртуальных аудиториях используется и во внеурочное время для обмена опытом с преподавателями других вузов, проведения тренингов, расширения кругозора студентов, участия в телемостах со студентами других вузов.

Выводы

В четвертой главе описан внешний вид и порядок работы с интерфейсом программного комплекса MathConference, который представляет собой веб-сайт, находящийся по адресу www.mathconference.ru. Разработаны принципы построения веб-интерфейса этой системы, основывающиеся на таблицах с графическим представлением данных в виде диаграмм и иерархических списков.

Разработано клиентское приложение, встраиваемое в веб-интерфейс указанной системы и реализующее режим общения в формате веб-конференции. Разработаны дополнительные функции, расширяющие функциональность клиентского приложения: демонстрация презентаций, работа с виртуальной доской, обмен файлами, демонстрация рабочего стола, решение учебных заданий. Реализованы интерфейсы студента и преподавателя, позволяющие проводить решение математических задач в онлайн-режиме. Разработан порядок работы студента и преподавателя при решении задач, а также порядок создания и редактирования задач преподавателем при помощи приложения-редактора.

Заключение

Основным итогом диссертационной работы является разработка математических методов и программного комплекса для реализации одной из форм дистанционного обучения на базе веб-конференций в режиме реального времени, что выразилось в достижении следующих научных и практических результатов.

1. Разработан метод индивидуализации траектории обучения студента при обучении в режиме веб-конференции, предложена модель одной из дистанционных форм учебного процесса, разработан принцип построения индивидуальной траектории обучения посредством выбора рекомендуемого уровня сложности заданий для каждого студента перед проведением занятия [6, 8, 15, 19].

2. Разработана структура иерархической нечеткой экспертной системы, реализующей предложенную математическую модель индивидуализации процесса обучения каждого студента. Экспертная система обрабатывает экспертную информацию об уровне сложности каждого занятия и данные об успеваемости студентов и автоматически составляет расписание занятий, в котором каждому студенту назначен рекомендуемый уровень сложности занятий. Разработано математическое обеспечение принципа работы экспертной системы на основе методов нечеткой логики и нейронных сетей [8, 15, 16, 19].

3. Разработаны принципы создания, архитектура и требования к компонентам программного комплекса, объединяющего систему поддержки веб-конференций и экспертную систему, и обеспечивающего дистанционное обучение в режиме реального времени [11, 14].

4. Выполнена реализация программного комплекса системы дистанционного обучения MathConference, включающего веб-приложение для проведения учебных веб-конференций, редактор учебных заданий, серверное приложение с экспертной системой, мультимедийный сервер, СУБД и веб-сервер. Реализованы методы взаимодействия между компонентами программного комплекса, разработаны структуры данных, используемых для хранения и передачи информации [1, 4, 5, 7, 11, 21].

5. Реализованы пользовательские интерфейсы компонентов комплекса, предназначенные для проведения занятий в форме веб-конференции, управления учебным процессом, просмотра результатов обучения, создания учебных заданий, администрирования системы [1, 2, 3, 9, 10, 12, 13, 17, 18, 20].

Разработанные программные средства создают эффективную систему дистанционного обучения с приятным, дружественным интерфейсом, позволяющую организовывать удобное и увлекательное онлайн-обучение в любой точке земного шара. Сочетание

технологии веб-конференций с интерактивными мультимедийными средствами позволяет компенсировать недостаток личного контакта обучаемых с преподавателем при дистанционном обучении за счет виртуального общения. Внедрение данной системы открывает перспективы участия в учебном процессе для различных категорий студентов, в том числе студентов-инвалидов, что весьма важно для их социализации и устранения коммуникативных барьеров.

