

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Томский государственный университет систем управления и радиоэлектроники»
(ТУСУР)


На правах рукописи

КРЕЧЕТОВ ИВАН АНАТОЛЬЕВИЧ

**МОДЕЛИ, АЛГОРИТМЫ И ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА
АДАПТИВНОГО ОБУЧЕНИЯ**

05.13.10 «Управление в социальных и экономических системах»

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель
доктор технических наук
Кручинин Владимир Викторович

Томск – 2021

Оглавление

ВВЕДЕНИЕ.....	4
1 ОБЗОР И АНАЛИЗ ПРОГРАММНЫХ СРЕДСТВ И ТЕХНОЛОГИЙ РЕАЛИЗАЦИИ АДАПТИВНОГО ОБУЧЕНИЯ.....	13
1.1 Основные термины и понятия	13
1.2 Модели и методы адаптивного обучения	19
1.2.1 Модель контента	19
1.2.2 Модель обучаемого.....	22
1.2.3 Модель адаптации.....	23
1.3 Обзор программных средств реализации адаптивного обучения....	24
1.3.1 Cerego	25
1.3.2 IADLearning	29
1.3.3 Smart Sparrow.....	33
Выводы по главе 1.....	36
2 МОДЕЛИ И АЛГОРИТМЫ АДАПТИВНОГО ОБУЧЕНИЯ.....	39
2.1 Модель предметной области и модель обучаемого	39
2.2 Модель забывания информации	41
2.3 Постановка задачи.....	47
2.4 Алгоритм адаптивного обучения	51
Выводы по главе 2.....	59
3 ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА ПОСТРОЕНИЯ ТЕХНОЛОГИИ АДАПТИВНОГО ОБУЧЕНИЯ.....	60
3.1 Требования к программному обеспечению.....	60
3.2 Пользовательские интерфейсы системы	66
3.3 Программная реализация системы адаптивного обучения	75
Выводы по главе 3.....	79

4 АНАЛИЗ ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ ПОСТРОЕНИЯ ТЕХНОЛОГИИ АДАПТИВНОГО ОБУЧЕНИЯ.....	81
4.1 Описание методики построения технологии адаптивного обучения.....	81
4.2 Создание адаптивного электронного курса «Химия» и его апробация в учебном процессе	84
4.3 Создание адаптивного электронного курса «Информатика» и его апробация в учебном процессе	88
4.4 Сравнительный анализ инструментальных систем построения адаптивного обучения.....	90
Выводы по главе 4.....	96
ЗАКЛЮЧЕНИЕ.....	97
ЛИТЕРАТУРА	99
ПРИЛОЖЕНИЕ А Экранные формы системы адаптивного обучения	107
ПРИЛОЖЕНИЕ Б Инструкция разработчика по созданию адаптивных электронных курсов с применением системы Nexbe.....	114
ПРИЛОЖЕНИЕ В Свидетельства о государственной регистрации программ для ЭВМ	143
ПРИЛОЖЕНИЕ Г Акты внедрения	144

ВВЕДЕНИЕ

Актуальность

Применяемые в современном образовании формы обучения, будь то очная или дистанционная, ориентированы в первую очередь на «усредненного» обучаемого и практически не учитывают индивидуальные особенности и потребности студентов: их уровень знаний, способности к обучению, мотивацию, личные предпочтения и т.д. При этом эффективность учебного процесса определяется в том числе и технологией разработки и применения учебно-методических материалов, по которым обучаемые осваивают новые знания и навыки. Решением проблемы реализации индивидуального обучения является адаптивное обучение, призванное за счет адаптивных учебных материалов (в общем случае – адаптивного контента) повысить эффективность приобретения новых компетенций. Современная интерпретация понятия «адаптивное обучение» предполагает реализацию учебного процесса на основе применения электронных систем обучения (цифровых учебных платформ, систем дистанционного обучения), содержание учебного контента которых подбирается в автоматическом режиме таким образом, чтобы учесть характеристики и способности конкретного обучаемого.

Большой вклад в развитие адаптивного обучения внесли ученые: И.П. Норенков, А.В. Соловов, А.И. Башмаков, С.В. Тархов, Г.В. Рыбина, Л.А. Растрин, В.Н. Касьянов, Р. Brusilovsky и др. Однако в большинстве случаев их исследования не имели практического воплощения в силу сложности программной реализации систем, относящихся к классу интеллектуальных и экспертных, которые в свою очередь являются требовательными к аппаратно-программному обеспечению и до настоящего момента не были реализованы в виде клиент-серверных решений, способных функционировать для широкого круга удаленных пользователей.

Современные адаптивные системы обучения можно условно разделить на два класса: системы с готовым контентом и инструментальные системы.

Производители систем первого класса предоставляют потребителям (образовательным учреждениям) законченные адаптивные электронные курсы, выполненные под заказ или под конкретные образовательные программы. Подобные электронные курсы интегрируются в системы дистанционного обучения образовательного учреждения либо размещаются на платформе производителя, к которой открывается доступ пользователям. К таким поставщикам можно отнести компании: Knewton, McGraw Hill, Pearson. Системы данного класса не предоставляют инструменты для редактирования и создания контента, исходя из чего не представляют собой интерес для тех потребителей, которые ставят перед собой задачу разработки собственных адаптивных курсов для обеспечения учебного процесса. На решение этой задачи ориентированы системы второго класса, которые представляют собой инструментальное программное обеспечение, предназначенное для создания учебных ресурсов (электронных курсов, тренажеров) с элементами адаптивного обучения. Из известных и доступных для анализа систем можно отметить продукты зарубежных разработчиков: Cerego, IADLearning, Smart Sparrow. Однако данные системы закрыты для разработчиков, имеют высокую стоимость и ряд недостатков методологического и технического характера, не учитывают особенности образования в России. Стоит также отметить, что на сегодняшний день на рынке отечественного программного обеспечения не существует решений, предназначенных для разработки и внедрения адаптивных электронных ресурсов.

Реализация инструментальной учебной системы в технологии адаптивного обучения основывается на применении моделей обучаемого, модели учебного контента, а также модели адаптации. Разработка новых или выбор существующих моделей является вопросом научного исследования из-за их непосредственного влияния на функциональные возможности системы и адекватности получаемого результата.

Таким образом, актуальной является задача разработки моделей, методов и инструментальной системы для реализации технологии адаптивного обучения.

Цели и задачи

Целью настоящей диссертационной работы является разработка моделей, алгоритмов и инструментальной системы для технологии адаптивного обучения в системах дистанционного обучения.

Основные задачи диссертационного исследования:

1. Анализ моделей и методов адаптивного обучения.
2. Анализ программных средств разработки адаптивных электронных курсов.
3. Теоретическое исследование построения технологии адаптивного обучения, построение интегрированной модели, включающей модель предметной области, модель кривой забывания, модель обучаемого.
4. Разработка и исследование алгоритма адаптивного обучения.
5. Выбор технологии реализации и описание структуры программного обеспечения, описание программного и алгоритмического обеспечения.
6. Исследование полученных методик и инструментального средства, описание внедрения адаптивного обучения в реальный учебный процесс.
7. Сравнение полученных инструментальных средств с известными.

Объект исследования

Объектом исследования являются методики и технологии адаптивного обучения в системах дистанционного обучения вуза.

Предмет исследования

Предметом исследования являются модели, алгоритмы и инструментальные средства адаптивного обучения на основе кривой забывания.

Научная новизна

1. Предложена оригинальная модель адаптивного обучения, отличающаяся от известных тем, что в ней агрегированы модель забывания, модель предметной

области, модель обучаемого, и позволяющая автоматизировать процесс адаптивного обучения.

2. Получен новый алгоритм определения индивидуальной траектории обучения в системе дистанционного обучения, основанный на разработанной модели адаптивного обучения.

3. Разработан оригинальный функционал инструментальной системы организации процесса адаптивного обучения, отличающийся от существующих аналогов автоматическим формированием траектории, адаптацией траектории на основе изменения состояний модели обучаемого, функцией поиска контента по базе данных системы, возможностью использования одного и того же контента в разных курсах, поддержкой итеративного обучения, доступом обучаемого к своей истории обучения.

4. Получена новая методика, позволяющая уменьшить затраты на реализацию технологии адаптивного обучения.

Положения, выносимые на защиту

1. Предложенная агрегированная модель адаптивного обучения, включающая модель забывания, модель предметной области, модель обучаемого, позволяет автоматизировать технологию адаптивного обучения в современных системах дистанционного обучения.

2. Разработанный алгоритм определения индивидуальной траектории обучения, основанный на применении модели адаптивного обучения, позволяет получить требуемый уровень знаний на момент окончания курса.

3. Полученные оригинальные структура и функционал инструментальной системы являются полными и обеспечивают реализацию моделей и алгоритмов технологии адаптивного обучения в системах дистанционного обучения.

4. Методика построения технологии адаптивного обучения, включающая следующие основные этапы: получение списка субкомпетенций, разработка учебных модулей, разработка тестовых заданий, запуск адаптивного электронного

курса, – позволяет экономить время разработчиков на 37% по сравнению с известными.

Соответствие результатов диссертационной работы и направлений исследований специальности 05.13.10

Полученная агрегированная модель адаптивного обучения соответствует п. 3 «Разработка моделей описания и оценок эффективности решения задач управления и принятия решений в социальных и экономических системах».

Полученные алгоритм определения индивидуальной траектории обучения и методика построения технологии адаптивного обучения соответствуют п. 4 «Разработка методов и алгоритмов решения задач управления и принятия решений в социальных и экономических системах».

Полученная инструментальная система соответствует п. 5 «Разработка специального математического и программного обеспечения систем управления и механизмов принятия решений в социальных и экономических системах».

Достоверность результатов работы

Достоверность диссертационного исследования обеспечивается корректным применением аппарата математической теории забывания, использованием разработанных моделей и алгоритмов на практике, сравнением полученных результатов с известными. Проведена статистическая обработка результатов внедрения.

Теоретическая значимость

Теоретическая значимость заключается в развитии и конкретизации моделей и алгоритмов адаптивного обучения на основе использования кривой забывания.

Практическая значимость

Практическая значимость работы обуславливается возможностью использования разработанных моделей и программных средств для построения технологии адаптивного обучения в системах дистанционного обучения. Внедрение разработок позволяет:

- 1) снизить затраты на реализацию проектов по адаптивному обучению посредством применения инструментальной системы;
- 2) повысить качество обучения за счет применения в образовательном процессе технологий адаптивного обучения;
- 3) повысить конкурентоспособность российского высшего образования на внешних рынках.

Методы исследования

Методы системного анализа, методы, основанные на применении кривой забывания; методы оптимизации, методы объектно-ориентированного программирования, методы проектирования, разработки и тестирования программного обеспечения.

Внедрение

Основные результаты диссертационной работы используются в Томском государственном университете систем управления и радиоэлектроники, Национальном исследовательском Томском государственном университете, Национальном исследовательском техническом университете «МИСиС».

Компоненты структуры и функционала разработанного программного обеспечения зарегистрированы в Реестре программ для ЭВМ (свидетельство № 2021612218).

Результаты диссертационного исследования использованы в ФГБОУ ВО «ТУСУР» при выполнении государственного задания Министерства науки и высшего образования РФ, проект FEWM-2020-0036 «Методологическое и

инструментальное обеспечение принятия решений в задачах управления социально-экономическими системами и процессами в гетерогенной информационной среде».

Апробация результатов работы

Основные результаты диссертации были доложены на международных научных и научно-методических конференциях и симпозиумах, всероссийской научно-технической конференции, в том числе:

1) XXI Международная научно-методическая конференция «Современное образование: повышение конкурентоспособности университетов», г. Томск, ТУСУР, 2021;

2) SEFI 48th Annual Conference Engaging Engineering Education Proceedings, г. Энсхеде, Нидерланды, 2020;

3) Международная научно-методическая конференция «Современные тенденции развития непрерывного образования: вызовы цифровой экономики», г. Томск, ТУСУР, 2020;

4) Международная конференция по новым образовательным технологиям EdCrunch Томск, г. Томск, 2019;

5) Международная научно-методическая конференция «Современное образование: качество образования и актуальные проблемы современной высшей школы», г. Томск, ТУСУР, 2019;

6) Международная конференции eLearning Stakeholders and Researchers Summit, г. Москва, 2018;

7) Международная научно-методическая конференция «Современное образование: повышение профессиональной компетентности преподавателей вуза – гарантия обеспечения качества образования», г. Томск, ТУСУР, 2018;

8) Международная глобальная конференция по технологиям в образовании EdCrunch Moscow, г. Москва, 2018;

9) Международная глобальная конференция по технологиям в образовании EdCrunch Томск, г. Томск, 2018;

10) Международная конференции eLearning Stakeholders and Researchers Summit, г. Москва, 2017;

11) Международная научно-методическая конференция «Современное образование: проблемы взаимосвязи образовательных и профессиональных стандартов», г. Томск, ТУСУР, 2016;

12) Международная научно-методическая конференция «Современное образование: практико-ориентированные технологии подготовки инженерных кадров», г. Томск, ТУСУР, 2015;

13) II Международный Поспеловский симпозиум «Гибридные и синергетические интеллектуальные системы» (ГИСИС'2014), г. Светлогорск, 2014;

14) Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых «Научная сессия ТУСУР – 2013», г. Томск, ТУСУР, 2013;

15) Международная юбилейная научно-методическая конференция «Современное образование: проблемы обеспечения качества подготовки специалистов в условиях перехода к многоуровневой системе высшего образования», г. Томск, ТУСУР, 2012;

16) I Международный симпозиум «Гибридные и синергетические интеллектуальные системы: теория и практика» (ГИСИС'2012), г. Светлогорск, 2012.

Публикации

По результатам выполненных исследований было опубликовано 17 научных работ, в том числе 7 работ в изданиях, включенных в перечень ВАК, и изданиях Scopus.

Личный вклад

Содержательная и математическая постановка задач осуществлялись совместно с научным руководителем. Разработка моделей, алгоритмов и методик выполнена лично автором.

Объем работы

Диссертация состоит из введения, четырех глав, заключения, списка литературы, включающего 70 наименований, четырех приложений. Объем работы составляет 150 страниц машинописного текста.

1 ОБЗОР И АНАЛИЗ ПРОГРАММНЫХ СРЕДСТВ И ТЕХНОЛОГИЙ РЕАЛИЗАЦИИ АДАПТИВНОГО ОБУЧЕНИЯ

1.1 Основные термины и понятия

С развитием информационных технологий появляются новые методы и технологии обучения. Сегодня всё большую популярность обретает технология адаптивного обучения, призванная повысить эффективность обучения за счет учета индивидуальных особенностей обучаемых. В сфере образования эта сфера считается относительно новой, поскольку практическое воплощение систем адаптивного обучения для широкого доступа пользователей стало возможным только в начале XXI в. Модели и алгоритмы, лежащие в основе таких систем, имеют существенные недостатки. Для понимания сути описываемого ниже приведены термины и понятия, используемые в диссертации.

Адаптивное обучение. Поиск эффективных методов и форматов обучения является одной из фундаментальных задач образования, которая никогда не теряла актуальность. Набирающим сегодня популярность подходом является адаптивное обучение.

Анализируя работы современных исследователей, можно сказать, что понятие адаптивного обучения сегодня редко вызывает разночтения в интерпретации. В работе [1] автор рассматривает поиск образовательных ресурсов, набор и последовательность представления материала в которых выстроены с учетом целей, текущего поля знаний и характеристик обучаемого. В [2] исследуются адаптивные среды формирования образовательных ресурсов как решение для приспособления процесса обучения к уровню знаний и умений, а также психологическим особенностям обучаемых. Согласно [3], адаптивные системы используют модель пользователя для сбора информации о его знаниях, целях, опыте и т.д. с целью адаптации содержания и навигационной структуры. В этих системах основное внимание уделяется знаниям обучающихся, которые могут сильно различаться. Состояние знаний изменяется во время работы с системой.

Корректное моделирование изменяющегося состояния знаний, надлежащее обновление модели и способность делать правильные заключения на базе обновленной оценки знаний являются важнейшей составляющей обучающей гипермедиа-системы [4, 5]. Ряд авторов под адаптивностью понимают некоторую индивидуализацию обучения, освоение электронного курса по индивидуальным траекториям (маршрутам) обучения [6–8]. В [9] адаптивное обучение описывается как система, работающая на основе адаптивной модели обучаемого, в которой учитываются индивидуальные факторы процессов запоминания и забывания каждого конкретного обучаемого.

Понятие адаптивного обучения сегодня существует вне отрыва от электронного обучения, поскольку реализуется средствами информационных технологий – современными алгоритмами построения индивидуальных траекторий. Эти алгоритмы лежат в основе *систем адаптивного обучения* либо могут являться частью функциональных возможностей систем дистанционного обучения.

Система дистанционного обучения (СДО). СДО представляет собой программное обеспечение (платформу), функционирующее в сети Интернет и предназначенное для реализации дистанционного (электронного) обучения. Доступ к СДО производится посредством интернет-браузеров, при этом для различных типов пользователей применяется соответствующая система прав и ролей. Как правило, в СДО присутствуют инструменты, предоставляющие авторам возможность создавать своими силами электронные курсы, в которых обучаемым доступны учебные мероприятия (освоение лекционных материалов, прохождение компьютерного тестирования, коммуникации с преподавателем и т.д.). Адаптивное обучение не является обязательным элементом в таких системах, поскольку в большинстве случаев СДО разрабатываются для широкого спектра применения.

Система адаптивного обучения (САО). САО является типом системы дистанционного обучения, в которой обучение реализуется по технологии адаптивного обучения. В таких системах обучаемому предлагается к освоению

контент, соответствующий его индивидуальным особенностям. За подбор контента в таких системах отвечают алгоритмы, которые производят вычисление индивидуальных траекторий на основе учебного опыта обучаемых. Многие разработчики-издатели предлагают САО с готовыми адаптивными электронными курсами, без инструментов для создания собственных. С другой стороны, существуют системы адаптивного обучения, которые включают в себя инструменты создания адаптивных электронных курсов. В последнем случае для создания акцента на наличии соответствующего функционала в САО могут использоваться названия: *инструментальная система адаптивного обучения, инструментальная система построения технологии адаптивного обучения, инструментальное программное средство реализации адаптивного обучения*. САО может быть реализована как самостоятельная система или может являться подсистемой (программной надстройкой) СДО.

Наиболее распространенным подходом при реализации адаптивного обучения является представление образовательного контента электронных учебных курсов в виде элементарных *модулей*.

Контент (от англ. *content* – содержимое). Этот термин зародился во времена появления первых сайтов сети Интернет и закрепился в терминологии сетевых информационных технологий. В общем случае под контентом подразумеваются данные, размещаемые на информационном ресурсе для конечного пользователя (текст, аудио/видео, файлы). В сфере образования действуют стандарты, которые ориентированы на применение дистанционных образовательных технологий при реализации образовательных программ в различных формах, в них можно встретить термин «образовательный контент», который также означает данные, размещаемые в электронной информационной образовательной среде. Контент является содержимым электронного курса в СДО.

В рамках настоящего исследования под образовательным контентом будем понимать данные, которые предъявляются обучаемому в рамках его обучения посредством СДО – электронные конспекты, текстовые лекции, тестовые задания,

изображения, видеоролики, виртуальные интерактивные элементы (тренажеры, лабораторные) и т.д.

Модуль (учебный модуль) – логически завершенная минимальная единица информации образовательного характера, которая раскрывает один или несколько терминов или понятий и находится в связи с другими единицами. Модуль может быть представлен текстом, содержать графику, видео или аудио, а также любые иные интерактивные формы представления информации [44]. Вопросами развития форматов представления и интерактивных возможностей образовательного контента занимаются различные консорциумы (ADL, AICC, IMS и др.). Результатом работы таких организаций стала новая концепция построения электронных образовательных ресурсов на основе объектного принципа.

Концепция заключается в представлении образовательного контента в виде логически завершенных минимальных единиц. В результате разделения контента на объекты (отдельные модули) постепенно уходят в прошлое большие по объему и негибкие по структуре массивы учебной информации и становятся все более востребованными многократно используемые Module Learning Object (англ. – модульные образовательные объекты) [10]. Преимущество такого подхода заключается в легкой доступности таких модулей для поиска на сервере образовательного учреждения и включения их в состав электронного курса. В наиболее востребованном в сфере электронного обучения международном стандарте SCORM [11] дано определение термина Shareable Content Objects (SCO) (англ. – разделяемые объекты контента): SCO – автономная единица учебного материала, имеющая метаданные и содержательную часть. Модули SCO могут в различных сочетаниях объединяться друг с другом в составе электронных курсов, для подобных задач компиляции используются системы дистанционного обучения. В обзоре [12] автор ряда отечественных исследований в области адаптивного обучения подчеркивает преимущества технологии разделяемых объектов – возможность адаптации содержания электронных курсов к индивидуальным

особенностям обучаемых, упрощение и удешевление сопровождения электронных образовательных ресурсов.

К настоящему времени повсеместное применение в контексте адаптивного обучения обрел термин Learning Object (LO) [13]. В дальнейшем для простоты употребления будет использоваться термин «модуль» и в равном ему значении «учебный модуль».

По мнению авторов [14], признаком адаптивности технологий создания электронных образовательных ресурсов (ЭОР) и соответствующих им инструментальных сред является высокая степень вариативности обеспечиваемых маршрутов обучения при возможности настройки маршрутов (автоматической или ручной) локальным преподавателем (тьютором) или самим обучаемым. Возможности адаптации существенно зависят от способа структурирования материала, а именно от размера учебных модулей и от системы межмодульных связей. При крупномодульной структуре число модулей невелико, разработка ЭОР упрощается, однако адаптационные возможности технологии оказываются весьма ограниченными. С уменьшением размера модулей вариативность траекторий изучения материала и степень адаптивности возрастают.

Принцип построения ЭОР из автономных модулей широко распространен в задачах доставки обучаемому адаптивного контента. Примером использования данной технологии является концепция под названием «открытая образовательная модульная мультимедиа-система» (ОМС) [15]. Основным принципом организации данных в ОМС является разделение совокупного контента по предмету на автономные модули по тематическим элементам и компонентам учебного процесса (получение информации, практические занятия, контроль). Кроме электронных учебных модулей, содержащих образовательный контент по предмету, ОМС предусматривает так называемый *модуль методической поддержки*, который задает последовательность модулей, составляющих курс обучения по определенной траектории.

При анализе совокупного контента ОМС по предмету пользователь (преподаватель, обучающийся) выбирает комфортные для него варианты, т.е. создает индивидуальную траекторию в массиве совокупного контента. Программные компоненты ОМС образуют функциональную среду, обеспечивающую хранение, поиск, выбор и воспроизведение контента. Прочие подробности о создании ЭОР представлены в работе [15].

Субкомпетенция, результат обучения. Понятие модуля тесно сопряжено с понятием результатов, получаемых по итогу прохождения модуля. Это может быть усвоение темы, понятия или термина, освоение навыка. В методике разработки модулей они должны быть охарактеризованы входными результатами, которые студент должен получить ранее, а также выходными, которые студент получит после прохождения рассматриваемого модуля.

В зарубежных источниках укоренились термины Learning Objectives и Learning Outcomes, дословный перевод которых в отечественной литературе можно встретить в виде понятий «цели обучения» и «результаты обучения». В [16] автор приводит термин «субкомпетенция» – элементарная составляющая, получаемая при декомпозиции компетенций, прописываемых в государственных стандартах. В дальнейшем термины «цели обучения», «результаты обучения» и «субкомпетенция» употребляются как синонимичные.

Модель студента (модель обучаемого). Как уже было отмечено выше, при адаптивном обучении система по определенным алгоритмам осуществляет индивидуальный вывод контента для каждого студента. Соответственно, адаптивное обучение можно рассматривать как систему, в которой производится управление объектом (студентом). Для того чтобы управляющая система могла реализовывать алгоритмы управления, она должна иметь представление об объекте управления, работать с его моделью. В интеллектуальных обучающих системах такой моделью является модель обучаемого или модель студента. Модель студента представляет собой цифровой образ пользователя в системе, включающий информацию о целях обучения, входных и текущих знаниях, индивидуальных

особенностях и характеристиках, способностях к обучению, свойствах памяти и т.д. Также модель студента должна включать в себя правила (методы) накопления и обработки информации об объекте [17]. Наличие и состав модели студента будут определять возможности управлением процессом обучения и его эффективность.

1.2 Модели и методы адаптивного обучения

Архитектуру адаптивных систем обучения рассматривают в виде трёх составляющих: модель предметной области (контента), модели обучаемого, модели адаптации [4, 60, 62]. Рассмотрим их подробнее.

1.2.1 Модель контента

Среди моделей представления контента в цифровых системах самой распространенной является гипертекстовая модель [67], лежащая в основе стандарта разметки документов HTML и являющаяся главной технологией вывода информации посредством сети Интернет на экраны пользователей.

В задачах разработки и публикации учебного контента в веб-ориентированных образовательных системах гипертекстовая модель успешно решает задачи визуализации учебной информации и навигации по ней. Но сама по себе гипертекстовая модель контента не определяет систему, в которой она используется, как адаптивную: слабая структурированность данных, представленных в такой модели (в частности, отсутствие типизации данных), существенно ограничивает возможности их программной обработки, что в свою очередь сказывается на функциональных возможностях образовательных ресурсов. Классические гипертекстовые системы способны решать задачи предоставления обучаемому контента, но в большинстве случаев такой контент статичен и не учитывает характеристики обучаемых [18–20].

Соответственно, одним из вопросов, требующих рассмотрения, является модель учебного контента, используемая при построении систем адаптивного обучения.

Для того чтобы контент был гибким к задаче адаптации, большинство авторов в своих научных работах сходятся во мнении, что он должен быть представлен в виде элементарных единиц. Рассмотрим некоторые из таких работ.

В исследовании [4] авторы рассматривают модель предметной области как совокупность концептов и межконцептных отношений, где концепт – абстрактный объект, используемый для представления элементов информации предметной области. Межконцептные отношения могут быть типизированы для определения последовательности их изучения, требуемых уровней знаний о концептах и др. (рис. 1.1).

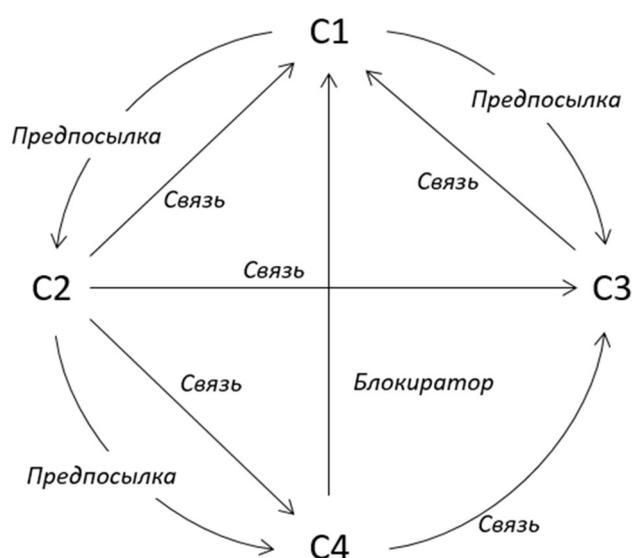


Рисунок 1.1 – Пример структуры межконцептных отношений

Фреймовая модель [5] предполагает наличие у концептов внутренней структуры в виде множества атрибутов, где концепты различных типов могут иметь различные множества атрибутов.

В работе [21] автор определяет в качестве модели предметной области семантическую сеть (рис. 1.2), состоящую из множества модулей и концептов. Концепт в данном случае – понятие в электронном курсе или пособии, которое должно быть освоено. Модуль характеризуется входными (которые требуется освоить) и выходными концептами (осваиваются в рассматриваемом модуле), а также включает в себя тело модуля и метаданные.

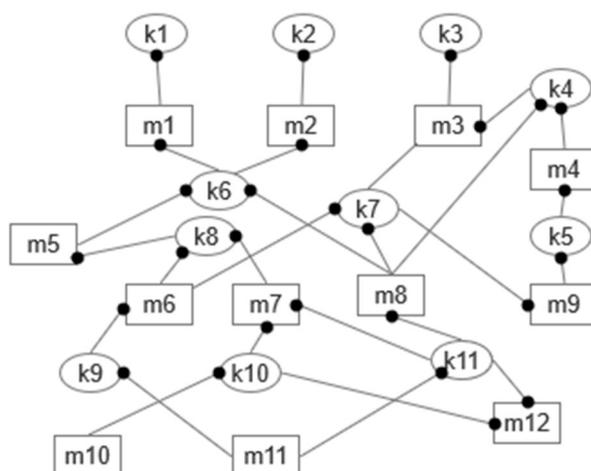


Рисунок 1.2 – Пример семантической сети базы ЭОР

Любой концепт, как отмечает автор, может оказаться в списках выходов нескольких модулей. Модули могут различаться методическими особенностями: стилем изложения материала, сложностью, объемом, форматом представления, ссылками на примеры, подмножеством входных концептов и т.д. Такая вариативность позволяет обеспечивать широкие возможности индивидуализации образовательных траекторий.

Подобный подход, при котором модель предметной области представляет собой множество неделимых объектов с отношениями, можно встретить во многих исследованиях. В работе [22] предлагается структура учебного контента, в которой модули имеют типизированные отношения между собой, описываемые в соответствующих метаданных (рис. 1.3).

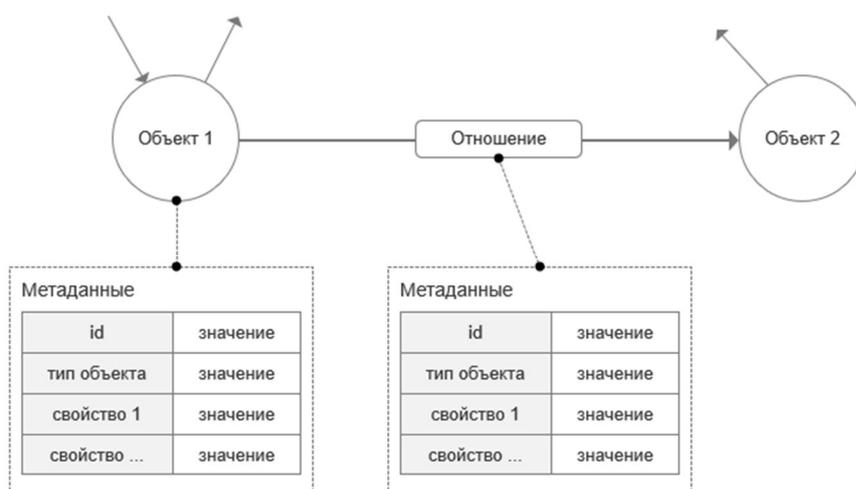


Рисунок 1.3 – Пример структуры отношений между единицами контента

Формализация предметной области в виде модели (продукционной, фреймовой, логической, на основе онтологий или в виде семантической сети) предоставляет возможность сопоставить концептам предметной области набор метаданных, связей или правил, что расширяет применимость частично адаптивных и адаптивных методов сборки блоков учебного контента и контрольных заданий [22]. Схожие по своему основному назначению модели можно встретить в работах [23–27, 58, 63].

1.2.2 Модель обучаемого

В практике построения систем адаптивного обучения широкое применение обрела *оверлейная* модель обучаемого [18, 22, 28]. Оверлейная модель представляет собой структуру предметной области, состоящей из понятий, находящихся в связи между собой. Понятия в различных системах именуется по-разному – концептами, темами, элементами знания, объектами, результатами обучения, субкомпетенциями [12, 16]. Во всех случаях они являются элементарными единицами знаний для рассматриваемой предметной области.

Оверлейная модель позволяет представить знания обучаемого в виде набора значений по каждому структурному элементу модели предметной области (пары «понятие – значение»). Значение является оценкой уровня знания для конкретного обучаемого по конкретному понятию и может измеряться различными шкалами: двоичной (знает – не знает), качественной (отлично – хорошо – удовлетворительно), количественной (в баллах или процентах). Отмечается, что оверлейные модели мощны и гибки, способны независимо измерять знание обучаемым различных тем [18].

Модель обучаемого динамична и изменяется в ходе его работы с адаптивной системой. Оценка знаний, как правило, производится посредством компьютерного тестирования или диагностики действий обучаемого в ходе его диалога с системой.

1.2.3 Модель адаптации

В литературе, посвященной реализации систем адаптивного обучения, практически повсеместно используются термины *гипертекстовая система* и *гипермедиа система*. Объясняется это тем, что в практике разработки электронных образовательных ресурсов широкое применение обрели гипертекстовая модель представления контента и системы, построенные на смежных (веб-) технологиях. Эти системы способны хранить информацию в виде электронного текста, устанавливать электронные связи между информационными единицами, хранящимися в ее памяти, и вызывать их на экран монитора одним нажатием кнопки [6]. Любая гипертекстовая система основана на гипертекстовой модели данных (языках гипертекстовой разметки данных). В настоящее время под гипермедиа системой можно подразумевать веб-приложение, архитектура которого основана на модели клиент-сервер. Принципиально разделять системы по типу гипермедиа и гипертекст сегодня не имеет смысла, поскольку современные стандарты гипертекста (HTML) поддерживают все современные форматы представления данных на экране (текст, графика, видео, аудио). Для разработчиков стало очевидным, что реализация адаптивной системы обучения для широкого круга пользователей должна производиться на базе веб-технологий. Так появился термин *адаптивные гипермедиа*, это технологии, призванные повысить функциональные возможности гипермедиа систем за счет формирования модели пользователя.

Модель адаптивных гипермедиа [5, 18, 29–33] определяет широкий перечень возможных сценариев поведения учебной системы, формируя тем самым для обучающегося уникальный образовательный опыт [34, 59]. В работе [18] автор приводит классификацию адаптивных гипермедиа:

- *системы с адаптивным представлением контента* – вывод контента в них производится в соответствии с моделью пользователя посредством сокрытия некоторой специализированной информации или добавления дополнительной информации;

- *системы с адаптивной навигационной поддержкой* – навигация по контенту в таких системах производится посредством адаптации ссылок и структуры навигации в соответствии с моделью пользователя.

Современные программные продукты с поддержкой адаптивного обучения, как правило, комбинируют оба способа адаптации контента. В классе систем с навигационной поддержкой также стоит отметить две технологии навигации. Первая технология имеет название *«полное руководство»*. Полное руководство применяется в системах, которые решают за пользователя, каким будет следующий наилучший модуль. При этом выдача следующего модуля основана на предшествующих результатах обучения и действиях обучаемого [69]. Реализацией технологии полного руководства может служить метод под названием *«следующий наилучший»*, который выражается в размещении интерфейса рабочего окна учащегося кнопки «далее», которая является основным управляющим элементом навигации по модулям и предполагает, что следующий модуль система подберет на основе предшествующего опыта обучаемого в системе. Примерами систем, использующих данный метод, могут служить платформы Knewton [35], McGraw Hill [36], Plario [41].

Вторая технология навигации – *адаптивное сокрытие ссылок*, смысл которой заключается в ограничении навигационного пространства путем скрывания ссылок к «нерелевантным» модулям. Нерелевантность может определяться неготовностью пользователя осваивать тот или иной образовательный материал, несоответствием целей обучения и т.д. [64, 68]. Подход сокрытия ссылок защищает пользователя от познавательной перегрузки в неограниченном гиперпространстве. Примером здесь является образовательная платформа Khan Academy [37].

1.3 Обзор программных средств реализации адаптивного обучения

При выборе программных средств для сравнительного анализа учитывалось следующее:

- 1) программное обеспечение предназначено для реализации электронного обучения с элементами адаптивного обучения. Признаком, в соответствии с которым рассматриваемые решения потенциально можно считать реализующими адаптивное обучение, будем считать зависимость получаемой в процессе обучения траектории от выполняемых пользователем действий;
- 2) программное обеспечение содержит инструментальные средства создания адаптивных электронных курсов;
- 3) разрабатываемые адаптивные электронные курсы могут быть интегрированы во внешние системы дистанционного обучения.

1.3.1 Cerego

Программное решение Cerego представляет собой веб-сервис для адаптивного обучения на базе применения кривой забывания [38].

В процессе прохождения электронного курса обучаемый работает с той последовательностью модулей, которую определил разработчик курса. В то же время алгоритмы Cerego автоматически добавляют в траекторию обучаемого элементы, направленные на проверку остаточных знаний или повторение ранее изученного материала, в зависимости от срабатывания кривой забывания. Cerego применяет различные типы оценивающих заданий. Например, первичная оценка знаний может быть осуществлена посредством тестового вопроса на множественный выбор, следующим может быть вопрос на заполнение пробелов и т.д.

На экран обучаемому выводится только один модуль (теоретический фрагмент или тест), переход к следующему осуществляется посредством кнопки Next (далее) (рис. 1.4). Каким будет следующий элемент – пользователю неизвестно. Навигация по содержанию курса также не предусмотрена.

The screenshot shows the Cerego Learning Science 101 interface. At the top left is the Cerego logo and the text 'Learning Science 101'. At the top right is a progress indicator showing '15%' and icons for mobile, help, settings, and 'Exit'. The main title is 'Distributed Learning'. On the left is a portrait of Hermann Ebbinghaus with the caption 'Hermann Ebbinghaus'. On the right is text explaining distributed learning, mentioning Ebbinghaus's work and the Ebbinghaus Forgetting Curve. A green 'Next' button is at the bottom right.

Рисунок 1.4 – Интерфейс обучаемого в образовательном веб-сервисе Cerego

Cerego отслеживает прогресс по каждому концепту (субкомпетенции) и предоставляет пользователю посредством интерфейса информацию об истории взаимодействия с курсом. Пользователю доступен функционал:

1. *Общий прогресс.* В соответствующем разделе интерфейса выводится принадлежность осваиваемых концептов условным уровням, отражающим, на какой период времени информация задержится в памяти.

2. *Последний просмотр и предстоящее напоминание/проверка.* Система отражает, когда был освоен тот или иной концепт, а также информирует о том, когда будет произведено его повторение (повторный вывод соответствующего модуля).

3. *Сложность материала.* В зависимости от того, как успешно пользователь справляется с заданиями в курсе, Cerego относит концепты к соответствующим уровням сложности. Одни и те же концепты для разных пользователей будут отмечены разной сложностью.

4. Раздел мониторинга, позволяющий производить *анализ затраченного на освоение концептов времени*.

Обратимся к особенностям разработки контента и создания адаптивного курса в системе Cerego. Работа начинается с создания набора элементов (Set). Элементами могут выступать различные формы представления контента или проверки знаний. По сути Set, включающий в себя несколько элементов, уже может являться курсом. В рамках каждого набора Set предлагается создать элемент из 11 зарезервированных типов (рис. 1.5).

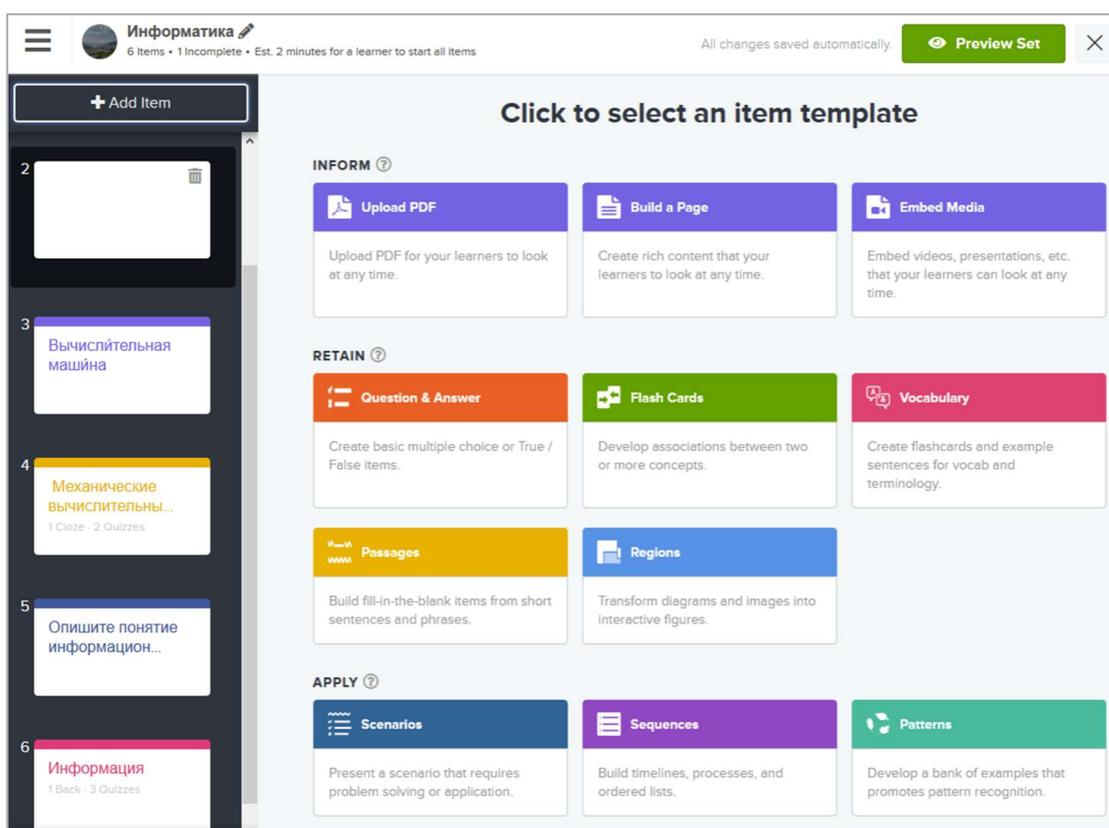


Рисунок 1.5 – Интерфейс редактора курса в Cerego

Создавая элементы в нужной очередности, разработчик тем самым определяет образовательную траекторию. Однако эта очередность скорее говорит о логической последовательности тем, а не о получившейся длине траектории, поскольку алгоритмы Cerego автоматически управляют включением в траекторию элементов, направленных на повторение или проверку знаний.

Cerego позволяет создавать элементы с содержимым информационного (группа Inrofm: PDF-файл, HTML-страница, медиа), а также интерактивного характера, подразумевающие необходимость выполнения пользователем действий (дать ответы на вопросы). Форма этих вопросов может быть разной, для их реализации предусмотрено 7 типов (выбор вариантов, установление соответствий, заполнение пробелов и т.д.). Все интерактивные элементы разбиты на две группы (Retain – закрепление; Apply – применение), хотя необходимость такого разбиения явно не прослеживается.

