

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

**РАЗРАБОТКА МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ И
АВТОМАТИЗИРОВАННЫХ СРЕДСТВ ПОДДЕРЖКИ
ИНДИВИДУАЛЬНО-ОРИЕНТИРОВАННОГО ОБУЧЕНИЯ**

Магистерская диссертация
обучающегося по направлению подготовки
02.04.01 «Математика и компьютерные науки»
очной формы обучения,
группы 07001531
Войтовой Юлии Юрьевны

Научный руководитель:
к.т.н., доцент В.В. Муромцев

Рецензент:
к.т.н., доцент В.В. Ломакин

БЕЛГОРОД 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 ИССЛЕДОВАНИЕ СУЩЕСТВУЮЩИХ ПОДХОДОВ К ОРГАНИЗАЦИИ ИНДИВИДУАЛЬНО-ОРИЕНТИРОВАННОГО ОБУЧЕНИЯ	8
1.1 Исследование подходов учета индивидуальных показателей обучающихся в системах электронного обучения	8
1.2 Анализ систем электронного обучения	20
2 ИССЛЕДОВАНИЕ СИСТЕМЫ ОБУЧЕНИЯ НИУ «БЕЛГУ».....	29
2.1 Исследование особенностей образовательного процесса НИУ «БелГУ»	29
2.2 Исследование системы электронного обучения НИУ «БелГУ».....	33
3 РАЗРАБОТКА ФОРМАЛЬНО-АЛГОРИТМИЧЕСКОГО ОБЕСПЕЧЕНИЯ ПОДДЕРЖКИ ПРОЦЕССА ИНДИВИДУАЛЬНОГО ОБУЧЕНИЯ	38
3.1 Проектирование процесса индивидуально-ориентированного обучения	38
3.2 Разработка базы знаний для системы индивидуально-ориентированного тестирования обучающихся	46
3.3 Построение алгоритма повышения качества тестовых заданий.....	52
3.4 Совершенствование базы данных системы электронного обучения НИУ «БелГУ».....	58
4 РЕАЛИЗАЦИЯ РАЗРАБОТАННЫХ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ И АЛГОРИТМОВ ИНДИВИДУАЛЬНО-ОРИЕНТИРОВАННОГО ОБУЧЕНИЯ.	66
4.1 Настройка и разработка модулей индивидуально-ориентированной системы обучения	66
4.2 Тестирование работы модулей индивидуально-ориентированной системы обучения	69

4.3 Оценка эффективности использования в СЭО алгоритмического и математического обеспечения индивидуально-ориентированной системы обучения.....	71
ЗАКЛЮЧЕНИЕ	73
СПИСОК ЛИТЕРАТУРЫ.....	76
ПРИЛОЖЕНИЯ.....	82
ПРИЛОЖЕНИЕ А	82
ПРИЛОЖЕНИЕ Б.....	105
ПРИЛОЖЕНИЕ В	121
ПРИЛОЖЕНИЕ Г.....	128

ВВЕДЕНИЕ

В настоящее время образование все больше становится капиталом, инструментом борьбы за рынок и решения геополитических задач. Состояние образования определяет интеллектуальный потенциал общества и на данный момент, и на длительную перспективу, создает предпосылки устойчивого социально-экономического прогресса. Все это приводит к необходимости проведения передовыми странами новой образовательной политики, опирающейся на «инженерный подход» к разработке системы образования. При этом активно используются различные стратегии и средства моделирования, реструктуризации, усовершенствования [7].

Современное образование стремится к применению индивидуально-ориентированного подхода к образовательному процессу на всех уровнях. Технология индивидуализированного обучения предполагает организацию учебного процесса, при которой индивидуальный подход и индивидуальная форма обучения являются приоритетными.

Главное достоинство данной модели обучения - полная адаптация содержания, методов и темпов учебной деятельности обучающегося к его особенностям, следить за его действиями и операциями при решении конкретных задач; следить за его продвижением от незнания к знанию, вносить вовремя необходимые коррективы в деятельность как обучающегося, так и тьютора, приспособлять их к постоянно меняющейся, но контролируемой ситуации со стороны обучающего и со стороны ученика.

Индивидуализацию обучения можно считать «проникающей технологией», т.к. она может быть реализована во всех существующих образовательных технологиях. Однако технологии, делающие индивидуализацию основным средством достижения целей обучения, можно рассматривать как самостоятельную систему, обладающую всеми качествами и признаками целостной педагогической технологии.

Одним из наиболее популярных и актуальных направлений в сфере образования является индивидуально-ориентированный подход к процессу обучения, который позволяет обучающемуся работать экономно, контролировать затраты своих сил, работать в оптимальное для себя время, что способствует достижению высоких результатов обучения. На сегодняшний день разработано и предложено очень много разработок и даже прикладных решений, обзор ряда которых можно изучить в работах: Астанин, С.В., Асадуллаев, Р.Г., Брусиловский П. Л., Олейник Н.М., Попов, Д.И.. Несмотря на популярность данного направления, вопросы учета индивидуальных особенностей обучающихся при построении индивидуальной образовательной траектории обучающегося по-прежнему остаются актуальными.

Отдельный интерес в данном направлении представляет образовательная деятельность высших учебных заведений. Преподавателям затруднительно применять индивидуальный подход к каждому студенту, т.к. численность академической студенческой группы, в среднем, около 20 человек, а для применения индивидуального подхода группа обучающихся должна составлять около 5-7 человек. Кроме того, дистанционное образование становится популярнее с каждым годом, а вот технологии его реализации в ВУЗах очень далеки от адаптивных подходов к процессу обучения. Таким образом, необходима реализация автоматизированных систем индивидуального-ориентированного обучения, работающих на основе адаптивных алгоритмов, способных формировать индивидуальные образовательные траектории за счет учета расширенного набора параметров обучаемых (психофизиологических характеристик, уровня фундаментальных базовых знаний, механизмов адаптивного тестирования, специфики восприятия информации и т.д.).

Объект исследования – система электронного обучения НИУ «БелГУ».

Предмет исследования – математические модели и автоматизированные средства поддержки системы электронного обучения НИУ «БелГУ».

Цель работы: повышение эффективности образовательного процесса НИУ «БелГУ», посредством учета индивидуальных особенностей обучающихся в системе электронного обучения.

Задачи:

- исследование существующих подходов к организации индивидуально-ориентированного обучения;
- исследование системы обучения НИУ «БелГУ»;
- проектирование формально-алгоритмических средств поддержки процесса индивидуального обучения;
- реализация разработанных моделей и алгоритмов индивидуально-ориентированного обучения.

Методы и средства исследований. При решении указанных задач использовались методы и средства теории множеств, теории систем, теории графов, сети Петри.

Достоверность научных положений и полученных результатов подтверждается корректным использованием алгоритмического обеспечения и формальных средств. Полученные результаты не противоречат существующим разработкам в области разработки систем индивидуально-ориентированного обучения.

Научная новизна диссертационного исследования:

- разработаны при помощи сетей Петри укрупненная и декомпозиционные схемы процесса индивидуально-ориентированного обучения;
- разработана нечеткая база знаний адаптивного тестирования;
- разработан алгоритм оценки качества тестовых заданий при малой выборке испытуемых;

– осуществлено совершенствование базы данных системы индивидуально-ориентированного тестирования.

Практическая ценность работы заключается в том, что:

– разработанные модели и алгоритмы индивидуально-ориентированного обучения могут быть реализованы в разных системах электронного обучения

– реализованные алгоритмы и модели могут быть использованы преподавателями кафедр ФГАУ ВО НИУ «БелГУ» для реализации индивидуально-ориентированного обучения.

Положения, выносимые на защиту:

– схемы процесса индивидуально-ориентированного обучения;

– нечеткая база знаний адаптивного тестирования;

– алгоритм оценки качества тестовых заданий при малой выборке испытуемых;

– база данных системы индивидуально-ориентированного тестирования.

– реализация предложенных моделей и алгоритмов в системе.

Публикации. По теме исследования подготовлено 2 тезиса доклада.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, списка литературы, приложений. Работа изложена на 130 страницах машинописного текста, включающего 23 рисунка, 28 таблиц, список литературы из 48 наименований, 4 приложений.

1 ИССЛЕДОВАНИЕ СУЩЕСТВУЮЩИХ ПОДХОДОВ К ОРГАНИЗАЦИИ ИНДИВИДУАЛЬНО-ОРИЕНТИРОВАННОГО ОБУЧЕНИЯ

1.1 Исследование подходов учета индивидуальных показателей обучающихся в системах электронного обучения

Одной из главных проблем компьютерного обучения является создание обучающих программ, обеспечивающих адаптивную организацию содержания учебного курса, образовательных траекторий и режимов взаимодействия ученика с машиной [8].

Выделяют такие виды технологий в индивидуально-ориентированных обучающих системах, как:

- построение последовательности курса обучения;
- интеллектуальный анализ ответов обучаемого;
- интерактивная поддержка практических заданий;
- основанная на примерах помощь в решении задач [8].

При этом все системы электронного обучения должны реализовывать основные функции:

- оперативная доставка учебной информации обучающемуся;
- осуществление взаимосвязи между всеми участниками учебного процесса и обратной связи с преподавателем;
- обеспечение индивидуальной и групповой самостоятельной работы [25].

Структурную схему процесса индивидуально-ориентированного обучения можно представить следующим образом (рис. 1.1):



Рис. 1.1. Структурная схема процесса адаптивного обучения

Так, индивидуально-ориентированное обучение предполагает применение в системах электронного обучения технологий адаптивного планирования или технологий адаптивной гипермедиа. Адаптивное планирование предполагает построение индивидуальных образовательных траекторий на основе интеллектуального анализа истории обучения каждого пользователя. При этом последовательность модулей знаний для обучения определяется также по результатам проведенного анализа. Данная технология помимо перечисленного предполагает возможность интерактивного решения задач (когда обучаемому предоставляется интеллектуальная помощь в виде подсказок, формирующихся на основании действий совершаемых обучаемым на каждом шаге решения задания), решения задач на примерах и совместной работы (используется в сетевых обучающих системах, когда формируются группы для совместного решения задач или системой находится человек с моделью, показывающей хорошее знание этой темы) [12].

Для реализации метода ставится задача построения оптимальной последовательности изучения курса. В системах электронного обучения, контент курса представляется в виде трех- или четырехуровневой иерархии: курс, тема, раздел, элементарный блок знаний, причем уровень разделов для небольших и несложных тем может быть опущен. Элементарный блок знаний - это элементарная порция информации: текст изучения какого-либо понятия темы, вопрос, задача, комментарий, пояснительный текст и т.п. [18]

Построение оптимальной последовательности элементов контента осуществляется на основе модели учебного материала, в качестве которой используется ориентированный граф с нагруженными ребрами $G(V, S)$ (рис. 1.2). Множество V вершин графа соответствует элементам контента (курсам, темам, разделам или элементарным блокам знаний), а множество S ребер - связям между ними. Возможны следующие степени связи:

S_1 - для изучения объекта необходимо иметь общее понятие о другом элементе контента;

S_2 - при изучении объекта используются частые ссылки на другой элемент контента;

S_3 - для изучения наиболее сложных (или редко используемых) понятий объекта необходимы знания из другого элемента контента;

S_4 - для изучения объекта и практического применения знаний необходимо четкое знание другого элемента контента [15].

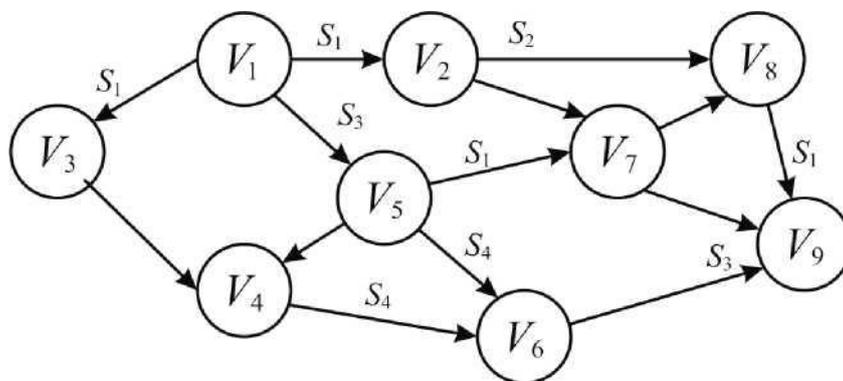


Рис. 1.2. Семантическая модель учебного материала

Каждой вершине i графа ставится в соответствие вектор $V_i = [R_n, R_{i2}, \dots, R_{im}]$, каждый элемент которого включает четыре параметра, т.е.

$$R_{ij} = (P_{ij}, t_{ij}, z_{ij}, u_{ij}, q_{ij}) \rightarrow O1$$

где i - номер элемента контента; j – номер образовательной программы обучающегося; P_{ij} – образовательная программа студента; t_{ij} - время изучения элемента контента; Z_{ij} - уровень знаний, который должен быть достигнут при изучении элемента; u_{ij} - уровень умений; q_{ij} - уровень навыков, которые должны быть сформированы в результате изучения контента [5,6].

Последние три параметра являются показателями цели обучения. Подграф G_k графа G , включающий вершины с одинаковыми значениями r_{ij} , является моделью контента курса для группы студентов, обучающихся по специальности r_k . На рис. 1.3 показан подграф с вершинами $V_1, V_2, V_4, V_5, V_6, V_7$. Такой подход позволяет осуществить адаптацию к разным группам студентов в зависимости от их специальности и (или) программы обучения.

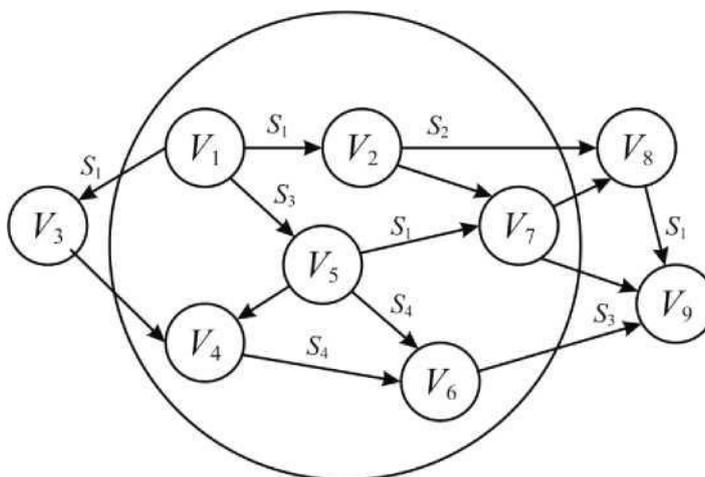


Рис. 1.3. Подграф G_k графа G как модель УМ для группы студентов, обучающихся по специальности r_k

Решение о включении новых элементарных блоков знаний принимается на основе модели студента, которая в общем случае, хотя и необязательно, является вектором [29]:

$$M = \{M_1, M_2, M_3, \dots, M_n\}$$

и может включать, к примеру, следующие компоненты-векторы: М1 - предыстория обучения; М2 - результаты текущей работы с курсом (тип выполненных заданий, время выполнения заданий, число обращений за помощью и т.д.); М3 - личностные психологические характеристики (тип и направленность личности, репрезентативная система, способность к обучению, уровень беспокойства тревоги, особенности памяти и др.); М4 - опыт работы с компьютерной системой; М5 - общий уровень подготовленности; М6 - стратегия обучения и др.

Область адаптивной гипермедиа включают в себя ряд решений по адаптивному представлению обучающих курсов и организации индивидуализированной навигации в процессе обучения [13].

Метод адаптивного представления информации предусматривает генерацию наиболее подходящих для студента блоков знаний и связан с видом и детальностью представления информации. Выбор вида и детальности информации осуществляется на основе модели студента (М2, М3, М5). Адаптивное представление подразумевает нахождение гипермедиа страниц в динамичном состоянии. При изменении модели обучаемого, меняется и содержание контента страницы [2].

Информация в процессе обучения может быть представлена в разном виде: текст, графика, анимация, звук и т.д. По форме восприятия информации различают людей с тремя системами (М3): аудиалы, визуалы и кинестики. Для аудиалов, необходима звуковая интерпретация учебного контента. Кинестики получают больше информации через практические действия. Для визуалов информация должна быть представлена в виде изображений, а текст должен быть визуально структурированным. Так, осуществляется адаптация в зависимости от вида представления информации. Но немаловажно предусмотреть в системе такое свойство адаптируемости, как обеспечение выбора вида выводимой информации, т.е. возможности включать и отключать звук, анимацию и презентацию [4,5].

Более детальное разделение и выдача информации зависит от текущего выполнения заданий (M2) и общего уровня подготовленности (M5) студента. Так, для более подготовленных учащихся достаточно демонстрирующего понимание материала краткого изложения изучаемого контента, для остальных нужна пояснительная информация различной степени детальности. Выбор степени детальности выдаваемой информации также можно предоставлять студенту [19].

Адаптивная навигация предполагает индивидуальную подстройку видимости ссылок на учебный контент, возможность адаптивной сортировки и аннотирования. Данная технология направляет обучаемого, разрешая перейти по той или иной ссылке, но не предлагает четкого плана переходов к тем или иным блокам курса [1,30].

Выдача комментариев – составная часть ряда методов индивидуализации (интеллектуальный анализ решений, диалоговая поддержка решения задач, решение задач на примерах, адаптивная поддержка сотрудничества). Комментарии, выдаваемые обучаемому, формируются и выводятся в зависимости от контекста и с учетом психологических характеристик студента, взятых из модели студента *M*. Различают такие виды адаптивных комментариев, как: реплика, помощь, разъяснение, примеры [26].

Еще один метод индивидуализации обучения – интеллектуальный анализ решений. При его использовании происходит анализ ответов студента с использованием интеллектуальных анализаторов, когда распознаванию подлежат не только правильные, но и неточные, неполные или полностью неправильные ответы. Такая методика позволяет определить, что именно не знает или не понимает студент, в результате чего выдаются соответствующие комментарии-разъяснения, а также генерируется дальнейший ход диалога. Так, для использования метода необходимо, во-первых, тщательно подобрать задания для проверки знаний и умений студента и подготовить комментарии

на все возможные варианты ответов и, во-вторых, разработать интеллектуальный анализатор [27].

Анализ индивидуализирующих технологий обучения, с точки зрения применения их в различных направлениях работы систем, представлен в таблице 1.1

Таблица 1.1

Технологии адаптации и основные направления работы обучающих систем

Технологии адаптации \ Направления работы	Адаптивное представление	Адаптивное планирование	Адаптивная поддержка навигации	Поддержка решения задач	Интеллектуальный анализ решения	Поддержка совместной работы
1	2	3	4	5	6	7
Представление	+	-	-	-	-	-
Оценка: тесты	+	+	+	-	-	-
Оценка: задания	+	+	+	+	+	+
Коммуникация	-	-	-	-	-	+
Управление: ограничение логики курса	+	+	+	-	-	-
Управление: руководство	+	+	+	-	-	-

Наиболее применимой ко всем направлениям работы систем обучения является технология адаптивного представления материала. Функция коммуникации обучающихся в ходе образовательного процесса поддерживается только технологией поддержки совместной работы. Можно сделать вывод о необходимости использования сразу нескольких адаптивных технологий для получения наиболее эффективного образовательного результата [6].

Реализация индивидуально-ориентированного подхода к обучению при помощи рассмотренных методов и технологий, подразумевает наличие соответствующего алгоритмического обеспечения. Так, один из главных вопросов организации индивидуальных образовательных траекторий –

определение подхода к реализации индивидуальной модели обучаемого. Не менее важна организация контрольных мероприятий по проверке знаний [44].

Одной из методологических основ построения модели обучаемого, является модель Раша, отражающая связь успеваемость обучающегося с уровнем его подготовленности и трудностью задания [29]. В модели подробно описываются способы определения дифференцирующей способности тестов, что важно при стремлении к объективной оценке знаний обучающихся и разделения их по группам успеваемости. Для расчета дифференцирующей способности теста учитываются такие характеристики, как валидность и надежность, которые рассматривает Item Response Theory [3,29].

Расчет латентных переменных обычно предусматривает участие не менее ста испытуемых, при этом, в результате испытаний и корректировок теста, число заданий в нем должно составить около 40-60. Наиболее устоявшийся метод расчета надежности тестовых заданий, применяемый в рассматриваемой методике, - построение матриц и вычисление коэффициента корреляции, что более подробно можно рассмотреть в работах Атанова «Обучение и искусственный интеллект, или основы современной дидактики» [8, 28].

Приемы расчета надежности теста совершенствуются с каждым годом, включая в себя все новые и новые критерии. Так, вместо трех основных понятий используемых ранее: знания, умения, навыки, теперь вводится интегрирующее их единое понятие – компетенция. Т.е. несмотря на смену понятий, на тестировании обучающийся должен продемонстрировать все три компонента. Отсюда возникает важность функциональной и содержательной валидности тестовых заданий [45].

Тесты становятся более адаптивно направленными за счет их разделения на уровни сложности, а также организации поэтапного тестирования при помощи ряда тематических тестов. Так, контрольная

оценка напрямую зависит от уровня сложности выполненных тестовых заданий.

Валидность и надежность тестов могут быть использованы для построения индивидуальных маршрутов обучения каждого из пользователей системы тестирования.

Родин из вариантов проверки надежности контрольных тестовых заданий - алгоритм Шана М.А., основанный на методе корреляции (рис. 1.4):

1) Ввод n (количество испытуемых), m (количество тестовых заданий), A (матрица результатов тестирования), B (матрица эталонных значений).

2) Делим тест на равные части (X и Y),

3) Для каждого очередного задания группы X определяем число тестированных, которые дали правильный ответ на текущее j -ое задание и затем находим их средний балл x_j .

4) То же самое для заданий группы Y .

5) Находим коэффициент корреляции X и Y : по формуле:

$$r_{XY} = \frac{\sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{\sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2} \cdot \sqrt{\sum_{i=1}^n y_i^2 - \frac{1}{n} (\sum_{i=1}^n y_i)^2}}$$

6) Находим надежность r рассматриваемого теста, используя формулу Спирмена-Брауна:

$$r = \frac{2r_{XY}}{1 + r_{XY}}$$

7) Вывод результатов.

8) Конец алгоритма. [43]

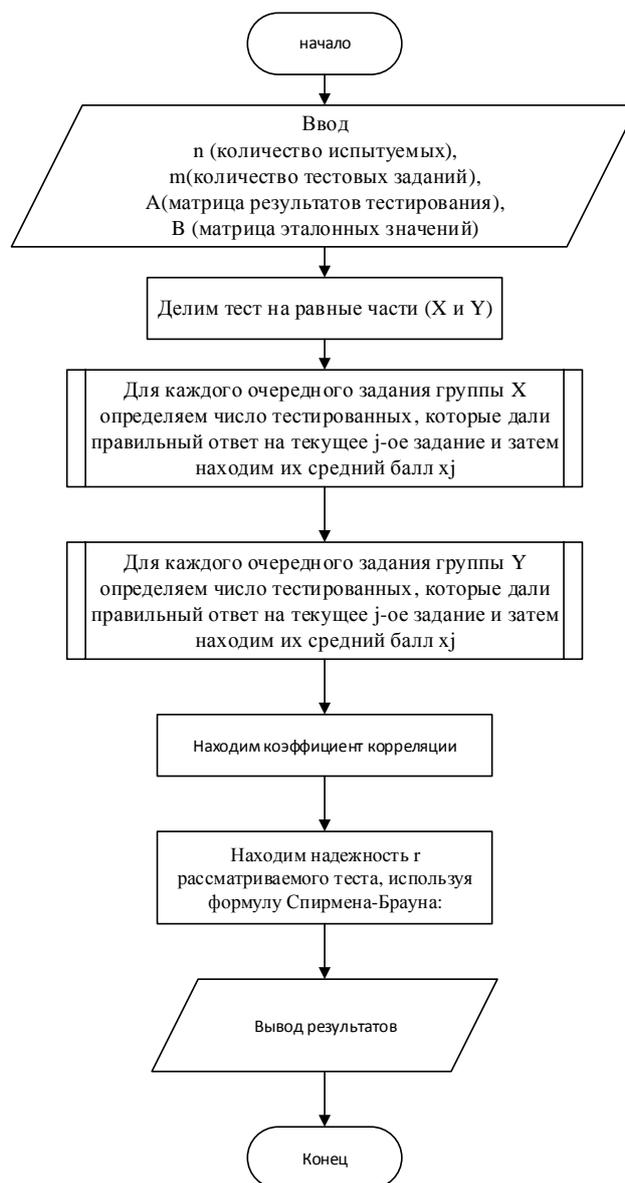


Рис. 1.4. Алгоритм расчета надежности теста

При построении индивидуальной модели обучаемого и организации адаптивного тестирования, необходимо определение текущего уровня подготовки обучаемого. Рассмотрим пример алгоритма, предназначенного для разбиения обучающихся на группы по уровням подготовки, где величина x – норма для тестовых результатов и величина k – масштабный коэффициент (см. рис. 1.6).

1) Ввод данных: n , m , A , k .

2) Для каждого из тестированных определяем его суммарный балл:

$$b_i = \sum_{j=1}^m a_{ij}$$

3) Разбиваем рассматриваемую выборку тестируемых на 3 группы. В первую группу попадают с высокими баллами: суммарный балл для попадающих в эту группу не ниже значения выражения

$$\min\{b_i\} + k(\max\{b_i\} - b_i)$$

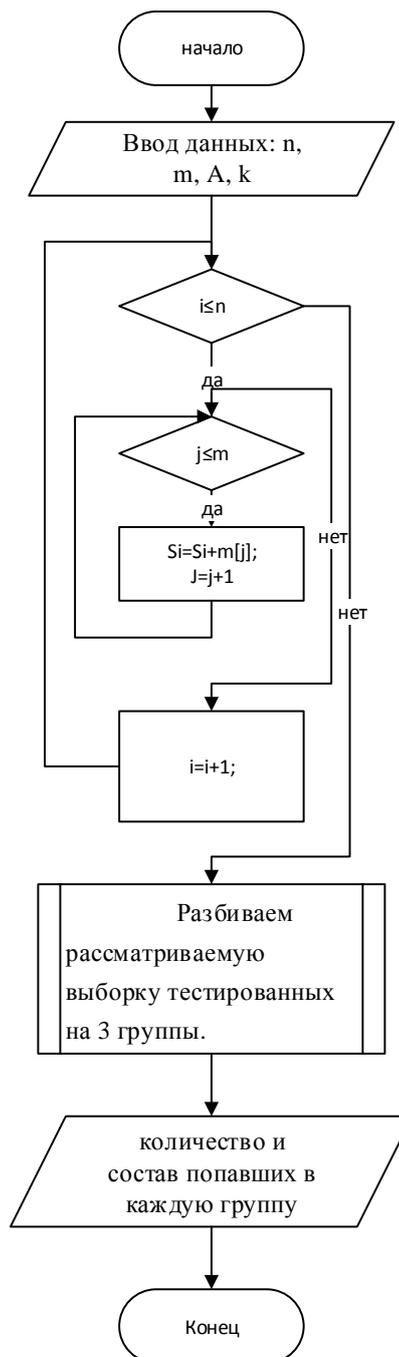


Рис. 1.5. Алгоритм разбиения обучающихся на группы по уровням подготовки

4) В третью группу попадают с низкими баллами – не выше значения выражения

$$\min\{b_i\} + (1 - k)(\max\{b_i\} - b_i)$$

5) Во вторую группу попадают все остальные (со средними баллами).

6) Выдача результатов: количество и состав попавших в каждую группу.

7) Конец алгоритма.

Работа индивидуально-ориентированных систем обучения предусматривает еще множество других алгоритмических решений, направленных, как на автоматизацию уже рассмотренных элементов системы, так и ряда других: формирование учебного контента, работа адаптивной навигации, поддержка решения задач, учет психологических особенностей обучаемого (темперамент, уровень восприятия, тип мышления) и т.д. [42]

Таким образом, в ходе исследования был проведен анализ индивидуально-ориентированных автоматизированных систем и адаптивных технологий, которые в них используются. Рассмотрен ряд адаптивных алгоритмов, использующихся при разработке индивидуально-ориентированных обучающих систем. Данные алгоритмы являются отличным решением для больших групп, однако требуют доработки, если речь пойдет о малой выборке испытуемых при оценке надежности теста. Приведенные примеры касались определения надежности тестовых заданий, распределения обучающихся по группам уровней сложности, построения процесса адаптивного тестирования. Результаты проведенного исследования алгоритмического обеспечения существующих систем обучения будут использованы при совершенствовании электронного обучения НИУ «БелГУ».