Список литературы

1. Алексейчук А. С. Контроль качества подготовки студентов по математическим дисциплинам в форме веб-конференции. – «Информационные и телекоммуникационные технологии», №19, 2013. – С. 16–21.
2. Алексейчук А. С. Интерактивная среда для обучения решению математических задач в режиме веб-конференции // Мат-лы 11-й Междунар. конф. «Авиация и космонавтика – 2012». – СПб.: Мастерская печати, 2012. – 412 с. – С. 369-370.
3. Алексейчук А. С. Интернет-семинары как форма дистанционного проведения занятий // Мат. XVII междунар. конф. по вычислительной механике и соврем. прикладным программным системам. – М., МАИ, 2011. – С. 776-777.
4. Алексейчук А. С. Использование телекоммуникационных технологий для создания систем онлайн-обучения // Молодые ученые – нашей новой школе. Материалы XI научно-практ. конф. – М.: МГППУ, 2012. – С. 440-441.
5. Алексейчук А. С. Клиентское приложение программного комплекса для проведения дистанционного обучения в форме веб-конференций // Свид-во о гос. регистрации программы для ЭВМ №2016662322 от 07.11.2016 г.
6. Алексейчук А. С. Модель взаимодействия студентов и преподавателя при решении учебных заданий в режиме реального времени // Научный альманах. Вып. 22: Материалы XIII Научно-практической конференции молодых ученых и студентов «Инновационный менеджмент в авиационной промышленности». – М.: «Доброе слово», 2017.
7. Алексейчук А. С. Мультимедийный серверный компонент программного комплекса для проведения дистанционного обучения в форме веб-конференций // Свид-во о гос. регистрации программы для ЭВМ №2016662678 от 17.11.2016 г.
8. Алексейчук А. С. Построение индивидуальной траектории обучения студента при помощи экспертной системы // Актуальные проблемы прикладной математики, информатики и механики. Сборник трудов Международной научно-технической конференции. – Воронеж: Изд-во «Научно-исследовательские публикации», 2016. – С. 349-351.
9. Алексейчук А. С. Применение веб-конференций с расширенными функциями в дистанционном обучении // Психологическая помощь социально незащищённым лицам с использованием дистанционных технологий (Интернет-консультирование и дистанционное обучение). Мат-лы. II междунар. конф. – М.: МГППУ, 2012. – С. 194-200.

10. Алексейчук А. С. Применение технологии веб-конференций для изучения математических методов в экономике // Мат. VIII научно-практ. конф. «Инновационный менеджмент в аэрокосмической промышленности». Научный альманах, вып. 16 – М.: «Доброе слово», 2012. – С. 142-148.
11. Алексейчук А. С. Принципы разработки телекоммуникационных образовательных систем на базе веб-конференций – Информационные и телекоммуникационные технологии, №14, 2012. – С. 31–36.
12. Алексейчук А. С. Программная система для обучения студентов экономических специальностей в режиме веб-конференции // Научный альманах. Вып. 21: Материалы XII научно-практической конференции молодых ученых и студентов «Инновации в экономике и менеджменте в аэрокосмической промышленности». – М.: «Доброе слово», 2016. – С. 190–198.
13. Алексейчук А. С. Системы дистанционного обучения на базе веб-конференций // Труды междунар. конф. «Информатизация инженерного образования» ИНФОРИНО-2012. – М.: МЭИ, 2012. – С. 343-344.
14. Алексейчук А. С. Технологии построения систем дистанционного обучения на базе веб-конференций // Материалы XI Международной конференции по неравновесным процессам в соплах и струях (NPNJ'2016), Алушта. – М.: Изд-во МАИ, 2016. – С. 547-549.
15. Алексейчук А. С., Пантелеев А. В. Индивидуализация процесса обучения в режиме веб-конференции на основе иерархической нечеткой экспертной системы. – «Информатика и её применения», том 11, выпуск 1, 2017. – С. 90-99.
16. Алексейчук А. С. Индивидуализация процесса обучения студентов в режиме веб-конференции при помощи интеллектуальной экспертной системы // Материалы XIX Международной конференции по вычислительной механике и современным прикладным программным системам (ВМСППС'2015), Алушта, 24–31 мая 2015 г.: Материалы конф. – М.: МАИ, 2015. – С. 707–709.
17. Алексейчук А. С. Интерактивная среда для решения математических задач в форме веб-конференции // Материалы XVIII Международной конф. по вычислительной механике и современным прикладным программным системам (ВМСППС'2013), Алушта. – М.: МАИ, 2013. – С. 830–832.
18. Алексейчук А. С. Применение веб-конференций для подготовки студентов Инжэкин к контрольным работам по математическим дисциплинам // Научн. альманах. Вып. 17: Мат. IX конф. «Инновационный менеджмент в аэрокосмической промышленности». – М.: «Доброе слово», 2013. – С. 174–179.

