

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
(ФГАОУ ВО «НИУ «ВШЭ»)**

УДК 004.414.2:371.32

На правах рукописи

МАКСИМЕНКОВА Ольга Вениаминовна

**СОВЕРШЕНСТВОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ УЧЕБНОГО
НАЗНАЧЕНИЯ НА ОСНОВЕ МОДЕЛЕЙ ПРОЦЕССОВ ЖИЗНЕННОГО
ЦИКЛА КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫХ МАТЕРИАЛОВ**

Специальность 05.25.05 – «Информационные системы и процессы»

Диссертация на соискание учёной степени
кандидата технических наук

*Научный руководитель:
доктор технических наук,
доцент
Зыков Сергей Викторович*

Москва – 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
ГЛАВА 1..... ИЗМЕРЕНИЯ В ОБРАЗОВАНИИ: МАТЕМАТИЧЕСКИЕ МОДЕЛИ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.....	25
1.1 Задача измерений в образовании и смежные с ней	25
1.1.1 Базовые определения, используемые в работе.....	25
1.1.2 Психофизиологические основы	26
1.1.3 Некоторые стандарты и указания	27
1.1.4 Постановка задачи эффективной информационной поддержки процессов измерений в образовании	27
1.2 Бизнес-процессы высших учебных заведений.....	28
1.3 Классификация тестовых заданий.....	30
1.3.1 Традиционная классификация тестовых заданий.....	31
1.3.2 Компьютерно-ориентированная классификация тестовых заданий.....	32
1.3.3 Классификация тестовых заданий, применённая в КИМРА	35
1.4 Форматы представления тестовых заданий	36
1.5 Современные средства автоматизации измерений в образовании	39
1.5.1 Системы и комплексы, включающие толстый клиент.....	40
1.5.2 Системы и комплексы, включающие тонкий клиент	42
1.6 Требования к ПСУН поддержки ЖЦ КИМ.....	50
1.7 Выводы и результаты по главе	51
ГЛАВА 2. МАТЕМАТИЧЕСКИЕ МОДЕЛИ ИС КИМРА	52
2.1 Математические модели тестирования.....	52
2.1.1 Классическая теория тестирования	53
2.1.2 Современная теория тестирования.....	57
2.1.3 Теория тестлетов и модели компьютерного адаптивного тестирования	61
2.2 Модель жизненного цикла контрольно-измерительного материала.....	64
2.3 Модель контрольно-измерительного материала	65
2.4 Модель результата контрольно-измерительного мероприятия, использующего контрольно-измерительный материал	71
2.5 Модель связи контрольно-измерительного материала с результатами обучения и учебными материалами	72
2.5.1 Формализация через таксономию дидактических единиц	72
2.5.2 Фолксономия логических атрибутов, служащих для сегментации	74
2.5.3 Анализ взаимосвязей между КИМ	75
2.6 Модель открытой экспертизы КИМ.....	79

2.7 Шкалирование	82
2.7.1 Типы шкал измерений	82
2.8 Выводы и результаты по главе	83
ГЛАВА 3.МЕТОДЫ И АЛГОРИТМЫ	85
3.1 Композиция КИМ.....	85
3.1.1 Сборка компьютерных тестов.....	86
3.2 Задача оптимальной композиции КИМ с учётом таксономии дидактических единиц.....	88
3.2.1 Интерактивный алгоритм оптимальной композиции КИМ	90
3.3 Доставка КИМ	93
3.3.1 Доставка КИМ при компьютерном взаимном оценивании	94
3.4 Выводы и результаты по главе	101
ГЛАВА 4.РЕАЛИЗАЦИЯ СИСТЕМЫ КИМРА	103
4.1 Архитектура распределённой информационной системы КИМРА	103
4.2 Подсистема «Банк КИМ»	106
4.2.1 Требования к интернационализации	107
4.2.2 Версионирование данных в КИМРА	109
4.2.3 Требования к банку КИМ информационной системы КИМРА	110
4.3 Подсистема «Редактор тестовых заданий»	111
4.3.1 Требования к визуальному редактору тестовых заданий	111
4.4 Подсистема «Открытая экспертиза КИМ»	113
4.5 Подсистема «Анализ результатов ИвО».....	115
4.5.1 Обзор пакетов R для обработки результатов тестирования	116
4.6 Подсистема «Управление жизненным циклом КИМ».....	117
4.7 Подсистема защищённого хранения персональных данных.....	118
4.8 Подсистема «Оповещения».....	119
4.9 Подсистема «Проведение контрольно-измерительных мероприятий».....	121
4.9.1 Модуль «Обеспечение доставки».....	121
4.9.2 Модуль «Проведение взаимного оценивания»	121
4.9.3 Средства разработки	125
4.10 Апробация и внедрение	125
4.11 Выводы и результаты по главе	127
ЗАКЛЮЧЕНИЕ	128
Обсуждение результатов	129
Текущее состояние программного комплекса КИМРА	130
Перспективные направления развития темы.....	131

БИБЛИОГРАФИЧЕСКИЙ СПИСОК	132
ПРИЛОЖЕНИЕ 1. ГЛОССАРИЙ.....	157
П 1.1. Процессы и роли измерений в образовании.....	157
П 1.2. Образование и измерения в образовании	157
П 1.3. Измерения	161
ПРИЛОЖЕНИЕ 2. ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ	164
ПРИЛОЖЕНИЕ 3. РЕАЛИЗАЦИЯ АЛГОРИТМА РАНДОМИЗАЦИИ.....	165
ПРИЛОЖЕНИЕ 4. РУБРИКИ ЗАДАНИЯ WEBQUEST	168
ПРИЛОЖЕНИЕ 5. АРТЕФАКТЫ ПРОЕКТИРОВАНИЯ ПО.....	172
П 1.4. Диаграммы прецедентов использования	172
ПРИЛОЖЕНИЕ 6. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ДАННЫХ	176
ПРИЛОЖЕНИЕ 7. АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ КИМРА	179

ВВЕДЕНИЕ

Актуальность. Современное образование отличается разнообразием подходов к обучению и контролю знаний. Развитие компьютерной техники и телекоммуникационных технологий позволяет постоянно снижать издержки уже существующих методик и технологий, повышая эффективность программных средств учебного назначения (ПСУН) [1], и предлагать новые решения в области образования. Так, развитие глобальных компьютерных сетей и облачных технологий позволило на новом уровне организовать коммуникацию в процессе обучения и обеспечить доступ к образовательным ресурсам во всемирном масштабе, а также сформировало такие направления как дистанционное обучение [*distance learning, e-learning*] и массовые открытые онлайн курсы [*massive open online courses, MOOCs*].

Повышение эффективности учебных процессов упомянутых направлений требует разработки специализированных многокомпонентных программных систем или тщательной селекции и компоновки уже существующих. Развитие информационно-телекоммуникационных технологий отчасти стало определять подходы к обучению и способы их реализации. Примерами таких учебных подходов являются образование, основанное на онтологиях [*ontology-based education*] [2, 3], обучение посредством облачных решений [*cloud-based learning*] [4, 5] и др. Поскольку подсистемы контроля знаний включены в надсистемы поддержки учебного процесса на разных уровнях или между ними присутствует межпрограммное взаимодействия, в данном исследовании сосредоточимся на изучении вопросов, связанных с программными системами, отнесённым в классификации А.И. Башмакова и И.А. Башмакова в [6] компьютерными системами контроля знаний (КСКЗ). Автор, тем не менее, отдаёт себе отчёт, что появление распределённых и облачных решений размыло границы между указанными классами программных средств и классификация нуждается в существенном пересмотре, поэтому мы будем говорить о *программных средствах поддержки измерений в образовании* (ПСПИВО) и сосредоточимся на классе *распределённых информационных систем учебного назначения* (РИСУН).

Многообразие видов контроля знаний и существенные отличия связанных с ними процессов определяет наличие разнообразных ПСПИвО. Например, программы проведения опросов и тестирований, системы поддержки взаимного оценивания [7, 8, 9], программные комплексы сопровождения олимпиад [10, 11, 12] и др. Практически любая подсистема РИСУН, автоматизирует лишь часть процесса жизненного цикла контрольно-измерительного материала, который начинается ещё до своего попадания в хранилище контрольно-измерительных материалов и не заканчивается участием в контрольно-измерительных мероприятиях [13].

Передача контрольно-измерительного материала из одной компьютерной системы в другую актуализирует вопросы интероперабельности особенно в рамках РИСУН. На уровне реализации учебных компьютерных систем интероперабельность непрерывно совершенствуется, соответствующие форматы для компьютерных тестовых систем обобщены автором совместно с коллегами [14], на мировом уровне существуют стандарты передачи данных и мета-данных в системах электронного обучения [15, 16, 17]. К настоящему моменту в какой-то мере решены задачи интероперабельности систем поддержки учебного процесса [*learning management systems, LMSs*] на уровне контингента, структуры дисциплин и выставленных оценок [16, 17].

Тем не менее, актуальными и пока неразрешёнными остаются следующие проблемы:

- представление контрольно-измерительных материалов в информационных системах, не зависимо от вида поддерживаемого мероприятия. Например, оптимизация, путём унификации, управления контрольно-измерительными мероприятиями в рамках *LMS*;
- корректное повторное использование контрольно-измерительных материалов, то есть соблюдение соглашений и регламентов, гарантирующих достоверность и валидность получаемых с их помощью результатов. Например, использование заданий, подготовленных для контроля знаний синтаксиса

языка программирования C# 4.0 на контингенте учащихся, изучающих C# 6.0;

- хранение и адекватное версионирование контрольно-измерительных материалов с поддержкой существенных для повторного использования и управления связей.

Объект данных, представляющий контрольно-измерительный материал, требует выявления наиболее полного набора характеристик по отношению к решаемым задачам и такой формализации, чтобы его реализация в компьютерных системах обеспечивала бесшовную интеграцию подсистем средств контроля с другими подсистемами.

Для РИСУН вопросы интеграции внутренних и внешних модулей приобрели особую актуальность и требуют решения как на технологическом уровне, так и на уровне математического моделирования.

Предложенное в работе исследование относится к контролю и измерению результатов обучения при помощи компьютеров и посвящена актуальной задаче построения математических моделей, лежащих в основе распределённых программных систем поддержки контрольно-измерительных мероприятий. В подобных системах контрольно-измерительный материал понимается как объект данных и согласован с требованиями к интероперабельности, предъявляемым к современным программным системам в образовании [16, 17].

История и актуальные направления исследований в области измерений в образовании в России и мире. Область измерений в образовании (ИвО) имеет сравнительно короткую историю – чуть больше ста лет. Наиболее проработанной является область компьютерного тестирования.

Первые работы, которые можно соотнести с измерениями в образовании, в их современном понимании начали появляться в конце XIX века в США. Большинство из них составляли попытки установить связь и получить численную характеристику этой связи между результатами психологических тестирований и успевае-

мостью учащихся университетов и колледжей. Наиболее значимой историкой математической статистики считают работу Ф. Гальтона [18], в которой он ввёл понятие коэффициента корреляции. Однако первые значимые результаты по описанию корреляций применительно к психологическим тестам были получены только в начале XX столетия. Ч. Спирмен критически пересмотрел исследование Ф. Гальтона в области корреляций, улучшил его результаты и представил её формулу [19]. В настоящее время эта формула известна как формула Спирмена и применяется в классической теории тестирования [*classical test theory, CTT*] при установлении валидности и надёжности тестов. В 1911 году Ч. Спирмен и Э. Браун независимо друг от друга представили формулу [20, 21] (формула Спирмена-Брауна). В работе [22] описаны её применение и интерпретации в рамках *CTT*.

Появление математического аппарата привело к стремительному развитию и популяризации тестов личности и тестов достижений [22]. Р.Е. Трауб [23] указывает, что в период с 1910 г. по 1940 г. формула Спирмена-Брауна широко обсуждалась и критиковалась исследователями. К значимым результатам в контексте измерений в образовании того периода можно отнести появление термина «коэффициент надёжности» в 1929 году, описание надёжности частей теста и ретестовой надёжности, а также опубликование формулы Кьюдера-Ричардсона [23].

В 30-е годы XX века профессор Калифорнийского университета Е. Нейман и профессор Лондонского университета К. Пирсон развили общую теорию проверки статистических гипотез, а А.Н. Колмогоров и Н.В. Смирнов заложили основы непараметрической статистики. А.Н. Колмогоров дал наиболее совершенное аксиоматическое построение теории вероятностей, связав ее с одним из важнейших разделов современной математики – метрической теорией функций. В тот же период формализуется понятие измерения физической величины, как совокупность операций по применению технического средства, хранящего единицу физической величины, обеспечивающих нахождение соотношения (в явном или неявном виде) измеряемой величины с ее единицей и получение значения этой величины [24].

Понятие измерений для социальных наук было введено психологом С. Стивенсом в 1946 году [25], он обобщил материалы по количественным измерениям,

проводимым в других предметных областях (физике, астрономии и др.) и ввёл определение измерения для социальных наук. Измерение было предложено трактовать в широком смысле как процедуру приписывания чисел некоторым характеристикам объектов в соответствии с определенными правилами. Работа Стивенса послужила отправной точкой для уточнения и систематизации математических методов психологических измерений, а также выделения из них измерений в области образования.

В 1950-е гг. опубликованы первые фундаментальные труды по *СТТ* и представлены её модели и методы [22]. Краеугольным камнем *СТТ* достаточно долго оставалась проблема зависимости характеристик качества измерительного инструмента (теста) от свойств популяции испытуемых. Такая зависимость проявлялась как результат применения методов математической статистики напрямую к обработке тестовых данных. Обобщение, описание и обоснование проблем классической теории тестирования приводятся в трудах по основаниям современной теории тестирования (например, [26]).

Первое решение было предложено в середине XX века учёным Д. Рашем в работе [27], которая считается одной из передовых работ, лёгших в основу современной теории тестирования [*Item Response Theory – IRT*], также известной в российских источниках как теория параметризации педагогических тестов [28, 29, 30]. Проблеме выбора наиболее подходящего русскоязычного эквивалента к термину «*item response theory*» посвящена работа [30], чаще всего используется термин «современная теория тестирования». В данной работе принято решение использовать латинские сокращения ввиду отсутствия в них разночтений.

К. Бирнбаум работал с двух и трёхпараметрическими моделями, которые в последствие были названы его именем. В работе 1968 года [31] он описал информационную функцию теста, основанную на понятии информации Фишера. Данный результат был бы невозможен без использования вероятностного математического аппарата, который окончательно сформировался после появления функции правдоподобия $L_x(\theta)$. Это был первый революционный результат, определённый Фишером как функция аргумента θ для заданного значения x [32].

IRT решила многие проблемы, связанные с применением *CTT*, но и она имеет ограничения, требующие специального внимания [33]: плохие результаты в условиях малой выборки; необходимость наличия банка апробированных и откалиброванных заданий; вычислительная сложность моделей *IRT* и необходимость аппроксимирования.

Кроме того, расчёт параметров математических функций (характеристических кривых заданий), описывающих свойства заданий, предполагает достаточно большое число испытуемых. При малом числе испытуемых получаемые значения параметров таких функций очень ненадёжны, а потому в таких случаях на результаты применения *IRT* нельзя полагаться в полной мере [34].

Попытки преодолеть барьер малых выборок в современной теории тестирования делались и делаются постоянно. Время от времени учёные и исследователи применяют к обработке результатов тестирования методы, позволяющие работать с малыми выборками, и их модификации. Так, в работе Д. Магисом и Б. Феконом в [34] приводится описание применения для построения *DIF* [*differential item functioning*] модификации классического метода Ангоффа [*Angoff's delta plot method*], предложенного в 1973 году в работе [35].

Отметим также, что в [34] указывают на слабую изученность точности стандартных методов построения *DIF* в условиях малой выборки, и приводят ссылку на работу С. Паршал и Т. Миллер [36], в которой проведено сравнение прямого и асимптотического критериев *Mantel-Haenszel* для получения *DIF* в условиях малой выборки. К 2014 году модификация для получения *DDIF* [*dichotomous DIF*] была реализована в виде пакета статистического проекта *R* [37], описанного в работе [38].

Основные идеи *IRT* привели к возрождению концепции адаптивного отбора заданий. Здесь говорится о возрождении ввиду того, что сама идея адаптивного предоставления испытуемому заданий не нова. Она зачастую реализуется экзаменаторами при проведении устных контрольных мероприятий, а одна из первых работ, в которой описывается предоставление заданий в зависимости от умственного возраста участника тестирования, принадлежит Т. Симону и А. Бине и появилась в

1905 году [39]. Несмотря на то, что *IRT* стала основой для развития адаптивного тестирования в его компьютерной форме – компьютерного адаптивного тестирования [*computerized adaptive testing* – *CAT*], потребовались десятилетия для того, чтобы вычислительные мощности компьютеров возросли настолько, что появилась возможность реализовывать модели *CAT*.

Первая реализация идей *CAT* появилась в разработке министерства обороны США *Armed services vocational aptitude battery* (ASVAB) в середине 80-х гг. XX века [40]. Повсеместный переход от бланкового тестирования к компьютеризированным версиям начался с появления адаптивной версии сертификационного экзамена *NCLEX/CAT*, проводимого *National Council of state Boards of Nursing*, вслед за которой последовала адаптивная версия экзамена *Graduate record examination* (GRE) и многие другие. Подробная история развития программ поддержки *CAT*, а также связи развития компьютерной техники с популяризацией компьютерных форм контроля знаний в США, изложены в [13].

Появление реализаций моделей *CAT* привело к постановке новых вопросов, относительно дальнейших путей развития теории и методов адаптивного тестирования. Во многом проблемы связаны не только с самими тестами, но и с технологией, то есть доставкой и предоставлением тестов испытуемому с помощью компьютера [41, 42]. Характер исследований и историю развития алгоритмов *CAT* в период конца 90-х гг. XX века и первого десятилетия XXI века легко проследить по статьям альманаха под редакцией Ван дер Линдена и Гласа [40] и [43].

Попытка разрешить проблему необходимости проведения масштабных апробаций, необходимых для калибровки заданий, включаемых в банки тестовых заданий, привела к привлечению байесовской модели, и в 1999 году научный коллектив под руководством Говарда Вайнера и др. [44] представил *TRT*, построенную на понятии тестлета. В рамках *TRT* под тестлетом понимают группу заданий, которые могут быть разработаны совместно (как блок заданий), что означает возможность единого управления ими [33]. Сейчас методы и модели *CAT* активно развиваются и после обоснования компьютерно-ориентированных моделей теории тестлетов

[*Testlet Response Theory, TRT*] [45] повсеместно внедряются на уровне массовых тестирований, таких как *AELST*, *SAT* [46] и проч.

История развития тестологии в России не насчитывает и полувека. Несмотря на вовлечённость таких идеологов математической статистики как А.Н. Колмогоров и В.Н. Смирнов в международное научное сообщество, результаты их работ находили отражение в развитии зарубежной тестологии. В СССР тестология до второй половины XX века фактически находилась под запретом.

Без сомнений, одним из первых ученых, посвятивших свою работу популяризации педагогических тестов в советском и российском образовании, является В.С. Аванесов. Согласно [28] первое упоминание о *IRT* в русскоязычной литературе датируется 1995 годом и вводится в учебном пособии М.Б. Челышковой [47]. Среди российских ученых, активно занимавшихся обработкой результатов педагогических измерений на базе *IRT*, стоит выделить Ю.М. Неймана и В.А. Хлебникова, работа которых связана не только с научными достижениями в области педагогических измерений, но и с организацией инфраструктуры. Так, на базе Лаборатории тестирования Московского педагогического государственного университета (МПГУ) был организован Центр тестирования выпускников общеобразовательных учреждений Российской Федерации при МПГУ.

В настоящее время активную работу в области математических моделей и вопросов автоматизации измерений в образовании ведут научные группы под руководством таких исследователей, как Е.Ю. Карданова [29], Д. Нардюжев [48].

В области измерений в образовании на протяжении многих лет остаются популярными узкоспециализированные исследования. Под ними понимаются исследования частных случаев, например, дифференцирующий способности тестовых заданий различных типов для студентов определённых дисциплин или направлений [49, 50]. Отметим, что частным случаям, например, разработке тестовых заданий определённых типов посвящаются циклы статей и книги [51].

В контексте математического аппарата измерений в образовании анализируются и исследуются границы применимости моделей и методов *CTT* и *IRT*, разрабатываются новые и совершенствуются существующие модели [52, 53, 54, 55, 56].

В связи с особенностями компьютерных форм контроля знаний (в том числе *СТ* и *CAT*), с 2000 года активно ставятся задачи, проводятся исследования и предлагаются реализации, посвящённые вопросам разработки, поддержки, использования и формирования банков заданий [57, 58, 59, 60]. Актуальность и насущность этих задач сейчас настолько высока, что один из лидеров мирового тестирования – Американская психометрическая ассоциация (*American Psychometric Association*) обновляет в 2015 году руководство по тестированию и контролю в психологии, включающее свежие версии разделов по разработке тестовых заданий, формированию и управлению [61].

В настоящее время на пике популярности на международном уровне находятся алгоритмы и математическое обеспечение *CAT* [62]. В связи с обоснованием и внедрением в практику массовых тестирований моделей *TRT* [45, 33] большую популярность приобрели задачи их уточнения и расширения, например, для политомических тестов [63, 64].

Активное обучение: основы, тенденции и автоматизация. Гибкие возможности автоматизации позволили реализовывать в информационных системах повсеместно обсуждаемые в настоящее время методы активного обучения [*active learning*]. Первоначально под активным обучением понимались только учебные практики, согласованные с моделями проектирования обучения, предложенными Б. Блумом, М. Скривеном и Р. Ганье [65, 66, 67], в которых учащиеся активно вовлечены в процесс [68].

Позднее в активное обучение были включены техники формирующего оценивания [*formative assessment*]. Понятие формирующего оценивания было введено М. Скривеном в 1967 году [67]. Он предложил считать учебные материалы окончательно сформированными только по результатам их апробации при обучении студентов. В результате формирующего оценивания получают данные, на основе которых перестраивают дальнейшее обучение.

Существенным компонентом оказывается понятие информативной обратной связи (ИОС) [*formative feedback*] как связи между студентом и преподавателем [69,

70]. Ввиду отсутствия единой терминологии на русском языке автором данной работы в 2014 году предложено следующее определение: ИОС – неотъемлемая часть *формирующего* контроля, обеспечивающая разнородные данные для работы механизма изменения процесса обучения. Разнородность данных подразумевает то, что они могут быть количественными или качественными, цифровыми или аналоговыми, могут записываться на любом носителе [71]. Актуальными на протяжении многих лет остаются исследования в области применения ИОС [70], достаточно полный обзор исследований, связанных с применением ИОС в высшем образовании, можно найти в работе С. Эванс [72].

Методы формирующего оценивания имеют сложную структуру бизнес-процессов и разнородные артефакты, появляющиеся при их реализации. Например, проведение взаимного оценивания [*peer assessment*] среди студентов требует от преподавателя поддержки сложно-структурированного процесса: подготовки контрольно-измерительных материалов (КИМ), включающих рубрики [*rubrics*] для оценивания, выдачи заданий студентам, сбора работ, их анонимизации и рандомизации, повторной выдачи работ на проверку, сбора проверок, проставления агрегированных результатов. Не смотря на серьёзную проработанность методики (см. обзоры [73, 74]), задачи взаимного оценивания актуализировались и стали популярными среди преподавателей в конце 1990-х [75], когда стали доступны сетевые ПСУН, реализующие специальные алгоритмы рандомизации и поддерживающие все перечисленные этапы процесса.

Последнее десятилетие ознаменовано формированием по всему миру междисциплинарных научных групп в области активного обучения. Основными направлениями исследований являются:

- 1) изучение и валидация уже существующих инструментов активного обучения и формирующего оценивания, а также конструирование новых [73, 76, 75, 77];
- 2) формализация, моделирование и автоматизация активного обучения [9, 78].

Открытое образование и наука. Развитие информационно-коммуникационных технологий привело к появлению и активному развитию в последние годы подходов открытого образования [*open education*] [79] и формирование хранилищ открытых образовательных ресурсов [*open educational resources, OER*] [80, 81, 82]. Открытым образовательным ресурсом называют любой учебный ресурс (включая учебные планы, видео, книги, КИМ и др.), находящийся в открытом доступе для любого участника учебного процесса, не требующий платы за доступ или лицензию [82, 83].

Принцип открытости образования, позволяет по-новому взглянуть на понимание качества КИМ. Качество учебных материалов определяется не узкой группой экспертов, а сообществом специалистов и заинтересованных лиц, задействованных в процедурах открытой экспертизы. Адаптация методов открытой экспертизы к процессам формирования банков КИМ отдельных учебных заведений и дисциплин, их автоматизация и внедрение позволит: 1) снизить издержки на экспертизу КИМ и 2) повысить их качество.

Возникновение и популяризация ведущими вузами мира *МООС* актуализировало среди исследователей проблему сравнимости образовательных результатов студентов, а также самих курсов [84]. Кроме того, международное учебное сообщество пытается разрешить проблему установления соответствия между показателями, полученными при прохождении *МООС* и принятыми в традиционном учебном процессе [85]. Например, можно ли, как и с какой оценкой засчитывать результаты *МООС* или как интегрировать *МООС* в очную учебную дисциплину [86, 87]?

В заключение отметим, что специфика учебного процесса серьёзно влияет на архитектурные решения в РИСУН и используемые в них модели, т.е. актуальными являются вопросы проектирования и разработки модулей и подсистем контроля (измерения) знаний для соответствующих РИСУН. Основные трудности связаны с выявлением процессов, подлежащих автоматизации, поскольку ключевым элементом автоматизации контроля результатов общения является поддержка процессов

контрольно-измерительных мероприятий. Задачи построения математических моделей КИМ, на базе которых строятся хранилища данных для КИМ, являются исключительно важными.

Удачно подобранные математические модели и способы их реализации влияют на гибкость РИСУН в условиях высокой вариативности задач в образовании, а также повышают достоверность результатов контроля.

Несмотря на достаточно глубокую проработанность затронутых тем, мало внимания при этом уделяется важнейшим направлениям, связанным с развитием инфраструктуры измерений, помощью в решении технологических проблем, повторным использованием КИМ. Многие из этих тем упираются в сложную мультидисциплинарную проблему управления жизненным циклом КИМ, особенно в областях со стремительно изменяющимися технологиями и сменяющимися парадигмами. Например, актуализация тестовых заданий по программированию в связи со сменой версии языка программирования.

Отметим, что при работе над диссертацией автору не удалось обнаружить системных исследований, посвящённых управлению жизненным циклом контрольно-измерительных материалов. Исходя из современного уровня развития информационных технологий и быстрого развития методов активного обучения, продуцирующих комплексные артефакты, бесспорно актуальными становятся вопросы построения моделей для РИСУН, поддерживающих процессы ЖЦ КИМ. В данном исследовании рассматриваются общие требования к программному комплексу КИМРА (Контрольно-Измерительные Материалы – Разработка и Анализ), предлагаются математические модели, подходящие для построения универсального хранилища КИМ, математические методы, оптимизирующие процесс доставки КИМ и реализация некоторых важнейших подсистем.

Объектом исследования являются распределённые информационные системы учебного назначения.

Предметом исследования являются информационные и математические модели, алгоритмы в РИСУН, ориентированных на ИвО.

Целью диссертационной работы является совершенствование РИСУН путём оптимизации процессов жизненного цикла контрольно-измерительных материалов на основе оригинальных математических и информационных моделей РИСУН, обеспечивающих унификацию и повышение эффективности РИСУН, ориентированных на ИвО.

Для достижения поставленной цели сформулированы следующие задачи диссертационного исследования:

- 1) предложить модель КИМ, включающую выявленные в результате анализа существенные характеристики КИМ, связанные с их использованием на различных стадиях ЖЦ КИМ и разграничивающую атомарные и композитные КИМ;
- 2) разработать модели связи КИМ с результатами обучения, позволяющие исследовать взаимоотношения между КИМ в рамках предметных областей;
- 3) исследовать уровни и направленность информативной обратной связи в процессах формирующего оценивания;
- 4) предложить формализацию процессов формирующего оценивания с учётом информативной обратной связи разных уровней;
- 5) разработать алгоритм оптимизации подбора КИМ при создании на основе ограничений на характеристики нового композитного КИМ;
- 6) предложить алгоритм рандомизации рецензентов по подачам при доставке КИМ взаимного оценивания;
- 7) спроектировать архитектуру и реализовать прототип РИСУН, реализующую модель КИМ, модели связей КИМ с результатами обучения и предложенные алгоритмы.

Научная новизна. В диссертационной работе предложены улучшения поддержки жизненного цикла контрольно-измерительных материалов и процессов доставки КИМ. В частности, можно выделить следующие результаты, характеризующиеся **научной новизной**:

- 1) обоснованы требования к РИСУН поддержки ЖЦ КИМ, предложен и апробирован набор подсистем, входящих в состав информационной системы поддержки процессов ЖЦ КИМ (пункт 6 паспорта специальности);
- 2) на основании указанных требований к средствам поддержки ЖЦ КИМ предложена модель КИМ, положенная в основу функционирования РИСУН поддержки ЖЦ КИМ (пункт 1 паспорта специальности);
- 3) разработаны графовые модели для представления связей КИМ с результатами обучения, позволяющие единообразно управлять КИМ при реализации сценариев ИвО (пункт 1 паспорта специальности);
- 4) предложен и реализован алгоритм многокритериальной оптимизации композитного КИМ путём подбора КИМ из банка КИМ (пункт 3 паспорта специальности);
- 5) предложен и реализован алгоритм рандомизации рецензентов в процессах взаимного оценивания (пункт 3 паспорта специальности).

Теоретическая значимость работы определена важностью совместного рассмотрения математических основ, процедур и бизнес-процессов измерений в образовании с учётом необходимости их реализации в программных средствах.

Практическая полезность определяется возможностью непосредственного применения разработанных методов, форматов, алгоритмов, информационной системы и её подсистем в учебном процессе.

Внедрение предложенных бизнес-процессов открытой экспертизы КИМ *позволит снизить* расходы на организацию процесса экспертизы КИМ. Кроме того, открытая экспертиза, поддержанная средствами программного комплекса КИМРА, представляет собой методику самоподготовки в рамках активного обучения и может быть использована для *повышения качества обучения*.

Методы исследований и достоверность результатов. При выполнении диссертационного исследования применялись методы общей теории измерений, теорий измерений в образовании, математической статистики, дискретной математики, численных методов, современных технологий моделирования (*UML, BPMN*)

и программирования (ООП, многокомпонентное программирование, распределённые системы, облачные технологии).

Достоверность изложенных в работе результатов подтверждается проверкой математических моделей с проведением имитационного моделирования, экспертными оценками архитектурных и других проектных решений, результатами тестирования и апробации моделей, методов и программных средств.

Реализация результатов. Результаты работы использованы в учебном процессе Московского и Пермского кампусов Национального исследовательского университета «Высшая школа экономики» и Санкт-Петербургском Национальном университете информационных технологий, механики и оптики.

Апробация работы. Основные положения и результаты диссертации докладывались и обсуждались на:

- семинаре «Мобильные сценарии в инфраструктуре современного вуза» технологического центра *Microsoft*, 2017;
- методической мастерской ФГБОУ ВО «НИУ «ВШЭ» «Взаимное оценивание студентов: от идеи к реализации», научных семинарах департамента программной инженерии факультета компьютерных наук и других подразделений ФГБОУ ВО «НИУ «ВШЭ»;
- 20th, 19th и 18th International Conference on Interactive Collaborative Learning (ICL 2017, 2016 и 2015).
- 20th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES-2016);
- X-ой и XIV-ой открытой Всероссийской конференции «Преподавание информационных технологий в Российской Федерации», 2012, 2016;
- 10th International Conference on Computer Science & Education (ICCSE 2015);
- III-ей и IV-ой Всероссийской научно-практической конференции «Информационные технологии в образовании XXI века», 2013, 2014;

- Международной научно-практической конференции «Инновационные информационные технологии», 2013;
- Международной научно-практической конференции «Информатизация образования – 2011», 2011;
- Spring/Summer young researcher's colloquium of Software Engineering (SYRCoSE 2010, SYRCoSE 2011), 2010, 2011.

Результаты диссертации были внедрены:

- при разработке требований и реализации веб-платформы взаимопомощи обучаемых в АНО «Научный центр социально-экономического развития малых городов и сельских поселений»;
- при разработке прототипа системы адаптивного обучения *FALS (Formal Concept Analysis Adaptive Learning System)* в Центре исследования СПО ИО НИУ ВШЭ;
- в общеуниверситетском магистерском курсе ФГБОУ ВО «НИУ «ВШЭ» «Проектирование взаимодействия с пользователем»;
- в дисциплине «Поведение потребителей», преподаваемом в рамках программы НИУ ВШЭ Пермь «Smart marketing: данные, аналитика, инсайты» для первого курса;
- в учебный курс повышения квалификации «Школа педагогического мастерства» дирекции общего образования ФГБОУ ВО «НИУ «ВШЭ».

Основные результаты, полученные в процессе выполнения диссертационной работы, опубликованы в 19 печатных работах, включая 8 – в журналах, рекомендованных ВАК, 5 – в изданиях, индексируемых *Scopus*, 6 – в материалах общероссийских и международных конференций.

Публикации в журналах из перечня ВАК

1. Незнанов А.А., Максименкова О.В. Концептуальные и логические модели для распределённых программных систем поддержки контрольно-измерительных мероприятий // Программные системы: теория и приложения. 2017. Т. 8. № 4. С. 31-46.

2. Подбельский В.В., Бабич К.С., Максименкова О.В. Об обеспечении интероперабельности программных средств поддержки тестирования в образовании // Информационные технологии. 2016. № 7. С. 535-541.
3. Kolomiets A., Maksimenkova O., Neznanov A. On business processes of computer-supported collaborative learning: A case of peer assessment system development // Business informatics, No. 4, 2016. pp. 35-46.
4. Максименкова О.В., Подбельский В.В. Практика использования открытых данных в курсе «Программирование» образовательной программы бакалавриата «Программная инженерия» // Образование и наука. 2016. № 10.
5. Максименкова О.В., Папушина Ю.О. Формирующее оценивание при внедрении метода интеллект-карт в процесс обучения на магистерской программе «Маркетинг» // Вестник Российского университета дружбы народов. Серия: Информатизация образования. 2015. № 3. С. 24-35.
6. Максименкова О.В., Незнанов А.А., Подбельский В.В. О формирующем контроле и информативной обратной связи при проектировании учебных курсов по программированию // Вестник Российского университета дружбы народов. Серия: Информатизация образования. 2014. № 4. С. 37-48.
7. Подбельский В.В., Максименкова О.В. Особенности формулировок тестовых заданий по программированию // Качество. Инновации. Образование. 2013. № 4. С. 3-10.
8. Подбельский В.В., Максименкова О.В., Казаков М.А. Визуализация структуры педагогического теста // Ярославский педагогический вестник. 2012. Т. III. № 1. С. 21-31.