В Cerego не существует принципиального деления сущностей «концепты» и «модули». Модулем здесь выступает создаваемый элемент. При этом механизм формирования концепта выглядит просто: система выявляет текст в элементах некоторых типов (например, определяемое слово в элементе Vocabulary или текст вопроса в элементе Questing & Answer) и фиксирует его в качестве концепта. Управлять концептами при таком подходе затруднительно: отсутствует возможность произвести поиск по базе концептов, определить связанные с ними элементы, назначать одни и те же концепты разным модулям. Также в системе невозможно создать несколько элементов, которые бы были привязаны к одному и тому же концепту.

Технически элементы в Cerego не находятся в связи друг с другом. Порядком следования тем курса управляет исключительно разработчик. Соответственно, система не в состоянии отследить возможные зацикливания, нелогичную последовательность или отсутствие требуемых элементов или концептов.

В Cerego возможно повторное использование контента при создании разных дисциплин, однако реализация такой идеи на практике неудобна. Cerego позволяет объединять отдельные наборы (Set) в серии (Series), существует возможность объединения наборов разных предметных областей, и такая серия также может быть освоена как самостоятельный электронный курс. Однако объединить в серию можно лишь полный набор, а не его составные элементы. Соответственно, если рассматривать все элементы всех наборов как единую базу образовательного

контента, в Cerego невозможно управление, которое бы позволило создавать уникальные курсы из разных элементов.

Элементы в Cerego не содержат настройки, которые бы оказывали влияние на траекторию. Сложность контента определяется только по факту взаимодействия пользователя с ним и влияет лишь на частоту повторения материала, а не на построение траектории под индивидуальные потребности.

Создаваемые курсы не имеют ограничений по продолжительности, поскольку элементы не характеризуются временем, отведенным их на освоение. Соответственно, определить требуемый объем содержания модулей и их количество можно лишь эмпирически.

Сделаем вывод: Cerego основан на одном из известных и используемом в адаптивном обучении методе кривой забывания. Однако особенности технической реализации сервиса накладывают ограничения на его полноценное использование в учебном процессе. Пользовательские сценарии разработчика электронного курса в Cerego требуют автоматизации.

1.3.2 IADLearning

IADLearning [39] представляет собой веб-сервис для адаптивного обучения, основанный на модели рекомендаций.

Интерфейс главной страницы обучаемого представляет собой визуализированную иерархию содержания курса, которая предполагает переходы к соответствующему контенту (рис. 1.6).

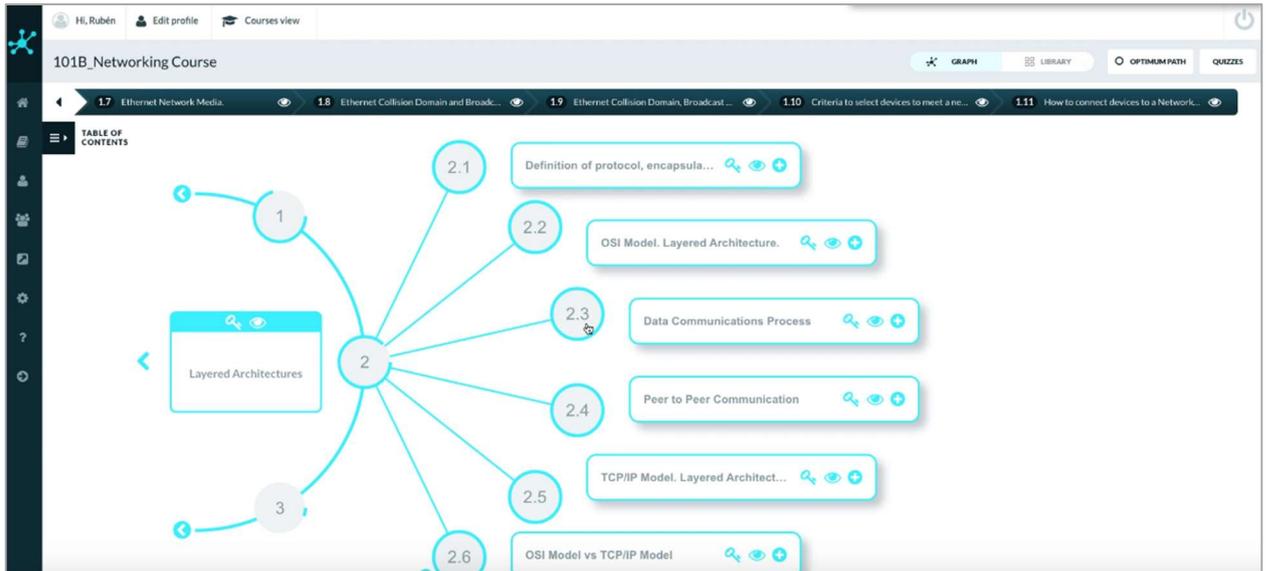


Рисунок 1.6 – Интерфейс обучаемого в среде IADLearning

Любой элемент курса может иметь ссылки на другие элементы, отличные от дочерних и родительских элементов. Также доступен альтернативный вариант представления содержания курса в виде блоков с предварительным просмотром контента (рис. 1.7).

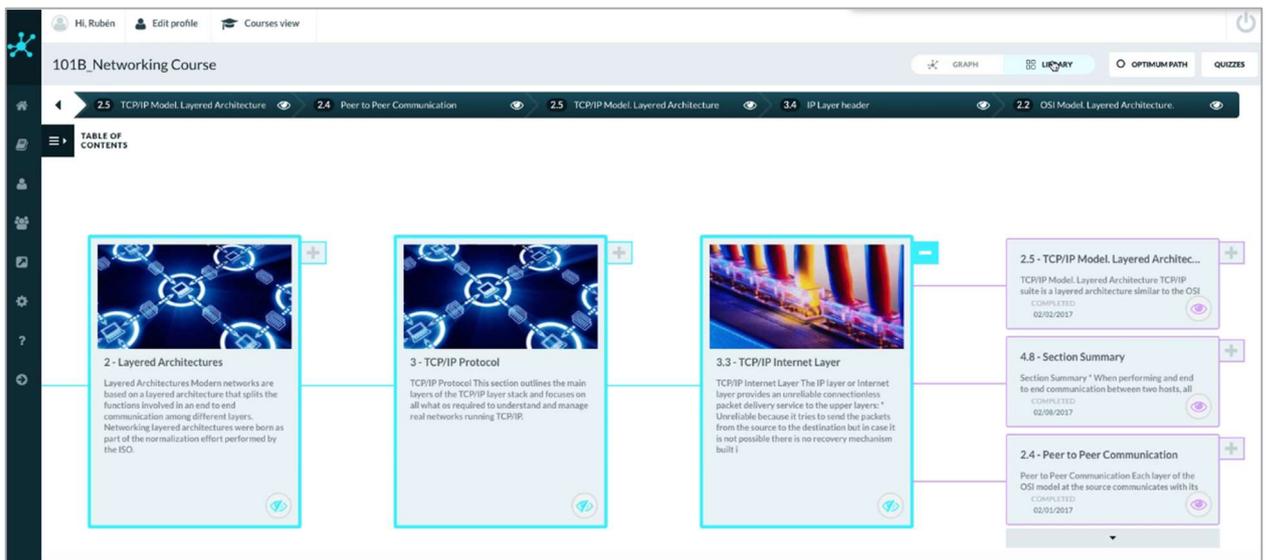


Рисунок 1.7 – Альтернативное представление содержания курса в среде IADLearning

Для перехода к контенту обучаемый должен перейти в желаемый модуль посредством экрана навигации. Контент модуля может быть представлен текстом,

видео или в любом другом формате, который поддерживает стандарт HTML (рис. 1.8).

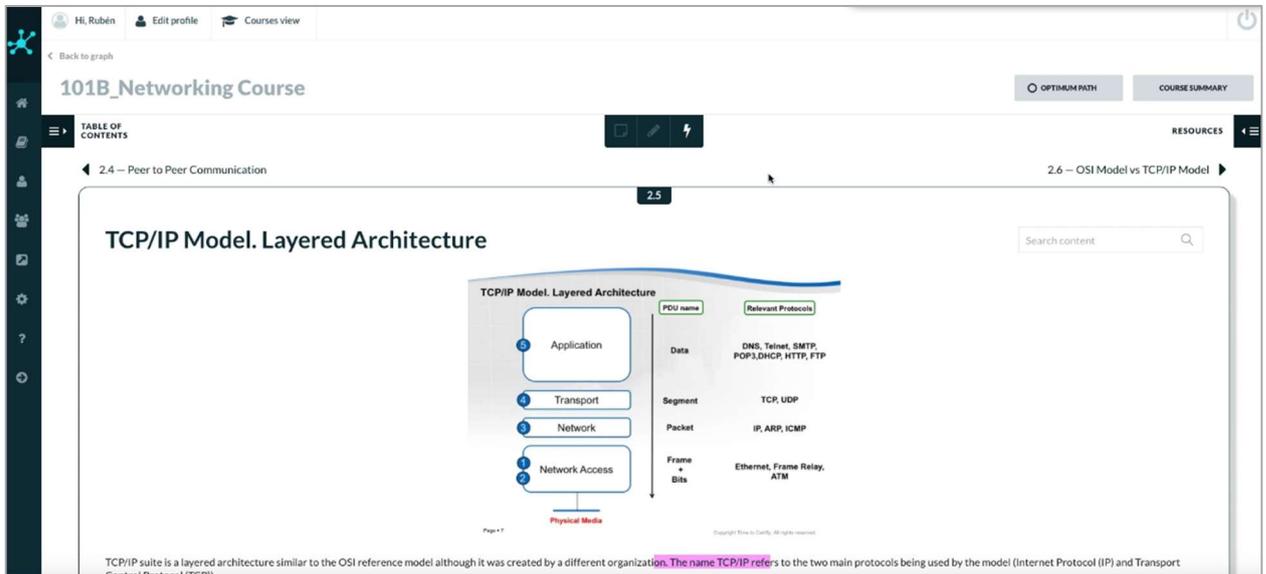


Рисунок 1.8 – Интерфейс обучаемого при работе с теоретическим контентом в среде IADLearning

IADLearning позволяет включить режим Optimum Path (оптимальный путь), который подразумевает инициализацию адаптивных рекомендаций по навигации. Алгоритмы системы производят вычисление рекомендаций, основываясь на профиле обучаемого, истории посещения разделов курса, а также времени, затраченном на освоение того или иного модуля. Рекомендация выводится в виде ветви содержания, которая имеет зеленую окраску. Рекомендованные к освоению элементы закрепляются в верхней части экрана и остаются в этом положении независимо от того, на каком уровне структуры курса находится обучаемый. Таким образом, рекомендованные элементы не ограничивают обучаемого в совершении перемещений по курсу, предоставляя ему возможность самостоятельно принимать решение относительно того, к какому разделу перейти.

Обучаемому также доступны к выполнению тестовые задания, однако они не привязаны к элементам контента. Также обучаемый имеет возможность в ходе работы оценивать контент и отслеживать историю прохождения курса.

При создании адаптивного курса платформа IADLearning предлагает несложный сценарий создания электронного курса, заключающийся в производстве общих настроек курса (рис. 1.9), в которых задаются метаданные (название, описание, сложность и продолжительность курса), далее производится загрузка контента в систему.

Рисунок 1.9 – Интерфейс редактора курса в IADLearning

IADLearning предлагает удобную функцию импортирования контента в систему: разработчик может загружать текстовые файлы, которые системой распознаются и преобразуются в формат системы с сохранением исходной структуры, которую также возможно в дальнейшем отредактировать. К недостаткам IADLearning можно отнести следующее:

1. Получаемый курс имеет статичную структуру, основанную на иерархии внутреннего содержимого загружаемого документа. Контент предварительно можно подготовить в виде модулей, однако получаемая таким образом траектория обучения не будет меняться в зависимости от динамического характера поведения показателей обучаемого.

2. Принцип навигации по содержанию курса на основе рекомендаций действительно можно рассматривать как адаптивное обучение, однако для

определенной модели, в основе которой лежат предиктивные алгоритмы, которые эффективны лишь на основе большой выборки данных, получаемых путем обучения множества обучаемых, проходящих одну и ту же траекторию. Должно пройти несколько запусков курса, прежде чем рекомендации начнут давать адекватный результат.

Сделаем вывод. Адаптивность в IADLearning проявляется в форме рекомендаций по дальнейшему следованию по маршруту обучения, в остальном пользователь может свободно передвигаться по статичной структуре курса. Для эффективной работы рекомендательных алгоритмов требуется наличие истории прохождения этого курса другими обучаемыми. При разработке контента учебным элементам не задаются какие-либо параметры, которые могли бы оказать влияние на формируемую траекторию обучения или рекомендательные алгоритмы. Автоматизация проверки знаний ограничена возможностями платформы. Связь тестовых заданий с теоретическим контентом не задается.

1.3.3 Smart Sparrow

Smart Sparrow [40] представляет собой программное решение для создания адаптивных электронных курсов, включающее инструментальные средства для производства контента с нуля.

Попадая в электронный курс, созданный средствами Smart Sparrow, пользователь видит экран с информацией о предстоящих активностях и подлежащих освоению разделах (рис. 1.10).

Обучаемому доступны лишь те разделы, которые были определены сценариями разработчика и алгоритмами Smart Sparrow. Раздел может быть представлен темами следующего уровня, при этом система делает рекомендации о том, к каким темам следует приступить в первую очередь (рис. 1.11).

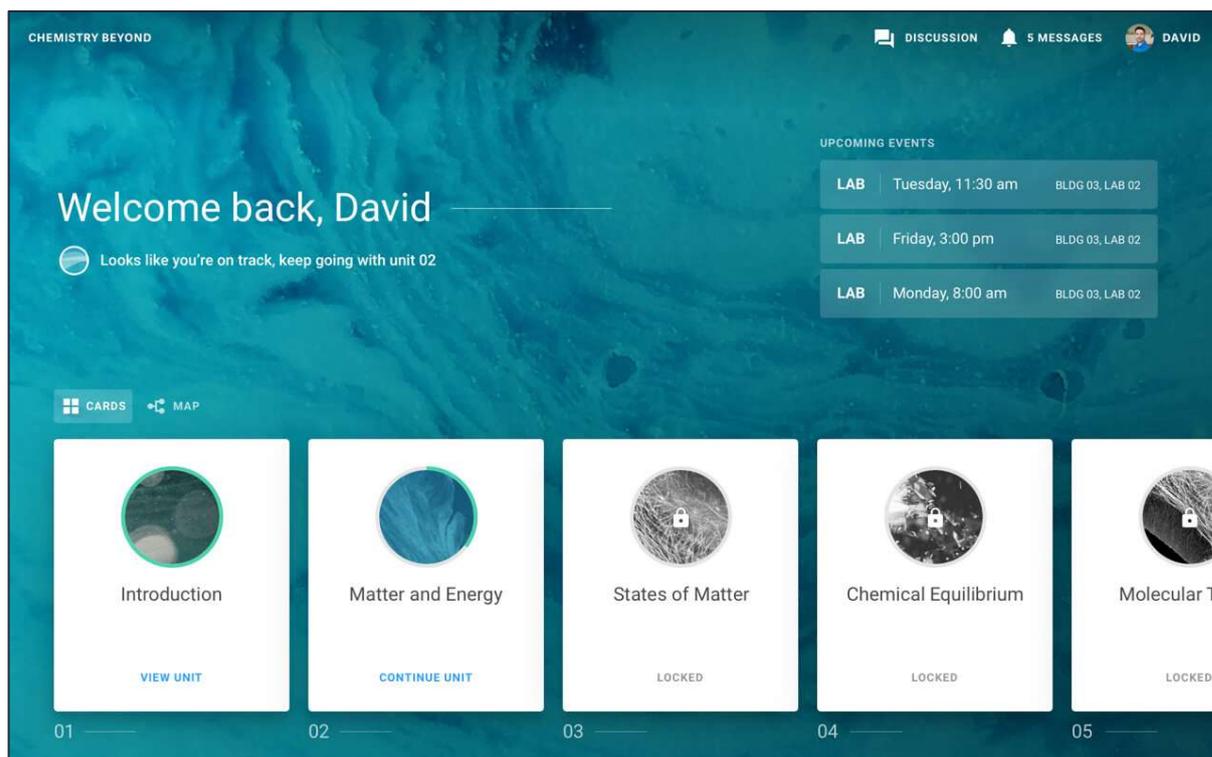


Рисунок 1.10 – Стартовый экран обучаемого в Smart Sparrow

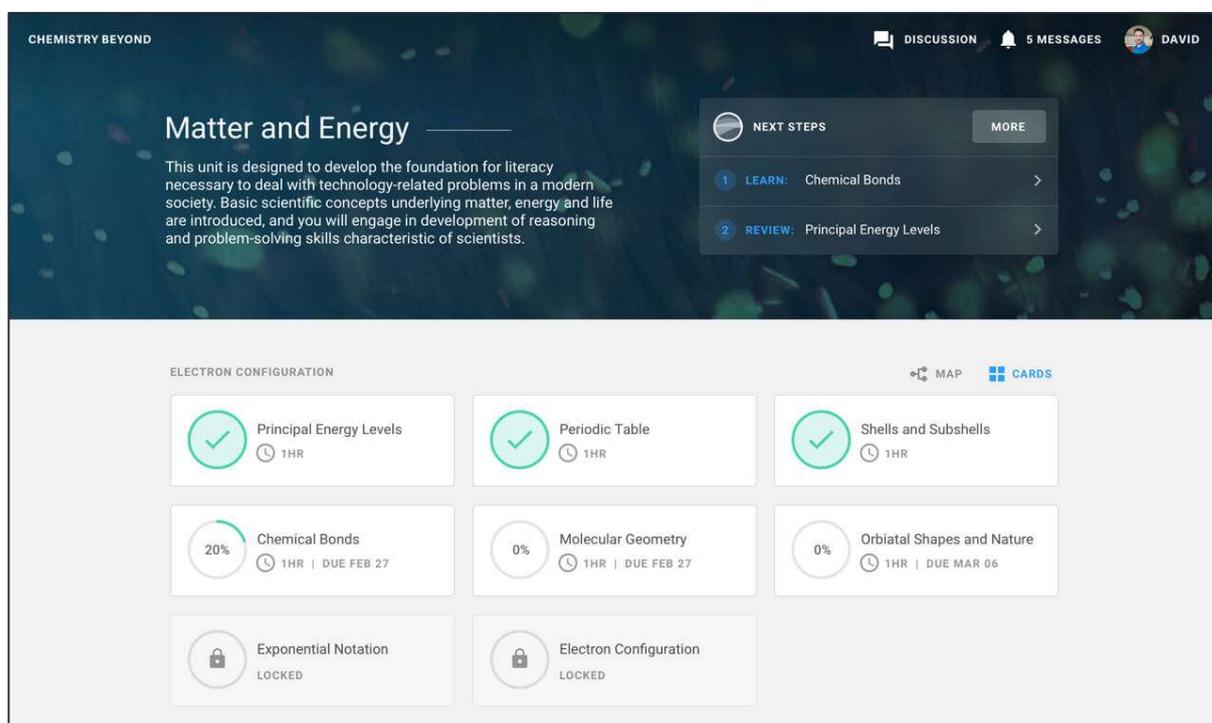


Рисунок 1.11 – Содержание раздела в Smart Sparrow представлено темами и сопровождается рекомендациями к прохождению (блок Next steps в правой верхней части экрана)

Процесс создания электронного курса заключается в создании учебных элементов, называемых «экранами». В качестве экрана может быть содержимое любого типа, включая комбинацию типов, для этого предусмотрен широкий перечень инструментов, позволяющих визуализировать учебную информацию различными способами (рис. 1.12). Экраны находятся в отношениях друг с другом. Эти отношения характеризуются правилами, которые реализуют переходы с одного экрана на другой. Правила настраиваются индивидуально для каждого экрана в виде условий. Это может быть переход в зависимости от ответа пользователя на вопрос на текущем экране, количества баллов, полученных на предыдущих этапах, зависимости от количества просмотренных экранов, времени нахождения на экране и т.д. При создании правил можно создать неограниченное количество условий, определяя при этом необходимость выполнения только одного из условий или всех одновременно. Экран может характеризоваться временем, отведенным на его освоение, иных свойств не предусмотрено.

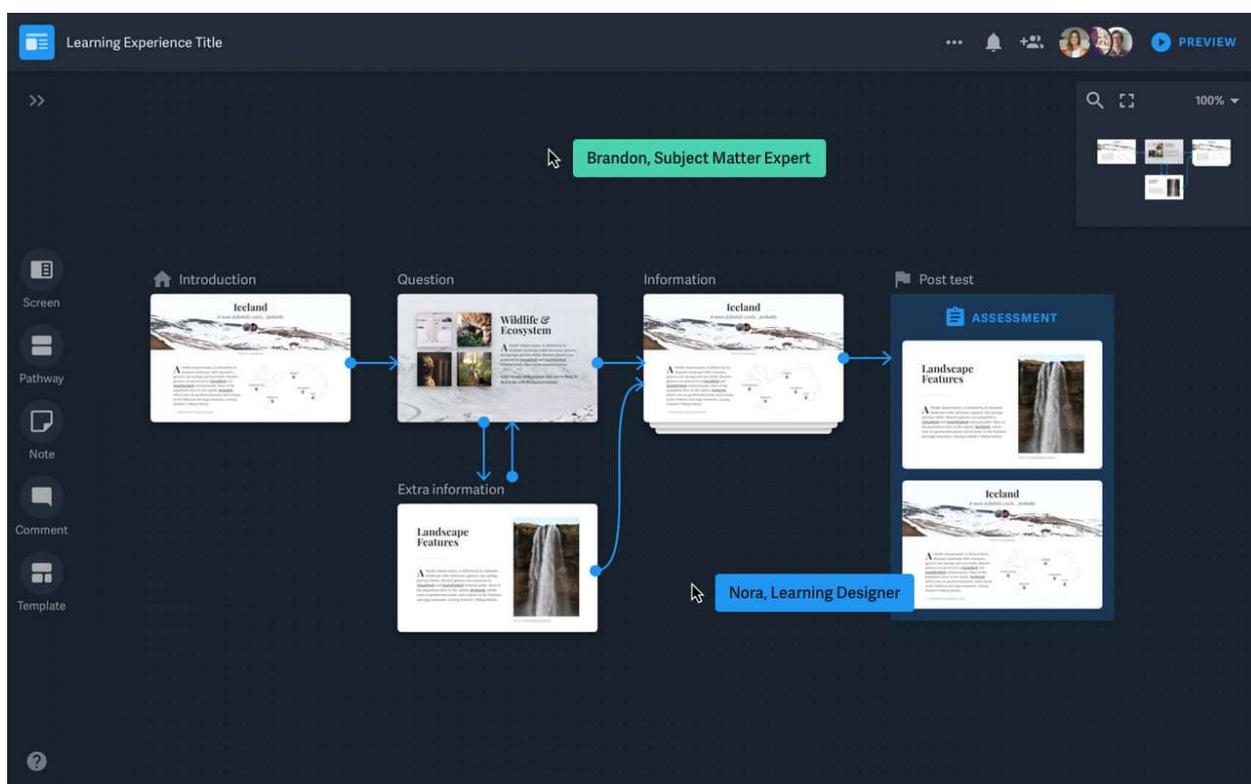


Рисунок 1.12 – Интерфейс редактора курса в Smart Sparrow

Как и другие современные системы, Smart Sparrow позволяет экспортировать созданные электронные курсы на внешнюю платформу через стандарт LTI.

В Smart Sparrow модули могут находиться в связи друг с другом, однако эти связи не имеют типов и не характеризуются входными и выходными субкомпетенциями. Единой базы контента в Smart Sparrow нет – каждый экран является элементом, доступным в рамках рабочей области разрабатываемого курса. Траектория обучения у каждого обучаемого может быть индивидуальной, однако алгоритмы автоматической корректировки траектории, основанные на изменении состояний модели обучаемого, не обнаружены – любые реакции системы на действия пользователя проектируются разработчиком курса вручную.

Выводы по главе 1

1. Анализ моделей контента показал, что в большинстве случаев они базируются на следующих принципах:

- в качестве элементарного носителя контента выступает модуль;
- модули находятся в отношениях между собой, эти отношения могут быть охарактеризованы определенными типами и метаданными;
- в основе структуры модуля может лежать формализованная структура представления информации (например, онтология), которая помимо описания непосредственно контента содержит метаданные, необходимые для работы адаптивных алгоритмов.

2. Анализ моделей обучаемого показал, что многие исследователи сходятся во мнении о необходимости моделирования перечня различных характеристик обучаемого, но в то же время возникают разногласия о том, какие характеристики принесут эффективность в адаптивные алгоритмы и как их измерять. Очевидно, что однозначно верного набора характеристик здесь не может быть, и в зависимости от задач проектируемой системы модель обучаемого может объединять в себе различные подходы [18, 28].

3. Анализ моделей адаптации показал, что системы адаптивного обучения, как правило, комбинируют в себе различные способы адаптации контента, при этом на практике широкое применение получила модель полного руководства и её метод под названием «*следующий наилучший*», подразумевающий, что система автоматически подбирает следующую порцию контента (следующий модуль) для изучения на основе ранее совершенных действий обучаемого.

4. Анализ выбранных программных решений позволяет сделать заключения:

- во всех программных средствах не поддерживается возможность использовать одни и те же модули в разных курсах;

- система управления контентом отсутствует, что затрудняет осуществление навигации по базе созданных модулей и их поиск;

- в большинстве случаев проверка остаточного уровня знаний обучаемого не автоматизирована и требует от разработчика создания сценариев тестирования;

- средства измерения знаний, учебные модули и модель обучаемого имеют слабые связи между собой либо не имеют их вовсе, что характеризует уровень автоматизации адаптационных механик как низкий;

- траектории задаются жестко и не вычисляются заново на основе актуализации модели обучаемого;

- на рынке программного обеспечения, предназначенного для создания адаптивных средств обучения, не существует решений, локализованных для отечественных потребителей. Специализированного инструментального ПО российского происхождения также не обнаружено. Широко распространенные системы дистанционного обучения не обладают механизмами адаптивного обучения [42].

5. Большинство существующих систем и инструментальных средств (в том числе и рассмотренные) ориентированы на реализацию скорее персонализированного, а не адаптивного обучения. Исчерпывающее пояснение отличий персонализированного обучения от адаптивного представлено в [43].

Таким образом, подтверждается актуальность проблемы, связанной с отсутствием решений для реализации технологии адаптивного обучения посредством инструментальных средств авторской разработки.

2 МОДЕЛИ И АЛГОРИТМЫ АДАПТИВНОГО ОБУЧЕНИЯ

2.1 Модель предметной области и модель обучаемого

Введем обозначения модели предметной области и модели обучаемого, необходимые для дальнейшего описания алгоритма адаптивного обучения.

1. Множество компетенций курса $K = \{K_j\}$, $j = 1, 2, \dots, m$, где m – количество компетенций (ранее было введено понятие *субкомпетенции*, однако для краткости предлагается называть их просто компетенциями).

2. Множество образовательных модулей курса $M = \{M_i\}$, $i = 1, 2, \dots, n$, где n – количество модулей. Модуль описывается четверкой $M_i = (TM_i, KI_i, KO_i, RO_i)$, $i = 1, 2, \dots, n$, где TM_i – время изучения модуля, KI_i и KO_i – списки входных ($KI_i = \{comp_k\}$, $comp_k \in [1, m]$, $k = 1, 2, \dots, ni$) и выходных ($KO_i = \{comp_k\}$, $comp_k \in [1, m]$, $k = 1, 2, \dots, no$) компетенций, RO_i – получаемый уровень знаний для каждой выходной компетенции.

3. Множество групп тестов $T = \{T_j\} = \{\{T_{jk}\}\}$, $j = 1, 2, \dots, m$, $k = 1, 2, \dots, nt_j$, предназначенных для оценки знаний компетенций K . При этом группа тестов T_j предназначена для проверки знаний по компетенции K_j и может включать один или множество тестов T_{jk} .

4. Время начала курса $t_{нач}$, время окончания курса $t_{кон}$, текущее время $t_{тек}$.

5. Модель обучаемого $S = (HM, HR, RK)$. Здесь компонент HM – это личная история изучения студентом модулей, она содержит идентификаторы модулей и время их изучения, $HM = \{(mod_k, th_k)\}$, $mod_k \in [1, n]$, $k = 1, 2, \dots, nh$. Компонент HR – это история изменения уровня знаний по каждой компетенции, $HR = \{HR_j\} = \{\{(RA_{jk}, tr_{jk})\}\}$, $j = 1, 2, \dots, m$, $k = 1, 2, \dots, nr_j$, где RA_{jk} – актуальный уровень знаний студентом компетенции K_j на момент времени tr_{jk} . Если

множество $HR_j = \emptyset$, т.е. компетенция K_j еще ни разу не была изучена, то полагаем $RA_{j0} = 0$. И, наконец, компонент RK – это коэффициенты кривой забывания для различного количества повторений студентом учебного материала, $RK = \{(k_r, c_r)\}$, где коэффициенты k_r и c_r соответствуют количеству повторений r , $r = 1, 2, \dots$

6. Уровень удовлетворительного усвоения учебного материала R_{norm} .

7. Множества достаточно (KS) и недостаточно (KF) изученных компетенций. Если уровень RA_{jl} усвоения знаний по какой-либо компетенции K_j (причем $l = nr_j$, т.е. берем последний элемент множества HR_j) ниже R_{norm} , то она считается недостаточно изученной, в противном случае – достаточно изученной. То есть, в терминах исчисления предикатов:

$$KS = \{comp_k \mid RA_{comp_k l} \geq R_{norm}\}, \quad comp_k \in [1, m], \quad k = 1, 2, \dots, ns, \quad (1)$$

$$KF = \{comp_k \mid RA_{comp_k l} < R_{norm}\}, \quad comp_k \in [1, m], \quad k = 1, 2, \dots, nf. \quad (2)$$

8. Множество модулей $A = \{mod_k\}$, $mod_k \in [1, n]$, $k = 1, 2, \dots, na$, реализующих недостаточно изученные компетенции, т.е.

$$\bigcup_{i \in A} KO_i \subseteq KF, \quad (3)$$

но при этом

$$KO_{mod_k} \not\subseteq KS, \quad k = 1, 2, \dots, na. \quad (4)$$

Последнее условие означает, что множество A не включает модули, все выходные компетенции которых уже изучены на достаточном уровне.

9. $G = (V, C)$ – двудольный граф, образующийся из вершин V и множества связей между вершинами C , где $V = \{M, K\}$. Модуль M_i может иметь несколько как входных, так и выходных компетенций. Очевидно, что некоторая K_j будет являться одновременно входной компетенцией для одного модуля и выходной для другого.

10. Траектория обучения студента представляет собой ориентированный граф-путь P графа G , который определяет последовательность прохождения

образовательных модулей. Поскольку набор модулей, который обучаемый должен освоить, меняется по мере того, как обучаемый движется по траектории, имеет смысл рассматривать путь P как непротиворечивый порядок изучения модулей множества A , т.е. $P = \{mod_k\}$, $mod_k \in [1, n]$, $k = 1, 2, \dots, np$ (не обязательно всех). В результате изучения модулей, входящих в P , должен быть повышен уровень знаний по всем компетенциям, для которых $RA_{jl} < R_{norm}$. Требование непротиворечивости означает, что к моменту изучения каждого модуля из P должны быть изучены все требуемые входные компетенции, т.е.

$$KS \cup \bigcup_{i < k} KO_{mod_i} \subseteq KI_{mod_k}, k = 1, 2, \dots, np. \quad (5)$$

2.2 Модель забывания информации

Для экстраполяции уровня остаточных знаний на момент окончания курса, отталкиваясь от промежуточных результатов тестирования, была использована модель, основанная на скорости забывания информации.

Других достаточно обоснованных моделей, пригодных для численного прогнозирования на практике уровня знаний обучающихся в будущие периоды времени на основе их предыдущей истории обучения, практически не существует. Использование байесовских сетей не дает более высокой точности прогнозирования (за пределами задач адаптивного тестирования), зато требует существенных вычислительных ресурсов [45]. Хорошую скорость принятия решений дает использование технологии машинного обучения (как, например, в *Snappet*¹), но для достижения достаточной точности прогнозов в учебном курсе, содержащем хотя бы 150–200 модулей, потребуется база из десятков тысяч уже пройденных траекторий обучения. Поэтому на начальном этапе внедрения учебного курса необходимо использовать статистические модели (например,

¹ <https://nl.snappet.org/>

байесовские сети доверия) или модели, основанные на скорости забывания информации (например, [46]).

Первое экспериментальное исследование памяти и процессов забывания выполнил в конце XIX в. немецкий психолог Герман Эббингауз. Его результаты были опубликованы в 1885 г. в книге «О памяти» [47]. Эббингауз показал, что скорость забывания удовлетворительно аппроксимируется функцией

$$R(t) = \frac{k}{\lg t + c}, \quad (6)$$

где нужно положить $k = 1,84$ и $c = 1,25$, если время t выражено в минутах [48]. Эти значения были получены для запоминания информации, лишенной ассоциативной связи.

Примечание. Уровень знаний не может быть ниже 0 и выше 100%, поэтому должно выполняться условие $R(t) \in [0,1]$. Выражение (6) при достаточно больших значениях t дает только положительные значения, т.к.

$$\lim_{t \rightarrow \infty} \frac{k}{\lg t + c} = 0.$$

Однако при малых значениях t выражение плохо обусловлено и может давать произвольный результат от $+\infty$ до $-\infty$. На практике для вычисления (6) предлагается использовать следующий подход:

$$R(t) = \begin{cases} 1, & t < 1; \\ \min\left(\frac{k}{\lg t + c}, 1\right), & t \geq 1. \end{cases} \quad (7)$$

Таким образом, $R(t)=1$ при $\lg t + c \leq k$. Это можно интерпретировать следующим образом: в течение некоторого времени после изучения информация не забывается, т.е. полностью сохраняется в памяти. Это время тем больше, чем больше коэффициент k .

Имея две экспериментальные точки на кривой забывания, можно определить значения коэффициентов k и c :

$$\begin{cases} R_1 = \frac{k}{\lg t_1 + c}, \\ R_2 = \frac{k}{\lg t_2 + c}, \end{cases}$$

откуда

$$c = \frac{R_2 \lg t_2 - R_1 \lg t_1}{R_1 - R_2}, \quad (8)$$

$$k = R_1 (\lg t_1 + c) \text{ или } k = R_2 (\lg t_2 + c). \quad (9)$$

Более поздние исследования Эббингауза, произведенные для изучения процесса забывания осмысленной информации, дали следующий процент воспроизведений: спустя 1 день – 75%, через 4 дня – 70%. Последние результаты имеет смысл взять за основу, т.к. учебные материалы относятся к осмысленной информации.

Учитывая, что сутки состоят из 1440 минут, из (8) и (9) получим:

$$c = \frac{0,7 \cdot \lg 5760 - 0,75 \cdot \lg 1440}{0,75 - 0,7} \approx 5,270,$$

$$k = 0,75 \cdot (\lg 1440 + 5,270) \approx 6,322,$$

что можно принять за значения данных коэффициентов по умолчанию, уточняя их по мере тестирования уровня остаточных знаний студента.

В данном примере был рассмотрен обобщенный сценарий процесса освоения информации студентом. Очевидно, что приобретение и забывание информации – сложные умственные процессы, результат влияния на которые в настоящее время невозможно спрогнозировать с предельной точностью. Однако в целом неоспоримым является эффект, которого можно достичь за счет итеративного научения [70], а именно: зафиксировать знания обучаемого к концу обучения на требуемом уровне. Так, в работе [49] авторы рассматривают разработку вероятностной модели принципа повторений и выводят формулу обобщенной функции забывания, сформированной в результате произвольного числа циклов обучения. Вид этой зависимости представлен на рисунке 2.1.

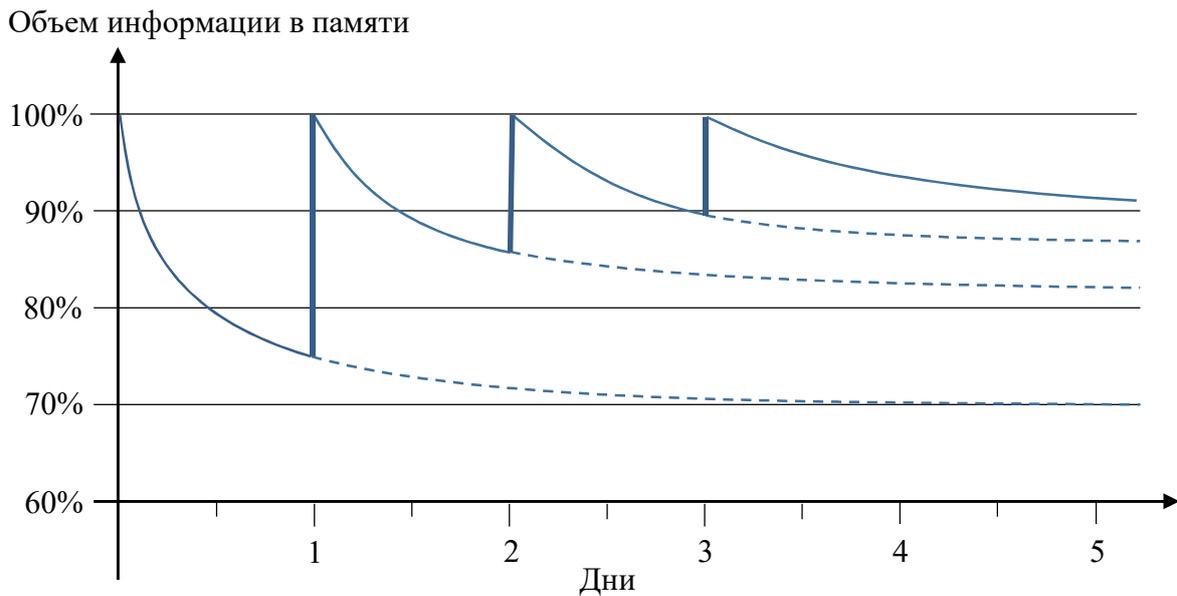


Рисунок 2.1 – Кривая забывания при итеративном обучении

Таким образом, при повторном изучении информации объем остаточных знаний на момент окончания курса увеличится пропорционально некоторому коэффициенту α :

$$R(t, r) = R(t) \cdot \alpha(r), \quad (10)$$

где r – количество повторений.

Вид зависимости $\alpha(r)$ неизвестен, кроме того, она будет индивидуальной для каждого студента. Предположим, что она имеет вид:

$$\alpha(r) = \frac{1,5}{1 + \exp\left(-\frac{r}{1,5}\right)},$$

т.е. в пределе, при достаточно большом количестве повторений значение $\alpha(r)$ стремится к 1,5 (как уже было сказано, данные допущения будут использованы только для вычисления значений коэффициентов зависимости $R(t, r)$ по умолчанию, а в дальнейшем они будут уточняться на основе фактических результатов обучения студента).

Определим параметры зависимости $R(t, r)$. Предположим, что через минимальное время $t=1$ мин уровень знаний студента сохраняется на уровне 100%, тогда из (6) и (10) имеем:

$$R(1, r) = R(1) \cdot 1 = \frac{k(r)}{\lg 1 + c(r)} = 1 \Rightarrow k(r) = c(r). \quad (11)$$

На момент окончания курса

$$R(t_{\text{кон}}, r) = \frac{k(r)}{\lg t_{\text{кон}} + c(r)} = R(t_{\text{кон}}) \cdot \alpha(r).$$

С учетом (11) получим:

$$k(r) = c(r) = \frac{R(t_{\text{кон}}) \cdot \alpha(r)}{1 - R(t_{\text{кон}}) \cdot \alpha(r)} \cdot \lg t_{\text{кон}}.$$

Таким образом, окончательно получим:

$$R(t, r) = \begin{cases} R(t), & r = 1, \\ \frac{k(r)}{\lg t + c(r)}, & r > 1. \end{cases} \quad (12)$$

При вычислении данного выражения также необходимо использовать подход (7). Вид кривой забывания для разного количества повторений учебного материала r приведен на рисунке 2.2.

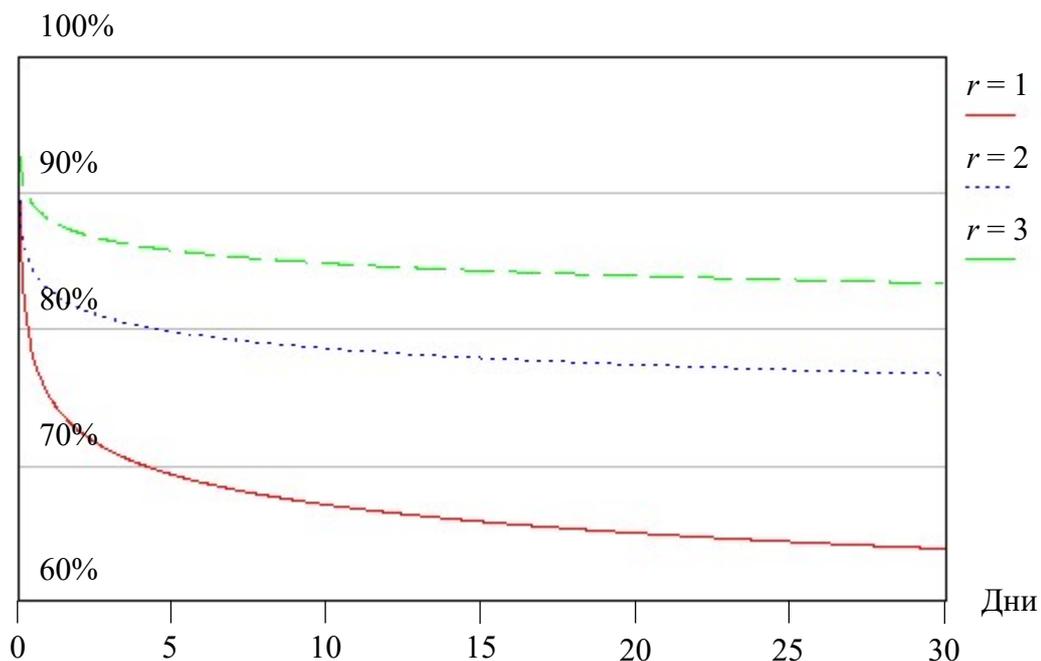


Рисунок 2.2 – Поведение кривой забывания при разных значениях параметра r

Эта формула, однако, была получена исходя из предположения, что при каждом повторении материала студент восстанавливает уровень знаний до 100%. На практике это не так, и последующее тестирование может показать меньший результат. Например, на рисунке 2.3 рассматривается ситуация, когда студент после первой итерации обучения выполнил выходной тест с результатом 100%, а после второй итерации – с результатом 80%. К этому моменту убывание уровня знаний $R(t, 1)$ уже замедлилось, и в дальнейшем кривая убывает уже не так существенно (средняя кривая), но новая кривая $R(t, 2)$ вначале убывает с высокой скоростью, и получается, что повторное изучение материала привело к тому, что студент стал забывать его быстрее (нижняя кривая). Поэтому формула (12) при практической реализации будет применяться не ко всему объему полученных знаний, а только к новым полученным знаниям. Так, для ситуации на рисунке 2.3 на момент повторного изучения материала прогнозируемый объем остаточных знаний составлял 75%. Повторное изучение показало, что реально эти знания находятся на уровне 80%, поэтому далее кривая будет вычисляться по формуле

$$R = 100\% \cdot R(t - t_1, 1) + (80\% - 75\%) \cdot R(t - t_2, 2),$$

где t_1 и t_2 – время первого и повторного изучения материала соответственно (верхняя кривая).