19. Алексейчук А. С. Применение экспертной системы для индивидуализации процесса обучения студентов в режиме веб-конференции // Научный альманах. Вып. 20: Материалы XI научно-практической конференции молодых ученых и студентов «Инновации в экономике и менеджменте в авиации». – М.: «Доброе слово», 2015. – С. 174–182.
20. Алексейчук А. С. Система поддержки онлайн-обучения на базе веб-конференций // Научно-техн. междунар. конф. «Системы, методы, техника и технологии обработки медиаконтента». Сборник тезисов. – МГУП им. И. Федорова, 2011. – С. 7-8.
21. Алексейчук А. С., Айсмонтас Б. Б., Пантелеев А. В. Телекоммуникационные системы дистанционного обучения // Проблемы авиастроения, космонавтики и ракетостроения: Сб. науч. тр. – М.: «Ваш полиграфический партнер», 2012. – С. 374–379.
22. Алтунин А.Е., Семухин М.В. Модели и алгоритмы принятия решений в нечетких условиях: Монография. Тюмень: Издательство Тюменского государственного университета, 2000. 352 с.
23. Благодатских В. А. Стандартизация разработки программных средств: учеб. пособие / Под ред. О. С. Разумова. – М.: Финансы и статистика, 2006.
24. Блюмин С.Л., Шуйкова И.А., Сараев П.В. Нечеткая логика: алгебраические основы и приложения. – Липецк: ЛЭГИ, 2002 г. – 111 с.
25. Борисов А.Н., Алексеев А.В., Крумберг О.А. и др. Модели принятия решений на основе лингвистической переменной. Рига: Зинатне, 1982. -256 с.
26. Бураков М. В., Коновалов А. С. Нейронечеткие системы управления // Информационно-управляющие системы, №1, 2002.
27. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя – М.: Изд-во ДМК, 2006.
28. Веда – бесплатная интерактивная СДО. [Электронный ресурс] // URL: <http://www.veda-soft.com/>
29. Власенко А. Н., Кучеренко Е. И. Иерархическая нечетко-вероятностная модель в задачах управления рисками – Обработка информации в сложных организационных системах. Харьков: Харьковский национальный университет радиоэлектроники, 2012.
30. Вятчинин Д. А. Нечеткие методы автоматической классификации – Минск: УП «Технопринт», 2004.
31. Гагарина Л. Г., Виснадул Б. Д., Игошин А. В. Основы технологии разработки программных продуктов: Учебное пособие. – М.: Инфра-М, 2006.