Публикации в изданиях, индексируемых Scopus

9. Maksimenkova O., Neznanov A. The PASCA: A Mail Based Randomized Blinded Peer Assessment System for Complex Artifact // Procedia Computer Science, Vol. 96, 2016. pp. 826-837.
10. Radchenko I., Maksimenkova O. Principles of Citizen Science in Open Educational Projects Based on Open Data // Proceedings of the 12th Central and Eastern European Software Engineering Conference in Russia. 2016.

11. Maksimenkova O., Papushina I., Kolomiets A. Digital educational mind maps: A Computer-Supported Collaborative Learning Practice on Marketing Master Program // Advanced in Intelligent Systems and Computing, 2016. P. 17-30.
12. Neznanov A., Maksimenkova O. Blended Learning in Software Engineering Education: the Application Lifecycle Management Experience with Computer-Supported Collaborative Learning // Proceedings of 18th International Conference on Interactive Collaborative Learning (ICL 2015). 2015. pp. 646-653.
13. Maksimenkova O., Podbelskiy V. On practice of using open data in construction of training and assessment tasks for programming courses // 10th International Conference on Computer Science & Education (ICCSE). 2015. pp. 233-236.

Публикации в других научных журналах и материалах конференций

14. Максименкова О.В., Подбельский В.В. О применении некоторых специальных критериев проверки нормальности распределения результатов педагогических измерений // Законодательная и прикладная метрология. 2014. № 6. С. 28-33.
15. Максименкова О.В. Вопросы локализации образовательного программного обеспечения с открытым кодом на примере EdX // Новые технологии в образовании. – Троицк-Москва, 2014. – С. 497-499.
16. Ахметсафина Р.З., Максименкова О.В. Активный обучающий подход при повышении квалификации преподавателей информатики // Информационные управляющие системы и технологии (ИУСТ-ОДЕССА-2013). Материалы Международной научно-практической конференции. – Одесса, 2013. – С. 50-52.
17. Максименкова О.В., Незнанов А.А. О роли и эффективных методах информативной обратной связи при контроле знаний по дисциплинам цикла «Программирование» // III-я Всероссийская научно-практическая конференция «Информационные технологии в образовании XXI века». Сборник научных трудов. – М., 2013. – С. 119-123.
18. Максименкова О.В., Подбельский В.В. Подготовительное тестирование по программированию на базе системы управления учебным процессом университета // Инновационные информационные технологии: Материалы международной научно-практической конференции. – М., 2013. – С. 338-345.

19. Maksimenkova O., Podbelskiy V. Educational tests in «Programming» academic subject development // SYCRoSE. Proceeding of the Spring/Summer Young Researcher's Colloquium on Software Engineering. 2011.

Личный вклад диссертанта. Основные результаты и положения, выносимые на защиту, получены автором лично. Подготовка результатов к публикации проводилась автором самостоятельно и в соавторстве со специалистами из прикладных областей, в которых внедрялись результаты, причём вклад диссертанта был определяющим.

Содержание работы по главам. Диссертация изложена на 153 страницах (без приложений), включает 11 таблиц, 24 рисунка, состоит из введения, четырёх глав, заключения, списка литературы из 247 наименований и 7 приложений.

В первой главе вводятся основные термины и определения, связанные с измерениями в образовании. Впервые в русскоязычной литературе формализовано понятие контрольно-измерительного материала (КИМ) и его жизненного цикла (ЖЦ КИМ). На основе анализа существующих классификаций и таксономий тестовых заданий предлагается оригинальная классификация тестовых заданий, ориентированная на использование в ПСУН и реализованная в КИМРА; исследуются форматы интероперабельности тестовых заданий и учебной информации в ПСУН; предлагается внутренний JSON-формат представления КИМ, включая тестовые задания, в КИМРА и определяются форматы интероперабельности; проводится обзор современных средств автоматизации измерений в образовании. Проводится обзор ПСУН, позволяющих накопление банков тестовых заданий. В заключении главы формулируются требования к хранилищу КИМ.

Во второй главе проведён обзор наиболее популярных теорий измерений в образовании. По результатам обзора существующих моделей и методов обработки результатов контрольно-измерительных мероприятий, выявлены характеристики КИМ и предложена формальная модель КИМ, использующегося в процессах компьютерных измерений и обеспечивающая поддержку ЖЦ КИМ. Обобщение задач сегментации и селекции КИМ с понятием дидактических единиц позволило пред-

ложить графовую модель связи КИМ с результатами обучения и учебными материалами (граф структуры содержания КИМ). По предложенной модели построена вторая графовая модель, отражающая взаимосвязи между КИМ (граф взаимосвязей КИМ).

В третьей главе введено понятие компьютерной доставки КИМ и предложены численные методы доставки КИМ в процессе взаимного оценивания. Поскольку методы доставки КИМ сильно зависят от типа КИМ, в главе сделан акцент на особенностях доставки КИМ в процессе взаимного оценивания, поставлена задача рандомизации рецензентов по рецензиям, предложен численный метод рандомизации, реализованный в модуле проведения взаимного оценивания РИСУН КИМРА.

В четвёртой главе описана архитектура РИСУН «Контрольно-измерительные материалы: разработка и анализ» (КИМРА), реализующей предложенную в работе модель КИМ, сформулированные методы и алгоритмы. Кроме того, сформулированы требования к подсистемам и модулям КИМРА, описаны особенности реализации интерфейсов и протоколов, обсуждаются результаты прототипирования, апробации и внедрения отдельных подсистем.

ГЛАВА 1. ИЗМЕРЕНИЯ В ОБРАЗОВАНИИ: МАТЕМАТИЧЕСКИЕ МОДЕЛИ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В данной главе ставится задача эффективной информационной поддержки ИвО. Для этого вводятся основные определения, связанные с ИвО и их автоматизацией. Формализуется понятие КИМ и ЖЦ КИМ, обсуждаются его разные варианты, например, тестовые задания (ТЗ), задания взаимного оценивания. С целью позиционирования ИвО и выявления основных задач по их автоматизации автором анализируются бизнес-процессы высших учебных заведений, выделяются подпроцессы контроля знаний и формулируются основные вопросы, связанные с ними.

Для уточнения требований к банку КИМ распределённого РИСУН приведён обзор ПСУН, имеющих подсистемы контроля знаний и представляющие возможность формирования банков. Для выявления требований к обобщённому внутреннему представлению КИМ в современных РИСУН будут исследованы классификации тестовых заданий, пригодных для реализации в ПСУН. Предполагается выявить сильные и слабые стороны существующих классификаций и предложить собственную, учитывающую самые современные типы тестовых заданий. Поскольку современное РИСУН предполагает управление ЖЦ КИМ не только тестовой формы, то в данной главе рассматриваются форматы интероперабельности. На основе существующих будет предложен собственный внутренний формат представления КИМ в РИСУН, позволяющий проводить гибкую конвертацию в наиболее распространённые форматы для упрощения межпрограммного взаимодействия.

1.1 Задача измерений в образовании и смежные с ней

1.1.1 Базовые определения, используемые в работе

Измерение [*measurement*] – сравнение конкретного проявления измеряемого свойства (измеряемой величины) со шкалой (частью шкалы) измерений этого свойства (величины) в целях получения результата измерения (оценки свойства или значения величины) [88].

Контрольный измерительный материал (КИМ):

- целенаправленно разрабатываемые материалы для осуществления контроля уровня сформированности профессиональных компетенций обучающихся. КИМ определяются в качественных и количественных показателях, которые способны ярко показать степень овладения знаниями, умениями и навыками [89];
- методический материал, который связан с какой-либо системой методических материалов, например, учебной дисциплиной [*course*]; подразумевает возможность получения ответа [*answer*] студента на задание [*assignment*] из КИМ; имеет схему и правила оценивания, позволяющие ставить в соответствие ответу качественную или количественную оценку [*grade*]; обладает характеристиками корректности, трудности, дискриминативности и другими, устанавливаемыми по результатам анализа ответов.

Жизненный цикл контрольного измерительного материала (ЖЦ КИМ) – набор активностей, отношений и зависимостей, в которых участвует КИМ и период времени от постановки задачи на разработку КИМ (или идеи его создания) до удаления из банка КИМ.

Тестовое задание [*test question, test item, item*] – элемент теста (утверждение, вопрос, упражнение, задание) и набор непротиворечивых правил его оценивания.

Тест [*test*] – описание процедуры педагогических измерений (тестирования), состоящее из проверенных на валидность и надежность системы тестовых заданий, стандартизированной схемы проведения и заранее спроектированной технологии обработки и анализа полученных результатов.

Не введенные здесь определения можно найти в [90, 91, 92]. Используемый в диссертационной работе глоссарий в 0.

1.1.2 Психофизиологические основы

В психометрической литературе иногда модели поведения при прохождении теста рассматриваются совместно с психофизиологическим моделированием участника тестирования [93]. Мы будем понимать модели поведения участника тестирования как математические модели его успеха или неуспеха в работе с КИМ.

Вопросы психофизиологического моделирования поведения участника тестирования, связанные с его личным опытом, мотивацией, усталостью и другими личностными характеристиками, оставим за рамками данной работы и детально рассматривать не будем. Вопросы тестирования личности рассмотрены в работах [94, 95].

1.1.3 Некоторые стандарты и указания

В данной работе используются материалы из следующих профессиональных стандартов и рекомендаций:

- Стандарты тестирования в образовании и психологии американской ассоциации исследований в образовании [95];
- ГОСТ Р 50779.10–2000 (ИСО 3534-1–93) Статистические методы. Вероятность и основы статистики. Термины и определения [96];
- ГОСТ Р ИСО 5479-2002. Статистические методы. Проверка отклонения распределения вероятностей от нормального распределения [97];
- ГОСТ Р 50779.21-96. Статистические методы. Правила определения и методы расчёта статистических характеристик по выборочным данным [98];
- РМГ 29-99 ГСИ. Метрология. Основные термины и определения [24];
- РМГ 83-2007. Шкалы измерений. Термины и определения [88];

1.1.4 Постановка задачи эффективной информационной поддержки процессов измерений в образовании

Середина 80-х годов XX века стала отправной точкой повсеместного внедрения компьютерных форм поддержки учебного процесса. Для различных областей и форм обучения (в том числе и контроля знаний) показан рост эффективности при использовании информационных технологий [99, 100, 101].

А.И. Башмаковым и И.А. Башмаковым в [6] обосновывается, что автоматизация бизнес-процессов образования и применение ПСУН на различных этапах обучения способствует повышению эффективности, а также стимулирует развитие существующих и позволяет формировать новые формы, средства и способы обучения. Кроме того, вводится классификация компьютерных средств обучения. Как и

многие другие классификации, введённые до 2010 года, она не отражает изменений в технологическом базисе и методологиях коллаборативного и смешанного обучения.

Для ПСУН и, в особенности, РИСУН, разрабатываемых в современных условиях, требуется учитывать высокую адаптивность [102], коллаборативность [103], развитие открытого образования [80], МООС-платформ [104] и т.п.

1.2 Бизнес-процессы высших учебных заведений

КИМРА автоматизирует процессы, связанные с ИвО и интегрирована в информационное пространство высшего учебного заведения (вуза). В связи с этим необходимо обозначить место КИМРА в рамках процессов высшего учебного заведения. Воспользуемся классификацией процессов, разработанной в рамках проекта *HEI-UP Business Process Management in Higher Education Institutions* [105]. В ней выделены такие группы процессов верхнего уровня как:

- «Преподавание» (*HEI-KPA1: Teaching processes*);
- «Учёба» (*HEI-KPA2: Studying processes*);
- «Исследование» (*HEI-KPA3: Research processes*);
- «Разработка» (*HEI-KPA4: Development processes*);
- «Управление и сопровождение» (*HEI-KPA5: Management and supporting processes*)

Рисунок 1.2 содержит фрагмент переведённой на русский язык классификации процессов. Система КИМРА не касается групп *HEI-KPA* 3, 4, 5 и затрагивает только участников процессов из групп «Преподавание» и «Учёба», то есть студентов и профессорско-преподавательский состав. Основными процессами со стороны преподавателя являются: «Оценивание дисциплины» (*HEI-PG-1.4 Course evaluation*), «Руководство КР и ВКР и экзамены» (*HEI-PG-1.5. Thesis supervision and examination procedure*). Со стороны студента: «Учёба» (*HEI-PG-2.4. Studying*), «Экзамены» (*HEI-PG-2.6. Examination*).

Поскольку выделенные процессы не предполагают детализации, для данной работы имеет смысл выделить связанные с ними подпроцессы, имеющие отношение к измерениям в образовании.

За основу возьмём этапы разработки теста, представленные в рекомендациях американской ассоциации психологов (*American Psychological Association, APA*). Согласно [13] основные этапы разработки теста состоят из создания банка заданий, формирования теста, доставки теста, получения данных о прохождении теста, анализа результатов теста и могут быть представлены в виде циклического графика (рисунок 1.1).

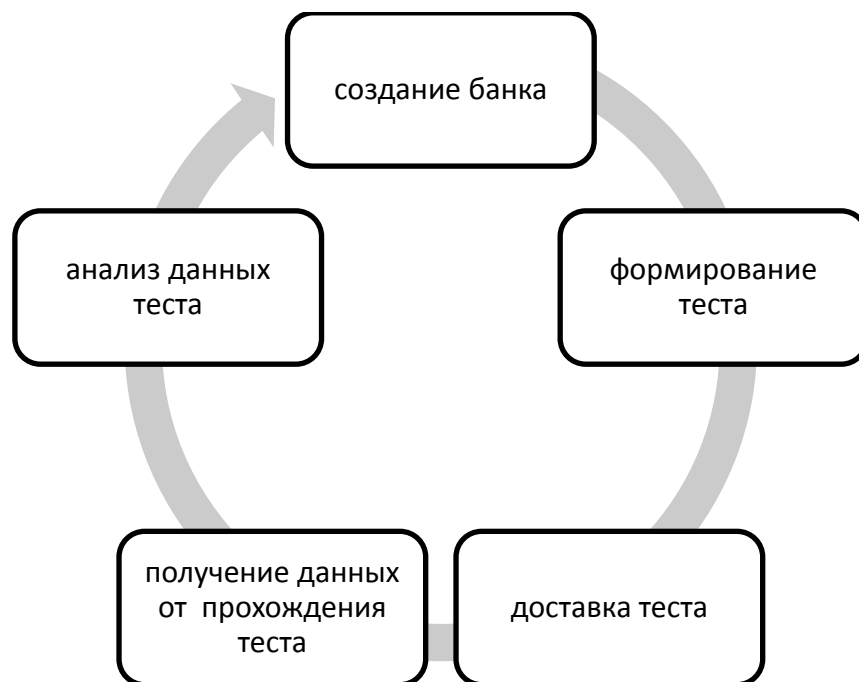


Рисунок 1.1. Основные этапы разработки теста

Приведённые этапы позволяют говорить о подпроцессах контроля знаний только на уровне теста в целом, не учитывая тестовые задания и их жизненный цикл. Разработка компьютерной системы, позволяющей работать на разных этапах жизненного цикла отдельных КИМ, требует детализации следующих процессов:

- до добавления в банк: разработка КИМ, проверка (экспертиза) КИМ, способ добавления КИМ в банк;
- для формирования композитного КИМ: правила формирования композитного КИМ и селекция форматов доставки КИМ;

- прохождение контрольно-измерительного мероприятия: способ накопление данных о прохождении и результатах КИМ, предоставление обратной связи;
- после анализа данных: правила обновления характеристик КИМ на основе анализа данных, полученных от измерений, актуализация КИМ согласно обновлённым характеристикам.

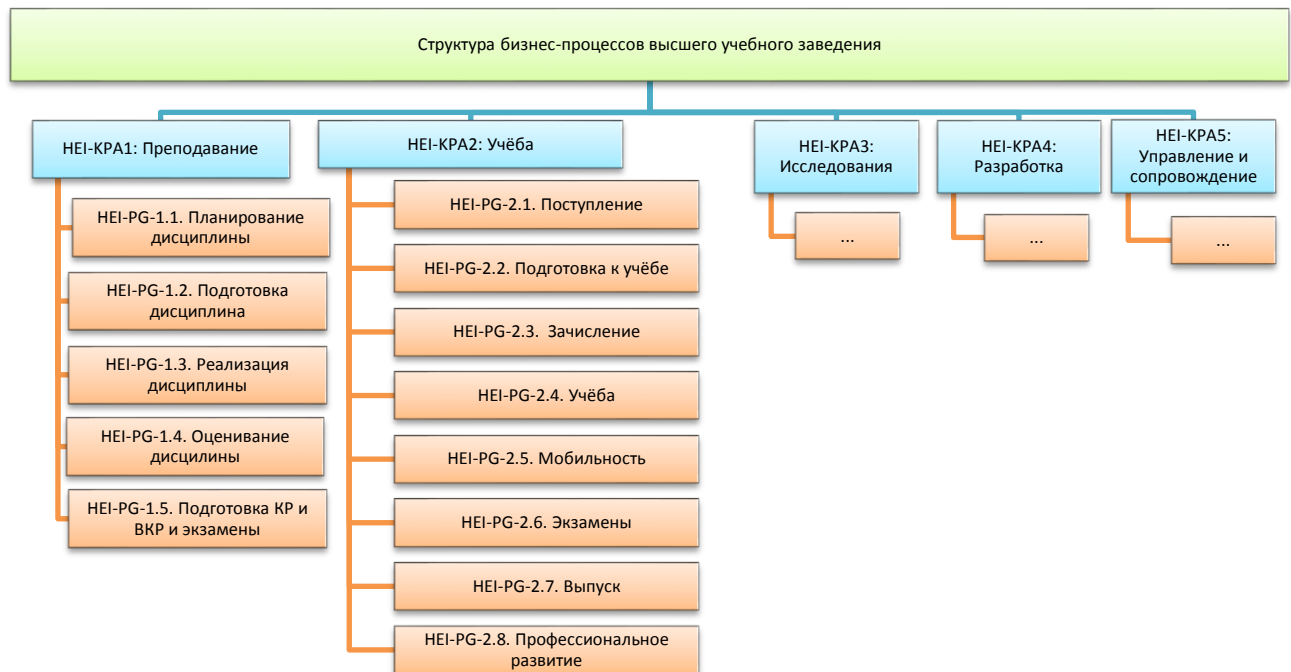


Рисунок 1.2. Структура бизнес-процессов высшего учебного заведения

Разрабатываемая РИСУН КИМРА нацелена на поддержание целостности процессов измерения в образовании и полностью согласован с ЖЦ КИМ, то есть поддерживает все его процессы.

1.3 Классификация тестовых заданий

В данном разделе мы вернёмся к самой распространённой форме компьютерного контроля знаний: тестированию. Классификации тестовых заданий в связи с необходимостью их автоматизации достаточно подробно проработаны в литературе, сформулированы их существенные особенности в связи с компьютерной формой представления. Заметим, однако, что эти классификации зачастую предложены вне контекста разработки ПСУН, что существенно усложняет их реализации ввиду неочевидности отношений между классами заданий.

1.3.1 Традиционная классификация тестовых заданий

Назовём традиционной [*traditional taxonomy*] классификацию тестовых заданий, в которой деление тестовых заданий на типы зависит от способа внесения студентом ответа в бланк варианта теста. Иными словами, к традиционным тестовым заданиям относятся те виды заданий, для выполнения которых не требуется возможностей, предоставляемых компьютером, то есть тестовое задание может быть включено в вариант теста, напечатанного на бумаге – бланкового теста [*paper-and-pencil test*].

В русскоязычной литературе [106, 107] задания традиционной таксономии подразделяют на две группы: задания с закрытым и задания с открытым ответом.

Задания с закрытым ответом:

- задания с двумя ответами («да/нет», «истина/ложь») [*true/false question*]. Предполагается выбор одного из двух альтернативных вариантов ответа;
- задание с одним верным ответом [*multiple-choice question, MCQ*]. Предполагается выбор одного из нескольких альтернативных вариантов ответа;
- задание с несколькими верными ответами [*multiple-response question, MRQ*]. Предполагается выбор нескольких верных вариантов ответа из предложенного списка;
- задание на установление соответствия [*matching question*]. Предполагается установление соответствия элементов из одного списка с элементами другого;
- задание на восстановление верной последовательности [*sequencing question*]. Предполагается расположение объектов (текста, рисунков, цифр и проч.) в верной последовательности.

Перечисленные задания с закрытым ответом иногда уточняются такими типами тестовых заданий, как задание с тремя ответами (вариант задания с одним верным ответом, когда выбор предлагается из трёх альтернатив) [106]; версией задания с несколькими верными ответами [108], представляющей по существу набор

заданий с двумя верными ответами, то есть когда в задании предполагается установление (ложности) нескольких высказываний и проч. Эти и другие подобные задания мы оставим без внимания, как допускающие конструирование путём комбинации предложенных пяти видов задания.

Задания с открытым ответом:

- задания с кратким ответом [*short answer question, SAQ*]. Предполагают краткий ответ, который необходимо внести в бланк. Например, слово, словосочетание, формулу, цифру и т.п.;
- задания с развёрнутым (свободно-конструируемым) ответом [*essay question*]. Предполагают занесение в бланк развёрнутого ответа на задание, например, в виде текста, схемы, диаграммы и т.п.

В некоторых источниках дополнительно выделяют задания с кратким ответом на дополнение [*completion*] [109] (текст требуется вписать в места пропусков); с развёрнутым структурированным ответом [*structured essay question*], предполагающие ограничения структуры ответа на задания, и многие другие модификации указанных двух типов заданий.

1.3.2 Компьютерно-ориентированная классификация тестовых заданий

В данной работе предполагается опираться на классификацию, отличную от традиционной. Будем учитывать специфику компьютеризированного тестирования и дополнительные возможности, которые появляются у разработчиков и организаторов тестов. Динамичное развитие области компьютеризированного тестирования в 90-е годы XX века привело к появлению некоторых дополнительных классификаций тестовых заданий. В книге Ван дер Лидена и Гласа [43] тестовые задания, реализация которых непременно требует проведения тестирования в компьютерной форме, названы инновационными тестовыми заданиями (ИТЗ) [*innovative items*]. Глава, посвящённая таксономии ИТЗ, написана авторами в области проведения и анализа результатов SAT во главе с С. Паршел.

Отметим, что в указанной выше книге авторы во многом обобщают опыт предшествующего десятилетия. Так, ещё 1993 году Д. Кох в статье [110] представил классификацию ИТЗ в виде четырёхуровневой иерархии:

- задания, входящие в традиционную классификацию;
- задания, позволяющие более полно использовать графику и графические возможности;
- «многомерные» задания (требуют визуализации или управления информацией);
- ситуативные задания, тесно связанные с контекстом предметной области.

Позже, в 1996 в работе [111] С. Паршел, Р. Стюарт и Д. Риттер ввели понятие измерительного «континуума», классификация видов тестовых заданий в котором приобрела непрерывную форму. Типы тестовых заданий, таким образом, изменялись от жёстко ограниченных (задания с множественным выбором) до крайне свободных (написание эссе).

В работе [112], представленной в 1998 году на Коллоквиуме по компьютерному тестированию в Филадельфии, обсуждались «комплексные» тестовые задания. Согласно этой статье комплексность тестового задания является результатом множества обстоятельств, начиная с природы тестовых заданий и заканчивая допустимыми для студента вариантами действия в рамках данного задания.

В седьмой главе книги [43] С. Паршел, Т. Деви и П. Пашлей приводят детальную таксономию инновационных тестовых заданий. Разграничение заданий производится по следующим пяти компонентам:

- формат тестового задания. Под форматом понимается способ получения ответа от студента. В русскоязычной литературе, касающейся традиционных тестовых заданий, формат обычно называют типом;
- способ задания ответа – подразумевает средства, необходимые для получения ответа от студента, например, клавиатуру, мышь, экран планшета и пр.;
- включение мультимедиа – предполагается ли применение таких элементов, как звук или видео в задании;

- уровень интерактивности – подразумевает наличие реакции задания на ответ или действия студента;
- алгоритм оценивания – предполагает способ преобразования ответа студента в количественную оценку.

Согласно [19] введение в таксономию нового типа тестового задания требует следующих обязательных действий:

- описание структуры задания – то, как составлено задание, какие элементы включает и пр.;
- описание структуры ответа – как выглядят ответы на задание (список, поле ввода, активное изображение и пр.);
- разработка способа построения ответа – в какой форме студент задаёт ответ;
- определение способов и правил оценивания задания – в самом простом случае: в каком случае задание зачтено, а в каком нет;
- проверка адекватности, т.е. соответствия задания теме, предполагаемому уровню обученности и т.п.

Как в традиционной, так и в новой классификации тестовые задания в первую очередь отличаются друг от друга способом построения ответа испытуемым (маркирование элементов списка, вписывание или впечатывание текста в специально выделенное место, активация областей изображения и проч.).

Отметим также, что для проектирования и разработки ПСУН таксономия тестовых заданий должна быть такой, чтобы были очевидны отношения между тестовыми заданиями разных типов и каждое тестовое задание определённого типа однозначно отличалось от прочих заданий. Это, в свою очередь, означает, что для проектирования программных средств поддержки процессов педагогического тестирования существенными являются следующие вопросы:

- форма отображения задания и ответов;
- способ конструирования ответа на задание.

1.3.3 Классификация тестовых заданий, применённая в КИМРА

С учётом выделенных требований на основе имеющихся таксономий для КИМРА предложена следующая классификация тестовых заданий:

1. Альтернативы. Стандартное представление – радиокнопки [*radio buttons*].
 - 1.1.      Альтернатива [*choice; alternative*] – выбор из нескольких взаимоисключающих вариантов.
 - 1.2.    Бинарная альтернатива [*binary choice*] (сужение «Альтернативы») – выбор из двух взаимоисключающих вариантов.
2. Опции. Стандартное представление – чекбоксы [*check boxes*].
 - 2.1.      Опции (набор опций) [*set of options*] – выбор от 0 до N опций.
 - 2.2.    Опция [*option*] – выбор от 0 до 1 опции.
3. Отображения. Стандартное представление – сортируемый список [*list with sorting*].
 - 3.1.      Сопоставление [*matching*] – установление соответствия между элементами двух множеств.
 - 3.2.      Упорядочение [*ranking*] – установление соответствия между элементами двух множеств, из которых одно упорядочено.
4. Простые редакторы. Стандартное представление – поле ввода текста [*text area*].
 - 4.1.   Короткий текст [*short text*] – ввод строки символов. Стандартное представление – редактор однострочного текста [*text editor*].
 - 4.2.   Число [*number*] – ввод единственного числа. Стандартное представление – редактор целого или вещественного числа [*integer or float number editor*].
 - 4.3.   Формула [*formula; mathematical expression*] – ввод математической формулы. Стандартное представление – редактор математической формулы [*mathematical formula editor*].
 - 4.4.   Эссе [*essay*] – ввод многострочного текста, возможно – форматированного. Стандартное представление – редактор многострочного текста [*memo*].

5. Комплексные редакторы. Стандартного представления нет – используется специализированный конструктор ответа [*answer designer*].

5.1.  Заполнение пропусков [*filling gaps*] – ввод нескольких отдельных строк символов в поля ввода в виде пропусков в форматированном тексте.

5.2.  Комплексный ответ (конструктор ответа) [*Complex designer; Complex constructor*] – произвольная интерактивная форма ввода артефакта некоторого типа. Артефактом может быть исходный код на конкретном языке программирования, диаграмма и др.

1.4 Форматы представления тестовых заданий

Проблема представления тестовых заданий в форме, повышения удобства импорта/экспорта тестов и тестовых заданий в системы тестового контроля знаний не нова. Известны и используются различные форматы представления тестовых заданий (таблица 1.1), достаточно подробная их подборка представлена на сайте [113].

Таблица 1.1 – Некоторые форматы представления тестовых заданий

Название формата	Краткое описание	Примеры поддерживающего ПО	Ссылка
<i>AIEKEN</i>	Текстовый формат <i>импорта</i> тестовых заданий альтернатив	<i>LMS Moodle, LMS E-Front</i>	https://docs.moodle.org/29/en/Aiken_format [114]
<i>GIFT</i>	Текстовый UTF-8 формат <i>импорта</i> тестовых заданий, поддерживающий представление альтернатив, опций, отображений, а также простых редакторов (короткий текст и число)	<i>LMS Moodle, LMS E-Front</i>	https://docs.moodle.org/29/en/GIFT_format [115]
<i>MoodleXML</i>	XML-представление для <i>экспорта/импорта</i> тестовых заданий. Формат поддерживает те же виды тестовых заданий, что и GIFT, но допускает импорт изображений	<i>LMS Moodle Quiz module</i>	https://docs.moodle.org/29/en/Moodle_XML_format [116]

<i>Blackboard</i>	Текстовый формат <i>импорта</i> тестовых заданий и элементов опросников. Поддерживает тестовые задания: альтернативы, опции, отображения, простые редакторы (короткий текст и число), а также несколько заданий комбинированного и специального вида, например, шкала Ликерта.	<i>Blackboard, LMS Moodle</i>	https://help.blackboard.com/en-us/Learn/9.1_SP_10_and_SP_11/Instructor/070_Tests_Surveys_Pools/106_Upload- ing_Questions#question_format [117]
<i>SCORM/AICC</i>	XML-формат <i>импорта/экспорта</i> электронных учебных курсов, включающий среди прочего, представление некоторых тестовых заданий: альтернатив, опций и простых редакторов (короткий текст).	<i>LMS Moodle, Blackboard, IMS Content package in Cecil 7, Share-Point LMS, Scorm assessment engine</i>	https://docs.google.com/document/d/1zzLk8sp3MHsnLCNeAJbqRcMv-DIg8Y4J0oRAUro5gbk/edit [118]
<i>WebCT</i>	Текстовый формат <i>импорта</i> тестовых заданий: альтернатив, опций и простых редакторов (короткий текст, число, эссе). Формат предусматривает включение рисунков в текст заданий. Для короткого текста возможно использование нескольких пропусков.	<i>Blackboard, LMS Moodle</i>	https://docs.moodle.org/29/en/Import_questions#WebCT_format [119]
<i>QTI/APIP</i>	Спецификация формата <i>импорта/экспорта</i> тестовыми заданиями, позволяет описать любое тестовое задание или задание опросника, в том числе задания, содержащие графику, мультимедиа, а также задания для адаптивного тестирования.	<i>Respondus 4.0, TAO v.2.6.5, ONYX Test-suite v5.3.1</i>	http://www.imsglobal.org/question/ [120]

Отметим, что большинство из них является частными форматами для отдельных систем, нацеленными на возможность разработки тестовых заданий преподавателями в отдельных файлах и их импорта в тестовые системы, например, *MoodleXML*, *WebCT*, *Blackboard* и проч. Отдельные форматы получили широкое распространение. Например, исходно разработанные для опросников [quiz] *LMS Moodle* форматы, которые поддерживаются и другими *LMS* и тестовыми системами.

Форматы экспорта/импорта учебных курсов, включающие в себя представления для тестовых заданий и других элементов контроля знаний, не нацелены на гибкое представление именно тестовых заданий. Например, в *SCORM* допустимо

представление отдельных (не всех) видов тестовых заданий, входящих в традиционную классификацию (таблица 1.1).

Наиболее системный подход к решению проблемы совместимости форматов тестовых заданий был предложен консорциумом *IMS Global learning* [121]. Специалистами *IMS* к 2012 году была разработана спецификация *IMS Question and Test interoperability specification (QTI v2.1)* [120], регламентирующая совместимость тестов и тестовых заданий для банков и тестовых систем, разработанных участниками альянса *QTI/APIP*. Основанный на *QTI v2.1* протокол *Accessible portable item protocol (APIP v1.0)* [122] представляет собой модель данных, стандартизирующую формат файлов для импорта/экспорта тестовых заданий.

Несмотря на проработанность и наличие полноценного методического сопровождения, *QTI/APIP* совместимость пока не стала повсеместной практикой – так, большинство продуктов обеспечивает поддержку стандарта *Learning tools interoperability (LTI v1.0 и LTI v1.1)*. Среди более ста разработчиков образовательного программного обеспечения, представленных на сайте *IMS*, *QTI/APIP* совместимость в своих продуктах обеспечивают не более десяти.

Изучив классификации тестовых заданий, сложившийся рынок программных средств поддержки тестирования и актуальные требования к автоматизации процессов поддержки жизненного цикла тестовых заданий автор разработал оригинальный формат представления артефактов, фигурирующих в техническом задании.

Для внутреннего представления тестовых заданий системы КИМРА используется специальный *JSON*-основанный формат. Тестовые задания указанного формата преобразуются к наиболее общему из рассмотренных распространённых форматов экспорта/импорта – *QTI/APIP*. Отметим, что он поддерживается последними версиями модулей контроля распространённых систем управления учебным процессом [*learning management systems, LMSs*] и средств контроля знаний, например, *Respondus, ONYX, Canvas*.

1.5 Современные средства автоматизации измерений в образовании

В данном разделе предложен обзор автоматических средств контроля знаний. Рассмотрены как комплексы программ, ориентированные только на поддержку процессов тестирования в образовании, так и модули комплексов, ориентированных на поддержку традиционного (очного) процесса обучения, дистанционного обучения, массовых открытых онлайн курсов и прочие.

В литературе [123, 13, 124, 125, 57, 58] описаны требования к банкам тестовых заданий (БТЗ). Учитывая их, а также необходимость версионирования, использование банка для хранения разнородных КИМ и необходимость интеграции с ПСУН сформулируем требования к хранилищу КИМ, которое может быть использовано в том числе как БТЗ. В первую очередь для этого будут проанализированы ПСУН, ориентированные на поддержку процессов тестирования в образовании и модули комплексов, поддерживающие традиционный (очный) процесс обучения, дистанционное обучение, массовые открытые онлайн курсы и прочие.