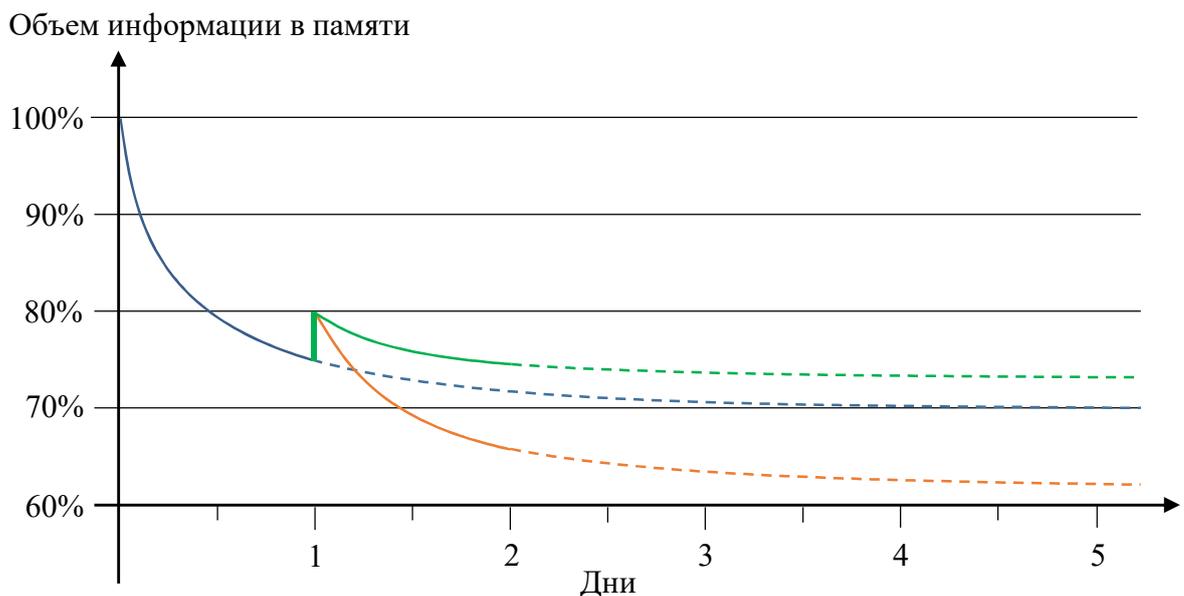


Рисунок 2.3 – Вычисление уровня остаточных знаний при повторении материала

2.3 Постановка задачи

Основная идея адаптивного обучения заключается в построении оптимальной траектории изучения модулей курса. Рассмотрим подробнее, в чем состоит эта задача.

Пусть задано множество компетенций K , необходимых для освоения обучаемым, а также множество модулей M , реализующих K . При этом одну и ту же произвольную компетенцию K_j может реализовывать более чем один модуль. У каждого модуля имеется как минимум одна выходная компетенция и ноль или более входных. За счет того что одна и та же компетенция для одного модуля является входной, а для другой выходной, множество модулей и компетенций образуют двудольный граф $G=(V,C)$, состоящий из вершин модулей и вершин компетенций $V=\{M,K\}$, а также множества связей C между ними. Пример такого графа представлен на рисунке 2.4.

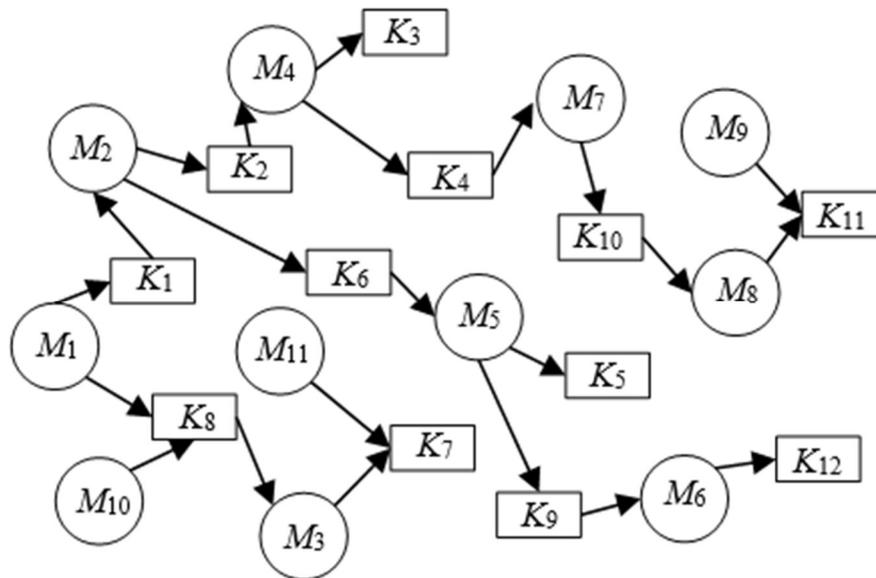


Рисунок 2.4 – Пример графа учебных модулей

Пусть задано время t , отведенное на обучение. При освоении модуля M_i обучаемый приобретает знание по соответствующей этому модулю выходной компетенции из набора K , измеряемое по условной шкале от 0 до 100%. Пусть R – суммарное значение уровней остаточных знаний всех компетенций на момент

окончания обучения. Задача системы – найти такой путь P на графе G , чтобы на момент окончания курса R имело максимальное значение, т.е.

$$R(P, t) \rightarrow \max. \quad (13)$$

Ограничения задачи (13):

1. Должно выполняться условие непротиворечивости прохождения модулей (5). Например, на рисунке 2.4 допустимым фрагментом пути будет являться последовательность $M_1 \rightarrow M_2$, но не наоборот. При этом возможной и допустимой также является последовательность $M_1 \rightarrow M_{11} \rightarrow M_2$, несмотря на то, что прямая связь между M_1 и M_{11} не задана.

2. При завершении освоения обучаемым модуля и переходе к следующему, к только что приобретенным компетенциям применяется закономерность (6), т. е. с течением времени уровень знаний начинает спадать. Согласно концепции итеративного обучения, один и тот же модуль может участвовать в P более чем один раз. Поскольку модуль характеризуется временем, отведенным на его изучение, то порядок модулей разной продолжительности (с учетом условия непротиворечивости прохождения), а также повторное включение в P ранее изученных модулей оказывают прямое влияние на R к моменту окончания обучения.

Пример. Имеются модули M_1 , M_2 и M_3 продолжительностью 5, 30 и 10 мин соответственно, которые реализуют компетенции K_1 , K_2 и K_3 . Продолжительность обучения задана как 90 мин. Условимся, что любая последовательность модулей допустима. На рисунках 2.5 и 2.6 приведены кривые забывания для возможных последовательностей: $M_1 \rightarrow M_2 \rightarrow M_3$ и $M_2 \rightarrow M_3 \rightarrow M_1$ соответственно.

На рисунке 2.7 приведены кривые забывания при итеративном обучении. В P были включены в качестве повторного изучения модули M_1 и M_3 , в результате чего путь стал представлять собой последовательность: $M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_1 \rightarrow M_3$.

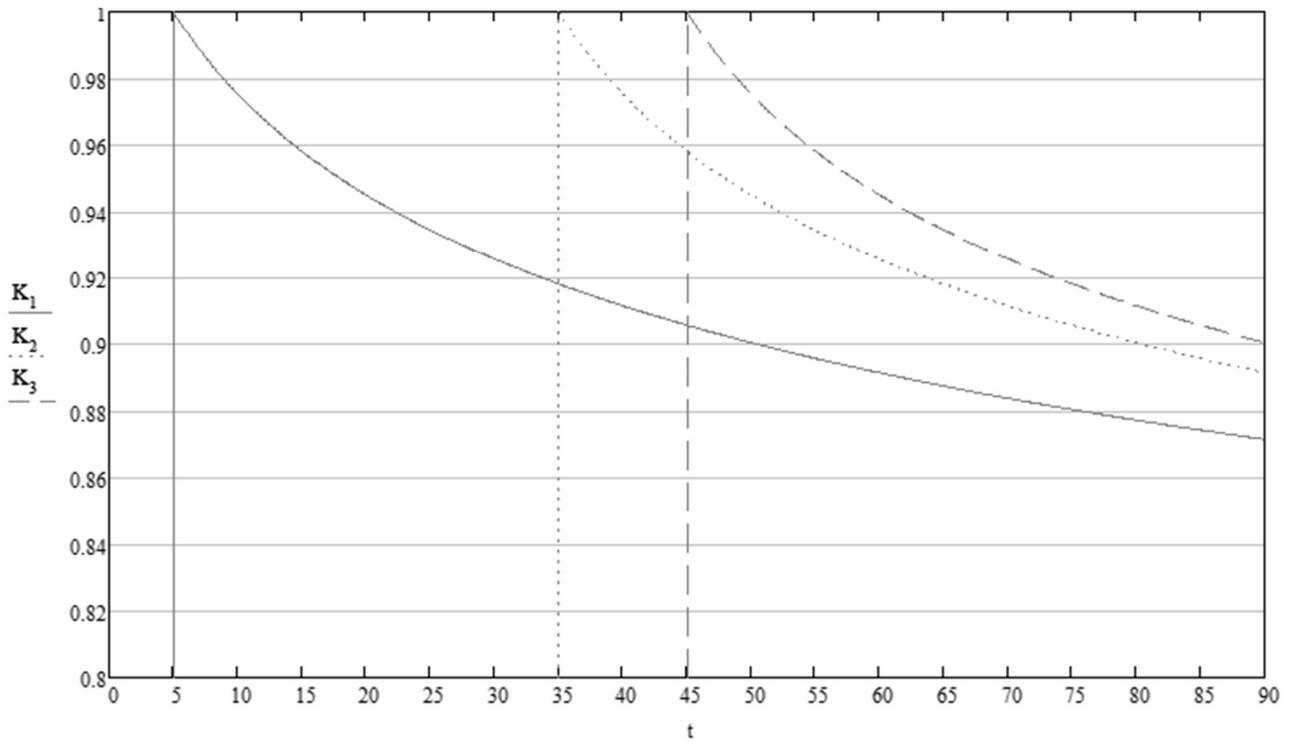


Рисунок 2.5 – Кривые забывания компетенций для последовательности $M_1 \rightarrow M_2 \rightarrow M_3$

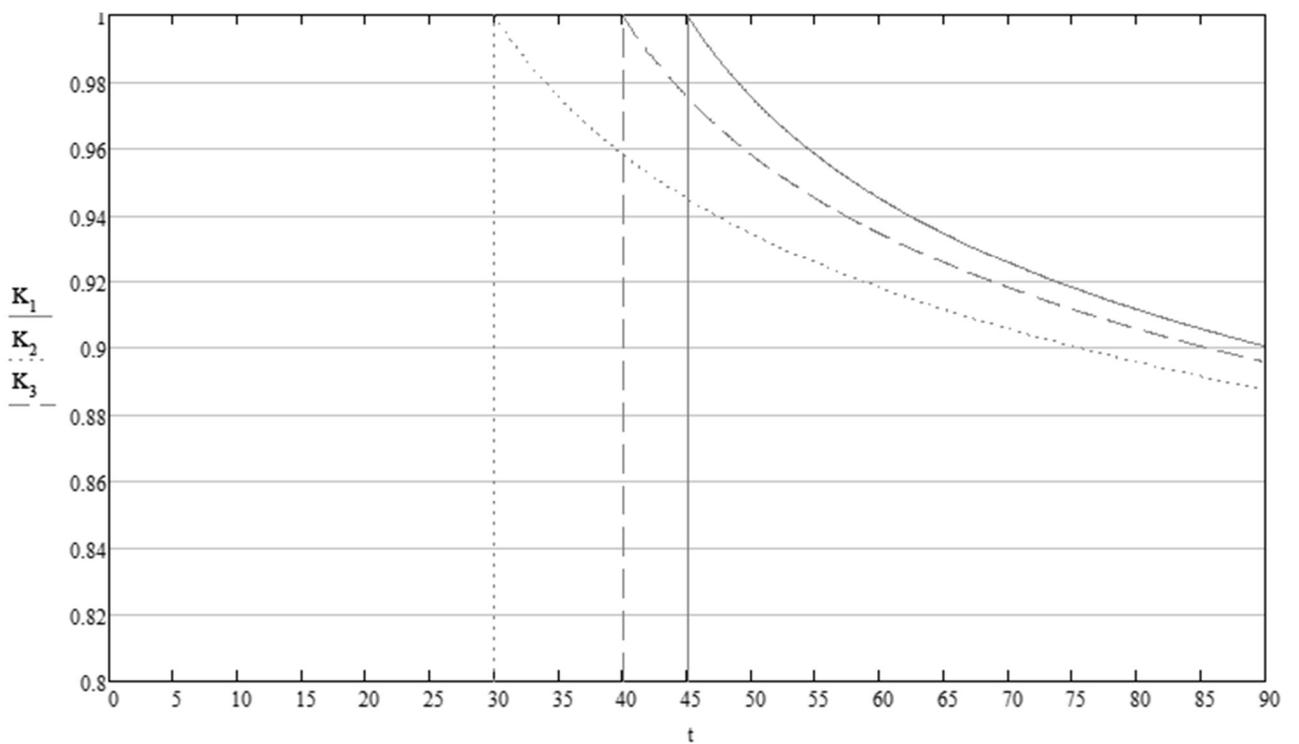


Рисунок 2.6 – Кривые забывания компетенций для последовательности $M_2 \rightarrow M_3 \rightarrow M_1$

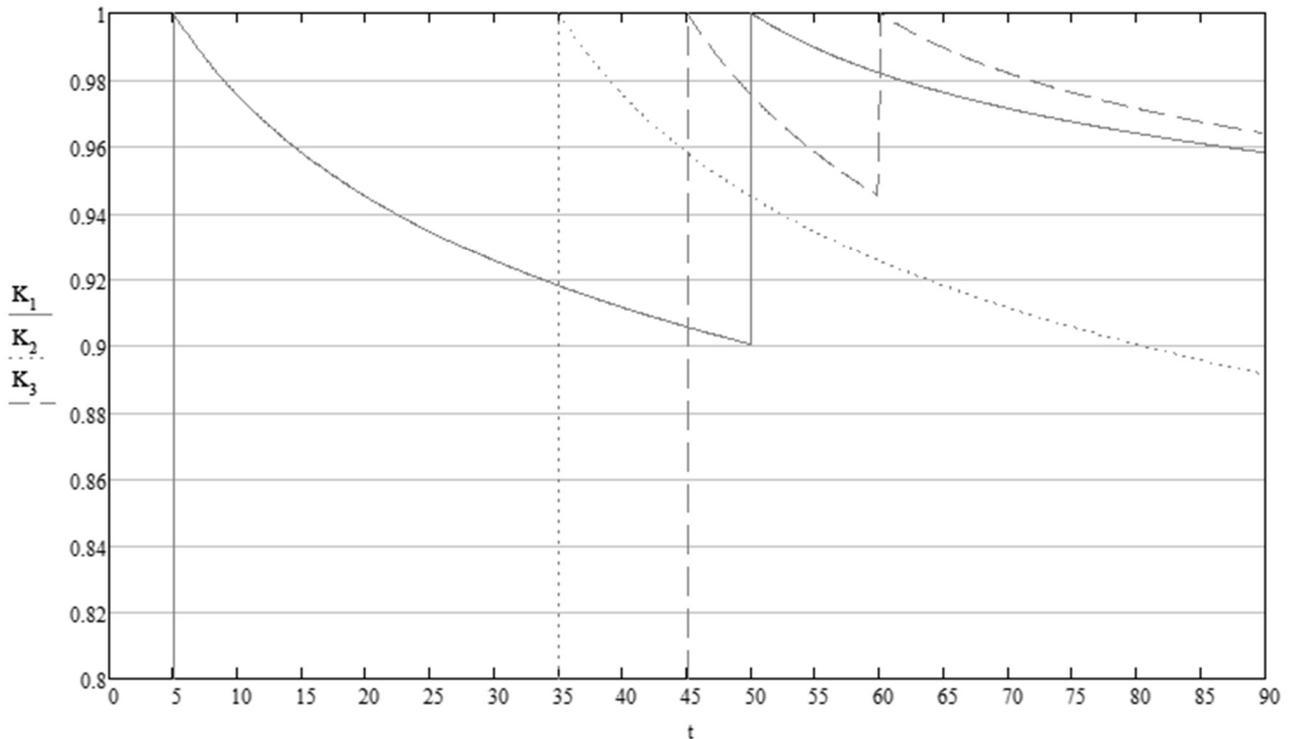


Рисунок 2.7 – Кривые забывания компетенций для последовательности

$$M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_1 \rightarrow M_3$$

Из рисунков видно, что при разной последовательности модулей, а также при включении в путь ранее изученных модулей для повторения значение R будет разным.

3. Одна и та же компетенция может быть выходной у более чем одного модуля в наборе M , при этом для однократного освоения обучаемому достаточно предъявить только один модуль из существующих альтернатив.

4. Суммарное время прохождения траектории (всех модулей, с учетом их повторного предъявления) не должно превышать время, отведенное на освоение курса.

Данную оптимизационную задачу на графе сложно свести к классической задаче оптимизации для того, чтобы выбрать соответствующие методы решения. Кроме того, пространство решений задачи является дискретным. В связи с этим целесообразным является применение генетического алгоритма [57], преимущество которого состоит в том, что за относительно короткое время он находит приближительные оптимальные решения [61, 66].

2.4 Алгоритм адаптивного обучения

Общая блок-схема алгоритма адаптивного обучения приведена на рисунке 2.8 [16]. В качестве модели навигационной поддержки будет взята модель «полное руководство», как наиболее распространенная в системах адаптивного обучения [18]. Это значит, что система будет решать за пользователя, какой модуль предъявить к освоению на каждом шаге прохождения электронного курса. Свободная навигация по базе модулей пользователю недоступна.

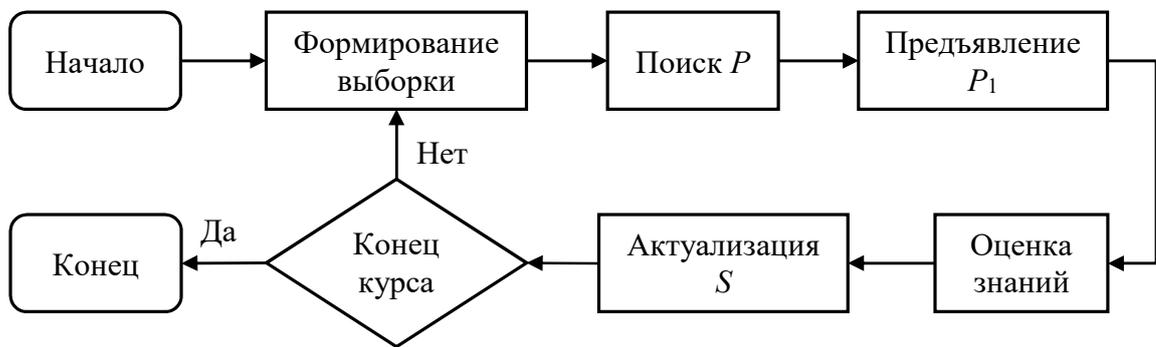


Рисунок 2.8 – Общая блок-схема алгоритма адаптивного обучения

Приведем описание алгоритма адаптивного обучения.

1. *Формирование выборки.* На данном этапе формируется множество A , состоящее из модулей, реализующих недостаточно изученные компетенции, т.е. из модулей, удовлетворяющих условиям (3) и (4). Компетенция K_j считается недостаточно изученной в двух случаях. Во-первых, если ранее она студентом вообще не изучалась, т.е. $HR_j = \emptyset$. Во-вторых, если компетенция является *утраченной*, т.е. уровень ее освоения, согласно кривой забывания, со временем опустился ниже уровня R_{norm} .

2. *Поиск P.* Для поиска траектории обучения используется генетический алгоритм, описанный ниже.

3. *Предъявление P_1 .* Обучаемому предоставляется для обучения первый модуль из P .

4. *Оценка знаний.* На данном этапе обучаемому предоставляется тест для проверки уровня знаний по выходным компетенциям модуля.

5. *Актуализация S*. После проведения теста обновляется актуальный уровень знаний студента в истории HR_j . При этом обновляется компонент модели обучаемого RK на основе результатов тестирования.

6. Проверка на окончание курса. Окончание курса наступает в двух случаях. Во-первых, при истечении времени курса, т.е. если $t_{тек} \geq t_{кон}$. Во-вторых, если все компетенции изучены на удовлетворительном уровне, т.е. $KS = K$, $KF = \emptyset$ (см. (1), (2)).

Описание генетического алгоритма. Базовые положения генетического алгоритма генерации последовательности образовательных модулей изложены в [44]. Его общая блок-схема изображена на рисунке 2.9.

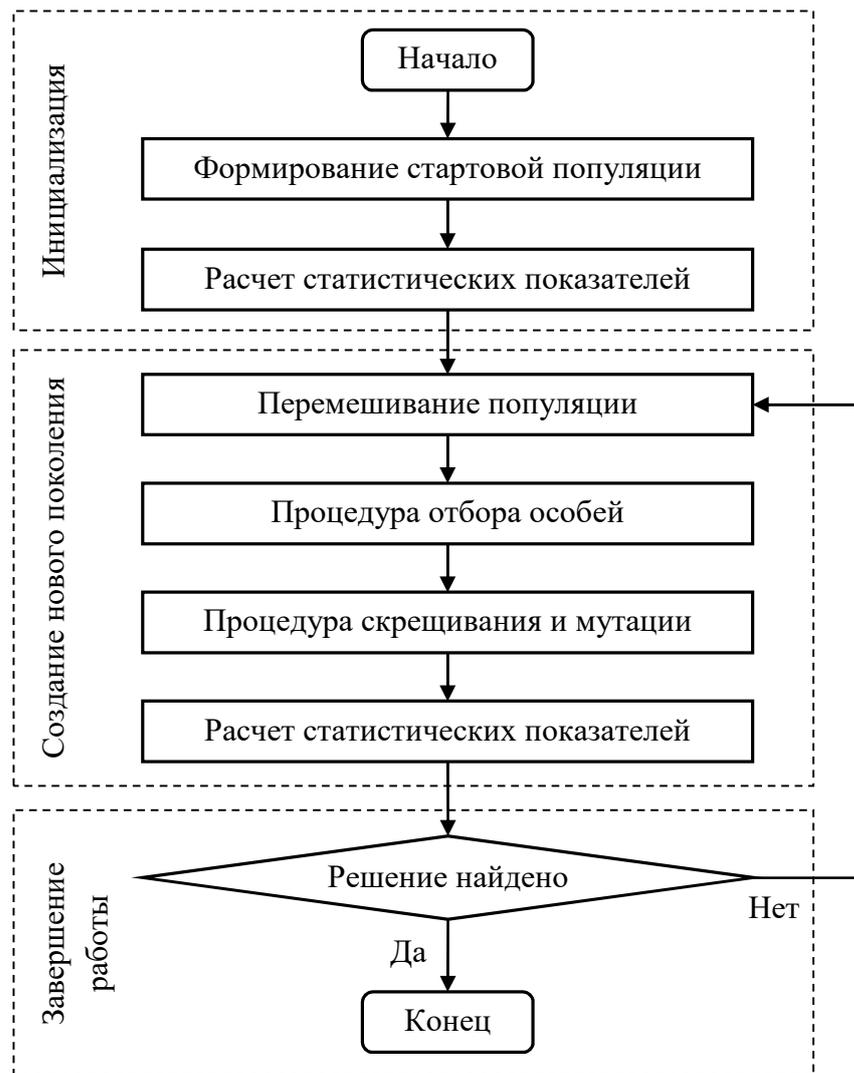


Рисунок 2.9 – Общая блок-схема генетического алгоритма

В классическом генетическом алгоритме популяция описывается как совокупность особей, каждая из которых представляет возможное решение задачи. У каждой особи есть набор хромосом, определяющих ее фенотип. При этом хромосомы обычно представляют собой битовый массив, которым относительно легко можно манипулировать при проведении процедур скрещивания, мутации и т.п. При решении задачи генерации адаптивного контента фенотип особи должен определять траекторию обучения P , т.е. порядок изучения модулей из выборки A . Закодировать его в виде битового массива не представляется возможным, т.к. не любая комбинация битов будет соответствовать допустимому порядку следования модулей. Поэтому в реализованном варианте генетического алгоритма хромосомы описывают упорядоченные цепочки модулей, что повлекло за собой его существенное усложнение.

Рассмотрим работу процедур, составляющих генетический алгоритм. Параметры алгоритма:

- размер популяции n_{pop} (выбран равным 100 особям);
- максимальное число поколений n_{gen} (выбрано значение 100);
- вероятность мутации $p_{mutation}$ (1%);
- вероятность скрещивания p_{cross} (90%);
- вероятность появления элитной особи p_{elite} (5%).

1. Формирование стартовой популяции. Генерируем n_{pop} особей. Каждая особь стартовой популяции генерируется однотипно.

2. Расчет статистических показателей. Заключается в выполнении двух шагов. Во-первых, здесь нужно для каждой особи оценить функцию пригодности (или целевую функцию (ЦФ)), которая запишется как:

$$F(P) = \frac{\sum_i TM_{P_i}}{\sum_j R_j} \rightarrow \min,$$

где $i = 1, 2, \dots, np$, $j = 1, 2, \dots, m$. С одной стороны, при минимизации ЦФ $F(P)$ мы пытаемся найти последовательность модулей P с минимальной общей

продолжительностью изучения. С другой стороны, при этом мы пытаемся максимизировать суммарный уровень знаний по всем компетенциям K_j на момент окончания курса. Во-вторых, считаем для статистики такие показатели, как минимальное, максимальное и среднее значение ЦФ в рамках популяции (F_{\min} , F_{\max} , F_{avg}). В качестве приближения оптимального решения выбирается особь с наименьшим значением ЦФ.

3. Перемешивание популяции. В реализованном варианте генетического алгоритма скрещивание происходит между «соседними» особями популяции. Для обеспечения генетического разнообразия перед скрещиванием особи в популяции перемешиваются случайным образом.

4. Отбор особей. «Соседние» особи сравниваются попарно, и лучшая из них (имеющая меньшее значение ЦФ) попадает в новое поколение особей. Таким образом можно получить только половину популяции, поэтому далее производится еще одно перемешивание и процедура отбора повторяется второй раз.

5. Скрещивание и мутация. Процедура скрещивания выполняется в два шага.

Шаг 1. Выделяем *элитных* особей. Необходимость их ввода в алгоритм поясняет рисунок 2.10.

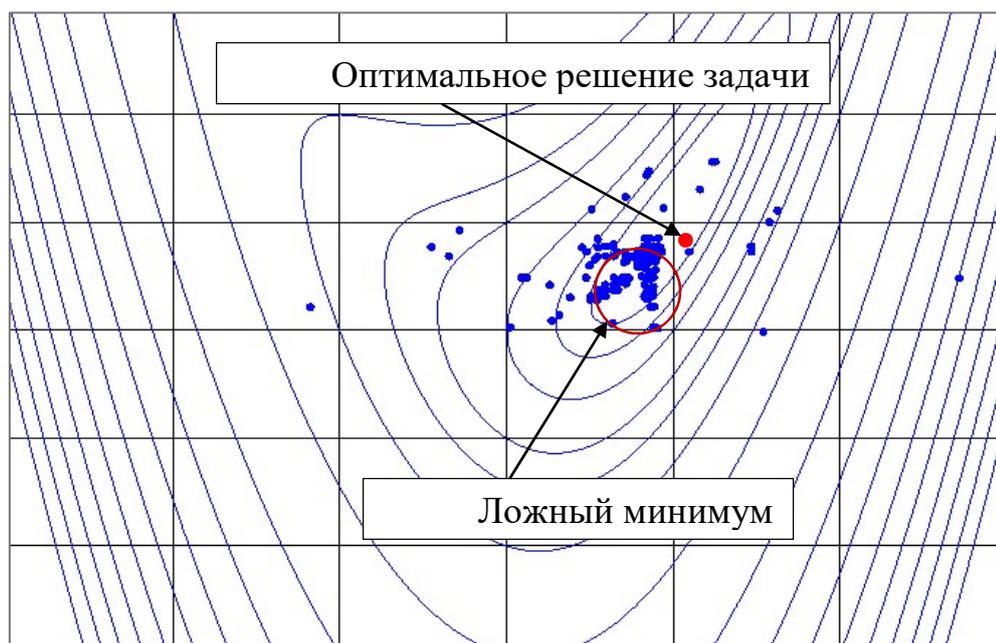


Рисунок 2.10 – Ложная сходимость генетического алгоритма

Начальное исследование работы генетического алгоритма осуществлялось на решении оптимизационных задач, а именно – минимизации функции Розенброка. Это функция выраженного овражного типа, поэтому многие оптимизационные алгоритмы не могут обеспечить для нее глобальную сходимость. В результате выяснилось, что зачастую генетический алгоритм обеспечивает ложную сходимость. Если вблизи какой-либо точки создавалась высокая концентрация особей популяции, то их генетический материал начинал перевешивать генетический материал остальных особей. Даже более удачные особи, расположенные ближе к оптимальной точке, в результате скрещивания с «мусорным» генетическим материалом становились менее пригодными. В итоге через несколько поколений практически вся популяция вырождалась в точку ложного минимума.

Чтобы решить эту проблему, в популяцию были введены элитные особи. Это небольшое число особей популяции (p_{elite}), имеющих наилучшее значение ЦФ.

Шаг 2. Выбираем две произвольные особи в старой популяции (в предыдущем поколении) и получаем из них две особи новой популяции. Если в скрещивании участвует хотя бы одна элитная особь, то ее хромосомы переходят потомкам следующего поколения без изменений. В противном случае с вероятностью p_{cross} выполняется одноточечное скрещивание.

Для этого сначала выбирается произвольная точка сечения $mate_1$ для первой родительской особи.

Затем несколько раз пробуем найти допустимую точку сечения $mate_2$ для второй родительской особи (рис. 2.11).

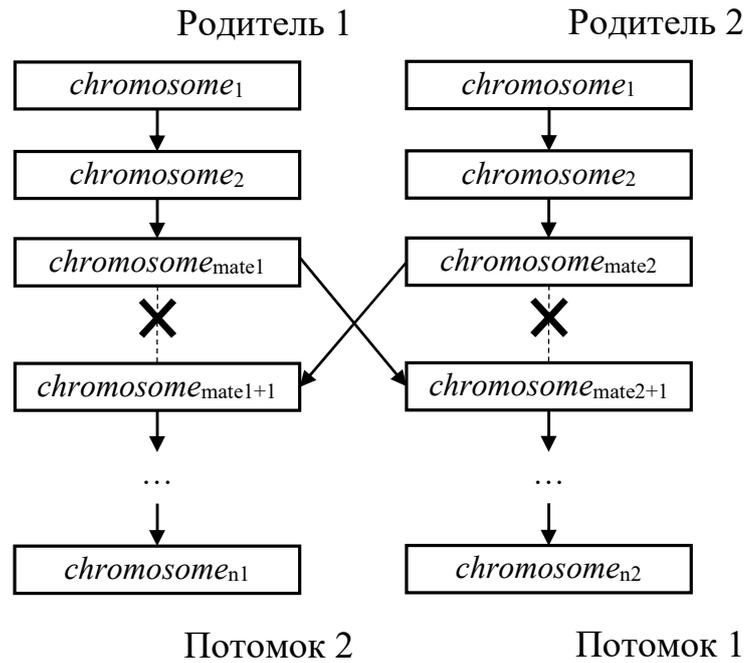


Рисунок 2.11 – Иллюстрация скрещивания особей

С этой целью выбираем случайным образом точку и проверяем, возможно ли скрещивание двух родительских особей в точках $mate_1$ и $mate_2$ соответственно. Для этого программа формирует «волны» компетенций для каждого родителя – две волны изученных компетенций, распространяющихся от начала цепочек хромосом (KS_1 , KS_2), и две волны неизученных компетенций, распространяющихся с конца цепочек (KF_1 , KF_2):

$$KS_k = KS \cup \bigcup_{i=1}^{mate_k} KO_i,$$

$$KF_k = \left(KF \cup \bigcup_{i=mate_k+1}^{n_k} KI_i \right) - \left(KS \cup \bigcup_{i=mate_k+1}^{n_k} KO_i \right).$$

Скрещивание возможно, если $KS_1 \subseteq KF_2$ и $KS_2 \subseteq KF_1$.

Если точки скрещивания найти удалось, то выполняем его по схеме на рисунке 2.8.

Эксперименты показывают, что число удачных скрещиваний находится на уровне 30%.

При выполнении процедуры мутации с вероятностью $p_{mutation}$ происходит одно из двух событий:

- либо замена одного модуля в последовательности другим модулем из базы, если это не нарушает описанных выше условий,
- либо перестановка двух модулей в последовательности местами при тех же условиях.

6. Проверка оптимальности решения. Увеличиваем счетчик поколений, и если его значение превышает n_{gen} , то работа алгоритма закончена. Кроме того, считается, что оптимальное решение найдено, если значение F_{min} не изменялось на протяжении последних 10 поколений.

В качестве языка реализации был выбран язык C#. Было выполнено несколько оптимизаций генетического алгоритма, после чего проведено исследование скорости его работы в зависимости от количества модулей и компетенций, входящих в адаптивный курс (рис. 2.12). Также была осуществлена реализация алгоритма на языке PHP с целью адаптации программного решения для веб-среды. Однако C# показал большую производительность по сравнению с языком PHP.

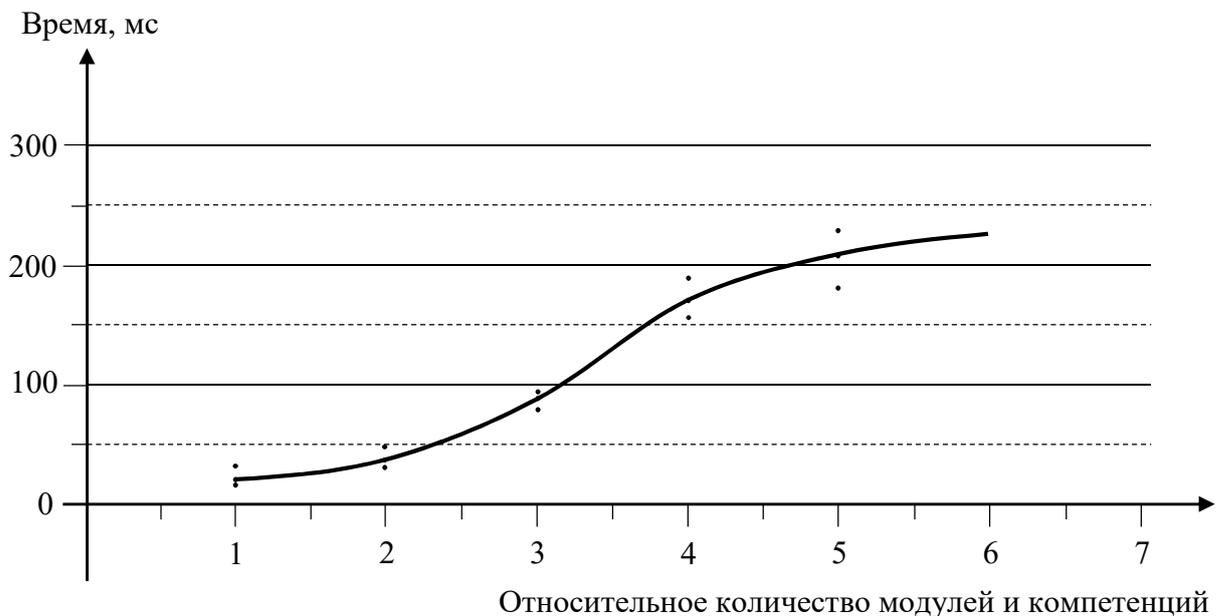


Рисунок 2.12 – Исследование скорости работы генетического алгоритма

На графике за единицу по оси абсцисс принято условное количество модулей и компетенций тестового фрагмента электронного курса (по дисциплине

«Информатика»), состоящего из 43 модулей и 57 компетенций. Остальные значения были получены при кратном увеличении этого количества. По оси ординат в каждом случае отложено по три точки – минимальное и максимальное время работы алгоритма при данном количестве модулей и компетенций, а также медианное значение, найденное как точка с минимальным среднеквадратичным отклонением:

$$t: \sum_{i=1}^n (t-t_i)^2 \rightarrow \min.$$

Некоторый разброс в результатах объясняется применением датчика случайных чисел. Поэтому при одних и тех же входных данных может быть сформировано различное количество поколений популяции, различное количество скрещиваний и т.п. Как показывают эксперименты, максимальное количество поколений, которые требуются генетическому алгоритму для нахождения оптимального решения, зависит от вариативности базы модулей и компетенций. Если вариативность не слишком высокая, то оптимальное решение может быть найдено уже к 5-му поколению. При повышении вариативности требуемое количество итераций работы алгоритма возрастает.

Таким образом, время работы алгоритма зависит от общего количества модулей и компетенций, имеющихся в курсе, а также от степени вариативности модулей. От общего количества модулей и компетенций в БД, а также общего количества студентов зависимости нет. Проседание производительности сервера генетического алгоритма может наблюдаться только в том случае, если к нему поступит одновременно множество запросов от разных клиентов.

Как видно из рисунка 2.9, до некоторой отметки скорость роста функции на графике превышает линейную, но затем замедляется, т.к. некоторые этапы работы алгоритма, имеющие приблизительно фиксированное время, перестают заметно влиять на общее время его работы. Это дает возможность экстраполировать скорость работы алгоритма для курсов с произвольным количеством модулей и компетенций.

Выводы по главе 2

1. Для реализации адаптивного обучения был получен алгоритм, применяющий модель навигации «*полное руководство*» [18]: на каждом шаге обучаемому предлагается к освоению только один модуль. После прохождения тестирования, направленного на измерение уровня знаний по только что изученным компетенциям, алгоритм произведет новое вычисление траектории P с учетом текущего состояния модели обучаемого S , за счет чего каждый обучаемый будет продвигаться по индивидуальной траектории.

2. Алгоритм реализует модель итеративного обучения на основе кривой забывания: в моменты, когда уровень знаний по какой-либо компетенции упадет ниже установленного, соответствующий модуль будет включен в набор популяции. Таким образом, вывод нового модуля будет зависеть как от результатов прохождения предыдущего, так и от наличия утраченных компетенций.

3. Генетический алгоритм всегда будет иметь решение, поскольку заданы условия остановки поиска оптимальных решений.

3 ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА ПОСТРОЕНИЯ ТЕХНОЛОГИИ АДАПТИВНОГО ОБУЧЕНИЯ

3.1 Требования к программному обеспечению

Анализ требований является необходимым этапом в системном проектировании для определения способов реализации программных средств, а также при разработке пользовательских интерфейсов. Рассмотрим некоторые нефункциональные и функциональные требования к разрабатываемому программному обеспечению.

Нефункциональные требования. К нефункциональным требованиям относят требования, характеризующие программную систему в целом, а не отдельные сценарии ее поведения.

1. *Технологическая платформа и среда эксплуатации.* Наиболее перспективным направлением развития программного обеспечения является реализация веб-ориентированных систем. Это могут быть облачные решения, как открытого доступа, так и функционирующие в ограниченных локальных сетях. Преимуществом реализации систем на основе веб-технологий является:

- возможность решать задачи из любого места практически с любого компьютера, подключенного к сети и имеющего предустановленный веб-браузер;
- такие системы, в отличие от большей части настольного программного обеспечения, не нужно устанавливать, что в свою очередь позволяет снизить требования к компетенциям конечного пользователя системы;
- стоимость приобретения доступа к таким системам как правило ниже, по сравнению со стоимостью устанавливаемого настольного ПО, поскольку целевая аудитория пользуется услугами системы, а не приобретает весь программный комплекс.

Таким образом, реализовывать систему адаптивного обучения целесообразно на основе веб-технологий. Это позволит:

- интегрировать адаптивное обучение в инфраструктуру системы электронного обучения учебного заведения, обеспечивающей формат дистанционного обучения;
- снизить требования к программному обеспечению устройства конечного пользователя (потребуется только веб-браузер);
- реализовать кросс-платформенность доступа.

2. *Технология реализации.* При реализации веб-ориентированных систем используется широкий перечень технологий: PHP, Python, .Net, Java и другие. Преимущества одной технологии перед другой явно не прослеживаются, поэтому фактором выбора языка программирования могут послужить свойства системы или инфраструктуры, в которую интегрируется разработка. В качестве такой системы была выбрана система дистанционного обучения Moodle [50] как самая распространенная СДО среди учебных заведений по всему миру. Moodle определяет собственные стандарты разработки подсистем (модулей), предназначенные для расширения функциональных возможностей системы [51]. В соответствии с этими стандартами для реализации модуля адаптивного обучения был выбран язык программирования PHP.

Однако, как уже было сказано ранее, опыты показали, что скриптовые языки типа PHP слабо подходят для решения задач по реализации генетических алгоритмов ввиду низкой скорости поиска решений. Таким образом, целесообразным является представление архитектуры системы в виде отдельных компонентов:

- модуль, отвечающий за взаимодействие пользователя с системой и интеграцию с СДО, реализуется на технологии PHP (клиент);
- модуль, отвечающий за работу алгоритма адаптивного обучения, реализуется на языке C# (сервер).

3. *Язык представления данных.* Стандартом представления информации в веб-среде является гипертекстовый язык HTML. Гипертекстовая модель позволяет успешно решать задачи навигации и визуализации контента, однако слабая

структурированность данных, представленных в данной модели (в частности, отсутствие типизации данных), существенно ограничивает возможности их программной обработки и функциональные возможности системы [19, 20].

Рекомендованной консорциумом W3C технологией описания данных для баз знаний является язык OWL [52]. Основным преимуществом применения OWL является способность реализовывать семантический поиск по базе данных контента. Спецификация OWL содержит в себе все необходимые конструкции для реализации фреймовой модели, подходящей для описания структуры учебного модуля: описание классов, свойств, экземпляров классов и отношений между ними.