32. Гроцев А. Р. Протоколы передачи данных для систем дистанционного обучения // "Образовательные технологии и общество" (электронный журнал). – 2011.– №2. URL:
33. Жуков Л.А., Решетникова Н.В. Учебное пособие по дисциплине «Приложения нейронных сетей». Красноярск: ИПЦ КГТУ, 2007.
34. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений – М.:Мир, 1976.
35. Захаров Р. Е. Разработка логико-лингвистических моделей управления и принятия решений на базе нечеткой логики // Дисс. на соискание уч. степ. канд. техн. наук – Владикавказ, 2004.
36. Иванова Л. В. Разработка интегрированной среды формирования индивидуальной образовательной траектории при подготовке персонала для предприятий транспортного комплекса // Диссертация на соискание уч. степ. канд. техн. наук, М., 2004.
37. Ивлева Е. В. Разработка и исследование интеллектуальных контролирующих систем с настраиваемой нечеткой экспертной подсистемой выставления оценок // Диссертация на соискание уч. степ. канд. техн. наук, Рязань, 2004.
38. Каллан Р. Основные концепции нейронных сетей – Пер. с англ.: М.: Издательский дом «Вильямс», 2001. – 287 с.
39. Карл И. Вигерс. Разработка требований к программному обеспечению –М.: Русская редакция, 2004. – 576 с.
40. Комашинский В. И., Смирнов Д. А. Нейронные сети и их применение в системах управления и связи – М.: Горячая линия–Телеком, 2003. – 94 с.
41. Коськин А. В., Константинов И. С., Мельников А. А., Гаценко О. Б. Выбор технологии информационного обмена в телекоммуникационной образовательной сети Орловской области // Индустрия образования. Сб. науч. трудов. – М.: МГИУ, 2002. – Вып. 2. – С. 76-84.
42. Крючин О. В., Арзамасцев А. А. Параллельные алгоритмы обучения искусственной нейронной сети QuickProp и RPROP // Вестник ТГУ, т. 17, вып. 1, 2012. Тамбов.
43. Кузьмина С. Н., Долгих А. П. Организация обучающей среды образовательного комплекса на основе использования информационно-коммуникационных технологий // Ученые записки Санкт-Петербургской академии управления и экономики. – СПб., 2010. – №3.
44. Лузгина В.Б., Майстренко В.А., Шамец С.П. Видеоконференции в дистанционном обучении: от революции к эволюции. // Материалы XV Всероссийской научно-методической конференции «Телематика'2008».

45. Лупанов В.Н. Интерактивные видеоконференции в системе открытого образования: опыт, проблемы и перспективы // Проблемы современного образования. – Москва, 2010. –№2.
46. Матвеев Е.В., Глинчиков В.А. Нечеткий логический вывод в системе управления беспилотного летательного аппарата. – Журнал Сибирского федерального университета. Техника и технологии. 2011, №4 (1) – С. 79-91.
47. Наумов А. В., Джумурат А. С., Иноземцев А. О. Система дистанционного обучения математическим дисциплинам CLASS.NET // Вестник компьютерных и информационных технологий, №10, 2014.
48. Нефункциональные требования к ПО [Электронный ресурс] // URL: http://life-prog.ru/1_16778_nefunktsionalnie-trebovaniya.html
49. Нифагин В. А., Бокуть Л. В. Методические аспекты применения информационных технологий при изучении математики в техническом вузе // Материалы II международной научно-практической конференции «Современные информационные технологии и ИТ-образование». М.: МГУ, 2006.
50. Орловский С.А. Проблемы принятия решений при нечеткой исходной информации. М: Наука, 1981. 203с.
51. Панарин С. И. Математическое и программное обеспечение системы дистанционного обучения по математическим дисциплинам // Диссертация на соискание уч. степ. канд. ф.-м. наук, Москва, 2011.
52. Полковникова Н. А., Курейчик В. М. Разработка модели экспертной системы на основе нечёткой логики // Известия ЮФУ. Технические науки, №1, 2014.
53. Программные проекты. Цикл статей по управлению проектами [Электронный ресурс] // URL: <http://project.dovidnyk.info/>
54. Прометей – система дистанционного обучения [Электронный ресурс] // URL: <http://www.prometeus.ru/>
55. Пятнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2016 (3–7 октября 2016 г., г. Смоленск). Труды конференции. В 3-х томах. Т 2. – Смоленск: Универсум, 2016. – 374 с.
56. Реунова Л. В. Развитие дистанционных технологий в образовании // Материалы II Всероссийской научно-методической конференции «Инновации в системе высшего образования». – Челябинск, 2011.
57. Ротштейн А.П., Штовба С.Д. Влияние методов дефаззификации на скорость настройки нечеткой модели // Кибернетика и системный анализ, 2002. - № 5. - С.169-176.