1.2 Анализ систем электронного обучения

В настоящее время, большая часть университетов использует возможности электронного обучения для организации учебного процесса разных форм обучения. Высшие учебные заведения разрабатывают свои системы обучения либо приобретают лицензии на использование настраиваемых образовательных платформ. Набирают популярность интернет-университеты, которые позволяют окончить по только отдельно взятые курсы, но и полноценные образовательные программы по направлениям подготовки. Функционал электронных систем обучения студентов может быть, как ограничен универсальной подачей материала и возможностью проведения контрольных мероприятий, так и расширен до использования адаптивных технологий [39].

Так, в таблице 1.2 приведены типы систем электронного обучения, использование которых повышает эффективность достижения различных уровней усвоения знаний, и мотивации к обучению [14, 41].

Наиболее популярными аналогами систем электронного обучения, использующихся сегодня являются: Moodle, E-front, LAMS, Sakai, ATutor, Claroline, Dokeos, OLAT, OpenACS, ILIAS.

В последние годы приобрела популярность программная среда MOODLE (Modular Object-Oriented Dynamic Learning Environment), реализующая взаимодействие между преподавателем и обучающимися в процессе обучения. Система предусматривает обучение по курсам, которые могут содержать контент, организованный при помощи различных настраиваемых элементов: страница, книга, web-ресурс, файл и т.д. Так же в системе предусматривается ряд элементов для организации контроля: задания, игры, тесты, опросы, форумы и т.д. Система MOODLE имеет открытый код, и лицензия программного обеспечения предусматривает возможность его изменения. Использование системы Moodle в учебном

процессе было рассмотрено рядом исследователей: А. В. Белозубов, Д. Г. Николаев, А. П. Толстобров, И. А. Коржик [10,20, 47].

Таблица 1.2

Типы обучающих систем

Уровень усвоения знаний	Уровни мотивации учения	Типы компьютерных обучающих систем	Управление учебной деятельностью
Уровень идентификации ранее изученного	Первый-четвертый	Тесты первого уровня	Компьютер
		Электронные лекции	Компьютер, преподаватель
Уровень воспроизведения знаний	Первый-четвертый	Контролирующие системы, тесты второго уровня	Компьютер
		Электронные учебники	Компьютер, преподаватель
Уровень использования знаний в стандартных ситуациях	Второй-четвертый	Контроль третьего уровня	Преподаватель
		Электронные учебники, программы-тренажеры, компьютерные практикумы, моделирующие системы	Компьютер
Творческий уровень (использование знаний в нетипичных ситуациях)	Третий-четвертый	Творческие задания	Преподаватель
		Специальные системы по управлению решением учебных творческих задач, экспертные системы, САПР	Компьютер

Так, в работе А. В. Белозубова и Д. Г. Николаева хорошо описаны основы работы с системой электронного обучения Moodle, в работе А. П. Толстоброва и И. А. Коржик рассматриваются вопросы обеспечения качества тестовых заданий при использовании электронных систем управления обучения.

Следующий рассматриваемый продукт – система электронного обучения нового поколения eLearning систем – eFront. Данная платформа имеет функционал для управления образовательными траекториями, а также включает в себя базовые функции всех систем электронного обучения по

созданию и изменению учебных материалов. Возможности eFront позволяют решать задачи организации как учебного процесса в учебных заведениях, так и задачи повышения квалификации, аттестации и отбора сотрудников в организациях различного масштаба [34]. Система распространяется со свободной лицензией, реализующая основные функции LMS/LCMS. Некоторые из многочисленных достоинств системы:

- имеет сертифицированную поддержку формата SCORM, что предполагает использование системы, как в бизнесе, так и в учебных заведениях [23];
- естественный и ясный интерфейс;
- применение технологий Ajax, Unicode, LDAP, мультиязычность;
- нацеленность на привлечение учащихся и сохранение их интереса;
- концепция управления навыками;
- настройка на структуру организации и распределение функций и полномочий в соответствии с ней;
- широкий набор компонент, помогающих создавать элементы учебного курса, следить за успеваемостью и использованием системы, создавать сертификаты [21].

Система обучения LAMS основана на языке Java и также имеет свободную лицензию распространения GNU General Public License (GPL). Система также поддерживает ряд языков, однако в России она не распространена, а потому поддержка русскоязычного интерфейса в данной системе электронного обучения также отсутствует. В основу системы положены результаты работы Открытого университета Нидерландов (Open University of the Netherlands – OUNL) по языку образовательного моделирования EML, при помощи которого описывается «метамодель» разработки учебного процесса [22].

На основе данной спецификации была создана «Система управления последовательностью учебных действий» - LAMS, которая предоставляет преподавателям визуальные средства для разработки структуры учебного процесса, позволяющие задавать последовательность видов учебной деятельности. Система представляет преподавателю интуитивно понятный интерфейс для создания учебных элементов, в которые могут входить различные индивидуальные задания, задания для групповой работы.

Sakai - программная оболочка с открытым исходным кодом на Java, которую разрабатывает и использует международное сообщество крупнейших вузов мира: Университет Индианы, Массачусетский технологический институт, Стэнфордский университет и Университет Мичигана, Северный (Арктический) федеральный университет имени М.В. Ломоносова, Санкт-Петербургские университеты и т.д.

Sakai включает в себя набор программных инструментов, предназначенных для помощи преподавателям и студентам в поддержке учебного процесса разных форм обучения. СЭО может служить средой для взаимодействия, например, исследовательских групп. Sakai предоставляет дополнительные возможности для интенсификации и повышения эффективности образовательного процесса. Для совместной работы в системе есть инструментарий, обеспечивающий коммуникацию и групповую деятельность обучающихся. Пользователи могут выбирать набор инструментов на сайте курса, изменяя функциональность для своих целей. В рамках Sakai могут быть созданы следующие сайты:

- персональный сайт пользователя создается автоматически при регистрации пользователя и служит хранилищем личных ресурсов и рабочим пространством, а также собирает в одном месте всю необходимую пользователю информацию;

- сайт учебного курса, где студенты имеют возможность ознакомиться с программой и календарем курса, получить доступ к материалам занятий и контрольным учебным элементам, опраивать свои

работы преподавателю, воспользоваться функцией интерактивного онлайн-занятия с использованием аудио- и видеосвязи, виртуальной классной доски; т.е. на данном сайте сосредоточен весь процесс интерактивного взаимодействия преподавателя со студентами;

- сайт-проект для внутреннего взаимодействия, где руководитель или ответственный сотрудник может помещать объявления и осуществлять рассылку, предоставлять доступ к необходимым электронным ресурсам; здесь возможно выполнение функции форума студентов, преподавателей, сотрудников подразделения или группы исследователей;

- сайт-портфолио может представлять информацию персонального или демонстративного характера, отражающую выполнение пользователем системы заранее определенных требований; данный инструмент позволяет организовать эффективный контроль качества как отдельных компонентов учебного процесса, так и в целом его институциональных характеристик[37].

ATutor – PHP-приложение, работающее на любом сервере с операционной системой семейства Unix. Богатая функциональность ATutor позволяет преподавателям и студентам назначать и проходить курсы (в том числе и в формате SCORM), создавать рабочие группы, помогать друг другу в освоении новых тем, посылать сообщения, создавать блоги, обмениваться учебными материалами и делать еще массу других вещей [31].

ATutor создана канадскими специалистами, но имеет русскоязычный интерфейс. ATutor является системой электронного обучения с открытым исходным кодом и разработана с возможностью к адаптации. Функционал и интерфейс системы легко и быстро настраиваются. Так, преподаватели могут быстро собрать, упаковать, и перенести Web-содержание, а также открыть их для доступа в Интернете. Обучающиеся могут учиться в настраиваемой адаптивной среде обучения. Технология ATutor с открытым исходным кодом используется как малыми, так и крупными организациями, представляющими свои учебные материалы в Интернете, так и для доставки полностью независимых онлайн-курсов [31].

ClarolineLMS - платформа электронного обучения и электронной деятельности, обеспечивающая создание и использование эффективных онлайн-курсов, управление процессом обучения и совместными действиями на основе веб-технологий. ClarolineLMS мультиязычна и имеет 35 национальных языков интерфейса [32].

Как и предыдущие рассмотренные системы электронного обучения, ClarolineLMS имеет лицензию с открытым кодом. Она применяется во многих организациях различных стран и позволяет осуществлять работу в режиме онлайн. Каждый курс содержит инструменты, позволяющие преподавателю: описать назначение курса; опубликовать документы; администрировать форумы; разрабатывать групповые или индивидуальные траектории обучения; объединять студентов в группы; подготавливать онлайн упражнения; управлять повесткой дня с задачами и сроками выполнения; публиковать анонсы; вывешивать онлайн информацию о текущих заданиях; просматривать статистику активности пользователей.

ClarolineLMS используется школами, университетами, тренинговыми центрами, ассоциациями и компаниями. Платформа настраиваемая и предлагает гибкую среду для разработки под конкретную образовательную организацию.

Dokeos - платформа для разработки сайтов электронного дистанционного обучения, основанная на оболочке Claroline, которая представляет собой клон свободно распространяемого программного продукта для изменения приложения-оригинала в том или ином направлении. Например, были разработаны такие модификации Claroline, как:

- изменение ориентации приложения в пользу организаций;
- организация набора дополнительных сервисов для платформы [33].

Dokeos-сообщество предлагает набор различных сервисов к платформе: хостинг, интегрирование контента, разработка дополнительных модулей, техническая поддержка и т.д.

Dokeos так же имеет свободно распространяемую лицензию, поскольку лицензия Claroline (GNU/GPL) предполагает, что ветки подпадают под ту же лицензию. Оба приложения сейчас относительно похожи друг на друга.

OLAT – система управления обучением с открытым исходным кодом, содержащая гибкую систему онлайн курсов. Изначально система создана для государственных учреждений: университетов, академий и колледжей, однако, она может подойти и для других предприятий. OLAT расшифровывается, как «Online Learning And Training». Система предлагает максимально гибкий интуитивно-понятный подход подготовки изучения её содержания и возможностей. OLAT поддерживает возможность построения сценариев обучения. Система также включает в себя различные независимые возможности: общее руководство учебным ресурсом, включающее индексацию, а также мощный редактор инструментов для создания курсов, тестов и анкет.

Архитектура OLAT предлагает обучающую среду в соответствии с установленными методами обучения, а также предоставляет множество возможностей для подготовки и структурирования содержания обучения, чтобы проверить знания студентов и поддерживать их во время обучения [36].

Система предполагает поддержку даже социальных навыков при помощи синхронного или асинхронного взаимодействия.

ILIAS – бесплатная и кросс-платформенная система электронного обучения, которая популярна в высших учебных заведениях, т.к. разработана именно для них. Материалы системы записываются в формате XML и соответствуют стандарту SCORM [35].

Программа поддерживает такие возможности, как:

- обеспечение русификации интерфейса;
- подсказки преподавателям при создании учебных курсов;

– организация личного рабочего пространство каждого пользователя, со статусами студентов «онлайн — офлайн», справочной системой, уведомлениями о поступающей почте и так далее;

– предоставление чатов, форумов, личных сообщений в качестве средств коммуникации;

– контроль времени выполнения учебных заданий и проведение тестов.

По результатам анализа рассмотренных систем, составлена таблица 1.3, отражающая их сравнение по различным характеристикам [11].

Таблица 1.3

Сравнительная таблица СЭО

	MOOD LE	E- front	LAM S	Sakai	ATuto r	Clarolin e	Dokeo s	OLA T	ILI AS
1	2	3	4	5	6	7	8	9	10
SCOR M	+	+	-	+	+	+	+	+	+
IMS	+	+	-	+	+	+	+	+	-
Язык и прило жения	PHP	PHP	Java	Java	PHP	PHP	PHP	Java	PH P
СУБ Д	MySQL	MySQ L	MyS QL	MySQ L, Oracle , hsqldb	MySQ L	MySQL	MySQ L	MyS QL Postg reSQ L	My SQ L
Лице нзии	GNU/G PL	GNU/ GPL	GNU/ GPL	GNU/ GPL	GNU/ GPL	GNU/G PL	GNU/ GPL	GNU/ GPL	GN U/G PL
Русск ий Язык	+	+	-	+	+	+	+	+	+

Таблица 1.3 продолжение

1	2	3	4	5	6	7	8	9	10
Другие языки	>54	>30	20	28	>50	36	38	34	43
Система проверки знаний	тесты, задания, семинары, активность на форумах	тесты, задания, семинары, активность на форумах	тесты	тесты, задания, активность на форумах	тесты	тесты, упражнения	тесты	тесты, задания	тесты
Демонстрационный сервер	+	+	+	-	+	+	+	+	-

Таким образом, в результате исследования рассмотрены СЭО, используемые при в университетах мира при организации разных форм обучения. По результатам анализа систем составлена сравнительная таблица их возможностей и характеристик. Исходя из полученных результатов, можно сделать вывод о весьма широком выборе программного обеспечения для автоматизации обучения. Важно отметить, что рынок программных продуктов в данной предметной области предлагает большое количество платформ с бесплатной лицензией и открытым программным кодом, что позволяет доработать систему в соответствии с предпочтениями образовательной организации. Для реализации разработанных в диссертации алгоритмов будет использована образовательная платформа E-front, т.к. она подходит по функционалу и имеет открытый для разработки код и базу данных.

2 ИССЛЕДОВАНИЕ СИСТЕМЫ ОБУЧЕНИЯ НИУ «БЕЛГУ»

2.1 Исследование особенностей образовательного процесса НИУ «БелГУ»

На сегодняшний день, НИУ «БелГУ» предоставляет образовательные услуги для 23 тыс. студентов из всех регионов России, более 2,5 тыс. иностранных студентов из 80 стран мира. Университет позволяет студентам осуществить выбор направления обучения деятельности из 190 лицензированных и аккредитованных специальностей среднего и высшего профессионального образования. НИУ «БелГУ» предлагает возможность обучения сразу на нескольких учебных программах одновременно. С 2016/2017 года в университете запущен проект обучения студентов заочного обучения в режиме онлайн-трансляции, что так же делает обучение в ВУЗе более привлекательным для жителей отдаленных районов. До этого дистанционное обучение предполагало лишь самостоятельную удаленную работу студентов с системой электронного обучения[38].

Структурными подразделениями, координирующими учебный процесс, после Ректора являются отделы:

- Управление образовательной политики;
- Управление заочного обучения и электронных образовательных технологий.

Данные подразделения следят за соответствием стандартам всех образовательных программ выпускающих кафедр и готовность их к использованию в системе дистанционного обучения. Организационная структура (описывающая координирование учебного процесса университета), отображена на рис. 2.1.

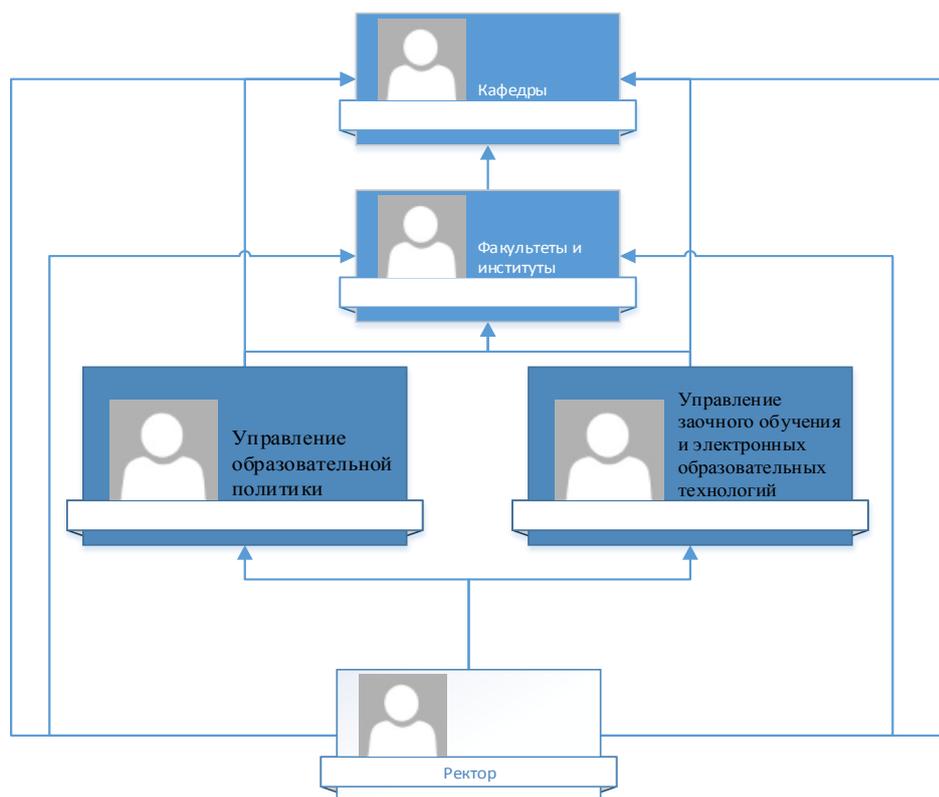


Рис. 2.1. Организационная схема учебного процесса НИУ БелГУ

Деятельность управления образовательной политики определяется Законом Российской Федерации от 29.12.2012 № 273 «Об образовании в Российской Федерации», нормативными правовыми актами Российской Федерации, в частности приказом Министерства образования и науки Российской Федерации от 13.06.2013 № 455 «Об утверждении порядка и оснований предоставления академического отпуска обучающимся», постановлением Правительства РФ от 15.08.2013 № 706 «Об утверждении Правил оказания платных образовательных услуг», типовым положением об образовательном учреждении высшего профессионального образования (высшем учебном заведении) Российской Федерации (утв. постановлением Правительства Российской Федерации от 14.02.2008 № 71), Уставом НИУ «БелГУ», положением об Управлении образовательной политики [38].

Главной целью деятельности Управления является планирование, организация и контроль осуществления учебной работы, процедур

лицензирования и аккредитации, а также координация и организация учебного процесса.

Управление включает в себя отделы:

- 1) Отдел планирования и организации учебного процесса
- 2) Отдел развития образовательного комплекса
- 3) Отдел методической работы и качества образования
- 4) Отдел проектирования информационных ресурсов и систем
- 5) Отдел разработки программного обеспечения

Управление заочного обучения и электронных образовательных технологий включает в себя следующие основные направления деятельности:

– разработка нормативно-правовых и организационно-методических документов, регламентирующих использование технологий электронного обучения НИУ «БелГУ»;

– координация деятельности подразделений НИУ «БелГУ» по подготовке к осуществлению образовательного процесса с использованием технологий электронного обучения;

– организация подготовки профессорско-преподавательского состава к осуществлению образовательного процесса с использованием технологий электронного обучения;

– формирование и развитие единой информационно-образовательной среды электронного обучения;

– организация и контроль учебного процесса с использованием технологий электронного обучения;

– совершенствование и развитие системы электронного обучения НИУ «БелГУ»;

– комплексное развитие системы дополнительного профессионального образования НИУ «БелГУ», эффективное её использование на рынке образовательных услуг в интересах НИУ «БелГУ» и его структурных подразделений.

Структура управления включает в себя:

- Отдел электронных образовательных ресурсов
- Отдел заочного и электронного обучения
- Отдел дополнительного профессионального образования

Кафедры институтов и факультетов под руководством названных управлений, осуществляют разработку электронных учебно-методических комплектов, в дальнейшей публикующихся в системе электронного обучения НИУ «БелГУ». Важно отметить, что Университет уже несколько лет обучает студентов по балльно-рейтинговой системе (БРС). Система предполагает разработку преподавателями учебных курсов технологических карт БРС (для студентов очной формы обучения) или графика изучения дисциплины (для студентов заочной формы обучения). Карта БРС представляет таблицу, содержащую в себе все виды деятельности, за которые студент может получить баллы в течение семестра. График изучения дисциплины описывает последовательность и сроки обучения, а также балльную оценку работы студентов заочной формы в системе дистанционного обучения.

Некоторые кафедры применяют в работе со студентами дополнительные системы электронного обучения, не интегрированные с общеуниверситетской системой «Пегас». Но именно при помощи системы электронного обучения университета реализуется контрольное тестирование для всех форм обучения, как в межсессионный период, так и в период экзаменационных сессий, зачетных недель. Студенты университета автоматически подключаются к курсам обучаясь как на очной (не в обязательном порядке), так и на заочной основе (в обязательном порядке) после распределения заведующими кафедрой учебной нагрузки. Стоит отметить, что студент может попасть на нужный курс по дисциплине просто открыв расписание.

Стоит отметить, что система обучения студентов заочной формы, даже при использовании системы электронного обучения, имеет свои недостатки: отсутствует контроль за выполнением заданий и тестирований в СЭО, нельзя

проследить за усвояемостью материала, предоставить студенту материал в наиболее усвояемом для него виде. Данные проблемы можно решить путем совершенствования существующих алгоритмов системы электронного обучения НИУ «БелГУ», что позволит при помощи учета индивидуально-ориентированных технологий обучения повысить качество образовательных услуг, предоставляемых не только на очной, но и на заочной форме обучения.

2.2 Исследование системы электронного обучения НИУ «БелГУ»

Дистанционное обучение в НИУ «БелГУ» реализовано при помощи электронной системы обучения «Пегас», разработанной на платформе модульной динамической учебной среды «Moodle». СЭО выполняет функцию автоматизации процессов управления обучением посредством предоставления доступа к электронному образовательному контенту из любой точки доступа к глобальной сети. Следует отметить, что система отлично реализует образовательные гипермедиа технологии. В середине 2015-2016 учебного года осуществлен переход системы электронного обучения на платформу Moodle 3.0 [10].

СЭО «Пегас» поддерживает разграничение пользователей по различным ролям. Создание ролей и присвоение им прав осуществляет администратор системы. Чаще всего применяются такие типы ролей, как: администратор, студент, преподаватель-редактор. Роль «студент» предполагает следующий набор функций: электронные услуги (графики консультаций и экзаменов, заказ экзаменационных листов и справок об обучении, формирование квитанций за проживанием в общежитии и по оплате обучения), электронный журнал, изучаемые учебные курсы, анкетирование пользователей.

Роль преподавателя-редактора включает значительно больше возможностей. Преподаватели имеют полный пакет прав к курсам, разработчиками которых они являются, а так же к курсам, которые они преподают.

Курсы системы электронного обучения могут содержать разные типы элементов: HotPot, анкета, база данных, видеоконференция BigBlueButton, вики, внешний инструмент, глоссарий, задание, игры, книга с вопросами, лекция, обратная связь, опрос, пакет SCORM, рабочая тетрадь, тест, форум, чат. Как же в курсе есть возможность использования следующих типов ресурсов: гиперссылка, книга, папка, пояснение, страница, файл.

При выполнении заданий, представленных в виде разных элементов, студенты отправляют ответы на проверку преподавателю (задания, форум, рабочая тетрадь и т.д.), либо система оценивает выполнение задания автоматически (тест, игры). Взаимодействие обучающегося с преподавателем напрямую по возникающим вопросам может осуществлять как во время проведения занятия, при помощи видеоконференции, так и после занятий при помощи обмена электронными сообщениями.

При подробном рассмотрении элементов системы «Пегас», можно отметить, что каждый курс универсален для всех обучающихся. Система не предусматривает подстройку контента под модель обучающегося. Настройки тестирования весьма разнообразны, однако они применяются либо для всех учащихся курса либо для конкретных групп. Разбиение учебной группы на более мелкие подгруппы невозможно. Кроме того, параметры ограничений по тесту преподаватель задает вручную. Теоретический материал и лабораторные задания универсальны для всех обучающихся, четко структурированы, доступны студентам сразу и в полном объеме.

Тестовая база предполагает разделение тестов по типам: множественный выбор, верно/неверно, краткий ответ, числовой ответ, вычисляемый, на соответствие, случайный вопрос на соответствие, вложенные ответы, выбор пропущенных слов, множественный вычисляемый,

перетаскивание в текст, перетаскивание маркеров, перетащить на изображение, простой вычисляемый. При выполнении теста, студент может просматривать понятия, содержащиеся в глоссарии, при наведении курсора на слово, что отчасти является недостатком системы тестирования, т.к. такие подсказки снижают надежность итоговых показателей оценки знаний студентов.

Оценивание каждого из элементов курса преподавателем отражается в разделе настроек курса «Оценки». Если в курсе есть задания, не предусматривающие ответа обучающегося (например - посещение лекции), преподаватель проставляет баллы по данным заданиям сразу в журнале оценок курса. Стоит отметить, что для каждого курса преподаватель формирует БРС в журнале оценок при помощи создания категории «БРС» и наполнения ее заданиями, необходимыми для получения оценки обучающимися по окончанию курса. Обучающиеся по курсу студенты могут в любой момент просмотреть текущее состояние их рейтинга в том же разделе: Настройки курса – Оценки.

Внутри курса существует удобная страница просмотра последних изменений, где за выбранный промежуток времени преподаватель может увидеть новых зачисленных студентов, новые сообщения на форумах, законченные попытки прохождения тестов и другую информацию о ходе обучения студентов на курсе.

Исходя из описания системы электронного обучения, можно прийти к выводу о том, что на данный момент, использование обучающимися СЭО «Пегас» в процессе обучения является универсальным. Для организации индивидуально-ориентированного обучения в системе электронного обучения необходима реализация таких функций, как: управление учебной деятельностью учащихся; контроль выполнения заданий; формирование индивидуальных наборов учебно-тренировочных задач; адаптация структуры курса под пользователя.

Однако, совершенствование существующих алгоритмов в системе «Пегас», достаточно затруднительно, т.к. для реализации части перечисленных функций необходимо создавать новые модули. Поэтому для реализации алгоритмов предлагается система электронного обучения e-front, которая уже включает в себя необходимые для индивидуально-ориентированного обучения модули, т.е. достаточно будет переписать лишь часть кода уже готового модуля. Система будет использоваться для обучения студентов инженерных специальностей.

Так, в систему e-front возможно внедрение адаптивных технологий типа: гипертекстовый компонент; адаптивное планирование; поддержка адаптивной навигации; интеллектуальный анализ решений; адаптивное представление.

Таблица 2.1

Недостатки системы и возможности их совершенствования на платформе efront

	Область функционирования	Недостатки СЭО «Пегас»	Возможности усовершенствования
1	2	3	4
1	Контрольное тестирование	1) Настройки курсу универсальны для всех обучающихся, подключенных к нему. 2) Время доступа к тестированию настраивается для только для групп, но не для каждого обучающегося. 3) Отсутствует возможность автоматической генерации групп обучающихся по индивидуальным особенностям; 3) Отсутствует возможность изменения заданий в ходе тестирования.	1. Разработка алгоритмического обеспечения, позволяющего сократить время контрольных испытаний за счет технологии адаптивного тестирования. 2. Разработка алгоритма, задающего индивидуальное время прохождения тестирования.
2	Формирование контента	1) Наполнение курса универсально для всех обучающихся	1. Разработка группой экспертов контента для различных форм восприятия 2. Разработка алгоритма адаптивного представления 3. Разработка алгоритма адаптивного планирования

Таблица 2.1 продолжение

1	2	3	4
3	Навигация по курсу	1) Стандартная навигация для всех обучающихся, подключённых к курсу	1. Настройка блокировки ссылок, недопустимых для обучения на текущий момент. Пока не будет доступен предыдущий материал
4	Контроль выполнения заданий	1) Отсутствует анализ результатов тестирования на предмет надежности теста	1. Сопоставление результатов тестирования с индивидуальной моделью обучаемого
6.	Контрольное тестирование	1) Настройки курсу универсальны для всех обучающихся, подключенных к нему. 2) Время доступа к тестированию настраивается для только для групп, но не для каждого обучающегося. 3) Отсутствует возможность автоматической генерации групп обучающихся по индивидуальным особенностям; 3) Отсутствует возможность изменения заданий в ходе тестирования.	1. Разработка алгоритмического обеспечения, позволяющего сократить время контрольных испытаний за счет технологии адаптивного тестирования. 2. Разработка алгоритма, задающего индивидуальное время прохождения тестирования.

Таким образом, в результате проведения анализа системы электронного обучения НИУ «БелГУ» выделены основные рекомендации по ее совершенствованию. Составлена сравнительная таблица 2.1, отражающая текущие функции системы и возможности ее усовершенствования по различным пунктам.

3 РАЗРАБОТКА ФОРМАЛЬНО-АЛГОРИТМИЧЕСКОГО ОБЕСПЕЧЕНИЯ ПОДДЕРЖКИ ПРОЦЕССА ИНДИВИДУАЛЬНОГО ОБУЧЕНИЯ

3.1 Проектирование процесса индивидуально-ориентированного обучения

При разработке или совершенствовании системы индивидуально-ориентированного электронного обучения требуется построение моделей, отражающих основополагающие свойства и процессы. Так, укрупненная модель системы позволяет отобразить ее основные объекты, процессы, операции. Декомпозиция, в свою очередь, дает представление об аспектах функционирования системы на уровне элементов, соответствующих минимальным объектам системы.

Рассмотрим контекстную диаграмму IDEF0 процесса индивидуально-ориентированного обучения, представленную на рис. 3.1. Согласно схеме на процесс обучения помимо основных сведений о курсе и форме обучения влияют данные об индивидуальных показателях обучающегося.