Критерии включения/исключение ПСУН в обзор. Поскольку доминирующее большинство средств контроля знаний являются сетевыми, то в обзор включены сетевые ПСУН или модули потенциально пригодные для поддержки БТЗ, исключены полностью проприетарные программные средства, а также системы поддержки жизненного цикла тестовых заданий, используемые в закрытых масштабных тестированиях регионального и мирового уровней. Модули и ПСУН (БТСИМ, Moodle Quiz Module, Respondus, ТАО, Learnosity, СДО «Прометей», OpenEdx) оценены по 21 критерию.

Критерии сравнения ПСУН: 1.1. Поддержка традиционной таксономии ТЗ; 1.2. Поддержка компьютерно-ориентированной таксономии ТЗ; 1.3. Поддержка пользовательских типов ТЗ; 2. Возможность поддержки БТЗ; 3. Поддержка первичных баллов тестирования; 4. Наличие возможностей анализа результатов тестирования по одной из теорий тестирования; 5. Экспорт результатов в форматах CSV, JSON, XML; 6. Экспорт результатов в других форматах; 7. Доступность по сети Интернет; 8. Наличие режима защищённого тестирования; 9. Поддержка версиониро-

вания БТЗ; 10. Наличие *API*; 11. Поддержка частных (собственных) форматов импорта/экспорта ТЗ; 12. Поддержка *SCORM* импорта/экспорта; 13. Поддержка *RDF* импорта/экспорта; 14. Поддержка *QTI/APIP* импорта/экспорта; 15. Проприетарность; 16. Наличие мобильного клиента или возможности интеграции со сторонним мобильным клиентом; 17.1. Поддерживается **nix* ОС; 17.2. Поддерживается ОС *Windows*; 17.3. Поддерживается *MacOS*; 18. Не требует обязательного использования стороннего платного программного обеспечения.

Поскольку доминирующее большинство средств контроля знаний являются сетевыми, то в предлагаемом обзоре программные средства сгруппированы по наличию в них толстого и/или тонкого клиента.

1.5.1 Системы и комплексы, включающие толстый клиент

1.5.1.1 Комплекс «Банкир-Тестер-СтатИнфо-Маркер»

Популярный среди российских университетов комплекс, разработанный В.И. Нардюжевым и И.В. Нардюжевым «Банкир-Тестер-СтатИнфо-Маркер» (БТСИМ), состоит из отдельных программ, поддерживающих основные процессы тестирования в образовании [48].

Программа «Банкир» предназначена на накопления БТЗ и генерации теста с заданными характеристиками (времени, количества заданий, параметры переключения заданий в процессе тестирования). В БТЗ могут быть сохранены задания видов: альтернативы, опции и простой редактор (число и короткий текст). Экспорт теста возможен только в формате программы проведения тестирования «Тестер». Выгрузка заданий в тест организована в зашифрованном виде.

Программа «Тестер» позволяет проводить тестирование в двух режимах: локальном и сетевом. Автоматически осуществляется контроль времени тестирования, генерируется протокол тестирования для каждого участника. В результате прохождения тестирования каждый участник может получить интегральную информацию по тесту, формирующая обратная связь по тесту не предусмотрена. Просмотреть протокол с предъявлением верных и неверных ответов по каждому заданию позволяет программа «Маркер».

Существенным плюсом программы «Тестер» является ограничения пользовательского доступа к прочим программам и ресурсам компьютера, что существенно снижает плагиат. Отметим, однако, что поскольку первые версии БТСИМ появились ещё на заре тестирования в России, в 1999 году, то в настоящее время в сети Интернет свободно распространяются средства несанкционированного доступа к защищённой информации БТСИМ.

Поскольку авторами БТСИМ банк трактуется «как набор заданий, каждое из которых имеет минимум один вариант» [48], следует отметить, что программа «Банкир» не является полноценным средством поддержки БТЗ. Это объясняется тем, что БТЗ помимо тестовых заданий позволяет хранить характеристики или связи с характеристиками тестовых заданий, версии и данные о состояниях тестовых заданий и прочую организационно и технически значимую информацию. В комплекс БТСИМ архитектурно не заложена возможность интеграции данных между «Банкиром» и «СтатИнфо» – программой «оперативного статистического анализа результатов компьютерного тестирования» [48].

В свою очередь программа «СтатИнфо» применяется для расшифровки протоколов тестирования, сгенерированных программой «Тестер» и предоставляет возможность генерации статистических отчётов общего вида: количество тестируемых, распределение тестируемых по оценкам используемой шкалы, дисциплинам и проч. Среди функций «СтатИнфо» отсутствуют средства автоматизации методов обработки результатов тестирования в образовании, начиная с *СТТ*. Также не реализованы модели *IRT* и *TRT*, нацеленные на обработку результатов компьютерного тестирования. «СтатИнфо» предоставляет возможность экспорта результатов тестирований в виде текста с разделителями. Анализ полученной выгрузки возможен только с привлечением сторонних средств.

Ограничения БТСИМ состоят в том, что он:

- имеет существенные ограничения по видам допустимых тестовых заданий, не поддерживает полностью даже задания традиционной классификации;

- на уровне БТЗ не поддерживает версионирования заданий и хранения существенных для организации тестирования характеристик, полученных в итоге анализа результатов;
- не автоматизирует процессов управления БТЗ;
- не поддерживает статистического анализа результатов тестирования методами *CTT*, *IRT* и *TRT*;
- оперирует только собственным форматом экспорта тестовых заданий и не поддерживает ни одного известного формата экспорта/импорта;
- функционирует только на компьютерах под управлением ОС *Windows*.

1.5.2 Системы и комплексы, включающие тонкий клиент

1.5.2.1 Подсистема поддержки опросов *Moodle Quiz Module*

Подсистема поддержки опросов *Quiz Module LMS Moodle* позволяет преподавателю разрабатывать тестовые задания, сохранять, и объединять задания в опросники.

На уровне преподавателя управление банком вопросов в рамках дисциплины организовано при помощи *Question Bank* [126]. *Moodle Quiz* предоставляет возможности по организации совместного использования вопросов в нескольких дисциплинах. Эта функциональность доступна на уровне администратора банка вопросов.

Как и в БТСИМ БТЗ в *Quiz Module* не является БТЗ, а представляет собой неверсионированное хранилище вопросов для составления опросников.

Поддерживаются тестовые задания: альтернативы, опции, отображения, простые конструкторы (число и короткий текст), эссе, а также их модификации. Предусмотренная архитектурой *Moodle* возможность подключения плагинов позволяет создавать интерактивные версии традиционных тестовых заданий, например, drag and drop отображения, тестовые задания с простыми конструкторами, например, для ввода математических или химических формул, а также задания с комплексными конструкторами ответа [127].

Экспорт тестовых заданий возможен в форматах *GIFT*, *MoodleXML* и *XHTML*. Поддерживается импорт заданий, представленных в форматах *GIFT*, *MoodleXML*, *AIKEN*, а также в форматах других программ, например, *WebCT*, *Blackboard*, *Diploma 6*, полный обновляемый список доступен на сайте документации *Moodle* [128].

Поскольку *Quiz Module* не является специализированным средством для проведения тестирований в образовании и не гарантирует контролируемой среды для участников тестирования, его использование для нужд тестирования возможно при использовании *Safe Exam Browser*, в архитектуру которого заложено приложение, переводящее компьютер в режим ограниченного доступа («режим информационного киоска»), когда запуск сторонних приложений возможен только этим приложением [129].

Вопросник может быть настроен таким образом, чтобы по итогам прохождения тестирования участникам помимо набранных баллов предоставлялась обратная связь по содержанию теста (верно и неверно отвеченные вопросы, комментарии к ответам).

Для первичной статистической обработки результатов компьютерного тестирования, полученных средствами *Quiz Module*, используются методы *СТТ* [130]. Отметим, что не производится дополнительных проверок согласия распределения первичных баллов с нормальным, поэтому результаты, полученные по итогам нельзя считать безоговорочно валидными.

1.5.2.2 Система управления контролем знаний *Respondus*

Система управления контроля знаний *Respondus 4.0* [131] предоставляет инструменты по созданию и управлению контрольными мероприятиями в компьютерной форме. Подготовленные задания могут быть представлены как в бумажном виде, так и с помощью *LMS*, например, *Blackboard*, *Moodle*, *Canvas* и др.

Respondus 4.0 позволяет работать с тестовыми заданиями, полностью отвечающими традиционной классификации. Экспорт тестовых заданий доступен в част-

ные форматы популярных *LMS* и формат *IMS QTI*. Импортировать задания допустимо из документов *MS Word* (включая изображения), текстовых редакторов, форматов *CSV* и *IMS QTI*.

Без сомнений отличительной особенностью и неоспоримым преимуществом является возможность использования совместно с *Respondus 4.0* банков тестов, доступных через сервис *Respondus Test Bank Network*. Эти банки конструируются ведущими издательствами на основе учебников для высшего образования. Преподаватели, использующие соответствующие книги в учебных курсах могут получить бесплатный доступ к этому сервису.

Тесты и тестовые задания, представленные в банках *Respondus Network*, являются фиксированными, не содержат дополнительных характеристик, необходимых для представления тестовых заданий в БТЗ. Поэтому в случае *Respondus Test Bank Network* мы имеем дело с ещё одним неверсионированным хранилищем тестовых заданий.

Браузер *Respondus LockDown Browser* представляет собой средство обеспечения контролируемости среды при проведении тестирования. На время тестирования браузер блокирует ресурсы компьютера, делая невозможным изменение размеров окна, переключение между программами, вызовы контекстных меню и т.д. Кроме того *Respondus LockDown Browser* интегрирован с *Respondus Monitor* – дополнительной программой, связывающей браузер с веб-камерой.

Программы и сервисы *Respondus* не предоставляют инструментов статистического анализа результатов тестирования. Выгрузки результатов осуществляются в файлы формата *CSV* для использования в специализированных программных средствах или подсистемах конкретных *LMS*.

Respondus 4.0 является платным программным обеспечением, что является существенным препятствием в использовании его и дополнительных программ и сервисов отдельными преподавателями. Кроме того, *Respondus 4.0* функционирует только на компьютерах под управлением ОС *MS Windows (Vista, 7 и выше)*.

1.5.2.3 TAO

Система с открытым кодом TAO [132] предназначена для поддержки процессов компьютерного тестирования.

Ввиду отсутствия привязки к бланковым тестам, TAO позволяет создавать и хранить задания более чем 17 интегрированных типов, включая задания традиционной классификации, задания с мультимедиа содержимым, а также любые пользовательские задания. Специальный редактор для подготовки пользовательских заданий полностью отвечает нуждам подготовки тестовых заданий, то есть кроме самого задания позволяет создавать правила (схему) его оценивания.

Хранилище заданий TAO не является версионированным, а также не предполагает сохранение характеристик задания, соответствующих хотя бы одной из наиболее распространённых теорий тестирования. Это не позволяет использовать TAO как полноценное средство поддержки и управления БТЗ.

Поскольку TAO сертифицирована на QTI по нескольким пунктам, в том числе и по поддержки форматов импорта/экспорта тестовых заданий, то TAO-тесты могут быть импортированы/экспортированы в QTI формат. Экспорт также возможен в формат *Resource Description Framework (RDF)* [133].

Несмотря на позиционирование TAO как системы поддержки компьютерных тестирований, предлагаемые средства весьма ограничены в обеспечении контролируемости среды. Предполагается автоматический контроль дат и времени прохождения теста, ресурсы компьютера не блокируются и остаются доступными при прохождении теста. Таким образом, подсистема тестирования TAO эффективно может быть использована только для организации опросников или подготовительных тестирований.

Результаты тестирований TAO не могут быть обработаны средствами этой системы. Выгрузки результатов по каждому тестированию осуществляются в файлы формата CSV и могут быть обработаны специализированными программами.

1.5.2.4 *Learnosity*

Набор *API* для поддержки процессов организации и проведения компьютерных контрольных мероприятий *Learnosity* [134].

Редактор создания тестовых заданий может быть интегрирован с *CMS* разработчика заданий, чтобы создавать задания в знакомом редакторе (рисунок 1.3). Допустимы тестовые задания традиционной классификации и несколько видов компьютерноориентированных заданий с поддержкой мультимедиа и различными конструкторами ответа: формулами, графиками, звуковым рядом и проч. Практически каждое тестовое задание в контексте *Learnosity* состоит из постановки задачи, типа ответа и правила оценивания.

Разработчики *Learnosity* предлагают два варианта работы с БТЗ: использование пользовательского БТЗ или поддержка *Learnosity Item Bank*. Важным преимуществом *Learnosity Item Bank* является возможность тегирования заданий. Отметим, что поскольку метки тегов могут выбираться произвольно, то сохраняемые тестовые задания могут быть связаны и со своими тестологическими характеристиками. Однако при этом *Learnosity Item Bank* является неверсионированным хранилищем.

Поддерживается адаптивное тестирование, которое может быть основано как на связях между заданиями, так и на связях между тестлетами. Проведение тестирования обеспечивается интерфейсом *Assess API*. Помимо временного контроля по тесту или каждому заданию *Assess API* позволяет в современных браузерах обеспечить режим полного экрана для повышения контролируемости тестирования. После завершения тестирования участниками предоставляются различные отчёты по каждому заданию, группам заданий и тесту в целом, предусмотрена возможность предоставления обратной связи.

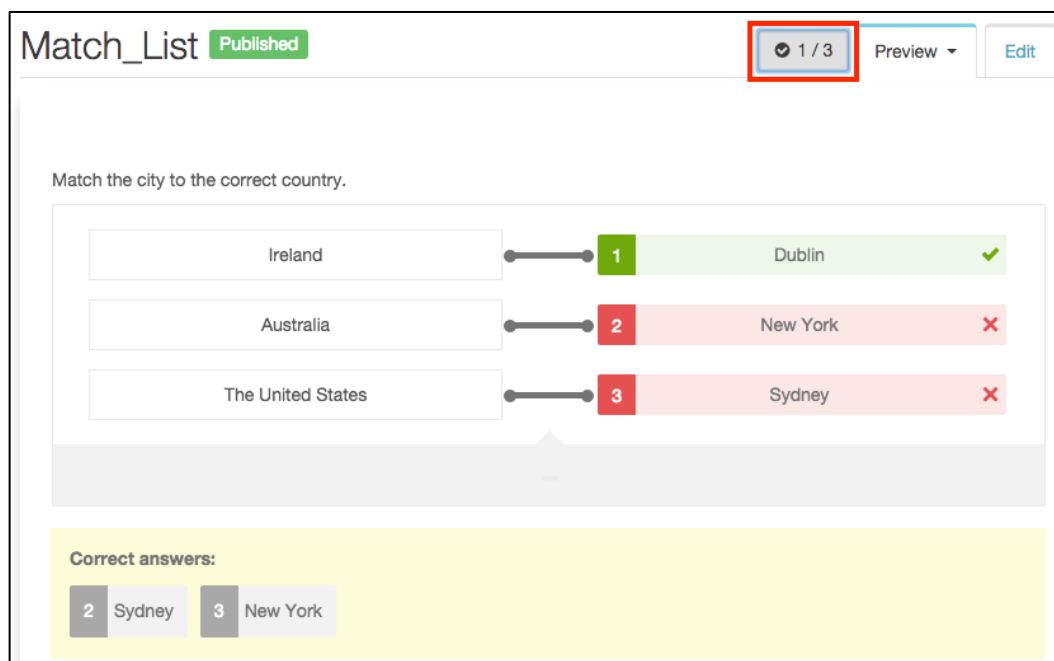


Рисунок 1.3. Пример проверки задания типа «Отображение» в *Learnosity*

Learnosity не имеет *API* анализа результатов педагогического тестирования, но допускает экспорт сырых данных в форматах *JSON* и *CSV*.

На данный момент ощутимым недостатком *API*, предоставляемых *Learnosity* является незавершённость функционала, а также несовместимость со стандартными форматами импорта/экспорта заданий. *QTI/APIP* совместимость заявлена как сервис, но к настоящему времени не опубликована.

1.5.2.5 Система дистанционного обучения «Прометей»

Популярная в российских университетах система дистанционного обучения «Прометей» [135] позиционируется как ПО, позволяющее поддерживать процессы виртуального университета. Контроль знаний реализован в формате тестирования.

Для участников тестирования доступны два вида тестирований: самопроверка (с не лимитированным количеством повторений) и экзамена (однократный). Сводные результаты тестирования доступны сразу после прохождения теста, подробные результаты формируются в отчёте по тестированию. В режиме самопроверки пользователю к заданиям может быть сформирована обратная связь.

В официальном описании решения для вузов и учебных центров СДО «Прометей» на сайте [135] автору не удалось обнаружить информации по типам допустимых тестовых заданий, возможностях проведения анализа результатов и проч.

Разработчики «Прометей» указывают на *IMS* и *SCORM* совместимость без указания версий и подтипов соответствующих стандартов.

Прометей 4.5 является платным программным обеспечением, что является существенным препятствием в использовании его и дополнительных программ и сервисов отдельными преподавателями. Кроме того, Прометей 4.5 серверная часть функционирует только на компьютерах под управлением ОС *MS Windows* не выше 7 версии и требует установленного *Microsoft Windows Server* и *Microsoft SQL Server* версий 2008 или 2012. Указанные сервера также являются проприетарным ПО, недоступным большинству преподавателей. Более того, они требуют специальных навыков при установке и настройке.

1.5.2.6 Платформа массовых открытых онлайн курсов *OpenEdX*

Система поддержки массовых открытых онлайн курсов *OpenEdX* [136] разработана специалистами Массачусетского технологического университета (*Massachusetts Institute of Technology, MIT*) и Гарвардского университета (*Harvard University*). В настоящее время платформа является свободно-распространяемой и имеет открытый программный код [*open source*].

Наименьшей единицей дробления учебного курса, реализованного на платформе *EdX*, является учебный блок [*unit*], который, в частности, может представлять собой контрольное задание или упражнение.

Компонент «Задания» [*Problems*] системы *OpenEdX* предоставляет возможность использования традиционных заданий в тестовой форме. В режиме расширенного [*advanced*] редактора тестовых заданий доступен пакет тестовых заданий, входящих в компьютерно-ориентированную классификацию [137]. Средства, предоставленные разработчикам *OpenEdX* курсов, предусматривают различные конструкторы ответов, например, для химических и математических формул, электрических цепей. В части оценивания расширенный редактор *OpenEdX* допускает создание пользовательской схемы оценивания задания, например, с помощью пользовательского задания с собственной оценкой на *Javascript* [*custom javascript problem*].

Наличие большого количества узкоспециализированных заданий, например, требующих рисования электрической цепи в ответе связано с общей негибкостью стандартных средств *OpenEdX* в области написания самостоятельного оценивания для заданий. Загрузка специальных средств, необходимых для отдельных тестовых заданий, например, редактора/подсветчика программного кода, осложняется необходимостью загрузки на сервер одиночных файлов и невозможностью создания структуры папок. Решение данной проблемы лежит в привлечении внешних по отношению к *OpenEdX* веб-сервисов и ресурсов, что существенно усложняет работу преподавателя при создании заданий.

Задания *OpenEdX* не экспортируются ни в одном из распространённых форматов, используется собственный формат *edXML*. Разработчики планируют описание собственного *EdX API*, а также обеспечение *QTI/APIP* совместимости.

Создание полноценного теста стандартными средствами оценивания платформы *OpenEdX* невозможно. Каждое тестовое задание представляется отдельно, без привязки к остальным заданиям. Тем не менее, допустимо, например, использование заданий, оценка за которые проставляется *JavaScript* или *Python*-скриптом, для представления теста в целом. В этом случае тест загружается в одно *OpenEdX*-задание и вся схема получения первичных баллов должна быть прописана в соответствующем скрипте. Однако, вычисление интегральных характеристик теста и отдельных тестовых заданий, а также шкалирование результатов остаются за рамками средств платформы *OpenEdX* и требуют от преподавателя привлечения сторонних программ.

Таблица 1.2 – Сравнение ПСУН поддержки ЖЦ КИМ

Критерий	<i>БТСИМ</i>	<i>Moodle Quiz</i>	<i>Respondus</i>	<i>TAO</i>	<i>Learnosity</i>	СДО «Прометей»	<i>OpenEdX</i>	<i>КИМРА</i>
1.1		+	+	+	+		+	+
1.2				+	+		+	+
1.3		+			+		+	+
2					+			+
3	+	+		+	+	+	+	+
4		+						+

Критерий	<i>БТСИМ</i>	<i>Moodle Quiz</i>	<i>Respondus</i>	<i>TAO</i>	<i>Learnosity</i>	<i>СДО «Промей»</i>	<i>OpenEdX</i>	<i>КИМРА</i>
5		+	+	+	+			+
6	+						+	+
7		+	+	+	+	+	+	+
8	+	+	+					+
9								+
10					+			+
11		+	+		+		+	+
12		+						+
14		+	+	+				+
13				+				+
15			+			+		+
16		+	+		+		+	+
17.1		+		+	+		+	+
17.2	+	+	+	+	+	+	+	+
17.3		+		+	+		+	+
18	+	+	+	+	+		+	+

1.6 Требования к ПСУН поддержки ЖЦ КИМ

Для ПСУН, соответствующего современному состоянию компьютерных технологий, и отражающего перспективные направления развития образования, можно сформулировать следующий набор требований.

1. Управление всеми стадиями ЖЦ КИМ с поддержкой версионирования и повторного использования.
2. Поддержка как можно большего набора методологий ИвО на основе современных онтологий учебного процесса.
3. Учёт специфики различных дисциплин на основе онтологий предметных областей.
4. Поддержка банка КИМ с возможностью генерации контента для доставки материала к контрольно-измерительному мероприятию.
5. Организация коллаборативной среды для работы территориально распределённых групп пользователей.
6. Обеспечение совместимости с популярными форматами и протоколами интероперабельности ПСУН.

7. Учёт технологических и идеологических трендов развития *LMS* для обеспечения интеграции со следующими поколениями.

1.7 Выводы и результаты по главе

В главе введено понятие контрольно-измерительного материала (КИМ) и жизненного цикла КИМ (ЖЦ КИМ). На основе классификации бизнес-процессов высших учебных заведений и основных процессов измерений в форме тестирования определены подпроцессы, подлежащие автоматизации и связанные с ЖЦ КИМ. Поставлена задача эффективной информационной поддержки измерений в образовании.

Проанализированы существующие классификации тестовых заданий и предложена собственная компьютерно-ориентированная таксономия тестовых заданий, учитывающая связи между типами заданий и согласованная с необходимостью сохранения их во внутреннем представлении компьютера.

Изученные форматы представления тестовых заданий в компьютерных системах в основном являются частными (внутренними) форматами отдельных систем. Более широкие форматы, применяемые для обмена учебными курсами, например, *SCORM*, предоставляют возможность передачи между программными продуктами лишь заданий нескольких видов, не покрывающих полностью даже традиционную классификацию тестовых заданий. Наиболее широкие возможности для поддержки обмена тестовыми заданиями предоставляет спецификация *QTI/APIP*. Поэтому принято решение использовать *QTI/APIP* совместимость для экспорта/импорта заданий из КИМРА. Для внутреннего представления тестовых заданий в КИМРА разработан собственный *JSON*-основанный формат.

Приведён обзор ПСУН, автоматизирующих некоторые из описанных процессов с разной степенью детализации, акцент сделан на исследовании возможностей по поддержке хранилища заданий. По результатам обзора сформулированы требования к ПСУН поддержки ЖК КИМ, согласно которым разрабатывается РИСУН КИМРА. Вопросам отбора параметров математических моделей, реализуемых в КИМРА, посвящена следующая глава.

ГЛАВА 2. МАТЕМАТИЧЕСКИЕ МОДЕЛИ ИС КИМРА

В главе приводится описание, критический анализ, селекция и уточнение математических моделей и методов, положенных в основу математического обеспечения КИМРА, предложены математические модели, использованные в информационной системе. Особое внимание уделено математическим основам тестирования, как наиболее технологичному и формализованному виду контрольно-измерительных мероприятий.

На основе анализа параметров существующих моделей и процедур проведения контрольно-измерительных мероприятий выявляются и формализуются существенные параметры математических моделей, реализованных в КИМРА.

2.1 Математические модели тестирования

Введём следующие обозначения.

N – общее количество участников тестирования, номера участников $i = 1, 2, \dots, n$

n – количество участников тестирования, входящих в подгруппу

k – количество заданий в варианте теста, номера заданий $j = 1, 2, \dots, k$.

$A = (a_{ij})$ – матрица ответов, где

$$a_{ij} = \begin{cases} 1, & \text{если } i\text{-й участник верно решил } j\text{-ое задание} \\ 0, & \text{в противном случае} \end{cases} \quad (2.1)$$

RS_i – первичный балл i -го участника,

$$RS_i = \sum_{j=1}^k a_{ij} \quad (2.2)$$

IRS_j – первичный балл j -го задания

$$IRS_j = \sum_{i=1}^n a_{ij} \quad (2.3)$$

В рамках IRT a_{ij} являются случайными величинами, принимающими значение 1 с вероятностью $p_{ij} = p(\theta_i, \delta_j)$ и 0 с вероятностью $q_{ij} = 1 - p_{ij}$, содержание параметров θ и δ раскрыто в разделе 2.1.2.1.

2.1.1 Классическая теория тестирования

Классическая теория тестирования – психометрическая теория, представляющая факторы, воздействующие на эмпирически-наблюдаемый результат выполнения теста [138, 139].

Основная задача *СТТ* – получение для каждого i -го испытуемого истинного балла T_i , исходя из эмпирически-наблюдаемого результата X_i , с учётом ошибки измерения E_i [1]. *СТТ*, базируется на постулате [22]:

$$X_i = T_i + E_i \quad (2.4)$$

Среди задач *СТТ* также выделяют следующие [22].

1. Задачи селекции тестовых заданий.
2. Вычисление результата – балла для каждого участника тестирования.
3. Определение точности результата (надёжность или ошибка измерения).
4. Определение прогностической способности результатов тестирования (валидность или погрешность ошибки измерения).

2.1.1.1 Характеристики теста

Рассмотрим математические модели, относящиеся к тесту, как инструменту измерений, в целом. Из поставленных в предыдущем разделе задач *СТТ* следует, что наиболее существенными характеристиками тестов являются надёжность и валидность [140].

Выделяют следующие виды надёжности [104]:

- надёжность повторного тестирования – устойчивость во времени результатов, полученных при помощи теста. Вычисляется как коэффициент корреляции Пирсона для результатов, полученных при помощи одного и того же теста в разные моменты времени;
- надёжность параллельных форм теста применяется для определения надёжности или проверки эквивалентности надёжности параллельных форм теста. Отметим, что согласно [22] параллельными считаются тесты, для которых совпадают истинные значения баллов заданий, а также сред-

ние квадратические отклонения. Вычисляется как коэффициент корреляции Пирсона для результатов, полученных при помощи каждой формы теста;

- внутренняя согласованность теста используется для определения того, что все задания теста направлены на измерение одного и того же результата:
 - метод расщепления теста – наиболее простой и распространённый способ установления внутренней согласованности теста. Тест разделяется на две части, после чего между ними вычисляется мера связи. Традиционно применяется формула Спирмена-Брауна [20, 21]:

$$R_{KK} = \frac{K_{r_{11}}}{1 + (K - 1)r_{11}}, \quad (2.5)$$

где: r_{11} – надёжность части теста; K – отношение количества заданий в части теста к общему количеству заданий N ; R_{KK} – надёжность теста из K заданий. Легко видеть, что формула Спирмена-Брауна вводит зависимость надёжности теста от количества заданий в нём;

- вторым способом установления внутренней согласованности теста является вычисление альфы Кронбаха [141]:

$$\alpha = \frac{N}{N - 1} \left(\frac{s_y^2 - \sum s_i^2}{s_y^2} \right), \quad (2.6)$$

где: N – общее количество заданий; s_y^2 – среднеквадратическое отклонение (СКО) всех эмпирически-наблюдаемых результатов; $\sum s_i^2$ – сумма СКО по каждому заданию;

- для дихотомических тестов внутренняя согласованность устанавливается посредством формулы Кьюдера-Ричардсона [142]:

$$KR_{20} = \frac{N}{N - 1} \left(\frac{s^2 - \sum PQ}{s^2} \right), \quad (2.7)$$

где: N – общее количество заданий; s^2 – всех итоговых результатов теста; P – для каждого задания доля верно решивших; Q – для каждого задания доля неверно решивших; $\sum PQ$ – сумма произведений по всем заданиям.

Валидность – свойство измерительного инструмента, показывающее, что инструмент измеряет именно те характеристики, которые заявлены. Выделяют несколько видов валидности: содержательная, критериальная и конструктивная. Поскольку мы говорим преимущественно о тестах достижений, то в данной работе рассмотрим математические модели содержательной валидности (или валидности содержания). Определения и исследования, посвящённые другим видам валидности, могут быть найдены в работах [139, 143, 28].

Результатом вычисления валидности содержания является установление факта включения в тест всех вопросов, необходимых для измерения требуемого свойства. Количественным выражением валидности содержания является коэффициент содержательной валидности (*content validity ratio, CVR*), определяемый формулой [104]:

$$CVR = \frac{n_e - N/2}{N/2}, \quad (2.8)$$

где: n_e – количество экспертов, отметивших задание как подходящее для данного теста, N – общее количество экспертов.

2.1.1.2 Характеристики тестовых заданий

Перейдём к рассмотрению математических моделей, представленных в *СТТ* и описывающих характеристики тестовых заданий: дискриминативность и трудность.

Трудность тестового задания – свойство тестового задания, представляемое пропорцией испытуемых верно ответивших на задание к общему числу испытуемых. Коэффициент трудности (p -коэффициент) представляет долю студентов, ответивших на задание верно:

$$p_j = \frac{n_j}{N}, \quad (2.9)$$

где: n_j – количество студентов, ответивших на j -ое задание верно, а N – общее количество участников тестирования, ответивших на задание.

Допустимо вычисление коэффициента трудности с использованием метода контрастных групп, тогда формула приобретает вид:

$$p_j = \frac{n_{upper} - n_{lower}}{N}, \quad (2.10)$$

где: n_{upper} – количество студентов, ответивших на j -ое задание верно в группе лучших; n_{lower} – количество студентов, ответивших на j -ое задание верно в группе худших; N – общее количество участников тестирования, ответивших на задание.

Дискриминативность – различающая (дифференцирующая) способность тестового задания. Способность задания отличать тех, кто знает от тех, кто не знает, тех, кто владеет навыком от тех, кто не владеет навыком в зависимости от того, на измерение чего нацелено тестовое задание. Количественной характеристикой, описывающей дискриминативность, является коэффициент дискриминативности, который также допускает несколько вариантов вычисления.

Например, с привлечением контрастных групп:

$$d_j = \frac{n_{upper} - n_{lower}}{N}, \quad (2.11)$$

где: n_{upper} – количество студентов, ответивших на -ое задание верно в группе лучших; n_{lower} – количество студентов, ответивших на j -ое задание верно в группе худших; N – общее количество участников тестирования, ответивших на задание.

Для дихотомически оцениваемых заданий дискриминативность устанавливается при помощи точечно-бисериальной корреляции [144]:

$$d_j = \left(\frac{\bar{X}_j - \bar{X}}{S_y} \right) \sqrt{\frac{P_x}{1 - P_x}}, \quad (2.12)$$

где N – общее количество заданий; $\bar{X}_j$ – средний балл результатов теста участников тестирования, ответивших верно на -ое задание; S_y – стандартное отклонение для теста; P_x – доля участников тестирования, верно решивших j -ое задание.

2.1.1.3 Характеристики частей тестовых заданий

При анализе и интерпретации результатов тестирования для заданий закрытого типа с одним или несколькими верными ответами вычисляют характеристики не только заданий, но и некоторых их конструктивных частей: дистракторов. При определении вероятности угадывания верного ответа в задании, или выявлении

равнозначности в привлекательности неверных ответов, говорят об эффективности дистракторов [*effectiveness of distracters*] [145, 146].

Коэффициенты эффективности дистракторов для задания определяются:

$$de_k = \frac{n_{upper} + n_{lower}}{N} \cdot 100, \quad (2.13)$$

где: n_{upper} – количество студентов из группы лучших, выбравших k -й дистрактор, n_{lower} – количество студентов из группы худших, выбравших k -й дистрактор; N – общее количество участников тестирования.

2.1.2 Современная теория тестирования

Основой *IRT* являются два предположения [26]:

- 1) достижения участника тестирования по тесту могут быть спрогнозированы при помощи набора признаков (умений);
- 2) зависимость между достижениями участника тестирования и набором этих признаков может быть описано монотонной возрастающей функцией, которую в русскоязычной литературе принято называть функцией успеха [28, 27]:

$$p = p(s, t), \quad (2.14)$$

где: s – уровень подготовленности испытуемого; t – уровень трудности задания ($s, t \in (0; \infty)$).

Иными словами, вероятность верного решения участником тестирования определённого задания представляет собой функцию по крайней мере двух аргументов – уровня подготовленности испытуемого s и уровня трудности задания t [26].

Функции успеха моделей, рассмотренных ниже, мы будем записывать в экспоненциальной форме, для этого представим s и t в логарифмической форме и введём обозначения [28]:

$$\text{Ln}(s) = \theta, \ln(t) = \delta \quad (2.15)$$

В настоящее время используется много *IRT*-моделей, практически все из них включают один и более параметров, описывающих тестовое задание и один и более

параметров, описывающих участника тестирования. Кроме того, в отличие от моделей *СТТ*, модели *IRT* являются фальсифицируемыми. Та или иная модель может оказаться неподходящей для конкретных результатов тестирования.

Таким образом, основными задачами *IRT* являются следующие [28]:

- 1) установление (оценивание) параметров модели;
- 2) оценка адекватности модели данным (точность параметризации);
- 3) статистическая проверка гипотез тестирования.

2.1.2.1 Базовые модели *IRT*

Однопараметрическая логистическая модель (Раша, *1PL*) [26]:

$$p_i(\theta) = \frac{e^{(\theta - b_i)}}{1 + e^{(\theta - b_i)}}, i = 1, 2, \dots, n, \quad (2.16)$$

где: $\theta \in (-\infty; \infty)$ – уровень подготовленности испытуемого; $b_i \in (-\infty; \infty)$ – трудность i -го задания; n – общее количество заданий теста.

Двухпараметрическая логистическая модель (Бирнбаума, *2PL*) [26]:

$$p_i(\theta) = \frac{e^{Da_i(\theta - b_i)}}{1 + e^{Da_i(\theta - b_i)}}, i = 1, 2, \dots, n, \quad (2.17)$$

где: D – коэффициент масштабирования; $a_i \in [0; \infty)$ – дискриминативность; а θ, b_i определяются также, как в модели *1PL*.