58. Рубанов В. Г., Филатов А. Г., Рыбин И. А. Интеллектуальные системы автоматического управления. Нечеткое управление в технических системах - БГТУ им. В. Г. Шухова. Электронное пособие. URL: <http://nrsu.bstu.ru/>
59. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. - М.: Горячая линия -Телеком, 2006. - 452 с.
60. Рыбина, Г. В. Основы построения интеллектуальных систем: учебное пособие – М.: Финансы и статистика, 2010. - 432 с.
61. Рыжов А. П. Элементы теории нечетких множеств и измерения нечеткости – М.: Диалог-МГУ, 1998.
62. Сахнюк П. А. Нечеткие отношения и операции над ними // Лекции по нечеткой логике и нейронным сетям, Ставропольский аграрный университет [Электронный ресурс] // URL: <http://www.stgau.ru/company/personal/user/7684/files/element/view/96632>
63. Система дистанционного обучения «WebTutor» [Электронный ресурс] // URL: http://www.websoft.ru/db/wb/root_id/webtutor/doc.html
64. Система дистанционного обучения «Доцент» [Электронный ресурс] // URL: <http://www.uniar.ru/docent.php>
65. Сологуб Г. Б. Разработка математических методов и комплекса программных средств имитационного тестирования знаний на основе семантических моделей // Дисс. на соискание уч. степ. канд. ф.-м. наук, Москва, 2013.
66. Спенсер П. XML. Проектирование и реализация – М.: Лори, 2001 г. 510 с.
67. Старыгин А. XML: разработка Web-приложений – СПб.: BHV, 2003. 592 с.
68. Трайнёв В.А., Трайнёв О.В. Обзор систем дистанционного обучения в России и за рубежом // Информационные и телекоммуникационные технологии. – 2009.– № 9.
69. Трофимец Е.Н. Информационные технологии математического моделирования в экономических вузах // "Образовательные технологии и общество" (электронный журнал). – 2012.– №1.
70. Устюгова В.Н., Валитов Р.А. О процессе создания системы дистанционного обучения в Татарском государственном гуманитарно-педагогическом университете (ТГГПУ) // "Образовательные технологии и общество" (электронный журнал). – 2010.– №2. URL:
71. Хайкин С. Нейронные сети: полный курс – М.: Вильямс, 2006. – 1104 с.