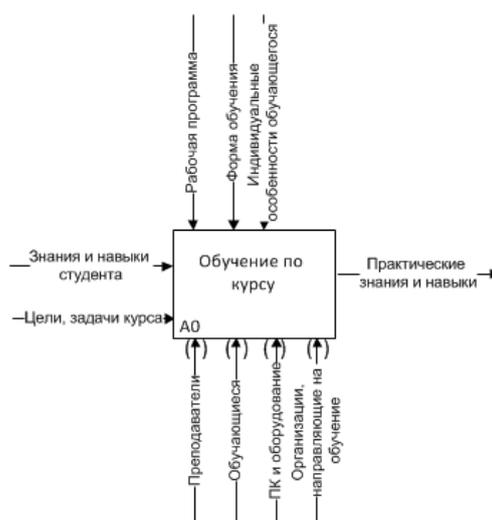


Рис. 3.1. Контекстная диаграмма процесса индивидуально-ориентированного обучения

На рис. 3.2 представлена декомпозиция процесса обучения студента по курсу в нотации IDEF0. На диаграмме показано, что индивидуальные данные о студенте и его остаточных знаниях используются в процессах «Электронное обучение» и «Контрольное тестирование».

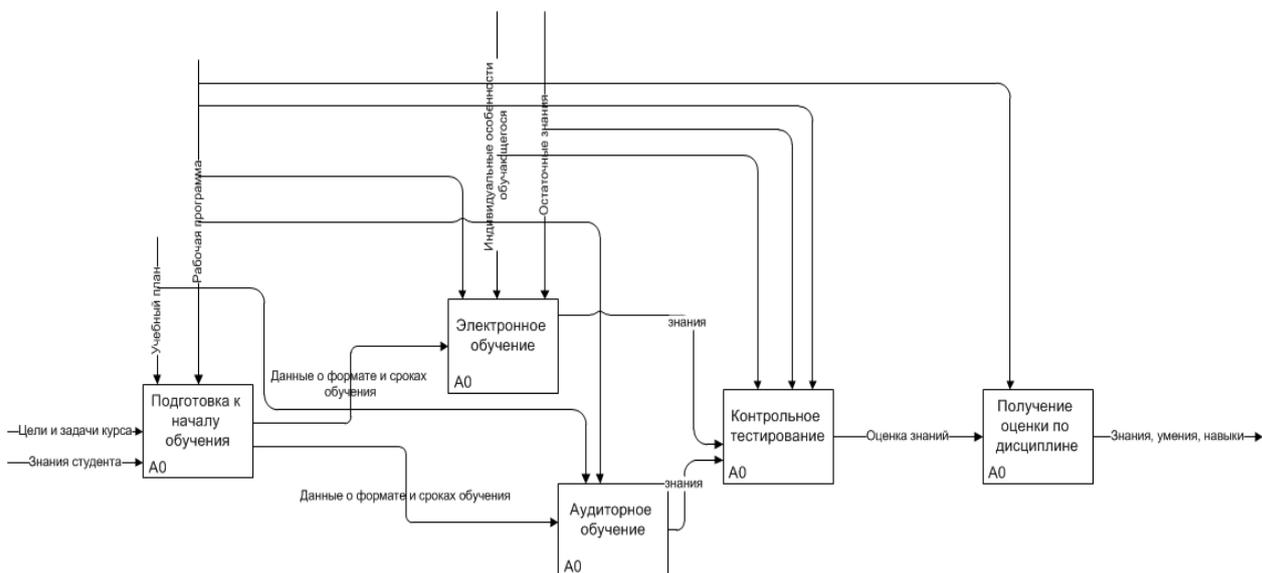


Рис. 3.2. Процесс обучения студента по дисциплине с использованием индивидуальных показателей студента

Для более детального рассмотрения представленных на рис. 3.2. описания точек соприкосновения нескольких процессов индивидуально-ориентированного обучения, при работе в системе пользователей с различными учетными записями выбран аппарат сетей Петри, позволяющий отразить параллельные процессы, происходящие в системе.

Разрабатываемая система электронного обучения основывается на независимом от преподавателя управлении процессом индивидуального усвоения знаний обучающихся с вмешательством в систему управления лишь при условиях, когда система не способна самостоятельно принять решение о дальнейшем формировании индивидуальной образовательной траектории обучающегося. На рис. 3.3 отображена укрупненная схема работы системы, отражающая основные состояния и переходы, необходимые для функционирования индивидуально-ориентированной системы обучения и определения тактики обучения.

Так, работа сети начинается из состояния 1 – программа в незапущенном состоянии. Для перехода к состоянию ввода данных авторизации необходимо срабатывание перехода запуска программы: I. На схеме показано, что после идентификации пользователя, система включает в себя параллельные процессы: при срабатывании перехода II – работа преподавателя, при срабатывании перехода III – работа обучающегося (см. рис. 3.3.), где работа первого включает состояния 3, 6, 7, а второго – состояния 4, 5, 8, 10. Важно отметить участок сети с переходами VIII и XI – для их срабатывания необходима обязательная активность обоих входящих в него состояний. То есть, для перехода к полному изучению контента обучающимся, преподаватель должен окончить редактирование курса.

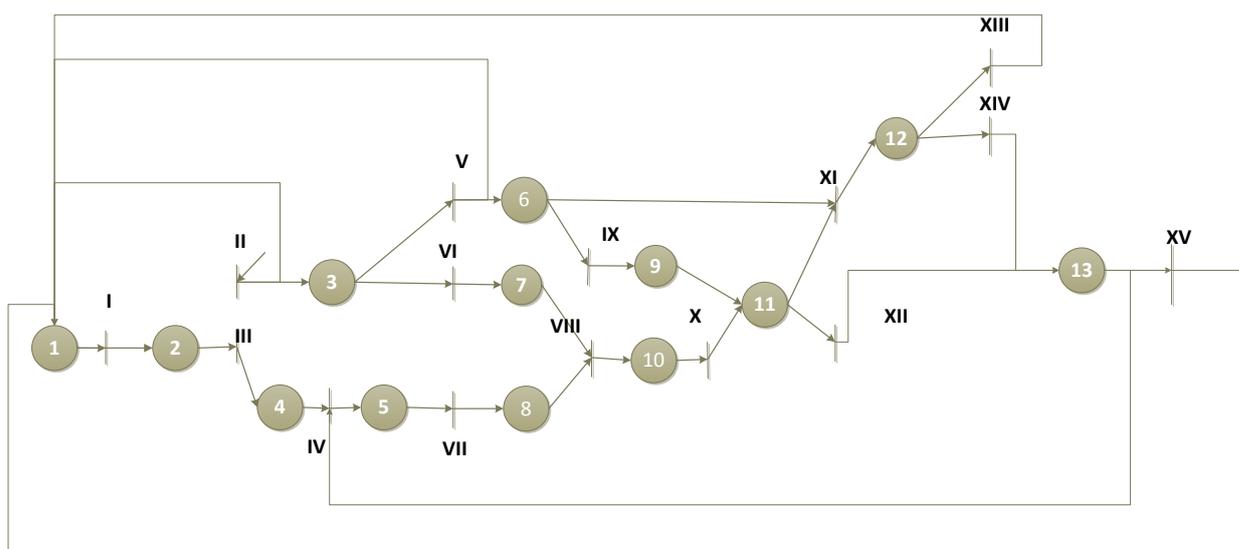


Рис. 3.3. Схема процесса индивидуально-ориентированного обучения

Описание позиций и переходов, схемы рис. 3.3 представлены в таблицах 3.1 и 3.2

Таблица 3.1

Позиции сети Петри схемы процесса индивидуально-ориентированного обучения

№ поз.	Описание позиции
1	2
1	Приложение не запущенно
2	Ожидание авторизации
3	Вход преподавателя-редактора выполнен
4	Вход обучающегося выполнен
5	Отображение главного меню со списком доступных курсов
6	Наблюдение эксперта за выполнением тестовых заданий
7	Изменения, внесенные преподавателем вступили в силу
8	Изучен доступный контент курса
9	Преподаватель завершил редактирование теста
10	Контент освоен обучающимся
11	Обучаемый прошел тестирование
12	Решение о дальнейшем обучении учащегося
13	Обучение по теме закончено

Таблица 3.2

Переходы сети Петри схемы процесса индивидуально-ориентированного обучения

Обозначение	Описание действия перехода
I.	Запуск программы
II.	Авторизация преподавателя
III.	Авторизация обучающегося
IV.	Загрузка личных данных обучающегося
V.	Корректировка контрольных заданий
VI.	Коррекция учебного курса, с его временной блокировкой
VII.	Формирование контента и проверка доступа
VIII.	Изучение представленного к изучению контента
IX.	Изменение и анализ тестов экспертом
X.	Завершение сеанса пользователя
XI.	Изменение преподавателем решения о дальнейшей траектории обучения обучающегося
XII.	Выполнение теста обучающимся
XIII.	Отправка решения об отчислении обучающегося
XIV.	Переход к формированию нового контента
XV.	Выход из программы

Протекающие внутри системы процессы (см. рис. 3.3) декомпозированы с целью более детального рассмотрения. В первой схеме (см. рис. 3.4) представлен процесс формирования индивидуального учебного контента. Так, после авторизации пользователя возможно осуществление двух параллельных процессов: работы обучающегося и преподавателя. При подключении к курсу обучающегося, система переходит к меню выбора курса, после чего пользователь последовательно выбирает раздел и тему. При выборе элементов контента возможен выход из системы или же переход в предыдущее меню по желанию обучаемого, в том числе в случае, когда отсутствует доступа к необходимому ресурсу. Одновременно с обучающимся в системе может работать преподаватель. От его действий зависит подключение обучаемого к курсу и обновление или переработка при необходимости существующего контента. В случае, редактирования контента преподавателем доступ к соответствующему материалу в обучающегося блокируется.

Контент курса различается в зависимости от глубины научного проникновения в сущность учебного элемента. Возможны четыре ступени абстракции при отображении контента:

Ступень 1 – феноменологическая теория (описание на естественном языке).

Ступень 2 – качественная теория (использование специфического языка науки с терминологией и символикой).

Ступень 3 – количественная теория (создание математической модели объекта науки и обеспечение наиболее полного знания закономерностей его поведения).

Ступень 4 – аксиоматическая теория (высшая ступень развития науки в целом, учитывающая общность законов функционирования объектов, принадлежащих разным наукам).

Каждый курс может быть изложен на любой ступени абстракции при условии, что область знания по курсу достигла ее в своем развитии. На каком

языке излагать информацию об учебном элементе, т.е. на каком языке надо научить учащихся говорить, думать и мыслить зависит от направления подготовки обучающегося.

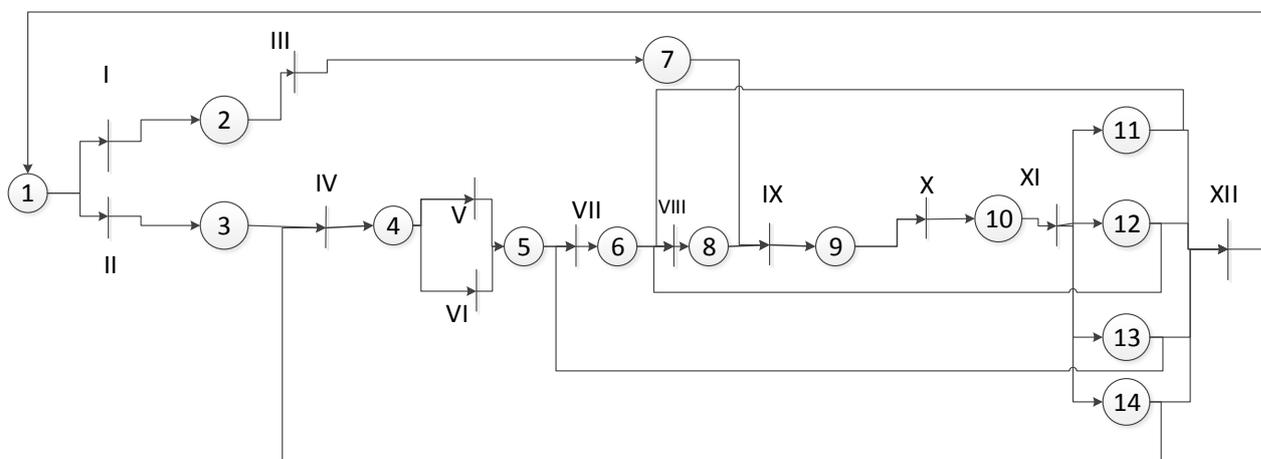


Рис.3.4. Процесс генерации контента для обучаемого

В таблицах 3.3 и 3.4 представлены расшифровки значений позиций и переходов, соответствующих рис. 3.4.

Таблица 3.3

Позиции сети Петри процесса генерации контента для обучаемого

№ поз.	Описание позиции
1.	Система неактивна
2.	Вход преподавателя выполнен
3.	Вход обучающегося выполнен
4.	Вход в меню курса
5.	Характеристики обучаемого загружены
6.	Меню темы
7.	Контент готов для использования
8.	Меню раздела
9.	Контент сформирован
10.	Изучен подраздел, отображение меню подразделов
11.	Ссылка в меню для повторения добавлена
12.	Изучена тема
13.	Изучен раздел
14.	Изучен курс

Переходы сети Петри процесса генерации контента для обучаемого

Обозн.	Описание действия перехода
I.	Ввод данных преподавателя
II.	Ввод данных обучающегося
III.	Редактирование курса
IV.	Подключение к курсу обучающегося
V.	Загрузка истории обучения
VI.	Выполнение теста на остаточные знания
VII.	Выбор темы
VIII.	Выбор раздела
IX.	Выбор подраздела
X.	Изучение подраздела
XI.	Контрольные мероприятия
XII.	Выход из системы обучающегося

В случае, когда уровень усвоения учебного материала соответствует модели обучаемого, осуществляется проверка: является ли текущая тема последней в разделе курса. Если да, то следует переход к прохождению тестового контроля по разделу изучаемого курса. Если изучаемый раздел последний в курсе - обучающийся переходит к прохождению итогового тестового контроля по изучаемому курсу. Если оба данных условия не выполняются, осуществляется формирование следующего индивидуального контента обучаемого. Если обучаемый многократно неуспешно проходит тестирование и количество повторов изучения больше порогового, то текущий раздел курса блокируется, в результате чего, обучающийся не может продвинуться в обучении, пока не будет проведена консультация с преподавателем-редактором, который принимает решение о последующем формировании контента для данного обучаемого либо окончании обучения.

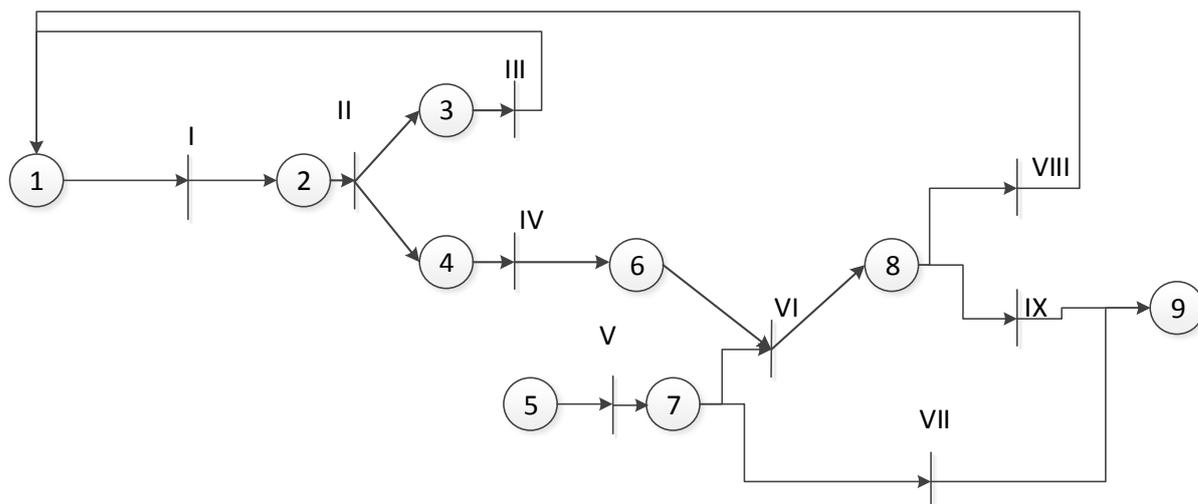


Рис. 3.5. Процесс тестирования обучаемого

В таблицах 3.5 и 3.6 представлены расшифровки значений позиций и переходов отображенной на рис. 3.5 сети Петри.

Таблица 3.5

Позиции сети процесса тестирования обучаемого

№ поз.	Описание позиции	№ поз.	Описание позиции
1	Сформирован контент	6	Ожидание консультации эксперта
2	Контент изучен	7	Эксперт готов к работе с обучаемым
3	Тест выполнен успешно	8	Решение по обучающемуся
4	Количество неправильных попыток превышено	9	Программа неактивна
5	Вход эксперта выполнен		

Таблица 3.6

Переходы сети Петри процесса тестирования обучаемого.

Обозначение	Описание действия перехода
I	Изучение контента
II	Прохождение теста
III	Переход к следующему блоку знания
IV	Блокировка курса
V	Проверка необходимости консультаций для обучающихся
VI	Консультация с обучающимися
VII	Выход из программы
VIII	Формирование нового контента
IX	Решение об окончании обучения студента

Таким образом, разработаны контекстная и декомпозиционная диаграммы процесса обучения по курсу в нотации IDEF0, выделяющие основные составляющие процесса и демонстрирующие применимость индивидуальных показателей обучающихся. Также разработана укрупненная и декомпозиционные схемы процесса при помощи инструмента - сети Петри, позволяющего продемонстрировать работу системы при наличии параллельных процессов в системе. Данные схемы удобны при реализации системы на различных формальных языках, т.к. данное представление предполагает достаточно глубокое рассмотрение любого процесса в системе. Так, проведенная декомпозиция алгоритма функционирования системы при изучении курса, сокращает время и трудоемкость разработки адаптивных алгоритмов системы обучения.

3.2 Разработка базы знаний для системы индивидуально-ориентированного тестирования обучающихся

Одним из наиболее перспективных направлений совершенствования систем электронного обучения является применение моделей, основанных на нечеткой логике, при разработке адаптивных алгоритмов построения индивидуальных образовательных траекторий. Такая тенденция объясняется тем, что процесс обучения является сложной слабоструктурированной системой. При формализации ряда процессов типа: четкой классификации обучаемых по уровню подготовленности, задания сложности тестовых заданий и т.д. возникают сложности с выбором точных критериев, определяющих управляющие воздействия [4,47]. К подобным процессам можно отнести определение уровня сложности для обучающихся в ходе проведения промежуточного или итогового контрольного тестирования. Написание данного модуля системы предусматривает предварительную разработку нечеткой базы знаний [7,9].

Как уже было описано в главе 3.1, в начале процесса обучения, каждый

пользователь проходит тест на определение начального уровня знаний, результаты которого заносятся в базу данных, как характеристики обучающегося. Полученные результаты предлагается использовать следующим образом: исходя из начального уровня подготовки, система будет предлагать обучающемуся при контрольном тестировании задания соответствующего уровня сложности. Следует отметить, что уровень сложности тестовых заданий может меняться в зависимости от текущих результатов выполнения теста. Таким образом, для определения уровня сложности тестовых заданий необходимо учитывать: начального уровня знаний и текущего результата тестирования. Согласно этим зависимостям, спроектирована система нечеткого логического вывода типа Мамдани при помощи пакета Fuzzy Logic Toolbox MatLab (см. рис. 3.6).



Рис. 3.6. Укрупненная схема системы нечеткого логического вывода Мамдани

Для каждой переменной на входе составлены графики, описывающие ее нечеткие значения, которые можно также графически сгенерировать. Результаты оценки знаний принято представляться в форме столбальной шкалы. Выполнение одного и того же количества тестовых заданий из множеств вопросов, не может считаться равнозначным, в как задания имеют весовые коэффициенты. Соотношение шкал задается следующее: (41-60) баллов - «3»; (61-80) баллов - «4»; (81-90) баллов - «5»; (90-100) баллов - «эксперт».

На рис. 3.7 отображен график нечетких значений переменной

«ТекущийТест». На оси ординат располагается шкала, отражающая количество баллов, полученных по результатам тестирования. На оси абсцисс отражается вероятность приобретения переменной одного из 5 заданных значений (неудовлетворительно, удовлетворительно, хорошо, отлично, эксперт). Вершины графиков соответствуют заданной системе оценивания. В соответствии с рекомендацией эксперта, опишем лингвистическую переменную «Текущий тест». Множество целых чисел из интервала $X: [0, 100]$; множество нечетких переменных, для каждого значения $T(x)$:

«неудовлетворительно»	$\mu(x) = \begin{cases} 1, & x \leq 15 \\ \text{нелинейная аппроксимация}, & 15 < x < 53 \\ 0, & x \geq 53 \end{cases}$
«удовлетворительно»	$e^{-\frac{(x-14)^2}{2 \cdot 42^2}}$
«хорошо»	$e^{-\frac{(x-16)^2}{2 \cdot 69^2}}$
«отлично»	$\mu(x) = \begin{cases} 0, & x \leq 71 \\ \text{нелинейная аппроксимация}, & 71 < x < 92 \\ 1, & x \geq 92 \end{cases}$

Множество синтаксических правил для образования имен новых значений G , включает в себя новую переменную «эксперт», которая образуется при помощи семантического правила CON(«отлично»).

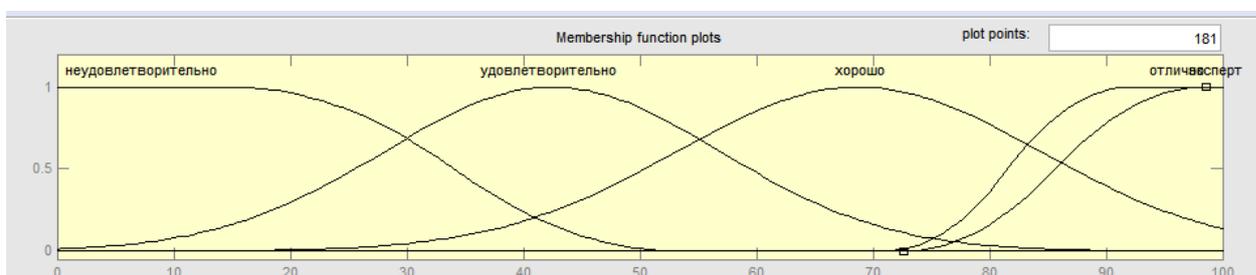


Рис. 3.7. График нечетких значений переменной «ТекущийТест»

График значений переменной «НачальныйУровеньПодготовки»

отображен на рис. 3.8. Здесь присутствуют значения, как: слабый, средний, сильный, эксперт, соответствующие начальному уровню подготовки обучающегося и определяющие, какого уровня успеваемости от него ожидать. При помощи учета значения данной переменной система значительно экономит время, отведенное на тестирование.



Рис. 3.8. График нечетких значений переменной «Начальный Уровень Подготовки»

Опишем лингвистическую переменную «Текущий Уровень Подготовки». Множество целых чисел из интервала $X: [0, 1]$; множество нечетких переменных, для каждого значения $T(x)$:

«слабый» $\mu(x) = \begin{cases} 1, & x \leq 0,13 \\ \text{нелинейная аппроксимация}, & 0,13 < x < 0,5 \\ 0, & x \geq 0,5 \end{cases}$

«средний» $\mu(x) = \frac{-(x-0,17)^2}{e^{2*0,55^2}}$

«сильный» $\mu(x) = \begin{cases} 0, & x \leq 0,57 \\ \text{нелинейная аппроксимация}, & 0,57 < x < 0,89 \\ 1, & x \geq 0,89 \end{cases}$

Множество синтаксических правил для образования имен новых значений G , включает в себя новую переменную «эксперт», которая образуется при помощи семантического правила $CON(\text{«сильный»})$;

На рис. 3.9 отображен график нечетких значений переменной выхода - «Уровень Сложности». Значения данной лингвистической переменной описывают уровни сложности тестовых заданий и напрямую зависят от значений входных переменных «Текущий Тест» и

«Начальный Уровень Подготовки».

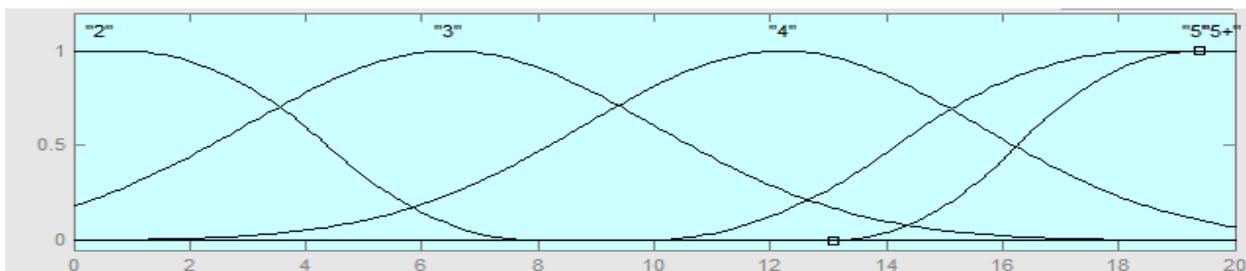


Рис. 3.9. График нечетких значений переменной «Уровень сложности»

Опишем лингвистическую переменную «Уровень Сложности». Множество целых чисел из интервала $X: [0, 20]$; множество нечетких переменных, для каждого значения $T(x)$:

$$\langle\langle 2 \rangle\rangle \quad \mu(x) = \begin{cases} 1, & x \leq 0,8 \\ \text{нелинейная аппроксимация}, & 0,8 < x < 8 \\ 0, & x \geq 8 \end{cases}$$

$$\langle\langle 3 \rangle\rangle \quad \frac{-(x - 3,5)^2}{e^{2 * 3,5^2}}$$

$$\langle\langle 4 \rangle\rangle \quad \frac{-(x - 3,4)^2}{e^{2 * 12,1^2}}$$

$$\langle\langle 5 \rangle\rangle \quad \mu(x) = \begin{cases} 0, & x \leq 9,8 \\ \text{нелинейная аппроксимация}, & 9,8 < x < 18,5 \\ 1, & x \geq 18,5 \end{cases}$$

Множество синтаксических правил для образования имен новых значений G , включает в себя новую переменную «5+», которая образуется при помощи семантического правила $CON(\langle\langle 5 \rangle\rangle)$. База знаний, формируется на основе рассмотренных переменных. Так, при проведении теста система будет предлагать обучающемуся попробовать себя на более высокую оценку, в случае, если при решении теста коэффициентом правильности выполнения будет выше 0,7. Для понижения уровня сложности выдаваемых заданий применяется коэффициент правильности ниже 0,4. Правила, на основе которых осуществляются переходы от одного уровня сложности тестовых заданий к другому, приводятся ниже:

- 1) If (ТестПоТеме is неудовлетворительно) and

- (ТекущийУровеньПодготовки is слабый) then (УровеньСложности is «3»)
- 2) If (ТестПоТеме is удовлетворительно) and (ТекущийУровеньПодготовки is слабый) then (УровеньСложности is «3»)
 - 3) If (ТестПоТеме is хорошо) and (ТекущийУровеньПодготовки is слабый) then (УровеньСложности is «4»)
 - 4) If (ТестПоТеме is неудовлетворительно) and (ТекущийУровеньПодготовки is средний) then (УровеньСложности is «3»)
 - 5) If (ТестПоТеме is удовлетворительно) and (ТекущийУровеньПодготовки is средний) then (УровеньСложности is «3»)
 - 6) If (ТестПоТеме is хорошо) and (ТекущийУровеньПодготовки is средний) then (УровеньСложности is «4»)
 - 7) If (ТестПоТеме is отлично) and (ТекущийУровеньПодготовки is средний) then (УровеньСложности is «5»)
 - 8) If (ТестПоТеме is неудовлетворительно) and (ТекущийУровеньПодготовки is сильный) then (УровеньСложности is «4»)
 - 9) If (ТестПоТеме is удовлетворительно) and (ТекущийУровеньПодготовки is сильный) then (УровеньСложности is «4»)
 - 10) If (ТестПоТеме is хорошо) and (ТекущийУровеньПодготовки is сильный) then (УровеньСложности is «5»)
 - 11) If (ТестПоТеме is отлично) and (ТекущийУровеньПодготовки is сильный) then (УровеньСложности is «5+»)
 - 12) If (ТестПоТеме is эксперт) and (ТекущийУровеньПодготовки is эксперт) then (УровеньСложности is «5+»)
 - 13) If (ТестПоТеме is отлично) and (ТекущийУровеньПодготовки is эксперт) then (УровеньСложности is «5+»)

Графически базу знаний можно представить в виде трехмерного графика, который представлен на рис. 3.10.