Трёхпараметрическая логистическая модель (*3PL*) [77]:

$$p_i(\theta) = c_i + (1 - c_i) \frac{e^{(\theta - b_i)}}{1 + e^{(\theta - b_i)}}, i = 1, 2, \dots, n, \quad (2.18)$$

где: $c_i \in [0; 1]$ – вероятность угадывания ответа на i -ое задание, а коэффициенты θ, a_i, b_i определены как в модели *2PL*.

2.1.2.2 Методы оценивания параметров моделей *IRT*

В различные годы были предложены методы оценивания параметров моделей *IRT*. Наиболее простым является алгоритм усреднения значений функции измерения [28]. Оценки параметров, полученные при помощи этого метода, являются состоятельными, но не эффективными.

Достаточно просто вычислить эффективные оценки можно, например, при помощи метода моментов. Для моделей *IRT* предлагается приравнивать первичные

баллы участников тестирования и тестовых заданий соответствующим математическим ожиданиям:

$$RS_i = E[RS_j] = \sum_{j=1}^k p_{ij}, i = 1, 2, \dots, N \quad (2.19)$$

$$IRS_j = E[IRS_i] \equiv \sum_{i=1}^N p_{ij}, j = 1, 2, \dots, k \quad (2.20)$$

Алгоритм вычислений оценок по методу моментов для результатов тестирования описан в [28].

Наиболее распространённым среди практиков является метод максимального правдоподобия, основанный на понятии максимального правдоподобия, введённого Фишером [32]. Оценки, получаемые при помощи этого метода, являются состоятельными, асимптотически несмещённым и асимптотически эффективными. Оценки имеют наименьшую дисперсию по сравнению с другими асимптотически нормальными и распределены асимптотически нормально с параметрами [28]:

$$E[\hat{\theta}] = \theta, E[\hat{\delta}] = \delta \quad (2.21)$$

$$D[\hat{\theta}] = - \left\{ E \left[\frac{\partial^2 L}{\partial \theta^2} \right] \right\}^{-1} \approx \left[\frac{\partial^2 L}{\partial \theta^2} \right]_{\theta=\hat{\theta}, \delta=\hat{\delta}}^{-1} \quad (2.22)$$

$$D[\hat{\theta}] = - \left\{ E \left[\frac{\partial^2 L}{\partial \delta^2} \right] \right\}^{-1} \approx \left[\frac{\partial^2 L}{\partial \theta \partial \delta^2} \right]_{\delta=\hat{\delta}, \theta=\hat{\theta}}^{-1} \quad (2.23)$$

Если существуют эффективные оценки, то метод максимального правдоподобия даёт эти оценки. Также если существуют достаточные оценки, то метод даёт и их.

Для оценивания параметров различных моделей *IRT* методом максимального правдоподобия были предложены несколько процедур, широко применяемых на практике. Так, в работе Ф. Лорда [147, 148] описана автоматизация вычислений безусловного максимального правдоподобия (*unconditional maximum likelihood, UML*) и объединённого максимального правдоподобия (*joint maximum likelihood, JML*). Р. Боком в [149] описан метод оценивания с использованием минимизации маргинального правдоподобия (*minimizing the marginal likelihood, MML*). В работе

[150] Е. Андерсена установлены оценки свойств, получаемые по методу условного максимального правдоподобия (*conditional maximum likelihood, CML*), предложенного Д. Рашем в [27]. Несколькими коллективами учёных в ряде работ [151, 152, 153, 154] была описана процедура объединённого и маргинального Байесовского оценивания (*joint and marginal Bayesian estimation*). Кроме того, были предложены метод эвристического оценивания – в работах В. Ури [155, 156] – и метод, основанный на нелинейном факторном анализе в работах Р. Макдональда [157, 158].

2.1.2.3 Методы проверки адекватности модели данным

Преимущества методов *IRT* проявляются только в том случае, если выбранная модель подходит для данных, полученных в результате тестирования. В случае слабого соответствия модели данным полученные оценки параметров заданий и участников тестирования не продуктивны.

Согласно [26] решение о соответствии модели данным основывается на выраженности трёх признаков:

- 1) валидность предположений модели для данных, полученных в результате тестирования;
- 2) диапазоны, в которых ожидается изменение полученных параметров модели;
- 3) точность прогнозов модели для реальных (или имитированных) данных.

Подходы, алгоритмы и примеры оценивания указанных признаков подобраны и описаны в [26].

2.1.2.4 Модели характеристик тестовых заданий

Дискриминативность тестового задания, то есть его способность различать участников тестирования по уровню подготовки, представляется в *IRT* коэффициентом дискриминации задания. Коэффициент дискриминации вычисляется как выборочный коэффициент точечно-бисериальной корреляции [28]:

$$\hat{\rho}_{pb} = \frac{\bar{x}_1 - \bar{x}_0}{\hat{\sigma}_x} \sqrt{\frac{n_1 n_0}{n(n-1)}}, \quad (2.24)$$

где: $\bar{x}_1$ и $\bar{x}_0$ – среднее значение критериального показателя выполнивших и не выполнивших исследуемое задание; соответственно, n_1 и n_0 – количество выполнивших и не выполнивших; $n = n_1 + n_0$; $\hat{\sigma}_x$ – оценка среднего квадратического отклонения показателей критерия.

Чем ближе значение коэффициента дискриминации к единице, тем эффективнее задание позволяет дифференцировать испытуемых по уровню их подготовленности. Если коэффициент корреляции меньше нуля, то задание считается неудачным и подлежит переработке.

2.1.2.5 Методы работы с частями заданий

Говоря об эффективности дистракторов заданий опций и альтернатив, среди признаков удачного подбора дистракторов выделяют факт выбора различных дистракторов одинаково часто [28].

В контексте *IRT* говорят о проверке статистической гипотезы о равномерном распределении частот выбора различных дистракторов. Вопросы, связанные с равномерностью распределений, лежат за рамками данной работы. Достаточно подробно эффективные критерии согласия распределения с равномерным рассмотрены в [159]. Пример расчётов и использования критерия χ^2 [160] для проверки гипотезы о типе распределения описан в [28].

2.1.3 Теория тестлетов и модели компьютерного адаптивного тестирования

Математические модели *TRT* тесно связаны со специальной трактовкой понятия «тестлет», введённой в работе Г. Вайнера и Д. Къели [161]. В данной работе мы будем опираться на определение тестлета, приведённое в работе [33]: **тестлет** – это группа заданий, которые разрабатываются как единый блок и управляются совместно.

Несмотря на то, что за последние полвека модели *IRT* были согласованы с тестлетами и могут быть использованы при управлении и обработке соответствующих тестов [45, 40], наиболее современные модели *TRT* связаны с использованием байесовского подхода.

В основе моделей *TRT* лежат две базовые модели, пригодные для дихотомически оцениваемых и политомически оцениваемых тестовых заданий. Для дихотомической трёхпараметрической модели:

$$p(Y_{ij} = 1) = c_j + (1 - c_j)\text{logit}^{-1}(t_{ij}) \quad (2.25)$$

где: Y_{ij} – ответ i -го испытуемого на j -ое задание; c_j – нижняя асимптота (параметр угадывания для j -го дихотомически оцениваемого тестового задания; $\text{logit} = \log\left(\frac{1}{1-x}\right)$.

Для политомически оцениваемых заданий – модель *TRT*, представленная в работе Ф. Самеджимы [162]:

$$p(Y_{ij} = r) = \Phi(d_r - t_{ij}) - \Phi(d_{r-1} - t_{ij}), \quad (2.26)$$

где: d_r и d_{r-1} – латентные границы (предельные значения результата) для политомических заданий; Φ – кумулятивная функция нормального распределения; t_{ij} – латентная линейная экстраполирующая функция результата.

Согласно [33] для двухпараметрических дихотомических заданий, *3PL* модель принимает специальный вид ($c_j = 0$):

$$p(Y_{ij} = 1) = \text{logit}^{-1}(t_{ij}). \quad (2.27)$$

Дополнительные зависимости в терминах тестлетов промоделированы расширением линейной экстраполирующей функции результата: $t_{ij} = a_j(\theta_i - b_j)$, путём введения, так называемого эффекта тестлета $\gamma_{id(j)}$ для j -го испытуемого и i -го задания, включённого в тестлет $d(j)$. Для двух заданий j и j' из одного тестлета справедливо $d(j) = d(j')$.

При этом a_i , b_i и θ – дискриминативность, трудность и функция успеха испытуемого.

Заметим, что трёхпараметрическая модель *3PL* байесовская модель тестлетов отличается от стандартной *IRT* *3PL* модели всё тем же коэффициентом $\gamma_{id(j)}$.

$$p(Y_{ij} = 1) = c_j + (1 - c_j)\text{logit}^{-1}\left(a_j(\theta_i - b_j - \gamma_{id(j)})\right) \quad (2.28)$$

Для обеих моделей предварительные распределения для параметров моделей: $\theta_i \sim N(0,1)$, $\log(a_j) \sim N(\mu_a, \sigma_a^2)$, $b_j \sim N(\mu_b, \sigma_b^2)$.

Модели *TRT* порождены и тесно связаны с компьютерным адаптивным тестированием. Подробное описание математического аппарата, предлагаемого в рамках *TRT*, а также связанных с ними исследований можно найти в работах [44, 54, 64, 77, 33].

Для моделей *CAT* атрибуты заданий обобщают в следующие группы [77]:

- 1) категориальные (авторы, содержимое, формат, сегмент БТЗ, к которому относится задание);
- 2) количественные (трудность, предполагаемое время решения, допустимые значения);
- 3) логические (взаимосвязи между заданиями).

Эти атрибуты используются при проведении *CAT* для решения двух следующих основных задач:

- 1) сегментирования [*partitioning*] хранилища заданий;
- 2) подбора заданий [*item selection*] в процессе адаптивного тестирования.

Рассмотренные выше тестлеты представляют собой один из самых современных и наиболее технологичных вариантов сегментирования хранилища заданий для нужд *CAT*, однако существуют и другие способы сегментации [77, 40, 163, 164].

Поскольку КИМРА предполагает возможность применения КИМ для проведения разных видов тестирований, модель КИМ должна поддерживать как минимум параметры, предлагаемые рассмотренными теориями, а также дополнительные параметры, позволяющие решать задачи сегментации КИМ произвольного типа. Поскольку КИМРА не поддерживает процессы *CAT*, задачи селекции при адаптивном тестировании ложатся на внешнюю систему, возможность интеграции с которой учитывается в архитектуре КИМРА.

2.2 Модель жизненного цикла контрольно-измерительного материала

Разработка компьютерных систем поддержки измерений в образовании требует однозначного понимания стадий жизненного цикла КИМ как основного артефакта проектирования и управления. В разделе 1.2 уже упоминалось, что специалистами в области организации массовых тестирований выделены этапы разработки тестов [13], которые в расширенном и обобщённом виде могли бы лечь в основу подобной модели: создание банка, формирование теста, доставка теста, получение данных от прохождения теста и анализ данных по тесту. Заметим, что для однозначного управления тестом в РИСУН или согласовании взаимодействия нескольких систем необходима детализация указанных процессов и понимание характеристик теста в каждом из них. Для обеспечения поддержки произвольных КИМ в РИСУН следует учесть следующие особенности: способы добавления/получения/удаления КИМ из банка, правила формирования композитного КИМ при получении из банка, информативная обратная связь по КИМ, правила обновления характеристик КИМ и актуализации КИМ.

Заметим, что методологический фундамент для управления ЖЦ КИМ, определённого в разделе 1.1.1, предполагает управление конфигурацией [*configuration management*] в понимании системной инженерией. Предложим упрощённые версии определений из стандарта ISO/IEC/IEEE 24765-2010 «System and software engineering – Vocabulary».

Конфигурация [*configuration*] – это совокупность версий значимых артефактов в системе. Версия [*version*] – это состояние артефакта, которое может быть восстановлено в любой момент времени независимо от истории изменения системы. Версионирование [*versioning*] – процесс создания, отслеживания и восстановления версий.

Таким образом, в данной работе ЖЦ КИМ понимается, как классической модели многостадийного жизненного цикла версионизируемого информационного артефакта, встроенного в систему с отношениями часть/целое и обобщения/детализации [165]:

$$EAM_LC \stackrel{\text{def}}{=} \langle EAM, LCD, AResults, LO \rangle, \quad (2.29)$$

где: EAM – собственно КИМ; $AResults \stackrel{\text{def}}{=} (AR_i), i \in [1, ARCount]$ – вектор результатов контрольно-измерительных мероприятий, элементы которого упорядочены по отметкам времени начала мероприятия; LO – результаты обучения, представленные онтологией предметной области, набором таксономий дидактических единиц или другим вариантом формализации, позволяющим ссылаться на отдельные объекты; LCD – параметры жизненного цикла, которые не задаются в форме единовременного определённых объектов (тип, тип хранилища, вариант формализации LO , возможные способы доставки и др.).

Поскольку каждый следующий результат в векторе $AResults$ имеет увеличенную отметку времени, то это позволяет проанализировать полученный временной ряд и на любой момент времени вычислить актуальные характеристики КИМ и сравнить их.

Технологически сложным является организация глобально непротиворечивого версионирования в распределённой РИСУН, когда КИМ используется в различных, территориально удалённых пунктах и требуется специальная система синхронизации версионизируемого артефакта, в нашем случае КИМ [166].

2.3 Модель контрольно-измерительного материала

Формализуем КИМ в виде четвёрки [165]:

$$EAM \stackrel{\text{def}}{=} \langle TaskDesc, Body, CCCS, Pr \rangle, \quad (2.30)$$

- $TaskDesc$ – кортеж элементов описания КИМ;
- $Body$ – тело КИМ, то есть содержательная часть КИМ, непосредственно используемое в контрольно-измерительном мероприятии;
- $CCCS$ – непустое множество ссылок на элементы описания результатов обучения;
- Pr – множество пререквизитов КИМ.

Раскроем компоненты формализации КИМ.

Кортеж $TaskDesc$ включает следующие обязательные элементы.

- *GUID* – глобальный уникальный идентификатор конкретного состояния (версии) КИМ;
- *Version* – идентификатор версии КИМ;
- *Description* – полное название;
- *Slug* – короткое название;
- *Comment* – комментарий;
- *Authorship* – множество идентификаторов авторов КИМ;
- *Type* – тип задания, определяющее правила интерпретации *Body*;
- *Tags* – множество логических атрибутов (тегов) КИМ;
- *TimeStamps* – кортеж отметок времени создания и модификаций КИМ.

Тело КИМ есть четвёрка $Body \stackrel{\text{def}}{=} \langle CEAM, Assignment, Context, GR \rangle$, представляющая собой контейнер с данными, необходимыми для проведения контрольно-измерительного мероприятия. Компоненты четвёрки *Body* имеют следующий смысл:

- $CEAM = \{EAM_1 \dots EAM_z\}$ – множество вложенных КИМ. Если множество *CEAM* пусто, то КИМ будем называть *атомарным*, в противном случае – *композиционным*;
- *Assignment* – задание, понимаемое как инструкции для студента по конкретному КИМ, включающее форму представления *результата*. *Assignment* может ссылаться на задания $Assignment_1, \dots, Assignment_z$ вложенных КИМ;
- *Context* – контекст КИМ, то есть некоторая информация, необходимая для адекватного восприятия и выполнения задания *Assignment*. *Context* может содержать элементы онтологии учебного процесса, комментарии к результатам обучения, указания по выполнению КИМ и др. *Context* также может ссылаться на контексты $Context_1, \dots, Context_z$ вложенных КИМ;

- *GR* – схема оценивания КИМ, являющаяся отображением *результата* выполнения КИМ в *оценку* по произвольной шкале (обзор шкал измерений см. в разделе 2.7). Эта функция может включать правила и сценарии (алгоритмы) оценивания.

Возможные типы контейнеров:

- тестовое задание [*test item*];
- задание для взаимного оценивания [*peer assessment task*];
- индивидуальное задание по программированию [*individual programming assignment*];
- веб-квест [*web quest*];
- задание построения интеллект-карты [*mind map*];
- и др.

Существенно, что задание *Assignment* некоторых типов контейнеров может содержать другие КИМ (например, частью задания на взаимное оценивание может быть тест или программный проект).

Контекст КИМ может быть двух типов: *локальный* и *глобальный*. *Локальный* контекст тела КИМ. Локальный контекст никак не привязывает КИМ к курсу, учебной программе и проч. Глобальный контекст содержит как минимум одну ссылку на *модуль*, где модуль может быть уроком, блоком, учебной дисциплиной, образовательной программой, то есть произвольной частью учебного процесса. При изменении модуля ссылка на него не изменяется, что удобно для организации версионирования КИМ. Минимально возможный контекст – *пустой*, что обозначает самодостаточность задания *Assignment* с точки зрения элементов *CCCS*. Максимально возможный контекст – *образовательная программа*. Контекст *Context* может наследоваться. Например, контекст курсовой работы есть дисциплина или группа дисциплин, а контекст выпускной квалификационной работы – образовательная программа.

Множества *CCCS* и *Pr* являются подмножествами объектов из *LO* в модели жизненного цикла и их можно формализовать по-разному в зависимости от проработанности сущностей, описывающих результаты обучения: от простейшего тематического членения результатов обучения до онтологии предметной области.

Осознавая недостаточную проработанность и сложность интеграции доступных средств онтологического моделирования содержания обучения, предлагается в текущей архитектуре КИМРА воспользоваться *таксономией дидактических единиц* (ДЕ), которой посвящён следующий раздел.

Предложенная модель снимает проблемы универсального хранения КИМ, версионирования КИМ в БКИМ, а также облегчает постановку задачи автоматической генерации композитных КИМ. Отметим однако, что не исчезает проблема совместимости КИМ разного типа, например, в рамках одного композитного КИМ. Требуется вводить ограничения на композицию, которые накладываются известными типами контейнеров, реализованных в конкретной ПСУН.

Пример атомарного задания «Веб квест» с рубриками оценивания (задание использовалось в курсе «*Mobile Game Development*» магистерской программы «Системная и программная инженерия», ФКН, НИУ ВШЭ)

Context: Course: Mobile games development

Web Quest topic: “Genre, Positioning, Monetizing”

Variant 1: Air Simulator

Assignment: Introduction

The goal of this Web Quest is to round up theory from the first lectures and to study monetizing models for mobile games.

You should use suggested websites and any related material to complete the task.

Task

1. Examine the links from *Links* section.
2. Complete the assignments from *Assignments* section.
3. Prepare a report, which contains the results of each assignment.

Links

- 1) Mobile Air Simulators:
 - a. Google Play (<https://play.google.com/store/search?q=air%20simulator&c=apps>)
 - b. Windows Market (<https://www.microsoft.com/ru-ru/search/result.aspx?q=air+simulator&form=apps>)
- 2) Monetization models:

- a. Monetizing Mobile Gaming (<https://techcrunch.com/2016/02/28/mobile-gaming-trends/>)
- b. Choosing a monetization model for your mobile game (<https://crowdsourcedtesting.com/resources/choosing-monetization-model-mobile-game/>)
- c. The Design of Free-To-Play Games: Part 1 (http://www.gamasutra.com/view/feature/6552/the_design_of_freetoplay_games_.php)
- d. The Design of Free-to-Play Games, Part 2 (http://www.gamasutra.com/view/feature/134959/the_design_of_freetoplay_games_.php?print=1)
- e. What are some new mobile app monetization strategies for games? (<https://www.quora.com/What-are-some-new-mobile-app-monetization-strategies-for-games>)
- f. Advertising as mobile game monetization tactic (<https://clickky.biz/blog/2016/03/advertising-as-mobile-game-monetization-tactic/>)
- g. Understanding Your Options When Monetizing Mobile Games (<https://appdeveloperomagazine.com/1275/2014/4/1/Understanding-Your-Options-When-Monetizing-Mobile-Games>)

Assignments

- 1) Design the short-lists of free, freemium, and paid Air Simulators (3 of each type, 9 in total). Give a link to a description of each game.
- 2) From your shortlist give overall picture of a niche (casual game, hardcore game, etc.), audience (use gamer's classification), game goals and the requirements for their achievements.
- 3) Analyze differences and similarities of air simulators and create the matrix of features (obligatory and distinctive) for a mobile air simulator based on your short-lists.
- 4) Think about the originality. Make up and explain 3 original features of your air simulator.
- 5) Describe UI Patterns used in applications from the short-lists. Give examples.
- 6) Examine the types of user's input/output in the simulators from the short-list. Give examples.
- 7) Explain the user's input/output modes in your simulator.
- 8) Explain your set of UI Patterns.
- 9) Analyze game engines used in your short-lists and create the pool of engines suitable for your genre.
- 10) Suggest a monetization model for your air simulator, expose the characteristics of the model.

GR: Evaluation Rubrics. Полная таблица рубрик с правилами конструирования оценок за задание представлена в 0.



Пример. Композитное КИМ «Тест»

Тест составляется, как КИМ, состоит из тестовых заданий (тоже КИМ). *TaskDesc* для теста через спецификацию [167] (см. таблицу) связан с кортежами описания каждого тестового задания, но не является их прямым объединением.

Каждое КИМ из теста оценивается дихотомически или политомически в согласии с математическим аппаратом той теории тестирования, которая будет использоваться при шкалировании, то есть правило оценивания всего теста определяется шкалой оценивания (подробнее о шкалах см. раздел 2.7).

Тема	Разбиение темы	Количество вопросов		
		Всего	С кодом	
Идентификаторы, лексемы, служебные слова		3		
Всего вопросов по теме:				2
Типы данных		3	1	
Всего вопросов по теме:				2
Операции	Арифметические	1	1	
	Логические	1	1	
	Битовые	1	1	
	Приведение типов	1	1	
	Тернарная	1	1	
Всего вопросов по теме:				6
Переменные		2	1	
Всего вопросов по теме:				3
Комментирование кода		1		
Всего вопросов по теме:				
Форматирование при выводе		2	2	
Всего вопросов по теме:				2
Операторы	Условный	3	1	
	Переключатель	2	1	
	Цикл с предусловием	2	1	
	Цикл с постусловием	2	1	
	Универсальный цикл	2	1	
Всего вопросов по теме:				11
Одномерные массивы	Методы класса Array	2		
	Работа с массивами	2	2	
Всего вопросов по теме:				4
Итого:			16	30
Время прохождения теста: 40 минут				

Таблица 2-1. спецификации теста по дисциплине «Программирование», 1 модуль 2012-2013 уч. год, образовательная программа «Программная инженерия», департамент «Программной инженерии» факультета «Бизнес-информатика», НИУ ВШЭ

вичные баллы студентов получены по аддитивной функции заданий, каждое задание оценено дихотомически и взято с единичным весом. Шкалирование результатов проведено по относительной шкале, где за максимум взят максимальный балл после выбраковки.

Пример. Экзаменационная работа по дисциплине «Программирование» 1 курса бакалаврской образовательной программы «Программная инженерия» факультета компьютерных наук НИУ ВШЭ

Пример подробного описания особенностей правил оценивания работ по дисциплине «Программирование» можно найти в работе [168]. КИМ для экзамена по программированию состоит из двух частей: письменной контрольной работы и компьютерного теста.

Пример задания для письменной контрольной работы по программированию.

Разработать метод **CreateArray()**, возвращающий сформированный и заполненный массив из **N** целочисленных элементов со случайными значениями из диапазона **[-2;2]**.

Разработать метод **ShowArray()** для вывода в консоль массива, ссылка на который передана в параметре. Элементы массива при выводе разделены символом пробела и выводятся на экран по десять в строке.

Разработать метод **MergeArray()**, формирующий массив **C** как объединение двух массивов **A** и **B** по следующему правилу: в массив **C** записываются попеременно элементы массивов **A** и **B**, имеющие положительные значения. Ссылки на массивы **A**, **B** и **C** – параметра метода.

В основной программе при помощи метода **CreateArray()** сформировать массивы **A** и **B** из **K** и **M** элементов, соответственно (**K**, **M** – целые числа вводятся пользователем). По массивам **A** и **B** сформировать, используя метод **MergeArray()**, массив **C**. Исходные и результирующий массивы вывести на экран при помощи метода **ShowArray()**.

Правила оценивания для контрольной работы (**P**) зафиксированы в программе учебной дисциплины «Программирование» (<https://www.hse.ru/edu/courses/184771486>), оценка по компьютерному тесту (**T**) составляется согласно принятой на дисциплине регрессивной шкале. Правила оценивания экзамена (экзаменационного композитного КИМ) (**E**) определены в программе учебной дисциплины и формализованы в формуле:

$$E = [0,8 \cdot \min(T, P) + 0,2 \cdot \max(T, P)]$$

2.4 Модель результата контрольно-измерительного мероприятия, использующего контрольно-измерительный материал

Результат подразумевает фиксацию данных, связанных с конкретным контрольно-измерительным мероприятием и описывающих используемый КИМ, полученные ответы (*Answer*), контингент испытуемых (*Cohort*) и среду проведения испытания (*Env*). Формализация среды проводится с учётом компьютерной доставки КИМ [165].

$$AM_ER \stackrel{\text{def}}{=} \langle (EAM_1 \dots EAM_z), Cohort, Answer, Env \rangle, \quad (2.31)$$

Отметим, что в случае композитного КИМ модель результата ссылается на сам КИМ, обозначенный EAM_1 , и на составляющие его КИМ ($EAM_2 \dots EAM_z$). Примером может служить получение характеристик тестом и каждым входящим в него тестовым заданием после прохождения апробации.

2.5 Модель связи контрольно-измерительного материала с результатами обучения и учебными материалами

Главная характеристика КИМ с точки зрения преподавателя – проверяемые результаты обучения, потому что собственно проверка результатов обучения и является целью создания КИМ. Наиболее популярная процедура отбора тестовых заданий в тест основана на указании набора проверяемых результатов обучения. Классики тестологии считают результаты обучения латентной характеристикой обучаемого, а системные аналитики считают это частью онтологии предметной области, если эти результаты относятся к актуальным знаниям. Понятно, что соответствие КИМ результатам обучения требует эмпирической проверки и валидации, по результатам которых может изменяться либо состав проверяемых результатов, либо содержание КИМ. Таким образом, необходимо формализовать связь между каким-либо представлением результатов обучения и выбранной моделью КИМ.

2.5.1 Формализация через таксономию дидактических единиц

Дидактической единицей (ДЕ) в российской традиции (например, согласно Письму Министерства образования Российской Федерации от 31 января 2002 г. № 18-52-116ин/18-15 «О рекомендациях по разработке примерных программ учебных дисциплин по специальностям среднего профессионального образования») принято считать элемент содержания учебного материала, изложенного в виде утверждённой в установленном порядке программы обучения в рамках определённой профессиональной дисциплины или общеобразовательного предмета. ДЕ – одна из предметных тем, подлежащих обязательному освещению в процессе подготовки специалистов, обучающихся по данной дисциплине (предмету).

Там же, как одно из основных направлений использования для ДЕ, определено сравнение учебных программ. В данной работе для формализации контрольных мероприятий и связанных с ними объектов предлагается определить набор таксономий ДЕ, где каждая таксономия соответствует некоторой предметной области (отметим, что предметная область не формализуется и выбирается преподавателем/экспертом произвольно).

Отметим важнейшее требование к формализации результатов КИМ через ДЕ. Таксономии ДЕ должны участвовать в жизненном цикле КИМ, то есть быть версионизируемыми. Это означает, что конкретный КИМ связывается не просто с «названием» ДЕ, а с её версией, актуальной для данного КИМ на момент последнего изменения. Например, в заданиях по программированию для одной и той же ДЕ допустимы вариации КИМ по версиям языка программирования.

Пример. Тестовое задание по языку С# (таксон «Программирование / Языки программирования / Язык С# / Форматированный вывод»)

Версия для С# 5.0

В результате выполнения следующего фрагмента программы:

```
int value1 = 10;
int value2 = 15;
string output = string.Format("{0}-{1}", value1, value2);
Console.Write(output);
```

на экран будет выведено: _____.

Версия для С# 6.0

В результате выполнения следующего фрагмента программы:

```
int value1 = 10;
int value2 = 15;
string output = $"{value1}-{value2}";
Console.Write(output);
```

на экран будет выведено: _____.

■

Введём обозначения. Пусть *ССС* – ДЕ, *Root* – корень таксономии для данной *ССС*. Будем считать корень таксономии тегом, определяющим и называющим «предметную область», что довольно удобно, так как позволяет не вводить новые сущности для именования предметных областей. Введём также обозначение

$ancestors(CCC)$ для множества предков ДЕ CCC , то есть всех таксонов в пути от родителя CCC до $Root$.

Предметную область определяет автор соответствующей таксономии, никаких ограничений на число таксономий не накладывается. Это может вызывать проблему дублирования таксономий, что решается отождествлением ДЕ из разных таксономий, фактически представляющих одну и ту же предметную область.

Далее множество всех КИМ будем обозначать $S_{EAM} = \{EAM_i\}$, а множество всех ДЕ будем обозначать $S_{CCC} = \{CCC_j\}$, если не указано иное.

Одной из главных характеристик КИМ является множество ссылок на CCC , обозначенное в модели КИМ как $CCCS$, и формализуемое в случае таксономического членения ДЕ как $CCCS \subseteq S_{CCC}$. По построению, множества КИМ и ДЕ образуют двудольный орграф структуры содержания $GR_{CoEAM} = \langle S_{EAM}, S_{CCC}, E \rangle$, где $E = \bigcup_{a \in S_{EAM}} CCCS(a)$ так как все ссылки, очевидно, идут из множества КИМ (S_{EAM}) в множество CCC (S_{CCC}).

2.5.2 Фолксономия логических атрибутов, служащих для сегментации

С учётом принципов открытой экспертизы, характеристика которой будет представлена в главе 3, наиболее удобным оказывается построить множество логических атрибутов *Tags* КИМ в виде фолксономии [*folksonomy*], являющейся объединением пометок пользователей, оценивающих сходство КИМ с точки зрения решающего их студента. Понимая проблемы фолксономий, такие как необходимость привлечения средств обработки естественного языка (хотя и на самом низком уровне), возможность появления тегов с опечатками и др., пока кажется невозможным предложить другой универсальный механизм создания ограничений на автоматический отбор множества КИМ с требованием отсутствия «слишком сходных» заданий [169, 170]. Возможное развитие этого подхода описано в [171] с учётом напрашивающегося перехода к представлению CCC в виде элемента онтологии предметной области, надёжность данного подхода и вариант реализации предложен в [172].

2.5.3 Анализ взаимосвязей между КИМ

Существуют тесты (например, относящиеся по В.С. Аванесову [1] к нетрадиционным, адаптивным тестам), для которых важна возможность анализа взаимосвязей непосредственно между тестовыми заданиями. Очевидно, что, имея связи тестовых заданий с единицами содержания учебных дисциплин, можно визуализировать связи и между тестовыми заданиями.

Графы, отображающие связи между тестовыми заданиями в тесте, будем называть графами *взаимосвязей между тестовыми заданиями*.

Однако построение и анализ подобных графов вручную – задача трудоемкая и требующая специальных знаний в области математики и тестологии. Автоматизация процесса позволяет упростить данную процедуру, сократив время на построение и анализ графа, и предоставить:

- разработчику – базу для предварительного анализа теста, минимизирующую необходимость привлечения экспертов;
- эксперту – удобный и наглядный материал, упрощающий и ускоряющий проведение экспертиз.

Постановка задачи построения (и визуализации) структуры педагогического теста [173]. Для формализации задачи визуализации необходимо:

- 1) рассмотреть априорную информацию о педагогическом тесте, аккумулированную в описании его содержания и спецификации теста;
- 2) получить визуальное представление связей между:
 - тестовыми заданиями и структурными единицами (разделами) учебной дисциплины вне зависимости от вида теста (текущего, промежуточного или итогового контроля);
 - заданиями одного теста;
 - индивидуальными результатами тестирования испытуемых с конкретными единицами учебной дисциплины;
- 3) Ввести обозначения, описать правила и механизмы, связанные с предполагаемой графической моделью.

4) Проанализировать возможность оценки предварительного качества тестов на основе полученных графических моделей.

Тестовые задания (и иногда тесты в целом) являются частным случаем КИМ. В общем виде ставится задача построения (и визуализации) графа взаимосвязей КИМ $GR_{EAM} = \langle V, E, A \rangle$, в котором КИМ являются вершинами и связываются друг с другом *дугами* (со смыслом «часть-целое» по тематике) и *рёбрами* (со смыслом «племянники» по отношению к некоторой ДЕ) в зависимости от топологии *графа структуры содержания*.

Дуги «часть-целое» (ЧЦ). Дуга ЧЦ связывает два КИМ, для которых хотя бы пара проверяемых ими ДЕ связана отношением «таксон-подтаксон» в таксономии предметной области: $(a, b) \in A \Leftrightarrow \exists d \in CCCS(b): d \in ancestors(g), g \in CCCS(a)$

Вес w дуги (a, b) :

$$w(a, b) = \sum_{\substack{g \in CCCS(a), \\ d \in CCCS(b)}} I(g, d)$$

$$I = \begin{cases} 1, & d \in ancestors(g) \\ 0 & \end{cases}$$

Рёбра «племянники». Такие рёбра связывают два КИМ, если они имеют общего предка, то есть $\{a, b\} \in E \Leftrightarrow (a \neq b) \& (\exists d \in CCCS(a), g \in CCCS(b): (\exists f: f \in ancestors(d) \& f \in ancestors(g)))$.

Вес w ребра $\{a, b\}$:

$$w(\{a, b\}) = \min_{\substack{f \in ancestors(d) \cap \\ ancestors(g)}} (length(f, d) + length(f, g))$$

Для визуализации графа взаимосвязи КИМ удобно сделать настройку, которая ограничивает показ рёбер и дуг только такими, веса которых меньше заданного значения, что обеспечивает удобство выявления КИМ сходной тематики.

Заметим, что связанные по структуре содержания КИМ могут быть весьма различны по использованию в композитных КИМ.

Пример. Построим граф взаимосвязей КИМ для КИМ, представленных задачами по физике и композитным КИМ – контрольной работой (рисунок 2.1). Используются задачи и структура ДЕ из сборника: *Рымкевич А.П., Рымкевич П.А. Сборник задач по физике для 8-10 классов средней школы. – 4-е изд. – М.: Просвещение, 1979. – 160 с.*

654. (Закон Кулона) Найти величину каждого из двух одинаковых зарядов, если в масле на расстоянии 6 см друг от друга они взаимодействуют с силой 0,4 мН.