4. *Модульное построение архитектуры.* Требование, необходимое для обеспечения в проектируемых системах расширяемости и замены составных компонентов. Расширяемость – возможность адаптации программного обеспечения к изменениям спецификации. Сценарии, по которым производится моделирование учебных процессов, могут подвергаться изменениям, модели могут меняться, соответственно необходим подход, при котором внесение изменений будет происходить с наименьшими затратами. Чем более автономны модули программной архитектуры, тем выше вероятность того, что простое изменение затронет только один или небольшое количество модулей и не вызовет цепную реакцию изменений во всей системе [53].

5. *Совместимость.* Данное требование обуславливает необходимость взаимодействия компонентов системы между собой и с внешними системами. Поскольку в реализации системы адаптивного обучения участвует набор различных технологий, для корректного взаимодействия модулей необходимо обеспечить функционирование на основе общих стандартов передачи данных в программных интерфейсах API.

Функциональные требования – требования, идентифицирующие задачи или действия, которые должна выполнить система. Система адаптивного обучения должна позволять выполнять следующие основные функции:

- наполнять БД образовательными объектами – модулями, компетенциями, тестами;
- организовывать в рамках СДО Moodle навигацию по адаптивному курсу;
- предоставлять обучаемому и преподавателю функциональность личного кабинета с историей обучения и текущей успеваемостью.

Эффективным способом определения элементов интерфейса разрабатываемого продукта является анализ пользовательских требований [65]. Фиксировать пользовательские требования удобно с помощью описания прецедентов. Прецеденты, иными словами сценарии использования, определяют взаимодействия между пользователем и системой, направленные на достижение цели.

Ключевыми пользователями проектируемой системы являются:

- *обучаемые* – участники этой роли имеют возможность изучения адаптивных курсов, а также доступ к личному кабинету;
- *преподаватели* или *авторы адаптивных учебных курсов* – участники этой роли имеют доступ к инструментарию для наполнения базы данных образовательными объектами, а также к истории обучения и успеваемости студентов;
- *администраторы* – участники этой роли имеют доступ к настройке серверов системы, к управлению БД, а также к управлению СДО Moodle.

Рассмотрим каждую роль в отдельности.

1. Роль «Обучаемый». Обучаемому в системе доступны следующие функции:

- авторизация в системе для доступа к выбору адаптивного курса;
- доступ к кабинету успеваемости с возможностью просмотра данных о прогрессе в изучении адаптивных курсов;
- выбор адаптивного курса из списка доступных курсов;
- выбор любого модуля курса для изучения, если это разрешено настройками;
- просмотр рекомендуемого модуля;

- переход к следующему модулю после просмотра текущего;
- прохождение тестирования после изучения модуля.

Диаграмма вариантов использования роли «Обучаемый» изображена на рисунке 3.1.

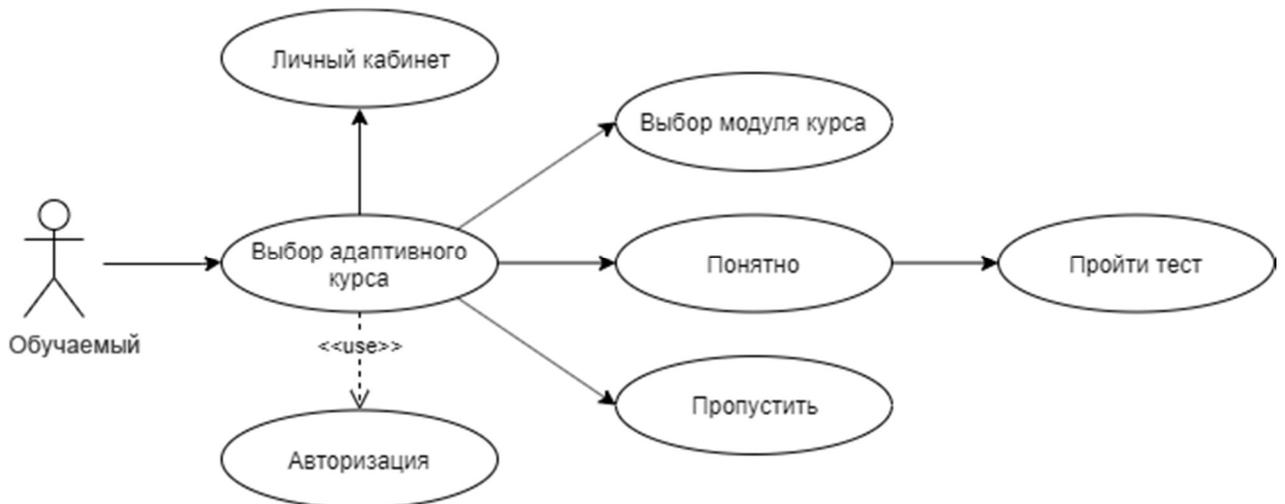


Рисунок 3.1 – Диаграмма вариантов использования роли «Обучаемый»

Переход к следующему модулю реализован двумя способами: через нажатие кнопки «Понятно» или кнопки «Пропустить». Последняя введена для того, чтобы исключить возможность заикливания на модуле, если обучаемый не может пройти тест.

2. Роль «Преподаватель». Преподавателю в системе доступны следующие функции:

- авторизация в системе для доступа к просмотру, созданию и редактированию адаптивных курсов;
- доступ к кабинету успеваемости с возможностью просмотра данных о прогрессе студентов в изучении адаптивных курсов, автором которых является данный преподаватель;
- редактирование настроек адаптивных курсов;
- добавление, удаление и редактирование записей в справочнике категорий для хранения компетенций и модулей адаптивного курса по произвольному объединяющему признаку;

- добавление, удаление и редактирование записей в базах модулей и компетенций адаптивного курса;
- добавление, удаление и редактирование тестов в адаптивном курсе;
- все функции обучаемого.

Диаграмма использования вариантов роли «Преподаватель» изображена на рисунке 3.2.

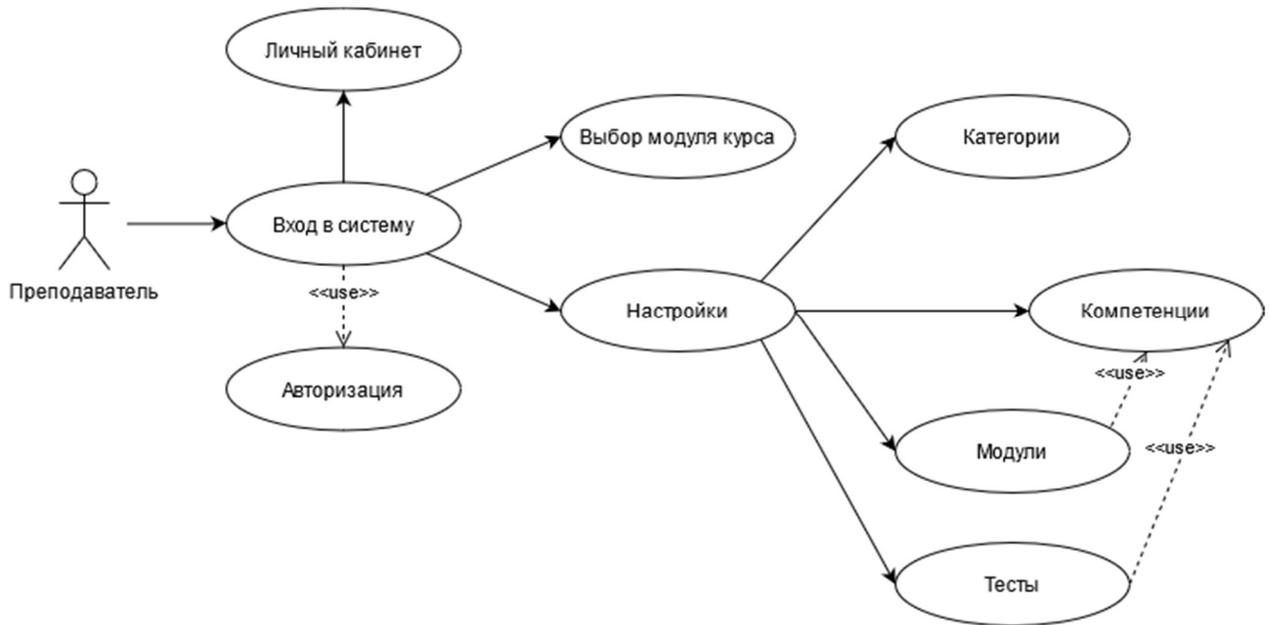


Рисунок 3.2 – Диаграмма вариантов использования роли «Преподаватель»

3. Роль «Администратор». Администратору системы доступны следующие функции:

- полный доступ к функциям СДО Moodle;
- доступ к серверам системы;
- доступ к исходному коду плагинов, инструментальных средств и скриптов, обеспечивающих работу системы;
- доступ к серверу для загрузки файлов программной реализации адаптивного алгоритма;
- доступ к базе данных для непосредственного редактирования обучающих объектов адаптивных курсов;
- доступ к логам работы системы;
- все функции преподавателя.

3.2 Пользовательские интерфейсы системы

Рассмотрим экранные формы (интерфейс), посредством которых пользователи будут взаимодействовать с системой в контексте функциональных возможностей, описанных ранее.

Роль «Обучаемый»

Авторизация и Выбор адаптивного курса из списка курсов. Так как система адаптивного обучения реализуется на базе СДО Moodle, многие сценарии данная СДО закрывает своим штатным набором функционала. Так, авторизация пользователя и навигация по курсам производится через стандартные интерфейсы Moodle, поэтому дополнительные экраны для этих действий не требуются.

Изучение модуля курса. Макет интерфейса модуля курса изображен на рисунке 3.3.

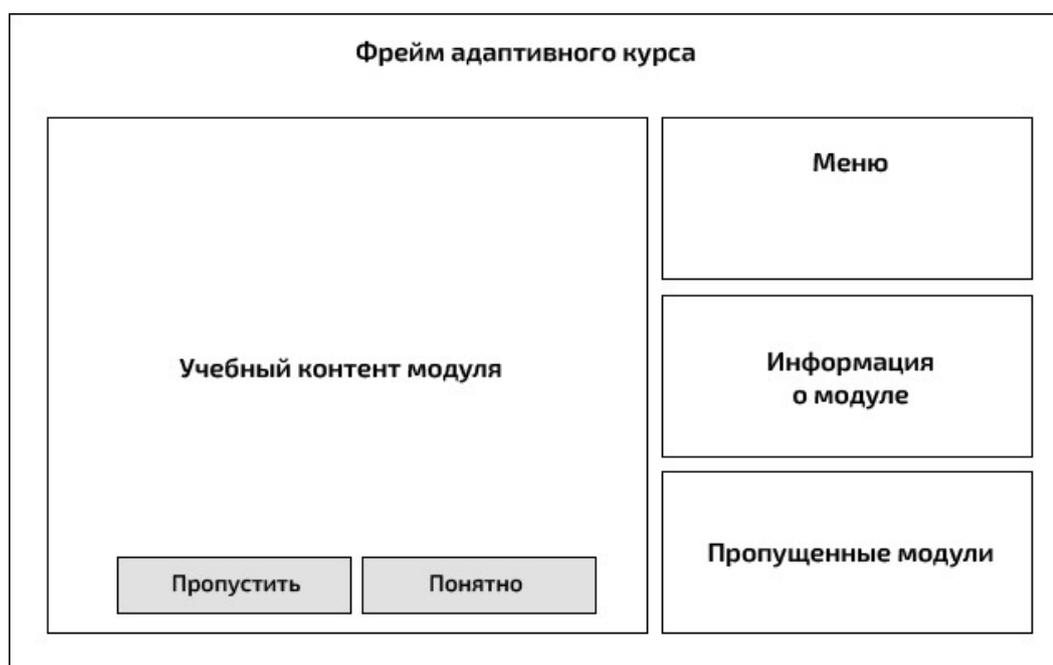


Рисунок 3.3 – Макет интерфейса просмотра модуля курса

В правом верхнем углу размещается *меню* курса, в котором содержатся функциональные ссылки, позволяющие:

- посмотреть причину, по которой алгоритм адаптивного обучения (ААО) рекомендовал для изучения данный модуль;
- перейти к окну выбора модуля;

– перейти к истории обучения и успеваемости.

Ниже отображается блок с *информацией о модуле*, который содержит перечень знаний (компетенций), требуемых для его изучения, и знаний, получаемых в результате его изучения.

В нижнем правом углу выводится список пропущенных модулей, чтобы к ним было проще вернуться.

В центральной части экрана выводится контент учебного модуля. Для завершения просмотра модуля обучаемый должен нажать кнопку «Пропустить» (отображается только при повторном изучении модуля) или «Понятно». В первом случае модуль попадает в список пропущенных модулей, а обучаемый переходит к следующему модулю курса. Во втором случае обучаемый должен пройти тестирование по изученному материалу для определения дальнейшей траектории обучения.

Алгоритм адаптивного обучения может включать в траекторию обучаемого тесты, которые не обеспечены учебными модулями. Это может быть входное диагностическое тестирование или тестирование, направленное на измерение остаточного уровня знаний. Для того чтобы обучаемым появление таких модулей на экране интерпретировалось как понятное событие, целесообразно использовать специальный тип модулей «без учебного контента» (рисунок 3.4). Этот модуль используется в качестве информационного экрана, текст сообщения которого может содержать пояснения о действиях системы в случаях, когда ААО выводит обучаемому тест, не связанный с предыдущим модулем.

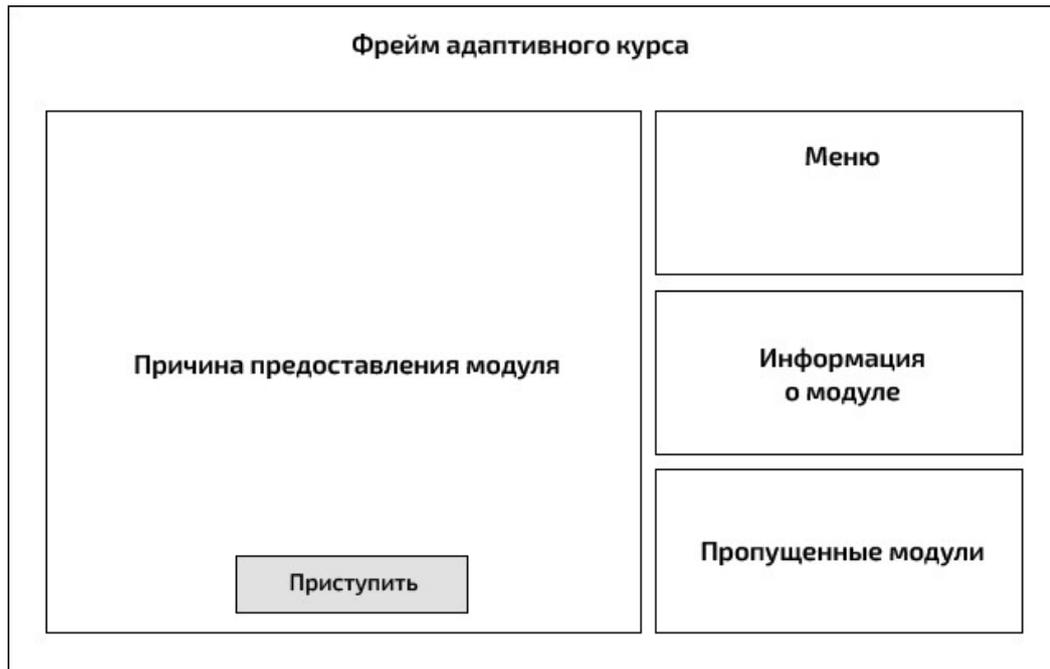


Рисунок 3.4 – Макет интерфейса для модуля без контента

Выбор модуля адаптивного курса. Если в настройках курса включена свободная навигация, обучаемый имеет возможность обратиться к произвольному модулю курса, а не только к рекомендуемому системой. Макет интерфейса выбора модуля изображен на рисунке 3.5.

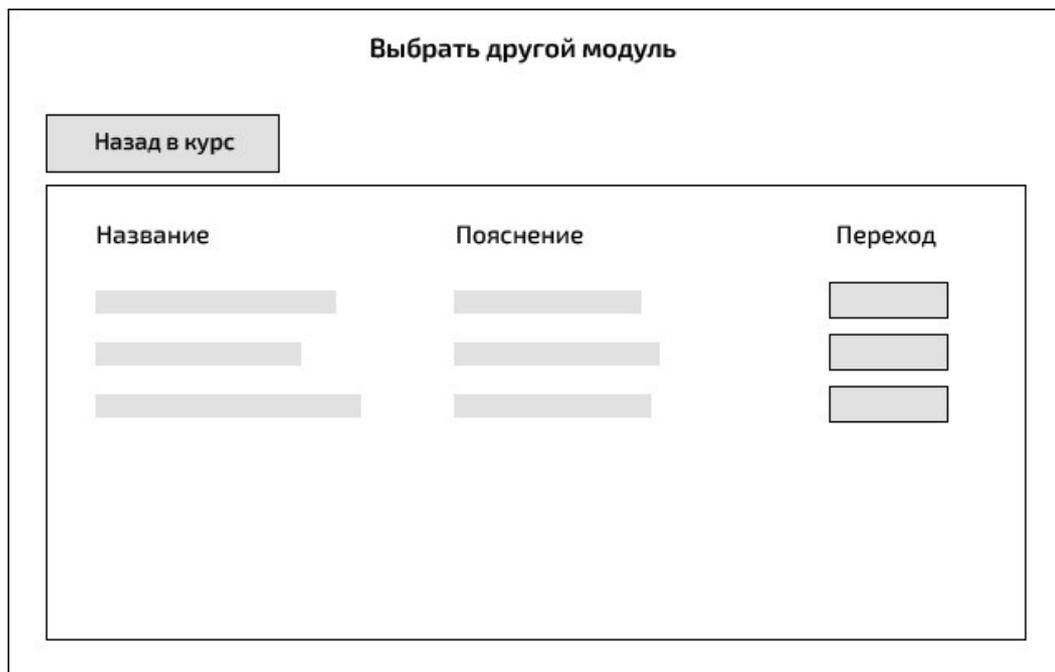


Рисунок 3.5 – Макет интерфейса выбора модуля

Пояснение помогает определиться с выбором и может указывать, что модуль:

- рекомендован к изучению системой (ААО) в настоящий момент;
- доступен для изучения (т.е. все его входные компетенции ранее были освоены);
- уже изучен.

Прохождение тестирования. Тестирование в системе осуществляется стандартными средствами СДО Moodle, никакие специальные интерфейсы для его реализации не требуются.

История обучения. Макет экрана истории обучения студента изображен на рисунке 3.6. В левом верхнем углу размещается кнопка для возврата в адаптивный курс. Ниже представлен список изученных модулей и обеспечиваемых ими компетенций.

История обучения			
Назад в курс			
Модуль	Компетенции	Тип модуля	Время изучения
█	█ █	█	█
█	█ █	█	█
█	█ █	█	█

Рисунок 3.6 – Макет интерфейса истории прохождения адаптивного курса

Успеваемость. Макет формы, предоставляющей сведения об успеваемости студента, представлен на рисунке 3.7.

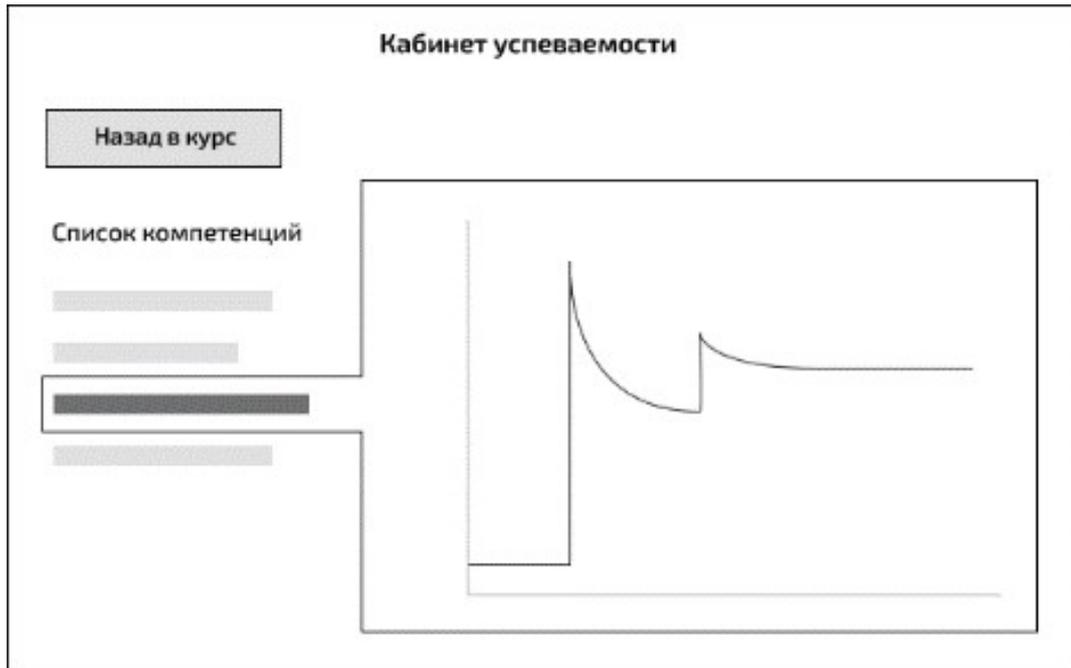


Рисунок 3.7 – Макет интерфейса кабинета успеваемости обучаемого

В левой части окна представлен список компетенций, для каждой из которых указан текущий уровень ее усвоения обучаемым. При выборе компетенции из списка в правой части окна появляется график, показывающий историю изменения уровня усвоения компетенции.

Роль «Преподаватель»

Авторизация и Навигация по курсам. Как и в случае с ролью «Обучаемый», авторизация преподавателя и переход к курсам в системе осуществляется стандартными средствами СДО Moodle, никакие специальные интерфейсы не требуются.

Наполнение адаптивного курса. Для наполнения адаптивного курса образовательными объектами необходимо зайти на страницу настроек, макет которой изображен на рисунке 3.8.



Рисунок 3.8 – Макет интерфейса страницы настроек курса

По умолчанию на этой странице отображаются общие настройки курса, включающие настройки сервера (адрес сервера и номер порта), и настройки курса (даты начала и окончания курса, требуемый уровень изучения компетенций, опция свободного перемещения).

Редактирование категорий компетенций и модулей. Редактирование категорий компетенций и модулей доступно по нажатию кнопки «Категории» (рис. 3.9). Категории нужны для объединения компетенций и модулей по некоторому объединяющему признаку, для удобства хранения и обращения к данным базы адаптивного контента.

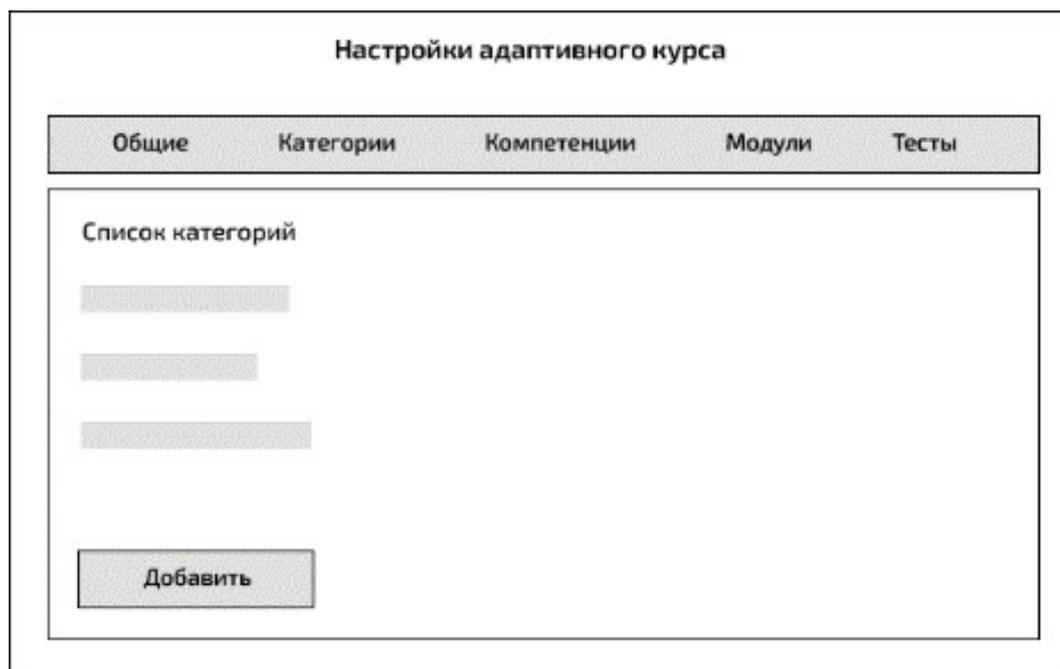


Рисунок 3.9 – Макет интерфейса для редактирования категорий

Предусмотрена возможность добавления категорий, а также просмотра списка категорий. Выбранную категорию можно редактировать или удалить.

Редактирование компетенций. Редактирование компетенций доступно по нажатию кнопки «Компетенции» (рис. 3.10). В верхнем левом углу окна предусмотрен выпадающий список для выбора категории. В списке ниже отображаются компетенции только из выбранной категории. Отображение категорий предусмотрено в виде карточек, на которых выводится наименование компетенции, ее основные атрибуты, кнопки для быстрого доступа к настройкам компетенции и ее удаления. Первая карточка в списке пустая, нажатие на нее инициирует создание новой компетенции.

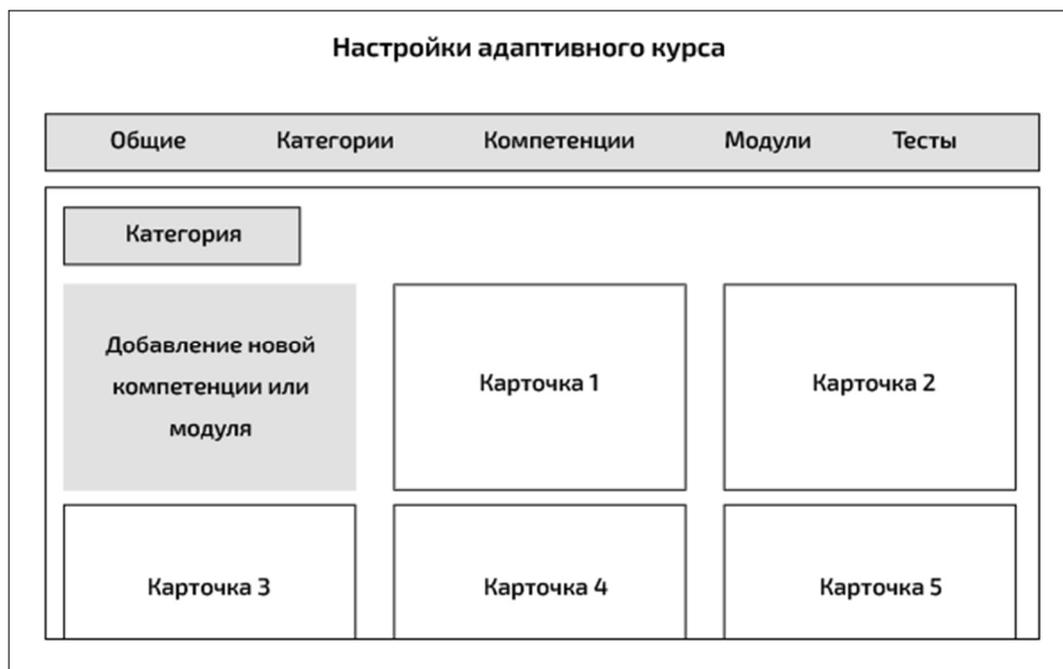


Рисунок 3.10 – Макет интерфейса для редактирования модулей и компетенций

При добавлении или удалении новой компетенции необходимо указать такие ее атрибуты, как:

- наименование;
- тип оценивания компетенции (автоматически или вручную преподавателем);
- уникальный идентификатор;
- категория;
- описание.

Редактирование модулей. Редактирование модулей доступно по нажатию кнопки «Модули». Структура интерфейс аналогична той, что приведена для интерфейса редактирования компетенций (рис. 3.10). Соответственно, нажатие на первую карточку в списке инициирует создание нового модуля.

При добавлении или удалении нового модуля необходимо указать такие его атрибуты, как:

- наименование;
- уникальный идентификатор;

- категория;
- контент модуля.

Редактирование связи модулей. Для определения связей между модулями необходимо определить их входные и выходные компетенции таким образом, чтобы образовался двудольный граф образовательных объектов. Для доступа к редактированию связей (входных и выходных компетенций модуля) предусмотрена специальная кнопка на карточках модулей. Интерфейс редактирования связей изображен на рисунке 3.11.

Настройки адаптивного курса

Общие	Категории	Компетенции	Модули	Тесты												
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Компетенция</th> <th style="width: 20%;">Тип</th> <th style="width: 30%;">Уровень</th> </tr> </thead> <tbody> <tr> <td style="height: 20px;"> </td> <td style="height: 20px;"> </td> <td style="height: 20px;"> </td> </tr> <tr> <td style="height: 20px;"> </td> <td style="height: 20px;"> </td> <td style="height: 20px;"> </td> </tr> <tr> <td style="height: 20px;"> </td> <td style="height: 20px;"> </td> <td style="height: 20px;"> </td> </tr> </tbody> </table>					Компетенция	Тип	Уровень									
Компетенция	Тип	Уровень														

Рисунок 3.11 – Макет интерфейса для редактирования связей модулей и компетенций

Здесь для каждой компетенции можно указать ее отношение к текущему модулю – либо она к нему никак не относится, либо является входной, либо выходной (настраивается в столбце «Тип»). Если компетенция является выходной, необходимо указать, на каком уровне (по шкале от 0 до 100) студент может изучить ее в данном модуле (настраивается в столбце «Уровень»).

Редактирование тестов. Редактирование тестов в системе осуществляется стандартными средствами СДО Moodle, никакие специальные интерфейсы не требуются.

Кабинет успеваемости (режим преподавателя). Макет интерфейса кабинета успеваемости для роли преподавателя изображен на рисунке 3.12. В левом верхнем углу размещается кнопка для возврата в адаптивный курс. Ниже представлен список обучаемых и ссылки на их историю обучения и успеваемость.

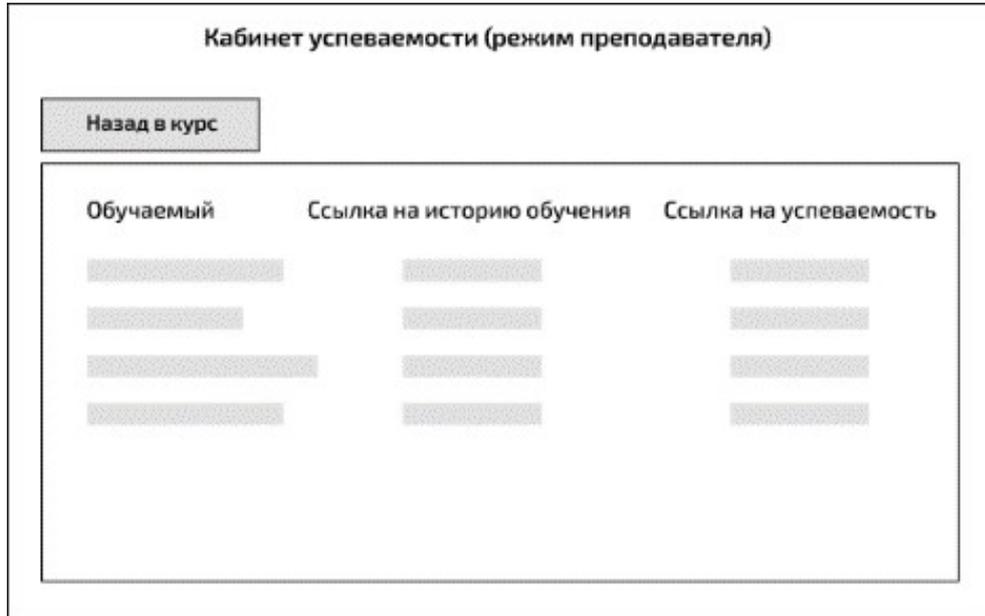


Рисунок 3.12 – Макет интерфейса кабинета успеваемости преподавателя

Интерфейс окна для просмотра истории обучения и успеваемости аналогичен тому, который предоставляется обучаемым (см. рис. 3.6 и 3.7).

3.3 Программная реализация системы адаптивного обучения

Архитектура полученного программного комплекса изображена на рисунке 3.13.

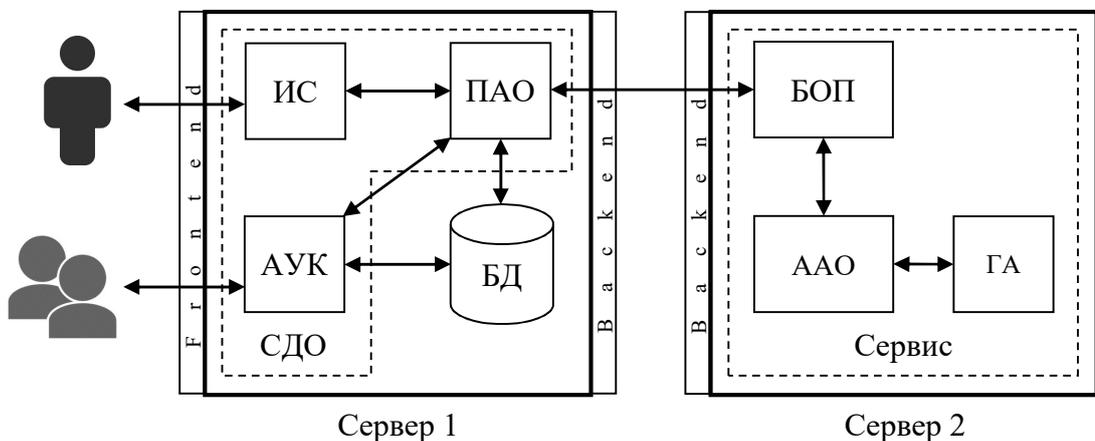


Рисунок 3.13 – Общая архитектура программного комплекса

Программный комплекс состоит из двух серверов, между которыми реализован обмен данными, и включает в себя:

Сервер 1 (Linux):

1. СДО – система дистанционного обучения Moodle, состоящая из компонентов:

- адаптивный учебный курс (блок «АУК»);
- инструментальная система для работы с базой адаптивного контента и разработки адаптивных электронных курсов (блок «ИС»);
- плагин адаптивного обучения (блок «ПАО»), реализующий сценарии адаптивного обучения и осуществляющий функции обмена данными между СДО (Сервер 1) и сервисом, развернутым на Сервере 2.

2. База данных (блок «БД»), реализующая хранение информации о модулях курса, тестах, профилях студентов и т.д.

Сервер 2 (Windows):

3. Сервис (служба Windows), обменивающийся данными с блоком ПАО и реализующий алгоритмы работы системы адаптивного обучения (блок «Служба»). Данная служба включает следующие компоненты:

- блок отправки и приема данных (БОП), осуществляющий обмен данными с ПАО в формате JSON по протоколу TCP – принимает информацию, необходимую для запуска адаптивного алгоритма (профиль пользователя, списки модулей, компетенций и других компонентов АУК), и отправляет в обратном направлении результаты его работы (следующий рекомендуемый модуль траектории, списки забытых компетенций, точки кривой забывания);
- программная реализация алгоритма адаптивного обучения (блок «ААО»), обеспечивающая вычисление коэффициентов кривой забывания, запуск генетического алгоритма, формирование точек кривой и т.п.;
- программная реализация генетического алгоритма (блок «ГА»), реализующая определение оптимальной траектории обучения.

Данная архитектура позволяет гибко менять конфигурацию решения. Она сделана максимально независимой от используемой СДО – при необходимости использования АУК в другой СДО потребуется только переработка блока ПАО. Реализация ААО (и особенно ГА) на отдельном сервере удобна также для балансировки нагрузки – работа алгоритмов требует существенных вычислительных ресурсов, поэтому использование облачного сервиса позволяет не увеличивать отклик сервера, на котором функционирует СДО.

Логическая структура базы данных для хранения всей информации, относящейся к адаптивному электронному курсу, изображена на рисунке 3.14 (на схеме не приведены таблицы СДО Moodle, в которых хранятся стандартные сущности (студент, курс, тест, попытка и т.д.), поскольку они не претерпели изменений и используются как есть).

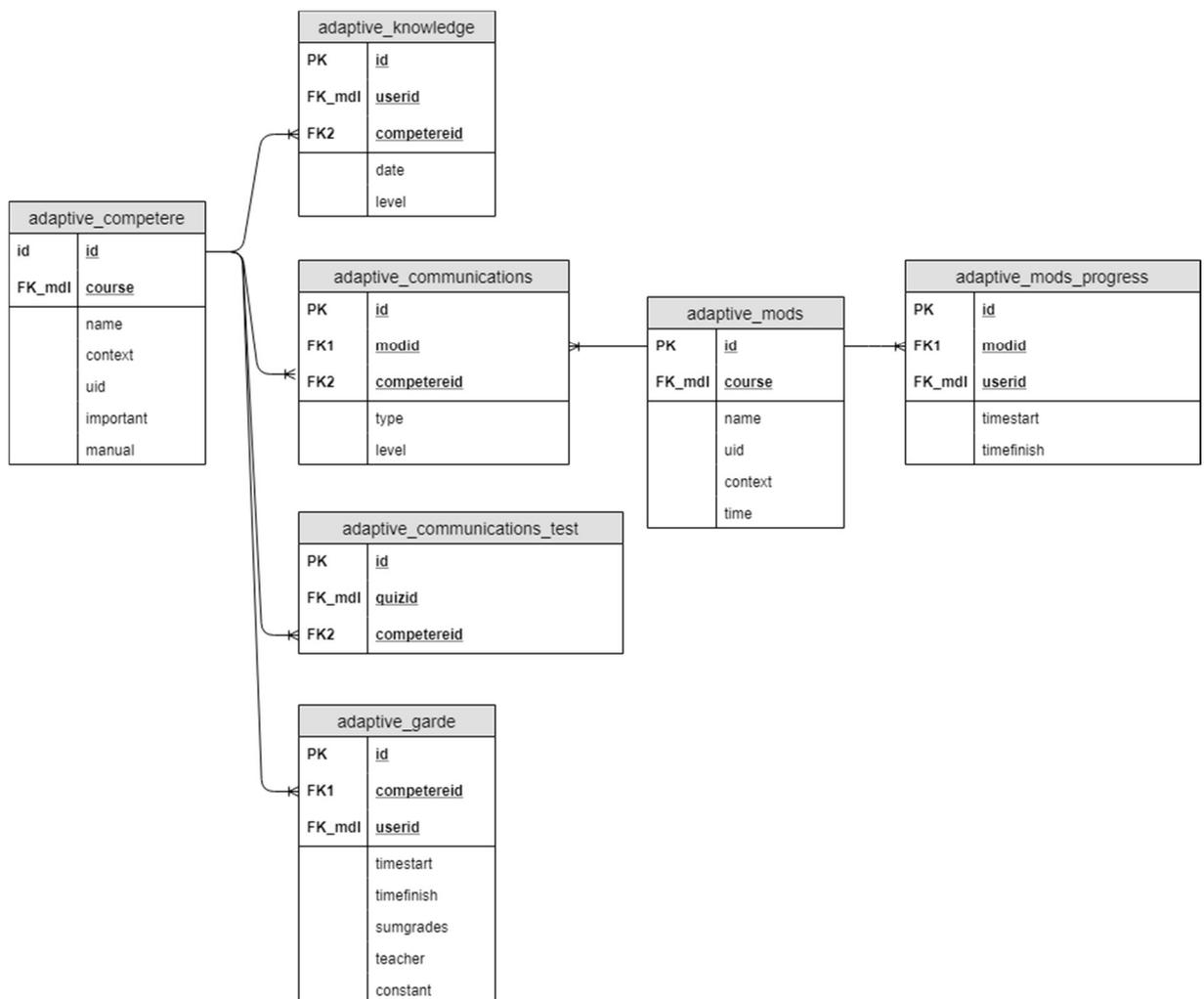


Рисунок 3.14 – Логическая схема БД

Здесь каждая запись таблицы `adaptive_competere` описывает компетенцию адаптивного курса, `adaptive_mods` – модуль курса, `adaptive_communications` – связь между модулем, его входными и выходными компетенциями, а также пререквизитами. Таблица `adaptive_knowledge` содержит информацию для построения графиков уровня усвоения компетенций в личном кабинете студента. Таблица `adaptive_communication_tests` связывает компетенцию с тестами СДО, которые позволяют проверить уровень ее освоения. Таблица `adaptive_grade` содержит оценки как за автоматическое тестирование, так и выставленные преподавателем вручную. В таблице `adaptive_mods_progress` хранится история изучения обучаемым модулей курса.

Общий алгоритм работы программного решения:

1. При формировании адаптивного учебного курса преподаватель производит наполнение БД модулями, компетенциями и другими объектами посредством инструментальной системы (ИС).

2. Когда в адаптивный курс заходит студент, ядро (ПАО) проверяет, определен ли следующий модуль для предоставления его студенту. Если нет, то запускается работа ААО. Модуль может быть уже определен, если студент не закончил его изучение при предыдущем сеансе работы с системой.

3. Сервис на Сервере 2 получает TCP-запрос, в который в числе прочих параметров входят данные профиля студента и содержимое адаптивного курса. После этого запускается компонент ААО.

4. ААО проверяет, есть ли модули, которые студент должен получить вне очереди (добавление которых в траекторию обусловлено механизмами пререквизитов и маркеров). Если такие модули есть, то результатом работы ААО будет идентификатор одного из них, выбранного случайным образом. Далее ААО вычисляет актуальные коэффициенты кривой забывания и определяет, закончено ли студентом изучение курса. Если закончено (т.е. все компетенции изучены на достаточном уровне), эта информация будет передана в ПАО. Если нет, для определения следующего рекомендуемого модуля запускается ГА.

5. Если ПАО получил информацию, что изучение курса окончено, студент получает сообщение об этом. В противном случае ему предоставляется следующий рекомендуемый модуль. После изучения модуля студент выполняет тест, проверяющий уровень усвоения модуля (точнее, обеспечиваемых им компетенций), после чего следует возврат на шаг 2.

Первые тестирования компонентов ААО и ГА производились на языке PHP, при этом данный язык показал крайне низкую скорость при обработке информации и проведении вычислений. Финальная реализация алгоритмов осуществлена на языке C# в виде сервиса Windows.