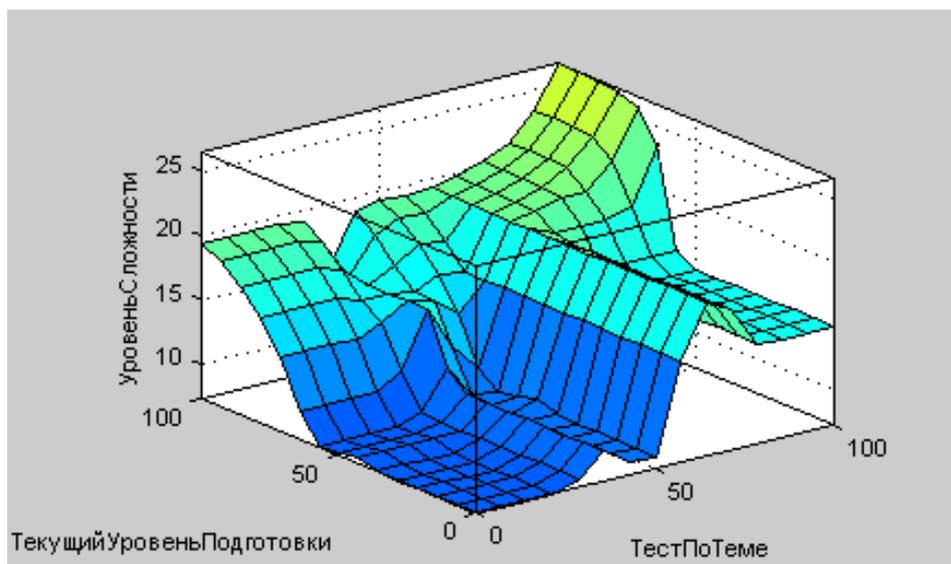


Рис. 3.10. График базы знаний индивидуально-ориентированной системы тестирования обучающихся

Таким образом, разработана нечеткая база знаний, позволяющая управлять процессом индивидуально-ориентированного тестирования, выявляющим знания обучающихся в зависимости от индивидуальных показателей. Настройка показателей построенных лингвистических переменных может изменяться в зависимости от требований конкретного преподавателя, а также корректироваться в процессе накопления статистической информации о ходе тестирования. Представленная база знаний решает проблему формализации слабо структурируемой системы, ввиду сложности оценки уровня знаний обучаемого четко заданными значениями. Данные, представляемые базой, будут использованы при совершенствовании процесса проведения контрольного тестирования студентов.

3.3 Построение алгоритма повышения качества тестовых заданий

В ходе построения алгоритмов тестирования важно включение в него алгоритмов расчета характеристик тестовых заданий, с целью их

использования для повышения качества тестовых заданий. Для определения любой качественной характеристики путем эксперимента, необходимо не менее 100-150 записей результатов ответов по каждому тестовому заданию. От количества респондентов зависит достоверность результатов исследования каждого тестового задания. Однако не все направления подготовки различных университетов имеют такой набор, отсюда возникает проблема выборки.

Для расчета качественных характеристик тестовых заданий исследователями разработан ряд подходов. В ходе их анализа выявлены основные недостатки, в число которых можно включить: обязательное условие наличия большой выборки испытуемых, отсутствие в алгоритме проверки качества тестовых заданий, отсутствие соотнесения результата тестирования обучающегося и его уровня подготовки, а так же отсутствие возможности повторного тестирования с целью определения параметров качества тестового задания (нет алгоритма, в обязательном порядке предоставляющем одно и то же тестовое задание обучающемуся дважды с целью увеличения выборки тестовых результатов).

Процесс оптимизации теста можно представить в виде линейной схемы (см. рис. 3.11).

Первый этап - предварительный анализ тестовых заданий. Для его осуществления необходимо провести такие мероприятия, как: отбор экспертов, определение уровня сложности каждого из заданий, апробация тестовых заданий на группе обучающихся (успешно освоивших материал данного курса), формирование рекомендаций по оптимизации уровня сложности теста на основе сопоставления оценки экспертов и результатов апробации теста на группе обучающихся [16].

После завершения данного этапа тест переносится в базу данных обучающей системы и начинается второй этап: расчет уровня подготовленности обучающегося и построение характеристической кривой

обучающегося. При построении графика определяется адекватность тестовых ответов обучающегося [17].

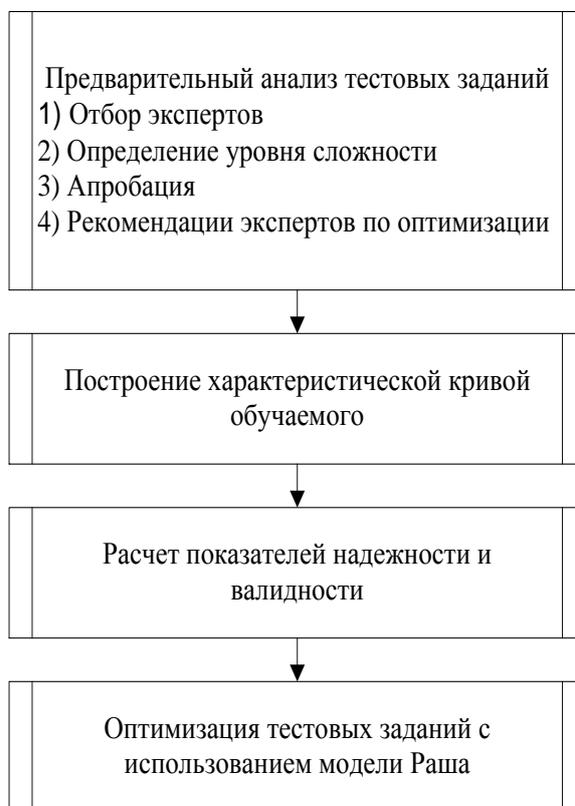


Рис. 3.11. Линейная схема оптимизации тестового задания

Для определения уровня знаний каждого учащегося проводятся предварительные тесты, апробированные ранее и отвечающие требованиям валидности и надежности. Такое тестирование осуществляется под контролем преподавателя систем видеонаблюдения за учебной аудиторией. После окончания тестирования, результаты заносятся в базу данных системы обучения, затем начинает работу модуль по определению уровня подготовленности обучаемого [24].

Для каждого обучаемого рассчитываются показатели уровня его подготовленности по дисциплине, на основе которых строятся характеристические кривые модели обучаемого, отражающей уровень подготовленности обучаемого, и тестового задания, показатели которой рассчитываются еще и по полученным ответам на текущее тестирование. На

основании построения двух кривых, делается вывод об адекватности результатов. При несовпадении кривых появляется вероятность списывания, помощи, плохого самочувствия и т.д. От результата сравнения показателей зависит работа алгоритмов, описанных далее.

При расчете надежности и валидности теста на третьем этапе, происходит построение матрицы, содержащей результаты тестирования, на основании которой, из теста удаляются неэффективные вопросы. Далее рассчитываются коэффициенты корреляции заданий между собой. Задания с коэффициентами меньше 0,3 отправляются в резерв, с целью достижения общего показателя надежности, который должен быть равен не менее, чем 0,8.

Оптимизация тестовых заданий – это заключительный, четвертый этап. Здесь осуществляется сопоставления сложности тестового задания и вероятности правильного ответа, учета объективности ответа обучаемого, а также учета коэффициента корреляции текущего тестового задания.

При определении вышеназванных характеристик, при малой выборке респондентов возникают сомнения в достоверности результата расчетов тестовых показателей. Для решения данной проблемы предлагается предоставлять каждое задание обучающемуся не менее 2-х раз, посредством их включения в тестовые блоки «Повторение изученного». При проверке теста и определении итогового балла, вводится коэффициент k , повышающий вероятность правильного ответа при повторном тестировании. Так, корректировку теста можно осуществлять каждые полгода. Алгоритм, описывающий применение данного положения отражен на блок-схеме рис. 3.12.

Рассмотрим подробнее алгоритм расчета качественных тестовых характеристик (рис. 3.13), применяемого с целью увеличения качества всего теста в целом.

В начале работы процедуры происходит сравнение коэффициента корреляции задания с установленной нормой, при несоответствии заданным

параметрам, система переходит к построению характеристической кривой тестового задания. Принцип построения графика и его использования аналогичен этим же процессам в отношении модели обучаемого. По окончании тестирования, на графиках каждого из заданий откладываются точки, по отклонению от нормы которых можно судить об адекватности тестового задания.

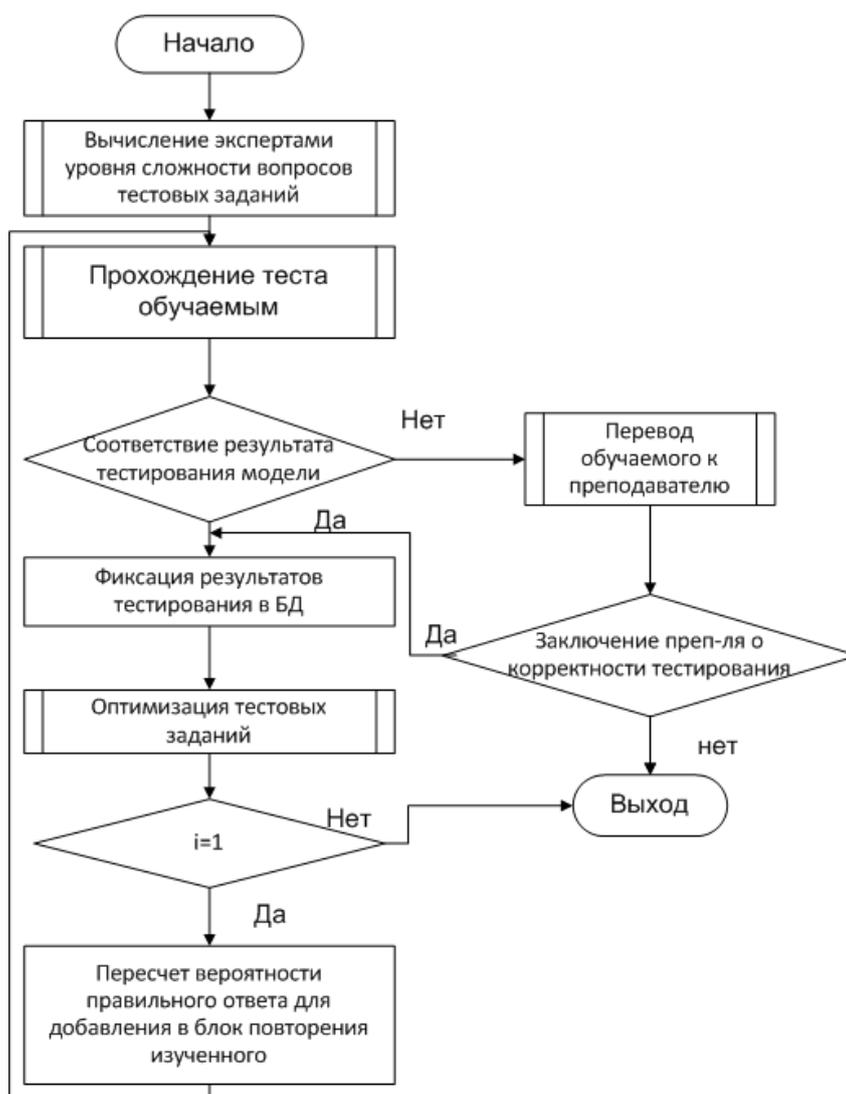


Рис. 3.12. Общий алгоритм работы с тестовым заданием

Стоит отметить, что задания, оцененные, как непригодные для тестирования отправляются в резерв, ввиду малой выборки испытуемых, и предлагаются следующим потокам обучающихся для прохождения, пока количество результатов респондентов не превысит 100 человек. В тест данные задания включаются из расчета 2 задания на один тест, ввиду

необходимости сохранения уровня надежности и валидности теста в целом. Если после тестирования заданного количества респондентов, задания по-прежнему неадекватны, они помечаются, как пригодные для удаления. Далее, эксперт, отвечающий за оптимизацию теста, либо отправляет данный тест на переработку группе экспертов, либо окончательно удаляет его.

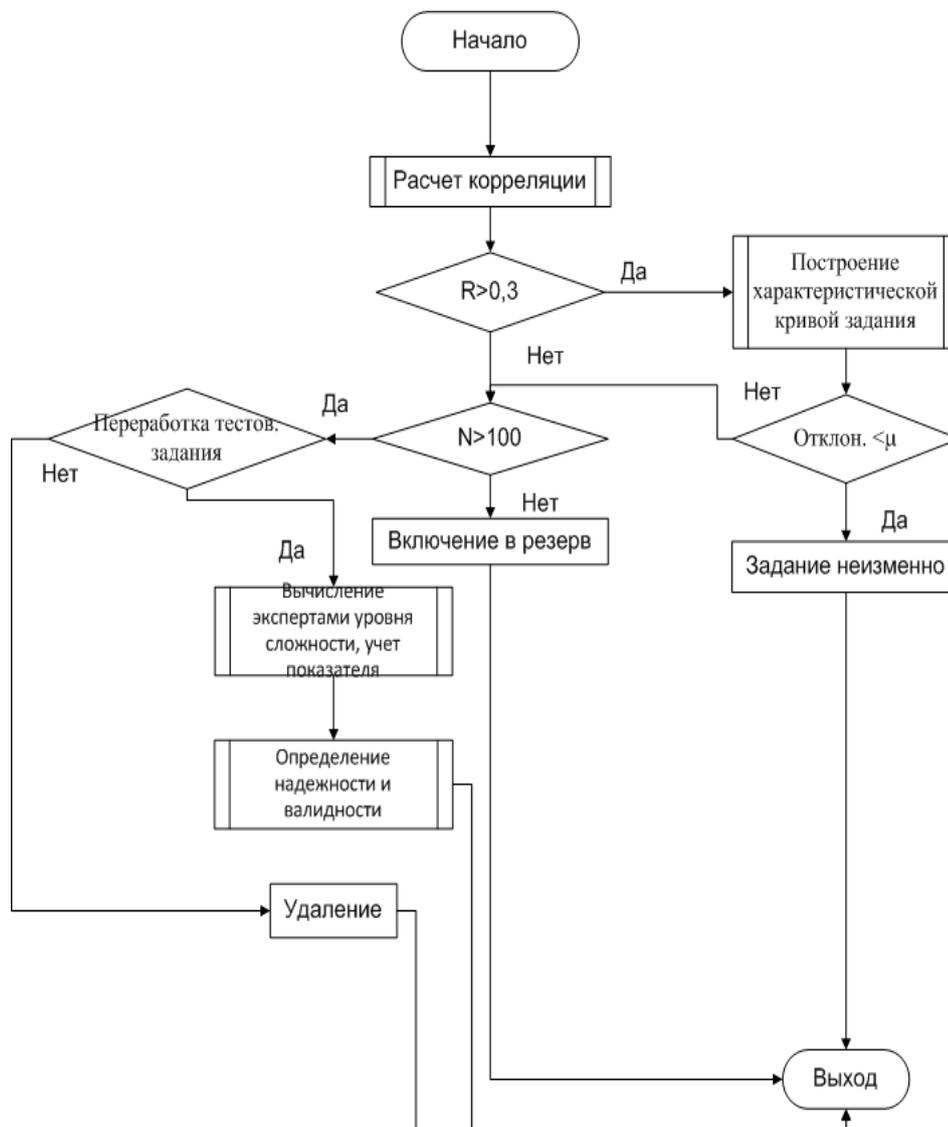


Рис. 3.13. Алгоритм оптимизации при расчете характеристик тестового задания

Таким образом, разработан алгоритм оценки качества тестовых заданий при малой выборке испытуемых, позволяющий организовать тестирование с анализом результатов на предмет их надежности.

Внедрение в СЭО «Пегас» разработанных схем, алгоритмов, и нечеткой базы знаний, позволит повысить эффективность образовательного

процесса, ввиду индивидуально-ориентированного подхода к процессу тестирования обучаемых. Использование разработанных схем и алгоритмов работы сократит время, затрачиваемое на контрольные мероприятия и повысит качество знаний студентов.

Согласно проведённому анализу исследователей рассматриваемой предметной области, применение алгоритмов индивидуально-ориентированного тестирования, позволяет повысить результативность обучения (уменьшение пробелов в знаниях) и сократить время, затрачиваемое на контрольные мероприятия.

3.4 Совершенствование базы данных системы электронного обучения НИУ «БелГУ»

Для хранения всей информации, хранящейся в системе электронного обучения, использующей разработанные в пунктах 3.1 – 3.3 алгоритмы и модели, используется представленная на рис. 3.14 структура базы данных. Рассмотрим подробнее сущности, представленные на схеме

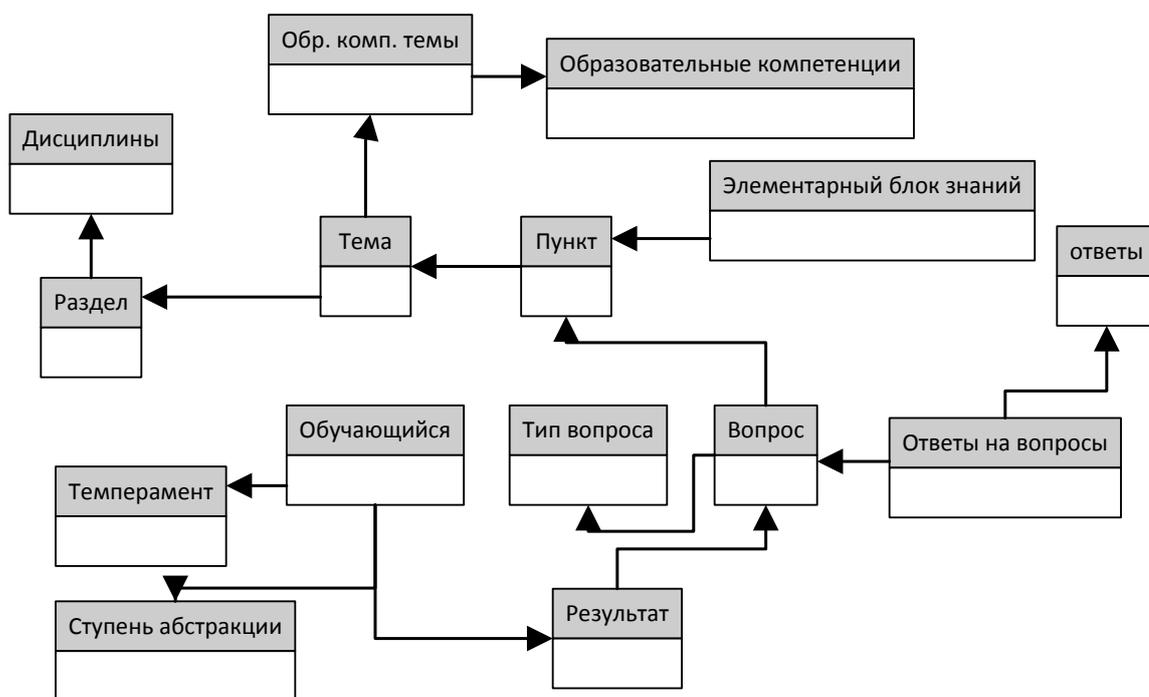


Рис. 3.14. Структура базы данных системы электронного обучения

Таблица Тип вопроса содержит информацию о типах вопросов, которые будут использоваться в процессе тестирования обучаемых. Атрибуты таблицы представлены в таблице 3.7

Таблица 3.7

Атрибуты таблицы «Тип вопроса»

Имя поля	Тип данных	Описание
Id_typeQuestion	Счетчик	Счетчик (первичный ключ)
nameQustion	Текстовый	Тип вопроса
note	Текстовый	Примечание

Таблица «Обучающиеся» включает в себя базовую информацию о каждом обучаемом. В нее входят как персональные данные, вводимые обучаемым или администратором при создании учетной записи пользователя, так и рассчитанные в ходе предварительного тестирования различного рода личностные показатели обучающегося. Атрибуты таблицы представлены в таблице 3.8

Таблица 3.8

Атрибуты таблицы «Обучающиеся»

Имя поля	Тип данных	Описание
Id_trainee	Счетчик	Счетчик (первичный ключ)
surname	Текстовый	Фамилия
name	Текстовый	Имя
Patronymic	Текстовый	Отчество
id_group	Числовой	Группа
Mail	Текстовый	Почта
login	Текстовый	Логин
password	Текстовый	Пароль
id_temperament	Числовой	Тип темперамента
capacity	Текстовый	Коэффициент времени тестирования
id_abstratskiyaStep	Числовой	Степень абстракции

Таблица «Образовательные компетенции темы» является средством сопоставления каждой темы с различными компетенциями. Атрибуты таблицы представлены в таблице 3.9

Таблица 3.9

Атрибуты таблицы «Образовательные компетенции темы»

Имя поля	Тип данных	Описание
Id_themeOk	Счетчик	Счетчик (первичный ключ)
Id_theme	Числовой	Тема
Id_ok	Числовой	Компетенция
Note	Текстовый	Примечание

Таблица «Темы» содержит в себе материал и описание блоков знания уровня «Тема» Атрибуты таблицы представлены в таблице 3.10

Таблица 3.10

Атрибуты таблицы «Тип вопроса»

Имя поля	Тип данных	Описание
Id_theme	Счетчик	Счетчик (первичный ключ)
codeTheme	Числовой	Шифр темы, учитывающий шифр раздела
Theme	Числовой	Тема
Purpose	Текстовый	Цель темы
Problem	Текстовый	Задачи темы
library	Текстовый	Список литературы
Id_section	Числовой	Раздел
baseTheme	Логический	Базовая тема или расширенная (вариатив)
note	Текстовый	Примечание

Таблица «Тип темперамента» содержит перечень типов темперамента и описание каждого из них Атрибуты таблицы представлены в таблице 3.11

Таблица 3.11

Атрибуты таблицы «Тип темперамента»

Имя поля	Тип данных	Описание
Id_temperament	Счетчик	Счетчик (первичный ключ)
temperament	Текстовый	Тип темперамента
description	Поле МЕМО	Краткое описание типа темперамента

Таблица «Разделы» содержит в себе материал и описание блоков знания уровня «Раздел» Атрибуты таблицы представлены в таблице 3.12

Атрибуты таблицы «Разделы»

Имя поля	Тип данных	Описание
Id_section	Счетчик	Счетчик (первичный ключ)
codeSection	Числовой	Шифр раздела, учитывающий шифр дисциплины
section	Текстовый	Раздел
Id_discipline	Числовой	Дисциплина
note	Текстовый	Примечание

Таблица «Ступени абстракции» включает описание различных ступеней абстракции, которые применяются для определения типа материала, который будет наиболее полезным в усвоении материала обучающимся. Атрибуты таблицы представлены в таблице 3.13.

Таблица 3.13

Атрибуты таблицы «Ступени абстракции»

Имя поля	Тип данных	Описание
Id_abstratskiyaStep	Счетчик	Счетчик (первичный ключ)
nameAbstratskiya	Текстовый	Степень абстракции
nameCutting	Текстовый	Краткое обозначение
Description	Поле МЕМО	Описание ступени абстракции

Таблица «Ответы на вопросы» соотношения вопросов и ответов тестовых заданий. Атрибуты таблицы представлены в таблице 3.14.

Таблица 3.14

Атрибуты таблицы «Ответы на вопросы»

Имя поля	Тип данных	Описание
Id_responseQuestion	Счетчик	Счетчик (первичный ключ)
Id_question	Текстовый	Вопрос
Id_response	Текстовый	ответ

Таблица «Дисциплины» содержит в себе материал и описание блоков знания уровня «Дисциплина» Атрибуты таблицы представлены в таблице 3.15

Таблица 3.15

Атрибуты таблицы «Дисциплины»

Имя поля	Тип данных	Описание
Id_discipline	Счетчик	Счетчик (первичный ключ)
codeDiscipline	Числовой	Шифр дисциплины
Discipline	Числовой	название
Purpose	Текстовый	Цель дисциплины
Problem	Текстовый	Задачи дисциплины
automationCoefficient	Числовой	коэффициент автоматизации, задается преподавателем в случае, когда требования к нему жесткие в зависимости от профессиональной деятельности
Id_assimilationLevel	Числовой	Уровень усвоения
Id_sensiblenessAssimilation	Числовой	Осознанность усвоения
Note	Текстовый	примечание

Таблица «Элементарный блок знаний» содержит в себе материал и различные показатели элементарных блоков знания Атрибуты таблицы представлены в таблице 3.16

Таблица 3.16

Атрибуты таблицы «Элементарный блок знаний»

Имя поля	Тип данных	Описание
1	2	3
Id_ebk	Счетчик	Счетчик (первичный ключ)
codeEbk	Числовой	код, используемый для определения к какой теме он относится
Ebk	Текстовый	имя элементарного блока знания
id_theme	Числовой	Тема
contentB1	Поле MEMO	содержимое блока 1 степень абстракции
contentB2	Поле MEMO	содержимое блока 2 степень абстракции
contentB3	Поле MEMO	содержимое блока 3 степень абстракции
contentB4	Поле MEMO	содержимое блока 4 степень абстракции

Таблица 3.16 продолжение

1	2	3
id_typeMaterial	Числовой	тип материала (понятие, теорема, пример, задача)
volumeInformation	Текстовый	объем информации
Note	Текстовый	Примечание

Таблица «Образовательные компетенции» содержит перечень компетенций, их описание и характеристики. Атрибуты таблицы представлены в таблице 3.17

Таблица 3.17

Атрибуты таблицы «Образовательные компетенции»

Имя поля	Тип данных	Описание
Id_ok	Счетчик	Счетчик (первичный ключ)
Ok	Текстовый	полное наименование компетенции
CuttingOk	Текстовый	сокращенное наименование компетенции
Know	Текстовый	Знать
Ability	Текстовый	Умение
Own	Текстовый	Владеть
Note	Текстовый	Примечание

Таблица «Группы» содержит перечень групп и их соотношение с конкретным профилем подготовки. Атрибуты таблицы представлены в таблице 3.18

Таблица 3.18

Атрибуты таблицы «Группы»

Имя поля	Тип данных	Описание
Id_group	Счетчик	Счетчик (первичный ключ)
codeGroup	Текстовый	Номер группы
id_preparationProfile	Числовой	Профиль подготовки
Note	Текстовый	Примечание

Таблица «История обучения» содержит историю обучения каждого пользователя. Атрибуты таблицы представлены в таблице 3.19

Атрибуты таблицы «История обучения»

Имя поля	Тип данных	Описание
Id_historyEducation	Счетчик	историяобучения history of education
dateRegistration	Дата/время	дата внесения записи
timeRegistration	Дата/время	время внесения записи
id_trainee	Числовой	Обучаемый
id_discipline	Числовой	дисциплина (параметры обучаемого по дисциплине)
id_assimilationLevel	Числовой	уровень усвоения дисциплины
sensiblenessAssimilation	Числовой	уровень осознанности дисциплины
assimilationSpeed	Числовой	Скорость усвоения материала (бит/сек) по дисциплине
id_ok	Числовой	образовательная компетенция (параметры обучаемого по компетенциям)
knowK	Числовой	коэффициент знания
abilityK	Числовой	коэффициент умения
ownK	Числовой	коэффициент владения
okK	Числовой	коэффициент компетенция

Таблица «Вопросы» содержит перечень вопросов тестовых заданий, а так же их числовые показатели и характеристики, разъяснение каждого из вопросов. Атрибуты таблицы представлены в таблице 3.20

Таблица 3.20

Атрибуты таблицы «Вопросы»

Имя поля	Тип данных	Описание
1	2	3
id_question	Счетчик	Счетчик (первичный ключ)
id_ebk	Числовой	элементарный блок знания, к которому относится вопрос
Question	Текстовый	Вопрос
id_typeQuestion	Числовой	тип вопроса
Image	Поле объекта OLE	Изображение
timeMinuteExpert	Числовой	время ответа на вопрос эксперта (минуты)

Таблица 3.20 продолжение

1	2	3
timeSecExpert	Числовой	время ответа на вопрос эксперта (секунды)
complexity	Числовой	сложность вопроса
explainingQuestion	Текстовый	разъяснение вопроса
Reliability	Числовой	Надежность вопроса
Validity	Числовой	валидность вопроса
Note	Текстовый	Примечание

Таблица «Ответы» содержит множество вариантов ответа на все тестовые задания обучающей системы; включает характеристики каждого ответа. Атрибуты таблицы представлены в таблице 3.21

Таблица 3.21

Атрибуты таблицы «Ответы»

Имя поля	Тип данных	Описание
id_response	Счетчик	Счетчик (первичный ключ)
Response	Текстовый	вариант ответа
responseLine	Текстовый	ответ на соответствие
Correct	Логический	правильный ответ
id_weightResponse	Числовой	вес ответа
explainingResponse	Текстовый	разъяснение ответа
Note	Текстовый	Примечание

Такаим образом, рассмотрена база данных системы электронного обучения e-front. Для реализации разработанных моделей и алгоритмов в базу данных дабовленных поля, в которых будут прописываться индивидуальные характеристики студентов.