72. Чванова М. С. Проблемы организации коммуникаций студентов наукоемких специальностей в системе открытого образования // "Образовательные технологии и общество" (электронный журнал). – 2011.– №2.
73. Шнейдер В. Е., Слуцкий А. И., Шумов А. С. Краткий курс высшей математики. Учеб. пособие для втузов. М., «Высш. школа», 1972. 640 с.
74. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику [Электронный ресурс] // URL: <http://matlab.exponenta.ru/fuzzylogic/book1>
75. Ястребова Н. Н. Исследование и разработка нечеткой модели и комплекса программ экологической экспертизы горнодобывающего производства // Диссертация на соискание уч. степ. канд. техн. наук, Ульяновск, 2008.
76. Adobe Connect™ – web conferencing software [Электронный ресурс] // URL: <http://www.adobe.com/products/adobeconnect.html>
77. Alexander, W., Higgison, C., Moge, N. Videoconferencing in teaching and learning – Case studies. LTDI and TALiSMAN, Institute for Computer Based Learning, Heriot-Watt University, Edinburg, 1999. URL: <http://www.icbl.hw.ac.uk/ltdi/vcstudies/vcstudies-all.pdf>
78. Anil K. Jain, Jianchang Mao, K.M. Mohiuddin. Artificial Neural Networks: A Tutorial, Computer, Vol.29, No.3, March/1996, pp. 31-44
79. Babuska R. Fuzzy Systems, Modeling and Identification – Delft University of Technology, 1996.
80. Berg J. Financial markets analysis by using a probabilistic fuzzy modeling approach / J. van den Berg, U. Kaymak, W.-M. van den Bergh // International Journal of Approximate Reasoning. – Elsevier. – 2004. – №35. – P. 291-305.
81. Christian Igel, Michael Husken. Empirical evaluation of the improved Rprop learning algorithms. – Neurocomputing 50 (2003), с. 105–123.
82. Comdi: агентство виртуальных мероприятий, занимающееся организацией онлайн-трансляций, виртуальных конференций [Электронный ресурс] // URL: <http://www.comdi.com/>
83. DeMarco T. Structured Analysis and System Specification. Englewood Cliffs: Prentice-Hall, 1979.
84. Fonseca C., Araújo F., Rodrigues M. Hierarchical Fuzzy Control. – Fuzzy Controllers – Recent Advances in Theory and Applications, 2012.
85. Fraternali P., Rossi G. Rich internet applications // IEEE Internet Computing (Volume: 14, Issue: 3, May-June 2010)
86. Haefner, J. Opinion: The Importance of Being Synchronous. Academic Writing.

87. Hartl M. Ruby on Rails Tutorial: Learn Web Development with Rails – Addison-Wesley, 2012. 600 p.
88. Hartog, den, M.H., R. Babuška, H.J.R. Deketh, M. Alvarez Grima, P.N.W. Verhoef and H.B. Verbruggen. Knowledge-based fuzzy model for performance prediction of a rockcutting trencher – International Journal of Approximate Reasoning 16(1), 1997, c. 43–66.
89. Huang, Shaobo, "Predictive Modeling and Analysis of Student Academic Performance in an Engineering Dynamics Course" (2011). – All Graduate Theses and Dissertations. Paper 1086.
90. Ibrahim A. Hameed and Claus G. Sorensen. Fuzzy Systems in Education: A More Reliable System for Student Evaluation, Fuzzy Systems, Ahmad Taher Azar (Ed.) INTECH, Croatia, 2010.
91. Ibrahim Saleh *, Seong-in Kim. A fuzzy system for evaluating students' learning achievement. Expert Systems with Applications 36 (2009).
92. Kelly L. Wilkinson, K. Virginia Hemby. An Examination of Perceptions of the Use of Virtual Conferences in Organizations: The Organizational Systems Research Association (OSRA) and The Association for Business Communication (ABC) Members Speak Out. // Information Technology, Learning, and Performance Journal. – 2000. – Vol. 18, No. 2. – P. 13-23.
93. Kosko B. Fuzzy systems as universal approximators // IEEE Transactions on Computers, vol. 43, No. 11, November 1994.-P. 1329-1333.
94. Leff A., Rayfield J. T. Web-application development using the Model/View/Controller design pattern // Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference, Seattle, WA, 2001, pp. 118-127.
95. MacIntosh J. Learner concerns and teaching strategies for video-conferencing // The Journal of Continuing Education in Nursing. November 2001. Volume 32. Issue 6.
96. Marlene Jones. Applications of artificial intelligence within education. - Computers & Mathematics with Applications –Volume 11, Issue 5, May 1985, Pages 517-526.
97. Martin Riedmiller, Heinrich Braun. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. Institut für Logik, Komplexität und Deduktionssysteme, University of Karlsruhe, 1993.
98. Michael M. Grant, Jongpil Cheon. The Value of Using Synchronous Conferencing for Instruction and Students. // Journal of Interactive Online Learning. 2007. – Vol. 6, No. 3. – P. 211-224.