Плагин адаптивного обучения реализован на языке PHP, однако может быть использовать любой язык, поддерживаемый сервером СДО.

Экранные формы реализованной системы адаптивного обучения приведены в приложении А.

Разработанная система адаптивного обучения при государственной регистрации программы ЭВМ получила название *Nexbe* (приложение В).

Выводы по главе 3

1. Получена программная реализация алгоритма адаптивного обучения, основанного на генетическом алгоритме поиска оптимальных траекторий обучения. Алгоритм адаптивного обучения реализован в виде сервиса Windows.

2. В качестве системы дистанционного обучения выбрана система Moodle, для которой был разработан плагин, обеспечивающий выдачу студенту модулей курса согласно рассчитанной траектории, а также предоставляющий доступ студенту и преподавателю к истории обучения и успеваемости в рамках адаптивного обучения. Обмен данными между плагином и алгоритмом происходит по протоколу TCP/IP.

3. Для СДО Moodle разработан инструментарий, позволяющий вести базу данных компонентов, необходимых для создания адаптивных курсов – создавать

модули, компетенции, устанавливать между ними связи, назначать компетенциям тесты для проверки уровней освоения.

4 АНАЛИЗ ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ ПОСТРОЕНИЯ ТЕХНОЛОГИИ АДАПТИВНОГО ОБУЧЕНИЯ

4.1 Описание методики построения технологии адаптивного обучения

Рассмотрим методику создания адаптивных электронных курсов посредством применения САО Nexbe.

1. *Получение набора компетенций.* На данном этапе производится декомпозиция компетенций на составные элементы (субкомпетенции). Такая декомпозиция может быть представлена в виде дерева (рис. 4.1).

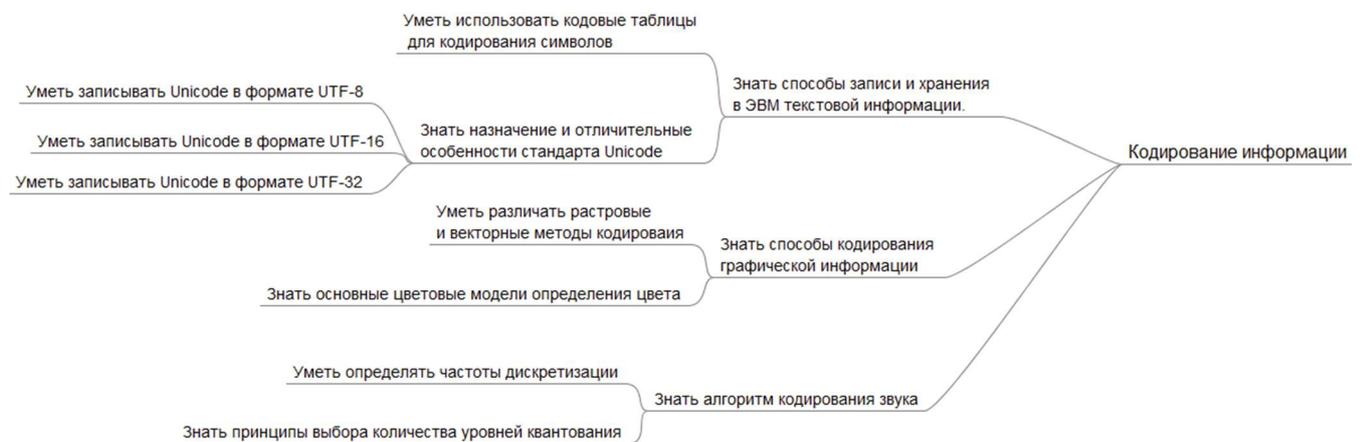


Рисунок 4.1 – Пример декомпозиции компетенций

На рисунке 4.1 показано, что тема «Кодирование информации» (из дисциплины «Информатика») на первом уровне декомпозиции представлена тремя элементами «Знать...», которые в свою очередь разбиты на составные субкомпетенции. Целью такого разбиения является получение списка элементарных компетенций, находящихся в листьях получаемого дерева. Декомпозиция производится до тех пор, пока не станет понятным, что уровень освоения субкомпетенции, находящейся в рассматриваемом листе дерева, может быть измерен с помощью тестовых заданий.

Полученные субкомпетенции вносятся в систему в разделе «Компетенции» посредством кнопки «Добавить» (см. приложение А, рис. А.8). Для каждой субкомпетенции необходимо указать название и тип. Для удобства работы с базой компетенций дополнительно могут быть введены номера и текстовое описание,

которые не влияют на поведение алгоритма построения траектории (приложение А, рис. А.9).

2. *Разработка модулей.* После того как в систему были введены субкомпетенции, необходимо внести данные о модулях, которые будут их обеспечивать. Под обеспечением в данном случае понимается создание модуля, у которого выходной будет являться та или иная субкомпетенция, полученная на этапе декомпозиции. При разработке модулей, обеспечивающих полученные субкомпетенции, возможны три подхода:

- модули разрабатываются только для листовых субкомпетенций;
- модули разрабатываются для всех субкомпетенций, по принципу от общего к частному (т.е. при изучении курса обучаемый сначала освоит субкомпетенции, и только потом – их родительскую компетенцию);
- модули разрабатываются для всех субкомпетенций, по принципу от частного к общему (т.е. при изучении курса обучаемый сначала освоит родительскую компетенцию, и только потом – её субкомпетенции).

Выбор одного из перечисленных подходов зависит от ряда факторов, таких как общая логика изложения материала курса, принципы декомпозиции компетенций и т.д.

Создание модулей производится в разделе «Модули» интерфейса системы (приложение А, рис. А.10) посредством нажатия кнопки «Добавить». При создании (приложение А, рис. А.11) указываются:

- название модуля;
- категория хранения;
- описание (контент модуля);
- время, отведенное на освоение модуля.

После того как модуль создан, необходимо указать для него входные и выходные субкомпетенции. Для этого предусмотрен специальный интерфейс (приложение А, рис. А.12), который вызывается через пиктограмму с

изображением карандаша в карточке модуля. В данном интерфейсе представлен список всех субкомпетенций курса, для которых указывается одно из значений:

- отсутствие связи с редактируемым модулем;
- отношение типа «выходная компетенция»;
- отношение типа «входная компетенция».

При присвоении выходной субкомпетенции здесь же указывается уровень, который обучаемый обретет по соответствующей субкомпетенции в результате работы с модулем.

3. Создание тестовых заданий. Система будет отслеживать процесс забывания студентом изученной информации на основе кривой забывания и в определенные моменты уточнять, забыл ли студент ту или иную субкомпетенцию или нет. Уточнение производится с помощью тестирования. Соответственно, для каждой субкомпетенции должен быть создан тест, состоящий из некоторого количества вопросов (одного или более). Эти же тесты будут предъявляться студенту после изучения соответствующего модуля.

При создании тестов соблюдается принцип: проверке одной субкомпетенции соответствует один тест. Тесты создаются встроенными средствами СДО Moodle. Все созданные тесты выводятся в разделе «Тесты» (приложение А, рис. А.13), где разработчику необходимо установить соответствие между тестом и компетенцией.

4. Запуск адаптивного электронного курса. Перед запуском производится установление общих настроек адаптивного курса. В соответствующем разделе разработчику необходимо указать дату начала и окончания обучения в курсе, а также задать режим навигации обучаемого по курсу (свободный или строгое следование алгоритму).

На этом создание адаптивного электронного курса завершено. Для того чтобы протестировать полученный результат, разработчику следует завести новую учетную запись и подписать её на созданный курс с ролью «Обучаемый», это производится средствами штатного функционала СДО.

Для разработчиков адаптивных электронных курсов также была разработана инструкция по проектированию адаптивного контента, которая содержит в себе описание концептуальных основ адаптивного обучения и методики проектирования модулей и компетенций для последующей загрузки в САО Nexbe (приложение Б).

4.2 Создание адаптивного электронного курса «Химия» и его апробация в учебном процессе

В рамках соглашения между ТУСУР и НИТУ «МИСиС» в осеннем семестре 2018–2019 уч.г. кафедрой ОиНХ НИТУ «МИСиС» была проведена апробация адаптивного электронного курса по дисциплине «Общая химия» (раздел «Химическая кинетика») [54], разработанного с помощью САО Nexbe. При преподавании дисциплины использовалась модель смешанного обучения, основанная на технологии «перевернутого класса».

Реализация курса осуществлялась по разработанной методике создания адаптивных электронных курсов на основе системы адаптивного обучения (см. п. 4.1) и включала в себя мероприятия:

- проектирование карты результатов обучения по дисциплине в виде связного графа, терминальные вершины которого представляют декомпозированные до элементарного уровня знания, умения и навыки;
- сопоставление результатов обучения с учебными модулями, определение типа и назначения модулей;
- формирование описаний для модулей;
- разработка учебного контента и контролирующих материалов для модулей с учетом вариативности и назначения модулей.

Были определены базовые результаты обучения (РО):

РО 1. Формулировать базовые понятия и законы химии.

РО 2. Выполнять химические расчеты, используя базовые понятия и законы химии.

РО 3. Проводить лабораторные исследования с последующей обработкой экспериментальных данных.

Количественные характеристики для раздела «Химическая кинетика»:

- 21 тест;
- 156 вопросов;
- 36 модулей;
- 46 субкомпетенций.

Для каждого модуля авторами были заданы:

– выходные субкомпетенции – результаты, формирование которых обеспечивает модуль. Авторы условились на том, что каждому модулю в соответствии приведена только одна субкомпетенция;

– входные субкомпетенции – результаты, определяющие знания/умения/опыт, которыми должен владеть студент для освоения текущего модуля. Авторы условились на том, что модули могут иметь как несколько входных субкомпетенций, так и не иметь ни одной (для ряда тем).

Для некоторых модулей была реализована вариативность форматов в представлении содержания.

Традиционная структура курса «Общая химия» включает в себя 3 вида аудиторных занятий: лекции, практические и лабораторные занятия. В ходе изучения дисциплины студентами каждый вид аудиторных занятий дополняется работой с адаптивным электронным курсом в САО Nexbe.

Работа с адаптивным курсом для обучаемого начинается с диагностического тестирования на знание материалов изучаемого раздела, полученных в ходе очных занятий. После чего формируется начальная траектория на основе выборки соответствующих модулей теоретического и практического блоков. После освоения первого модуля траектории оценивается уровень сформированности соответствующего ему результата обучения. Далее следует актуализация модели обучаемого – в историю добавляется информация о текущем уровне знаний. Если освоение курса не окончено, происходит актуализация траектории и предлагается

следующий модуль для изучения. По мере прохождения определенного количества теоретических блоков в траекторию начинают встраиваться модули практического и лабораторного блоков.

По завершении прохождения курса на основе зафиксированных в системе данных были построены фактические маршруты обучения. Они показали, что:

- нарушений в логической последовательности модулей нет;
- студенты не имели доступ к следующему модулю без достижения требуемой оценки за тест;
- в зависимости от состояния кривых забываний в траектории были включены модули для повторения ранее изученного материала.

Эти факты свидетельствуют о том, что модель адаптивного обучения построена верно.

В качестве испытуемых, проходящих обучение с применением технологии адаптивного обучения, были взяты две учебные группы (выборка в общем количестве 20 человек) из одного потока одного направления подготовки, обучающиеся у одного преподавателя.

Вторая пара групп (другие 20 человек) из того же потока проходила обучение без применения технологии адаптивного обучения. Оценка полученных знаний производилась посредством контрольных мероприятий у всех четырех групп одинаковым способом. Сравнительная диаграмма успеваемости представлена на рисунке 4.2. Ось ординат показывает долю обучающихся (в процентах), успешно справившихся с контрольными мероприятиями.

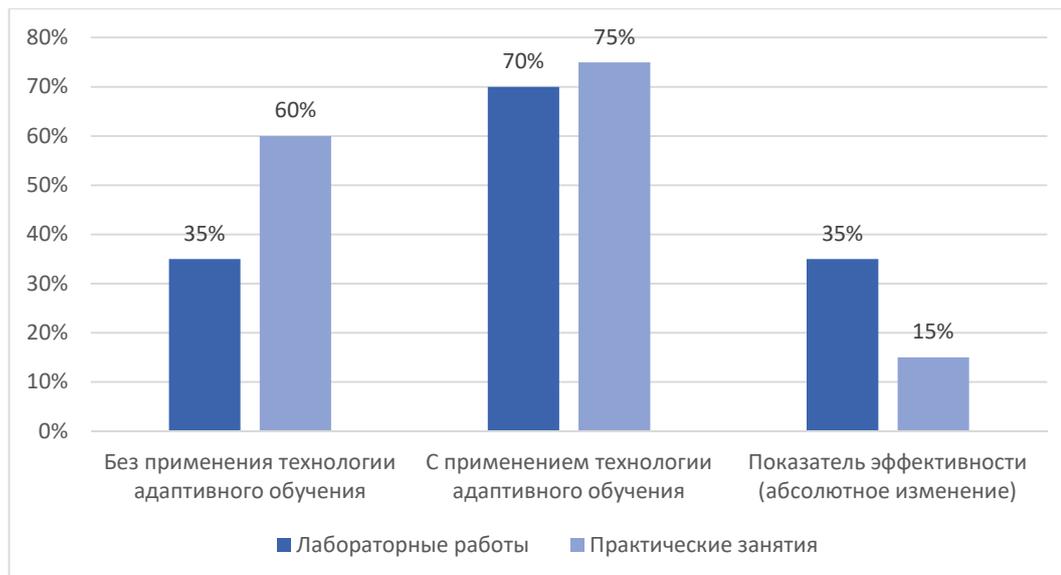


Рисунок 4.2 – Сравнительная диаграмма успеваемости обучающихся из разных групп

Из диаграммы видно, что успеваемость обучающихся из экспериментальной группы (с применением адаптивного обучения) по двум блокам мероприятий выше, чем у студентов, осваивающих программу в традиционном формате. Абсолютный прирост доли студентов, справившихся с лабораторным блоком, составил 35%, с блоком практических занятий – 15%.

Проверка результатов освоения теоретического, практического и лабораторного блоков, а также результаты проведения аудиторных практических занятий позволили преподавателю сделать следующие выводы:

1. В ходе практических и лабораторных занятий обучающимися было продемонстрировано хорошее понимание теоретического материала, который традиционно является сложным для усвоения.

2. Перенос решения прикладных задач 1-го и 2-го уровней сложности в онлайн-режим обеспечивает формирование устойчивых навыков и умений в решении типовых задач, а также существенный прирост аудиторного времени для рассмотрения задач повышенного уровня сложности.

3. Благодаря возможности встраивать в траекторию модули, относящиеся к школьной программе, к концу первого семестра у большинства студентов нивелируются пробелы в знаниях школьной программы. Ключевым моментом

здесь является факт реализации этого процесса в онлайн-режиме, без участия преподавателя.

4. Адаптивное обучение способствует повышению мотивации студентов. Отмечены повышение успеваемости и появление интереса у студентов, которые до поступления в университет не были ориентированы на изучение химии.

4.3 Создание адаптивного электронного курса «Информатика» и его апробация в учебном процессе

На факультете дистанционного обучения (ФДО) ТУСУР САО Nexbe была внедрена в промышленную эксплуатацию (приложение Г). С помощью данной системы был разработан и внедрен в учебный процесс электронный курс «Информатика», состоящий из 6 разделов, в котором первые два раздела («Общее представление об информации» и «Кодирование информации») были модифицированы на основе предшествующей версии курса и реализованы по технологии адаптивного обучения, остальные четыре раздела («Принципы работы компьютера», «Программное обеспечение», «Телекоммуникации», «Основы защиты информации») модификации не подвергались.

Модификация первых двух разделов курса осуществлялась по разработанной методике и включала в себя мероприятия, аналогичные описанным в п. 4.2.

Были определены базовые результаты обучения:

РО 1. Формулировать базовые понятия и свойства информации.

РО 2. Осуществлять кодирование информации.

Количественные характеристики курса:

- 56 тестов;
- 382 вопроса;
- 45 модулей;
- 54 субкомпетенции.

Каждому модулю назначалось не более 5 входных и не более 4 выходных субкомпетенций.

По завершении прохождения теоретического блока (для всех шести разделов курса) предусмотрено выполнение компьютерных тестов на знание соответствующих разделов, каждый тест состоит из 20 вопросов.

Учебный процесс на ФДО ТУСУР характеризуется следующими особенностями:

1) набор на обучение ведется непрерывно, студенты учатся по индивидуальным планам, планирование и выполнение учебных мероприятий (изучение лекционного блока, контроль знаний) производится студентами самостоятельно;

2) аттестация по дисциплине (зачет или экзамен) производится в формате компьютерного тестирования. Также предусмотрена промежуточная аттестация в виде компьютерных контрольных работ по соответствующим темам дисциплины;

3) обеспечением дисциплины является электронный курс, который работает в асинхронном режиме (т.е. без привязки к датам начала обучения студентов). Один и тот же электронный курс может обеспечивать разные направления подготовки.

Таким образом, основная доля электронных курсов ФДО ТУСУР (в том числе курс «Информатика») эксплуатируется автономно и непрерывно, что позволяет исследовать статистику учебных данных на различных отрезках времени, а не только в периоды классических семестров.

Оценка эффективности адаптивного обучения производилась на основе двух групп. В первую группу вошли 46 человек, завершивших обучение по адаптивному электронному курсу «Информатика» в 2020 г.; вторая группа была сформирована из 46 человек, проходивших обучение в 2017–2019 гг. по электронному курсу «Информатика», версия которого не включала в себя разделы с адаптивным обучением. При формировании групп учитывалось следующее:

- все студенты обучаются по одному направлению подготовки;
- для достижения однородности вторая группа формировалась таким образом, чтобы каждый участник этой группы соответствовал участнику первой группы по результатам прохождения промежуточных тестов по разделам с третьего

по шестой с абсолютной погрешностью 0,5 баллов для каждого теста (из расчета, что максимальный балл за каждый тест равен 10).

Результаты тестирования второй группы по первым двум разделам курса «Информатика» (средний балл):

I раздел – 8,1 баллов;

II раздел – 7,2 баллов.

В результате прохождения первой группой адаптивной части средний балл составил:

I раздел – 8,9 баллов;

II раздел – 8,6 баллов.

Таким образом, для первого раздела технология адаптивного обучения позволила повысить средний балл успеваемости на 9,8%, для второго раздела – на 19,4%. Для первого раздела значение t -критерия Стьюдента составило 5,94 (при числе степеней свободы 90), для второго – 11,62, при этом в обоих случаях рассчитанное значение получилось больше критического, соответственно, наблюдаемые различия статистически значимы (уровень значимости $p < 0,05$).

4.4 Сравнительный анализ инструментальных систем построения адаптивного обучения

На основе системного анализа были сформированы следующие критерии сравнения:

1. *Интеграция с внешними СДО.* Система может быть реализована в качестве самостоятельного клиент-серверного приложения, с возможностью интеграции адаптивных электронных курсов во внешние системы, либо являться надстройкой над системой дистанционного обучения.

2. *Поддержка русского языка интерфейса.* Необходима для эффективного применения системы на территории РФ.

3. *Использование контента в разных курсах.* Модель контента должна быть основана на модульном подходе [12], где каждый модуль характеризуется

входными и выходными компетенциями. При этом должна быть обеспечена возможность использовать один и тот же модуль при построении множества различных адаптивных электронных курсов.

4. *Автоматическое формирование траектории.* На основе информации о входных и выходных компетенциях модулей система должна посредством заложенных алгоритмов производить формирование траектории обучения без необходимости задавать последовательность прохождения модулей вручную. В качестве модели навигации может использоваться комбинация моделей адаптивных гипермедиа-систем [18].

5. *Адаптация траектории на основе изменения состояний модели обучаемого (МО).* МО должна включать в себя состояния всех компетенций. При этом система должна в автоматическом режиме производить оценку анализа состояний МО и осуществлять изменение траектории обучения на основе актуальной информации.

6. *Поиск контента по базе данных системы.* Поиск для авторов и преподавателей может являться сложной задачей, если базы данных хранят в себе записи множества дисциплин и это хранение не имеет системной организации. Кроме того, поиск может быть более эффективным, если средства измерения знаний, учебные модули и компетенции находятся в связи друг с другом – это даёт возможность разработчику применять семантический поиск.

7. *Поддержка итеративного обучения.* Реализация сценариев повторения ранее изученного материала должна производиться в автоматическом режиме.

8. *Доступ обучаемого к своей истории обучения.* Система должна предоставлять возможность обучаемому анализировать свою траекторию обучения и историю изменения состояний компетенций.

9. *Отчеты для преподавателя о прохождении курса обучаемыми.* Преподаватель должен иметь возможность формировать отчеты по всем обучаемым в адаптивном электронном курсе.

10. Настройки условий прохождения курса. Преподаватель должен иметь возможность задавать условия:

- курс завершается, когда по всем компетенциям происходит достижение заданных уровней;
- курс завершается, когда истекает время, выделенное на его прохождение.

В таблице 4.1 приведена сводная информация по разработанной системе CAO Nexbe в сравнении с рассмотренными ранее аналогичными решениями.

Таблица 4.1 – Сравнение систем адаптивного обучения

Показатель	Cerego	IADLearning	Smart Sparrow	CAO Nexbe
Интеграция с внешними СДО	+	+	+	+
Поддержка русского языка интерфейса	–	–	–	+
Использование контента в разных курсах	–	–	–	+
Автоматическое формирование траектории	–	+	–	+
Адаптация траектории на основе изменения состояний МО	+	–	–	+
Поиск контента по базе данных системы	–	–	–	+
Поддержка итеративного обучения	+	–	+	+
Доступ обучаемого к своей истории обучения	+	+	–	+
Отчеты для преподавателя о прохождении курса обучаемыми	–	–	+	+
Настройки условий прохождения курса	–	–	+	+

На основе таблицы 4.1 можно сделать заключения:

1. Все рассматриваемые программные средства имеют возможность интеграции с системами дистанционного обучения посредством программных

интерфейсов или стандартов передачи учебных данных, таких как LTI [55]. Однако при выстраивании полноценного учебного процесса с применением технологий адаптивного обучения электронная среда, в которой происходит обучение, становится зависимой от внешних ресурсов. В случае с CAO Nexbe необходимые компоненты интегрируются непосредственно в СДО организации и не имеют проблем с языковой локализацией.

2. Возможность применять одни и те же модули в разных учебных дисциплинах упрощает сопровождение базы контента и делает процесс разработки адаптивных электронных курсов более гибким: разработчики-составители могут создавать курсы различной конфигурации под широкий спектр учебных задач. Такие возможности в настоящий момент поддерживаются только в CAO Nexbe.

3. В отличие от аналогов, CAO Nexbe предоставляет как обучаемому, так и преподавателю инструменты анализа успеваемости, включающие в себя функции просмотра истории обучения и построения отчетов.

4. В целом рассмотренные программные средства уступают в функциональности CAO Nexbe и в большей степени могут быть использованы только в качестве инструментов создания вспомогательных элементов тренинга и самоконтроля знаний, интегрируемых в электронные курсы. В то же время на основе CAO Nexbe представляется возможным организовать процесс разработки и эксплуатации адаптивных электронных курсов, выступающих в роли основного обеспечения учебных дисциплин.

5. С учетом большого количества недостатков у существующих инструментальных программных решений и нецелесообразности их применения в учебном процессе нерациональной становится оценка затрат на приобретение данной продукции, а также трудовых затрат на получение результата с помощью этого ПО. Практическая ценность полученной системы CAO Nexbe может быть сравнима с разработкой заказного программного обеспечения, при этом, по некоторым оценкам [56], стоимость разработки системы адаптивного обучения для одной предметной области начинается от 100 млн рублей, что в современных

реалиях для образовательных организаций является мероприятием ограниченной доступности. Более дешевым и осуществимым вариантом реализации адаптивного обучения на сегодняшний день является подбор необходимых инструментальных средств авторской разработки, однако, как уже было показано ранее, ассортимент доступного и функционального ПО для соответствующих задач на рынке адаптивного обучения находится только в стадии зарождения.

Для анализа эффективности промышленной эксплуатации инструментальной системы был проведен сравнительный эксперимент, целью которого являлось определение временных затрат на создание адаптивных электронных курсов. В эксперименте участвовали программные средства Cerego, IADLearning, Smart Sparrow и CAO Nexbe.

Для эксперимента был выбран фрагмент дисциплины «Информатика». В качестве входных данных были взяты:

- теоретическая часть (контент), описывающая 30 модулей (время изучения каждого в среднем занимает 20 мин) в виде документов в формате MS Word;
- 45 тестовых заданий, направленных на проверку владения компетенциями, в формате MS Word (по 1–2 вопроса на каждую компетенцию);
- список из 30 компетенций и их связь с учебными модулями в формате таблицы MS Excel.

В эксперименте рассматривались только необходимые для создания адаптивного электронного курса этапы (действия пользователя). Выбранные программные средства в полной мере не пересекаются по функционалу, в связи с чем некоторые этапы могут быть присущи только конкретному ПО, если они необходимы для достижения требуемого результата.

Замеры времени производились для следующих этапов работ:

1. Разметка документа с теоретическим контентом для импорта в систему (если данный функционал поддерживается).

2. Редактирование сущностей модулей в системе (если невозможно произвести импорт данных либо требуется произвести постобработку после импорта).

3. Редактирование сущностей компетенций в системе (если невозможно произвести импорт данных либо требуется произвести постобработку после импорта).

4. Разработка тестовых заданий в среде системы.

5. Редактирование связей между моделями и компетенциями.

6. Редактирование связей между тестами и компетенциями.

Результат эксперимента представлен на рисунке 4.3. Для тех программных средств, для которых некоторые этапы неприменимы, затраты указаны как 0 мин. Из диаграммы видно, что по всем этапам разработки адаптивного электронного курса CAO Nexbe позволяет достичь результат с меньшими затратами.

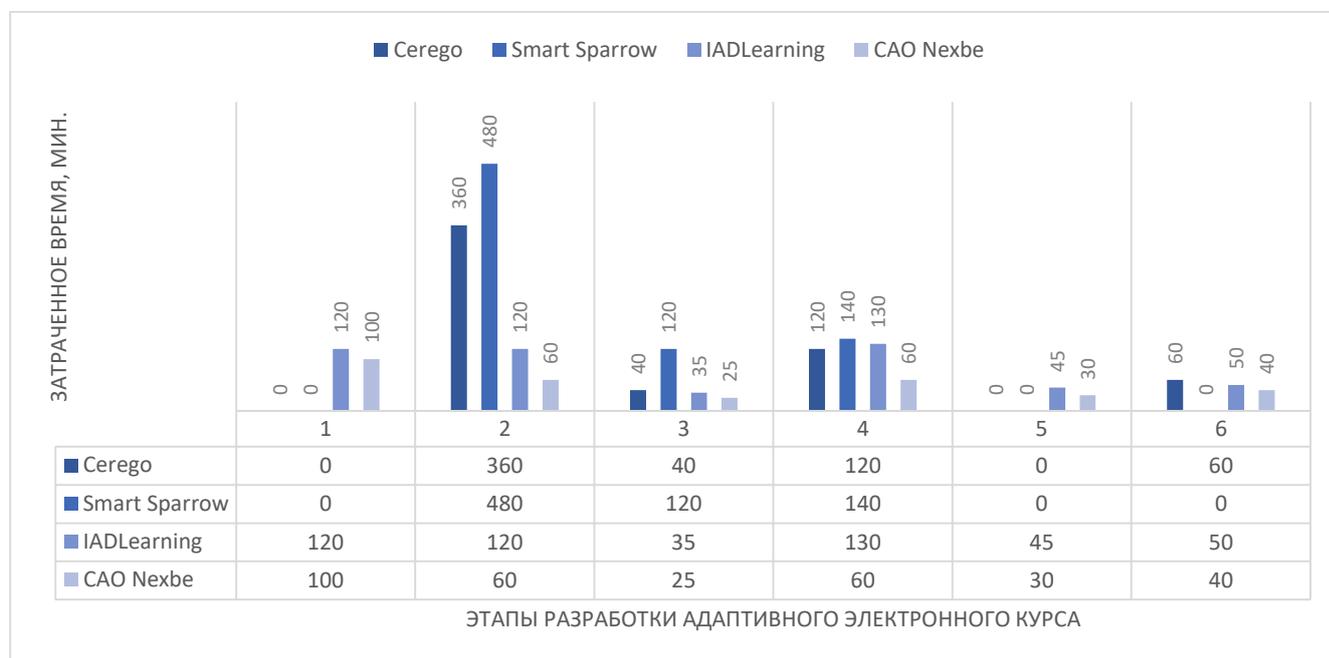


Рисунок 4.3 – Сравнительная диаграмма времени, затраченного на этапы разработки адаптивного электронного курса

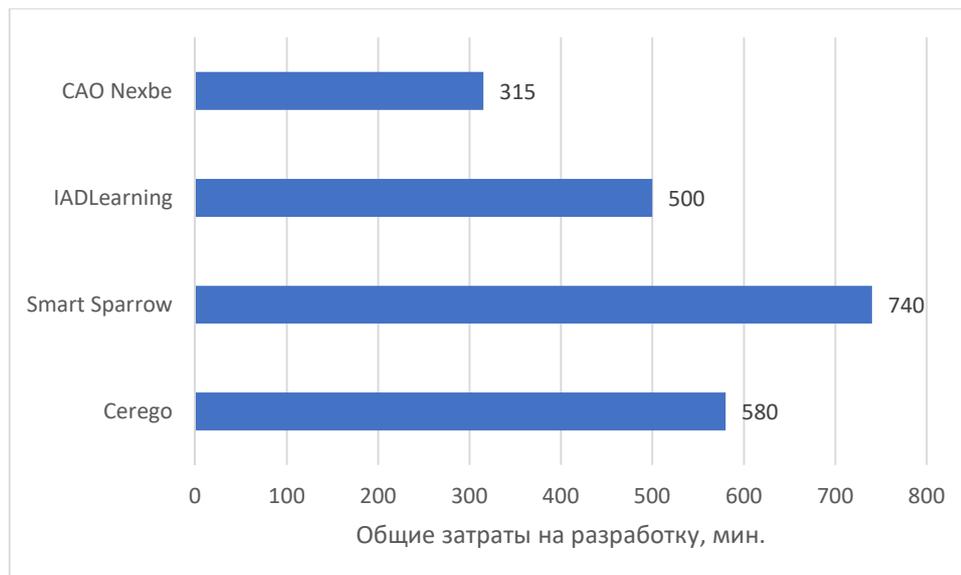


Рисунок 4.4 – Сравнительная диаграмма времени, затраченного на разработку адаптивного электронного курса

На рисунке 4.4 приведены итоговые данные по затратам на разработку адаптивного электронного курса. В сравнении с ближайшим по полученному значению затрат аналогом (IADLearning) сокращение времени на разработку курса при использовании CAO Nexbe составило 37%.

Выводы по главе 4

1. Эксперименты показали, что программная реализация выбранной модели адаптивного обучения произведена корректно. Разработанное программное обеспечение позволяет решать задачу создания адаптивных электронных курсов и их эксплуатации в системе дистанционного обучения.

2. По сравнению с аналогами CAO Nexbe обладает рядом преимуществ и позволяет сократить время на разработку адаптивных электронных курсов.

3. Эффективность и применимость разработанных решений доказана в ходе разработки электронных адаптивных курсов «Общая химия» и «Информатика» и их апробации в реальных учебных процессах НИТУ «МИСиС» и ТУСУР соответственно.

ЗАКЛЮЧЕНИЕ

1. Произведен обзор и анализ моделей и методов адаптивного обучения, рассмотрены инструментальные программные средства реализации адаптивных электронных курсов. Анализ показал, что большинство теоретических исследований вопросов адаптивного обучения не обрели практического воплощения. Известные системы адаптивного обучения в большинстве случаев закрыты. Доступные для анализа системы обладают рядом недостатков.

2. Построена оригинальная модель адаптивного обучения, включающая модель забывания, модель предметной области, модель обучаемого и позволяющая автоматизировать процесс адаптивного обучения в современных системах дистанционного обучения.

3. Построен новый алгоритм определения индивидуальной траектории обучения в системе дистанционного обучения, основанный на разработанной модели адаптивного обучения, позволяющий получить заданный уровень знаний на момент окончания курса.

4. Получены оригинальные структура и функционал инструментальной системы, представляющие собой двухсерверную конфигурацию (сервер системы дистанционного обучения с интегрированным плагином адаптивного обучения, а также сервер, исполняющий алгоритм адаптивного обучения) и обеспечивающие реализацию полученных моделей и алгоритмов адаптивного обучения в СДО.

5. Произведено сравнение полученной инструментальной системы с аналогами. Сравнение показало, что рассмотренные аналоги уступают в функциональности CAO Nexbe и в большей степени могут быть использованы только в качестве инструментов создания вспомогательных элементов тренинга и самоконтроля знаний, интегрируемых в электронные курсы. CAO Nexbe позволяет сократить время на разработку адаптивных электронных курсов.

6. Создана новая методика построения адаптивных электронных курсов на основе инструментальной системы, включающая следующие основные этапы: получение списка компетенций, разработка учебных модулей, разработка тестовых

заданий, запуск адаптивного электронного курса, и позволяющая уменьшить время и материальные затраты на реализацию адаптивного обучения.

7. Разработаны адаптивные электронные курсы «Химия» и «Информатика», которые были внедрены в учебный процесс Национального исследовательского технического университета «МИСиС» и Томского государственного университета систем управления и радиоэлектроники соответственно. Результаты внедрения показали правильность выбранных решений и позволили повысить успеваемость студентов, проходивших обучение с применением технологий адаптивного обучения, обеспечили гибкий подход к разработке адаптивных электронных курсов, обеспечили возможность проведения глубокого анализа индивидуальных траекторий.

ЛИТЕРАТУРА

1. Ужва, А.Ю. Метод адаптивного поиска образовательных ресурсов на основе онтологической модели представления знаний и алгоритма рассуждений по прецедентам : автореф. дис. ... канд. техн. наук : специальность 05.13.01 / А.Ю. Ужва. – Волгоград, 2013. – 24 с.
2. Соколов, Н.К. Адаптивные среды создания образовательных ресурсов для системы повышения квалификации и переподготовки персонала промышленных предприятий : автореф. дис. ... канд. техн. наук : специальность 05.13.06 / Н.К. Соколов. – М., 2012. – 27 с.
3. Живенков, А.Н. Реализация информационной адаптивной системы обучения на базе LMS MOODLE / А.Н. Живенков, О.Г. Иванова // Вестник Воронежского государственного университета. Серия: системный анализ и информационные технологии. – 2010. – № 2 – С. 88–92.
4. Касьянов, В.Н. Дистанционное обучение: методы и средства адаптивной гипермедиа / В.Н. Касьянов, Е.В. Касьянова // Программные средства и математические основы информатики / под ред. В.Н. Касьянова. – Новосибирск, 2004. – С. 80–141.
5. Brusilovsky, P. Adaptive Hypermedia / P. Brusilovsky // User Modeling and User-Adapted Interaction. – 2001. – Vol. 11. – P. 87–110.
6. Гура, В.В. Теоретические основы педагогического проектирования личностно-ориентированных электронных образовательных ресурсов и сред / В.В. Гура. – Ростов н/Д : Изд-во ЮФУ, 2007. – 320 с.
7. Жажа, Е. Ю. Мониторинг и управление кадровым потенциалом предприятия на основе моделирования процессов забывания и научения : автореф. дис. ... канд. техн. наук : специальность 05.13.06 / Е. Ю. Жажа. – М., 2013. – 23 с.
8. Башмаков, А. Разработка компьютерных учебников и обучающих систем / А. Башмаков, И. Башмаков. – М. : Информационно-издательский дом «Филинь», 2003. – 616 с.

9. Ковалев, И.В. Системные аспекты организации и применения мультилингвистической адаптивно-обучающей технологии / И.В. Ковалев, М.В. Карасева, Е.А. Суздалева // Образовательные технологии и общество (Educational Technology & Society). – 2002. – № 5. – С. 198–212.
10. Стебеньяева, Т.В. Формализация дидактического контента адаптивных электронных образовательных ресурсов на основе использования стандартов / Т.В. Стебеньяева, Т.С. Ларина // Экономика и социум. – 2013. – № 1. – С. 739–747.
11. Advanced Distributed Learning Initiative. SCORM Overview [Electronic resource]. – URL: <https://adlnet.gov/projects/scorm>, free access (accessed 24.11.2020).
12. Норенков, И.П. Технологии разделяемых единиц контента для создания и сопровождения информационно-образовательных сред / И.П. Норенков // Информационные технологии. – 2003. – № 8. – С. 34–39.
13. Hammad, R. A hybrid e-learning framework: Process-based, semantically-enriched and service-oriented [Electronic resource] / R. Hammad // University of the West of England. – URL: <http://eprints.uwe.ac.uk/35001/>, free access (accessed 24.11.2020).
14. Норенков, И.П. Адаптивные среды создания образовательных ресурсов / И.П. Норенков, Н.К. Соколов, М.Ю. Уваров ; МГТУ им. Н.Э. Баумана // Наука и образование : электронный журнал. – 2009. – № 3. – Режим доступа: <http://technomag.bmstu.ru/doc/115688.html> (дата обращения: 27.02.2017).
15. Осин, А.В. Открытые образовательные модульные мультимедиа системы / А.В. Осин. – М. : Агентство «Издательский сервис», 2010. – 328 с.
16. Кречетов, И.А. Об одном алгоритме адаптивного обучения на основе кривой забывания / И.А. Кречетов, В.В. Кручинин // Доклады ТУСУРа. – 2017. – № 1 (20). – С. 75–80.
17. Лащенко, Н.И. Построение модели обучаемого в автоматизированной обучающей системе / Н.И. Лащенко // Вестник Таганрогского государственного педагогического института. – 2012. – № 1. – С. 125–130.

18. Волянская Т.А. Методы и технологии адаптивной гипермедиа / Т.А. Волянская // Современные проблемы конструирования программ / под ред. В.Н. Касьянова. – Новосибирск : ИСИ СО РАН, 2002. – С. 38–68.

19. Семикин, В.А. Семантическая модель контента образовательных электронных изданий : автореф. дис. ... канд. техн. наук : специальность 05.13.18 / В.А. Семикин. – Тюмень, 2004. – 21 с.

20. Ярных, Ю.А. Структурированная семантическая модель контента текстов научно-теоретического характера: автореф. дис. ... канд. техн. наук / Ю.А. Ярных. – М. : Изд-во Рос. акад. гос. службы при Президенте РФ, 2005. – 25 с.

21. Соколов, Н.К. Синтез оптимальных траекторий обучения [Электронный ресурс] / Н.К. Соколов // Наука и образование : электрон. науч.-техн. изд. / МГТУ им. Н.Э. Баумана. – 2012. – № 1 (январь). – Режим доступа: technomag/edu/ru (дата обращения: 22.09.2012).

22. Косоногова, М.А. Метод и средства управления образовательной траекторией в системах электронного обучения : дис. ... канд. техн. наук : специальность 05.13.10 / М.А. Косоногова. – Белгород, 2016. – 160 с.

23. Зайцева, Л.В. Методы и модели адаптации к учащимся в системах компьютерного обучения / Л.В. Зайцева // Educational Technology & Society. – 2003. – Vol. 6, № 4. – P. 204–211.

24. Топчиев, А.В. Модели адаптивного обучения в компьютерных системах / А.В. Топчиев, В.А. Чулюков // Современные наукоемкие технологии. – 2010. – № 5. – С. 62–68.

25. Силкина, Н.С. Обзор адаптивных моделей электронного обучения / Н.С. Силкина, Л.Б. Соколинский // Вестник ЮУрГУ. Сер. Выч. матем. информ. – 2016. – № 4 (т. 5). – С. 61–76.

26. Касьянова, Е.В. Адаптивные методы и средства поддержки дистанционного обучения программированию : автореф. дис. ... канд. физ.-мат. наук : специальность 05.13.11 / Е.В. Касьянова. – Новосибирск, 2006. – 27 с.

27. Пахунов, А.В. Разработка принципов структуризации учебно-методических материалов для подготовки специалистов промышленных предприятий в системе электронных образовательных ресурсов : автореф. дис. ... канд. техн. наук : специальность 05.13.06 / А.В. Пахунов. – М., 2010. – 22 с.
28. Буль Е.Е. Сравнительный анализ моделей обучаемого / Е.Е. Буль // Телематика-2003 : труды X Всерос. науч.-метод. конф. (Санкт-Петербург, 14–17 апр. 2003 г.). – СПб. : СПбГУ ИТМО, 2003. – С. 362–364.
29. Brusilovsky, P. Methods and techniques of adaptive hypermedia / P. Brusilovsky // Adaptive Hypermedia and Hypermedia. – Dordrecht : Kluwer Academic Publishers, 1998. – P. 1–43.
30. Brusilovsky, P. Efficient techniques for adaptive hypermedia / P. Brusilovsky // Lect. Notes. Comput. Sci. – 1997. – Vol. 1326. – P. 12–30.
31. Brusilovsky, P. Methods and techniques of adaptive hypermedia / P. Brusilovsky // User Modeling and User-Adapted Interaction. – Vol. 6. – P. 87–129.
32. De Bra, P. Adaptive Hypermedia: From Systems to Framework / P. De Bra, P. Brusilovsky, G.-J. Houben // ACM Computing Surveys. – 1999. – Vol. 31, № 4.
33. De Bra, P. Adaptive Hypermedia on the Web: Methods, techniques and applications / P. De Bra // Proc. of the AACE WebNet'98 Conf. – Orlando, Fl., 1998. – P. 220–225.
34. Корпоративное обучение для цифрового мира / под ред. В.С. Катькало, Д. Л. Волкова. – М. : АНО ДПО «Корпоративный университет Сбербанка», 2017. – 200 с.: ил., табл.
35. Why Alta. Knewton, Inc. [Electronic resource]. – URL: <https://www.knewton.com/why-alta/>, free access (accessed 24.11.2020).
36. Higher Education Adaptive Technology at McGraw Hill [Electronic resource]. – URL: <https://www.mheducation.com/highered/learning-solutions/adaptive-learning.html>, free access (accessed 24.11.2020).
37. Khan Academy [Electronic resource]. – URL: <https://ru.khanacademy.org/>, free access (accessed 24.11.2020).