712. (Электроёмкость) Насколько увеличится потенциал шара радиусом 3 см при сообщении ему заряда 20 нКл?

725. (Электроёмкость, конденсаторы) К воздушному конденсатору, заряженному для напряжения 210 В, присоединили параллельно такой же незаряженный конденсатор, но с диэлектриком из стекла. Какова диэлектрическая проницаемость стекла, если напряжение на зажимах батареи оказалось равным 20 В?

734. (Электроёмкость, конденсаторы, энергия электрического поля) Во сколько раз изменится энергия поля заряженного конденсатора, если пространство между пластинами конденсатора залить маслом? Рассмотреть случаи: а) конденсатор отключён от источника напряжения; б) конденсатор остаётся присоединённым к источнику постоянного напряжения. Ответ объяснить, пользуясь законом сохранения энергии.

Контрольная работа (*Механика, электродинамика*)

- 1) (349). На вагонетку массой 800 кг, катящуюся по горизонтальному пути со скоростью 0,2 м/с, насыпали сверху 200 кг щебня. Насколько при этом уменьшилась скорость вагонетки?
- 2) (692). Какую работу совершает поле при перемещении заряда 20 нКл из точки с потенциалом 700 В в точку с потенциалом 200 В? Из точки с потенциалом –100 В в точку с потенциалом 400 В?

Таблица 2.2 – Соответствие дидактических единиц КИМ

КИМ	СССС
349-692	[«механика», «электродинамика»]
654	[«закон Кулона»]
712	[«электроёмкость»]
725	[«электроёмкость», «конденсаторы»]

734	[«электроёмкость», «конденсаторы», «энергия электрического поля»]
-----	---

Таблица 2.3 – Дуги «часть-целое»

	349-692	654	712	725	735
349-692	0	0	0	0	0
654	1	0	0	0	0
712	2	0	0	0	0
725	3	0	0	0	0
734	4	0	0	0	0

Таблица 2.4 – Отношения между дидактическими единицами КИМ 725 и 349-692

725→349-692	«механика»	«электродинамика»
«электроёмкость»	0	1
«конденсаторы»	0	1

Таблица 2.5 – Рёбра «племянники»

	349-692	654	712	725	735
349-692	0	4	4	4	4
654	4	0	2	2	2
712	4	2	0	2	2
725	4	2	2	0	2
734	4	2	2	2	0

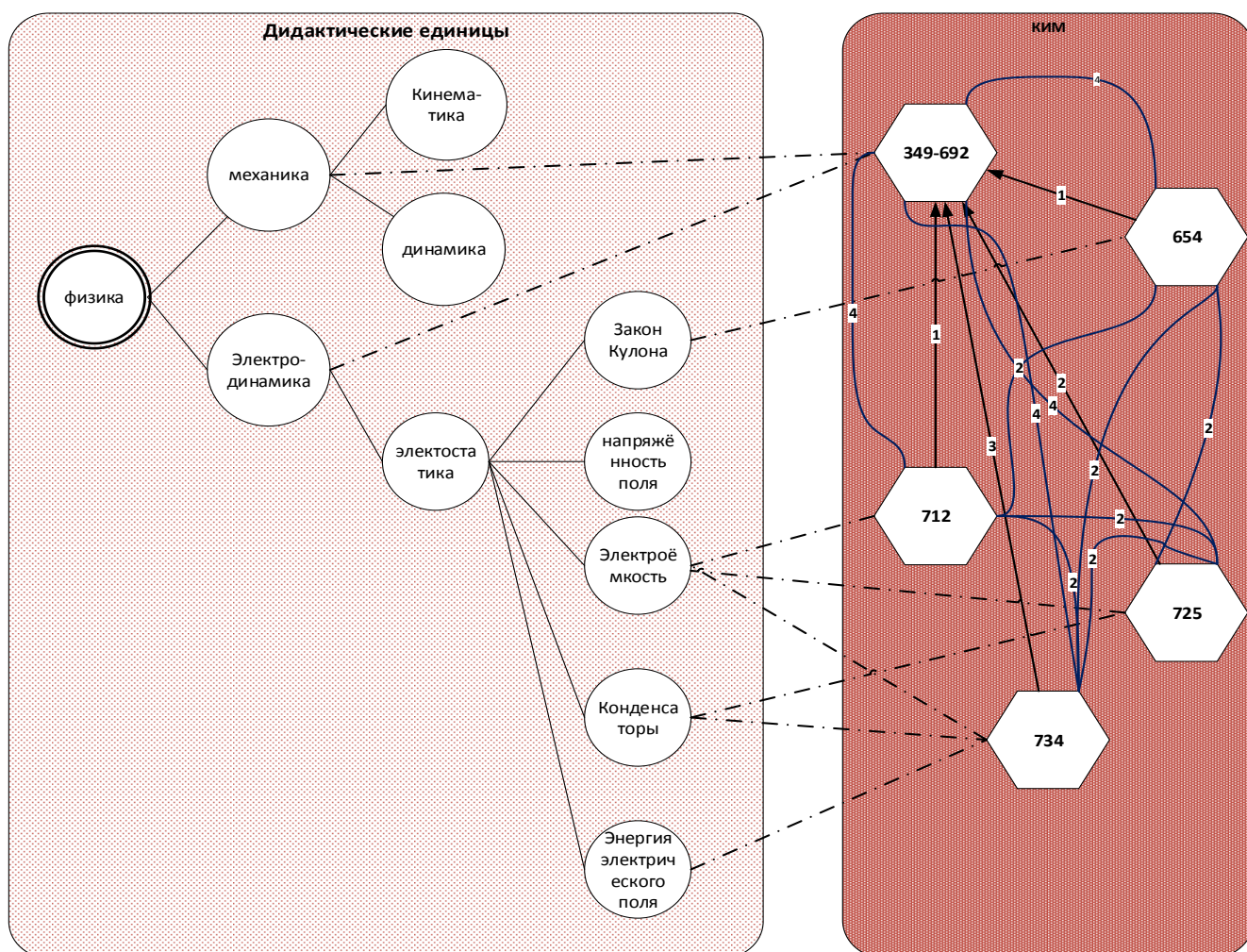


Рисунок 2.1. Пример графа взаимосвязи КИМ

2.6 Модель открытой экспертизы КИМ

Известно, что наиболее развита область экспертизы КИМ, представленных в тестовой форме. Экспертиза тестовых заданий традиционно проводится с привлечением группы экспертов и состоит из одного или нескольких этапов [13, 106], во время которых проверяется содержание тестовых заданий, оценивается соответствие тестовых заданий целям и задачам предстоящего тестирования, назначаются экспертные оценки. Привлечение сетевых компьютерных технологий существенно упростило разделение этапов, доставку заданий экспертам, получение отклика от экспертов и агрегацию данных, а также позволило иметь распределённые команды экспертов и организовывать экспертизы дистанционно [13].

Решение, принятое для КИМРА, состоит в совмещении традиционного подхода к экспертизе тестовых заданий с принципами открытого взаимного оценивания [*open peer review*], предлагаемого открытой наукой [*open science*] и его обобщение для произвольных типов КИМ. Далее экспертизу, поддерживаемую КИМРА, будем называть открытой экспертизой КИМ (ОЭ КИМ). Рисунок 2.2 формально описывает процессы ОЭ.

Цели ОЭ КИМ является повышение содержательного качества КИМ, применяемых для организации контрольно-измерительных мероприятий, снижение накладных расходов на формирование банка КИМ и повышение его качества.

Ниже перечислены принципы ОЭ КИМ.

1. Экспертиза инициализируется разработчиком КИМ и согласована с принципами *RFC* [*Request For Comments*], используемыми рабочими группами *IETF* [174].
2. Созданное и участвующее в ОЭ тестовое задание доступно любому пользователю КИМРА.
3. Комментарии к тестовым заданиям в рамках ОЭ могут оставлять любые подписчики системы.
4. Возможность выбора режимов просмотра КИМ:
 - а. При этом организуется стандартная доставка КИМ с отображением 1:1 (один КИМ для одного студента).

Возможность организации ОЭ КИМ в режиме выполнения КИМ позволяет интегрировать ОЭ КИМ в учебный процесс (например, ей может быть отведено отдельное место при проектировании учебной дисциплины) в качестве самооценивания [*self-assessment*] студентов. Эффективность самооценивания в онлайн и смешанных дисциплинах показана в работах [175, 176, 177]. При использовании ОЭ КИМ в учебном процессе автоматизируется получение ИОС от студентов к преподавателю. Распространение тестовых заданий, участвующих в ОЭ КИМРА, регламентируется лицензией *Creative Commons* [178]. При этом авторы не несут ответственности за качество заданий, не прошедших экспертизу.

В общем случае для программно-поддерживаемой ОЭ определены следующие роли участников процесса: подписчик – любой заинтересованный в КИМ пользователь (студент, член профессионального сообщества и т.п.); эксперт – пользователь, компетентный в области, которой принадлежит КИМ; разработчик КИМ – любой пользователь, заинтересованный в создании КИМ. Заметим, что ОЭ не накладывает ограничений на обязательную принадлежность указанных пользователей одному профессиональному сообществу, если такое ограничение необходимо, то оно регламентируется процедурно.

Различия между Подписчиком и Рецензентом обусловлены тегами статусов, доступными для назначения КИМ. Стандартный набор тегов эксперта определяется реализацией ОЭ и может включать следующие: *экспертиза пройдена*, *требуется доработка содержания КИМ*, *неверна ДЕ КИМ*, *отсутствуют ДЕ КИМ*, *не определены правила оценивания КИМ* и проч. Теги подписчика не используются для поддержки экспертизы КИМ, и появляются только контексте ОЭ, тем не менее, они могут подвергаться фолксономическому моделированию, анализироваться и использоваться для оптимизации ОЭ.

Отметим, что для подписчика ОЭ обеспечивает не только участие в ЖЦ КИМ, но и дополнительный учебный процесс самоподготовки Подписчиков в процессе ОЭ КИМ. Формирование наборов КИМ для ОЭ являются частными случаями задач конструирования композитного КИМ и доставки КИМ.

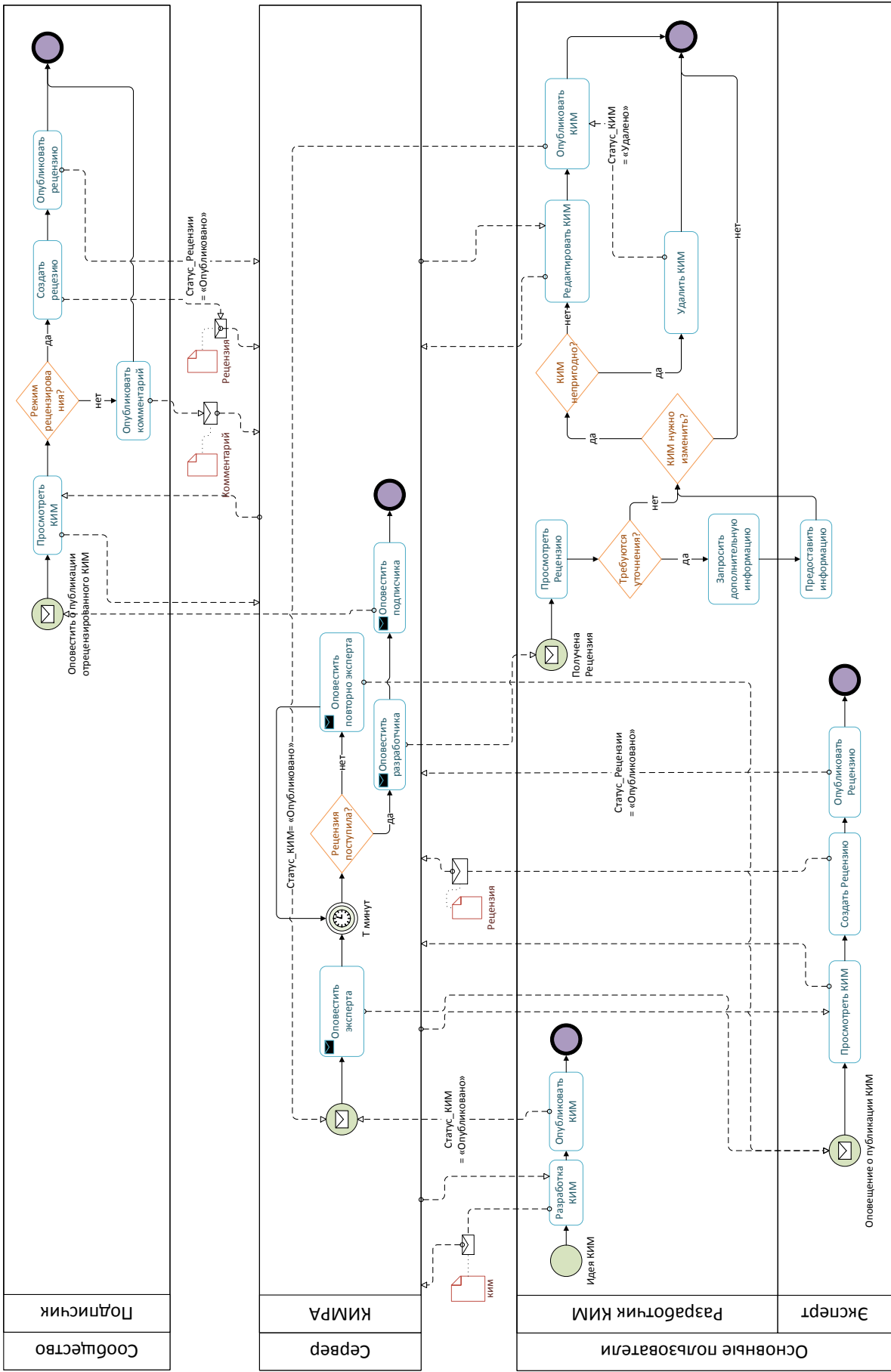


Рисунок 2.2. BPMN-диаграмма открытой экспертизы КИМ

2.7 Шкалирование

В данном разделе приведена справочная информация о шкалах измерений и их типах. Как уже было сказано выше, измерения в социологических науках были определены на базе достаточно проработанной на тот момент теории физических измерений. Поскольку для социальных наук соответствующий непротиворечивый глоссарий не сформирован, автор использует для данного раздела материалы из рекомендаций по межгосударственной стандартизации, метрологии и сертификации «РМГ 83-2007. Шкалы измерения. Термины и определения» [88] и работы С. Стивенса «*On the theory of scales of measurement*» [25].

Шкала (измерений) – отображение множества различных проявлений количественного или качественного свойства на принятое по соглашению упорядоченное множество чисел или другую систему логически связанных знаков (обозначений).

2.7.1 Типы шкал измерений

Под типом шкалы понимают специфический набор признаков, классифицирующий данную шкалу измерений и характеризующий совокупность присущих ей логических соотношений между различными проявлениями измеряемого свойства.

Шкала наименований (номинальная шкала). Шкала измерения качественного свойства, характеризующаяся только соотношениями эквивалентности или отличиями проявления этого свойства. Особенности:

- 1) неприменимы понятия «нуль», «единица измерения» и «размерность»;
- 2) допустимы только изоморфные и гомоморфные преобразования;
- 3) недопустимо изменение спецификаций, описывающих конкретные шкалы.

Шкала порядка (ранговая шкала). Шкала измерений количественного свойства (величины), характеризующаяся соотношениями эквивалентности и порядка по возрастанию (убыванию) различных проявлений свойства. Особенности:

- 1) неприменимы понятия «единица измерения» и «размерность»;
- 2) необязательность наличия нуля;
- 3) допустимы любые монотонные преобразования;

- 4) недопустимо изменение спецификаций, описывающих конкретные шкалы.

Шкала разностей (интервальная шкала). Шкала измерений количественного свойства (величины), характеризующаяся соотношениями эквивалентности, порядка, пропорциональности (допускающими в ряде случаев операцию суммирования) различных проявлений свойства. Особенности:

- 1) устанавливаемые по соглашению «нуль» и «единица измерений»;
- 2) допустимо применение понятия размерность;
- 3) допустимы линейные преобразования;
- 4) допустимо изменение спецификаций, описывающих конкретные шкалы.

Шкала отношений (абсолютная шкала). Шкала измерений количественного свойства (величины), характеризующаяся соотношениями эквивалентности, порядка, пропорциональности (в ряде случаев допускающими суммирование) различных проявлений свойства. Шкалы отношений, в которых не имеет смысла операция суммирования, называют *пропорциональными шкалами отношений* или шкалами отношений 1-го рода. Если операция суммирования имеет смысл, говорят об *аддитивных шкалах отношений* или шкалах отношений 2-го рода. Особенности:

- 1) наличие «естественного нуля» и устанавливаемой по соглашению «единицы измерений»;
- 2) применимо понятие размерности;
- 3) допустимы линейные преобразования;
- 4) допустимо изменение спецификаций, описывающих конкретные шкалы.

2.8 Выводы и результаты по главе

В главе приведён обзор математических моделей наиболее популярных теорий тестирования: классической теории тестирования (*СТТ*), современной теории тестирования (*ИРТ*) и теории тестлетов (*ТРТ*). В результате установлено, что *СТТ* и *ИРТ* не накладывают существенных ограничений на форму представления данных в хранилище ПСУН поддержки ЖЦ КИМ, так как их характеристики – наборы числовых значений, выражающих особенности тестов и тестовых заданий. В то же

время атрибуты моделей *TRT* и других моделей, применяемых для нужд *CAT*, используются не только для анализа результатов, но и для управления процессами доставки и проведения тестирования. В силу этого особенности *CAT* накладывают ограничения на модель КИМ, так как у средства проведения *CAT* должно быть достаточно данных для решения задач сегментации пула заданий и подбора задач в процессе проведения тестирования. Поскольку основной целью КИМРА не является поддержка процессов проведения *CAT*, задачи селекции задания при адаптивном тестировании ложатся на внешнюю систему, возможность интеграции с которой учитывается в архитектуре КИМРА.

На основе выделенных требований к атрибутам КИМ предложена математическая модель КИМ, параметры которой отражают этапы ЖЦ КИМ и обеспечивают целостность процессов РИСУН поддержки ЖЦ КИМ.

Обобщение задач сегментации и селекции КИМ с понятием дидактических единиц позволило предложить графовую модель связи КИМ с результатами обучения и учебными материалами (граф структуры содержания КИМ). По предложенной модели построена вторая графовая модель, отражающая взаимосвязи между КИМ (граф взаимосвязей КИМ). Численные методы, оптимизирующие процесс конструирования композитных КИМ предложены и подробно описаны в главе 3.

Детали реализации указанных моделей в информационных системах обсуждаются в главе 4.

ГЛАВА 3. МЕТОДЫ И АЛГОРИТМЫ

В главе вводится понятие композиции КИМ и ставятся задачи оптимизации состава композитного КИМ, формализованного моделью 2.29. Для построения оптимального композитного КИМ поставлена задача дискретной оптимизации и предложен реализованный в КИМРА интерактивный алгоритм, основанный на жадном алгоритме и методе динамического программирования.

Формализовано понятие компьютерной доставки КИМ. Предложен модифицированный алгоритм генерации беспорядков, использующийся в модуле взаимного оценивания системы КИМРА при доставке КИМ в рамках взаимного оценивания: процедура распределения (рандомизации) подач по рецензентам.

3.1 Композиция КИМ

При организации и проведении измерений в образовании существенным и персистентным является вопрос конструирования из КИМ композитных КИМ, например, объединение задач в вариант контрольной работы, тестовых заданий в тесты (примеры см. в разделе 3.1.1), тестов в батарее тестов [*test batteries*] и т.п.

Задачи генерации композитных КИМ возникают не только при организации контрольно-измерительных мероприятий, но и в процессе проведения экспертизы, ОЭ, при самоподготовке. Это означает, что ПСУН поддержки ЖЦ КИМ должны реализовывать методы и алгоритмы генерации композитных КИМ и допускать расширение соответствующими модулями.

Поставим задачу оптимизации состава композитного КИМ.

Формулировка. Подобрать оптимальный набор КИМ из банка КИМ для создания композитного КИМ с учётом набора критериев и ограничений. Очевидно, число критериев может превышать один, что позволяет отнести задачу к многокритериальной оптимизации. Задача осложняется тем, что в общем виде нас интересует не множество КИМ, а последовательность или более сложная структура.

Примером может служить построение *теста* на основе банка *тестовых заданий*, где задания могут образовывать априори заданную последовательность, а могут быть представлены деревом для адаптивного тестирования.

3.1.1 Сборка компьютерных тестов

В области CAT широко освещены задачи, связанные с генерацией теста, как композитного КИМ, в процессе предоставления студенту. Отметим наиболее распространённые критерии отбора заданий.

3.1.1.1 Критерии отбора заданий для CAT

Критерий максимальной информации [*maximum information criterion*] [31] на основе критерия выбирается k -е задание, так чтобы максимизировать ожидаемое количество информации Фишера $I_{U_{i_1 \dots U_{i_{k-1}}}(\theta) = E \left[J_{U_{i_1 \dots U_{i_{k-1}}}(\theta) \right]$ на $\hat{\theta}_{u_{i_1}, \dots, u_{i_{k-1}}}$, то есть

$$i_k \equiv \arg \max \left\{ I_{U_j} \left(\hat{\theta}_{u_{i_1}, \dots, u_{i_{k-1}}} \right) : j \in R_k \right\} \quad (3.1)$$

Согласно процедуре Оуэна i -е задание выбирается в соответствии с условием [179]:

$$|b_{i_k} - E(\theta | u_{i_1}, \dots, u_{i_{k-1}})| < \delta, \text{ где } \delta \geq 0 \quad (3.2)$$

$$\text{где } \hat{\theta}_{u_{i_1}, \dots, u_{i_{k-1}}}^{EAP} = \int \theta g(\theta | u_{i_1}, \dots, u_{i_{k-1}}) d\theta$$

Постериорное распределение для θ задаётся:

$$g(\theta | u_{i_1}, \dots, u_{i_{k-1}}) = \frac{L(\theta | u_{i_1}, \dots, u_{i_{k-1}}) g(\theta)}{\int L(\theta | u_{i_1}, \dots, u_{i_{k-1}}) g(\theta) d\theta} \quad (3.3)$$

Критерий максимума общей информации [180] основан на использовании меры Кульбака-Лейблера, значение которой в условиях модели 3PL для распределения ответов k -го задания при истинном значении способности (θ_0) студента и её оценки ($\hat{\theta}_{k-1}$) вычисляется как:

$$K_{i_k}(\hat{\theta}_{k-1}, \theta_0) = E \left[\log \frac{L(\theta_0 | U_{i_k})}{L(\hat{\theta}_{k-1} | U_{i_k})} \right] \quad (3.4)$$

Поскольку значение θ_0 неизвестно, используется её оценка $[\hat{\theta}_{k-1} - \delta_k, \hat{\theta}_{k-1} + \delta_k]$, где δ_k – убывающая функция рангов заданий в адаптивном тесте. Задание k выбирается согласно критерию:

$$i_k \equiv \arg \max \left\{ \int_{\hat{\theta}_{k-1}-\delta_k}^{\hat{\theta}_{k-1}+\delta_k} K_{i_k}(\hat{\theta}_{k-1}, \theta) d\theta : j \in R_k \right\} \quad (3.5)$$

Критерий взвешенного максимального правдоподобия [*likelihood-weighted information criterion*] является модификацией критерия максимума общей информации. В работе [181] предложено взвешивать информацию Фишера при помощи функции правдоподобия и выбирать k -е задание в тест по критерию:

$$i_k \equiv \arg \max \left\{ \int_{-\infty}^{+\infty} L(\theta | u_{i_1}, \dots, u_{i_{k-1}}) I_{i_k} d\theta : j \in R_k \right\} \quad (3.6)$$

Полностью байесовские критерии от предшествующих отличаются использованием постериорного распределения для θ . Аналог критерия максимума общей информации определяется как:

$$i_k \equiv \arg \max \left\{ \int I_{U_j}(\theta) g(\theta | u_{i_1}, \dots, u_{i_{k-1}}) d\theta : j \in R_k \right\} \quad (3.7)$$

Более современные байесовские критерии [77] основаны на предварительном анализе, они прогнозируют распределение ответов на оставшихся в хранилище заданиях $i \in R_k$ после того, как $k - 1$ задание уже были выбраны в тест. Важным элементов такого анализа является предсказание распределения для ответа на задании i , которое определяется как функция вероятности:

$$p(u_i | u_{i_1}, \dots, u_{i_{k-1}}) = \int p(u_i | \theta) g(\theta | u_{i_1}, \dots, u_{i_{k-1}}) d\theta \quad (3.8)$$

Критерий выбора k -го задания определяют, как:

$$i_k \equiv \arg \max \left\{ p_j(0 | u_{i_1}, \dots, u_{i_{k-1}}) J_{U_{i_1} \dots U_{i_{k-1}}}, U_j = 0 \left(\hat{\theta}_{u_{i_1}, \dots, u_{i_{k-1}}}, U_j = 0 \right) + \right. \\ \left. p_j(1 | u_{i_1}, \dots, u_{i_{k-1}}) J_{U_{i_1} \dots U_{i_{k-1}}}, U_j = 1 \left(\hat{\theta}_{u_{i_1}, \dots, u_{i_{k-1}}}, U_j = 1 \right) : j \in R_k \right\}, \quad (3.9)$$

где $p_j(0 | u_{i_1}, \dots, u_{i_{k-1}})$ – вероятность неверного ответа.

Также известны критерии, совершенствующие уже существующие и построенные на моделях, учитывающих дополнительные характеристики, например, время выполнения [*response time, RT*] заданий или специализированные для малых выборок и снижающие ошибки определения способности студента решить выбранное задание [40, 182].

3.2 Задача оптимальной композиции КИМ с учётом таксономии дидактических единиц

В процессе работы над диссертацией автором не было обнаружено задокументированных алгоритмов, позволяющих конструировать композитный КИМ по набору критериев и ограничений. Наиболее распространены 1) случайная генерация набора КИМ, при которой не гарантируется, например, покрытие ДЕ при помощи КИМ; 2) конструирование композитного КИМ по жёсткой спецификации и гарантии, что необходимые КИМ присутствуют в БКИМ. Высокоуровневые алгоритмы, в том числе использующие онтологии, являются промышленными ноу-хау и не публикуются в виде пригодном для воспроизведения или осмысленного развития [183].

Задача оптимальной композиции КИМ относится к задачам дискретной оптимизации [184]. Предположим, что для создания нового композитного КИМ $A = \langle TaskDesc_A, \langle CEAM_A, Assignment_A, Context_A, GR_A \rangle, CCCS_A \rangle$ необходимо выбрать оптимальное множество $CEAM_A \subseteq S_{EAM}$ в соответствии с последовательностью критериев $Criteria$ (номер критерия в последовательности задаёт его приоритет) и не нарушающим набор ограничений $Constraints$. Пусть задано начальное целевое множество ДЕ $CCCS_A^* \subseteq S_{CCC}$. Введём обозначение $CCCS_{comp} = \bigcup_{B \in CEAM_A} CCCS(B)$, чтобы удобнее ссылаться на объединение ДЕ всех КИМ, содержащихся в A .

Критериями, входящими в последовательность $Criteria$ могут быть следующие.

1. Критерий точности сопоставления результатов обучения $M_{CCC_Acc} = \max_{CCCS_{comp}} \left(\frac{|CCCS_A^* \cap CCCS_{comp}|}{|CCCS_{comp}|} \right)$. Чем ближе этот критерий к 1, тем точнее композиция КИМ оценивает целевое множество ДЕ, а чем ниже его значение, тем большее число дополнительных ДЕ затрагивается.
2. Критерий приближения к заданному общему времени T выполнения композитного КИМ: $\min_{CEAM_A} \left| \sum_{B \in CEAM_A} t(B) - T \right|$.

3. Критерий приближения к планируемому количеству L атомарных КИМ:

$$\min_{CEAM_A} ||CEAM_A| - L|.$$

Ограничениями, входящими в *Constraints*, могут служить следующие данные и условия на них.

1. Ограничение на тип задания $Type$, задаваемое множеством $ATypes$:
 $Type_A \in ATypes$.
2. Ограничение на логические атрибуты $Tags$, задаваемое множеством $ATags$: $\forall x \in Tags_A (x \in ATags)$.
3. Невозможность расширения набора ссылок на элементы описания результатов обучения по сравнению с $CCCS_A^*$ целевых ДЕ: $\forall x \in CCCS_{comp} (x \in CCCS_A^*)$.
4. Вектор четвёрок $\langle APartSize_i, APartCCCS_i, APartType_i, APartTags_i \rangle$, где $APartSize$ – относительный размер -го раздела ($\sum_i APartSize_i = 1$) композитного КИМ; $APartCCCS$ – множество ДЕ раздела, возможно, пустое; $APartType$ – тип КИМ раздела; $APartTags$ – множество логических атрибутов КИМ раздела, возможно, пустое.
5. Количество КИМ в $CEAM_A$.
 - а. Общее количество КИМ $ATotal_{min} \leq ATotal \leq ATotal_{max}$.
 - б. Количество КИМ, отобранных по признаку (например, трудность $AP_{min} \leq AP \leq AP_{max}$, дискриминативность $AD_{min} \leq AD \leq AD_{max}$ и проч.).
6. Общее время решения $ASolveTime_{min} \leq ASolveTime \leq ASolveTime_{max}$ композитного КИМ. Отметим, что сумма времён заданий не обязательно равна общему времени, которое может быть, как больше (что понижает общую сложность), так и меньше.
7. Среднее время решения отдельных КИМ $ItemSolveTime_{min} \leq \frac{\sum_i ASolveTime_i}{|CEAM_A|} \leq ItemSolveTime_{max}$.

8. Правила траектории прохождения композитного КИМ (эти правила задаются на этапе конструирования КИМ или на стороне клиента). Например, для моделей SAT почти вся селекция построена рекуррентно, то есть на знаниях о том, как пройдены задания до выбираемого¹.

Расширение постановки задачи оптимальной композиции КИМ для случая композиции набора из *ParVariants* параллельных вариантов КИМ. В отличие от создания одного композитного КИМ A , теперь нужно создать множество композитных КИМ $A_i = \langle TaskDesc_{A_i}, \langle CEAM_{A_i}, Assignment_{A_i}, Context_{A_i}, GR_{A_i} \rangle, CCCS_{A_i} \rangle$, где $i \in [1; ParVariants]$ и где множество критериев расширено критерием максимального сходства КИМ A_i .

Заметим, что конструирование теста как композитного КИМ полностью укладывается в предложенную постановку. Для стандартизованных и критериально-ориентированных тестов выполнены все требования, накладываемые определениям в [185, 95] за исключением наличия сразу после конструирования характеристик, полученных в результате апробаций на репрезентативных выборках участников.

3.2.1 Интерактивный алгоритм оптимальной композиции КИМ

Многокритериальная задача оптимизации композитного КИМ решается в различных ситуациях, многие из которых характеризуются высокой неопределённостью параметров. Отсюда запрос на инструмент, позволяющий преподавателю/эксперту в интерактивном режиме создать композитный КИМ, требования к которому уточняются в процессе создания. В основе интерактивного алгоритма оптимальной композиции КИМ (рисунок 3.1) лежат два подхода: быстрый приближённый *жадный алгоритм*, позволяющий последовательно уточнить выбор $CEAM_A$ по следующему критерию в последовательности; и *алгоритм на основе ме-*

¹ В данной работе не рассматривается, но должно учитываться на этапе постановки задачи и сбора требований.

тогда динамического программирования, позволяющий получить глобально-оптимальное решение по одному из критериев, переведя остальные критерии в ограничения.

Промежуточные результаты работы алгоритма визуализируются вместе с обобщающей характеристикой, что позволяет преподавателю точнее понять возможности получения решения при текущих ограничениях.

Особо значимым является предпросмотр параллельных вариантов композиции, поскольку, например, задача одновременного выравнивания уровня сложности и тематики заданий до сих пор требует экспертного мнения (за исключением случая наличия фасетов для всех *тестовых заданий*, используемых при генерации *теста*).

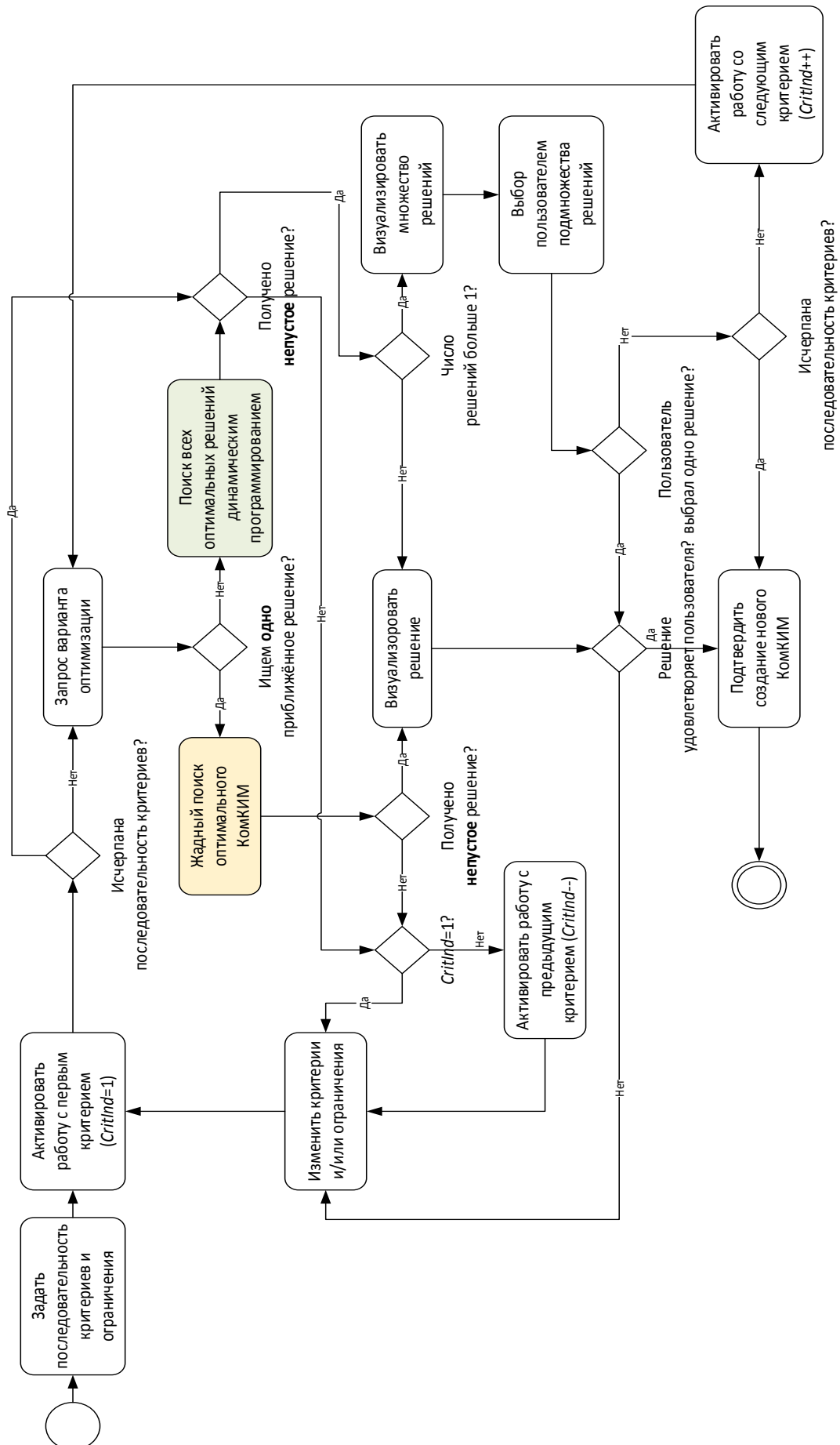


Рисунок 3.1. Общая схема интерактивного алгоритма композиции КИМ

В предложенном интерактивном алгоритме используются реализации жадного алгоритма [186] и современная реализация техники динамического программирования [187, 188].