4 РЕАЛИЗАЦИЯ РАЗРАБОТАННЫХ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ И АЛГОРИТМОВ ИНДИВИДУАЛЬНО- ОРИЕНТИРОВАННОГО ОБУЧЕНИЯ.

4.1 Настройка и разработка модулей индивидуально-ориентированной системы обучения

При реализации разработанного математического и алгоритмического обеспечения были изменены модуль системы электронного обучения «module_complete_test» (Приложение А) и обработчики content.js (Приложение Б), tests.js (Приложение В).

В первый модуль «module_complete_test» добавлен программный код, позволяющий записать индивидуальные характеристики обучающегося в качестве значения одного из атрибутов таблицы об обучающихся базы данных системы. Создание тестов, записывающих данные параметры, реализовано в виде отдельных функций скрипта. Вызов функции осуществляется при создании соответствующего нового элемента. Так, на рис. 4.1 приведён пример реализации таких тестов в блоке «Психофизиологические параметры»

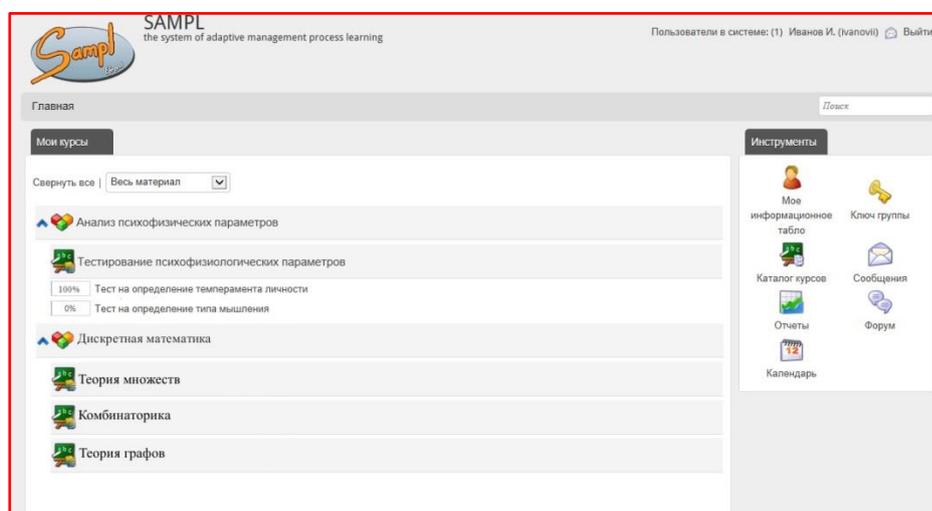


Рис. 4.1. Интерфейс главного меню студента

Так, к примеру, при создании теста на определение типа мышления, администратор системы указывает параметры присвоения того или иного типа мышления в зависимости от количества баллов, набранных по тесту. Создание и настройку самого теста, осуществляет преподаватель-эксперт.

Все характеристики, присеваемые студенту при помощи создания таких тестов, являются атрибутами таблицы «Обучающиеся», которая подробнее рассматривается в главе 3.4.

Оценка текущих знаний студента на основе ретроспективной информации реализуется также через обращение в базе данных системы, а именно к таблице «История обучения». Параметры, хранящиеся по каждому из студентов, перезаписываются в ходе освоения курса. Алгоритм расчета и изменения показателей также реализован через модуль «`module_complete_test`».

Настройка контента в зависимости от индивидуальных параметров обучающегося прописывается в функциях файла `content.js`. Так, работа функции `ContAbstrakciya` заключается в считывании информации об обучающемся из базы данных и последующем выводе на экран контента соответствующего ступени абстракции пользователя. Стоит отметить, что при создании курса, преподаватель может предоставить контент только для двух или даже одной ступени абстракции. В таком случае редактор курса указывает, на какой контент направлять студента в случае отсутствия нужной степени абстракции. Пример организации учебного материала приводится на рис. 4.2.

Реализация тестирования, работающего на основе нечеткой базы знаний, прописана в скриптовом файле системы `tests.js`. При старте тестирования, система предлагает тестируемому студенту задания уровня знаний, соответствующего текущей успеваемости студента. После прохождения 5 заданий система анализирует результаты и при слишком высоких или низких коэффициентах выдает обучающемуся задания более низкого или высокого уровня.

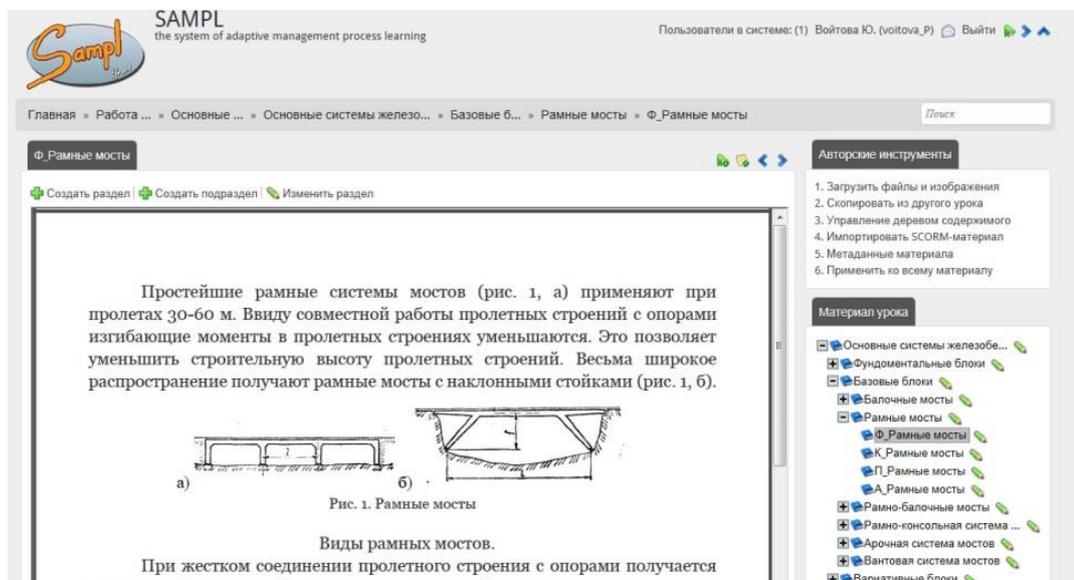


Рис. 4.2. Контент учебного курса в системе электронного обучения.

Пример работы системы тестирования приводится на рис. 4.3. На правой панели видны ссылки на фундаментальные, вариативны и базовые блоки курса. В зависимости от результата тестирования, система выводит пользователю ссылки на темы, в которых обучающийся плохо ориентируется, согласно результатам тестирования. Если тестирование имеет несколько попыток, обучающийся может его повторить после повторения материала.

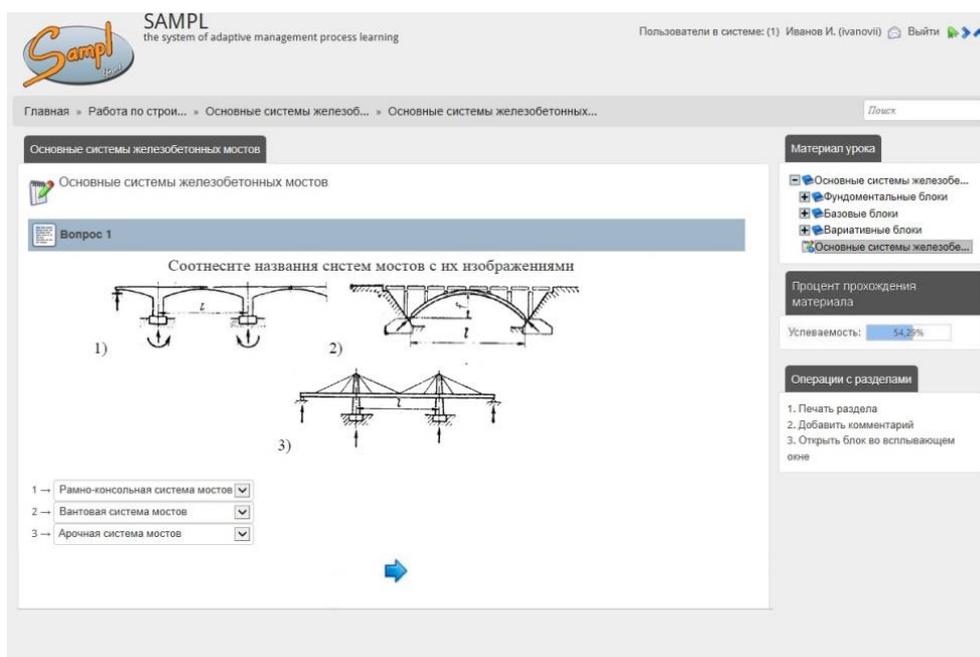


Рис. 4.3. Тестирование в системе индивидуально-ориентированного обучения

В этом же скриптовом файле находится функция, реализующая алгоритм анализа качества тестовой базы, описанный в главе 3.3. Так, для получения нужной выборки испытуемых, алгоритм генерации случайных вопросов настроен так, что для каждого обучаемого каждое из тестовых заданий выводится дважды: первый в одной попытке и второй в другой. В дальнейшем, при достижении нужной выборки будет реализован алгоритм автоматического подсчета характеристик теста.

Таким образом, описана реализация представленных в главе 3 алгоритмов и моделей поддержки индивидуально-ориентированного обучения.

4.2 Тестирование работы модулей индивидуально-ориентированной системы обучения

Для проверки работы системы была создана экспериментальная группа в количестве 20 человек. В состав группы входили студенты 2 курса НИУ «БелГУ». Группе предлагалось опробовать обучение по демо-курсу «Дискретная математика», разработанному для системы электронного обучения e-front.

В начале обучения студенты должны были пройти небольшие тесты, определяющие тип их темперамента и степень абстракции, при которой контент наиболее усваивается для студента. Результаты прохождения данного этапа представлены в таблице 4.1.

Таблица 4.1

Результаты вводного тестирования студентов

Наименование опрошенного	Среднее время на ответ	Темперамент	Текущая степень абстракции
1	2	3	4
Студент 1	0:21	Флегматик	Качественная
Студент 2	0:19	Сангвиник	Качественная

Таблица 4.1 продолжение

1	2	3	4
Студент 3	0:18	Меланхолик	Количественная
Студент 4	0:09	Холерик	Феноменологическая
Студент 5	0:20	Флегматик	Качественная
Студент 6	0:17	Меланхолик	Количественная
Студент 7	0:12	Сангвиник	Феноменологическая
Студент 8	0:10	Сангвиник	Количественная
Студент 9	0:22	Меланхолик	Качественная
Студент 10	0:11	Холерик	Качественная
Студент 11	0:08	Холерик	Качественная
Студент 12	0:22	Сангвиник	феноменологическая
Студент 13	0:17	Сангвиник	Количественная
Студент 14	0:10	Флегматик	Аксиоматическая
Студент 15	0:11	Сангвиник	Качественная
Студент 16	0:08	Холерик	Феноменологическая
Студент 17	0:14	Сангвиник	Качественная
Студент 18	0:21	Флегматик	Количественная
Студент 19	0:15	Меланхолик	Аксиоматическая
Студент 20	0:17	Сангвиник	Качественная

Далее студентам предлагалось перейти к изучению первой темы курса. С соответствии с результатами теста, система представила каждому обучающему контент, изложенный в наиболее доступной для понимания степени абстракции. Пример контента по первой теме представлен в Приложении Г. Незначительная часть студентов, сменили степень абстракции в процессе изучения материала.

После изучения теории, система предлагала студентам пройти тестирование по теме. Рассмотрим подробнее процесс тестирования одного из студентов, результат которого представлен в таблице 4.2. Тестируемый студент имеет среднюю успеваемость на уровне «хорошо». В соответствии с заданными правилами тестирования и разработанной базой знаний, система в ходе тестирования предлагает учащемуся решить более сложные задания. Попытка оказывается успешной. В результате тестирования обучающийся получает оценку «Отлично».

Ход тестирования студента №1, имеющего текущую успеваемость
«хорошо»

Номер вопроса	Коэф-т правильности выполнения	Сложность	Затраченное время
1	1	«4»	20 сек
2	1	«4»	23 сек
3	0,7	«4»	35 сек
4	0,75	«4»	32 сек
5	0,9	«4»	28 сек
6	0,94	«5»	32 сек
7	0,96	«5»	41 сек
8	0,97	«5»	40 сек

Таким образом, результаты тестирования системы показали корректные результаты, то позволяет судить о правильности функционирования реализованных алгоритмов.

4.3 Оценка эффективности использования в СЭО алгоритмического и математического обеспечения индивидуально- ориентированной системы обучения

Система электронного обучения на образовательной платформе e-front имеет бесплатную лицензию с открытым программным кодом. Поэтому эффективность работы системы стоит оценивать с точки зрения повышения показателей усвояемости материала. Так, на основе результатов тестирования разработанных элементов системы и опроса преподавателей и студентов, можно отметить улучшение таких показателей как: качество знаний, успеваемость, скорость усвоения, объективность тестирования, удовлетворенность обучающихся возможностями системы электронного обучения. Более подробно показатели эффективности обучения по названным критериям представлены в таблице 4.3.

Таблица 4.3

Показатели эффективности обучения в системе электронного обучения при использовании разработанных алгоритмов.

№	Показатель	Результат до внедрения	Результат после внедрения
1	Качество знаний (по одному курсу)	64%	83%
2	Успеваемость	70%	91%
3	Скорость усвоения темы (при использовании индивидуализированного контента), среднее значение по эксп. Группе	3 ч 24 мин	2 ч 12 мин
4	Объективность тестирования (на примере 1 из тестов при последующем опросе преподавателя)	67% ответов обоснованы	94% ответов обоснованы
5	Удовлетворенность обучающихся СЭО	6 из 10	9 из 10

Таким образом, проведена оценка эффективности образовательного процесса после внедрения разработанных моделей и алгоритмов в работу системы.

ЗАКЛЮЧЕНИЕ

В результате выполнения магистерской диссертации были выполнены все поставленные задачи. Проведен анализ подходов к учету индивидуальных особенностей обучающихся, включая их психологические особенности и ретроспективную информацию об обучении. Рассмотрен ряд адаптивных алгоритмов, использующихся для реализации индивидуально-ориентированных подходов в процессе работы электронных систем обучения.

Также проведено исследование систем электронного обучения, использующихся как на очных, так и на заочных отделениях образовательных организаций (Moodle, E-front, LAMS, Sakai, ATutor, Claroline, Dokeos, OLAT, OpenACS, ILIAS). В качестве критериев сравнения систем были использованы показатели: поддержка стандарта SCORM, интеллектуальное управление контентом, языки приложения, используемая системой СУБД, тип лицензии, наличие русскоязычного интерфейса, система проверки знаний, демонстрационный сервер. По результатам анализа систем составлена сравнительная таблица, на основании которой можно сделать вывод о распространенности бесплатной лицензии с открытым программным кодом у всех рассматриваемых систем. Большая часть прикладных решений имеет русскоязычный интерфейс приложения и возможность настройки ряда элементов для построения индивидуальных образовательных траекторий.

Проведен анализ использующейся в настоящее время в НИУ «БелГУ» системы электронного обучения, работающей на образовательной платформе Moodle. Выделены основные функции и элементы системы. По результатам анализа системы выделены основные недостатки СЭО и составлены

рекомендации по возможному ее усовершенствованию. Выявлено, что для реализации предлагаемых в результате анализа функций необходимо создавать новые модули. В связи с этим для реализации алгоритмов предлагается образовательная платформа e-front, включающая в себя необходимые для индивидуально-ориентированного обучения модули.

Проведено моделирование процессов системы индивидуально ориентированного обучения. Построены контекстная и декомпозиционная диаграммы в нотации IDEF0, описывающие работу процессов в ходе освоения курса. Для проектирования системы также использован инструмент сети Петри. Спроектированные сети позволяют отразить поведение системы в процессах тестирования обучаемого и формирования образовательного контента, когда от активности состояний двух параллельных процессов, подразумевающих работу двух пользователей, зависит совершение какого-либо перехода.

Разработана нечеткая база знаний, позволяющая управлять процессом индивидуально-ориентированного тестирования. Для ее функционирования построены лингвистические переменные, позволяющие настраивать показатели уровня знаний или текущих результатов тестирования в зависимости от требований преподавателя. Основное назначение разработанной базы знаний – корректировка количества и последовательности тестовых заданий в результате накопления статистической информации о ходе тестирования. Так, данные, предоставляемые базой, могут быть использованы для выявления реального уровня знаний обучающихся.

Разработан алгоритм оценки качества тестовых заданий при малой выборке испытуемых, позволяющий организовать процесс корректировки теста для его использования при работе в системе индивидуально-ориентированного электронного обучения студентов.

Предложена усовершенствованная структура базы данных системы электронного обучения. В ее таблицы добавлены поля, заполнение которых необходимо для реализации индивидуально-ориентированного обучения.

Предложена реализация разработанных моделей и алгоритмов в программной среде E-front. Разработанное в магистерской диссертации алгоритмическое и математическое обеспечение реализовано в системе электронного обучения e-front через модуль системы электронного обучения «module_complete_test» и обработчики content.js, tests.js. Проведено тестирование системы, реализованное в виде прохождения экспериментальной группой студентов входного тестирования для записи в систему индивидуальных показателей. Так же апробировано решение адаптивного представления контента в зависимости от уровня абстракции обучающихся. Опробовано тестирование, работающее на основе нечеткой базы знаний. В результате всех тестов получены корректные результаты работы системы.

В результате, внедрение в СЭО «Пегас» разработанных схем, алгоритмов, и нечеткой базы знаний, позволяет повысить эффективность образовательного процесса, за счет результативности обучения и сокращения времени, затрачиваемого на контрольные мероприятия. Таким образом, в ходе выполнения магистерской диссертации, поставленная цель исследования, которая ориентирована на повышение эффективности образовательного процесса посредством учета индивидуальных особенностей обучающихся в системе электронного обучения НИУ «БелГУ».

СПИСОК ЛИТЕРАТУРЫ

1. Brusilovsky P., Corbett A., de Rosis F. User Modeling. Springer, 2003.
2. Brusilovsky, P. Methods And Techniques Of Adaptive Hypermedia. User Modeling and User-Adapted Interaction. 6 (2-3). 1996. P. 87-129.
3. Аванесов В.С. Композиция тестовых заданий [Текст]/В.С. Аванесов// М.: АДЕПТ. – 1998. – стр. 217
4. Асадуллаев Рустам Геннадьевич. Автоматизация процесса индивидуальной подготовки кадров в системе управления персоналом промышленного предприятия: автореферат дис. ... кандидата технических наук: 05.13.06 / Асадуллаев Рустам Геннадьевич; [Место защиты: Государственный университет - учебно-научно-производственный комплекс - ФГБОУВПО].- Орел, 2013
5. Асадуллаев, Р.Г. Организация интеллектуального управления индивидуальными образовательными траекториями [Текст] / В.В. Ломакин, Р.Г. Асадуллаев // Научные ведомости Белгородского государственного университета: научный журнал. – Белгород: Издательский дом «Белгород» - 2013. - №22(165) вып.28/1. – С. 167–174.
6. Астанин, С.В. Разработка индивидуальной модели поведения обучаемого в системе дистанционного образования [Текст]/ С.В. Астанин, Т.Г. Калашникова // Перспективные информационные технологии и интеллектуальные системы – 2001 – №1 – С. 179-196.
7. Астанин. С.В. Сопровождение процесса обучения на основе нечеткого моделирования [Электронный ресурс]/ С.В. Астанин // Дистанционное образование. – 2000. – № 5. – Режим доступа: http://www.e-joe.ru/sod/00/5_00/as.html

8. Атанов, Г.А. Обучение и искусственный интеллект, или Основы современной дидактики высшей школы [Текст]/ Г.А/ Атанов, И.Н/ Пустынникова/ Донецк: Изд-во ДООУ, 2002. – 504 с.
9. Ахрем, А. А. Нечеткий логический вывод в системе принятия решений [Текст]/ А.А. Ахрем, М.Р. Ашинянц, С.А. Петров//Труды ИСА РАН. – 2007. – Т. 29. – с.265-275.
10. Белозубов, А. В. Система дистанционного обучения Moodle: [Электронный ресурс]: Учебно-методическое пособие/ А. В. Белозубов, Д. Г. Николаев//СПб.:СПбГУ ИТМО, 2007. - 108 с.
11. Богомолов, В.А. Обзор бесплатных систем управления обучением [Текст]/В.А. Богомолов// Educational Technology & Society, 2007. – V.10(3). – С. 460-466
12. Брусиловский П. Л. Адаптивные интеллектуальные технологии в сетевом обучении [Текст]/ П.Л. Брусиловский // Новости искусственного интеллекта, 2002. - № 5. – С. 25–31
13. Брусиловский, П.Л. Адаптивные обучающие системы в World Wide Web: обзор имеющихся в распоряжении технологий [Электронный ресурс]/ П.Л. Брусиловский// International Forum of Educational Technology & Society – Режим доступа: <http://roman.by/r-5716.html>, свободный.
14. Булыгин, В.Г. Автоматизация учебного процесса [Электронный ресурс] / В.Г. Булыгин// Основы автоматизации процесса обучения. – Режим доступа:http://www.eusi.ru/lib/bulygin_osnovy_avtomatizacii_processa_obucenia/2.shtml, свободный
15. Буль, Е. Е. Обзор моделей студента для компьютерных систем обучения [Текст]/ Е. Е. Буль // Educational Technology & Society. – 2003. – Vol. 6 (4). – P. 245–250
16. Васильев В. И., Основы культуры адаптивного тестирования [Текст] /В.И. Васильев, Т. Н. Тягунова // М.: Национальный институт бизнеса, 2003. – 370с

17. Войтова, Ю.Ю. Анализ подходов, используемых в процессе обучения кадров промышленных предприятий с применением информационных технологий [Текст]// Р.Г. Асадуллаев, Ю.Ю. Войтова/ «Современные проблемы и перспективы управления развитием инновационной экономики»: мат. Всерос. науч. - практ. конф. с межд. уч. (г. Белгород, 18-19 октября 2012 г) / отв. ред. И.С. Шаповалова. – Белгород: ИД «Белгород» НИУ «БелГУ», 2012. – 365-369 с.

18. Демидов, Д. Г. Разработка моделей и алгоритмов автоматизации процессов адаптивного обучения специалистов для предприятий [Текст]: автореферат дис. ... кандидата технических наук (05.13.06)7 / Дмитрий Григорьевич Демидов; ГОУ ВПО «Московский государственный университет печати им. Ивана Федорова». – Москва, 2011. – 21 с.

19. Денисова, И. Ю. Реализация адаптивной технологии обучения в информационной обучающей системе [Текст]// Известия Пензенского государственного педагогического университета им. В.Г. Белинского. – №28. – 2012. – 749-752с.

20. Живенков, А.Н. Формирование плагинов LMS Moodle для адаптивного построения структуры курса электронного обучения[Текст]/ А.Н. Живенков, О.Г. Иванова // Научные ведомости БелГУ. Серия «История. Политология. Экономика. Информатика». - 2010. - № 19(90). - С. 150 - 156.

21. Живенков, А.Н. Формирование плагинов LMS Moodle для адаптивного построения структуры курса электронного обучения / А.Н. Живенков, О.Г. Иванова // Научные ведомости БелГУ. Серия «История. Политология. Экономика. Информатика». - 2010. - № 19(90). - С. 150 - 156.

22. Зайцев, И.Е. Адаптивные технологии в современных автоматизированных обучающих системах [Текст]// И.Е. Зайцев/ Журнал «Известия Российского государственного педагогического университета им. А.И. Герцена». - № 51. – 2008. – 214-217с.

23. Зайцева Л.В. Адаптация в компьютерных системах на базе структуризации объектов обучения [Текст]/ Л.В. Зайцева, Е.Е. Буль // Educational Technology & Society – 2006 – Vol.9(1).– P. 1-6.
24. Келеберда, И.Н. Перспектива реализации индивидуальной траектории обучения для хранилищ с SCORM-курсами [Электронный ресурс]/И.Н. Келеберда, Д.С. Негурица, В.В. Сокол// Educational Technology & Society. – 2008. – V.11(3). – ISSN 1436-4522. - [Режим доступа]: http://ifets.ieee.org/russian/depository/v11_i3/html/10.htm
25. Ким В. С. Тестирование учебных достижений[Текст]: монография/ В.С. Ким. –Уссурийск: Издательство УГПИ, 2007. – 214 с.
26. Колесникова, И.А Основы андрагогики/ Колесникова И.А.// М.: «Академия», 2003. - 412с.
27. Кудрявцев, В.Б. Моделирование процесса обучения [Текст] / В.Б. Кудрявцев, П.А. Алисейчик, К. Вашик, Ж. Кнап, А.С. Строгалов, С.Г. Шеховцов // «Интеллектуальные системы», сб. статей./ Москва: РГГУ, 2006. – т. 10, вып. 1-4 – 189-270 с.
28. Мазурок, Т.Л. Интеллектуальные средства автоматизации управления обучением [Электронный ресурс]/ Т.Л. Мазурок// Educational Technology & Society. – 2012 – V. 15(3). – P. 502-521. – Режим доступа: http://ifets.ieee.org/russian/depository/v15_i3/html/14.htm, свободный.
29. Майоров, А.Н. Теория и практика тестов для системы образования. (Как выбирать, создавать и использовать тесты для целей образования) [Текст]/ А. Н. Майоров. – Москва: Интеллект-центр, 2001. – 296с.
30. Метрическая система Георга Раша Rasch Measurement (RM) Электронный ресурс. / В.С.Аванесов. - Режим доступа: <http://testolog.narod.ru>.
31. Олейник Н.М. Тест как инструмент измерения уровня знаний и трудности заданий в современной технологии обучения. Учебное пособие по спецкурсу [Текст] / Н.М. Олейник// Донецк: Донецкий государственный университет, 1991. – 168 с.

32. Официальный сайт ATutor [Электронный ресурс]. – Электрон. дан. – ATutor, 2002-2012. – Режим доступа: <http://www.atutor.ca>, свободный
33. Официальный сайт Claroline [Электронный ресурс]. – Электрон. дан. – Claroline, 2015. – Режим доступа <http://www.claroline.net/>, свободный
34. Официальный сайт Dokeos [Электронный ресурс]. – Электрон. дан. – Dokeos, 2015. – Режим доступа <http://www.dokeos.com/>, свободный
35. Официальный сайт eFront [Электронный ресурс]. – Электрон. дан. – Режим доступа: <http://www.efrontlearning.net>, свободный
36. Официальный сайт ILIAS [Электронный ресурс]. – Электрон. дан. – ILIAS, 2015. – Режим доступа <http://www.ilias.de/>, свободный
37. Официальный сайт OLAT [Электронный ресурс]. – Электрон. дан. – OLAT, 2013. – Режим доступа <http://www.olat.org/>, свободный
38. Официальный сайт Sakai [Электронный ресурс]. – Электрон. дан. – Apereo Foundation, 2014. – Режим доступа: <https://www.sakaiproject.org/>, свободный
39. Официальный сайт НИУ "БелГУ"[Электронный ресурс]. - Электрон.дан. - НИУ «БелГУ», 1999 – 2015. - Режим доступа: <http://www.bsu.edu.ru/bsu/>, свободный
40. Петрушин, В.А. Экспертно-обучающие системы [Текст] / В.А. Петрушин. – К.: Наук. Думка, 1992. – 196 с.
41. Попов, Д. И. Способ оценки знаний в дистанционном обучении на основе нечетких отношений [Текст] / Д. И. Попов // Дистанционное образование. – 2000. – № 6. – С. 26-28.
42. Потеев, М.И. Практикум по методике обучения во втузах [Текст]: учебное пособие / М.И. Потеев. – Москва: Высшая школа, 1990. – 94с.
43. Прокофьева, Н.О. Стратегии и методы проведения контроля знаний в компьютерном обучении [Электронный ресурс]/ Н.О. Прокофьева// Educational Technology & Society – 2010 – V.13(1). – P. 378-392 – Режим доступа: http://ifets.ieee.org/russian/depository/v13_i1/html/11.htm, свободный

44. Разыграева, В.А. Автоматизация процесса адаптивного электронного обучения [Текст]/В.А. Разыграева, А.В. Лямин// Труды XVIII Всероссийской научно-методической конференции «Телематика'2011» г.Санкт-Петербург, 2011 г. – С Пб: СПбГУ ИТМО, 2010. – Т.1. – С. 168-169.

45. Рыбина, Г.В. Основы построения интеллектуальных систем: учебное пособие [Текст]/ Г.В. Рыбина. – Москва: Финансы и статистика, 2010. – 432с.