38. Cerego [Electronic resource]. – URL: <https://www.cerego.com/>, free access (accessed 25.11.2020).
39. IADLearning [Electronic resource]. – URL: <https://www.iadlearning.com/>, free access (accessed 25.11.2020).
40. SmartSparrow [Electronic resource]. – URL: <https://www.smartsparrow.com/>, free access (accessed 12.01.2019).
41. Plario [Electronic resource]. – URL: <https://plario.com>, free access (accessed 25.11.2020).
42. Тархов, С.В. Методологические и теоретические основы адаптивного управления электронным обучением на базе агрегативных учебных модулей : автореф. дис. ... д-ра техн. наук : специальность 05.13.10 / С.В. Тархов. – Уфа, 2009. – 35 с.
43. Dilon, J.D. Adaptive Learning: Five Common Misconceptions [Electronic resource] / J.D. Dilon // Learning Solutions. – URL: <https://learningsolutionsmag.com/articles/2399/adaptive-learning-five-common-misconceptions>, free access (accessed 25.11.2020).
44. Кречетов, И.А. Алгоритм генерации последовательности образовательных модулей в технологии получения адаптивного образовательного контента / И.А. Кречетов // Материалы II международного Пospelовского симпозиума «Гибридные и синергетические интеллектуальные системы» / под ред. д-ра техн. наук, проф. А.В. Колесникова. – Калининград : Изд-во БФУ им. И. Канта, 2014. – С. 200–206.
45. Хлопотов, М.В. Модели и алгоритмы интеллектуального анализа образовательных данных для поддержки принятия решений : дис. ... канд. техн. наук : специальность 05.13.06 / М.В. Хлопотов. – СПб., 2014. – 127 с.
46. Харитонов, И.М. Прогнозирование уровня освоения компетенций выпускниками вуза на основе рейтинговых оценок абитуриентов [Электронный ресурс] / И.М. Харитонов, Е.Г. Крушель // Современные проблемы науки и

образования. – 2012. – № 6. – Режим доступа: <http://www.science-education.ru/ru/article/view?id=7531>, свободный (дата обращения: 09.12.2020).

47. Ebbinghaus, H. Memory: A Contribution to Experimental Psychology [Electronic resource] / H. Ebbinghaus. – URL: <https://archive.org/stream/memorycontri00ebbiuoft>, free access (accessed 09.12.2020).

48. Ланге, В.Н. О скорости забывания / В.Н. Ланге // Вопросы психологии. – 1983. – № 4. – С. 142–145.

49. Буймов, А.Г. Вероятностная модель эффекта повторений в обучении / А.Г. Буймов, Б.А. Буймов // Доклады ТУСУР. – 2010. – № 1. – С. 236–242.

50. Moodle [Electronic resource]. – URL: <https://moodle.org/>, free access (accessed 24.11.2020).

51. Moodle Coding Guidelines [Electronic resource]. – URL: <https://docs.moodle.org/dev/Coding>, free access (accessed 17.12.2020).

52. OWL Web Ontology Language Overview W3C Recommendation. [Electronic resource]. – URL: <http://www.w3.org/TR/owl-features>, free access (accessed 03.06.2013).

53. Бертран, М. Объектно-ориентированное конструирование программных систем / М. Бертран. – М. : Русская Редакция, 2005. – 1200 с.

54. Кречетов, И.А. Раскрываем потенциал адаптивного обучения: от разработки до внедрения / И.А. Кречетов, М.Ю. Дорофеева, А.В. Дегтярев // Материалы Международной конференции eLearning Stakeholders and Researchers Summit. – 2018. – С. 76–85.

55. Learning Tools Interoperability. [Electronic resource]. – URL: <https://www.imsglobal.org/activity/learning-tools-interoperability>, free access (accessed 15.01.2021).

56. Томский курс по адаптивной математике начали покупать российские вузы [Электронный ресурс] // Сайт РИА «Томск». – Режим доступа: <https://www.riatomsk.ru/article/20190826/kurs-po-adaptivnoj-matematike-plario-tgu-enbisys/>, свободный (дата обращения: 15.01.2021).

57. Hssina, B. A Personalized Pedagogical Objectives Based on a Genetic Algorithm in an Adaptive Learning System / B. Hssina, M. Erritali // *Procedia Computer Science*. – 2019. – Vol. 151. – P. 1152–1157. – doi: 10.1016/j.procs.2019.04.164.
58. Kerr, P. Adaptive learning / P. Kerr // *ETL Journal*. – 2016. – Vol. 70 (1). – P. 88–93. – doi:10.1093/elt/ccv055.
59. A tutorial on machine learning in educational science / L. Kidzinski [et al.] // *State-of-the-Art and Future Directions of Smart Learning (Lecture Notes in Educational Technology)* / Y. In Li et al. (eds.). – Springer, Singapore, 2015. – P. 453–459.
60. Martin, F. Adaptive learning modules / F. Martin, D. Markant // *The SAGE encyclopedia of higher education* / M.E. David, M.J. Amey (eds.). – London : Sage, 2020. – P. 2–4.
61. Газизов, Т.Т. Методология, алгоритмы и программное обеспечение для комплексной оптимизации элементов радиоэлектронных устройств : дис. ... д-ра техн. наук : специальность 05.13.18 / Т.Т. Газизов. – Томск, 2017. – 316 с.
62. Bontchev, B. Courseware Adaptation to Learning Styles and Knowledge Level / B. Bontchev, D. Vassileva // *E-Learning – Engineering, On-Job Training and Interactive Teaching*. – 2012. – P. 3–22.
63. Vassileva, D. Adaptive e-learning content design and delivery based on learning styles and knowledge level / D. Vassileva // *Serdica J. Computing*. – 2012. – № 6. – P. 207–252.
64. Esichaikul, V. Student modelling in adaptive e-learning systems / V. Esichaikul, S. Lamnoi, C. Bechter // *Knowledge Management & E-Learning : An International Journal*. – 2011. – № 3(3). – P. 342–355.
65. Кручинин, В.В. Разработка компьютерных учебных программ / В.В. Кручинин. – Томск : Изд-во Том. ун-та, 1998. – 211 с.
66. Панченко, Т.В. Генетические алгоритмы : учеб.-метод. пособие / Т.В. Панченко ; под ред. Ю.Ю. Тарасевича. – Астрахань : АГУ, 2007. – 87 с.

67. Мицель, А.А. Автоматизация разработки компьютерных учебных программ : монография / А.А. Мицель, В.В. Романенко, М.В. Веретенников, А.И. Щербаков. – Томск : Изд-во НТЛ, 2005. – 383 с.
68. Носков, С.И. Моделирование обучения для автоматизированной обучающей системы / С.И. Носков, Ю.И. Петров // Современные технологии. Системный анализ. Моделирование. – 2010. – № 1(25). – С. 200–206.
69. Dyro, A. Adapting To Adaptive Learning [Electronic resource] / A. Dyro // eLearning Industry. – URL: <https://elearningindustry.com/adapting-to-adaptive-learning>, free access (accessed 29.01.2021).
70. Новиков, Д.А. Закономерности итеративного научения / Д.А. Новиков. – М. : Институт проблем управления РАН, 1998. – 77 с.

ПРИЛОЖЕНИЕ А

Экранные формы системы адаптивного обучения

Система адаптивного обучения ▶ Математика

В начало ▶ Мои курсы ▶ Математика

Основные свойства функции (четность, нечетность, периодичность)

Множество X симметрично относительно начала координат, если $(-x) \in X$ для любого элемента $x \in X$. Примеры симметричных относительно начала координат множеств:

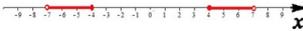
1. $A = [-3; 3]$



2. $B = [-5; -2] \cup [2; 5]$



3. $C = (-7; -4] \cup [4; 7)$



4. $D = (-\infty; -5) \cup (5; +\infty)$
5. $\mathbb{R} = (-\infty; \infty)$

Примеры несимметричных относительно начала координат множеств:

1. $A_1 = [-3; 3]$. Для элемента 3 из множества A_1 элемент (-3) не принадлежит множеству A_1 .



2. $B = [-5; -2] \cup [2; 5]$

[Продолжить](#)

Почему я здесь?
[Выбор модуля](#)
[История](#)
[Успеваемость](#)

Получаемые знания:

- Перечислять основные свойства функции (четность, нечетность, периодичность)

Рисунок А.1 – Интерфейс просмотра модуля курса

Система адаптивного обучения ▶ Математика

В начало ▶ Мои курсы ▶ Математика

Нахождение области определения и проверка основных свойств показательных функций (тест)

Почему я здесь?
[Выбор модуля](#)
[История](#)
[Успеваемость](#)

Получаемые знания:

- Определять $D(f)$, четность, периодичность показательных функций (тест)

Почему я здесь?

В диагностическом тесте были даны неверные ответы на вопросы, проверяющие уровень навыка «Определять $D(f)$, четность, периодичность показательных функций (тест)».

[Прислупить](#)

Рисунок А.2 – Интерфейс модуля без контента

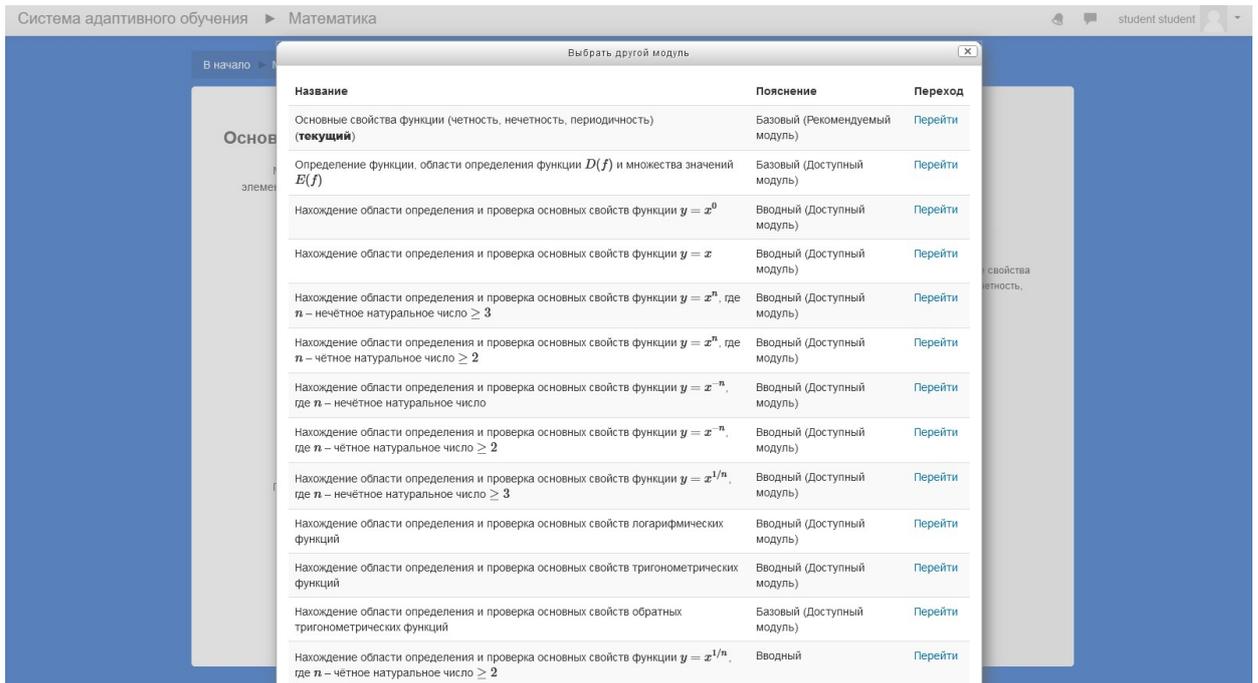


Рисунок А.3 – Интерфейс выбора модуля

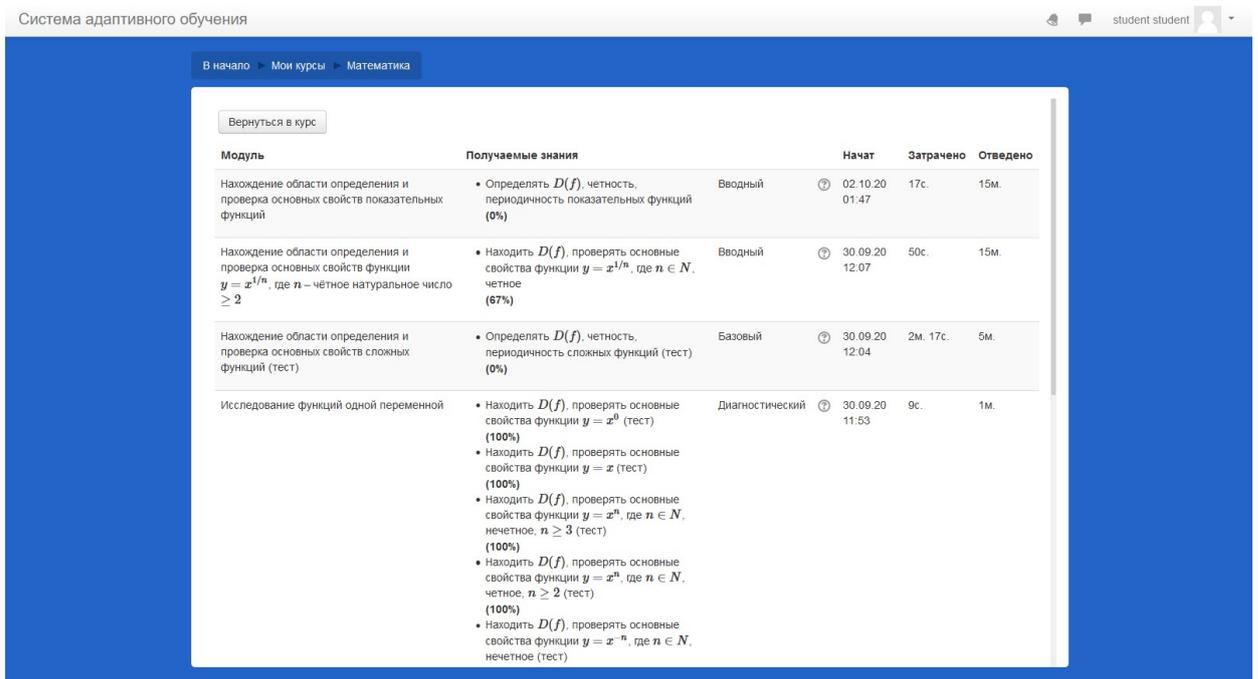


Рисунок А.4 – Интерфейс истории прохождения адаптивного курса

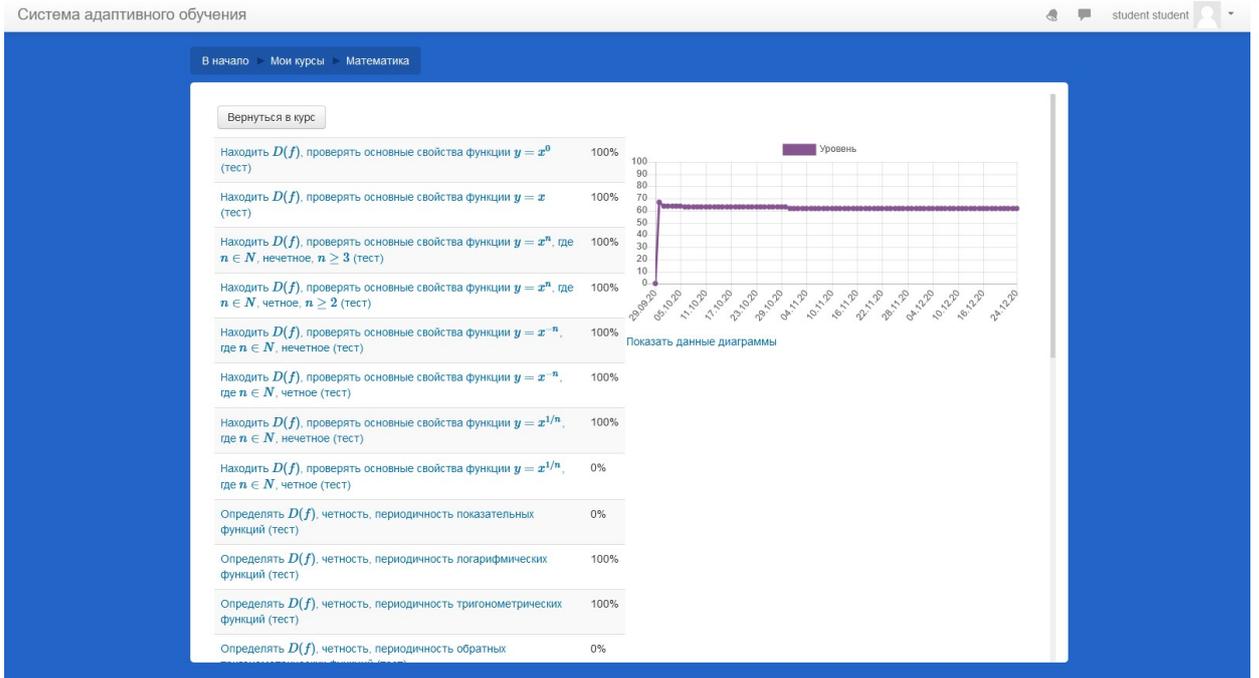


Рисунок А.5 – Интерфейс кабинета успеваемости студента

Система адаптивного обучения

Иван Кречетов

В начало ▶ Курсы ▶ Математика

Общие настройки Категории Компетенции Модули Тесты Назад в курс

core_header

Версия ядра 2

Сервер 172.16.9.33

Порт 21

course_header

Дата начала 29 Сентябрь 2020

Дата окончания 1 Январь 2021

course_normal 60

Свободное перемещение по курсу Да

Сохранить Отмена

Рисунок А.6 – Интерфейс страницы настроек курса

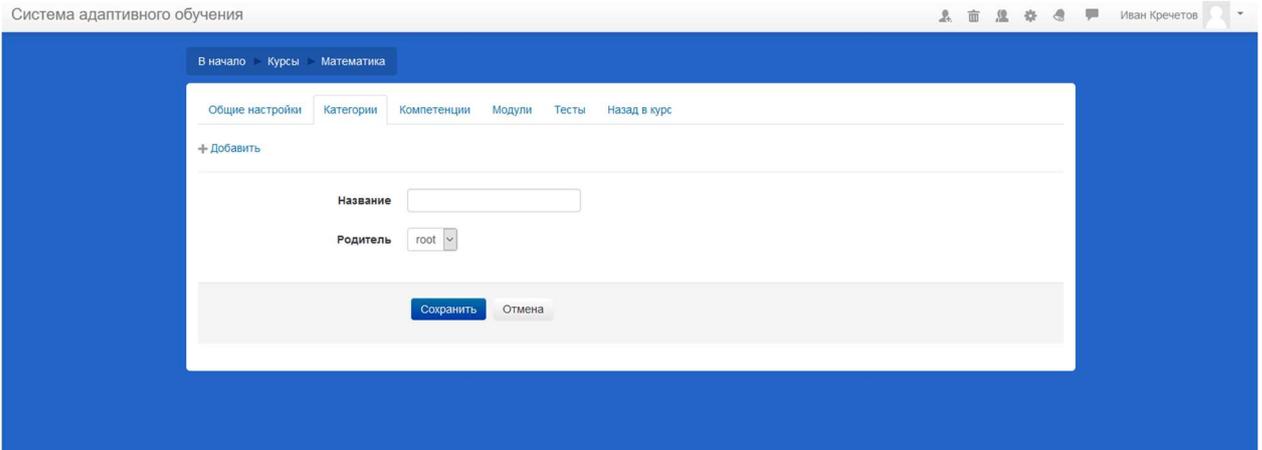


Рисунок А.7 – Интерфейс для редактирования категорий

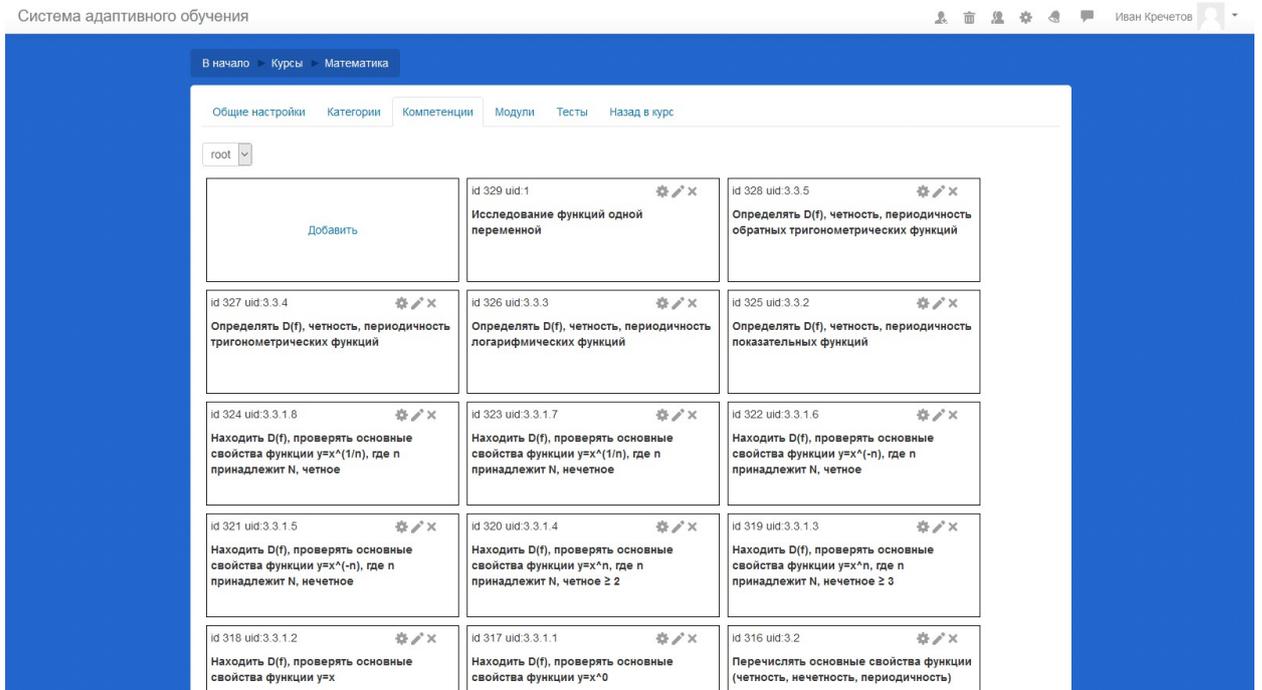


Рисунок А.8 – Интерфейс списка компетенций

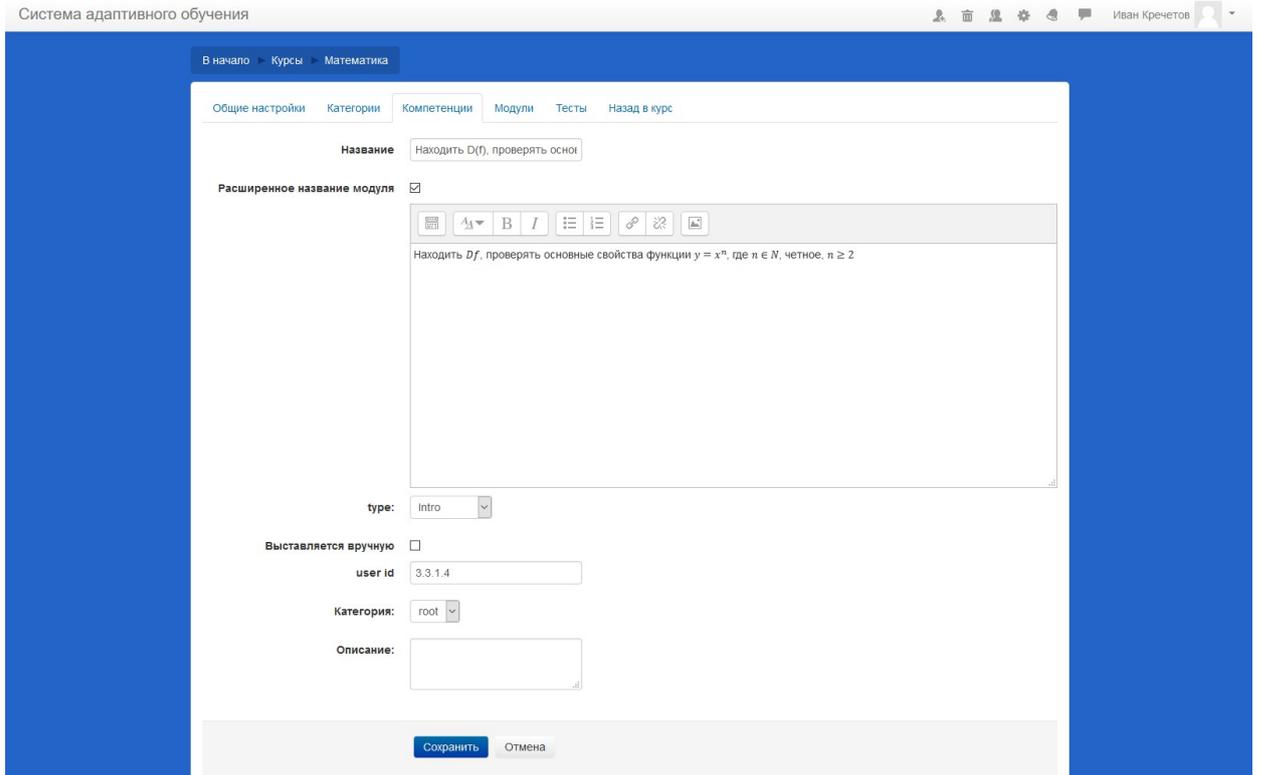


Рисунок А.9 – Интерфейс редактирования компетенции

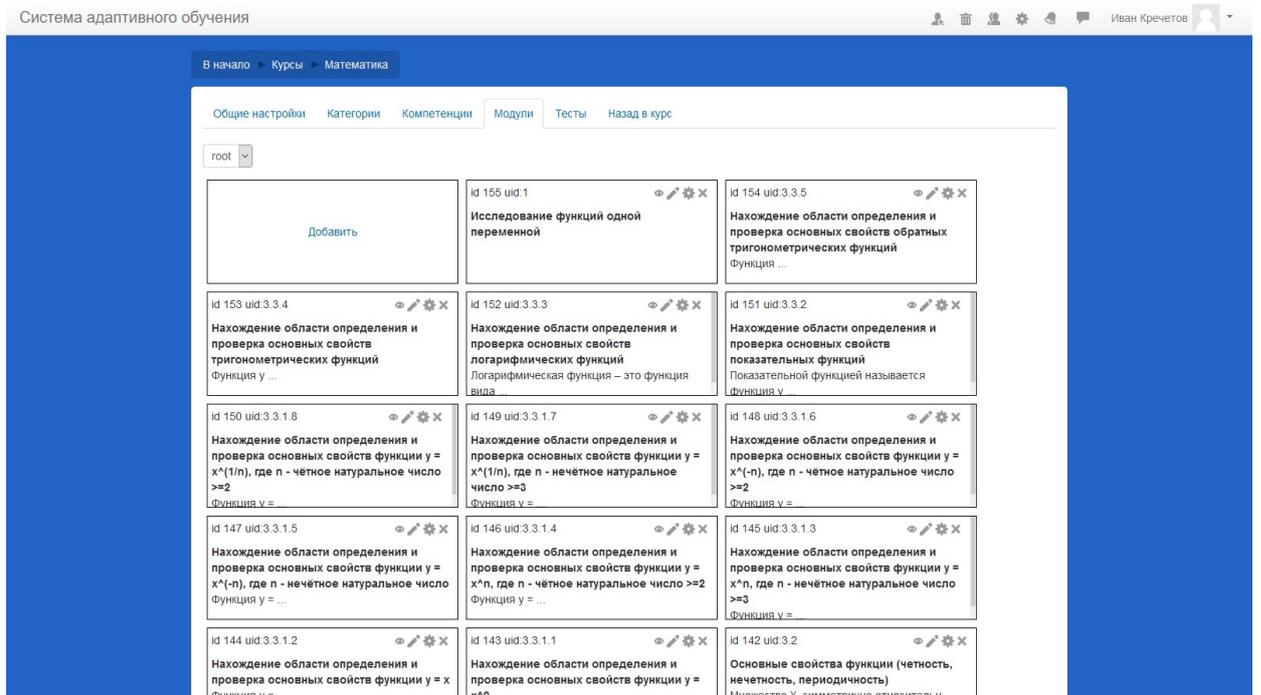


Рисунок А.10 – Интерфейс списка модулей

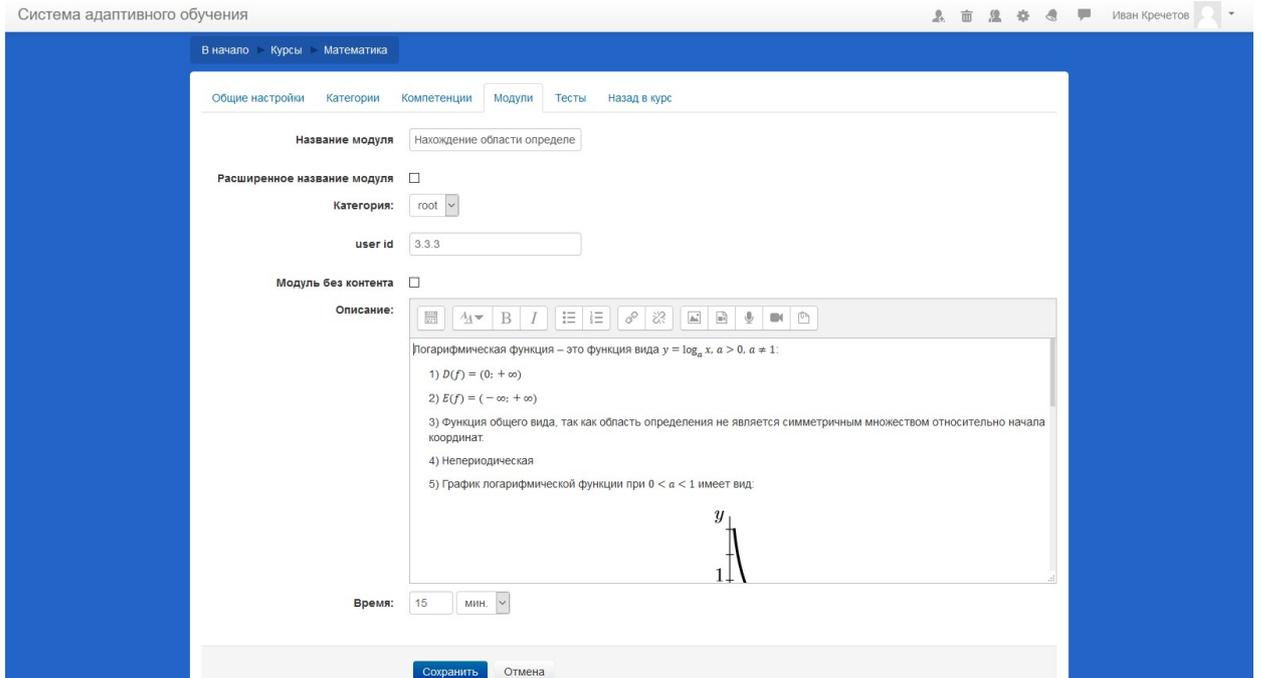


Рисунок А.11 – Интерфейс редактирования модуля

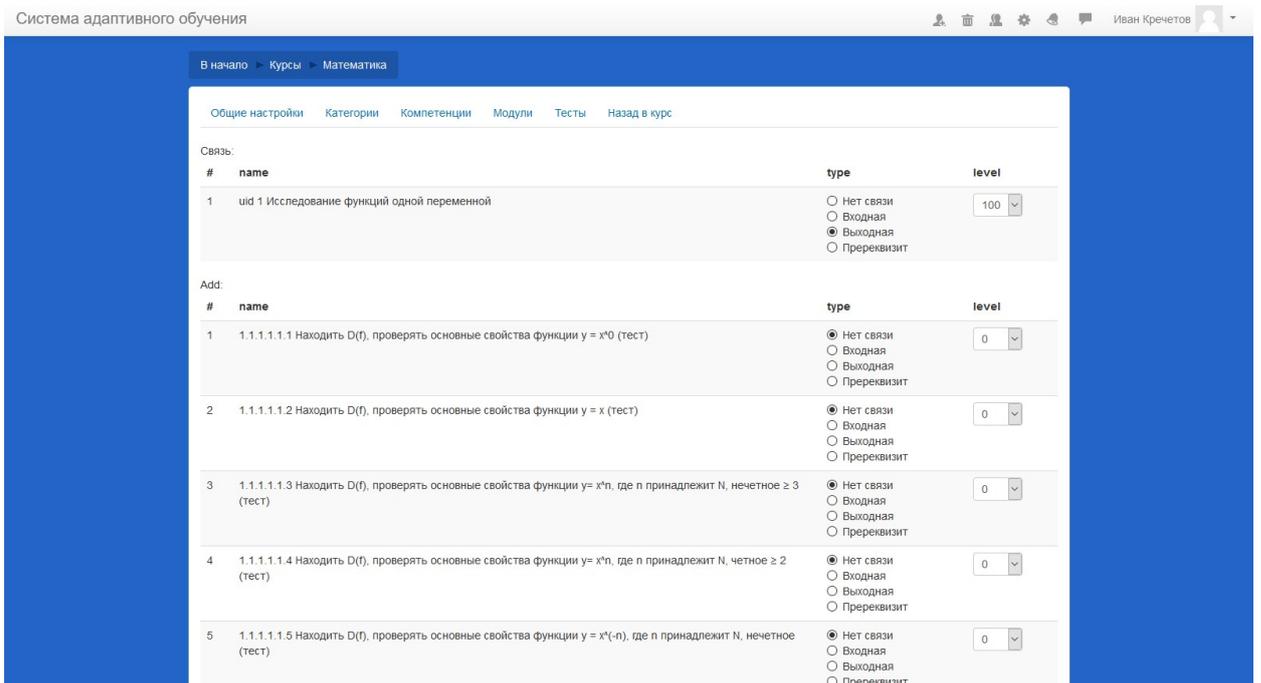


Рисунок А.12 – Интерфейс редактирования связей модулей и компетенций

ПРИЛОЖЕНИЕ Б

Инструкция разработчика по созданию адаптивных электронных курсов с применением системы Nexbe

I Общие понятия

Адаптивный учебный курс – это учебный курс, который в любой момент времени предлагает обучаемому оптимальный (с точки зрения формы представления, текущего уровня знаний и т.д.) элемент контента. Таким образом, для каждого обучаемого формируется персональная *траектория обучения*, исходя из его прогресса в изучении курса, а также индивидуальных показателей скорости усвоения и забывания учебного материала.

1 Модули и компетенции

В основе предлагаемой концепции адаптивного обучения лежат понятия *модуля* и *компетенции*.

Компетенция – это совокупность знаний, умений, навыков в определённой предметной области. *Субкомпетенция* – это составной элемент компетенции (т.е. результат декомпозиции компетенции). Субкомпетенции можно декомпозировать (рис. Б.1.1) и далее, получая таким образом дерево (или лес) компетенций учебного курса. В дальнейшем термин «компетенция» будет применяться как для компетенций, так и для субкомпетенций всех уровней.



Рисунок Б.1.1 – Фрагмент дерева компетенций

Модуль – это неделимая, автономная единица, несущая в себе достаточно информации (текстовой, графической, мультимедиа), чтобы можно было однозначно определить те выходные компетенции, которые приобретет обучаемый, освоив данный модуль (рис. Б.1.2).

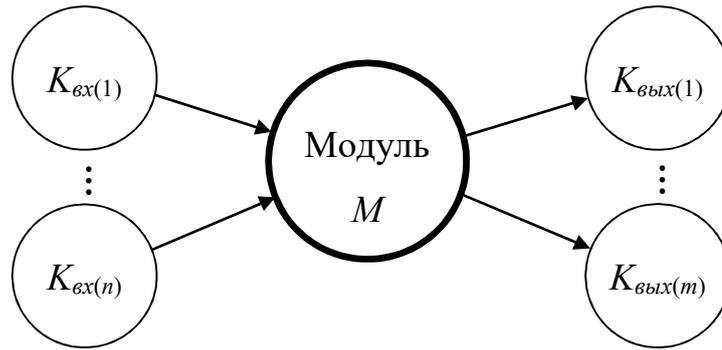


Рисунок Б.1.2 – Общее представление модуля

Результат освоения модуля – это совокупность выходных компетенций. Чтобы оценить результаты освоения модуля, необходимо измерить уровень знаний выходных компетенций, т.е. оценить объём владения обучаемым той или иной рассматриваемой компетенцией. Из вопросов, оценивающих уровень освоения всех выходных компетенций модуля, формируется его *выходной тест*. Если модуль имеет выходную компетенцию, это означает, что он её *обеспечивает*, т.е. является для неё *обеспечивающим модулем*.

Также модуль может (но не в обязательном порядке) иметь входные компетенции. Их наличие означает, что для того, чтобы приступить к изучению данного модуля, обучаемый должен предварительно получить по указанным компетенциям какой-то минимально необходимый уровень знаний. В свою очередь их отсутствие означает, что данный модуль может располагаться в самом начале траектории обучения.

Таким образом, обладая входными и выходными компетенциями, вместе модули образуют связанный двудольный направленный *граф учебного курса* (рис. Б.1.3).

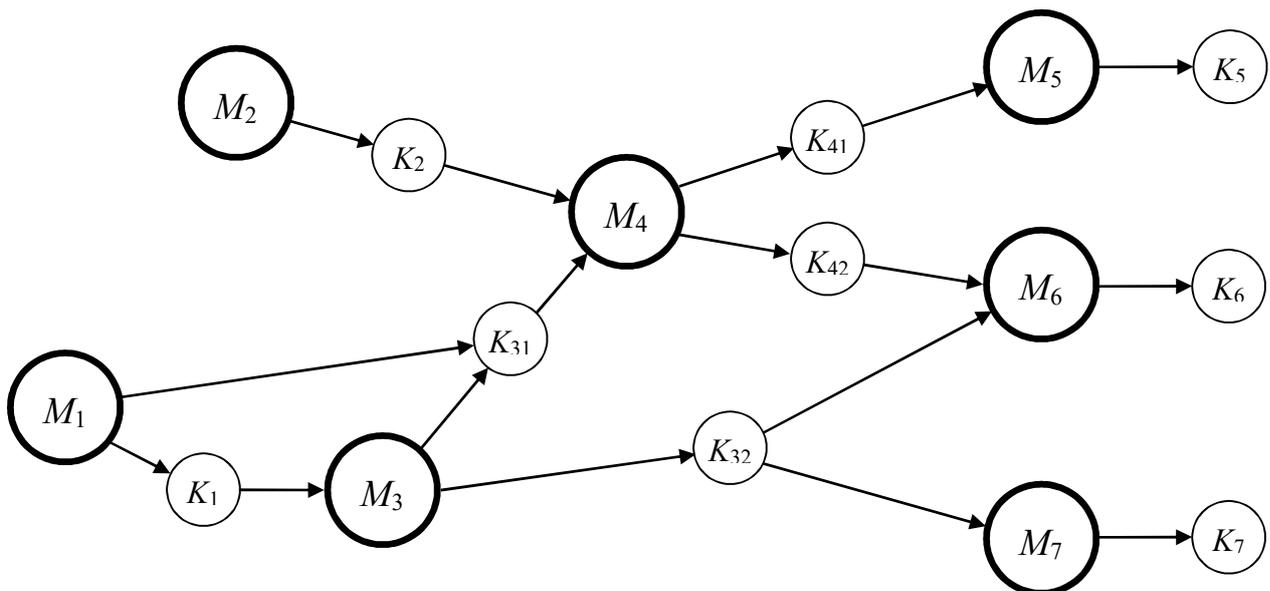


Рисунок Б.1.3 – Пример графа учебного курса

База модулей и база компетенций курса – это хранилища, в которых содержатся, соответственно, все модули и все компетенции курса.

Теперь, введя понятие модуля, можно формально определить использованный ранее термин «траектория обучения». Итак, траектория обучения – это последовательность предлагаемых обучаемому модулей курса. В следующем подразделе говорится о том, как она формируется.

2 Формирование траектории обучения

Формирование траектории определяется рядом факторов:

- 1) графом модулей и компетенций учебного курса;
- 2) текущим уровнем знаний обучаемого по всем компетенциям;
- 3) набором целевых компетенций курса.

Целевые компетенции курса – это набор компетенций, которые должен освоить обучаемый к моменту окончания курса. Например, если для рассмотренного выше в примере курса целевыми были бы только компетенции K_6 и K_7 , то модуль M_5 в траекторию не вошёл, а уровень освоения компетенции K_{41} не влиял на прогресс в обучении (т.к. она не целевая и не требуется для освоения какого-либо модуля, обеспечивающего целевую компетенцию).

По умолчанию, все компетенции курса являются целевыми. Их явное указание требуется лишь в том случае, если набор модулей в базе является избыточным. Например, имеется база модулей по математике, но необходимо сформировать курс, куда войдут только некоторые разделы дисциплины.

В начале обучения уровень знаний по всем компетенциям курса полагается нулевым (на это можно повлиять с помощью диагностики, о которой будет сказано далее в подразделе «Диагностика обучаемых»). Таким образом, если положить, что в рассмотренном выше примере все компетенции курса являются целевыми, траектория обучения может начаться только с модуля M_1 или M_2 (рис. Б.1.4).

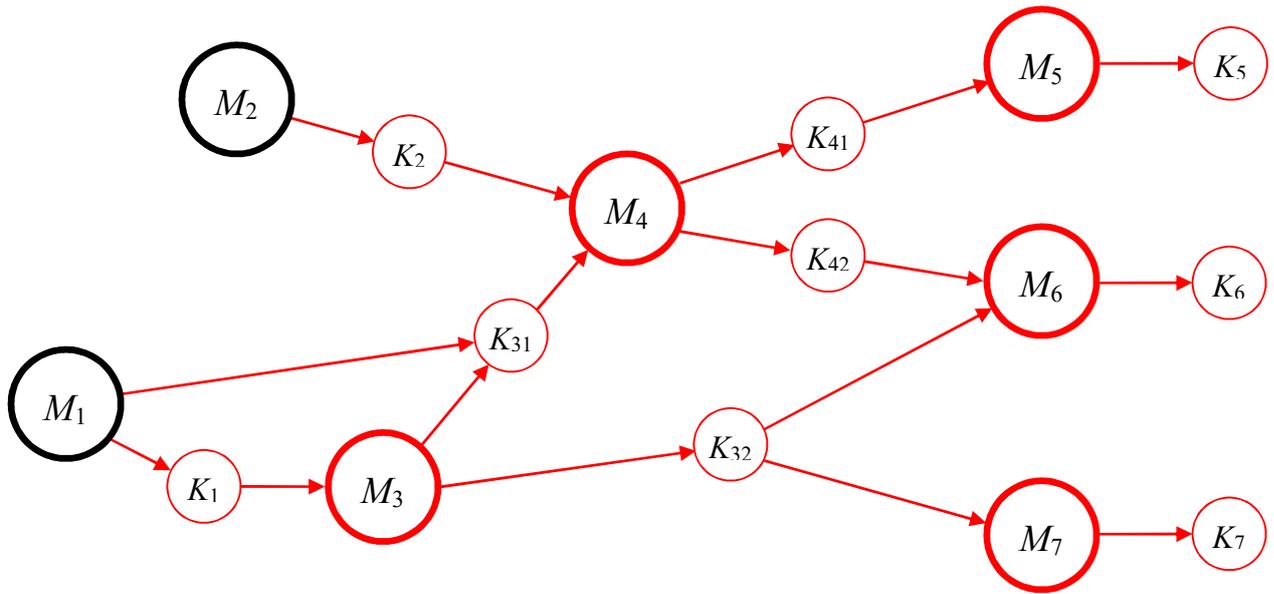


Рисунок Б.1.4 – Доступность модулей курса в начале обучения

Обучаемому будет предложен для изучения один из них (на основе решения генетическим алгоритмом оптимизационной задачи, где критерием является скорейшее освоение целевых компетенций до заданного уровня).