3.3 Доставка КИМ

Этап доставки [*delivery*] учебного материала студентам предполагает разнообразие подходов. Методы доставки неразрывно связаны с форматом дисциплины и с инфраструктурой поддержки учебного процесса.

Исследования показывают (см. обзор в [189]), что доставка учебных материалов, в том числе и КИМ с использованием информационно-телекоммуникационных технологий (далее – *компьютерная доставка*) оказывается предпочтительнее как по точности и качеству предоставления учебных материалов студентам, так и по эффективности из-за возможности журналирования и повторного использования, практически неограниченной масштабируемости, воспроизводимости и гибкости. Компьютерные способы доставки КИМ существенно снижают время, затрачиваемое на компоновку контрольно-измерительных мероприятий, требования к персоналу и накладные расходы. Отметим, что в данной работе компьютерная доставка КИМ понимается шире, чем представление КИМ клиентам (студентам) посредством устройств вывода клиентских терминалов.

Дадим определение компьютерной доставки КИМ, базируясь на [190, 191] и др., как совокупности процедур и используемым в этих процедурах программно-аппаратных средств, обеспечивающих: 1) отображение выбранного подмножества КИМ на априори заданное подмножество студентов; 2) надёжную и безопасную передачу КИМ в виде, ориентированном на проведение контрольно-измерительного мероприятия, на клиентские терминалы; 3) организацию контролируемой среды проведения контрольно-измерительного мероприятия; 4) сбор результатов; 5) полное журналирование процесса.

Такое определение позволяет в самом общем виде описать основные требования к доставке КИМ, включая требования к масштабированию, надёжности, информационной безопасности, работе с персональными данными, параметризации, конфигурации, автоматизации и другим аспектам.

3.3.1 Доставка КИМ при компьютерном взаимном оценивании

3.3.1.1 Основные понятия и определения

Автором данной работы совместно с к.т.н., доц. А.А. Незнановым предложена русскоязычная терминология для процесса взаимного оценивания. В данном разделе продублированы определения, необходимые для понимания процесса доставки КИМ. Полный перечень определений приведён в разделе 4.9.2.1.

Взаимное оценивание [*peer assessment (PA)*] – процедура оценивания, организованная в форме рандомизированного взаимного рецензирования [*peer review (PR)*] артефактов, трактуемых как результат выполнения учебного задания с предварительно формализованной схемой оценивания.

В общем случае, артефакт [*artifact*] – сущность, созданная человеком с определённой целью. В контексте взаимного оценивания артефакт – сущность, созданная студентом с целью получить рецензию.

Для программного средства поддержки взаимного оценивания уместно выделить следующие роли пользователей:

- автор [*initial author*] – студент, зарегистрированный в качестве будущего отправителя артефакта в текущей сессии взаимного оценивания (СВО);
- податель [*submitter*] – студент, направивший артефакт на рецензирование в рамках СВО;
- рецензент [*reviewer*] – студент, отославший валидный оценочный лист в рамках СВО.

Элементы данных, фигурирующие в каждой СВО:

- оценочный лист [*PR form*] – таблица, специфицирующая поля рецензии для конкретного типа артефакта и конкретного задания; содержит понятные рецензенту описания полей, примеры заполнения и правила валидации значений полей;
- подача [*submission*] – артефакт, направленный на рецензирование конкретным подателем;
- рецензия [*review*] – заполненный оценочный лист, отосланный рецензентом.

Наиболее трудоёмким процессом при реализации взаимного оценивания студентами друг друга является процедура распределение (рандомизации) подач по рецензентам.

3.3.1.2 Постановка задачи

Рассмотрим простейшую постановку задачи взаимного оценивания, когда множества подателей и рецензентов совпадают. Обозначим это множество как *Authors*. Пусть также число рецензий на каждую работу равно *PrA_Count* (число пиров на одну работу).

3.3.1.3 Определения

Перестановка [*permutation*] – взаимно однозначное отображение $f: X \leftrightarrow X$.

Число всех возможных перестановок обозначается $P_n = n!$ (n факториал).

Факториал $n! = \prod_{k=1}^n k$. $0! \stackrel{\text{def}}{=} 1$.

Неподвижная точка [*fixed point*] перестановки – любой элемент множества $\{x \in X | f(x) = x\}$, которое представляет собой дополнение носителя перестановки в X .

Беспорядок [*derangement*] – перестановка без неподвижных точек, то есть такая, что никакой элемент не отображается на себя ($f(x) \neq x$).

Число всех возможных беспорядков обозначается $D_n = !n = \sum_{k=0}^n (-1)^k \frac{n!}{k!}$. Это число часто называют субфакториал.

3.3.1.4 Математическая формулировка задачи

Дано множество *Authors* идентификаторов авторов. Требуется сгенерировать *PpA_Count* случайных беспорядков DR_i ($i \in [1..PpA_Count]$) множества *Authors* таким образом, чтобы для любого объекта $x \in Authors$ соответствующие элементы всех беспорядков были различны, то есть $(DR_i(x) \neq x) \& (\forall i, j \in [1..PpA_Count](DR_i(x) \neq DR_j(x)))$.

Таблица 3.1 содержит количество различных беспорядков и перестановок для множеств с мощностями от 0 до 10. Таблица 3.1 показывает, что беспорядков заметно меньше, чем перестановок с неподвижными точками, но при $|Authors| \geq 6$ их достаточно много, чтобы пробовать алгоритмы генерации случайных беспорядков.

Таблица 3.1 – Числа перестановок и беспорядков

<i>Authors</i>	Число перестановок	Число беспорядков
0	1	1
1	1	0
2	2	1
3	6	2
4	24	9
5	120	44
6	720	265
7	5040	854
8	40320	14833
9	362880	133496
10	3628800	1334961

3.3.1.5 Идеи решения

Поскольку нам необходимо сгенерировать лишь столько беспорядков, сколько рецензий должна получить одна работа (рисунок 3.2), то использовать методы конструктивного перечисления оказывается неэффективно. Подробно о конструктивном перечислении беспорядков рассказано в работе Ж.-Л. Барила и В. Вайновски [192].

	A	B	C	D	E	F	G	H	I	J	K
31	500683	187472	663613	165129							
32	910242	663613	165129	912500							
33	636879	165129	912500	196907							
34	334772	912500	196907	674937							
35	276439	196907	674937	851770							
36	139150	674937	851770	497369							
37	306626	851770	497369	625376							
38	481122	497369	625376	849698							
39	795616	625376	849698	633202							
40	431936	849698	633202	444270							
41	402316	633202	444270	138696							
42	196907	444270	138696	108299							
43	964689	138696	108299	500683							
44	400423	108299	500683	910242							
45	194732	500683	910242	334772							
46	151031	910242	334772	880420							
47	633202	334772	880420	976427							
48	327233	880420	976427	372173							
49	242254	976427	372173	400423							
50	749602	372173	400423	900607							

Рисунок 3.2. Результат рандомизации

Действительно, нам необходимо небольшое число «наиболее случайных беспорядков». В работе [193] приведён анализ достаточно простого алгоритма порождения случайного беспорядка и доказано, что, начиная с некоторой мощности исходного множества, проверки на выдачу генератором случайных чисел фиксированной точки не влияют на асимптотическую оценку временной вычислительной сложности, которая остаётся линейной, как у алгоритма, предложенного Р. Дурштенфельдом [194].

Поскольку цель генерации беспорядков в рамках КИМРА – это формирование отображения множества подач на множество рецензентов при взаимном оценивании, предлагается модификация алгоритма [193] учитывающая, что элементы каждого следующего беспорядка проверяются на несовпадение с элементами, стоящими в тех же позициях, в предыдущих беспорядках.

3.3.1.6 Алгоритм

Так как тестовая версия КИМРА разрабатывалась для внедрения в очные курсы, авторы и разработчики не стали заниматься дополнительной низкоуровневой оптимизацией, ожидая, что число участников взаимного оценивания не будет превосходить 10000.

Основой реализации является процедура *GenReMapOtherDR* порождения следующего беспорядка исходного множества чисел $[0..AuthorsCount - 1]$. При порождении элементы этого беспорядка проверяются на несовпадение с элементами предыдущих беспорядков с использованием функции *CheckPrevMaps*. Поскольку необходимо рандомизировать идентификаторы авторов, то после генерации беспорядков номера элементов беспорядков заменяются на соответствующие идентификаторы.

Далее алгоритмы записаны в псевдокоде, предложенном в книге [195], за исключением того, что типы данных указываем через двоеточие справа от имени параметров процедур и функций, переменные i, j, k целочисленные и локальны внутри процедур и функций.

- ▷ Глобальные структуры данных:
 - ▷ Массив целочисленных массивов *ReMaps* (набор беспорядков)
 - ▷ – Первый индекс – номер беспорядка (0 – само исходное множество)
 - ▷ – Второй индекс – номер элемента беспорядка
 - ▷ Целочисленный индекс текущего беспорядка *ACurDerangement*
 - ▷ Целочисленный элемент исходного множества *ARow*
 - ▷ Сгенерированный целочисленный элемент беспорядка *Avalue*
 - ▷ Целочисленный массив идентификаторов объектов (множество *Authors*) *AAutArr*
 - ▷ Целочисленное количество рецензентов одного артефакта *APeerCount*
 - ▷ Проверка кандидата в элемент беспорядка на наличие в предыдущих беспорядках
 - ▷ Возвращает результат проверки ()
- ```

func CheckPrevMaps(ReMaps, ACurDerangement, ARow, Avalue): Boolean
begin
 Result $\leftarrow$ False
 for $i \leftarrow 0$ to ACurDerangement $- 1$ begin
 if ReMaps[ARow, i] = Avalue then begin
 Result $\leftarrow$ True
 break
 end
 end ▷ i
 return Result
end func

```

- ▷ Генератор очередного беспорядка с номером *ACurDerangement*
- ▷ локальные переменные процедуры:
- ▷ целочисленные *p, tmp*
- ▷ логические *IsFound, IsFixed, IsOver*

```

proc GenReMapOtherDR(ReMaps, ACurDerangement)
begin
 IsFound $\leftarrow$ False
 while not IsFound do begin
 for j $\leftarrow$ 0 to High(ReMaps) do begin
 ReMaps[j, ACurDerangement] $\leftarrow$ j
 end; ▷ j
 j $\leftarrow$ High(ReMaps)
 IsFixed $\leftarrow$ False
 IsOver $\leftarrow$ False
 while not IsOver do begin
 p $\leftarrow$ Int((j) * Rnd)
 Tmp $\leftarrow$ ReMaps[p, ACurDerangement]
 if (Tmp = j) or
 CheckPrevMaps(ReMaps, ACurDerangement, j, Tmp) then begin
 IsFixed $\leftarrow$ True
 IsOver $\leftarrow$ True
 end else
 обменять ReMaps[p, ACurDerangement] $\leftrightarrow$ ReMaps[j, ACurDerangement]
 end
 j $\leftarrow$ j - 1
 if j = 0 then begin
 IsOver $\leftarrow$ True
 end
 end ▷ IsOver
 Tmp $\leftarrow$ ReMaps[0, ACurDerangement]
 if not IsFixed and (Tmp $\neq$ 0) and
 not CheckPrevMaps(ReMaps, ACurDerangement, 0, Tmp) than begin
 IsFound $\leftarrow$ True
 end
 end ▷ IsFound
end proc

```

- ▷ локальные переменные процедуры AuthorsRandomization
- ▷ целочисленные *ArrCount, u*

```

proc AuthorsRandomization(AAutArr, APeerCount)
begin
 ArrCount $\leftarrow$ High(AAutArr) + 1
 if ArrCount < 6 then begin
 сообщить, что количество авторов меньше 6
 Exit
 end
 if ArrCount < APeerCount then begin
 сообщить, что количество авторов меньше, чем количество рецензетов
 Exit
 end

```

```

ReMaps[0] ← AAutArr ▷ Load first array (Identifiers of Authors)
▷ Переназначить рецензентов
for j = 0 to APeerCount – 1
 GenReMapOtherDR(ReMaps, j);
end; ▷ j количество рецензентов на каждый артефакт

for i ← 0 to High(ReMaps) do begin
 for j ← 0 to High(ReMaps[i]) do begin
 ReMaps[i, j] = AAutArr[ReMaps[i, j]]
 end
end
return ReMaps
end proc

```

### 3.3.1.7 Реализация

Алгоритм реализован на языках *Smart Pascal* и *Visual Basic for Applications*.

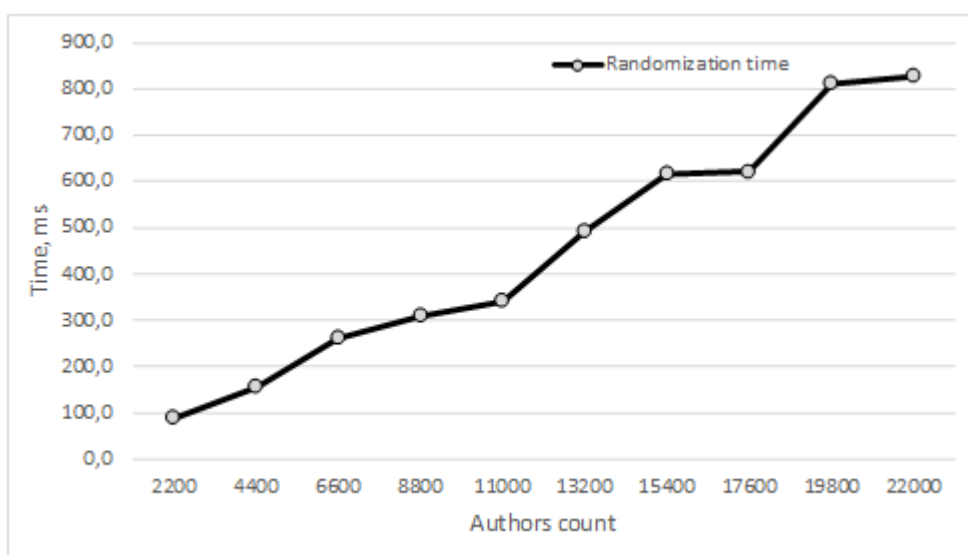
Отметим, что значения, которыми в итоге заменяются индексы в массивах *ReMaps* – это плохо различимые пользователем значения из диапазона [100000, 999999], которые выступают анонимизированными идентификаторами авторов (*Author\_Id*) в рамках СВО. Для каждого *Author\_Id* также формируется набор глобально-уникальных идентификаторов (*GUID*) для отслеживания подач и рецензий. 0 содержит полный код сценариев, реализующих алгоритм рандомизации.

### 3.3.1.8 Исследование вычислительной сложности

Исследуем время работы алгоритма рандомизации идентификаторов авторов работ для *PpA\_Count* = 4 с занесением в базу данных. Таблица 3.2 содержит усреднённое время по 4 прогонам теста для монотонно возрастающего количества авторов (*Authors count*). Рост времени работы проиллюстрирован графиком (рисунок 3.3). График соответствует ожиданиям и хорошо аппроксимируется линейной функцией.

Таблица 3.2 – Время работы алгоритма рандомизации

| Authors count | 2200 | 4400  | 6600  | 8800  | 11000 | 13200 | 15400 | 17600 | 19800 | 22000 |
|---------------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Time (ms)     | 88,8 | 156,5 | 260,6 | 309,9 | 341,6 | 491,2 | 617,3 | 621,6 | 812,0 | 828,4 |



**Рисунок 3.3. График роста времени работы**

### *3.3.1.9 Адаптивная доставка и оценка*

Рассмотренный вариант априорной рандомизации множества авторов является одним из самых простых. Намного более сложный случай состоит в учёте дополнительных ограничений для рецензентов (аффилированность, например, принадлежность одной группе или участие в одном проекте; недостаточный балл за предыдущие сессии; наличие подозрений на недобросовестность и т.п.), возможность изменения числа рецензентов конкретного артефакта и адаптивность схемы рандомизации к изменению состава участников с течением времени. При этом пока рассматривается рецензирование каждого артефакта независимо, что не является единственным вариантом организации рецензирования. Вариант качественных сравнительных оценок хорошо рассматривается в [196].

## **3.4 Выводы и результаты по главе**

В главе исследованы методы и алгоритмы сборки композитных КИМ и вопросы доставки КИМ.

Автором установлено, что наиболее проработанными являются численные методы конструирования компьютерных адаптивных тестов (как композиции тестовых заданий). Особенностью указанных методов является необходимость их реализации на стороне клиентского (по отношению к КИМРА) программного обеспечения, так конструирование адаптивного теста (как композитного КИМ) проис-

ходит в процессе прохождения теста. В то же время процессы ЖЦ КИМ предполагают необходимость конструирования композитных КИМ на разных этапах, относящихся не только к контрольно-измерительным мероприятиям, но и к ОЭ КИМ и самоподготовке студентов. Поэтому в данной главе поставлена задача оптимальной композиции КИМ, и предложен численный метод сборки композитных КИМ, основанный на графе связи КИМ с результатами обучения и графе связей КИМ.

Практика организации компьютерных форм измерений в образовании показывает, что процесс доставки КИМ серьёзно зависит от его типа *Type*. В главе обсуждаются особенности доставки КИМ в процессе взаимного оценивания, поставлена задача рандомизации рецензентов по рецензиям, предложен численный метод рандомизации, реализованный в модуле проведения взаимного оценивания программного комплекса КИМРА. О реализации и апробации модуля подробно рассказано в разделах 4.9.2 и 4.10.

## ГЛАВА 4. РЕАЛИЗАЦИЯ СИСТЕМЫ КИМРА

Глава посвящена особенностям программной реализации предложенных моделей и методов в рамках распределённой информационной системы КИМРА. Предложена её архитектура, сформулированы требования к его подсистемам и структуре банка КИМ, приведены детали реализации модуля проведения взаимного оценивания, описана апробация результатов диссертационного исследования и её результаты.

### 4.1 Архитектура распределённой информационной системы КИМРА

Работы по проектированию и конструированию КИМРА начинались ещё до облачной революции и популяризации новых подходов к построению распределённых систем. Отметим, что появление платформ *Microsoft Classroom* и *Google Classroom* (как следующего поколения *LMS*), а также *MOOC*-платформ перевело огромное число ПСУН в статус устаревших. Текущая спецификация требований описывает КИМРА как слабосвязную распределённую систему, базирующуюся на инфраструктуре интернет, использующую открытые интерфейсы (в основном базирующиеся на *JSON RESTful API*) и стандартные протоколы для связи подсистем.

Подсистемы «Управление жизненным циклом КИМ (УЖЦ КИМ)», «Банк КИМ» и «Открытая экспертиза КИМ» образуют фундамент полноценной интегрированной системы поддержки КИМ различных видов. С учётом их интерфейсов и методологии строятся формально независимые подсистемы «Проведение контрольно-измерительных мероприятий (Проведение ИвО)», «Редактор тестовых заданий», «Анализ результатов Измерений в Образовании (АР ИвО)». Перечисленные подсистемы могут использовать в своей работе технологические подсистемы: подсистему «Оповещения», позволяющую создавать и отслеживать наступление событий с автоматическим уведомлением подписчиков, и подсистему «Защищённое хранилище персональных данных (ЗХПД)», предоставляющую сервис защищённого хранения персональных данных, их локализации и аудита. Рисунок 4.1 даёт общее представление об архитектуре РИСУН КИМРА.

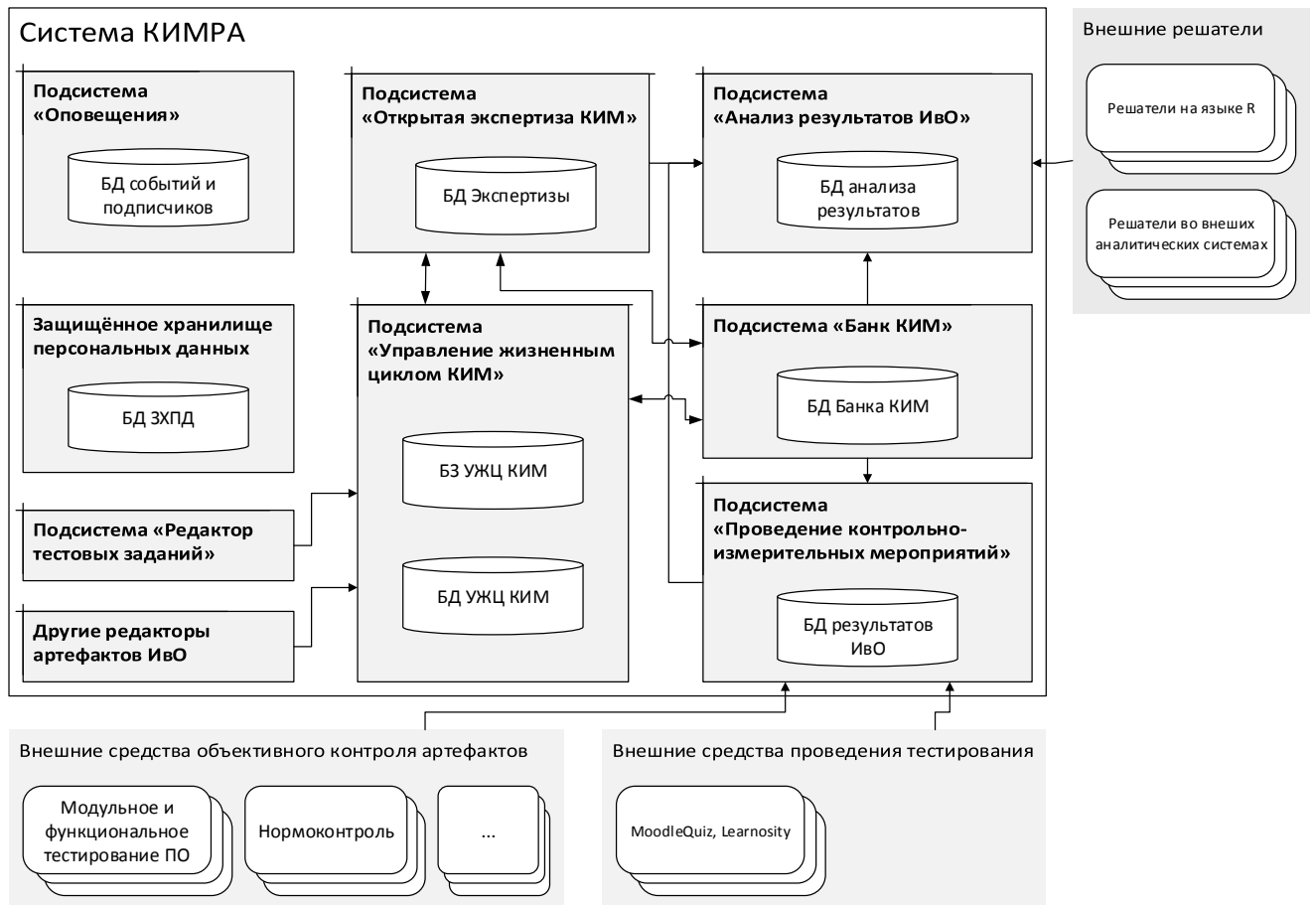


Рисунок 4.1. Архитектура программного комплекса КИМРА

Детализированная архитектура КИМРА представлена в 0.

Матамета модель КИМРА позволяет работать подсистемам как единому целому, поддерживая сквозное понятие контейнера КИМ (на основе *JSON Scheme*) и онтологию учебного процесса в самом общем виде.

#### Пример. JSON-Shema контейнера КИМ

```
{
 "$schema": "http://json-schema.org/draft-04/schema#",
 "title": "EAM Container",
 "description": "Schema for a EAM Container",
 "type": "object",
 "properties": {
 "allOf": [
 { "$ref": "BEP.json" },
 {
 "properties": {
 "ContainerType": {
 "enum": [
 "TestItem",

```



```

 "Test",
 "TestResults"
],
 },
 "Content": { "type": "object" }
},
"required":
[
 "ContainerType",
 "Content"
]
}
]
}
}
}

```

### Пример. JSON-Shema свойств базового типа

```

{
 "$schema": "http://json-schema.org/draft-04/schema#",
 "title": "BEP",
 "description": "Schema for a base entity properties",
 "type": "object",
 "properties": {
 "Id": { "type": "string" },
 "TS_LastChange": { "type": "string" },
 "Author": { "type": "string" }
 }
}

```

### Пример. JSON-Shema тестового задания «Альтернативы»

```

{
 "$schema": "http://json-schema.org/draft-04/schema#",
 "title": "Alternative Test Item",
 "description": "Schema for an alternative test item",
 "type": "object",
 "properties":
 {
 "TestItemType":
 {
 "enum":
 [
 "Choice",
 "BinaryChoice"
]
 },
 "Stem": { "type": "string" },
 "Choice":
 {
 "type": "array",
 "items": { "type": "object" }
 },
 "Answer":
 {
 "type": "array",
 "items": { "type": "object" }
 },
 "required":
 [
 "Stem",

```

"Answer"

```

 }
]
}

```

■

Метамодел ь КИМРА позволяет обрабатывать КИМ некоторых известных типов, например, тестовых заданий или заданий на взаимное оценивание. Метамодел ь поддерживает онтологии тестологии, взаимного оценивания и т.п., ссылаясь на онтологию метаметамодел и.

Модел ь данных КИМРА состоит из: 1) БЗ ЖЦ КИМ (базы знаний жизненного цикла КИМ); 2) БД УЖЦ КИМ (базы данных управления жизненным циклом КИМ); 3) БД банка КИМ (расширяющей понятие банка тестовых заданий и поддерживающей генерацию не только тестов, но и других «заданий»); 4) БД экспертизы КИМ (поддерживающей обсуждение КИМ в различных сценариях сообществами различной организации); 5) БД результатов ИОВ (накапливающей результаты любых измерений, например, результаты тестов); 6) БД анализа результатов измерений в образовании; 7) БД событий и подписчиков (для функционирования подсистемы «Оповещения»).

Общая слабосвязанная архитектура КИМРА и разделение модели данных допускает использование её подсистем другими ПСУН и банками КИМ.

#### 4.2 Подсистема «Банк КИМ»

Анализ применяемых математических модел ей измерений в образовании показал сосуществование на разных уровнях наиболее популярных теорий (*СТТ*, *ИРТ* и *ТРТ*), а также выявил потребность хранения и учёта характеристик заданий, используемых разными методами анализа результатов измерений. Этот результат положен в основу архитектур подсистем «Банк КИМ», «Проведение ИвО» и «АР ИвО», дающих возможность их использования в качестве основы для проведения контрольно-измерительных мероприятий разных видов и масштаба. Оценка популярных ПСУН по критерию поддержки банка КИМ показала отсутствие таких возможностей – БТЗ и программные средства управления ими доступны только на уровне масштабных национальных тестирований. Они дорогостоящи в поддержке

и использовании, а также, исходя из обзора литературы и опыта реального внедрения, реализованы на технологиях предыдущего поколения.

Возникновение банка КИМ (БКИМ) для РИСУН поддержки ЖЦ КИМ является продолжением традиций ИвО, заложенных компьютерным тестированием. Как уже отмечалось, главенство САТ на уровне национальных и межнациональных тестирований актуализировало вопросы, связанные с формированием банков тестовых заданий (БТЗ). Задача, поставленная в данной диссертации – шире и предполагает формализацию требований к БКИМ, который включает в себя БТЗ.

Говоря о требованиях к БТЗ, авторы публикаций иногда без дополнительных уточнений подразумевают не функциональные требования к информационной системе, а набор пожеланий к некоторой сущности [58, 57]. Автор данной работы полагает наиболее вероятным, что под БТЗ в таких случаях подразумевается пул тестовых заданий [*item pool*], обычно охватывающий полный набор заданий, разработанных для проведения тестирований по дисциплине (дисциплинам) или достаточный набор заданий для реализации САТ. Разумеется, отдельные функциональные требования могут вытекать из подобных пожеланий, например, в [13, 61, 123] отражено каким образом они, а также сценарии использования проявляются в структуре БТЗ.

Исследователи отмечают размытость границы между понятиями БТЗ [*item bank*] и пул (набор) тестовых заданий [*item pool*], подробное исследование этих двух понятий, их истории и уточнений для отдельных областей, например, САТ можно найти в работе [197].

В данной работе мы будем опираться на определение, приведённое в [123]. БТЗ – это база данных, содержащая уникально-идентифицируемые тестовые задания и необходимые данные, позволяющие осуществлять выбор [*selection*] для нужд сборки тестов.

#### 4.2.1 Требования к интернационализации

Локализацией [*localization, L10n*] называется процесс адаптации программного обеспечения под конкретные национальные требования. Отметим, однако, что

процессы, связанные с локализацией, являются лишь частью другого, более масштабного процесса – интернационализации. Интернационализация [*internationalization, i18n*] – это способ проектирования программного обеспечения, при котором возможность многоязыковой поддержки закладывается ещё на этапе его проектирования.

Возможность интернационализации образовательного программного обеспечения особенно значима для веб-средств, например, *МООС*-платформ, применяемых по всему миру. Идеология интернационализации заложена в такие популярные *МООС*-платформы, как *Coursera* и *EdX*, и автоматизирована при помощи средств платформы *Transifex* [198]. Некоторые вопросы локализации её средствами *Transifex* рассмотрены в докладе [199].

Существенным вопросом интернационализации для образовательного ПО является возможность интернационализации не только интерфейсов, но и контента. Это связано с тем, что существенную ценность и практическую значимость имеют методические и образовательные ресурсы [*educational resources*], доступные через ПО.

Заметим, что интернационализация образовательного контента обделена вниманием специалистов по разработке программного обеспечения. Одной из инициатив, на базе которой возможно подобие подобной интернационализации, является проект *Trinket* (<https://trinket.io>), построенный по принципам системы контроля версий [*concurrent version system*]. Учебная единица – *trinket* (от одного примера до полного содержания учебной дисциплины) может быть унаследована и использована в качестве базы для новой учебной единицы, в том числе в производной единице может быть произведена и интернационализация содержимого. Заметим, что при таком подходе существует необходимость порождения производного *trinket* для каждого нового языка, т.е. речь идёт об эмуляции интернационализации, а не об интернационализации в полном смысле слова.

В информационной системе КИМРА заложена возможность интернационализации как интерфейса (с использованием средств, предоставляемых платформой *.NET*), так и учебного содержимого, начиная с модели данных.

### 4.2.2 Версионирование данных в КИМРА

ЖЦ КИМ разделяется на временные периоды: *КИМ доступен пользователям*, *КИМ недоступен пользователям*, *КИМ изменён и доступен пользователям в изменённом виде*. Для информационной системы, поддерживающей процессы разработки и повышения качества КИМ, существенным является требование сохранения временного периода, во время которого задание находилось в том или ином состоянии. Данные, таким образом, должны поддерживать один из интервальных вариантов версионирования. Интервальное версионирование подразумевает сохранение для каждой версии данных значений временных промежутков, в которые данная версия была актуальной [200].

Для КИМРА используем версионирование реляционной модели на основе хранения универсальных кортежей с данными об изменениях [*tuple-versioning*] [201]. Для общего предка, таблицы **BaseEntity**, есть таблица **BaseEntityVersioning**, которая является таблицей версий. В таблице версий сохраняется кортеж в формате **key=value** и даты актуальности этой версии кортежа (дата изменения **StartDate**, дата следующего изменения **EndDate**). Удобство такого способа версионирования в том, что про каждый объект известно в какой период он находился в заданной версии.

Порядок обновления таблиц состоит в обновлении **BaseEntity** и последующим обновлении таблицы **BaseEntityVersioning**. В таблицу добавляется кортеж вида:

(*ID, BaseEntityID, Tuple, VersionNumber, StartDate, EndDate, Operation* )  
*Operation* = {*insert, update, delete*}

Операция «вставка» (*insert*): в таблицу **BaseEntityVersioning** добавляется кортеж *Tuple*, и *StartDate*, *EndDate*, где *EndDate* открыт для этой версии.

Операция «обновление» (*update*): в таблицу **BaseEntityVersioning** у последней версии проставляется значение *EndDate* и добавляется новая строка, содержащая *Tuple*, и *StartDate*, *EndDate*, где *EndDate* открыт для этой версии.

Операция «удаление» (*delete*): в таблицу *BaseEntityVersioning* у последней версии проставляется значение *EndDate* и добавляется новая строка, содержащая *StartDate = EndDate*.

#### 4.2.3 Требования к банку КИМ информационной системы КИМРА

Клиентами БКИМ являются подсистемы и модули КИМРА и другие ПСУН.

##### Форматы входных/выходных данных.

1. Форма совместимости средств контроля знаний, определённый для протокола *APIP*.
2. Внутренний *JSON*-основанный формат программного комплекса КИМРА.