46. Титенко, С.В. Автоматизация построения тестовых заданий в системах дистанционного обучения на основе понятийно-тезисной модели [Электронный ресурс]/С.В. Титенко//Образовательные технологии и общество. – 2013. – 16(1). – С. 463-481

47. Толстобров, А. П. Возможности анализа и повышения качества тестовых заданий при использовании сетевой системы управления обучения[Текст]/ А.П. Толстобров, И.А. Коржик // Вестник ВГУ, 2008. - № 2. – С. 100-106

48. Штовба, С. Д. Проектирование нечетких систем средствами MATLAB[Текст]/С.Д. Штовба//М.: Горячая линия – Телеком, 2007. – 288 с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

module_complete_test.class.php

```
<?php
/**
 * Complete test class
 *
 * This class implements the Complete test eFront module
 * @version 0.1
 */
class module_complete_test extends EfrontModule
{
    public function getName() {
        return "Correct test";
    }
    public function getPermittedRoles() {
        return array("professor");
    }
    public function onInstall() {
        return true;
    }
    public function onUninstall() {
        return true;
    }
    public function getModule() {
        return true;
    }
    public function isLessonModule() {
        return true;
    }
    public function getLessonSmartyTpl() {
        return $this -> getControlPanelSmartyTpl();
    }
}

public function getSmartyTpl() {
    $smarty = $this -> getSmartyVar();
    $smarty -> assign("T_CURRENT_TEST_MODULE_BASEURL", $this ->
moduleBaseUrl);
    $currentUser = $this -> getCurrentUser();
    if ($currentLesson = $this -> getCurrentLesson()) {
        $currentContent = new EfrontContentTree($currentLesson);
        if (isset($_GET['test']) && isset($_GET['login'])) {
            $currentUnit = new EfrontUnit($_GET['test']);
            $user = EfrontUserFactory::factory($_GET['login']);
            $test = new EfrontTest($currentUnit['id'], true);
```

```

        $status = $test -> getStatus($user);
        $form = new HTML_QuickForm("test_form", "post", $this ->
moduleBaseUrl.'&login='.$_GET['login'].'&test='.$_GET['test'], "", null, true);
        switch ($status['status']) {
            case 'incomplete':
                if (!$testInstance =
unserialize($status['completedTest']['test'])) {
                    throw new
EfrontTestException(_TESTCORRUPTEDASKRESETEXECUTION,
EfrontTestException::CORRUPTED_TEST);
                }
                if ($testInstance -> time['pause'] &&
isset($_GET['resume'])) {
                    $testInstance -> time['pause'] = 0;
                    $testInstance -> time['resume'] = time();
                    //unset($testInstance -> currentQuestion);
                    $testInstance -> save();
                }
                $remainingTime = $testInstance ->
options['duration'] - $testInstance -> time['spent'] - (time() - $testInstance -> time['resume']);
                $nocache = false;
                if ($form -> isSubmitted() || ($testInstance ->
options['duration'] && $remainingTime < 0) || $status['status'] == 'incomplete') {
                    $nocache = true;
                }
                $testInstance -> options['pause_test'] = 0;
                $testInstance -> options['onebyone'] = 0;
                $testInstance -> options['duration'] = 0;

                $testString = $testInstance ->
toHTMLQuickForm($form, false, false, false, $nocache);
                $testString = $testInstance -> toHTML($testString,
$remainingTime);

                break;
            case 'completed':case 'passed':case 'failed':case 'pending':
                if (!$testInstance =
unserialize($status['completedTest']['test'])) {
                    throw new
EfrontTestException(_TESTCORRUPTEDASKRESETEXECUTION,
EfrontTestException::CORRUPTED_TEST);
                }
                //$url =
basename($_SERVER['PHP_SELF']).'?ctg=content&view_unit='.$_GET['view_unit'];
                //($testInstance -> options['redoable'] = 1);
                $testString = $testInstance ->
toHTMLQuickForm($form, false, true);
                $testString = $testInstance ->
toHTMLSolved($testString);

```

```

flash-chart.php';
$testInstance -> analyseTest();
$analysisCode);

user['login']);
$status);

moduleBaseUrl.'&login='.$user ->
user['login'].'&test='.$currentUnit['id'].'&test_analysis=1&selected_unit='.$_GET['selected_unit']
.'&show_chart=1&show_solved_test='.$_GET['show_solved_test'];
displayChart($url);

calculateChart($parentScores);

        if (isset($_GET['test_analysis'])) {
            require_once 'charts/php-ofc-library/open-
list($parentScores, $analysisCode) =
            $smarty -> assign("T_CONTENT_ANALYSIS",
            $smarty -> assign("T_TEST_DATA", $testInstance);
            $status = $testInstance -> getStatus($user ->
            $smarty -> assign("T_TEST_STATUS",
            if (isset($_GET['display_chart'])) {
                $url = $this ->
                echo $testInstance ->
                exit;
            } elseif (isset($_GET['show_chart'])) {
                echo $testInstance ->
                exit;
            }
        }
        break;
    default:
        if (isset($_GET['confirm'])) {
            $testInstance = $test -> start($user ->
            eF_redirect('".$_GET['confirm']).$this ->
            moduleBaseUrl.'&login='.$_GET['login'].'&test='.$_GET['test'];
            exit;
        } else {
            $testInstance = $test;
            $test -> getQuestions();
            //This way the test's questions are populated, and we will be needing this information
            $testInstance -> options['random_pool'] &&
            $testInstance -> options['random_pool'] >= sizeof($testIn) ? $questionsNumber = $testInstance
            -> options['random_pool'] : $questionsNumber = sizeof($testInstance -> questions);
        }
        break;
    }
    if (isset($_GET['ajax'])) {
        $testInstance -> handleAjaxActions();
    }
}

```

```

        //Calculate total questions. If it's already set, then we are visiting
an unsolved test, and the questions number is already calculated (and may be different that the
$testInstance -> questions size)
        if (!isset($questionsNumber)) {
            $questionsNumber = sizeof($testInstance -> questions);
        }
        //$smarty -> assign("T_REMAINING_TIME", $remainingTime);
        $smarty -> assign("T_TEST_QUESTIONS_NUM",
$questionsNumber);

        $smarty -> assign("T_TEST_DATA", $testInstance);
        $smarty -> assign("T_TEST", $testString);
        $smarty -> assign("T_TEST_STATUS", $status);

        if (!$status['status'] || ($status['status'] == 'incomplete' &&
$testInstance -> time['pause'])) { //If the user hasn't confirmed he wants to do the test,
display confirmation buttons
            $smarty -> assign("T_SHOW_CONFIRMATION", true);
        } else { //The user
confirmed he wants to do the test, so display it

            $form -> addElement('hidden', 'time_start', $timeStart);
//This element holds the time the test started, so we know the remaining time even if the user
left the system

            $form -> addElement('submit', 'submit_test',
SUBMITTEST, 'class = "flatButton" onclick = "if (typeof(checkedQuestions) != \'undefined\' &&
(unfinished = checkQuestions())) return
confirm(\'. _YOUHAVENOTCOMPLETEDTHEFOLLOWINGQUESTIONS.\': \'+unfinished+\'.
.\_AREYOUSUREYOUWANTTOSUBMITTEST.\');");

            if ($form -> isSubmitted() && $form -> validate()) {
                $values = $form -> exportValues();
                $submitValues = $form -> getSubmitValues();
                foreach($testInstance -> questions as $id =>
$question) {
                    $submitValues['question_time'][$id] ||
$submitValues['question_time'][$id] === 0 ? $question -> time =
$submitValues['question_time'][$id] : null;
                }
                //Set the unit as "seen"
                $testInstance -> complete($values['question']);
                $completedLesson = $user ->
setSeenUnit($currentUnit, $currentLesson, 1);
                eF_redirect('".$this ->
moduleBaseUrl.'&login='.$_GET['login'].'&test='.$_GET['test']);
                exit;
            }
            $renderer = new
HTML_QuickForm_Renderer_ArraySmarty($smarty);
            $form -> accept($renderer);

```

```

        $smarty -> assign('T_TEST_FORM', $renderer -> toArray());
    }
} else {
    $form = $this -> buildImportCsvForm();
    list($selectedTest, $uploadedFile) = $this ->
handleImportCsvForm($form);
    $renderer = prepareFormRenderer($form);
    $smarty -> assign('T_UPLOAD_FORM', $renderer -> toArray());
    $form = $this -> buildCorrelateDataForm($selectedTest,
$uploadedFile);
    list($errorDuringImport, $numImported) = $this ->
handleCorrelateDataForm($form);
    $renderer = prepareFormRenderer($form);
    $smarty -> assign('T_CORRELATE_FORM', $renderer -> toArray());
    if (isset($_GET['ajax']) && $_GET['ajax'] == 'usersTable') {
        $lessonUsers = $currentLesson -> getUsers('student');
//Get all users that have this lesson
        $testsIterator = new EfrontTestsFilterIterator(new
EfrontVisitableFilterIterator(new EfrontNodeFilterIterator(new RecursiveIteratorIterator(new
RecursiveArrayIterator($currentContent -> tree), RecursiveIteratorIterator :: SELF_FIRST)));
        foreach ($testsIterator as $key => $value) {
            $tests[$key] = $value['name'];
        }
        $select_units = & HTML_QuickForm ::
createElement('select', 'tests');
        $select_units -> loadArray($tests);
        $smarty -> assign("T_TESTS_SELECT", $select_units ->
toHTML());

        isset($_GET['limit']) && eF_checkParameter($_GET['limit'],
'uint') ? $limit = $_GET['limit'] : $limit = G_DEFAULT_TABLE_SIZE;

        if (isset($_GET['sort']) &&
eF_checkParameter($_GET['sort'], 'text')) {
            $sort = $_GET['sort'];
            isset($_GET['order']) && $_GET['order'] == 'desc' ?
$order = 'desc': $order = 'asc';
        } else {
            $sort = 'login';
        }
        $lessonUsers = eF_multiSort($lessonUsers, $sort, $order);
        $smarty -> assign("T_USERS_SIZE", sizeof($lessonUsers));
        if (isset($_GET['filter'])) {
            $lessonUsers = eF_filterData($lessonUsers,
$_GET['filter']);
        }
        if (isset($_GET['limit']) &&
eF_checkParameter($_GET['limit'], 'int')) {

```



```

        $testsIterator = new EfrontTestsFilterIterator(new
EfrontVisitableFilterIterator(new EfrontNodeFilterIterator(new RecursivelteratorIterator(new
RecursiveArrayIterator($currentContent -> tree), RecursivelteratorIterator :: SELF_FIRST)));
        foreach ($testsIterator as $key => $value) {
            $tests[$key] = $value['name'];
        }
        $form = new HTML_QuickForm("upload_form", "post", $this ->
moduleBaseUrl.'&tab=import', "", null, true);
        $form -> addElement('file', 'upload_file', _UPLOADFILE);
        $form -> addElement('select', 'select_test', _TEST, $tests);
        $form -> addElement('submit', 'submit', _SUBMIT, 'class = "flatButton");

        return $form;
    }

    private function handleImportCsvForm(&$form) {
        $currentUser = $this -> getCurrentUser();
        $smarty      = $this -> getSmartyVar();
        if ($form -> isSubmitted() && $form -> validate()) {
            $values = $form -> exportValues();
            if (!is_dir($currentUser -> user['directory']."/temp")) {
                mkdir($currentUser -> user['directory']."/temp", 0755);
            }
            $filesystem = new FileSystemTree($currentUser ->
user['directory']."/temp");
            $uploadedFile = $filesystem -> uploadFile('upload_file');

            if (($handle = fopen($uploadedFile['path'], "r")) !== FALSE) {
                while (($data = fgets($handle, 1000, "")) !== FALSE) {
                    $parsedContents[] = $data;
                }
                fclose($handle);
            }

            $selectedTest = new EfrontTest($values['select_test'], true);
            $smarty -> assign("T_TEST_QUESTIONS", $selectedTest ->
getQuestions());
            $smarty -> assign("T_COMPLETED_TEST_PARSED_CONTENTS",
array_slice($parsedContents, 0, 10));
        }
        return array($selectedTest, $uploadedFile);
    }

    private function buildCorrelateDataForm($selectedTest, $uploadedFile) {
        if (isset($selectedTest)) {
            $selectedTestId = $selectedTest -> test['id'];
        } elseif ($_GET['test_id']) {
            $selectedTest = new EfrontTest($_GET['test_id']);
        }
        $form = new HTML_QuickForm("correlate_form", "post", $this ->
moduleBaseUrl.'&tab=import&test_id='.$selectedTestId, "", null, true);

```

```

        $form -> addElement('hidden', 'uploaded_file');
        $form -> addElement('hidden', 'test_id');
        $form -> addElement('hidden', 'start_data_row', "", 'id = "start_data_row"');
        $form -> addElement('hidden', 'date_source_hidden', "", 'id =
"date_source_hidden"');
        $form -> addElement('hidden', 'user_source_hidden', "", 'id =
"user_source_hidden"');
        $form -> addElement('hidden', 'score_source_hidden', "", 'id =
"score_source_hidden"');
        if (isset($selectedTest)) {
            $form -> setDefaults(array('test_id' => $selectedTest -> test['id'],
'uploaded_file' => $uploadedFile['path']));
            foreach($selectedTest -> getQuestions() as $key => $value) {
                $form -> addElement('hidden', $key.'_answer_source_hidden', "",
'id = "'. $key.'_answer_source_hidden"');
                $form -> addElement('hidden', $key.'_score_source_hidden', "", 'id
= "'. $key.'_score_source_hidden"');
            }
        }
        $form -> addElement('advcheckbox', 'complete_course',
_COMPLETE_TEST_COMPLETECOURSEWITHLESSON, array(0,1));
        $form -> addElement('submit', 'submit', _SUBMIT, 'class = "flatButton"');
        return $form;
    }

private function handleCorrelateDataForm(&$form) {

    if ($form -> isSubmitted() && $form -> validate()) {
        $currentLesson = $this -> getCurrentLesson();
        $lessonUsers = $currentLesson -> getUsers('student');
        $formValues = $form -> exportValues();
        $selectedTest = new EfrontTest($formValues['test_id']);
        $uploadedFile = new EfrontFile($formValues['uploaded_file']);
        if (($handle = fopen($uploadedFile['path'], "r")) !== FALSE) {
            while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
                $parsedContents[] = $data;
            }
            fclose($handle);
        }
        $parsedContents = array_slice($parsedContents,
$formValues['start_data_row']);

        $dateColumn = $formValues['date_source_hidden'];
        $userColumn = $formValues['user_source_hidden'];
        $scoreColumn = $formValues['score_source_hidden'];
        foreach ($parsedContents as $key => $value) {
            if ($value[$userColumn] && $value[$dateColumn]) {
                //Meaning a valid row
            }
        }
    }
}

```

```

        $testDates[] = $value[$dateColumn];
        $testUsers[] = $value[$userColumn];
        $testScores[] = $value[$scoreColumn];
    } else {
        unset($parsedContents[$key]);
    }
}
$timestamps = $this -> analyzeContentsToFindDateFormat($testDates);
$userLogins = $this -> analyzeContentsToFindUserFormat($testUsers,
    $allUsers);

//pr($lessonUsers);
$numImported = 0;
foreach ($testUsers as $key => $user) {
    $login = $userLogins[$key];
    if (!$login) {
        $errorDuringImport[$user] = "The user "$user"
does not exist in the system";
    } elseif (in_array($login, array_keys($lessonUsers))) {
        $errorDuringImport[$user] = "The user "$user"
is not enrolled to this lesson";
    } else {
        try {
            $userAnswers = array();
            foreach ($selectedTest -> getQuestions(true) as $id
=> $question) {
                $questionColumn =
                $formValues[$id.'_answer_source_hidden'];
                $parsedAnswer =
                $parsedContents[$key][$questionColumn];
                $userAnswers[$id] = $this ->
                translateParsedAnswersToUserAnswers($parsedAnswer, $question);
                $questionScoreColumn =
                $formValues[$id.'_score_source_hidden'];
                $parsedQuestionScore =
                $parsedContents[$key][$questionScoreColumn];
            }
            $completedTest = $selectedTest -> start($login);
            $completedTest -> complete($userAnswers,
            $results);
            $completedTest -> time['start'] =
            $timestamps[$key]; //The time that this test has started
            $completedTest -> time['end'] =
            $timestamps[$key]+1; //The time that this test ends
            $completedTest -> time['spent'] = 1; //The
            time that this test ends
            $completedTest -> completedTest['status'] =
            'passed'; //The time that this test ends

```

```

                                $completedTest -> completedTest['score'] =
(float)$parsedContents[$key][$scoreColumn];

                                foreach ($completedTest -> getQuestions(true) as
$question) {
                                $questionScoreColumn =
$formValues[$question.' _score_source_hidden'];
                                $parsedQuestionScore =
$formValues[$questionScoreColumn];
                                $question -> score = $parsedQuestionScore
<= 1 ? $parsedQuestionScore*100 : $parsedQuestionScore;
                                }
                                $completedTest -> save();
                                $currentUser = EfrontUserFactory::factory($login);
                                $completedLesson = $currentUser ->
setSeenUnit($selectedTest -> test['content_ID'], $_SESSION['s_lessons_ID'], 1);
                                $result = eF_getTableData("users_to_lessons",
"completed", "users_LOGIN=".$currentUser -> user['login']. " and
lessons_ID=".$_SESSION['s_lessons_ID']);
                                if ($result[0]['completed'] &&
$formValues['complete_course']) {
                                $lessonCourses = $currentUser ->
getCourses(true);
                                //Get the courses that this lesson is part of. This way, we can auto complete a course, if
it should be auto completed

                                //Filter out courses that the student doesn't
have
                                $result =
eF_getTableDataFlat("users_to_courses", "courses_ID", "users_LOGIN=".$currentUser ->
user['login']. " ");
                                $userCourses = $result['courses_ID'];
                                foreach ($lessonCourses as $id => $course) {
                                    if (!in_array($id, $userCourses)) {
                                        unset($lessonCourses[$id]);
                                    } else {
                                        $currentUser ->
completeCourse($course -> course['id'], 100, _AUTOCOMPLETEDCOURSE);
                                    }
                                }
                                }
                                $numImported++;
                                } catch (Exception $e) {
                                    $errorDuringImport[$user] = "Error during
importing values for &quot;$user&quot;:" . $e -> getMessage();
                                }
                            }
                    }
}

```

```

        if (sizeof($errorDuringImport) > 0) {
            $this -> setMessageVar("Successfully imported $numImported
results, but ".sizeof($errorDuringImport)." users could not be imported:<br> ".implode("<br>",
$errorDuringImport), "failure");
        } else {
            $this -> setMessageVar("Successfully imported $numImported
results", "success");
        }
        return array($errorDuringImport, $numImported);
    }
}

private function translateParsedAnswersToUserAnswers($parsedAnswer, $question) {
    $userAnswers = array_fill(0, sizeof($question -> options), 0);
    $parsedAnswer = explode("/", $parsedAnswer);
    foreach ($parsedAnswer as $key => $value) {

        if (!is_numeric($parsedAnswer)) {
            $value = chr(ord(strtoupper($value)) - 16);
        }

        if ($value > 0 && $value <= sizeof($userAnswers)) {
            $userAnswers[$value - 1] = 1;
        }
    }
    return $userAnswers;
}

private function buildMappingsForm($selectedTest, $uploadedFile) {
    // $dateFormat
}

/*
private function importCSVContents($parsedContents) {
    $lessonUsers = $currentLesson -> getUsers('student'); //Get all users
that have this lesson
    foreach ($parsedContents as $value) {
        $login = "";
        $userNameInCSV = $value[$userColumn];

        if ($userFormat == 'login') {
            $userLogin = $userNameInCSV;
        } elseif ($userFormat == 'surname, name') {
            $userLogin = array_search($userNameInCSV, $surnameName);
        } elseif ($userFormat == 'name, surname') {
            $userLogin = array_search($userNameInCSV, $nameSurname);
        }

        if ($userLogin && in_array($userLogin, array_keys($lessonUsers))) {
            $login = $userLogin;
        } else if ($userLogin) {

```

```

        $existingUsersButNotInLesson[] = $userLogin;
    } else {
        $notFoundUsers[] = $userNameInCSV;
    }

}
pr($existingUsersButNotInLesson);
pr($notFoundUsers);
exit;
pr($testDates);
pr($parsedContents);
pr($values);exit;
}
*/
private function analyzeContentsToFindUserFormat($testUsers, $allUsers) {
    $allUsers = eF_getTableData("users", "login, name, surname");
    foreach ($allUsers as $value) {
        $surnameName[$value['login']] = $value['surname'].'.'.$value['name'];
        $nameSurname[$value['login']] = $value['name'].'.'.$value['surname'];
        $logins[$value['login']] = $value['login'];
    }
    $count = 0;
    $userFormat = false;
    while (!$userFormat && $testUsers[$count]) {
        $value = $testUsers[$count++];
        if (in_array($value, $logins)) {
            $userFormat = 'login';
            $sourceArray = $logins;
        } elseif (in_array($value, $surnameName)) {
            $userFormat = 'surname, name';
            $sourceArray = $surnameName;
        } elseif (in_array($value, $nameSurname)) {
            $userFormat = 'name, surname';
            $sourceArray = $nameSurname;
        } else {
            $userFormat = false;
        }
    }
    foreach($testUsers as $key => $value) {
        $testUsers[$key] = array_search($value, $sourceArray);
    }
    return $testUsers;

    //return $userFormat;
}
private function analyzeContentsToFindScoreFormat($testScores) {
    pr($testScores);
}

```

```

private function analyzeContentsToFindDateFormat($testDates) {
    $count = 0;
    $dateFormat = false;
    while (!$dateFormat && $testDates[$count]) {
        $value = $testDates[$count++];
        $parts = explode("/", $value);
        if (sizeof($parts) == 1) {
            $dateFormat = 'timestamp';
        } elseif ($parts[0] > 1000) {
            $dateFormat = 'YYYY/MM/DD';
        } elseif ($parts[0] > 12) {
            $dateFormat = 'DD/MM/YYYY';
        } elseif ($parts[1] > 12) {
            $dateFormat = 'MM/DD/YYYY';
        }
    }
    foreach ($testDates as $key => $value) {
        if ($dateFormat != 'timestamp') {
            $parts = explode("/", $value);
            switch ($dateFormat) {
                case 'YYYY/MM/DD': $testDates[$key] =
mktime(0,0,0,$parts[1], $parts[2], $parts[0]); break;
                case 'DD/MM/YYYY': $testDates[$key] =
mktime(0,0,0,$parts[1], $parts[0], $parts[2]); break;
                case 'MM/DD/YYYY': $testDates[$key] =
mktime(0,0,0,$parts[0], $parts[1], $parts[2]); break;
                default: break;
            }
        }
    }
    return $testDates;
    //return $dateFormat;
}
private function importCsvFile() {
}
private function parseCSVContents() {
}
}
?>

```

module_complete_test.tpl

```

{* smarty template for complete_test *}
{if $smarty.get.login && $smarty.get.test}
    {capture name = 't_test_code'}
        {if $T_SHOW_CONFIRMATION}
            {assign var =
't_show_side_menu' value = true}

```

```

= "sideContentTree"}
{capture name
    {eF_template_printSide title = $smarty.const._LESSONMATERIAL data =
    $T_CONTENT_TREE id = 'current_content'}
{/capture}
<table class =
"navigationHandles">
<tr><td>{eF_template_printPreviousNext previous = $T_PREVIOUS_UNIT next =
    $T_NEXT_UNIT}</td></tr>
</table>
<table class =
"testHeader">
<tr><td
id = "testName">{$T_TEST_DATA->test.name}</td></tr>
<tr><td
id = "testDescription">{$T_TEST_DATA->test.description}</td></tr>
<tr><td>
<table class = "testInfo">
<tr><td rowspan = "6" id = "testInfolmage"><img src =
    {$T_COMPLETETEST_BASELINK|cat:"images/tests.png"} alt = "{$T_TEST_DATA-
    >test.name}" title = "{$T_TEST_DATA->test.name}"/></td>
<td id = "testInfoLabels"></td>
<td></td></tr>
<tr><td>{$smarty.const._TESTDURATION};&nbsp;</td>
<td>
    {if $T_TEST_DATA->options.duration}
        {if $T_TEST_DATA-
        >convertedDuration.hours}{$T_TEST_DATA->convertedDuration.hours}
        {$smarty.const._HOURS}&nbsp;&nbsp;&nbsp;{/if}
        {if $T_TEST_DATA-
        >convertedDuration.minutes}{$T_TEST_DATA->convertedDuration.minutes}
        {$smarty.const._MINUTES}&nbsp;&nbsp;&nbsp;{/if}
        {if $T_TEST_DATA-
        >convertedDuration.seconds}{$T_TEST_DATA->convertedDuration.seconds}
        {$smarty.const._SECONDS}{/if}
    {else}
        {$smarty.const._UNLIMITED}

```

```

        {/if}
    </td></tr>

    <tr><td>{$smarty.const._NUMOFQUESTIONS}&nbsp;</td>

        <td>{$T_TEST_QUESTIONS_NUM}</td></tr>

    <tr><td>{$smarty.const._QUESTIONSARESHOWN}&nbsp;</td>

        <td>{if $T_TEST_DATA-
>options.onebyone}{$smarty.const._ONEBYONEQUESTIONS}{else}{$smarty.const._ALLT
OGETHER}{/if}</td></tr>

        {if $T_TEST_STATUS.status == 'incomplete' && $T_TEST_DATA->time.pause}

            <tr><td>{$smarty.const._YOUPAUSEDTHISTESTON}&nbsp;</td>

                <td>#filter:timestamp_time-{$T_TEST_DATA-
>time.pause}#</td></tr>

            {else}

                <tr><td>{$smarty.const._DONETIMESSOFAR}&nbsp;</td>

                    <td>{if
$T_TEST_STATUS.timesDone}{$T_TEST_STATUS.timesDone}{else}0{/if}&nbsp;<td>{$smarty.c
onst._TIMES}</td></tr>

                <tr><td>{if $T_TEST_STATUS.timesLeft != false
}{$smarty.const._YOUCANDOTHETEST}&nbsp;</td>

                    <td>{$T_TEST_STATUS.timesLeft}&nbsp;<td>{$smarty.const._TIMESMORE}{/if}</td><
/tr>

                {/if}

            </table>

        </td>

    <tr><td
id = "testProceed">

        {if
$T_TEST_STATUS.status == 'incomplete' && $T_TEST_DATA->time.pause}

            <input class = "flatButton" type = "button" name = "submit_sure" value =

```

```

"{$smarty.const._RESUMETEST}&nbsp;&raquo;" onclick =
"javascript:location=location+'&resume=1'" />
{else}

<input class = "flatButton" type = "button" name = "submit_sure" value =
"{$smarty.const._PROCEEDTOTEST}&nbsp;&raquo;" onclick =
"javascript:location=location+'&confirm=1'" />
{/if}

</td></tr>
</table>
{elseif $smarty.get.test_analysis}
{assign var = 'title'
value = "`$title`&nbsp;&raquo;&nbsp;&lt;a class = 'titleLink' href =
`$smarty.server.PHP_SELF`?ctg=content&view_unit=`$smarty.get.view_unit`&test_anal
ysis=1`>`$smarty.const._TESTANALYSISFORTEST` &quot;`$T_TEST_DATA-
>test.name`&quot;</a>"}
{capture name =
"t_test_analysis_code"}
<div class =
"headerTools">
<span>
<img src = {$T_COMPLETETEST_BASELINK|cat:"images/arrow_left.png"} alt =
"{$smarty.const._VIEWSOLVEDTEST}" title = "{$smarty.const._VIEWSOLVEDTEST}">
<a href =
"{$smarty.server.PHP_SELF}?ctg=tests&view_unit={$smarty.get.view_unit}&show_solve
d_test={$smarty.get.show_solved_test}">{$smarty.const._VIEWSOLVEDTEST}</a>
</span>
{if
$T_TEST_STATUS.testIds|@sizeof > 1}
<span>
<img src = {$T_COMPLETETEST_BASELINK|cat:"images/go_into.png"} alt =
"{$smarty.const._JUMPTOEXECUTION}" title = "{$smarty.const._JUMPTOEXECUTION}">
&nbsp;{$smarty.const._JUMPTOEXECUTION}
<select style = "vertical-align:middle" onchange =
"location.toString().match(/show_solved_test/) ? location =
location.toString().replace(/show_solved_test=\d+/,
'show_solved_test='+this.options[this.selectedIndex].value) : location = location +
'&show_solved_test='+this.options[this.selectedIndex].value">
{if $smarty.get.show_solved_test}{assign var = "selected_test" value =
$smarty.get.show_solved_test}{else}{assign var = "selected_test" value =
$T_TEST_STATUS.lastTest}{/if}