Далее будет приведён пример построения одной из возможных траекторий обучения. Только одной, потому что у каждого обучаемого, как уже было сказано, траектории будут индивидуальными. Это объясняется, во-первых, тем, что каждый обучаемый будет при тестировании показывать результаты, отличные от результатов других обучаемых. Во-вторых, тем, что генетический алгоритм оперирует вероятностями, поэтому его результат, даже для одинаковых исходных данных, может несколько отличаться.

Итак, пусть обучаемому на первом шаге обучения был предоставлен модуль M_1 . После его изучения, ему был предложен тест, чтобы проверить уровень освоения компетенций K_1 и K_{31} . По компетенции K_1 при тестировании был получен результат 90%, а по компетенции K_{31} – 70% (при минимальном требуемом уровне 60%). Тогда на втором шаге (рис. Б.1.5) доступными для изучения будут модули M_2 и M_3 (но не M_4 , т.к. для того, чтобы к нему приступить, необходимо дополнительно освоить компетенцию K_2).

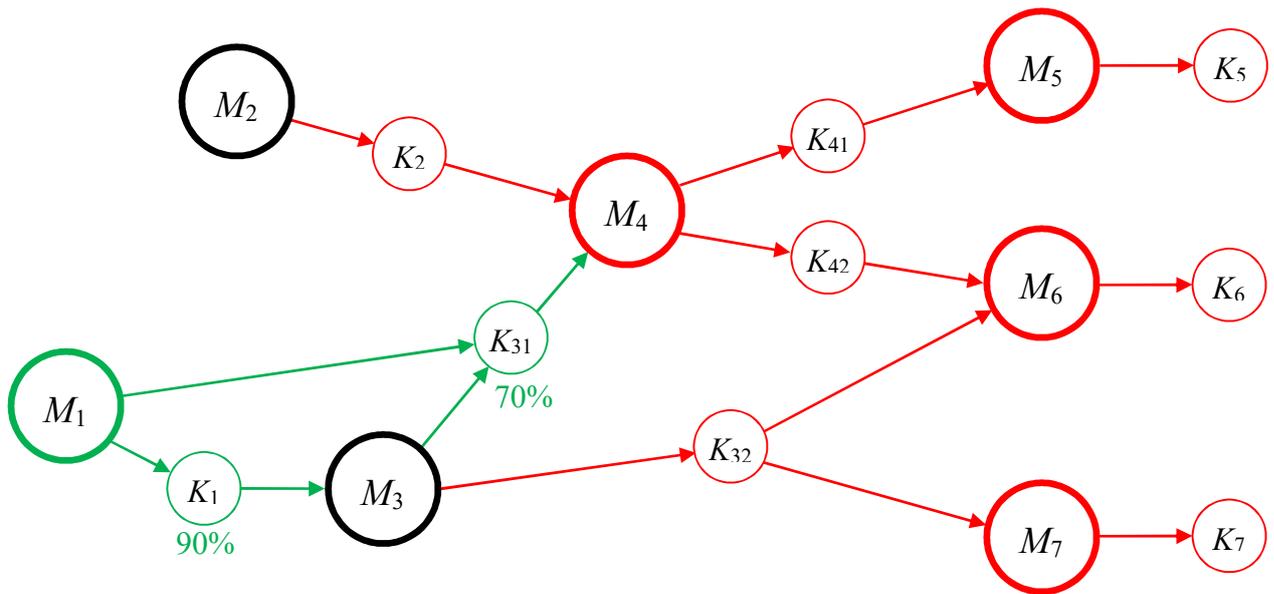


Рисунок Б.1.5 – Доступность модулей на втором шаге

Пусть адаптивный алгоритм следующим предоставил обучаемому модуль M_2 , а при тестировании уровня освоения компетенции K_2 обучаемый показал результат 80%. Тогда на третьем шаге (рис. Б.1.6) доступными для изучения будут модули M_3 и M_4 .

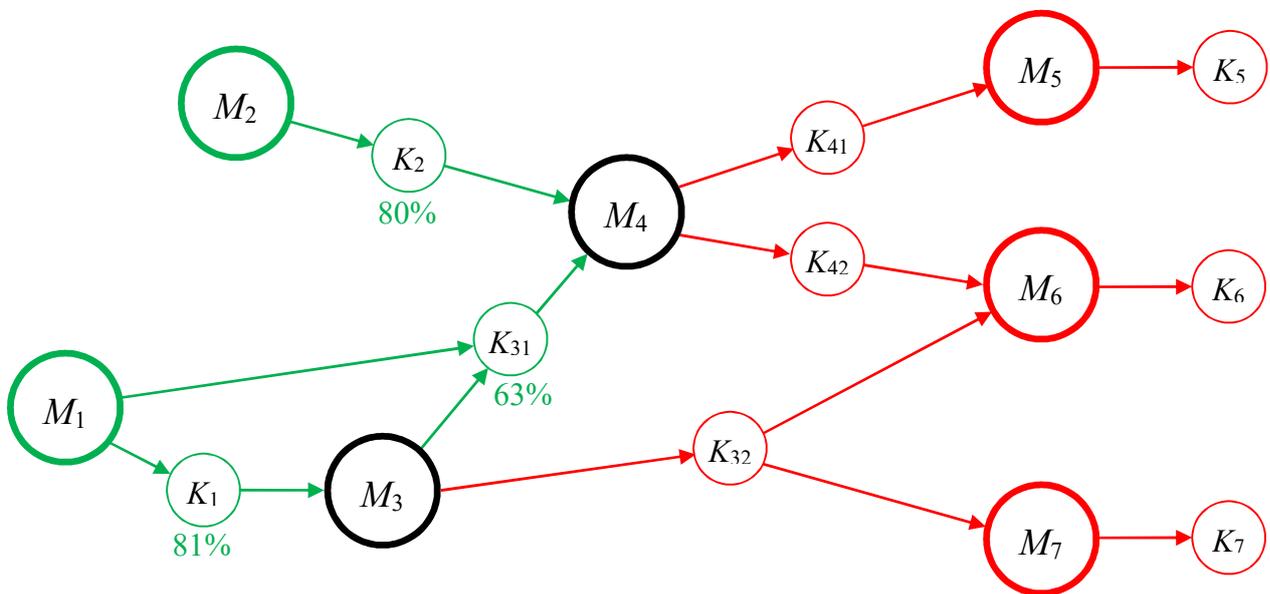


Рисунок Б.1.6 – Доступность модулей на третьем шаге

Как можно заметить, уровень усвоения компетенций K_1 и K_{31} при этом снизился. Система учитывает, что обучаемый со временем забывает полученную информацию. Скорость забывания (определяемая кривой Эббингауза) изначально для всех обучаемых полагается одинаковой, но в процессе обучения уточняется таким образом, что каждый студент по каждой изученной компетенции будет иметь индивидуальные коэффициенты скорости забывания.

Пусть адаптивный алгоритм следующим предоставил модуль M_4 , а при тестировании уровня освоения компетенций K_{41} и K_{42} обучаемый показал результаты 50% (что ниже требуемого уровня) и 100% соответственно. Также, в соответствии с кривой забывания, опустился уровень освоения изученных ранее компетенций K_1 , K_2 и K_{31} , причем у последней, как видно на рисунке Б.1.7, ниже требуемого уровня.

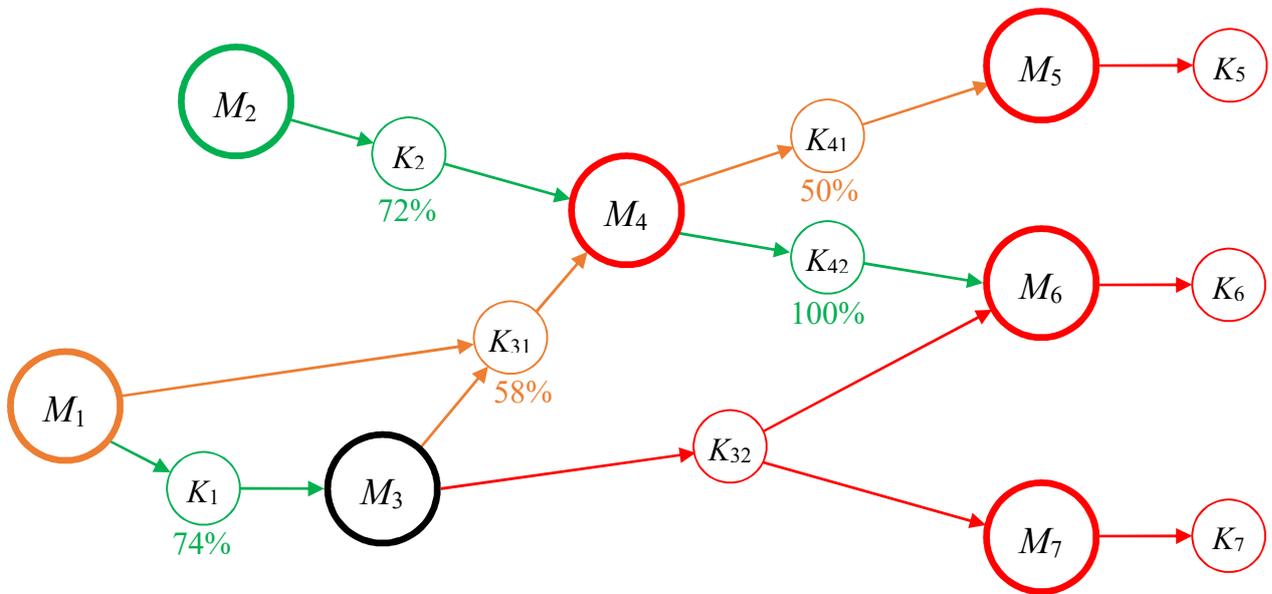


Рисунок Б.1.7 – Доступность модулей на четвертом шаге

Соответственно, модуль M_4 , который необходим для повторного освоения компетенции K_{41} , стал недоступен. Таким образом, на четвертом шаге алгоритм может предложить два варианта дальнейшей траектории:

1. Провести повторное тестирование уровня освоения компетенции K_{31} . Если будет выбран этот вариант, то коэффициенты кривой забывания будут скорректированы: либо в сторону уменьшения скорости забывания, если окажется, что обучаемый забывает информацию медленнее, чем предполагалось (т.е. результат окажется выше 58%), либо в сторону увеличения скорости забывания, если быстрее (т.е. результат окажется ниже 58%). Если при тестировании будет показан уровень выше 60%, модуль M_4 снова станет доступным. Если же ниже 60%, то на следующем шаге обучаемому будет предложен какой-то модуль, обеспечивающий компетенцию K_{31} , для ее повторного изучения. Как видно из графа, это может быть модуль M_1 или M_3 .

2. Предоставить обучаемому модуль M_3 , т.к. он позволит как повторно обратиться к компетенции K_{31} , так и освоить не изученную ранее K_{32} , которая откроет доступ сразу к двум новым модулям – M_6 и M_7 .

Пусть адаптивный алгоритм выбрал второй вариант, а при тестировании уровня освоения компетенций K_{31} и K_{32} обучаемый в обоих случаях показал результат 80%. Тогда на пятом шаге (рис. Б.1.8) доступными для изучения будут модули M_4 (чтобы

повторно изучить неосвоенную компетенцию K_{41}), а также неизученные ранее модули M_6 и M_7 .

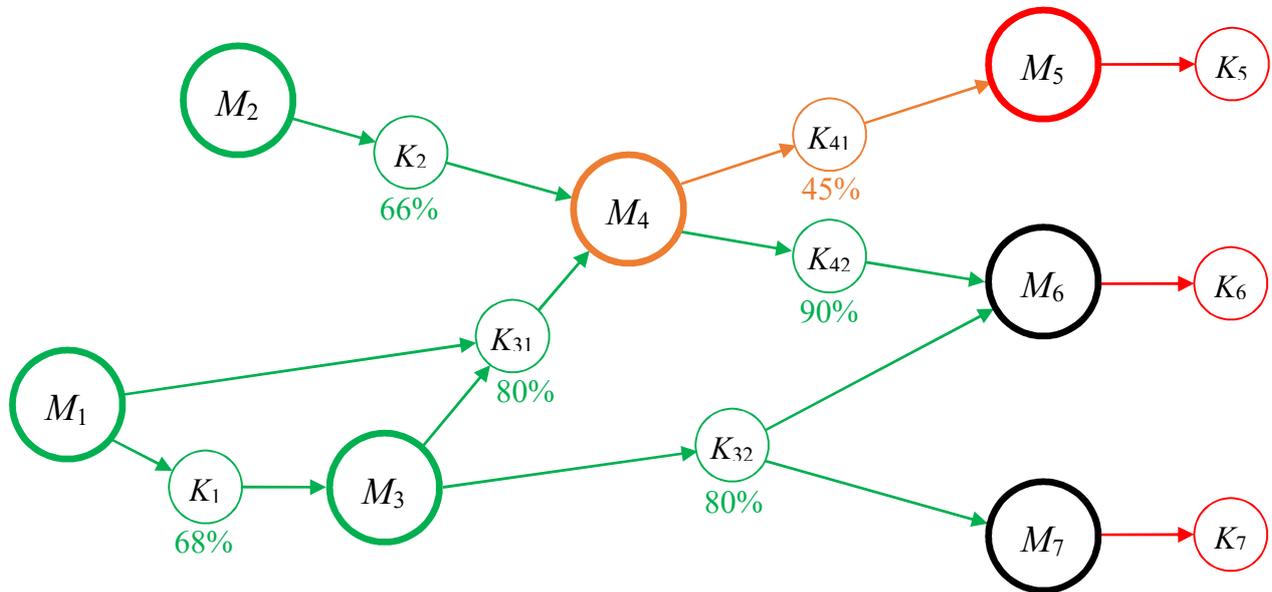


Рисунок Б.1.8 – Доступность модулей на пятом шаге

Пусть алгоритм предоставил обучаемому (рис. Б.1.9) модуль M_6 (получив при тестировании компетенции K_6 уровень 100%) а на следующем шаге – M_7 (получив при тестировании компетенции K_7 уровень 90%).

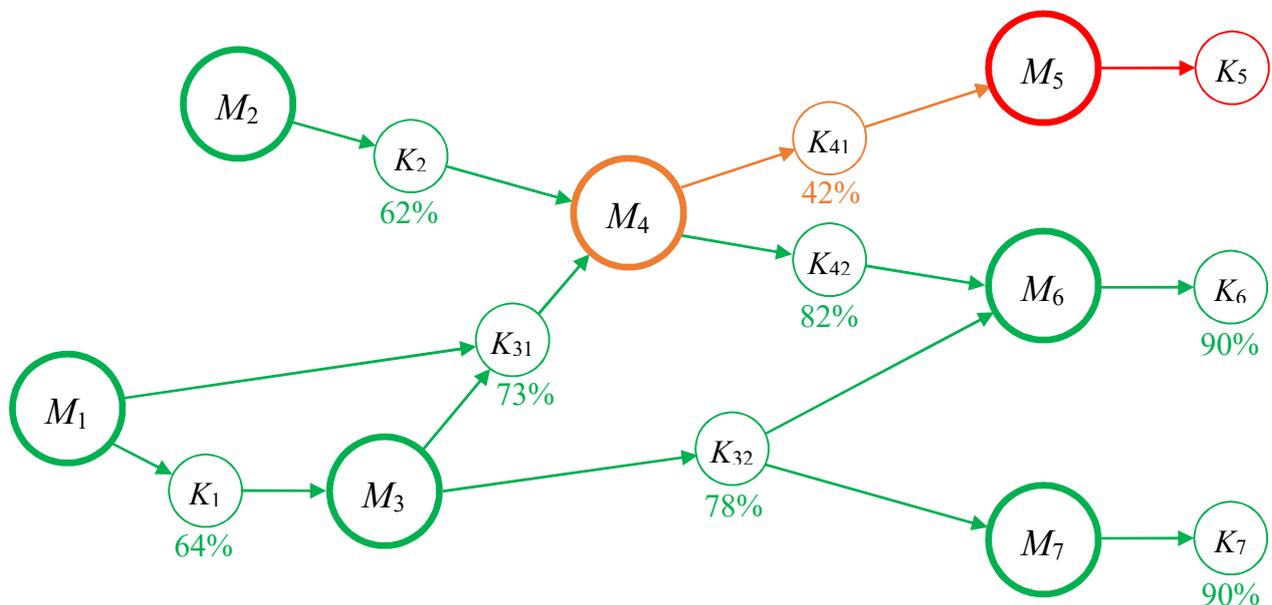


Рисунок Б.1.9 – Доступность модулей на седьмом шаге

Также здесь учтено забывание изученных ранее (на шагах 1–6) компетенций. На следующем шаге обучаемому будет безальтернативно снова представлен модуль M_4 . В этом случае возможно *заикливание* траектории обучения – если по каким-то причинам студент не может освоить компетенцию K_{41} , он не сможет продвигаться

далее в изучении дисциплины, снова и снова возвращаясь к модулю M_4 . Другого же модуля, обеспечивающего данную компетенцию, в графе нет.

Выходом из ситуации является пропуск модуля. Если обучаемый пропускает модуль M_4 , то компетенция K_{41} считается условно освоенной и изучение дисциплины можно продолжать (т.е. следующим ему будет предоставлен модуль M_5). Пропущенные модули включаются в долг, и в будущем обучаемый должен к ним вернуться, чтобы его ликвидировать.

3 Типы модулей и компетенций



Материал данного подраздела является дополнительным. Указание типов компетенций адаптивного курса не является обязательным, но позволяет определить особенности поведения модулей.

Типы компетенций определяют поведение модулей, которые их обеспечивают (формируют). Тип компетенции не является обязательным атрибутом. Если он не указан явно, по умолчанию полагается базовый тип.

Типы компетенций:

- **Базовые** – это обычные компетенции без каких-либо особенностей.
- **Значимые** – отличаются тем, что алгоритм обеспечивает повышенную вероятность их успешного освоения на момент завершения курса, т.е. при прочих равных модулям, обеспечивающим значимые компетенции, будет отдаваться предпочтение при формировании траектории.
- **Повышенной сложности** – являются необязательными для изучения. Обучаемый может успешно завершить курс, даже если они не освоены. Если обучаемый пропускает модули, обеспечивающие компетенции повышенной сложности, то они не попадают в долг.
- **Вводные** – компетенции, которые соответствуют изученному ранее материалу (в школе, в предшествующих учебных курсах и т.п.). Обеспечивающие их модули по умолчанию обучающемуся не предоставляются. Предполагается, что изложенный в них материал обучающийся уже изучил ранее (условно уровень освоения таких компетенций полагается равным 100%, а при необходимости его можно уточнить диагностикой). Рекомендуемый сценарий использования вводных компетенций – указание их в качестве *пререквизитов* каких-либо модулей (понятие пререквизита вводится в следующем подразделе «Маркеры и пререквизиты»).
- **Диагностические** – позволяют обеспечить входное тестирование обучающихся. В результате можно выяснить, какие разделы учебного материала (какие компетенции) обучающийся уже знает или освоил ранее, и не вставлять их в траекторию, сконцентрировавшись на проблемных разделах. Подробнее о диагностике говорится в подразделе «Диагностика обучаемых».

- **Оцениваемые вручную** – это компетенции, оценить уровень освоения которых может только преподаватель.

Тип модуля определяется типом его выходных компетенций:

- Если у модуля только одна выходная компетенция, то тип модуля совпадает с типом этой компетенции (соответственно, получаем базовый, значимый, сложный, вводный или диагностический модуль).
- Если все выходные компетенции модуля имеют один и тот же тип, то и сам модуль имеет такой же тип.
- Если типы выходных компетенций модуля отличаются, то типом модуля будет тип самой «строгой» компетенции. Порядок «строгости» (по возрастанию): диагностическая → вводная → базовая → значимая → повышенной сложности → оцениваемая вручную.

Например, если модуль имеет две выходные компетенции, первая из которых является базовой, а вторая – значимой, то модуль будет иметь значимый тип. Поведение модуля определяется его типом аналогично компетенциям.

4 Маркеры и пререквизиты



Материал данного подраздела является дополнительным. Использование маркеров и пререквизитов не является обязательным, но позволяет управлять формированием траектории обучения.

Маркеры и пререквизиты не являются обязательными элементами курса, но иногда могут помочь более точно построить дальнейшую траекторию обучения. Причем они используются только тогда, когда обучаемый завершил тест на оценку уровня освоения какой-либо компетенции с отрицательным результатом (т.е. показав уровень ниже требуемого).

Пререквизит модуля – это компетенция, которая, как предполагается, освоена обучаемым ранее (либо до изучения текущего учебного курса, либо в его рамках – далее рассмотрим оба случая) и необходима для понимания материала модуля. Если выходной тест модуля обучаемым не пройден, все пререквизиты данного модуля будут вставлены следующими в траекторию обучения.

Если пререквизит указывает на компетенцию, изученную в рамках текущего курса, то обычно она имеет базовый (или значимый) тип.

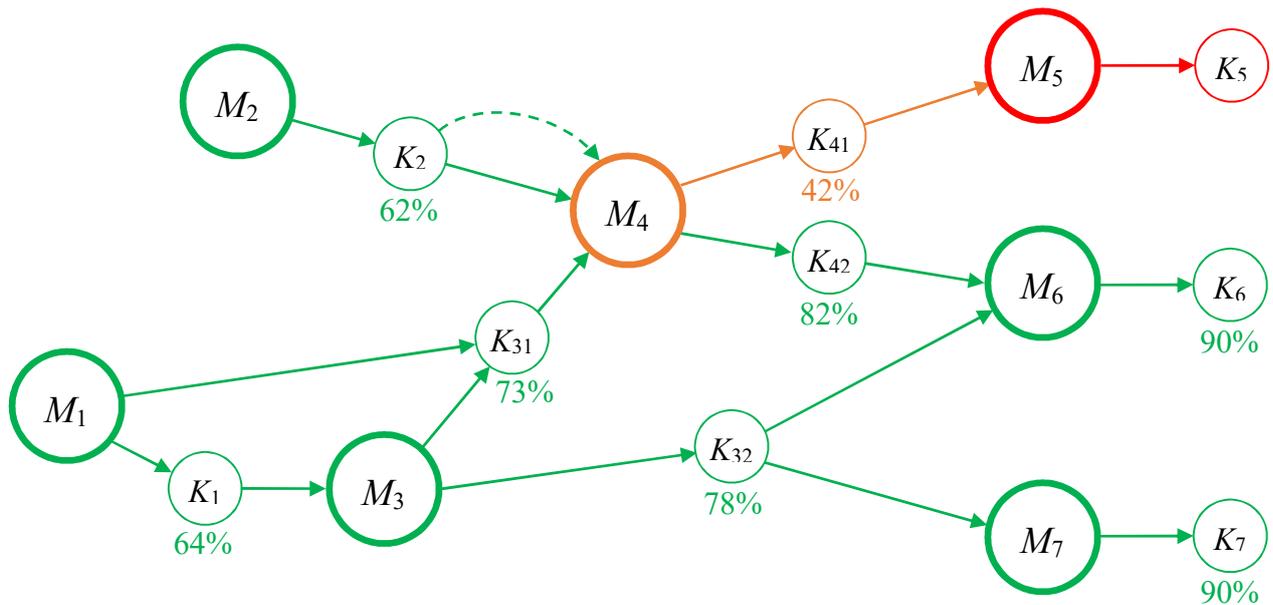


Рисунок Б.1.10 – Добавление базовой компетенции-пререквизита

Вернемся к примеру (рис. Б.1.10), где обучаемый никак не может освоить компетенцию K_{41} . Почему так происходит? Алгоритм не знает. Возможно, причина в том, что обучаемый уже забыл материал, изученный при освоении предыдущих разделов курса. Например, компетенцию K_2 . Когда (и если) предполагаемый уровень ее освоения, согласно кривой забывания, снизится ниже допустимого, обучаемый будет снова по ней протестирован. И только после этого, если тестирование будет «провалено», в траектории снова появится модуль, обеспечивающий компетенцию K_2 (в данном случае он один – M_2). Но пока этого не произошло, пререквизиты выступают в роли механизма, позволяющего гарантированно перейти к повторному изучению компетенции (т.е. вставить в траекторию модуль, который ее обеспечивает). Если сделать компетенцию K_2 пререквизитом модуля M_4 , то каждый раз, когда обучаемый будет «проваливать» выходной тест модуля M_4 , ему в траекторию будет добавляться модуль M_2 .

Данной возможностью нужно пользоваться обдуманно:

- во-первых, если у модуля большое количество пререквизитов, то после каждого неудачного прохождения выходного теста все модули, обеспечивающие соответствующие компетенции, будут добавлены в траекторию, что сделает её существенно длиннее;
- во-вторых, она, в общем случае, не спасает от заикливания, хотя в цикл будет входить уже не один модуль, а несколько (для примера выше – M_4 и M_2).

Если пререквизит указывает на компетенцию, изученную обучаемым когда-то ранее, до изучения текущего учебного курса, то обычно она имеет вводный тип. Например (рис. Б.1.11), пусть у модуля M_4 имеется пререквизит P_4 .

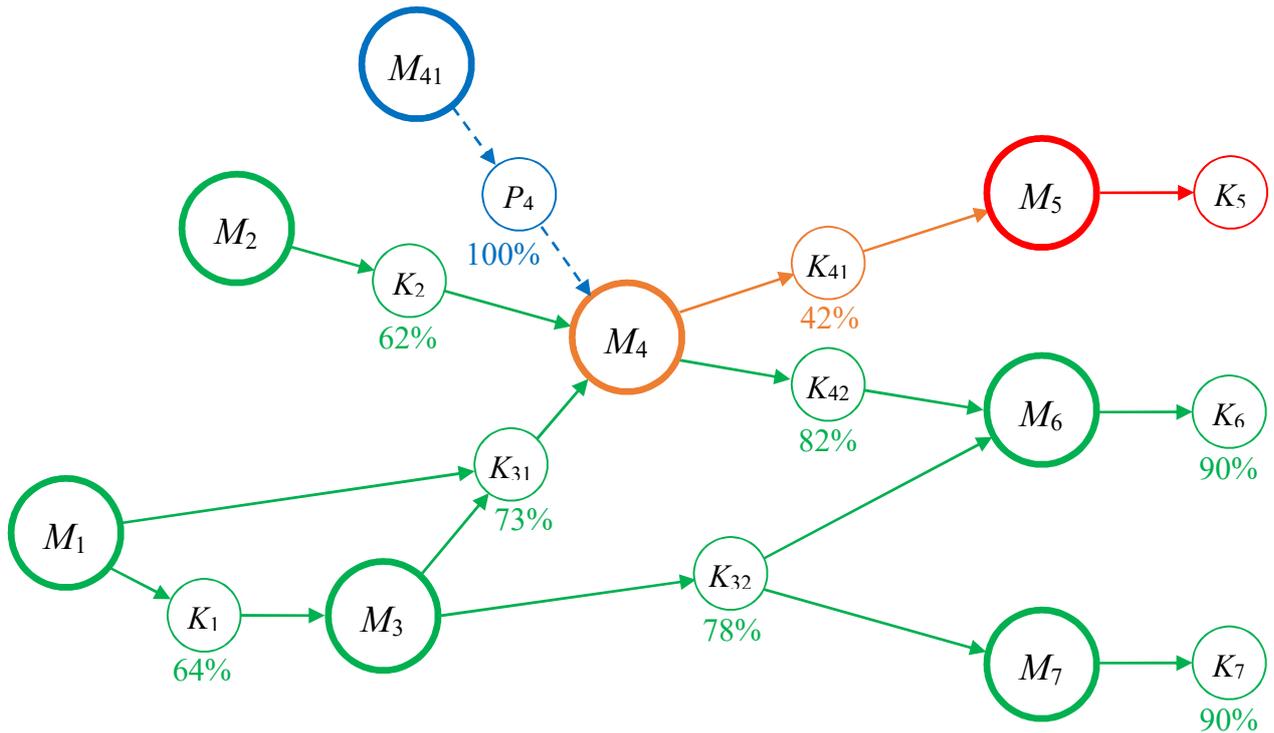


Рисунок Б.1.11 – Добавление вводной компетенции-пререквизита

Он соответствует знаниям, которые обучаемый получил в рамках школьного курса, и которые необходимы для освоения модуля M_4 (а точнее, его выходных компетенций K_{41} и K_{42}). Сама компетенция P_4 и обеспечивающий ее модуль M_{41} , соответственно, имеют вводный тип.

Как уже было сказано выше, изначально адаптивный алгоритм полагает, что данная компетенция обучаемым уже освоена на 100%, и модуль M_{41} в траектории не появляется. Также на эту оценку 100% не действует механизм забывания. Но, как только обучаемый не смог справиться с выходным тестом модуля M_4 , модуль M_{41} попадет в траекторию, и затем, по итогам тестирования, уровень освоения компетенции P_4 будет актуализирован.

Маркер – это атрибут тестового вопроса, содержащий идентификатор какой-либо компетенции. Если при выполнении теста обучаемым был дан неверный ответ на этот вопрос, и в итоге тест не был пройден, то в траекторию обучения будет добавлен модуль, обеспечивающий соответствующую компетенцию. Если же тест был пройден успешно, маркеры игнорируются (за исключением проведения диагностики, о чем сказано в следующем подразделе «Диагностика обучаемых»).

Как можно заметить, маркеры – более тонкий механизм, чем пререквизиты. То есть они «бьют точно», а не «по площади». Но, с другой стороны, их расстановка является более трудоемким занятием.

Итак, пусть в тест, оценивающий уровень освоения компетенции K_{41} , был добавлен вопрос, для ответа на который нужно также владеть компетенцией K_2 . Тогда его можно пометить маркером $\#K_2$. Если на этот вопрос будет дан неверный ответ и тест в целом будет «провален», следующим в траекторию будет добавлен модуль,

обеспечивающий компетенцию K_2 (в данном случае есть только один вариант – это будет модуль M_2). Аналогично в тест, оценивающий уровень освоения компетенции K_{42} был добавлен вопрос, для ответа на который нужно также владеть компетенцией K_{31} , и тогда его можно пометить маркером $\#K_{31}$. При неудачном завершении теста и (одновременно) неверном ответе на данный вопрос следующим в траекторию будет добавлен модуль, обеспечивающий компетенцию K_{31} (т.е. либо M_1 , либо M_3).

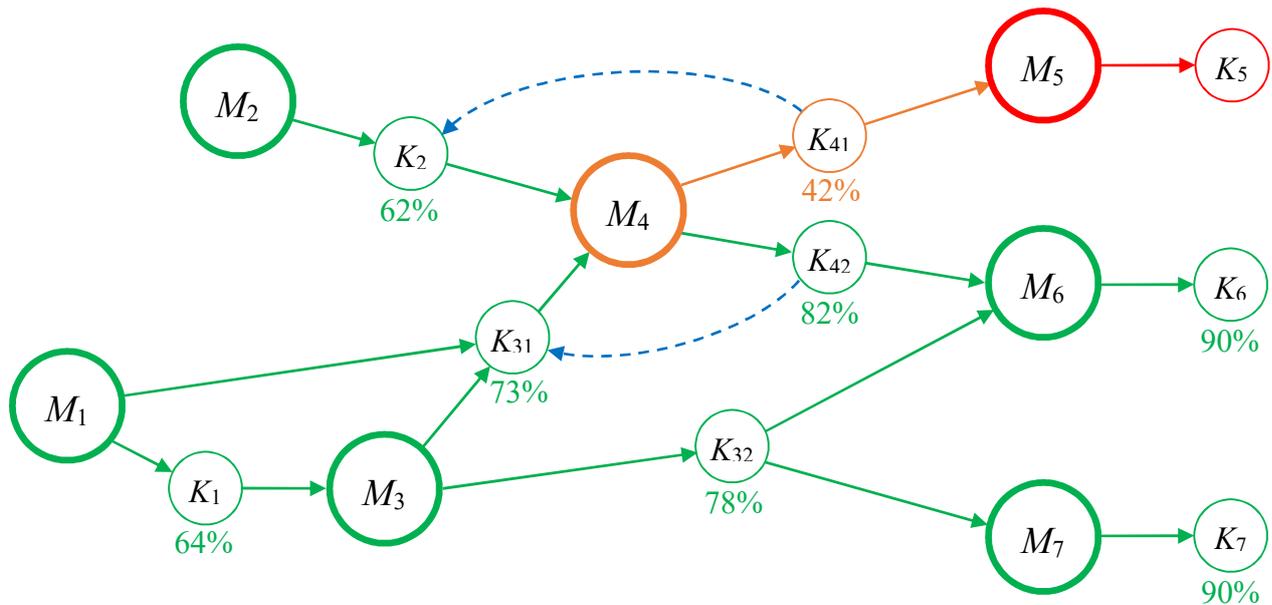


Рисунок Б.1.12 – Добавление маркеров

Если в тесте есть несколько вопросов, помеченных одним и тем же маркером, по ним вычисляется средняя оценка. Например, пусть в тесте для компетенции K_{41} есть четыре вопроса, помеченных маркером $\#K_2$. Тогда, при минимально необходимом уровне 60%, если обучаемый ответит правильно на 3 или 4 из них (показав уровень освоения 75 или 100% соответственно), модуль M_2 в траекторию включён не будет, а если правильных ответов менее 3 – будет включён (опять же, при дополнительном условии, что в целом тест не пройден).

Таким образом, рассмотренное ранее общее представление модуля расширяется такими необязательными элементами, как маркеры и пререквизиты. Пререквизиты, как и входные компетенции, указываются на входе модуля. А маркеры указываются на входе и выходе компетенций, т.к. они связывают компетенции друг с другом. Например, на рисунке Б.1.12 маркер $\#K_2$ является выходным для компетенции K_{41} и входным для K_2 , а маркер $\#K_{31}$ является выходным для компетенции K_{42} и входным для K_{31} .

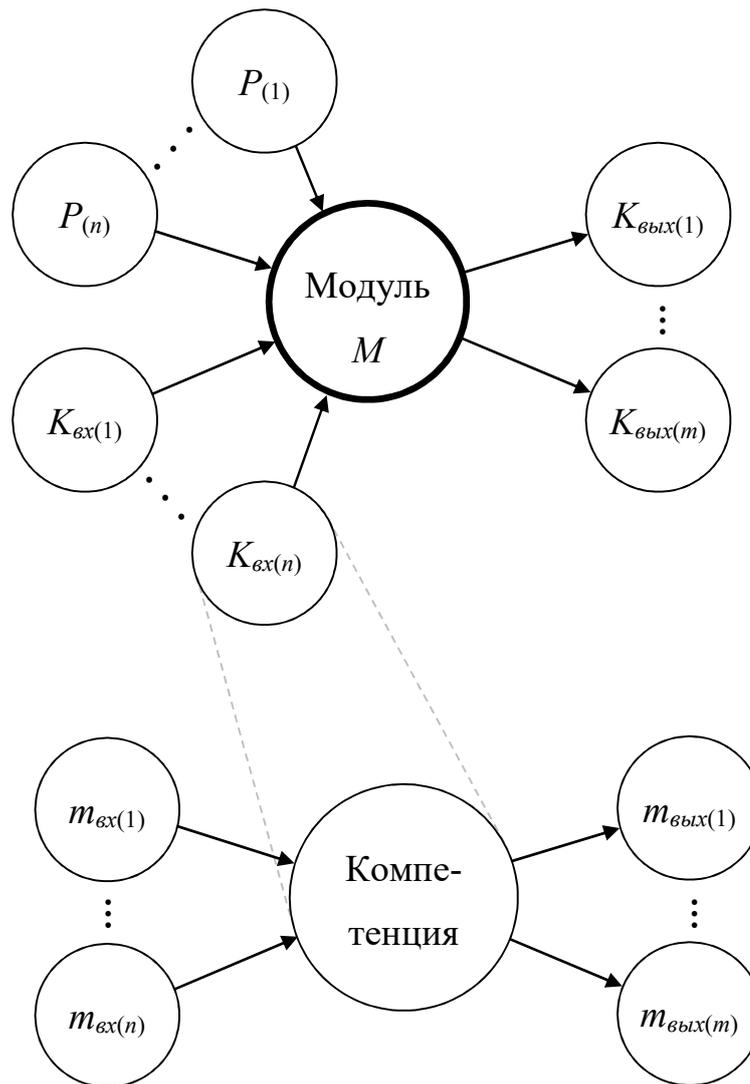


Рисунок Б.1.13 – Расширенное представление модуля

Как уже отмечено выше, адаптивный алгоритм может работать без пререквизитов и маркеров, используя другой механизм повторения ранее изученного материала (кривую забывания). Использовать эти возможности нужно обдуманно, на тех этапах изучения курса, где это оправданно. Во-первых, по той причине, что задача **качественной** декомпозиции (рис. Б.1.13) учебных материалов дисциплины на модули и компетенции для адаптивного курса сама по себе является непростой, а добавление в граф дополнительных связей, соответствующих пререквизитам и маркерам, делает её ещё сложнее. Во-вторых, если задачу построения траектории полностью (или почти полностью) возложить на пререквизиты и маркеры, тогда не будут в достаточной степени задействованы возможности адаптивного алгоритма по формированию оптимальной траектории обучения, а также учету индивидуальных характеристик скорости усвоения и забывания информации обучаемым. Поэтому нужно соблюдать баланс между тем, чтобы жёстко расставлять переходы в графе и тем, чтобы этим занимался адаптивный алгоритм.

5 Диагностика обучаемых



Материал данного подраздела является дополнительным. Наличие диагностических модулей и компетенций в учебном курсе не является обязательным.

Диагностика в учебном курсе требуется тогда, когда обучающиеся на входе могут иметь различный уровень подготовки. Если какие-то разделы курса обучаемым уже изучены ранее и уровень остаточных знаний по ним достаточно высок, то нет смысла добавлять их в траекторию обучения.

Определены следующие особенности диагностических модулей и компетенций:

- Во-первых, диагностические компетенции могут являться выходными только для диагностических модулей, а компетенции других типов диагностические модули обеспечивать не должны.
- Во-вторых, диагностические модули не содержат учебный материал, иными словами, они ничему не учат. Поэтому диагностическая компетенция имеет «служебное» назначение: она определяет, во-первых, положение диагностического модуля в траектории обучения, во-вторых, перечень модулей, по материалам которых проводится диагностика. Но уровень её освоения нигде не отображается и на результат обучения не влияет.
- В-третьих, диагностический модуль появляется в траектории обучения только один раз. С каким бы результатом ни была пройдена диагностика, повторно она не проводится.
- В-четвертых, все вопросы в диагностическом тесте (это тест, который привязан к диагностической компетенции) должны иметь маркеры. Именно маркеры в итоге определяют, какие разделы курса обучаемым уже освоены, а какие – ещё нет.

Например, на рисунке Б.1.14 изображен диагностический модуль M_d , проверяющий уровень освоения компетенций K_1 , K_2 и K_{31} . Как уже было сказано выше, использование служебной диагностической компетенции K_d , во-первых, гарантирует, что модуль M_d будет в траектории первым, а модули M_1 и M_2 – только после него. Во-вторых, т.к. она является входной для модулей M_3 и M_4 , они, потенциально, могут быть исключены из дальнейшей траектории, а их выходным компетенциям будут присвоены оценки, исходя из результатов диагностики.

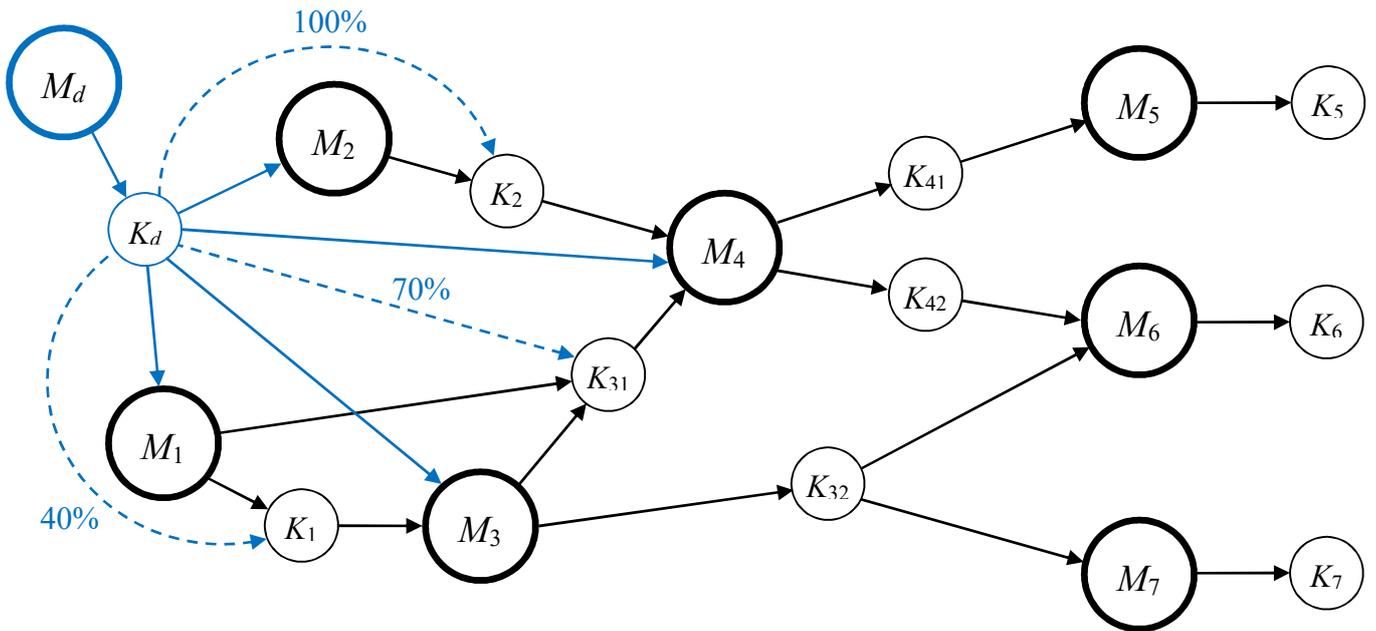


Рисунок Б.1.14 – Пример графа учебного курса с диагностикой

Например, пусть при тестировании компетенции K_1 обучаемым был показан уровень 50%, компетенции K_2 – 100%, компетенции K_{31} – 70%. Тогда на следующем шаге (рис. Б.1.15) обучаемому станут доступны модули M_1 , M_3 , M_5 , M_6 и M_7 , а модули M_2 и M_4 будут из дальнейшей траектории исключены.