99. Ming-Ling Lee, Hung-Yuan Chung, Fang-Ming Yu. Modeling of hierarchical fuzzy systems. – *Fuzzy Sets and Systems*, 138 (2003), С. 343 – 361
100. Moodle – система управления обучением [Электронный ресурс] // URL: <https://moodle.org/>
101. Myers B. A brief history of human-computer interaction technology // *Interactions*, Volume 5 Issue 2, March/April 1998.
102. Oladokun, V.O., A.T. Adebajo, and O.E. Charles-Owaba. 2008. “Predicting Students’ Academic Performance using Artificial Neural Network: A Case Study of an Engineering Course”. *Pacific Journal of Science and Technology*. 9(1):72-79.
103. OpenMeetings: open source web conferencing [Электронный ресурс] // URL: <https://openmeetings.apache.org/>
104. Passino K. M., Yurkovich S. *Fuzzy control* – Addison Wesley Longman, 1998. 502 p.
105. Reby D. Artificial neural networks as a classification method in the behavioural sciences // *Behavioural Processes* 40, 1997, p. 35–43.
106. Richardson C. *ORM in Dynamic Languages* // *Queue*. Vol. 6, issue 3, May 2008.
107. Sanders W. *Learning Flash Media Server 3* – "O'Reilly Media, Inc.", 2008. 273 p.
108. Sher, Ali. Assessing the relationship of student-instructor and student-student interaction to student learning and satisfaction in Web-based Online Learning Environment. // *Journal of Interactive Online Learning*. – 2009. – Vol. 8, No. 2. – P. 102-117.
109. Skype – приложение для видеосвязи и веб-конференций. [Электронный ресурс] // URL: www.skype.com
110. Stathacopoulou, R. and Magoulas, George D. and Grigoriadou, M. and Samarakou, M. (2005) Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis. *Information Sciences* 170 (2-4), pp. 273-307.
111. Sugeno M., Yasukawa T. A fuzzy-logic-based approach to qualitative modeling – *IEEE Transactions on Fuzzy Systems* 1, 1993. – С. 7–31.
112. Sugeno, M., Kang G. T. Fuzzy modelling and control of multilayer incinerator. – *Fuzzy Sets and Systems* 18, 1986. – С. 329
113. Sugeno, M., Tanaka K. Successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems* 42, 1991. – С. 315-334.
114. Su-Houn Liu. Applying the Technology Acceptance Model and Flow Theory to Online E-Learning Users’ Acceptance Behavior. – *Issues in Information Systems*. Volume VI, No. 2, 2005.

115. Takács M. Soft Computing-Based Risk Management - Fuzzy, Hierarchical Structured Decision-Making System / Márta Takács // Risk Management Trends. – InTech Europe. – 2011. – P.28-46.
116. Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on Systems, Man and Cybernetics 15(1). – С. 116–132.
117. Thomas D. Agile Web Development with Rails – Pragmatic Bookshelf, 2006. 719 p.
118. Thomas D. Programming Ruby 1.9: The Pragmatic Programmers' Guide. – The Pragmatic Programmers LLC, 2010. 863 p.
119. TrainingWare. Системы дистанционного обучения [Электронный ресурс] // URL: <http://www.tadviser.ru/>
120. TrueConf – решение для корпоративных коммуникаций [Электронный ресурс] // URL: <https://trueconf.ru/>
121. Tsekouras G. A hierarchical fuzzy-clustering approach to fuzzy modeling / G. Tsekouras, H. Sarimveis, E. Kavakli, G. Bafa // Fuzzy Sets and Systems. – Elsevier. – 2005. – №107. – P. 245-266
122. Wang, LX. Analysis and design of hierarchical fuzzy systems – I-EE transactions on Fuzzy Systems, (Volume:7 , Issue: 5) 1999.
123. WiziQ: Online learning delivery platform for universities [Электронный ресурс] // URL: <https://www.wiziq.com/>