```

```

        {foreach name = "test_analysis_list" item = "item" key = "key" from =
$T_TEST_STATUS.testIds}

            <option value = "{$item}" {if $selected_test ==
$item}selected{/if}>#{Smarty.foreach.test_analysis_list.iteration} -
#filter:timestamp_time-{$T_TEST_STATUS.timestamps[$key]}#</option>

        {/foreach}

    </select>

                                                    </span>
                                                    {/if}
        </div>
        <table style =
"width:100%">
                                                    <tr><td
style = "vertical-align:top">{$T_CONTENT_ANALYSIS}</td></tr>
                                                    <tr><td
style = "vertical-align:top"><iframe width = "750px" id = "analysis_frame" height =
"550px" frameborder = "no" src =
 "{$T_CURRENT_TEST_MODULE_BASEURL}&login={Smarty.get.login}&test={Smarty.ge
t.test}&display_chart=1&selected_unit={Smarty.get.selected_unit}&test_analysis=1&s
how_solved_test={Smarty.get.show_solved_test}"></iframe></td></tr>
                                                    </table>
        {/capture}

        {eF_template_printBlock title = ""Smarty.const._TESTANALYSIS`
`Smarty.const._FORTEST` &quot;`$T_TEST_DATA->test.name`&quot;
`Smarty.const._ANDUSER` &quot;`$T_TEST_DATA->completedTest.login`&quot;;" data
= Smarty.capture.t_test_analysis_code image=
$T_COMPLETETEST_BASELINK|cat:'images/tests.png' absoluteImagePath = 1}
            {else}
                {if $T_TEST_STATUS.status == " | |
$T_TEST_STATUS.status == 'incomplete'}
                    {capture name =
"test_footer"}
                        <table class =
"formElements" style = "width:100%">
                            <tr><td colspan
= "2">&nbsp;</td></tr>
                            <tr><td colspan
= "2" class = "submitCell" style = "text-
align:center">{$T_TEST_FORM.submit_test.html}&nbsp;{$T_TEST_FORM.pause_test.ht
ml}</td></tr>
                        </table>
                    {/capture}
                {else}

```

```

't_show_side_menu' value = true}

"sideContentTree"}

        {eF_template_printSide title=$smarty.const._LESSONMATERIAL data =
$T_CONTENT_TREE}

        {/capture}
        <table class =

"navigationHandles">

        <tr><td>{eF_template_printPreviousNext previous = $T_PREVIOUS_UNIT next =
$T_NEXT_UNIT}</td></tr>

        </table>
        {/if}
        {if !$T_NO_TEST}

        {$T_TEST_FORM.javascript}

        <form
{$T_TEST_FORM.attributes}>

        {$T_TEST_FORM.hidden}

        {$T_TEST}

        {$smarty.capture.test_footer}

        </form>
        {/if}
        {/if}
        {/capture}
        {eF_template_printBlock title =
"$smarty.const._COMPLETE_TEST_SETVALUES" data = $smarty.capture.t_test_code
image = $T_COMPLETETEST_BASELINK|cat:'images/theory.png' absoluteImagePath = 1
main_options = $T_TABLE_OPTIONS}
{else}
        {capture name = 't_users_code'}
        {capture name = 't_upload_form_code'}
        {$T_UPLOAD_FORM.javascript}
        <style>
        {literal}
        td.ct_hoverCell{background:#C7D7F3;}
        {/literal}
        </style>
        <script>
        var dataSources = new Array('row', 'date', 'user', 'score');
        var dataSourcesColors = new Array('blue', 'red', 'green', 'orange');
        {foreach name = 'test_questions_loop' item = "question" key = "id" from =
$T_TEST_QUESTIONS}
                dataSources.push('{ $id }_answer');

```

```

        dataSourcesColors.push('#'+Math.round(Math.random()*16777215).toString(16)
    );
        dataSources.push('{${id}_score');

    dataSourcesColors.push('#'+Math.round(Math.random()*16777215).toString(16)
    );
    {/foreach}

    {literal}
    var currentDataSource = 0;
    var hasSelectedRow = false;
    function selectRow(idx) {
        if (!hasSelectedRow) {
            hasSelectedRow = true;
            $('import_results_table').select('tr').each(function (s) {
                if (s.hasClassName('oddRowColor')) {

s.removeClassName('oddRowColor').addClassName('oddRowColorNoHover');
                } else if (s.hasClassName('evenRowColor')) {

s.removeClassName('evenRowColor').addClassName('evenRowColorNoHover');
                }
            });
            dataSourceColor = dataSourcesColors[currentDataSource];

            $('start_row_result').update('&nbsp;#'+idx).setStyle({color:dataSourceColor});
            $('start_data_row').value = idx;
            setRowAsStartData(idx, dataSourceColor);
            currentDataSource++;
            $(dataSources[currentDataSource]+'_source').show();
        }
    }
    function setRowAsStartData(idx, color) {

        $('import_results_table').select('tr.defaultRowHeight').each(function (s, i) {if
(i==idx) {s.setStyle({color:color});}});
    }
    function highlightColumn(idx) {
        $('import_results_table').select('tr').each(function (s) {if
(s.hasClassName('oddRowColorNoHover') || s.hasClassName('evenRowColorNoHover'))
{s.select('td').each(function (p, i) {if (i == idx) {p.addClassName('ct_hoverCell');}});}});
    }
    function unHighlightColumn(idx) {
        $('import_results_table').select('tr').each(function (s) {if
(s.hasClassName('oddRowColorNoHover') || s.hasClassName('evenRowColorNoHover'))
{s.select('td').each(function (p, i) {if (i == idx) {p.removeClassName('ct_hoverCell');}});}});
    }
    function setColumnAsDataSource(idx, color) {

```

```

        $('import_results_table').select('tr').each(function (s) {if
(s.hasClassName('oddRowColorNoHover') || s.hasClassName('evenRowColorNoHover'))
{s.select('td').each(function (p, i) {if (i == idx) {p.setStyle({color:color});});});});
    }
    function unsetColumnAsDataSource(idx, color) {
        $('import_results_table').select('tr').each(function (s) {if
(s.hasClassName('oddRowColorNoHover') || s.hasClassName('evenRowColorNoHover'))
{s.select('td').each(function (p, i) {if (i == idx) {p.setStyle({color:"});});});});
    }
    function setDataSource(idx) {
        dataSource          = dataSources[currentDataSource];
        if (dataSource) {
            dataSourceColor = dataSourcesColors[currentDataSource];

$(dataSource+'_source_result').update(':&nbsp;#+idx').setStyle({color:dataSource
eColor});

            $(dataSource+'_source_hidden').value = idx;
            setColumnAsDataSource(idx, dataSourceColor);
            currentDataSource++;
            if (dataSources[currentDataSource]) {

$(dataSources[currentDataSource]+'_source').show();
                } else {
                    $('continue_button').show();
                }
            }
        }
    }
    function unsetDataSource(idx) {

}
{/literal}
</script>
{if 0}

{elseif $T_COMPLETED_TEST_PARSED_CONTENTS}
<div class = "mediumHeader" style = "margin-bottom:30px;">Step
2: Correlate columns to data</div>
{$T_CORRELATE_FORM.javascript}
<form {$T_CORRELATE_FORM.attributes}>
{$T_CORRELATE_FORM.hidden}
    <table id = "import_results_table" style = "width:100%;">
        <tr><td colspan = "100%">
            <div id = "start_row_source" >Click
on the row where the data starts<span id = "start_row_result">: </span></div>
            <div id = "date_source" style =
"display:none">Click on the column that will serve as <b>Date</b><span id =
"date_source_result">: </span></div>

```

```

        <div id = "user_source" style =
"display:none">Click on the column that will serve as <b>User name</b><span id =
"user_source_result">: </span></div>
        <div id = "score_source" style =
"display:none">Click on the column that will serve as <b>Test score</b><span id =
"score_source_result">: </span></div>
        {foreach name = 'test_questions_loop' item
= "question" key = "id" from = $T_TEST_QUESTIONS}
        <div id = "{$id}_answer_source" style
= "display:none">Click on the column that will serve as answer for question
<b>{$question.text|@strip_tags|eF_truncate:30}</b><span id =
 "{$id}_answer_source_result">: </span></div>
        <div id = "{$id}_score_source" style =
"display:none">Click on the column that will serve as score for question
<b>{$question.text|@strip_tags|eF_truncate:30}</b><span id =
 "{$id}_score_source_result">: </span></div>
        {/foreach}
        <div id = "continue_button" style =
"display:none">{$T_CORRELATE_FORM.complete_course.label}:
{$T_CORRELATE_FORM.complete_course.html}<br/>{$T_CORRELATE_FORM.submit.ht
ml}</div>
        </td></tr>
        <tr class = "topTitle"><td class = "topTitle
centerAlign" colspan = "100%">{$smarty.const._RESULTS}</td></tr>
        {foreach name = 'parsed_contents_loop' key = "key" item =
"row" from = $T_COMPLETED_TEST_PARSED_CONTENTS}
        <tr class = "defaultRowHeight {cycle values =
"oddRowColor, evenRowColor}" onclick = "selectRow('{ $key}');" >
        {foreach name = 'parsed_contents_inner_loop' key
= "key2" item = "cell" from = $row}
        <td style = "padding:2px 4px 2px 4px;"
onmouseover = "if (hasSelectedRow) highlightColumn('{ $key2}');" onmouseout = "if
(hasSelectedRow) unHighlightColumn('{ $key2}');" onclick = "if (hasSelectedRow)
setDataSource('{ $key2}');" >{$cell}</td>
        {/foreach}
        </tr>
        {/foreach}
        </table>
    </form>

    {else}
        <div class = "mediumHeader">Step 1: Import csv file and select
test</div>
        {$T_UPLOAD_FORM.javascript}
        <form {$T_UPLOAD_FORM.attributes}>
        {$T_UPLOAD_FORM.hidden}
        <table class = "formElements">
        <tr><td class =
"labelCell">{$T_UPLOAD_FORM.upload_file.label}</td>

```

```

                <td class =
"elementCell">{$T_UPLOAD_FORM.upload_file.html}</td></tr>
                <tr><td class =
"labelCell">{$T_UPLOAD_FORM.select_test.label}</td>
                <td class =
"elementCell">{$T_UPLOAD_FORM.select_test.html}</td></tr>
                <tr><td class = ""></td>
                <td class =
"submitCell">{$T_UPLOAD_FORM.submit.html}</td></tr>
            </table>
        </form>
    </if>
</capture>

<!--ajax:usersTable-->

        <table style = "width:100%" class = "sortedTable" size =
"{$T_USERS_SIZE}" sortBy = "0" id = "usersTable" useAjax = "1" rowsPerPage =
"{$smarty.const.G_DEFAULT_TABLE_SIZE}" url = "{$T_COMPLETETEST_BASEURL}&">
            <tr class = "topTitle">
                <td class = "topTitle" name =
"login">{$smarty.const._LOGIN}</td>
                <td class = "topTitle" name =
"name">{$smarty.const._NAME}</td>
                <td class = "topTitle" name =
"surname">{$smarty.const._SURNAME}</td>
                <td class = "topTitle">{$smarty.const._TEST}</td>
                <td class = "topTitle
centerAlign">{$smarty.const._COMPLETE_TEST_SETVALUES}</td>
            </tr>
            <foreach name = 'users_to_lessons_list' key = 'key' item = 'user' from =
{$T_ALL_USERS}>
                <tr class = "defaultRowHeight {cycle values = "oddRowColor,
evenRowColor"} {if !$user.active}deactivatedTableElement{/if}">
                    <td>{$user.login}</td>
                    <td>{$user.name}</td>
                    <td>{$user.surname}</td>
                    <td>{$T_TESTS_SELECT}</td>
                    <td class = "centerAlign"><a href = "javascript:void(0)"
onclick = "location =
('{$T_CURRENT_TEST_MODULE_BASEURL}&login={$user.login}&test='+Element.extend(
this).up().previous().down().options[this.up().previous().down().options.selectedIndex].v
alue)"><img src = {$T_COMPLETETEST_BASELINK|cat:"images/arrow_right.png"} alt =
"{$smarty.const._COMPLETE_TEST_SETVALUES}" title =
"{$smarty.const._COMPLETE_TEST_SETVALUES}"></a></td>
                </tr>
            </foreach>
        </table>

```

```

<!--/ajax:usersTable-->

{/capture}
{capture name = "t_complete_test_code"}
<div class = "tabber">
    {eF_template_printBlock tabber = "main" title =
"$smarty.const._LESSONUSERS`" data = $smarty.capture.t_users_code image =
$T_COMPLETETEST_BASELINK|cat:'images/theory.png' absoluteImagePath = 1
main_options = $T_TABLE_OPTIONS}
    {eF_template_printBlock tabber = "import" title =
"$smarty.const._COMPLETE_TEST_IMPORTFILE`" data =
$smarty.capture.t_upload_form_code image = "32x32/import.png"}
</div>
{/capture}
{eF_template_printBlock title = "$smarty.const._COMPLETE_TEST`" data =
$smarty.capture.t_complete_test_code image = "32x32/tests.png"}

{/if}

```

content.js

```

/**
 * Set unit seen status
 * This function makes an ajax call to the server to set the current unit's status to either 'seen'
or 'unseen', based on
 * the status parameter
 */
function setSeenUnit(status) {
  if (typeof(status) == 'undefined') { //If "status" parameter is not set, then toggle the seen
status based on whether the user has seen the lesson
    hasSeen ? status = 0 : status = 1;
  } else { //If "status" parameter is set, then toggle the seen status based on this
parameter
    status ? status = 1 : status = 0;
    hasSeen = !status;
  }
  //el = $('seenLink'); //For backwards-compatibility, we don't
specify el in the parameters list
  if ($('#seenLink')) {
    $('#seenLink').blur();
    el = $('#seenLink').down();
  } else {
    el = new Element('div');
  }

  parameters = {set_seen:status, method: 'get'};
  var url = window.location.toString();
  ajaxRequest(el, url, parameters, onSetSeenUnit);
}
/**
 * This function is executed when the ajax call of setSeen returns, to set the appropriate text
and
 * depict the unit status, both in the 'set seen' icon and the content tree as well
 */
function onSetSeenUnit(el, response) {
  try {

    results = response.evalJSON();
    if (unitType == 'scorm') {
      unitType = 'theory'; //scorm does not have icons of its own any
more
    } else if (unitType == 'scorm_test') {
      unitType = 'tests';
    }

    if (hasSeen) {
      if ($('#seenLink')) {

```

```

        setImageSrc($('seenLink').down(), 32, 'unit.png');
        $('seenLink').down().next().update(sawunit);
    }
    if ($('#tree_image_'+unitId)) {
        setImageSrc($('#tree_image_'+unitId), 16, unitType+'.png');
    }
} else {
    if ($('#seenLink')) {
        setImageSrc($('seenLink').down(), 32, 'unit_completed.png');
        $('seenLink').down().next().update(otsawunit);
    }
    if ($('#tree_image_'+unitId)) {
        setImageSrc($('#tree_image_'+unitId), 16, unitType+'_passed.png');
    }
}
if ($('#seenLink')) {
    new Effect.Appear($('seenLink'));
}

hasSeen = !hasSeen;
if ($('#progress_bar')) {
    $('#progress_bar').select('span.progressNumber')[0].update(parseFloat(results[0]) + '%');
    $('#progress_bar').select('span.progressBar')[0].setStyle({width:parseFloat(results[0]) +
'px'});
    if ($('#passed_conditions')) {
        $('#passed_conditions').update(parseInt(results[1]));
    }
    if ($('#lesson_passed')) {
        if (results[2] == true) {
            $('#lesson_passed').down().removeClassName('failure').addClassName('success');
        } else {
            $('#lesson_passed').down().removeClassName('success').addClassName('failure');
            //$('completed_block').hide();
        }
    }
}
if (results[2] == true) {
    if ($('#completed_block')) {
        $('#completed_block').show();
    }
    if (results[6] == true) {
        $('#show_certificate_block').show();
    }
    Effect.ScrollTo('completed_block');
}
}
} catch (e) {}
}

```

```

function nextLesson(el) {
    parameters = {ajax:'next_lesson', method: 'get'};
    var url = window.location.toString();
    ajaxRequest(el, url, parameters, onNextLesson);
}
function onNextLesson(el, response) {
    if (response.evalJSON(true)) {
        if (response.evalJSON(true).url) {
            window.location = response.evalJSON(true).url;
        } else {
            alert(translations['_YOUAREATTHELASTLESSONYOUMAYVISIT']);
        }
    }
    //'/{$smarty.server.PHP_SELF}?lessons_ID={$T_NEXT_LESSON}{if
$smarty.session.s_courses_ID}&from_course={$smarty.session.s_courses_ID}{/if}'
}

/**
 * This function automatically navigates to the next unit, if any
 */
function nextUnit() {
    if (typeof(nextId) != 'undefined') {
        window.location = window.location.toString().replace(/view_unit=\d*/,
"view_unit="+nextId);
    }
}
/**
 * This function automatically navigates to the previous unit, if any
 */
function previousUnit() {
    if (typeof(previousId) != 'undefined') {
        window.location = window.location.toString().replace(/view_unit=\d*/,
"view_unit="+previousId);
    }
}

function insertatcursor(myField, myValue) {

    if (document.selection) {
        myField.focus();
        sel = document.selection.createRange();
        sel.text = myValue;
    }
    else if (myField.selectionStart || myField.selectionStart == '0') {
        var startPos = myField.selectionStart;
        var endPos = myField.selectionEnd;
        myField.value = myField.value.substring(0, startPos)+ myValue+
myField.value.substring(endPos, myField.value.length);
    }
}

```

```

    } else {
        myField.value += myValue;
    }
}
var checkToggle;
function togglePdf() {
    $('pdf_upload').toggle();
    $('pdf_upload_max_size').toggle();
    $('pdf_content').toggle();
    $('nonPdfTable').toggle();
    $('toggleTools').toggle();
    if (checkToggle == true) {
        if ($('#editor_content_data').value != '') {
            $('#content_toggle').value = $('#editor_content_data').value;
            $('#editor_content_data').value="";
        } else {
            $('#editor_content_data').value = $('#content_toggle').value;
        }
    }
}

function toggleAdvancedParameters() {

    if ($('#scorm_asynchronous')) {
        Effect.toggle($('#scorm_asynchronous'));
        Effect.toggle($('#scorm_asynchronous_explanation'));
        Effect.toggle($('#scorm_times'));
        Effect.toggle($('#scorm_logging'));
    }
    if ($('#ctg_type')) {
        Effect.toggle($('#ctg_type'));
    }
    Effect.toggle($('#maximize_viewport'));
    Effect.toggle($('#object_ids'));
    if ($('#no_before_unload')) {
        Effect.toggle($('#no_before_unload'));
    }
    Effect.toggle($('#indexed'));
    Effect.toggle($('#accessible_explanation'));
    if ($('#advnced_parameter_image').className.match("down")) {
        setImageSrc($('#advnced_parameter_image'), 16, 'navigate_up.png');
    } else {
        setImageSrc($('#advnced_parameter_image'), 16, 'navigate_down.png');
    }
}

function answerQuestion(el) {
    Element.extend(el);
    $('#correct_answer').hide();
}

```

```

        $('wrong_answer').hide();
        el.up().insert(new Element('img', {src:'themes/default/images/others/progress1.gif',
id:'progress_image'})).setStyle({verticalAlign:'middle', marginLeft:'5px'}));
        $('question_form').request({
            onFailure: function(transport) {
                $('progress_image').remove();
                showMessage(transport.responseText, 'failure');
            },
            onSuccess:function(transport) {
                if (transport.responseText == 'correct') {
                    new Effect.Appear($('correct_answer'));
                    setSeenUnit(1);
                } else {
                    new Effect.Appear($('wrong_answer'));
                    setSeenUnit(0);
                }
                $('progress_image').remove();
            }
        });
    }
}

```

```

function submitCompletionTerm(el) {
    Element.extend(el);
    $('accept_option').hide();
    $('reject_option').hide();
    el.up().insert(new Element('img', {src:'themes/default/images/others/progress1.gif',
id:'progress_image'})).setStyle({verticalAlign:'middle', marginLeft:'5px'}));
    $('accept_form').request({
        onFailure: function(transport) {
            $('progress_image').remove();
            showMessage(transport.responseText, 'failure');
        },
        onSuccess:function(transport) {
            if (transport.responseText == 'accept') {
                new Effect.Appear($('accept_option'));
                setSeenUnit(1);
            } else {
                new Effect.Appear($('reject_option'));
                setSeenUnit(0);
            }
            $('progress_image').remove();
            jQuery('#submit_completion_term').attr('disabled', 'disabled');
        }
    });
}

```

/**

* This function prevents a link from loading upon click.

```

*/
function disableLink (s)
{
    s.stop();
}

function updateProgress(obj) {
    if (!(w = findFrame(top, 'mainframe'))) {
        w = window;
    }

    try {
        var progress            = obj[0];
        var conditions          = obj[1];
        var lesson_passed      = obj[2];
        var unitStatus         = obj[5];
        var certified_course   = obj[6];
        var visits              = obj[7];

        if (w.$('progress_bar')) {

            w.$('progress_bar').select('span.progressNumber')[0].update(parseFloat(progress) +
            '%');

            w.$('progress_bar').select('span.progressBar')[0].setStyle({width:parseFloat(progress) +
            'px'});

            if (w.$('passed_conditions')) {
                w.$('passed_conditions').update(parseInt(conditions));
            }
            if (w.$('lesson_passed')) {
                if (lesson_passed == true) {

                    w.$('lesson_passed').down().removeClassName('failure').addClassName('success')
                    } else {

                    w.$('lesson_passed').down().removeClassName('success').addClassName('failure');
                    }
                }
                if (lesson_passed == true) {
                    if (w.$('completed_block')) {
                        w.$('completed_block').show();
                    }
                    if (certified_course == true && w.$('show_certificate_block')) {
                        w.$('show_certificate_block').show();
                    }
                }
            }

        }

        // if (visits > -1) {

```

```

//          if (w.$('scorm_times')) {
//              if (visits) {
//                  w.$('scorm_times').innerHTML = visits;
//              } else {
//                  w.$('scorm_times').up().innerHTML =
translations['_YOUCANNOTACCESSTHISCONTENTANYMORE'];
//                  w.$('ef-scorm-running').hide();
//              }
//          }
//      }

var status = "";
//console.log(unitStatus);
for (var i in unitStatus) {
    var status = "";

        if (unitStatus[i].completion_status == 'completed' || unitStatus[i].success_status
== 'passed') {
            status = '_passed';
        } else if (unitStatus[i].completion_status == 'incomplete' &&
(unitStatus[i].success_status == 'unknown' || unitStatus[i].success_status == 'incomplete')) {
            status = '_incomplete';
        } else if (unitStatus[i].success_status == 'failed') {
            status = '_failed';
        } else {
            status = "";
        }

        if (w.$('tree_image_'+i)) {
            w.$('tree_image_'+i).className.match(/tests/) ? type = 'tests' : type =
'theory';

            setImageSrc(w.$('tree_image_'+i), 16, type+status);
        }
    }

if (nodesStatus = obj[3]) {
    if ($('navigate_continue')) {
        if (nodesStatus['continue'] == 'enabled') {
            $('navigate_continue').removeClassName('inactiveImage');
            $('navigate_continue').onclick = "";
            $('navigate_continue').show();

        } else if (nodesStatus['continue'] == 'disabled') {
            $('navigate_continue').addClassName('inactiveImage');
            $('navigate_continue').onclick = function () {return false;};
            $('navigate_continue').show();

        } else if (nodesStatus['continue'] == 'hidden') {

```

```

        $('navigate_continue').hide();
    }
}
if ($('navigate_previous')) {
    if (nodesStatus['previous'] == 'enabled') {
        $('navigate_previous').removeClassName('inactiveImage');
        $('navigate_previous').onclick = "";

        $('navigate_previous').show();

    } else if (nodesStatus['previous'] == 'disabled') {
        $('navigate_previous').addClassName('inactiveImage');
        $('navigate_previous').onclick = function () {return false;};

        $('navigate_previous').show();

    } else if (nodesStatus['previous'] == 'hidden') {
        $('navigate_previous').hide();
    }
}
if ($('navigate_exitAll')) {
    if (nodesStatus['exitAll'] == 'enabled') {
        $('navigate_exitAll').removeClassName('inactiveImage');
        $('navigate_exitAll').onclick = "";
        $('navigate_exitAll').show();

    } else if (nodesStatus['exitAll'] == 'disabled') {
        $('navigate_exitAll').addClassName('inactiveImage');
        $('navigate_exitAll').onclick = function () {return false;};
        $('navigate_exitAll').show();

    } else if (nodesStatus['exitAll'] == 'hidden') {
        $('navigate_exitAll').hide();
    }
}
if ($('navigate_suspendAll')) {
    if (nodesStatus['suspendAll'] == 'enabled') {
        $('navigate_suspendAll').removeClassName('inactiveImage');
        $('navigate_suspendAll').onclick = "";
        $('navigate_suspendAll').show();

    } else if (nodesStatus['suspendAll'] == 'disabled') {
        $('navigate_suspendAll').addClassName('inactiveImage');
        $('navigate_suspendAll').onclick = function () {return
false;};

        $('navigate_suspendAll').show();

    } else if (nodesStatus['suspendAll'] == 'hidden') {
        $('navigate_suspendAll').hide();

```

```

    }
}
if ($('#navigate_abandon')) {
    if (nodesStatus['abandon'] == 'enabled') {
        $('#navigate_abandon').removeClassName('inactiveImage');
        $('#navigate_abandon').onclick = "";
        $('#navigate_abandon').show();

    } else if (nodesStatus['abandon'] == 'disabled') {
        $('#navigate_abandon').addClassName('inactiveImage');
        $('#navigate_abandon').onclick = function () {return false;};
        $('#navigate_abandon').show();

    } else if (nodesStatus['abandon'] == 'hidden') {
        $('#navigate_abandon').hide();
    }
}
if ($('#navigate_abandonAll')) {
    if (nodesStatus['abandonAll'] == 'enabled') {
        $('#navigate_abandonAll').removeClassName('inactiveImage');
        $('#navigate_abandonAll').onclick = "";
        $('#navigate_abandonAll').show();

    } else if (nodesStatus['abandonAll'] == 'disabled') {
        $('#navigate_abandonAll').addClassName('inactiveImage');
        $('#navigate_abandonAll').onclick = function () {return
false;};

        $('#navigate_abandonAll').show();

    } else if (nodesStatus['abandonAll'] == 'hidden') {
        $('#navigate_abandonAll').hide();
    }
}
}

for (var i in nodesStatus['choice']) {

    if (nodesStatus['choice'][i] == 'enabled') {
        $('#node'+i).select('a')[0].removeClassName('inactiveLink');
        $('#node'+i).select('a')[0].onclick = "";
        $('#node'+i).show();

    } else if (nodesStatus['choice'][i] == 'disabled') {
        $('#node'+i).select('a')[0].addClassName('inactiveLink');
        $('#node'+i).select('a')[0].onclick = function () {return false;};

        $('#node'+i).show();

    } else if (nodesStatus['choice'][i] == 'hidden') {
        $('#node'+i).hide();
    }
}

```

```

    }
  }
} catch (e) {
  //alert(e);
}
}

if (top.sideframe && !usingHorizontalInterface) {
  if (typeof(show_left_bar) == 'undefined' || parseInt(show_left_bar)) {
    showLeftSidebar();
  } else {
    hideLeftSidebar();
  }
}
/**
 * This function is used to resize scorm iframe, so that it spans through the entire page
 */
function eF_js_setCorrectIframeSize(setHeight)
{
  //Event.observe($('scormFrameID').contentWindow, 'beforeunload', function (s)
  {alert('b');});

  if (frame = window.document.getElementById('scormFrameID')) {
    innerDoc = (frame.contentDocument) ? frame.contentDocument :
frame.contentWindow.document;

    //Some contents send the final commit after the page closes, thus causing the tree and
    progress to not be updated on time.
    //For this reason, we copy the onunload event to a beforeunload event. In order to make
    for some weird contents that use the
    //onunload in other circumstances as well, there is a flag called "noBeforeUnload" that
    disables this event copying
    try {
      if (typeof(noBeforeUnload) == 'undefined' || !noBeforeUnload) {
        //Firefox
        if ($('scormFrameID').contentWindow.onunload) {
          Event.observe($('scormFrameID').contentWindow, 'beforeunload',
$('scormFrameID').contentWindow.onunload);
        }
        //IE
        else if ($('scormFrameID').contentWindow.document.body.onunload) {
          Event.observe($('scormFrameID').contentWindow, 'beforeunload',
$('scormFrameID').contentWindow.document.body.onunload);
        }
        //Sub frames: in case the main doc is frameset, we must go through its
        frames and apply the same
        else if ($('scormFrameID').contentWindow.frames &&
$('scormFrameID').contentWindow.frames.length > 0) {