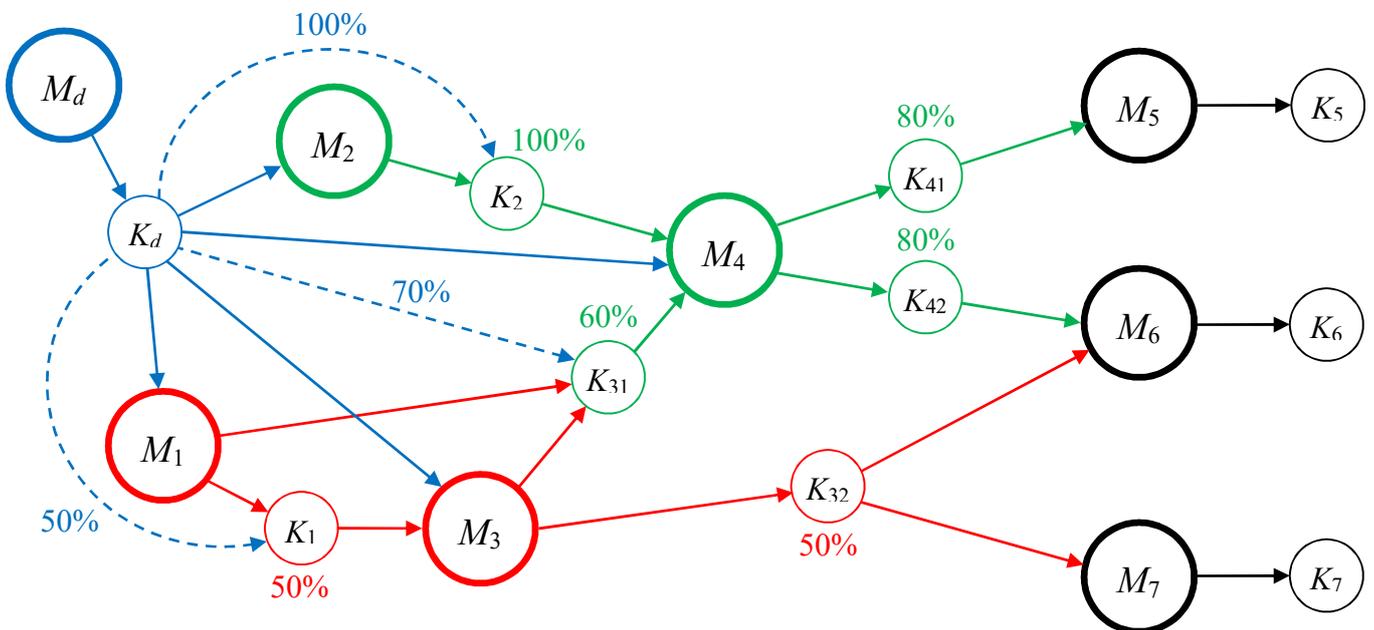


Рисунок Б.1.15 – Оценка успеваемости студента после диагностики

Модуль M_2 из траектории будет исключен, т.к. диагностика показала, что его выходная компетенция K_2 освоена на достаточном уровне (100%). Модуль M_4 будет исключен потому, что средняя оценка его входных компетенций составит 80% (эти же оценки будут также присвоены выходным компетенциям данного модуля, т.к. он является диагностируемым, т.е. имеет входную диагностическую

компетенцию). При этом оценка компетенции K_2 на уровне 100% была получена при диагностике напрямую, а оценка за компетенцию K_{31} на уровне 60% – как среднее взвешенное арифметическое от полученных при диагностике напрямую 70% и 50%, полученные при диагностике компетенции K_1 . Таким образом, компетенция K_{31} является выходной для диагностируемых модулей M_1 и M_3 , поэтому их средняя оценка опосредованно распространяется и на неё саму. При вычислении среднего взвешенного арифметического вес каждой оценки обратно пропорционален количеству маркеров, которые участвовали в её формировании. В данном случае положим, что при оценке всех компетенций участвовало по одному маркеру, поэтому получили обычное среднее арифметическое. Таким образом, в результате диагностики оценки получают все выходные компетенции диагностируемых модулей.

Диагностический тест нельзя пройти успешно или неуспешно. Какой бы результат ни был показан, диагностическая компетенция считается освоенной, а оценки за вопросы-маркеры позволяют определить, какие разделы курса будут включены в траекторию, а какие – нет.

II Формат описания модулей

Минимально необходимый набор материалов для разработки адаптивного курса – это документы с описанием модулей (рассмотрены в данном разделе), а также документы с описанием компетенций (рассмотрены в разделе III).

Описание модулей включает их карту, а также отдельные документы (в формате HTML, Microsoft Word, LibreOffice Writer и т.п.), соответствующие каждому модулю курса. Документ (рис. Б.2.1) с описанием модуля должен включать его метаданные, а также содержание (контент).

1 Метаданные модуля

Метаданные модуля задаются таблицей следующего вида:

Номер модуля	Номер модуля ⁽¹⁾
Название модуля	Название модуля
Входные компетенции⁽²⁾	Номер компетенции 1 ⁽¹⁾ Номер компетенции 2 ⁽¹⁾ ... Номер компетенции N ⁽¹⁾
Выходные компетенции⁽³⁾	Номер компетенции 1 ⁽¹⁾ (уровень L_1) Номер компетенции 2 ⁽¹⁾ (уровень L_2) ... Номер компетенции M ⁽¹⁾ (уровень L_M)

Пререквизиты ⁽⁴⁾	Номер компетенции 1 ⁽¹⁾ Номер компетенции 2 ⁽¹⁾ ... Номер компетенции P ⁽¹⁾
Время на изучение модуля	T мин
Комментарий	(если необходим)

Примечания:

⁽¹⁾ Модули и компетенции должны образовывать дерево (или лес). Желательно, чтобы номера модулей и компетенций указывали на их место в общей иерархии. Например:

- компетенция 1.2.1 – первое дерево компетенций, вторая ветвь от корня, первая подветвь;
- модуль 1.2.1 – первый раздел учебного материала, второй подраздел, первый под-подраздел.

Номер модуля должен быть явно указан в имени файла, описывающего модуль (например, «1.2.1.docx», «1.2.1.odt», «Раздел 1.2.1.docx», «1.2.1. Перевод чисел из десятичной системы счисления в двоичную.docx» и т.п.).

Система нумерации модулей и компетенций на алгоритм не влияет, она скорее нужна программистам, методистам и авторам для того, чтобы удобнее ориентироваться в контенте.

⁽²⁾ Входные компетенции – это компетенции, которые необходимо освоить для доступа к модулю. Если хотя бы одна входная компетенция не освоена на достаточном уровне (этот уровень задаётся в системных настройках и по умолчанию составляет 60%), доступ к модулю запрещён. Модуль может не иметь входных компетенций, это значит, что ему разрешено появляться в самом начале траектории обучения. В то же время алгоритм может предоставить такой модуль и в середине траектории (но до модулей, входные компетенции которых являются выходными у рассматриваемого модуля).

⁽³⁾ Выходные компетенции – это компетенции, которые формируются при изучении модуля. У каждого модуля должна быть как минимум одна выходная компетенция. У каждой выходной компетенции можно указать уровень L её раскрытия в материале модуля. Например, один модуль может раскрывать компетенцию на 75%, другой модуль – эту же компетенцию на 90% и т.д. Если уровень не указан, по умолчанию используется значение 100%.

⁽⁴⁾ Пререквизиты – это компетенции, которые будут предложены обучаемому для освоения в том случае, если он не пройдет выходной тест модуля, т.е. это указание на то, какие компетенции обучаемому нужно изучить повторно, прежде чем

возвращаться к данному модулю. Не являются обязательным атрибутом модуля. Если пререквизиты не указаны, система будет использовать другие механизмы для повторения забытого материала (расчёт кривой забывания, маркеры в вопросах).

Дополнительные сведения о модулях, компетенциях, пререквизитах и др. можно получить в разделе «Общие понятия».

При наличии карты модулей (см. подраздел «Карта модулей») в метаданные допускается включать только следующие поля:

Номер модуля	Номер модуля
Время на изучение модуля	T мин
Комментарий	(если необходим)

Остальные атрибуты указываются в карте.

2 Содержание модуля

Включает любой учебный контент: теоретический материал (текст, формулы, рисунки, видео и т.п.), примеры решения практических задач и пр. Содержание модуля может быть пустым (например, если это диагностический модуль либо модуль, содержащий только тестовые вопросы или задачи, и т.п.).

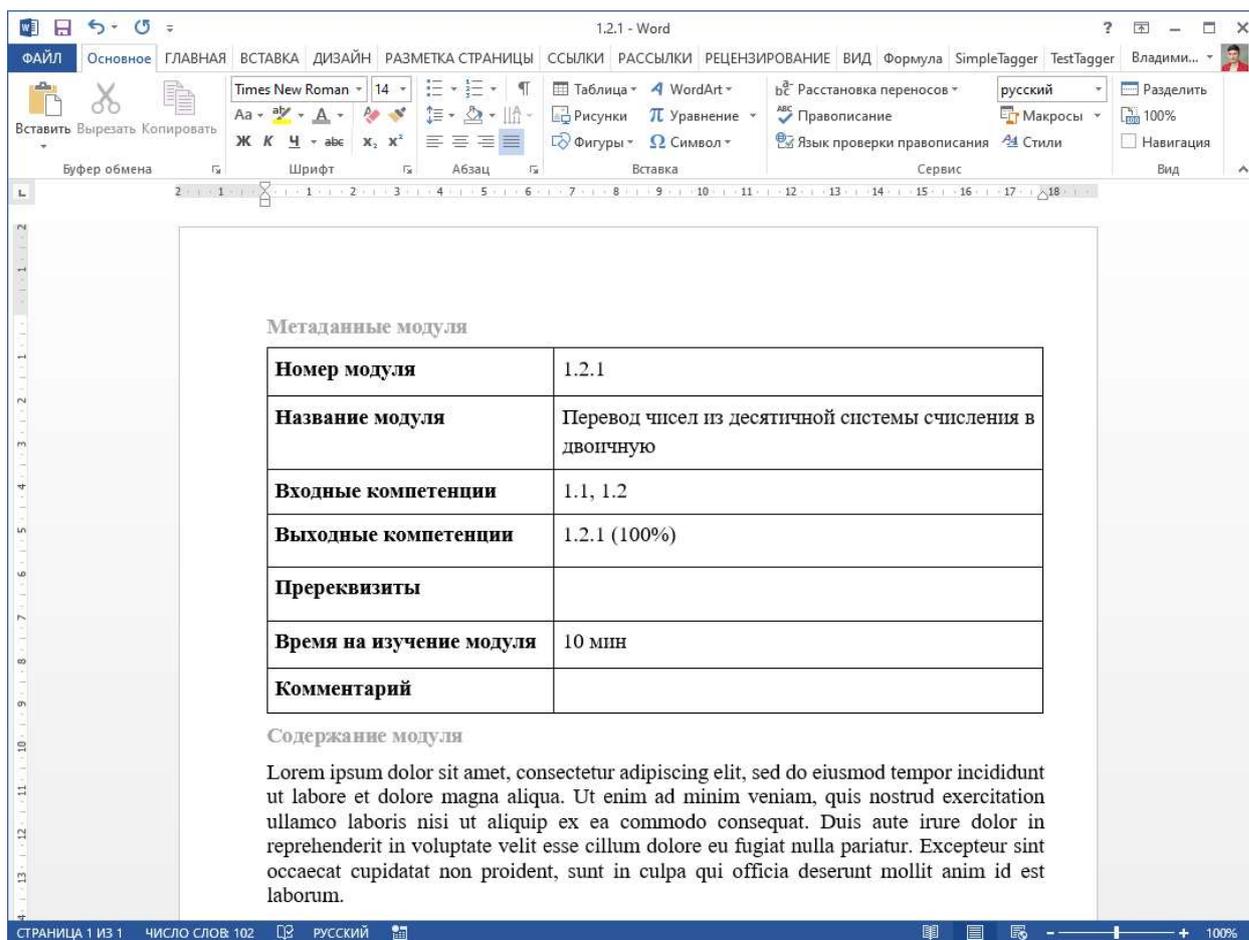


Рисунок Б.2.1 – Пример оформления модуля

3 Карта модулей

Чтобы видеть картину в целом, необходимо иметь *карту модулей* – сводный перечень модулей и их связей с компетенциями в виде таблицы (Microsoft Excel, LibreOffice Calc и т.п.) следующего вида:

№	Номер модуля	Название модуля	Входные компетенции	Выходные компетенции	Пререквизиты	Комментарий

На рисунке Б.2.2 приведен фрагмент карты модулей.

№	Номер модуля	Название модуля	Входные компетенции	Выходные компетенции	Прerequisites	Комментарий
1	1	Проверка знаний		1		
2	1.1	Десятичная система счисления	1	1.1		
3	1.2	Двоичная система счисления	1	1.2		
4	1.2.1	Перевод чисел из десятичной системы счисления в двоичную	1.1 1.2	1.2.1		
5	1.2.2	Перевод чисел из двоичной системы счисления в десятичную	1.1 1.2	1.2.2		
6	1.3	Шестнадцатеричная система счисления	1	1.3		
7	1.3.1	Перевод чисел из десятичной системы счисления в шестнадцатеричную	1.1 1.3	1.3.1	1.2.1	
8	1.3.2	Перевод чисел из шестнадцатеричной системы счисления в десятичную	1.1 1.3	1.3.2	1.2.2	
9	1.4	Двоично-десятичная система счисления	1 1.1 1.2 1.3	1.4		
10	1.4.1	Перевод чисел из десятичной системы счисления в двоично-десятичную	1.2.1 1.3.1 1.4	1.4.1		
11	1.4.2	Перевод чисел из двоично-десятичной системы счисления в десятичную	1.2.2 1.3.2 1.4	1.4.2		
12	1.5	Системы счисления с произвольным основанием	1.2.1 1.2.2 1.3.1 1.3.2 1.4.1 1.4.2	1.5 1.5.1 1.5.2	1.1 1.2 1.3	

Рисунок Б.2.2 – Пример оформления карты модулей

III Формат описания компетенций

Описание компетенций включает их карту (см. подраздел «Карта компетенций»), а также отдельные документы (в формате HTML, Microsoft Word, LibreOffice Writer и т.п.), соответствующие каждой компетенции курса. Документ с описанием компетенции должен включать ее метаданные, а также тестовые вопросы для оценки уровня освоения компетенции.

1 Метаданные компетенции

Метаданные компетенции задаются таблицей следующего вида:

Номер компетенции	Номер компетенции ⁽¹⁾
Название компетенции	Название компетенции
Тип компетенции	D/I/B/Z/E/M ⁽²⁾
Комментарий	(если необходим)

Примечания:

(1) Рекомендации по способу нумерации модулей и компетенций приведены в подразделе «Метаданные модуля».

Номер компетенции должен быть явно указан в имени файла, описывающего компетенцию (например, «1.2.1.docx», «1.2.1.odt», «Компетенция 1.2.1.docx», «1.2.1. Уметь переводить числа из СС10 в СС2.docx» и т.п.).

(2) Тип компетенции можно указывать словами (см. подраздел «Типы модулей и компетенций») либо использовать условные обозначения: D – диагностическая, I – вводная, B – базовая, Z – значимая, E – повышенной сложности, M – оцениваемая вручную. Если тип не указан, по умолчанию полагается тип B.

При наличии карты компетенций (см. подраздел «Карта компетенций») указание метаданных не требуется. В этом случае для отдельных компетенций достаточно предоставить только тестовые вопросы.

2 Тестовые вопросы

Из описанных в данном разделе вопросов формируется тест, который будет предложен обучаемому для оценки уровня освоения компетенции (выходной тест).

Описание каждого вопроса должно иметь следующий формат:

[Название категории ⁽¹⁾] [Мощность категории ⁽²⁾] [Вопрос №... ⁽³⁾] [Название вопроса ⁽⁴⁾] [(Список маркеров) ⁽⁵⁾] Формулировка вопроса ⁽⁶⁾
--

Здесь квадратными скобками помечены необязательные атрибуты.

Примечания:

(1) Все вопросы теста могут быть разделены на несколько категорий. Если категория вопроса не указана, то он будет отнесён к той же категории, что и предыдущий вопрос. Если в документе категории вообще не указаны, по умолчанию создается одна категория для всех вопросов, название которой состоит из номера и названия компетенции.

(2) Мощности категории – это количество вопросов, включаемых в выходной тест из данной категории. Если категории не описаны, то мощность указывается для всего теста. Если мощность не указана, в тест будут включены все описанные вопросы.

Пример:

Тестовые вопросы Выдать 5 случайных вопросов из 10: Вопрос № 1 ...
--

Вопрос № 2 Вопрос № 10

В данном примере категории не указаны, а в тест будут включены 5 вопросов из 10 описанных.

(3) Нумерация вопросов задается для удобства, поэтому является необязательным атрибутом и особых требований к ней нет. Вопросы можно нумеровать по порядку, или использовать шаблон «Номер категории.Номер вопроса».

(4) Название вопроса также является необязательным атрибутом, но если оно указано, то будет показано обучаемому при тестировании.

(5) Здесь можно (но не обязательно) указать один или несколько маркеров. Описание каждого маркера имеет следующий формат:

#номер_компетенции

Как уже было сказано в подразделе «Маркеры и пререквизиты», если обучаемый выполнит выходной тест неудачно (то есть на оценку ниже минимального допустимого уровня) и при этом неправильно ответит на вопрос, помеченный маркером, то в дальнейшем ему в траекторию будет добавлен модуль, обеспечивающий указанную компетенцию.

Пример:

Тестовые вопросы	
1) Категория «Вопросы для оценки компетенции 1.2.2» (выдать 3 вопроса из 10)	Вопрос № 1.1
...	Вопрос № 1.2
...	...
...	Вопрос № 1.10
2) Категория «Вопросы для оценки компетенции 1.1» (выдать 1 вопрос из 3)	Вопрос № 2.1 (#1.1)
...	Вопрос № 2.2 (#1.1)
...	Вопрос № 2.3 (#1.1)
...	
3) Категория «Вопросы для оценки компетенции 1.2» (выдать 1 вопрос из 3)	

...	Вопрос № 3.1 (#1.2)
...	Вопрос № 3.2 (#1.2)
...	Вопрос № 3.3 (#1.2)
...	

В данном случае обучаемому в тесте будут предложены пять вопросов: три из первой категории и по одному вопросу из второй и третьей категорий. Чтобы успешно завершить тест, обучаемый (при настройках по умолчанию) должен правильно ответить на 3 вопроса из 5 (это будет соответствовать минимальной допустимой оценке 60%). В этом случае механизм маркеров задействован не будет. Если же он ответит неверно на 3 и более вопроса и среди них будет вопрос, помеченный маркером #1.1 или #1.2, то следующим в траекторию будут вставлены модули, обеспечивающие соответствующие компетенции.

Отдельно рассмотрим случай, когда в тест входят несколько вопросов с одним и тем же маркером, например, три вопроса с маркером #1.2. В этом случае, чтобы механизм маркеров был задействован, обучаемый должен ответить правильно только на один такой вопрос, получив за маркированную компетенцию оценку 33,3...%, либо ни на один, получив оценку 0%. В остальных случаях оценка будет больше минимально необходимого по умолчанию уровня 60%.

Сказанное выше относится к компетенциям типа I/V/Z/E. Первым исключением являются диагностические компетенции (типа D). Во-первых, в них наличие маркеров является обязательным. Во-вторых, при выполнении диагностического теста механизм маркеров будет задействован в любом случае, независимо от результатов тестирования. В-третьих, алгоритм работы механизма маркеров для компетенций типа D отличается:

- оценка за маркированные вопросы будет использована для установки начального уровня освоения маркированной компетенции;
- если для какой-либо компетенции обучаемым будет показан уровень её освоения ниже требуемого, то обеспечивающий данную компетенцию модуль не будет вставлен в траекторию немедленно – когда его предоставлять, определит адаптивный алгоритм.

Вторым исключением являются компетенции, оцениваемые вручну (типа M). Для них тесты составлять вообще не нужно, т.к. оценить эти компетенции может только преподаватель. Такие компетенции полезно использовать, когда обучение проходит в смешанном очно-дистанционном режиме. Например, обучаемые могут изучать дистанционно теоретический и практический материал дисциплины, а потом очно выполнить лабораторную работу (если для этого требуется специализированное лабораторное оборудование или по другим причинам). Соответствующая лабораторной работе компетенция типа M будет оценена преподавателем и внесена в адаптивный курс через специальный интерфейс.

Допуск же к лабораторной работе будет получен в адаптивном курсе посредством освоения компетенций других типов. В процессе изучения дисциплины может наступить момент, когда дальнейшее продвижение по траектории обучения невозможно без получения оценок за компетенции типа М. Система оповестит обучаемого об этом, показав сообщение вида «Для продолжения обучения необходимо получить у преподавателя оценку за компетенцию...», и далее будет указано название компетенции.

Необходимо отметить следующее. Если тесты для оценки двух и более компетенций содержат одинаковые вопросы, повторно их описывать не следует. Вместо этого нужно указать ссылку – какие вопросы (или категории вопросов) следует использовать в данном тесте. Это относится ко всем типам компетенций.

Пример:

Тестовые вопросы

- 1) Категория «Десятичная система счисления»
Выдать 2 случайных вопроса для компетенции 1.1 (#1.1)
- 2) Категория «Двоичная система счисления»
Выдать 2 случайных вопроса для компетенции 1.2 (#1.2)
- 3) Категория «Шестнадцатеричная система счисления»
Выдать 2 случайных вопроса для компетенции 1.3 (#1.3)
- 4) Категория «Двоично-десятичная система счисления»
Выдать 2 случайных вопроса для компетенции 1.4 (#1.4)

В данный диагностический тест включены 8 вопросов, по 2 из каждой категории. Сами вопросы берутся из тестов для проверки уровня освоения компетенций 1.1, 1.2, 1.3 и 1.4. Вопросы будут помечены указанными маркерами. Соответственно, в зависимости от количества правильных ответов обучаемого, каждой из перечисленных компетенций будет присвоен уровень 0%, 50% или 100% (как уже было сказано, для компетенций других типов маркеры так не работают, они обеспечивают только переходы, но не установку уровня освоения маркированных компетенций). В процессе дальнейшего изучения дисциплины в траекторию обучения не войдут модули, обеспечиваемые которыми компетенции получили оценку 100%, но войдут модули, обеспечивающие компетенции с оценками 0% и 50%, а также компетенции, которые не были продиагностированы.

Диагностический тест не обязательно должен содержать вопросы для других компетенций, здесь можно описать и новые, уникальные вопросы.

⁽⁶⁾ Вопрос формулируется в свободной форме. Допустимы любые типы вопросов с автоматическим оцениванием, поддерживаемые СДО Moodle:

1. Вопросы на одиночный и множественный выбор. Сначала задается тело вопроса, а затем, на отдельных маркированных или нумерованных строках – альтернативы ответов. Правильные альтернативы должны быть выделены (жирным начертанием, цветом шрифта или фона либо специальным знаком) или перечислены ниже вариантов ответов.

Пример вопроса на одиночный выбор (правильный ответ выделен):

<p>Какие цифры можно использовать для записи чисел в десятичной системе счисления?</p> <ul style="list-style-type: none"> – от 1 до 10 – от 0 до 10 – от 0 до 9

Пример вопроса на одиночный выбор (правильный ответ указан ниже):

<p>Какие цифры можно использовать для записи чисел в десятичной системе счисления?</p> <ul style="list-style-type: none"> – от 1 до 10 – от 0 до 10 – от 0 до 9 <p>Ответ: 3.</p>
--

Пример вопроса на множественный выбор:

<p>Какие цифры можно использовать для записи чисел в шестнадцатеричной системе счисления?</p> <ul style="list-style-type: none"> – от 1 до 16 – от 0 до 16 – от 0 до 9 – от 0 до 15 – от A до F – от 0 до F

2. Вопросы на ввод текстового или числового ответа. Кроме тела вопроса необходимо указать правильный ответ (или несколько правильных ответов). Для текстового вопроса можно дополнительно указать, чувствителен ли ответ к регистру. Для числового вопроса можно указать допустимую погрешность ответа.

Пример вопроса на ввод текстового ответа:

<p>Как называется единица информации, состоящая из 8 бит?</p> <p>Ответ: байт.</p>
--

Пример вопроса на ввод числового ответа:

<p>Сколько бит содержит один байт?</p> <p>Ответ: 8.</p>
--

3. Вопросы на соответствие и упорядочение. Кроме тела вопроса состоят из пунктов, которые необходимо сопоставить или упорядочить. В вопросе на соответствие могут быть «лишние» варианты ответов.

Пример вопроса на соответствие:

Сопоставьте единицы информации с их емкостью в байтах:

Килобайт	1024
Мегабайт	1048576
	1000
	1000000
	1024000

Пример вопроса на упорядочение:

Расположите единицы информации в порядке возрастания их емкости:

Терабайт
Петабайт
Эксабайт
Зеттабайт

4. Составной вопрос. Может содержать несколько вложенных вопросов на одиночный выбор, а также ввод текстового или числового ответа.

Наряду с килобайтом существует такая единица измерения объема информации, как (килобод/килобит), которая состоит из **(128)** байт.

Здесь в вопросе есть два вложенных вопроса – первый на выбор альтернативы, второй – на ввод числового ответа.

Остальные возможности по настройке параметров вопросов см. в документации СДО Moodle.

Если уже имеется готовая база вопросов, внесенная в СДО Moodle, ее можно предоставить в формате MoodleXML. Дополнительно в документе с описанием компетенции в этом случае указываются только маркеры, если они требуются.

Если для компетенции не указаны тестовые вопросы, то студент будет получать за неё оценку автоматически (оценка будет равна уровню *L*, который присвоен выходной компетенции в описании модуля, см. подраздел «Метаданные модуля»). Эту возможность можно использовать для компетенций, соответствующих ознакомительному материалу, т.е. если проверка уровня его усвоения не требуется, но при этом его нужно разместить в определенном месте графа (траектории) обучения.

На рисунке Б.3.1 показан пример описания компетенции. Здесь в первой категории «Вопросы для оценки компетенции 1.2.1» описаны 10 вопросов, из которых обучаемому в тесте будут предоставлены 3. Первый вопрос – на выбор правильной альтернативы, последний – на ввод числового (или текстового) ответа. Во второй категории «Вопросы для оценки компетенции 1.1» описаны 3 вопроса, из которых

обучаемому будет предоставлен один. При этом вопрос 2.1 помечен маркером #1.1. То есть, если тест не будет пройден и одновременно будет дан неверный ответ на этот вопрос, следующим в траектории может появиться какой-либо модуль, обеспечивающий компетенцию 1.1 (но не обязательно сразу, т.к. в тесте могут иметься другие маркеры, а модуль может иметь пререквизиты – все соответствующие им компетенции попадают в пул, из которого выбираются адаптивным алгоритмом в произвольном порядке).

1.2.1 - Word

Метаданные компетенции

Номер компетенции	1.2.1
Название компетенции	Уметь переводить числа из десятичной системы счисления в двоичную
Тип компетенции	В
Комментарий	

Тестовые вопросы

1) Категория «Вопросы для оценки компетенции 1.2.1» (выдать 3 вопроса из 10)

Вопрос № 1.1

Какие числа можно переводить из СС10 в СС2?

- только такие, которые состоят из 0 и 1;
- **любые (правильный ответ);**
- только числа в диапазоне от 0 до 2.

Вопрос № 1.2

...

Вопрос № 1.10

Переведите число 53 из СС10 в СС2.

Ответ: 110101.

2) Категория «Вопросы для оценки компетенции 1.1» (выдать 1 вопрос из 3)

Вопрос № 2.1 (#1.1)

Какие цифры можно использовать для записи чисел в СС10?

- от 1 до 10;
- от 0 до 10;
- **от 0 до 9 (правильный ответ).**

СТРАНИЦА 1 ИЗ 2 ЧИСЛО СЛОВ 176 РУССКИЙ 100%

Рисунок Б.3.1 – Пример оформления компетенции

3 Карта компетенций

Карта компетенций может быть представлена древовидным (иерархическим) списком любого формата (рис. Б.3.2). Рекомендуется использовать формат *ментальных карт* (MindMap), который поддерживается, например, свободно распространяемой программой FreeMind.

Для каждой компетенции в карте необходимо указывать ее метаданные (см. раздел «Метаданные компетенции»):

- номер компетенции;
- название компетенции;
- тип компетенции (необязательный атрибут).

Карта компетенций не является обязательной, однако, как показывает практика, ее разработка однозначно рекомендуется. Она позволяет автору более наглядно представлять декомпозицию материалов курса.

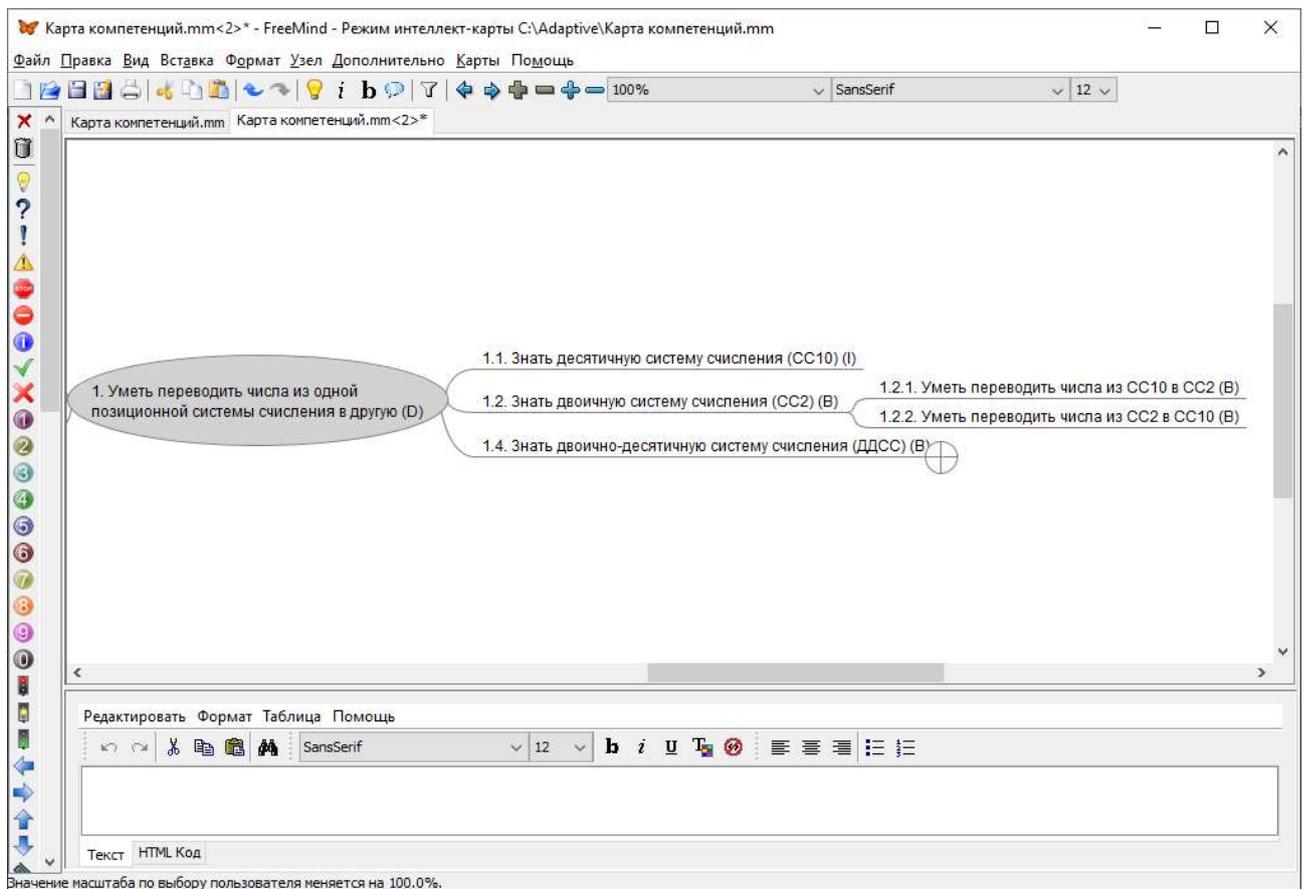


Рисунок Б.3.2 – Пример разработки карты компетенций

Разработка карты начинается с выделения одной или нескольких компетенций верхнего уровня, освоение которых является задачей учебного курса. Далее эти компетенции декомпозируются на субкомпетенции до тех пор, пока не будут определены конечные элементарные компетенции, находящихся в листьях полученного дерева (леса).

При разработке модулей, обеспечивающих полученные компетенции, возможны три подхода:

- модули разрабатываются только для листовых компетенций;

- модули разрабатываются для всех компетенций, по принципу от общего к частному (т.е. при изучении курса обучаемый сначала освоит субкомпетенции и только потом – их родительскую компетенцию);
- модули разрабатываются для всех компетенций, по принципу от частного к общему (т.е. при изучении курса обучаемый сначала освоит родительскую компетенцию, и только потом – её субкомпетенции).

Выбор одного из перечисленных подходов зависит от ряда факторов, таких как общая логика изложения материала курса, принципы декомпозиции компетенций и т.д.

ПРИЛОЖЕНИЕ В

Свидетельства о государственной регистрации программ для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2021612218

Система адаптивного обучения Nexbe

Правообладатель: *Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)*

Авторы: *Кречетов Иван Анатольевич (RU), Романенко Владимир Васильевич (RU), Мазитов Артём Захарович (RU), Городович Андрей Викторович (RU)*

Заявка № 2021611335

Дата поступления 10 февраля 2021 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 12 февраля 2021 г.



Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев

ПРИЛОЖЕНИЕ Г**Акты внедрения**

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ

«Национальный исследовательский технологический университет «МИСиС»

(НИТУ «МИСиС»)

АКТ

«05» марта 2020 г.

Москва

№ 2/239

**О внедрении в учебный процесс технологий адаптивного обучения,
разработанных в рамках диссертационного исследования
Кречетова Ивана Анатольевича**

Комиссия в составе доцента кафедры общей и неорганической химии НИТУ «МИСиС», кандидата химических наук, Дегтярева А.В., директора центра «Школа педагогического мастерства» НИТУ «МИСиС», кандидата технических наук, Дорофеевой М.Ю., и.о. проректора по образованию НИТУ «МИСиС», кандидата технических наук, доцента Волкова А.А., составила настоящий акт в том, что результаты диссертационного исследования Кречетова Ивана Анатольевича на соискание степени кандидата технических наук по специальности 05.13.10 – «Управление в социальных и экономических системах» используются в учебном процессе НИТУ «МИСиС».

Отмечаем, что:

1. Внедрены:

- инструмент создания адаптивного образовательного контента, включающий средства управления базой учебных модулей и базой субкомпетенций, в т.ч. средства описания учебных модулей в среде СДО, средства присвоения субкомпетенций и назначения входного / выходного тестирования;
- модель адаптивного обучения, состоящая из модели контента, модели обучаемого и алгоритма построения траектории обучения;
- методика создания адаптивных электронных курсов, включающая в себя этапы: декомпозиции компетенций, проектирования учебных модулей и тестовых заданий, создания контента в СДО через предложенный инструмент.

Результаты внедрения позволили:

– разработать адаптивный электронный курс «Химия» в объеме 3 з.е., соответствующий рабочей программе дисциплины «Химия» НИТУ «МИСиС». Адаптивный электронный курс включает 300 учебных модулей, 2000 тестовых вопросов, более 300 субкомпетенций;

– повысить успеваемость студентов группы, обучавшейся с применением технологий адаптивного обучения (20 чел.) в среднем на 25 % по сравнению с группой, обучавшейся по традиционной технологии (20 чел.).

Доцент кафедры ОиНХ
к.х.н., доцент

А.В. Дегтярев

Директор центра «Школа
педагогического мастерства»
к.т.н.

М.Ю. Дорофеева

И.о. проректора по образованию
к.т.н., доцент



А.А. Волков

УТВЕРЖДАЮ
Директор Института дистанционного
образования Томского государственного
университета



М.О. Шепель

М.П.

«05» февраля 2021 г.

№ 1

АКТ

о внедрении (использовании) результатов
кандидатской диссертационной работы
Кречетова Ивана Анатольевича

Комиссия в составе:

председатель Дубровская Виктория Сергеевна, заместитель директора
Института дистанционного образования ТГУ,

члены комиссии: Фещенко Артем Викторович, заведующий лабораторией
компьютерных средств обучения Института дистанционного образования
ТГУ, Лозинская Анастасия Петровна, заместитель директора Института
дистанционного образования ТГУ, Заседатель Вячеслав Сергеевич,
программист Института дистанционного образования ТГУ

составили настоящий акт о том, что результаты диссертационной работы
«Модели, алгоритмы и инструментальные средства адаптивного обучения»,
представленной на соискание ученой степени кандидата технических наук,
использованы при совершенствовании системы электронного обучения в
Томском государственном университете в виде:

1. Модели адаптивного обучения.
2. Методики разработки адаптивных электронных курсов.
3. Инструментальной системы разработки адаптивного учебного контента.

Использование указанных результатов позволяет:

- 1) создавать адаптивные электронные курсы;
- 2) реализовывать технологию адаптивного обучения;
- 3) сократить затраты на проведение работ, связанных с разработкой адаптивных электронных курсов.

Председатель комиссии:



В.С. Дубровская

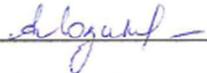
05.02.2021 г.

Члены комиссии:



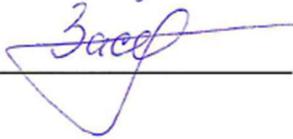
А.В. Фещенко

05.02.2021 г.



А.П. Лозинская

05.02.2021 г.



В.С. Заседатель

05.02.2021 г.

Министерство науки и высшего образования Российской Федерации
**Федеральное государственное бюджетное образовательное учреждение
 высшего образования**
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»



УТВЕРЖДАЮ

Проректор по научной работе и инновациям

_____ А.Г. Лошилов

« 30 » 03 2021 г.

АКТ

внедрения в Томском государственном университете систем управления и радиоэлектроники

Комиссия в составе:

председатель зам. директора ИИ ТУСУР Городович Андрей Викторович,

члены комиссии: и.о. декана ФДО ТУСУР Черкашина Ирина Петровна, к.т.н.,

доцент каф. ТЭО ТУСУР Перминова Мария Юрьевна,

составили настоящий акт о том, что результаты диссертационной работы Кречетова Ивана Анатольевича «Модели, алгоритмы и инструментальные средства адаптивного обучения», представленной на соискание ученой степени кандидата технических наук, внедрены в промышленную эксплуатацию на факультете дистанционного обучения Томского государственного университета систем управления и радиоэлектроники. А именно внедрены:

1. Модели адаптивного обучения.
2. Методика разработки адаптивных электронных курсов.
3. Инструментальная система разработки адаптивного учебного контента.
4. Система адаптивного обучения.

Использование указанных результатов позволяет:

- 1) создавать адаптивные электронные курсы;
- 2) реализовывать технологию адаптивного обучения;
- 3) сократить затраты на проведение работ, связанных с разработкой адаптивных электронных курсов.

Результаты использования программного обеспечения показали:

1. Инструментальная система разработки адаптивного учебного контента:

- 1) обеспечивает простой и гибкий подход к разработке адаптивных электронных курсов;
 - 2) обеспечивает возможность создания междисциплинарных связей при создании адаптивных электронных курсов;
 - 3) обеспечивает возможность проведения глубокого анализа индивидуальных траекторий, получаемых в результате прохождения студентами электронных курсов.
2. Инструментальная система разработки адаптивного учебного контента может использоваться:
- 1) для реализации технологии адаптивного обучения при учебно-методическом обеспечении учебных дисциплин;
 - 2) для реализации уникальных учебных задач и создании индивидуальных адаптивных электронных курсов по принципу конструктора;
 - 3) в качестве средства повышения эффективности образовательных программ при реализации дистанционной формы обучения.
3. Использование указанных результатов позволило:
- 1) реализовать на факультете дистанционного обучения ТУСУР технологию адаптивного обучения;
 - 2) разработать адаптивный электронный курс «Информатика», соответствующий рабочей программе дисциплины «Информатика» ТУСУР (156 учебных модулей, 1146 тестовых вопросов, 182 субкомпетенции);
 - 3) повысить успеваемость студентов, обучавшихся с применением технологии адаптивного обучения (46 чел.) в среднем на 15% по сравнению со студентами, обучавшимися по традиционной технологии (46 чел.);
 - 4) сократить затраты на построение технологии адаптивного обучения на 37%.

Председатель комиссии:



подпись

Городович А.В.

ФИО

30.03. 2021 г.

Члены комиссии:



подпись

Черкашина И.П.

ФИО

30.03. 2021 г.



подпись

Перминова М.Ю.

ФИО

30.03. 2021 г.

Министерство науки и высшего образования Российской Федерации
**Федеральное государственное бюджетное образовательное учреждение
 высшего образования**
**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
 РАДИОЭЛЕКТРОНИКИ»**



УТВЕРЖДАЮ

Проректор по учебной работе

 П.В. Сенченко

« 15 » марта 2021 г.

АКТ

о внедрении результатов диссертационного исследования в учебном
 процессе ТУСУР
 Кречетова Ивана Анатольевича

Подтверждаю, что основные научные положения, выводы и рекомендации кандидатской диссертации Кречетова Ивана Анатольевича на тему: «Модели, алгоритмы и инструментальные средства адаптивного обучения» внедрены в учебный процесс кафедры «Промышленной электроники» для подготовки магистров по направлению 11.04.04 «Электроника и нанoeлектроника» по дисциплине «Компьютерные технологии в научных исследованиях».

Использование результатов диссертационного исследования позволило повысить уровень подготовки магистров в области разработки электронных курсов.

Зав. каф. промышленной
 электроники



д.т.н., доцент
 Михальченко С.Г.