##### Требования к *Web-API*.

1. *Web-API* удовлетворяет базовым принципам построения *REST*-интерфейсов и обеспечивает двухфакторную аутентификацию по протоколу *OAuth 2.0*.
2. Организовать асинхронное *RESTful*-взаимодействие на основе уникального токена сессии доступа к БКИМ:
  - a) запрос «Создать новое КИМ»;
  - b) запрос «Получить актуальную версию КИМ»;
  - c) запрос «Получить КИМ с версиями»;
  - d) запрос «Обновить версию КИМ»;
  - e) запрос «Обновить существующее КИМ»;
  - f) запрос «Удалить КИМ»;
  - g) запрос «Обновить правила оценивания КИМ»;
  - h) запрос «Обновить характеристики КИМ»;
  - i) запрос «Обновить статус КИМ»;
  - j) сохранение/восстановление сессии.

##### Требования к БКИМ.

1. Сохранение версий [*versioning*] КИМ.
2. Возможность локализации КИМ.

- а) Статусы КИМ («Открыто для экспертизы», «Используется для контрольно-измерительных мероприятий», «Удалено», «Не активно» (не используется ни для экспертизы, ни для тестов).
- 3. Тегирование КИМ:
  - а) Теги;
  - б) теги под управлением фолксономии.
- 4. Сохранение характеристик КИМ:
  - а) показатели качества;
  - б) правила оценивания.
- 5. Поддержка связей между КИМ и проверяемыми дидактическими единицами (онтология учебного процесса, дисциплины, ДЕ).
- 6. Сквозное комментирование со ссылкой на элементы КИМ.
- 7. Поддержка связей между КИМ и учебной дисциплиной.
- 8. Конструирование КИМ, состоящих из других КИМ, соответствующих наборам учебных дисциплин, дидактических единиц, тегам, онтологиям учебного процесса и т.д.).
- 9. Обеспечение экспорта и импорта тестов и тестовых заданий в форматах (*GIFT*, *MoodleXML*, *QTI/APIP 2.0*).
- 10. Поддержка ИОС:
  - а) в форме комментария [*commentary*] к КИМ. От преподавателя к студенту – автоматически. Комментарий добавляется к КИМ при создании. От студента к преподавателю – комментарии к КИМ на уровне открытой экспертизы, на уровне прохождения теста;
  - б) в форме сообщения [*message*].

#### 4.3 Подсистема «Редактор тестовых заданий»

##### 4.3.1 Требования к визуальному редактору тестовых заданий

Клиентами визуального редактора тестовых заданий являются разработчики тестовых заданий и другие ПСУН, например, БТЗ, системы поддержки компьютерного тестирования и т.д.

### Форматы входных/выходных данных.

1. Формат совместимости средств контроля знаний, определённый для протокола APIP.
2. Внутренний JSON-основанный формат программного комплекса КИМРА.

### Требования к Web-API.

1. *Web-API* должен удовлетворять базовым принципам построения *REST*-интерфейсов и обеспечивать двухфакторную аутентификацию по протоколу *OAuth 2.0*.
  - а) Обеспечить согласование с *REST Web-API* хранилища заданий с поддержкой *CRUD*-операций.
2. Организовать асинхронное *RESTful*-взаимодействие на основе уникального токена сессии редактирования:
  - а) запрос «Создать новое задание» предоставляет интерфейс мастера создания нового задания;
  - б) запрос «Редактировать задание» предоставляет интерфейс редактора задания, переданного как параметр запроса;
  - в) запрос «Создать на основе шаблона» обеспечивает предоставление шаблона и примеров использования в интерфейсе пользователя;
  - г) запрос статуса сессии по токену;
  - д) запрос результатов редактирования;
  - е) сохранение/восстановление сессии.

### Требования к визуальному редактору тестовых заданий

Визуальный редактор должен предоставлять средства для разработки тестовых заданий, входящих в предложенную в главе 1 классификацию.

На экране (рисунок 4.2 представляет прототип интерфейса с пользователем) должны присутствовать несколько областей.

1. Область конструирования тестового задания.



2. Редакторские «ленты». Объекты – части заданий располагаются на редакторских лентах и перемещаются в область конструирования перетаскиванием [*drag and drop*]:

- а) лента составных частей заданий;
- б) [*Опционально*] лента версий заданий. Лента версий заданий отображается только в случае, если пользователь редактора поддерживает работу с версионированными объектами.

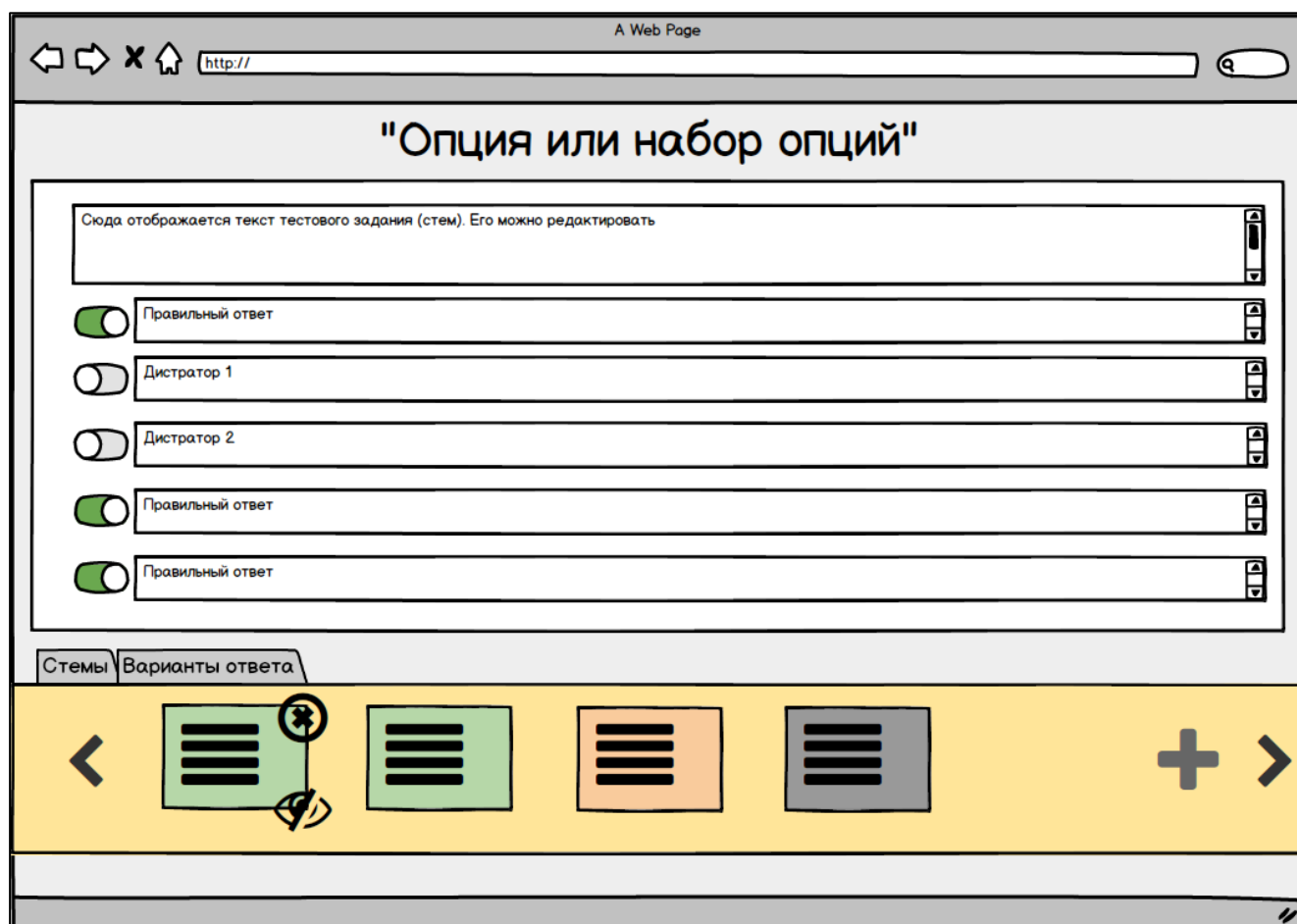


Рисунок 4.2. Прототип интерфейса создания тестового задания «Набор опций»

#### 4.4 Подсистема «Открытая экспертиза КИМ»

Анализ процессов ЖЦ тестовых заданий, принятых на уровне массовых тестирований и процессов открытого рецензирования, формализованных *открытой наукой*, позволил предложить и описать процесс проведения ОЭ КИМ [202, 61, 105]. Эта идеология положена в основу подсистемы «Открытая экспертиза КИМ» и позволила впервые согласовать общепринятые на уровне массовых тестирований

процедуры с принципами экспертизы, предлагаемыми открытой наукой и обобщить их до уровня произвольных КИМ. Реализуемые подсистемой сценарии позволяют поддерживать также учебные процессы, например, самоподготовки учащихся и подготовительных контрольно-измерительных мероприятий с получением ИОС и возможностью оповещения авторов КИМ.

Функции участников процесса ОЭ КИМ.

Разработчик КИМ.

1. Создать КИМ.
2. Редактировать КИМ.
3. Назначить статуса КИМ (специальное тегирование).
4. Инициализировать ОЭ.
5. Комментировать КИМ.
6. Просмотреть рецензию(и).

Подписчик.

1. Просмотреть КИМ
2. Комментировать КИМ.
3. Добавить тег к КИМ.
4. Создать рецензию.
5. Редактировать рецензию.
6. Выбрать режима ОЭ (экспертиза, комментирование).

Эксперт.

1. Просмотреть КИМ.
2. Комментировать КИМ.
3. Назначить статус КИМ (специальное тегирование).
4. Создать рецензию.
5. Редактировать рецензию.
6. Оценить КИМ по шкале.

ОЭ инициируется, когда разработчик КИМ выставляет КИМ на экспертизу или направляет приглашение провести экспертизу конкретному эксперту. Во втором случае экспертиза также инициируется, как открытая. Особенности модели

КИМ, привязывающей КИМ к системе учебно-методических материалов, позволяют направлять КИМ на рецензию конкретной группе экспертов. Рисунок 4.3 иллюстрирует ключевые процессы ОЭ КИМРА, детализируя основной сценарий ОЭ КИМ с участием ролей Эксперт и Подписчик.

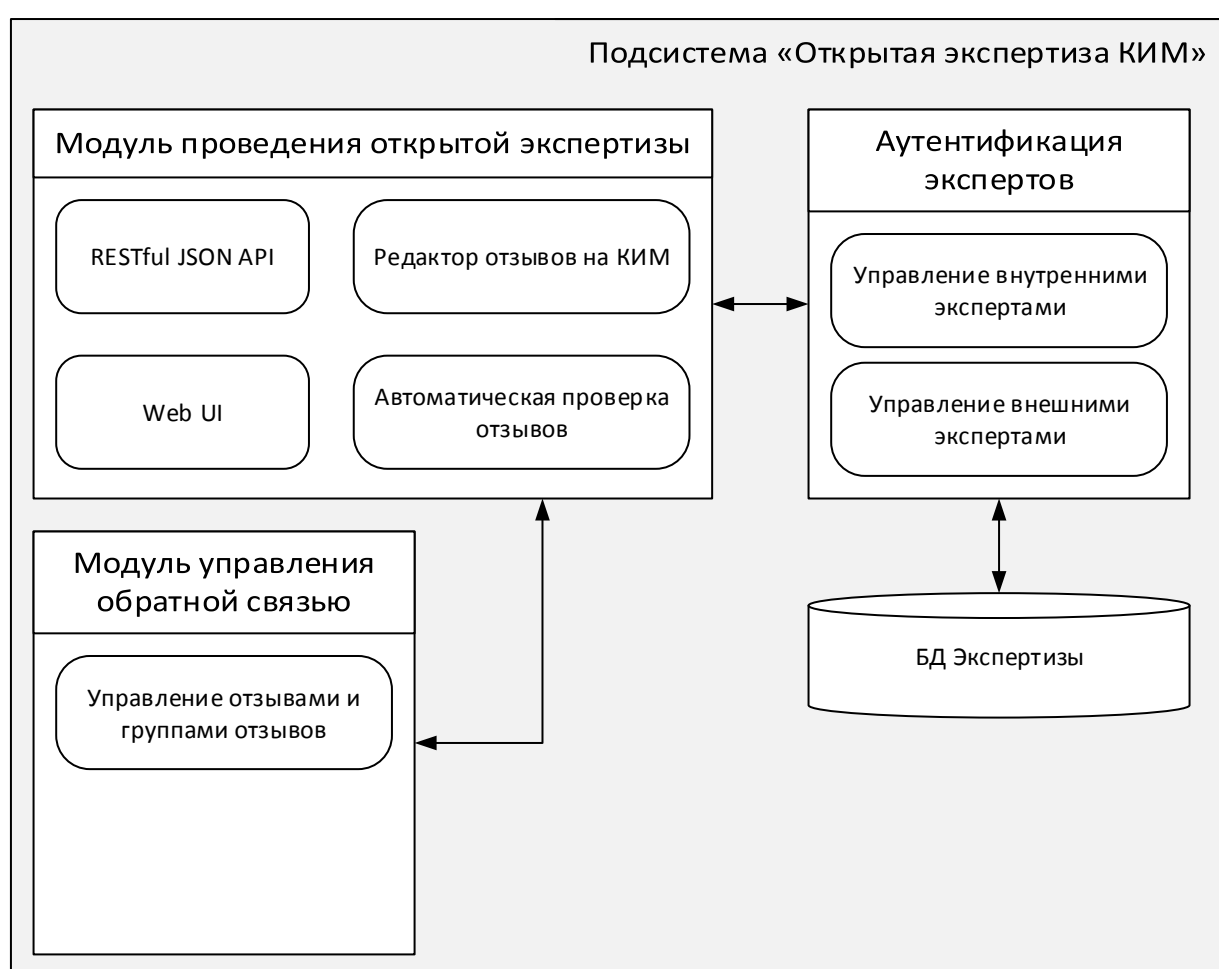


Рисунок 4.3. Архитектура подсистемы «Открытая экспертиза КИМ» РИСУН КИМРА

#### 4.5 Подсистема «Анализ результатов ИвО»

Математическое обеспечение систем анализа результатов измерений в образовании является серьёзно проработанным направлением (см. раздел 2.1). Математические методы и модели автоматизированы в различных ПСУН, а также в специализированных математических пакетах. В данном разделе приведён краткий обзор пакетов на языке *R*, поддерживающих анализ результатов тестов по *СТТ*, *IRT* и *TRT* и сформулированы требования к подсистеме «Анализ результатов ИвО» РИСУН КИМРА.

#### 4.5.1 Обзор пакетов R для обработки результатов тестирования

##### 4.5.1.1 Классическая теория тестирования

Пакет `ctt` (<https://cran.r-project.org/web/packages/CTT/CTT.pdf>) состоит из функций, позволяющих обрабатывать результаты тестов, состоящих из заданий типа «Альтернативы» (см. раздел 1.3.3). Методы позволяют вычислять показатели надёжности и валидности теста, определённые *СТТ*, строить характеристические кривые для заданий, проводить анализ дистракторов. Среди ограничений пакета: неприспособленность для работы с заданиями других типов; отсутствие функций, необходимых для вычисления характеристик заданий, а также функций, позволяющих приводить первичные баллы к различным шкалам.

##### 4.5.1.2 Современная теория тестирования

Вследствие высокого разнообразия математических моделей *IRT*, связанных с ней пакетов достаточно много. Например, пакет для расчёта характеристик при помощи многомерных моделей:

- `diffIRT` (<https://cran.r-project.org/web/packages/diffIRT/diffIRT.pdf>);
- пакет `irtrees` (<https://cran.r-project.org/web/packages/irtrees/irtrees.pdf>), реализующий модель *IRTrees* для работы неполными данными;
- `eRm` (<https://cran.r-project.org/web/packages/eRm/eRm.pdf>),
- `mixRasch` (<http://cran.at.r-project.org/web/packages/mixRasch/mixRasch.pdf>),
- `mRm` (<http://cran.r-project.org/web/packages/mRm/mRm.pdf>), содержащие методы расширенных и смешанных моделей *IRT* и другие.

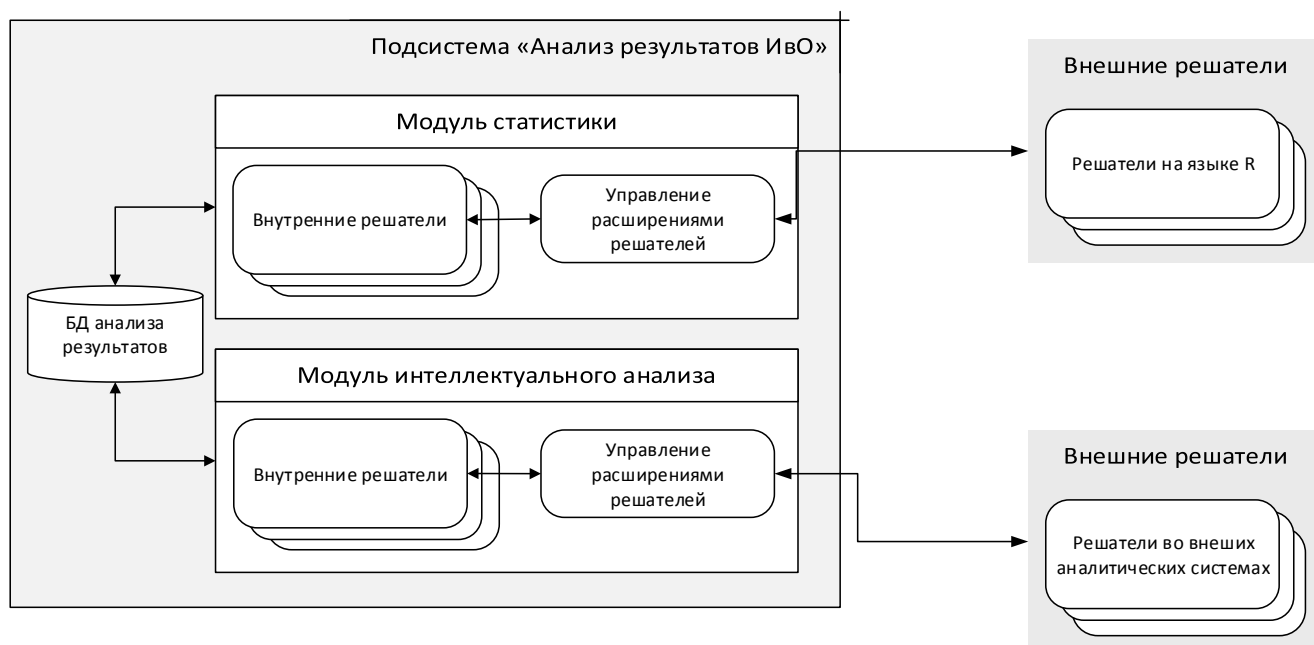
Для решения отдельных задач, например, расчёта характеристик заданий, генераций выборок, визуализаций также есть отдельные пакеты, например:

- `irtoys` (<http://cran.r-project.org/web/packages/irtoys/irtoys.pdf>);
- `pairwise` (<http://cran.r-project.org/web/packages/pairwise/pairwise.pdf>);
- `raschsampler` (<http://cran.r-project.org/web/packages/RaschSampler/RaschSampler.pdf>).

### 4.5.1.3 Теория тестлетов

Для решения задач *TRT* в репозитории *CRAN* доступен пакет *sirt* (<https://cran.r-project.org/web/packages/sirt/sirt.pdf>), вобравший в себя несколько моделей *IRT*, стандартные функции *R*, необходимые для вычисления и валидации этих моделей, а также функции для Байесовского анализа. Пакет *sirt* имеет сложную структуру, состоит из более чем ста функций, многие из которых связаны со внешними пакетами.

Обзор показал, что наиболее удачным проектным решением для КИМРА является поддержка подключения внешних решателей для обработки результатов ИвО. Таким образом, предполагается, что подсистема «Анализа результатов ИвО» предоставляет *Web-API*, удовлетворяющий базовым принципам построения *REST*-интерфейсов и обеспечивающий двухфакторную аутентификацию по протоколу *OAuth 2.0* (рисунок 4.4).



**Рисунок 4.4. Архитектура подсистемы «Анализ результатов ИвО» РИСУН КИМРА**

## 4.6 Подсистема «Управление жизненным циклом КИМ»

Анализ бизнес-процессов высших учебных заведений и процессов ЖЦ КИМ позволил сформировать набор состояний/статусов КИМ: «на рецензии», «актуально», «удалено» и т.п. (рисунок 4.5).

Важность обеспечения содержательной валидности инструментов измерений требует от ПСУН поддержки связей КИМ с базами знаний дисциплин или их онтологиями. В связи с перечисленными требованиями подсистема «УЖЦ КИМ» содержит модуль управления статусами и состояниями КИМ в банке, модуль онтологического моделирования, поддерживающий онтологию учебного процесса, и редактор онтологий.



Рисунок 4.5. Архитектура подсистемы «Управление жизненным циклом КИМ» КИМРА

Существенной особенностью данной подсистемы является возможность управления связями между КИМ и единицами баз знаний и онтологий. Наличие этих связей вкупе с характеристиками заданий позволяют использовать подмножество тестовых заданий, сохранённых в КИМРА для поддержки САТ.

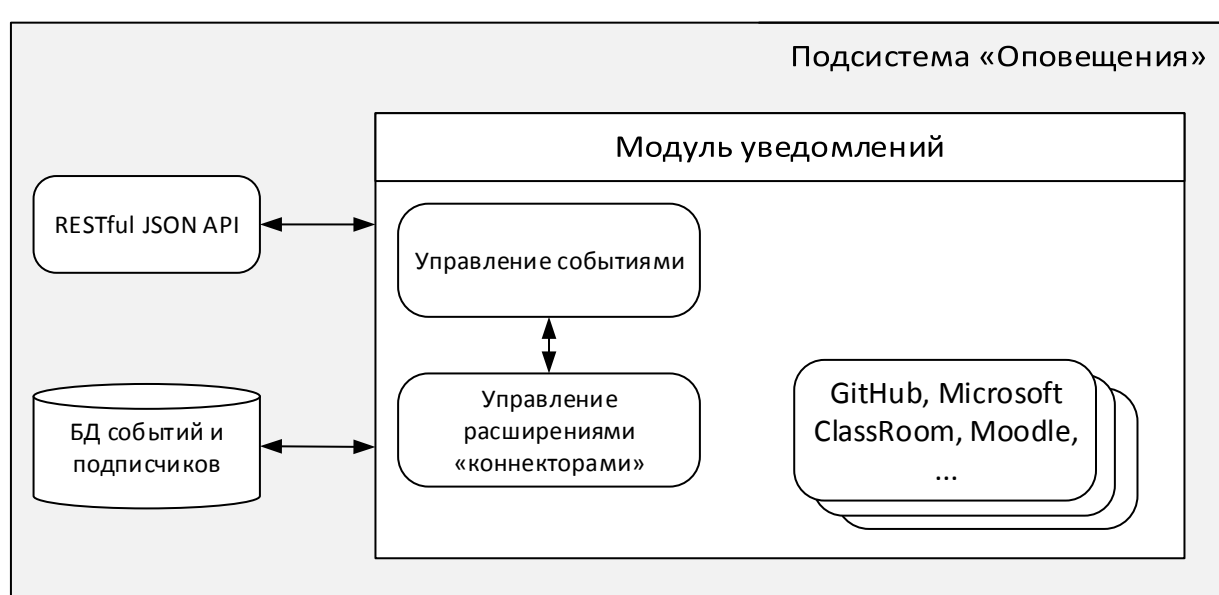
#### 4.7 Подсистема защищённого хранения персональных данных

Подсистема «Защищённое хранилище персональных данных (ЗХПД)»:

- 1) предоставляет сервис защищённого хранения персональных данных, их локализации и аудита;
- 2) требует сертификации;
- 3) должна соответствовать законам 149-ФЗ «Об информации, информационных технологиях и о защите информации», 152-ФЗ «О персональных данных», 63-ФЗ «Об электронной подписи».

#### 4.8 Подсистема «Оповещения»

Активные формы обучения и формирующее оценивание вводят понятие информативной обратной связи, как обязательной части учебного процесса (рисунок 4.6). Подсистема «Оповещения» обеспечивает поддержку системных свойств КИМРА, с точки зрения поддержки учебного процесса эта подсистема является ключевой в реализации информативной обратной связи.



**Рисунок 4.6. Архитектура подсистемы «Оповещения» комплекса КИМРА**

Одним из наиболее показательных процессов, злодействующих функции подсистемы «Оповещения», является процесс предоставление информативной обратной связи (ИОС). Рисунок 4.7 формально описывает протекание этого процесса в КИМРА с точки зрения ролей студента и преподавателя.

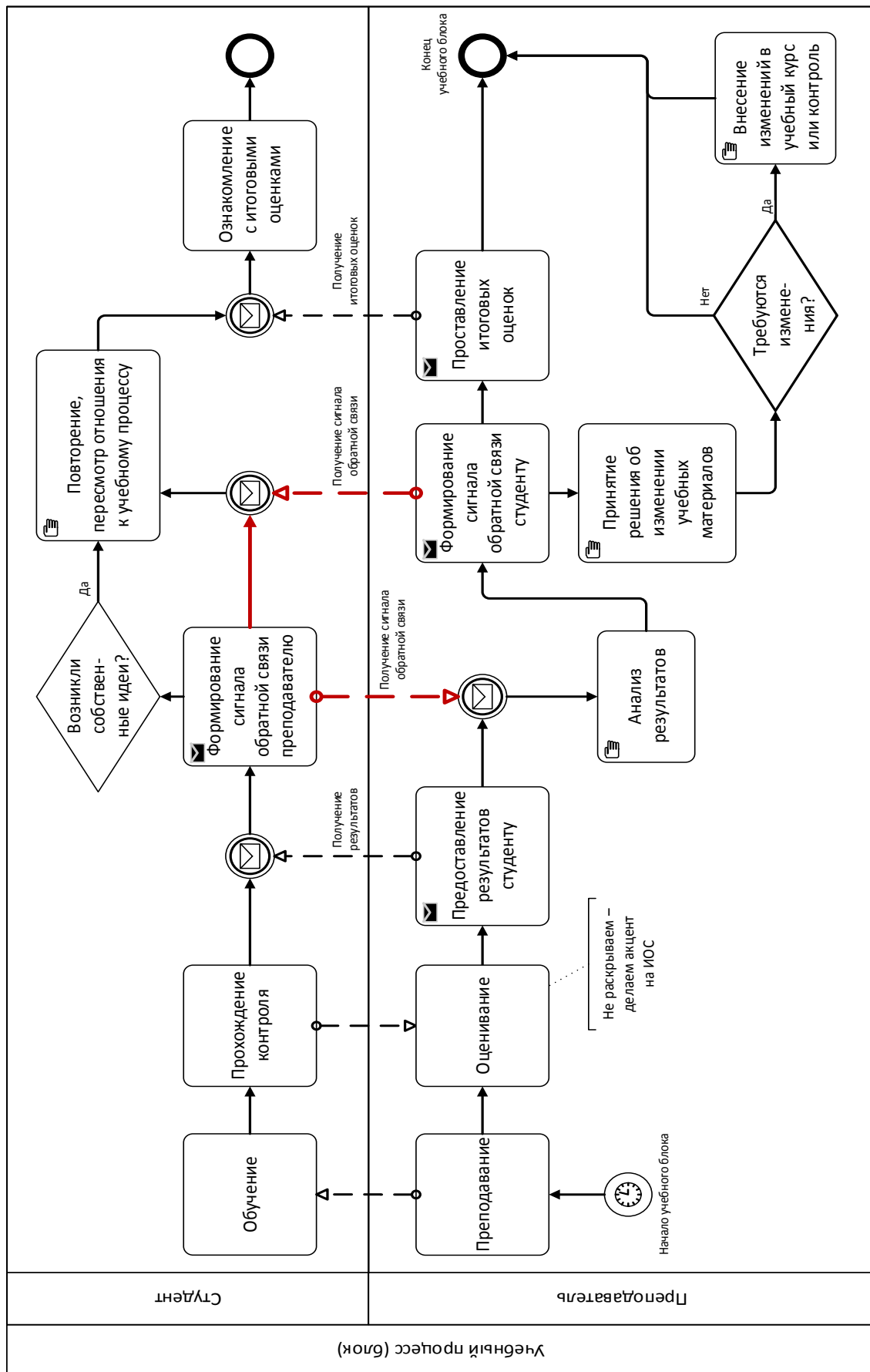


Рисунок 4.7. BPMN-диаграмма формирующего оценивания с информативной обратной связью



#### 4.9 Подсистема «Проведение контрольно-измерительных мероприятий»

При прототипировании подсистемы проведения ИвО было принято решение на данном этапе не разрабатывать собственный модуль проведения тестирования в образовании. Основанием для решения послужило наличие различных подсистем, модулей и сервисов, например, *Learnosity*, *Respondus*, *MoodleTest*, реализующих практически все желаемые функции с учётом контролируемости среды. Поддержка возможности интеграции с существующими средствами контроля обеспечена использованием формата интероперабельности доставки тестов *QTI/APIP*. Модуль этой подсистемы, реализующий измерения в форме взаимного оценивания, после прототипирования выделен в отдельное свободно распространяемое ПСУН *PASCA* (*Peer Assessment System for Complex Artifacts*), доступное в репозитории (<https://bitbucket.org/SiberianShaman/pasca>).

##### 4.9.1 Модуль «Обеспечение доставки»

Особенности доставки КИМ зависят от его типа (*Type*). В этом разделе будут рассмотрены детали реализации обеспечения доставки заданий взаимного оценивания. Поскольку основой идеологии КИМРА является поддержка подключения внешних модулей проведения контрольно-измерительных мероприятий, можно сформулировать следующие основные требования к модулю обеспечения доставки КИМ:

- 1) преобразование КИМ из внутреннего формата представления в один из форматов интероперабельности ПСУН;
- 2) предоставление *RESTful-API*.

##### 4.9.2 Модуль «Проведение взаимного оценивания»

###### 4.9.2.1 Основные понятия процессов взаимного оценивания

Определим взаимное оценивание [*Peer Assessment (PA)*] как контрольно-измерительное мероприятие, организованное в форме рандомизированного взаимного рецензирования [*Peer Review (PR)*] произвольных артефактов, являющихся результатом выполнения учебного задания с предварительно формализованными

критериями оценивания. Процесс *РА* одного задания назовём сессией взаимного оценивания (*CBO*). Выделим следующие роли в процессах *CBO* (рисунок 4.8).

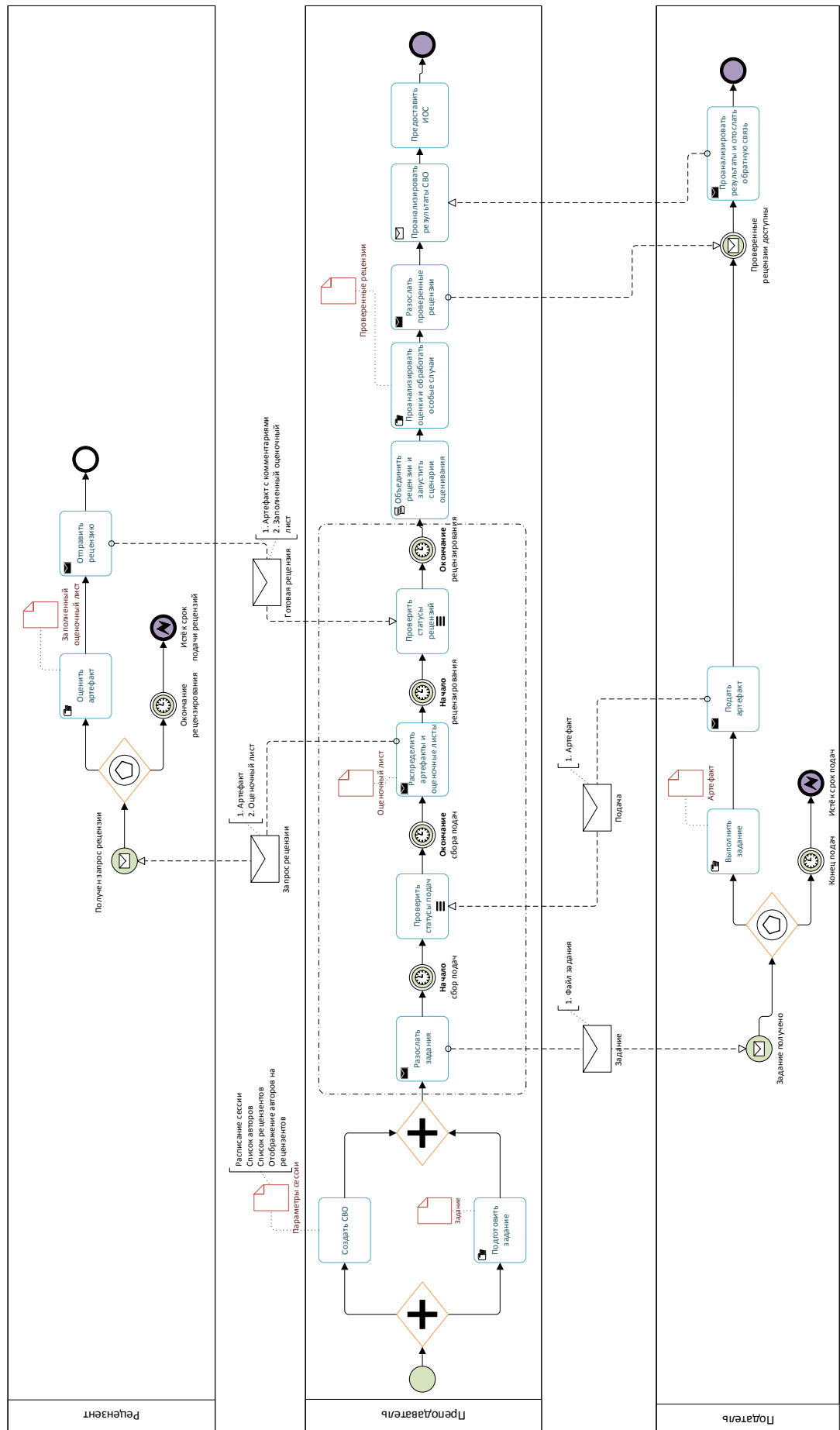
1. Преподаватель [*Teacher*] – любой организатор взаимного оценивания, имеющий полный доступ к данным текущей *CBO*.
2. Студент [*Student*] – любой студент, потенциально имеющий возможность участвовать в текущей *CBO*.
3. Автор [*Initial Author*] – Студент, зарегистрированный в качестве будущего подателя артефакта в текущей *CBO*.
4. Податель [*Submitter*] – Студент, направивший артефакт на рецензирование.
5. Рецензент [*Reviewer*] – Студент, отославший валидный Оценочный лист.