```

```

        for (var i = 0; i < $('scormFrameID').contentWindow.frames.length; i++) {
            w = $('scormFrameID').contentWindow.frames[i];
            //FF
            if (w.onunload) {
                Event.observe(w, 'beforeunload', w.onunload);
            }
            //IE
            else if (w.document.body.onunload) {
                Event.observe($('scormFrameID').contentWindow,
'beforeunload', w.document.body.onunload);
            }
        }
    }
} catch(err){} //Added due to strange #5567 (unable to get property onunload of
undefined or null reference)
objToResize = (frame.style) ? frame.style : frame;
if (setHeight) {
    objToResize.height = setHeight + 'px';
} else {
    if (frame.document) {
        objToResize.height = Math.max(innerDoc.body.scrollHeight,
frame.document.body.scrollHeight) + 'px';
    } else {
        objToResize.height = innerDoc.body.scrollHeight + 500 + 'px';
    }
}
}
Event.observe($('scormFrameID').contentWindow, 'beforeunload', function (s)
{setTimeout("", 1000)});
}

if (typeof(editPdfContent) != 'undefined' && editPdfContent) {
    togglePdf();
}

if (typeof(setIframeSize) != 'undefined' && setIframeSize) {
    eF_js_setCorrectIframeSize();
}

if (typeof(matchescreenobjectid) != 'undefined' && matchescreenobjectid) {
    var ids = matchescreenobjectid.split(','); //for more than one ids specified
    var viewport = document.viewport.getDimensions();
    var height = viewport.height;

    for(i = 0; i < ids.length; i++){
        var idString = ids[i].toString();
        if ($idString) {
            $(idString).setStyle({height:height+'px'});
        }
    }
}

```

```

        }
    }
}
if (typeof(autoSetSeenUnit) != 'undefined' && autoSetSeenUnit) {
    setSeenUnit(1);
}

function checkLessonConditions() {
    el = document.body;
    parameters = {check_conditions:true, method: 'get'};
    var url = window.location.toString();
    ajaxRequest(el, url, parameters, onCheckLessonConditions);
}

function onCheckLessonConditions(el, response) {
    results = response.evalJSON();
    if ($('#passed_conditions')) {
        $('#passed_conditions').update(parseInt(results[1]));
    }
    if ($('#lesson_passed')) {
        if (results[2] == true) {
            $('#lesson_passed').down().removeClassName('failure').addClassName('success');
            $('#completed_block').show();
            if (results[6] == true) {
                $('#show_certificate_block').show();
            }
            Effect.ScrollTo('completed_block');
        }
    }
}

function startContentTimer() {
    this.formatTime = function (v) {
        if (v == 0) {
            v = '00';
        } else if (v.length == 1) {
            v = "0" + v;
        }
        return v;
    }
}

if (typeof(seconds) != 'undefined') {
    if (seconds < 59) {seconds++;}
    else {
        if (seconds == 59 ) {seconds = 0;}
        if (minutes < 59) {minutes++;}
        else {
            if (minutes == 59) {minutes = 0;}
            hours++;
        }
    }
}

```

```

    }
  }
  min = this.formatTime(minutes.toString());
  sec = this.formatTime(seconds.toString());
  hours = this.formatTime(hours);

  //if (hours.length == 1) {hours = "0" + hours;}
  //if (min.length == 1) {min = "0" + min;}
  //if (sec.length == 1) {sec = "0" + sec;}

  $("user_time_in_unit_display").update(hours + ":" + min + ":" + sec);

  //$("user_current_time_in_unit").update(parseInt($("#user_current_time_in_unit").innerHTML)
+1);

  var newUserTotalTimeInUnit = parseInt($("#user_total_time_in_unit").innerHTML)+1;
  $('#user_total_time_in_unit').update(newUserTotalTimeInUnit);
  if (newUserTotalTimeInUnit == parseInt($('#required_time_in_unit').innerHTML)) {
    setSeenUnit(true);
  }

  if (lesson_seconds < 59) {lesson_seconds++;}
  else {
    if (lesson_seconds == 59) {lesson_seconds = 0;}
    if (lesson_minutes < 59) {lesson_minutes++;}
    else {
      if (lesson_minutes == 59) {lesson_minutes = 0;}
      lesson_hours++;
    }
  }
  lesson_min = this.formatTime(lesson_minutes.toString());
  lesson_sec = this.formatTime(lesson_seconds.toString());
  lesson_hours = this.formatTime(lesson_hours);

  $("user_time_in_lesson_display").update(lesson_hours + ":" + lesson_min + ":" +
lesson_sec);
  var newUserTimeInLesson = parseInt($("#user_time_in_lesson").innerHTML)+1;
  $("#user_time_in_lesson").update(newUserTimeInLesson);
  if (newUserTimeInLesson == parseInt($('#required_time_in_lesson').innerHTML)) {
    checkLessonConditions();
  }
}

//contentTimer = setTimeout("startContentTimer()", 1000);
}
if (typeof(start_timer) != 'undefined' && start_timer) {
  pe = false;

```

```

if (start_timer != 'flash') {
    var isIE = (navigator.appName == "Microsoft Internet Explorer");
    if (isIE) {
        document.onmouseout = function () {
            if (pe) {
                pe.stop();
                dimPage(true);
            }
        };
        document.onfocusin = function() {
            if (pe) {
                pe.stop();
                dimPage(false);
            }
        };
        pe = new PeriodicalExecuter(startContentTimer, 1);
    }
} else {
    window.onblur = function () {
        if (pe) {
            pe.stop();
            dimPage(true);
        }
    };
    window.onfocus = function() {
        if (pe) {
            pe.stop();
            dimPage(false);
        }
    };
    pe = new PeriodicalExecuter(startContentTimer, 1);
}
}

if (!pe) {
    pe = new PeriodicalExecuter(startContentTimer, 1);
}
}

function dimPage(mode) {
    if (typeof(timers) != 'undefined' && timers) {
        if (mode) {
            document.body.appendChild(new
Element('div').addClassName('dimmerDiv').addClassName('dimmerDivTemp').setOpacity(0.5).s
etStyle({height:getDocHeight()+ 'px'}));
        } else {
            $$('.dimmerDivTemp').each(function (s) {s.remove();});
        }
    }
}
}

```

```

function getDocHeight(D) {
  if (!D) {var D = document;}
  return Math.max(
    Math.max(D.body.scrollHeight, D.documentElement.scrollHeight),
    Math.max(D.body.offsetHeight, D.documentElement.offsetHeight),
    Math.max(D.body.clientHeight, D.documentElement.clientHeight)
  );
}

function handleDrop(s,d, e) {
  s.setStyle({left:'auto', top:'auto'});
  d.next().insert(s.remove());
  s.down().value = d.down().value;
  dragdrop[s.id]=d.id
  Droppables.remove(d);
}

function handleDrag(s, e,questionId) {
  if (dragdrop[s.element.id]) {
    Droppables.add($(dragdrop[s.element.id]), {accept:'draggable',
onDrop:handleDrop});

    $('source_'+questionId+'_'+s.element.id.match(/firstlist_\d+_(\d+)/)[1]).insert(s.element
.remove());
  }
}

function initDragDrop(questionId, keys) {
  for (var i = 0; i < keys.length; i++) {
    Droppables.add('secondlist_'+questionId+'_'+keys[i], {accept:'draggable',
onDrop:handleDrop});
    new Draggable('firstlist_'+questionId+'_'+keys[i], {revert:'failure',
onStart:function (s,e) {handleDrag(s,e,questionId);}});
  }
}

dragdrop = new Object();
if (typeof(dragDropQuestions) != 'undefined') {
  dragDropQuestions.each(function (s) {initDragDrop(s, dragDropQuestionKeys[s]);})
}

function retrieveInformation() {
  return _information_json;
}

function setUnitCompletionOptions(el) {
  Element.extend(el);
  if (el.options[el.options.selectedIndex].value == 2) {
    if ($('#complete_question_row')) {
      $('#complete_question_row').show();
    }
    if ($('#complete_time_row')) {

```

```

        $('complete_time_row').hide();
    }
} else if (el.options[el.options.selectedIndex].value == 3) {
    if ($('#complete_question_row')) {
        $('#complete_question_row').hide();
    }
    if ($('#complete_time_row')) {
        $('#complete_time_row').show();
    }
} else {
    if ($('#complete_question_row')) {
        $('#complete_question_row').hide();
    }
    if ($('#complete_time_row')) {
        $('#complete_time_row').hide();
    }
}
}

function setAllUnitsProperties(el, scorm) {
    Element.extend(el);
    if (el.previous().type == 'checkbox') {
        value = el.previous().checked ? 1 : 0;
    } else {
        value = el.previous().value;
    }

    ajaxRequest(el, location.toString(), {ajax:1, method:'get', option:el.previous().name,
value:value, scorm:scorm});
}

```

tests.js

```

function eF_js_printQuestionTimer(id) {
    //The time is up!
    if (questionHours[id] <= 0 && questionMinutes[id] <= 0 && questionSeconds[id] <= 0 &&
questionDuration[id]) {

        //Overlay a div to disable question.
        $("question_content_" + id).up().insert(new Element("div", {id:"question_overlay_" +
id}).addClassName("expiredQuestion").setOpacity(0.3));

        //If it is the current question that expired, disabled it
        if (id == $("question_" + current_question).down().innerHTML) {
            $("question_overlay_" + id).clonePosition($("question_" + current_question));
        }
    } else {
        if (questionSeconds[id] >= 1) {questionSeconds[id]--;}
        else {
            if (questionSeconds[id] == 0) {questionSeconds[id] = 59;}
            if (questionMinutes[id] >= 1) {questionMinutes[id]--;}
            else {
                if (questionMinutes[id] == 0) {questionMinutes[id] = 59;}
                if (questionHours[id] >= 1) {questionHours[id]--;}
                else
                    {questionHours[id] = 0;}
            }
        }
        questionMin[id] = questionMinutes[id].toString();
        questionSec[id] = questionSeconds[id].toString()
        if (questionMin[id].length == 1) {questionMin[id] = "0" + questionMin[id];}
        if (questionSec[id].length == 1) {questionSec[id] = "0" + questionSec[id];}

        $("question_" + id + "_time_left").update(questionHours[id] + ":" + questionMin[id] + ":" +
questionSec[id] + " " + remainingtime);
        $("question_" + id + "_time_left_input").value =
parseInt(questionHours[id])*3600+parseInt(questionMinutes[id])*60+parseInt(questionSecond
s[id]);
        setTimeout("eF_js_printQuestionTimer("+id+")", 1000);
    }
}

function showTestQuestion(question_num) {

    if (question_num == 'next') {
        current_question < total_questions ? question_num = parseInt(current_question) + 1 :
question_num = current_question;
    } else if (question_num == 'previous') {
        current_question > 1 ? question_num = parseInt(current_question) - 1 : question_num =
current_question;
    }
}

```

```

}

$('question_' + current_question).hide();
$('question_' + question_num).show();
var questionId = $('question_' + question_num).down().innerHTML;

if ($('#question_overlay_' + questionId)) {
    $('#question_overlay_' + questionId).clonePosition($('question_' + question_num));
}
current_question = question_num;
if ($('#previous_question_button')) {
    current_question <= 1 ? $('#previous_question_button').hide() :
$('#previous_question_button').show();
}
if ($('#next_question_button')) {
    current_question >= total_questions ? $('#next_question_button').hide() :
$('#next_question_button').show();
}
if ($('#goto_question')) {
    $('#goto_question').selectedIndex = current_question - 1;
}
if ($('#question_' + questionId + '_time_left')) {
    $('#question_' + questionId + '_time_left').up().select('span').each(function (s) {s.hide()});
    $('#question_' + questionId + '_time_left').show();
}

if (questionDuration && questionDuration[questionId]) {
    eF_js_printQuestionTimer(questionId);
}
}

function checkQuestions() {

    var finished = new Array();
    var count = 0;
    var editor_count = 0;

    $$('.unsolvedQuestion').each(function (r) {
        finished[count] = 0;
        if (r.hasClassName('trueFalseQuestion')) {
            r.select('input[type=radio]').each(function (s) {s.checked ? finished[count] = 1 :
null;});
        } else if (r.hasClassName('multipleOneQuestion')) {
            r.select('input[type=radio]').each(function (s) {s.checked ? finished[count] = 1 :
null;});
        } else if (r.hasClassName('multipleManyQuestion')) {
            r.select('input[type=checkbox]').each(function (s) {s.checked ? finished[count] = 1 :
: null;});
        } else if (r.hasClassName('gridQuestion')) {

```

```

                r.select('input[type=checkbox]').each(function (s) {s.checked ? finished[count] = 1
: null;});
            } else if (r.hasClassName('HotspotQuestion')) {
                r.select('input[type=text]').each(function (s) {s.value ? finished[count] = 1 :
null;});
            } else if (r.hasClassName('matchQuestion')) {
                //r.select('select').each(function (s) {s.checked ? finished[count] = 1 : null;});
                finished[count] = 1;
            } else if (r.hasClassName('emptySpacesQuestion')) {
                r.select('input[type=text]').each(function (s) {s.value ? finished[count] = 1 :
null;});
                r.select('select').each(function (s) {s.value ? finished[count] = 1 : null;});
            } else if (r.hasClassName('rawTextQuestion')) {
                if (typeof(tinyMCE) != 'undefined' && r.select('.simpleEditor').length) { // Added
r.hasClassName('simpleEditor') because of #4478
                    console.log(r);
                    if (typeof(tinyMCE.editors[editor_count]) != 'undefined' &&
tinyMCE.editors[editor_count++].getContent()) {
                        finished[count] = 1;
                    }
                } else {
                    r.select('textarea').each(function (s) {s.value ? finished[count] = 1 : null;});
                }

                r.select('input').each(function (s) {s.value ? finished[count] = 1 : null;}); //check
if a file is picked for upload

            } else if (r.hasClassName('dragDropQuestion')) {
                r.select('td.dragDropTarget').each(function (s) {s.childElements().length ?
finished[count] = 1 : null;});
            }

            count++;
        });

        var unfinished = new Array();
        for (var i = 0; i < finished.length; i++) {
            if (!finished[i]) { unfinished.push(i+1); }
        }
        if (unfinished.length) {
            if (force_answer_all == 1) {
                alert (translations['youhavetoanswerallquestions']+' '+unfinished);
                return false;
            } else if(force_answer_all == 0) {
                return confirm(translations['youhavenotcompletedquestions']+' '+
'+unfinished+' '+translations['areyousureyouwanttosubmittest']);
            }
        }
    }
}

```

```

}
function handleDrop(s,d, e) {
    s.setStyle({left:'auto', top:'auto'});
    d.next().insert(s.remove());
    s.down().value = d.down().value;
    dragdrop[s.id]=d.id
    Droppables.remove(d);
}
function handleDrag(s, e,questionId) {
    if (dragdrop[s.element.id]) {
        Droppables.add($(dragdrop[s.element.id]), {accept:'draggable',
onDrop:handleDrop});

        $('source_'+questionId+'_'+s.element.id.match(/firstlist_\d+_(\d+)/)[1]).insert(s.element
.remove());
    }
}
function initDragDrop(questionId, keys) {
    for (var i = 0; i < keys.length; i++) {
        Droppables.add('secondlist_'+questionId+'_'+keys[i], {accept:'draggable',
onDrop:handleDrop});
        new Draggable('firstlist_'+questionId+'_'+keys[i], {revert:'failure',
onStart:function (s,e) {handleDrag(s,e,questionId)}});
    }
}

function toggleAdvancedParameters() {

    $('onebyone').toggle();
    $('only_forward').toggle();

    if ($('action_on_submit')) {
        $('action_on_submit').toggle();
    }
/*
    if ($('given_answers')) {
        $('given_answers').toggle();
    }
    if ($('show_score')) {
        $('show_score').toggle();
    }
    if ($('show_answers_if_pass')) {
        $('show_answers_if_pass').toggle();
    }
    if ($('answers')) {
        $('answers').toggle();
    }
    if ($('redirect')) {
        $('redirect').toggle();

```

```

    }
*/
    $('shuffle_answers').toggle();
    $('shuffle_questions').toggle();
    if ($('#pause_test')) {
        $('#pause_test').toggle();
    }
    $('publish').toggle();
    $('display_list').toggle();
    if ($('#redo_wrong')) {
        $('#redo_wrong').toggle();
    }
    if ($('#answer_all')) {
        $('#answer_all').toggle();
    }
    if ($('#test_password')) {
        $('#test_password').toggle();
    }
    if ($('#custom_class')) {
        $('#custom_class').toggle();
    }
    if ($('#keep_best')) {
        $('#keep_best').toggle();
    }
    if ($('#display_weights')) {
        $('#display_weights').toggle();
    }
    if ($('#advanced_parameter_image').className.match("down")) {
        setImageSrc($('#advanced_parameter_image'), 16, 'navigate_up.png');
    } else {
        setImageSrc($('#advanced_parameter_image'), 16, 'navigate_down.png');
    }
}

//These must be run when a test is shown
if (typeof(showtest) != 'undefined' && showtest) {
    //alert(window.initTimer);
    if (window.initTimer) {initTimer();};
    if (typeof(current_question) != 'undefined') {
        showTestQuestion(current_question);
    }
}

dragdrop = new Object();
if (typeof(dragDropQuestions) != 'undefined') {
    dragDropQuestions.each(function (s) {initDragDrop(s, dragDropQuestionKeys[s]);})
}

```

```

function startAutoCompleter(el, id, options) {
    var arrayOptions = decodeURIComponent(options).evalJSON(true);
    Element.extend(el);
    if (!$('question_autocomplete_'+id)) {
        el.up().up().insert(new Element('div',
{id:'question_autocomplete_'+id}).addClassName('autocomplete'));
    }
    var completer = new Autocompleter.Local(el.identify(),
'question_autocomplete_'+id, arrayOptions, {});
    completer.activate();
}
/*
jQuery('div[id*=question_]').find('input,select,textarea,div.draggable').on('mouseup',
function(event) {
    var current_timestamp = Math.round(new Date().getTime() / 1000);
    if (typeof (window.last_autosave) == 'undefined') {
        window.last_autosave = current_timestamp;
    }
    //console.log(current_timestamp);
    //console.log(window.last_autosave);
    if (current_timestamp > window.last_autosave+10) {
        window.last_autosave = current_timestamp;
        //var url = jQuery(this).parents('form').attr('action');
        jQuery.post(jQuery(this).parents('form').attr('action')+'&auto_save=1',
jQuery(this).parents('form').serialize());
        event.stopPropagation();
    }
});
*/
if (typeof multipleManyQuestionsLimits !== 'undefined') {
    for (var key in multipleManyQuestionsLimits) {
        multipleManyQuestionsLimitsFunction(key, multipleManyQuestionsLimits[key])
    }
}
function multipleManyQuestionsLimitsFunction(question, limit) {
    var theCheckboxes = jQuery("input[name*='question["+question+"]']");
    theCheckboxes.click(function()
    {
        if (theCheckboxes.filter(":checked").length > limit)
            jQuery(this).removeAttr("checked");
    });
}

function setHotspotCoords(id){
    jQuery('#hotspot_image_'+id).click(function(e){
        var x = e.pageX - jQuery(this).offset().left;
        var y = e.pageY - jQuery(this).offset().top;
        jQuery('#question_'+id).val(x+', '+y);
    });
}

```

```

        jQuery("img[id*='pin_'+id+'"]').remove();
        var x_pin = x-4;
        var y_pin= y-10;
        x_pin = x_pin.toFixed(1);
        y_pin = y_pin.toFixed(1);
        jQuery("#hotspot_div_"+id).append('');
    });
}

jQuery(document).ready(function() {
    jQuery("div[id*='hotspot_div_']").each(function(index){
        var hotspot_question_ID = this.id.substr(12);
        var coordinates = jQuery("#question_"+hotspot_question_ID).val();
        var coord= coordinates.split(',');
        var x_pin = coord[0]-4;
        var y_pin= coord[1]-10;
        if (coord != "") {
            jQuery("#hotspot_div_"+hotspot_question_ID).append('');
        }
    });

    if (jQuery("#hotspot_answer").val()) {
        var answer = jQuery("#hotspot_answer").val();
        answer = answer.substr(0,answer.indexOf('| |'));
        answer = answer.replace(' ','');
        answer = answer.replace(' ','');
        answer = answer.replace('(','');
        var coord = answer.split(',');
        var coords = {x1 : coord[0], y1 : coord[1], x2 : coord[2], y2: coord[3]};

        jQuery("#hotspotSpace").imgAreaSelect(coords);
        //alert(answer);
    }

});

```

Аксиоматическая ступень абстракции.

1. Комбинаторные конфигурации

Б.1.3. Правила суммы и произведения

Пусть X – конечное множество, состоящее из n элементов. Тогда говорят, что объект x из X можно выбрать n способами, где $|X| = n$.

Рассмотрим набор попарно непересекающихся множеств X_1, \dots, X_k . Воспользуемся равенством, описанным в разделе теория множеств $|A \cup B| = |A| + |B| - |A \cap B|$. Условие попарно непересекающихся говорит о том, что мощности всех возможных пересечений на наборе множеств X_1, \dots, X_k равны нулю. Следовательно, **правило суммы** звучит следующим образом: если все комбинаторные объекты можно разбить на два непересекающихся множества, причем в первом множестве лежат A комбинаторных объектов, в во втором – B комбинаторных объектов, то всего комбинаторных объектов будет $A + B$.

Другим часто применяемым в комбинаторике правилом является **правило произведения**.

Рассмотрим систему множеств X_1, \dots, X_k , на которой мы хотим организовать выборку, выбрав по одному элементу из каждого множества. Данная выборка может быть осуществлена $|X_1| * |X_2| * \dots * |X_k|$ способами.

Правило произведения: Если объект x можно выбрать m способами и после каждого из таких выборов объект y в свою очередь может быть выбран n способами, то выбор « x и y » в указанном порядке можно осуществить $m * n$ способами.

Качественная ступень абстракции.

1. Комбинаторные конфигурации

Б.1.3. Правила суммы и произведения

Пусть X – конечное множество, состоящее из n элементов. Тогда говорят, что объект x из X можно выбрать n способами, где $|X| = n$.

Рассмотрим набор попарно непересекающихся множеств X_1, \dots, X_k . Воспользуемся равенством, описанным в разделе теория множеств $|A \cup B| = |A| + |B| - |A \cap B|$. Условие попарно непересекающихся говорит о том, что мощности всех возможных пересечений на наборе множеств X_1, \dots, X_k равны нулю. Следовательно, **правило суммы** звучит следующим образом: если все комбинаторные объекты можно разбить на два непересекающихся множества, причем в первом множестве лежат A комбинаторных объектов, в во втором – B комбинаторных объектов, то всего комбинаторных объектов будет $A + B$.

Для $k=2$ правило суммы выглядит следующим образом: если объект x можно выбрать m способами, а объект y – другими n способами, то выбор «либо x , либо y » можно осуществить $m+n$ способами.

Пример. Сколько существует наборов с двумя координатами из множества $B = \{0; 1\}$?

Решение. Все наборы с двумя координатами из множества B разобьем на два непересекающихся множества: наборы с первой координатой 0 и наборы с первой координатой 1.

В первом множестве два набора: (00);(01) ($A = 2$), во втором множестве также два набора: (10);(11) ($B = 2$). Следовательно, всего наборов $A + B = 4$.

Другим часто применяемым в комбинаторике правилом является **правило произведения**.

Рассмотрим систему множеств X_1, \dots, X_k , на которой мы хотим организовать выборку, выбрав по одному элементу из каждого множества. Данная выборка может быть осуществлена $|X_1| * |X_2| * \dots * |X_k|$ способами.

Так, например, если все комбинаторные объекты обладают двумя признаками, причем первый признак может принимать A различных значений, а для каждого комбинаторного объекта с фиксированным первым признаком второй признак может принимать B различных значений, то всего комбинаторных объектов будет $A * B$.

Правило произведения: Если объект x можно выбрать m способами и после каждого из таких выборов объект y в свою очередь может быть выбран n способами, то выбор « x и y » в указанном порядке можно осуществить $m * n$ способами.

Пример. Сколько существует наборов с двумя координатами из множества $B = \{0; 1\}$?

Решение. Все наборы с двумя координатами из множества B обладают двумя признаками: значением первой координаты и значением второй координаты. Первая координата может принимать два различных значения: 0 и 1 ($A = 2$), у каждого набора с фиксированной первой координатой вторая координата также может принимать два значения: 0 и 1 ($B = 2$).

Следовательно, всего наборов $A * B = 4$.

Прогностическая ступень абстракции.

1. Комбинаторные конфигурации

Б.1.3. Правила суммы и произведения

Пусть X – конечное множество, состоящее из n элементов. Тогда говорят, что объект x из X можно выбрать n способами, где $|X| = n$.

Рассмотрим набор попарно непересекающихся множеств X_1, \dots, X_k . Воспользуемся равенством, описанным в разделе теория множеств $|A \cup B| = |A| + |B| - |A \cap B|$. Условие попарно непересекающихся говорит о том, что мощности всех возможных пересечений на наборе множеств X_1, \dots, X_k равны нулю. Следовательно, **правило суммы** звучит следующим образом: если все комбинаторные объекты можно разбить на два непересекающихся множества, причем в первом множестве лежат A комбинаторных объектов, во втором – B комбинаторных объектов, то всего комбинаторных объектов будет $A + B$.

Другим часто применяемым в комбинаторике правилом является **правило произведения**.

Рассмотрим систему множеств X_1, \dots, X_k , на которой мы хотим организовать выборку, выбрав по одному элементу из каждого множества. Данная выборка может быть осуществлена $|X_1| * |X_2| * \dots * |X_k|$ способами.

Правило произведения: Если объект x можно выбрать m способами и после каждого из таких выборов объект y в свою очередь может быть выбран n способами, то выбор « x и y » в указанном порядке можно осуществить $m * n$ способами.

Феноменологическая ступень абстракции.

1. Комбинаторные конфигурации

Б.1.3. Правила суммы и произведения

Пусть X – конечное множество (с исчисляемым количеством входящих в него элементов), состоящее из n элементов. Тогда говорят, что объект x из X можно выбрать n способами, где $|X| = n$.

Рассмотрим набор попарно непересекающихся множеств X_1, \dots, X_k . Воспользуемся равенством, описанным в разделе теория множеств $|A \cup B| = |A| + |B| - |A \cap B|$. Условие попарно непересекающихся говорит о том, что мощности всех возможных пересечений на

наборе множеств X_1, \dots, X_k равны нулю (данное заключение становится очевидным из постановки комбинаторных задач, где говорится о том, что перечислить все возможные комбинации без повторения). Следовательно, мы можем сформулировать **правило суммы**, которое звучит следующим образом: если все комбинаторные объекты можно разбить на два непересекающихся множества, причем в первом множестве лежат A комбинаторных объектов, в во втором – B комбинаторных объектов, то всего комбинаторных объектов будет $A + B$.

Для $k=2$ правило суммы выглядит следующим образом: если объект x можно выбрать m способами, а объект y – другими n способами, то выбор «либо x , либо y » можно осуществить $m+n$ способами.

Из правила суммы следует: если Вы желаете посчитать мощность (количество элементов) множества, которое описано как некоторая комбинация из каких-то подмножеств, то старайтесь выделить не пересекающиеся множества (если этого не сделать, то будет вероятность того, что некоторые элементу будут подсчитаны дважды).

Пример. Сколько существует наборов с двумя координатами из множества $B = \{0; 1\}$?

Решение. Все наборы с двумя координатами из множества B разобьем на два непересекающихся множества: наборы с первой координатой 0 и наборы с первой координатой 1.

В первом множестве два набора: (00);(01) ($A = 2$), во втором множестве также два набора: (10);(11) ($B = 2$).

Следовательно, всего наборов $A + B = 4$.

Другим часто применяемым в комбинаторике правилом является **правило произведения**.

Рассмотрим систему множеств X_1, \dots, X_k , на которой мы хотим организовать выборку, выбрав по одному элементу из каждого множества. Данная выборка может быть осуществлена $|X_1| * |X_2| * \dots * |X_k|$ способами.

Так, например, если все комбинаторные объекты обладают двумя признаками (из двух множеств), причем первый признак может принимать A различных значений (выборка из первого множества), а для каждого комбинаторного объекта с фиксированным первым признаком второй признак (из второго множества) может принимать B различных значений, то всего комбинаторных объектов будет $A * B$.

Правило произведения: Если объект x можно выбрать m способами и после каждого из таких выборов объект y в свою очередь может быть выбран n способами, то выбор « x и y » в указанном порядке можно осуществить $m*n$ способами.

Пример. Сколько существует наборов с двумя координатами из множества $B = \{0; 1\}$?

Решение. Все наборы с двумя координатами из множества B обладают двумя признаками: значением первой координаты и значением второй координаты. Первая координата может принимать два различных значения: 0 и 1 ($A = 2$), у каждого набора с фиксированной первой координатой вторая координата также может принимать два значения: 0 и 1 ($B = 2$).

Следовательно, всего наборов $A*B = 4$.