Следующие объекты данных являются базовыми в рамках *CBO*.

1. Параметры *CBO* [*PA session parameters*] – набор формальных параметров текущей *CBO* (пути, идентификаторы, тема, дедлайны, ...).
2. Учебное задание [*Assignment*] – учебное задание, выполняемое студентами в рамках *CBO*.
3. Оценочный лист [*PR form*] – таблица, специфицирующая поля рецензии для конкретного типа артефакта и конкретного задания; содержит понятные рецензенту описания полей, примеры заполнения и правила валидации значений полей. Адекватный оценочный лист всегда содержит дополнительное текстовое поле «свободный комментарий» для неформальных впечатлений рецензента.
4. Подача [*Submission*] – 1) Артефакт, направленный на рецензирование конкретным Отправителем в рамках текущей *CBO*; 2) событие отправления артефакта.
5. Рецензия [*Review*] – 1) заполненный Оценочный лист, отосланный Рецензентом; 2) событие отправления заполненного оценочного листа.
6. Обратная связь [*Feedback*] – любая дополнительная информация от Студента, отличная от Подачи и Рецензии.

Некоторые замечания о базовых понятиях и терминологии:

- 1) в нашем варианте внедрения процедур взаимного оценивания предполагается, что множество рецензентов есть подмножество подателей (в терминах теории множеств). Но ничего не мешает построить построение этих множеств для других схем проведения взаимного оценивания;
- 2) глаголы «подать» [*Submit*] и «разослать» [*Send/Broadcast*] используются для различения действий на стадии сбора артефактов (*submissions*) и в других ситуациях.



**Рисунок 4.8. BPMN-диаграмма процесса взаимного оценивания в рамках сессии**

### 4.9.3 Средства разработки

Прототипирование и реализация подсистем проводилась с использованием различных технологий и инструментов:

- языки *C#, Python, Java, VBA, R*;
- СУБД *Postgres, SQLite, Microsoft Access*;
- среды разработки *Microsoft Visual Studio, Microsoft Office, R Studio*;
- платформы *Microsoft Azure* и *Office 365, Apache HTTP Server, Elasticsearch*.

### 4.10 Апробация и внедрение

Результаты, представленные в диссертации, обсуждались на более чем десяти конференциях и научно-практических мероприятиях. По результатам выступления на IV-ой Всероссийской научно-практической конференции «Информационные технологии в образовании XXI века» в 2014 году была уточнена русскоязычная терминология, связанная с процессами активного обучения. На XIV-ой открытой Всероссийской конференции «Преподавание информационных технологий в Российской Федерации» в 2016 году обсуждалась предложенная русскоязычная терминология, связанная с взаимным оцениванием, выделенные роли пользователей в ПСУН поддержки взаимного оценивания и структура бизнес-процессов автоматизации. По результатам конференции терминология и процессы были уточнены и обсуждены на методической мастерской ФГАОУ ВО «НИУ «ВШЭ» и семинаре методического центра ФГАОУ ВО «НИУ «ВШЭ» «Методические среды».

Особенности автоматизации процессов активного обучения и поддержки формирующего оценивания для различных образовательных программ и систематизация требований к правилам оценивания обсуждались на конференциях 19<sup>th</sup> и 18<sup>th</sup> *International Conference on Interactive Collaborative Learning (ICL 2016 и 2015)*. Исследования, посвящённые использованию компьютерных интеллект-карт как инструмента формирующего оценивания, проводились автором совместно с Ю.О. Папушиной (к.с.н., доцент пермского кампуса ФГАОУ ВО «НИУ «ВШЭ»). По результатам опубликован ряд работ [203, 204], сформулированы требования и

начаты работы по разработке и проектированию автоматического средства оценивания компьютерных интеллектуальных карт.

Предложенная в диссертации модель открытой экспертизы КИМ, согласованная с принципами открытого образования и науки, обсуждалась с главным координатором сообщества по открытой науке России «Российские открытые знания» И.А. Радченко. Результаты исследований в области использования открытых данных в обучении студентов и реализации подходов открытого обучения освещены на нескольких конференциях и опубликованы в ряде работ [205, 206, 207].

Студентами ОПИ ФКН НИУ ВШЭ под руководством автора исследовались – в рамках курсовых работ форматы интероперабельности ПСУН. Разработанный под руководством автора К.С. Бабиным полнофункциональный прототип *web-API* конвертера тестовых заданий в закрытой форме доступен в открытом доступе (<http://testconstructor.azurewebsites.net>), результаты исследования и реализации опубликованы в [208]. По результатам исследований был сформирован пул форматов интероперабельности КИМРА, а также разработан внутренний формат представления данных в КИМРА.

Система взаимного оценивания, получившая рабочее название *PASCA* (*Peer Assessment System for Complex Artifacts*), впервые была пилотно внедрена в учебный процесс в 2015 году в рамках факультатива «Введение в программирование» бакалаврской образовательной программы «Программная инженерия», факультета компьютерных наук НИУ ВШЭ (65 человек). После прохождения курса студенты участвовали в анкетировании, часть которого была уделена вопросам о функционировании *PASCA* и предложениям по её улучшению.

Модифицированная версия *PASCA* использовалась для проведения сессий взаимного оценивания на курсах повышения квалификации учителей информатики школ университетского округа ФГАОУ ВО «НИУ «ВШЭ», а также в процессе преподавания дисциплин: «Проектирование взаимодействия с пользователем» (Магалега НИУ ВШЭ, 60 человек), «Деловая (конкурентная) разведка. Защита бизнеса от угроз в области экономики и финансов» (Майнор факультета экономики, 303 человека), «*Introduction to Advanced Macroeconomics*» (магистерская программа

«Экономика» ФГАОУ ВО «НИУ «ВШЭ», 90 человек), «Открытые данные» (магистерская программа «Журналистика данных» ФГАОУ ВО «НИУ «ВШЭ», 35 человек). Модифицированная в результате перечисленных прогонов версия *PASCA 2.0* в настоящее время проходит серию тестовых запусков.

Формализация бизнес-процесса взаимного оценивания, особенности архитектуры и реализации *PASCA*, а также результаты пилотного внедрения опубликованы [209] и обсуждены на конференции *20<sup>th</sup> International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES-2016)*, после чего были обобщены в работе [210].

#### 4.11 Выводы и результаты по главе

В главе подробно обсуждается архитектура и проектные решения, принятые для РИСУН КИМРА, нацеленной на поддержку ЖЦ КИМ. Сформулированы требования ко всем подсистемам КИМРА и их программным интерфейсам. Для отдельных модулей и подсистем изложены детали реализации.

Особое внимание уделено требованиям к БКИМ, которые выдвинуты на основании анализа технологических требований к банкам тестовых заданий и особенностей реализации ПСУН нового поколения (в том числе облачных), таких как *Trinket*, *Knewoton* и пр. Предложенный набор требований к реализованному в КИМРА БКИМ позволяет банку версионировать КИМ, поддерживать взаимосвязь КИМ с учебным контекстом, а также взаимодействовать с внешними ПСУН и использоваться в автоматизированных процессах поддержки активного обучения.

В главе обсуждаются результаты апробации теоретических результатов диссертационного исследования в рамках конференций и научно-практических семинаров, и их внедрение в учебных дисциплинах московского и пермского кампусов НИУ ВШЭ и НИУ ИТМО. Описаны результаты прототипирования и итоги пилотных внедрений подсистем КИМРА в учебные дисциплины различных направлений обучения НИУ ВШЭ. Результаты данного исследования, их обсуждение и перспективы развития темы подробно описаны в заключении.

## ЗАКЛЮЧЕНИЕ

В данной работе изучены существующие исследования в областях, непосредственно связанных с математическим обеспечением, проектированием и разработкой РИСУН. На основе результатов этих исследований предложены математические модели, численные методы и распределённая информационная система КИМРА (Контрольно-измерительные мероприятия разработка и анализ), позволяющая поддерживать процессы: 1) активного обучения и формирующего оценивания; 2) открытой экспертизы; 3) жизненного цикла контрольно-измерительных материалов.

Результаты, характеризующиеся научной новизной.

1. Сформированы требования к РИСУН поддержки ЖЦ КИМ, предложена и апробирована архитектура программной системы поддержки процессов ЖЦ КИМ, включая их открытую экспертизу. Результат основан на решении проектировать КИМРА как ПСУН нового поколения с распределённой архитектурой на базе предложенной формализации процесса открытой экспертизы КИМ, согласованном с принципами открытого образования и науки. Проанализированы особенности существующих ПСУН, изучены форматы интероперабельности подобных систем и сформулированы требования к подсистемам и модулям КИМРА, которые уточнены и дополнены требованиями к средствам поддержки ЖЦ КИМ. Предложена русскоязычная классификация тестовых заданий, адаптированная для нужд разработки ПСУН поддержки ЖЦ КИМ.
2. Для КИМРА предложена математическая модель КИМ, положенная в основу функционирования универсальной модели хранения данных и функционирования КИМРА. Предложенная модель облегчает постановку задачи автоматической генерации композитных КИМ и обеспечивает целостность процессов КИМРА как информационной системы.
3. Разработаны графовые модели для представления связей КИМ с результатами обучения, позволяющие исследовать взаимоотношения между КИМ в рамках предметных областей. Построенные модели активно используются в процессах открытой экспертизы КИМ и для построения композитных КИМ.



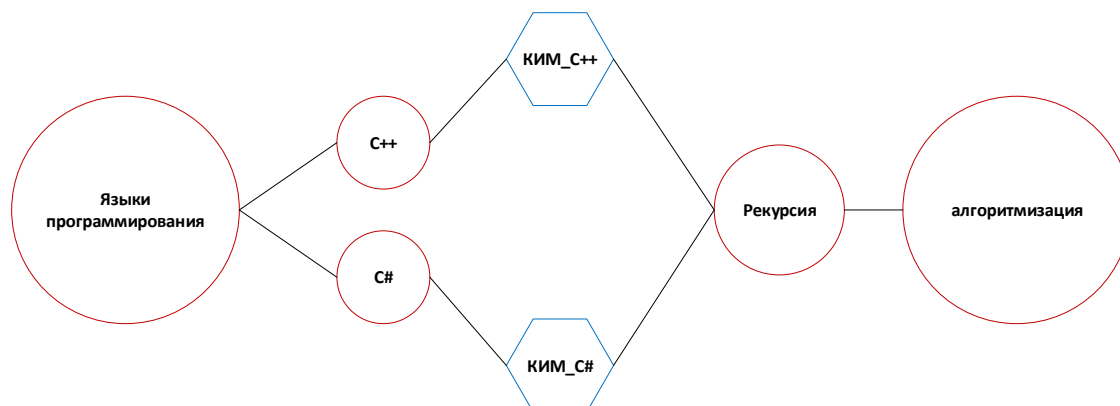
4. На основе математической модели КИМ и предложенных графовых моделей в КИМРА реализован в численный метод оптимизации подбора КИМ при создании композитного КИМ и его доставке средствами РИСУН.
5. Предложена модель бизнес-процесса и сформулированы требования к программной поддержке формирующего оценивания в форме взаимного оценивания и тестирования. Разработан и внедрён модуль проведения контрольно-измерительных мероприятий в форме взаимного оценивания *PASCA*. Процесс рандомизации рецензентов при взаимном оценивании реализован на основе предложенного в работе численного метода.

### Обсуждение результатов

Следует отметить учебные эффекты открытой экспертизы КИМ, поддерживаемый КИМРА. Участие в ОЭ открывает для студентов широкие возможности по самоподготовке, так как, по существу, реализует один из видов формирующего контроля, где ИОС представлена возможностью прямой коммуникации с разработчиками заданий посредством подсистемы комментирования и оповещения (раздел 4.7).

Тестирование и опытная эксплуатация выявили некоторые ограничения метода оптимальной композиции КИМ, проявляющиеся при его использовании для реальных данных. В некоторых случаях композитный КИМ не будет отвечать требованиям содержательной валидности (раздел 2.1.1.1). Например, если в параметрах композиции указана возможность расширения подмножества ДЕ (за счёт таксонов одного уровня) для обеспечения их предопределённого покрытия выбранными КИМ, то в случае нехватки заданий, проверяющих некоторую ДЕ, будут добавлены задания, проверяющие одну из смежных ДЕ, тогда высока вероятность, что композитный КИМ будет включать не только ДЕ, которые должны были подлежать проверке. Данная коллизия может быть раскрыта на примере. На **Ошибка!** **Источник ссылки не найден.** приведена схема связей между заданиями и элементами таксономий по программированию и алгоритмизации. В условиях задачи, когда требуется сконструировать композитный КИМ по языку C#, включающий за-

дания по реализации рекурсивных алгоритмов на языке *C#*, в случае нехватки заданий для обеспечения, заданного пользователем покрытия в композитный КИМ будут автоматически включены задания по языку *C++*, требования содержательной валидности о КИМ по *C#* будут нарушены:



**Схема связей между заданиями и элементами таксономий по программированию и алгоритмизации**

Отметим, что данная коллизия не возникает для достаточно большого количества заданий, которое обычно встречается в реальных банках КИМ, но уточнение параметров математических методов является одним из приоритетных направлений развития программного комплекса КИМРА.

#### Текущее состояние программного комплекса КИМРА

В настоящее время ведётся работа по развитию программного комплекса КИМРА в следующих направлениях:

- 1) тестирование и апробация второй версии подсистемы взаимного оценивания – плагина к *LMS Microsoft Classroom*;
- 2) разработка *Telegram*-бота, конвертирующего тестовые задания в формат *QTI/APIP*;
- 3) разработка визуального редактора *QTI/APIP*-совместимых тестовых заданий и его *RESTful API*;
- 4) разработка формата представления заданий по программированию в форме модульных тестов для хранения в банке КИМ информационной системы КИМРА;
- 5) разработка расширения для *IDE Microsoft Visual Studio* для решения заданий по программированию в форме модульных тестов.

### Перспективные направления развития темы

Нынешнее состояние тем, затронутых в диссертационном исследовании, выявленные ограничения позволяют выделить следующие направления для ближайшей дальнейшей работы.

1. Тестирование и усовершенствование математического обеспечения РИСУН КИМРА.
2. Разработка онтологии учебной дисциплины «Программирование».
3. Формализация бизнес-процессов методов активного обучения, разработанных для использования при обучении компьютерным наукам, наукам о данных и программной инженерии.
4. Интеграция с модулями ПСУН, поддерживающими режимы защищённого проведения компьютерных тестирований.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Роберт И.В., Лавина Т.А. Толковый словарь терминов понятийного аппарата информатизации образования. М: Институт информатизации образования (ИИО) РАО, 2006. 88 с.
2. Baker E.L. Ontology-based educational design: seeing is believing: Resource Paper No.13 // The Regents of the University of California, 2012.
3. Kabenov D., Muratkhan R., Satybaldina D., Razahova B. Ontology-based Testing System for Evaluation of Student's Knowledge // International Journal of Philosophy Study, Vol. 1, No. 1, 2013.
4. Boushman M., Labouidya O., El Kamoun N. Design of a Cloud Learning System Based on Multi-Agents Approach // International Journal of Advanced Computer Science and Applications, Vol. 6, No. 3, 2015.
5. Velez D.G. Challenges and Opportunities of Cloud-Based Mobile Learning // International Journal of Information and Education Technology, Vol. 4, No. 1, 2014.
6. Башмаков А.И., Башмаков И.А. Разработка компьютерных учебников и обучающих систем. М: Информационно-издательский дом "Филинь", 2003. 616 с.
7. Anson R., Goodman J.A. A Peer Assessment System to Improve Student Team Experiences // Journal of Education for Business, Vol. 89, No. 1, 2014. pp. 27-34.
8. Freeman M., McKenzie J. SPARK, A Confidential Web-Based Template for Self and Peer Assessment of Student Teamwork: Benefints of Evaluating across Different Subjects // British Journal of Educational Technology, No. 33, 2002. pp. 551-569.
9. Hyrynen V., Hamalainen H., Ikonen J., Porras J. MyPeerReview: An Online Peer-Reviewing System for Programming Courses // Proceedings of the

- 10th Koli Calling International Conference on Computing Education Research. Koli National Park. 2010. pp. 94-99.
10. Conlon M.P. RockTest: a programming contest management system // Journal of Computing Science in Colleges, Vol. 20, No. 5, 2005. pp. 27-35.
  11. Andrianoff S.K., Levine D.B., Gewand S.D., Heissenberger G.A. A testing-based framework for programming contests // Proceedings of the 2003 OOPSLA workshop on eclipse technology eXchange. 2003. pp. 94-98.
  12. Gewand S.D., Heissenberger G.Q. The Corona Project: a testing framework for programming contests // Journal of Computing Science in Colleges, Vol. 19, No. 5, 2004. pp. 348-349.
  13. Weiss D.J. Item Banking, Test Development, and Test Delivery // In: The APA Handbook on Testing and Assessment. Washington DC: American Psychological Association, 2011.
  14. Подбельский В.В., Максименкова О.В..Б.К.С. Об обеспечении интероперабельности программных средств поддержки тестирования в образовании // Информационные технологии, № 7, 2016. С. 535-541.
  15. Kurilovas E. Interoperability, Standards and Metadata for E-Learning // In: Intelligent Distributed Computing III. Berlin: Springer, 2009. pp. 121-130.
  16. IMS Global Learning Tools Interoperability Implementation Guide, Version 1.2 Final Specification [Электронный ресурс] // IMS Global Learning Consortium: [сайт]. [2015]. URL: <http://www.imsglobal.org/specs/ltiv1p2/implementation-guide> (дата обращения: 07.04.2017).
  17. Ed-Fi Alliance. Interchange Schema Documentation // Ed-Fi Data Standard v2.0. 2017. URL: <https://techdocs.ed-fi.org/display/EFDS20/Interchange+Schema+Documentation> (дата обращения: 28.04.2017).

18. Galton F. Co-Relation and their measurement, chiefly from antropometric data // Proceedings of the Royal society of London, Vol. 45, 1888-1889. pp. 135-145.
19. Spearman C. The proof and measurement of association between two things // The American journal of psychology, Vol. 15, No. 1, 1904. pp. 72-101.
20. Brown W. Some experimental results in the correlation of mental abilities // British journal of psycology, Vol. 3, No. 3, 1910. pp. 296-322.
21. Spearman C. Correlation calculated from faulty data // British journal of psychology, Vol. 3, No. 3, 1910. pp. 271-295.
22. Gulliksen H. Theory of Mental Tests. John Wiley & Sons, 1950. 504 pp.
23. Traub R.E. Classical test theory in historical perspective // Educational measurement: Issues and practice, Vol. 16, No. 4, 1997. pp. 8-14.
24. РМГ 29-99 ГСИ. Метрология. Основные термины и определения. Минск: ИПК Издательство стандартов, 2003. 65 с.
25. Stevens S.S. On the theory of scales of measurement // Science, Vol. 103, No. 2684, 1946. pp. 677-680.
26. Hambleton R.K., Swaminathan H., Rogers H.J. Fundamentals of item response theory. Newbury Park: SAGE, 1991.
27. Rasch G. Probabilistic models for some intelligence and attainment test. Copenhagen: Danish institute for educational research, 1960.
28. Нейман Ю.М., Хлебников В.А. Введение в теорию моделирования и параметризации педагогических тестов. Москва: Прометей, 2000. 168 с.
29. Карданова Е.Ю. Моделирование и параметризация тестов: основы теории и приложения. М.: ФГУ "Федеральный центр тестирования", 2008. 303 с.
30. Аванесов В.С. Item Response Theory: Основные понятия и положения. Статья первая // Педагогические измерения. 2007. № 3. С. 3-36.

31. Birnbaum A. Some latent trait models and their use in inferring an examinee's ability // In: Statistical theories of mental test scores / Ed. by Lord F.M., Novick M.R. MA: Addison-Wesley, 1968. pp. 397-479.
32. Fisher R.A. On the mathematical foundations of theoretical statistics // Philosophical transformations of the Royal society of London, Vol. 222, 1922. pp. 309-368.
33. Wainer H., Bradlow E.T., Wang X. Testlet response theory and its application. New York: Cambridge University press, 2007.
34. Magis D., Facon B. Angoff's delta method revisited: improving the DIF detection under small samples // British journal of mathematical and statistical psychology, Vol. 65, 2012. pp. 302-321.
35. Angoff W.H., Ford S.F. Item-race interaction on a test scholastic aptitude // Journal of educational measurements, Vol. 10, 1973. pp. 95-106.
36. Parshall C.G., Miller T.R. Exact versus asymptotic Mantel-Haenszel DIF statistics: A comparison of performance under small-sample conditions // Journal of educational measurement, No. 32. pp. 302-316.
37. // The R Project for Statistical Computing: [сайт]. URL: <http://www.r-project.org> (дата обращения: 04.09.2014).
38. Magis D., Facon B. deltaPlotR: An R package for differential item functioning analysis with Angoff's Delta Plot. // Journal of Statistical Software, Code Snippets, Vol. 59, No. 1, 2014. pp. 1-19.
39. Binet A., Simon T.A. Methodes nouvelles pour le diagnostic du niveau intellectuel des anormaux // l'Annet Psychologie, No. 11, 1905. pp. 191-336.
40. Van der Linden, W.J.; G.A.W. Glass. Computerized adaptive testing: theory and practice. Dordrecht: Kluwer academic publishers, 2000. 335 pp.
41. Thompson N.A., Weiss D.J. A framework for the Development of Computerized Adaptive Tests // Practical Assessment, Research & Evaluation, Vol. 16, No. 1, January 2011.

42. Budescu D. Efficiency of Linear Equating as a Function of the Length of the Anchor Test // *Journal of Educational Measurement*, Vol. 22, No. 1, 1985. pp. 13-20.
43. van der Linden W.J., Glas G.A.W. *Computerized Adaptive Testing: Theory and Practice*. Kluwer academic publishers , 2000.
44. Bradlow E.T., Wainer H., Wang X. A Bayesian random effects model for testlets // *Psychometrika*, Vol. 64, No. 2, 1999. pp. 153-168.
45. Wang X., Bradlow E.T., Wainer H. *A general Bayesian model for testlets: theory and applications*. Princeton, NJ: Educational testing service, 2002.
46. Lemann N. Behind the SAT // *Newsweek*, Vol. 134, No. 10, 1999. pp. 52-57.
47. Челышкова М.Б. Разработка педагогических тестов на основе современных математических моделей. Учебное пособие. – М.: Исследовательский центр, 1995. 32 с.
48. Куригин И.Н., Нардюжев В.И., Нардюжев И.В. Комплексная технология компьютерного тестирования // *Вестник РУДН Серия "Информатизация образования"*. 2013. № 2.
49. Lee H.S., Liu O.L., Linn M.C. Validating Measurement of Knowledge Integration in Science Using Multiple-Choice and Explanation Items // *Applied measurement in education* , Vol. 24, 2011. pp. 115–136.
50. Tasdemir M. A Comparison of Multiple-Choice Tests and True-False Tests Used in Evaluating Student Progress // *Journal of Instructional Psychology*, Vol. 37, No. 3, 2010. pp. 258-265.
51. Haladyna T.M. *Development and validating multiple-choice test items.. 3rd ed.* Lawrence Erlbaum associatedes, 2004.
52. Guler N., Uyanki G.K., Teker G.T. Comparison of classical test theory and item response theory in terms of item parameters // *European Journal of Research on Education*, Vol. 2, No. 1, 2014. pp. 1-6.



53. Noventa S., Stefanutti L., Vidotto G. An Analysis of Item Response Theory and Rasch Models Based on the Most Probable Distribution Method // *Psychometrika*, Vol. 79, No. 3, July 2014. pp. 377-402.
54. Bradshaw L., Templin J. Combining Item Response Theory and Diagnostic Classification Models: A Psychometric Model for Scaling Ability and Diagnosing Misconceptions // *Psychometrika*, Vol. 79, No. 3, July 2014. pp. 403-425.
55. Dorans N.J., Holland P.W. Population Invariance and the Equatability of Tests: Basic Theory and the Linear Case // *Journal of Educational Measurement*, Vol. 37, No. 4, 2000. pp. 281-306.
56. Gipps C.V. *Beyond Testing: Towards a Theory of Educational Assessment*. The Falmer Press, 1994.
57. Bjorner J.B., Kosinski M., Ware J.E. *Computerized adaptive testing and item banking*. Vol 2. Oxford: Oxford University Press, 2004.
58. Bjorner J.B., Chang C.H. Developing tailored instruments: item banking and computerized adaptive assessment // *Quality of Life Research*, Vol. 16, August 2007. pp. 95-108.
59. Vale C.D. Computerized Item Banking // In: *Handbook of Test Development*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., 2006. pp. 262-287.
60. Downing S.M., Haladyna T.M., Lane S., Raymond M. *Handbook of test development*. Lawrence Erlbaum Associates, 2006. 792 pp.
61. Weiss D.J. Item Banking, Test Development, and Test Delivery // In: *The APA Handbook on Testing and Assessment*. American Psychological Association, in press.
62. Dong Gi S. *Application of the Bifactor Model to Computerized Adaptive Testing*, University of Minnesota, Ann Arbor, Dissertations & Theses 3443919, 2011.

63. Koedsri A., Lawthong N., Ngudgratoke S. 5-th World Conference on Educational Sciences - WCES 2013 // Efficiency of Item Selection Method in Variable-Length Computerized Adaptive Testing for the Testlet Response Model: Constraint-Weighted A-Stratification Method. 2014. Vol. 116. pp. 1890-1895.
64. Paap M.C.S., Glas C.A.W. An Overview of Research on the Testlet Effect: Associated Features, Implications for Test Assembly, and the Impact of Model Choice on Ability Estimates, Law School Admission Council, Enschede, LSAC Research Report Research Report 13-03, 2013.
65. Bloom B.S. Taxonomy of Educational Objectives: the Classification of Educational Goals. Boston: Allyn and Bacon, 1956. 404 pp.
66. Gagne R. The conditions of learning. New York: Holt, Rinehart and Winston, 1965. 308 pp.
67. Scriven M. The methodology of evaluation // In: Perspectives of curriculum evaluation. Chicago, IL: Rand McNally, 1967. pp. 39-83.
68. Bonwell C.C., Eison J.A. Active Learning: Creating Excitement in the Classroom, School of Education and Human Development, The George Washington University, Washington, D.C., ASHE-ERIC Higher Education Report 1, 1991.
69. Dunn K.E., Mulvenon S.W. A Critical Review of Research on formative Assessment: The Limited Scientific Evidence of the Impact of Formative Assessment in Education // Practical Assessment, Research & Evaluation, Vol. 14, No. 7, 2009. P. 11.
70. Spiller D. Assessment: Feedback to Promote Student Learning Teaching Development. Wāhanga Whakapakari Ako, 2009. 19 pp.
71. Максименкова О.В., Незнанов А.А., Подбельский В.В. О формирующем контроле и информативной обратной связи при проектировании учебных курсов по программированию // Вестник Российского

университета дружбы народов. Серия: Информатизация образования, № 4, 2014. С. 37-48.

72. Evans C. Making Sense of Assessment Feedback in Higher Education // Review of Educational Research, Vol. 83, No. 1, 2013. pp. 70-120.
73. Falchikov N. Involving Students in Assessment // Psychology Learning and Teaching, Vol. 3, No. 2, 2004. pp. 102-108.
74. Falchikov N., Goldfinch J. Student Peer Assessment in Higher Education: A Meta-Analysis Comparing Peer and Teacher Marks // Review of Educational Research, Vol. 70, No. 3, 2000. pp. 287-322.
75. Dochy F., Segers M., Sluijsmans D. The Use of Self-, Peer and Co-assessment in Higher Education: a review // Studies in Higher Education, Vol. 24, No. 3, 1999. pp. 331-350.
76. Hundhausen C.D., Agrawal A., Agarwal P. Talking about Code: Integrating Pedagogical Code Reviews into Early Computing Courses // ACM Trans. Comput. Educ., Vol. 13, No. 3, August 2013. pp. 14:28.
77. van Zundert M., Sluijsmans D., van Merriënboer J. Effective Peer Assessment Processes: Research Findings and Future Directions // Learning and Instructions, No. 20, 2010. pp. 270-279.
78. Hamer J., Kell C., Spence F. Peer assessment using aropa // Proceedings of the ninth Australian conference on Computing education. Ballarat, Victoria. 2007. pp. 43-54.
79. Peters MA, Britez RG, editors. Open Education and Education for Openness. Rotterdam: Sense Publishers, 2008. 55 pp.
80. OER Commons [Электронный ресурс] // <https://www.oercommons.org>: [сайт]. [2016]. URL: <https://www.oercommons.org> (дата обращения: 17.08.2016).

81. What is open education? [Электронный ресурс] // opensource.com: [сайт]. [2016]. URL: <https://opensource.com/resources/what-open-education> (дата обращения: 01.12.2016).
82. Butcher N. A basic guide to Open Educational Resources (OER). Paris: commonwealth of Learning, 2015. 134 pp.
83. McGreal R, Kinuthia W, Marshall S, editors. Open Educational Resources: Innovation, Research and Practice. Vancouver: Commonwealth of learning, 2013. 268 pp.
84. Haggard S. The Maturing of the MOOC, Department for Business and Innovation Skills, Government of the UK, BIS Research Number 130, 2013.
85. Pangiotis A. Thirty Fourth International Conference on Information Systems // What Makes a Great MOOC? An Interdisciplinary Analysis of Student Retention on Online Courses. Milan. 2013. pp. 1-21.
86. Maher M.L., Latulipe C., Lipford H., Rorrer A. SICCSE'15 // Flipped Classroom Strategies for CS Education. Kansas City. 2015. pp. 218-223.
87. Cutrell E., O'Neill J., Bala S., Nitish B., Cross A., Gupta N., Kumar V., Thies W. L@S 2015 // Blended Learning in Indian Colleges with Massively Empowered Classroom. Vancouver. 2015. pp. 47-56.
88. РМГ 83-2007. Шкалы измерений. Термины и определения. М.: Стандартиформ, 2008.
89. Громова Л.А., Бавина П.А., Кондрашин А.В. Управление проектированием образовательных программ в рамках требований федеральных государственных образовательных стандартов высшего профессионального образования третьего поколения: Метод. пособие. СПб: Изд-во РГПУ им. А.И. Герцена, 2010.
90. European glossary on education. Volume 1 - Examinations, qualifications and Titles. 2nd ed. Euryclice European Unit, 2004. 268 pp.

91. Glossary of Standardized Testing Terms [Электронный ресурс] // ETS: [сайт]. [2016]. URL: [https://www.ets.org/understanding\\_testing/glossary/](https://www.ets.org/understanding_testing/glossary/) (дата обращения: 15.08.2016).
92. // Glossary of Important Assessment and Measurement Terms: [сайт]. URL: <http://ncme.org/resource-center/glossary/> (дата обращения: 4.Март.2014).
93. Аванесов В.С. Три источника становления метрической системы Георга Раша (RM) // Педагогические измерения, № 4, 2011. С. 18-29.
94. Анастаси А., Урбина С. Психологическое тестирование. М.: Директ-Медиа, 2008. 859 с.
95. American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. Standards for educational and psychological testing. Washington, DC: American Educational Research Association, 1999.
96. ГОСТ Р 50779.10–2000(ИСО 3534-1–93) Статистические методы. Вероятность и основы статистики. Термины и определения. М: Московский печатник, 2000. 42 с.
97. ГОСТ Р ИСО 5479-2002. Статистические методы. Проверка отклонения распределения вероятностей от нормального распределения. Москва: ИПК Издательство стандартов, 2002. 27 с.
98. ГОСТ Р 50779.21-96. Статистические методы. Правила определения и методы расчёта статистических характеристик по выборочным данным. М: ИПК Издательство стандартов, 1997. 48 pp.
99. Rada R. Efficiency and Effectiveness in Computer-Supported Peer-Peer Learning // Computers & Education, Vol. 30, No. 3, 1998. P. 137.
100. Hartel H. Efficiency of computer supported learning material (by integrating simulations and hypertext) // Frontiers in Education Conference, 1996. FIE '96. 26th Annual Conference. 1996.

101. Barrow L., Markman L., Rouse C. Technology's Edge: The Educational Benefits of Computer-Aided Instruction // American Economic Journal: Economic Policy, Vol. 1, No. 1, 2009. pp. 52-74.
102. The Knewton Blog [Электронный ресурс] // Knewton: [сайт]. [2017]. URL: <https://www.knewton.com/blog/adaptive-learning/> (дата обращения: 10.02.2017).
103. Neznanov A., Maksimenkova O. Blended Learning in Software Engineering Education: the Application Lifecycle Management Experience with Computer-Supported Collaborative Learning // Proceedings of 18th International Conference on Interactive Collaborative Learning (ICL 2015). Florence. 2015. pp. 646-653.
104. Kurpius S.E.R., Stafford M.E. Testing and measurement. A user-friendly guide. New York: SAGE, 2011. 184 pp.
105. Project results // HEI-UP Business Process Management in Higher Education Institution. 2015. URL: <http://www.bpm-hei.eu/index.php/news> (дата обращения: 19.08.2015).
106. Челышкова М.Б. Теория и практика конструирования педагогических тестов. М.: Логос, 2002. 432 с.
107. Майоров А.Н. Теория и практика создания тестов для системы образования. М.: Интеллект-центр, 2002. 296 с.
108. Bull J., McKenna C. Good Practice Guide in Question and Test Design. 2003.
109. Wierrma W., S.J. J. Educational measurement and testing. Allyn and Bacon, 1990.
110. Koch D.A. Testing goes graphical // Journal of Interactive Instruction Development, No. 5, 1993. pp. 14-21.
111. Parshall C.G., Stewart R., Ritter J. Annual meeting of the National Council on Measurement in Education // Innovations: Sound, graphics, and alternative response modes. New York. 1996.

112. Luecht R.M..C.B.E. ETS Colloquium on Computer-based testing: Building the foundation for future assessments // Test models of complex computer-based testing. Philadelphia, PA. 1998.
113. Help // Cruz roja Mexicana. URL: <http://www.cruzrojamexicanasonora.org.mx/capacitacion-en-linea/help.php?module=quiz&file=import.html> (дата обращения: 23.06.2015).
114. Aiken format // Moodle. 2015. URL: [https://docs.moodle.org/29/en/Aiken\\_format](https://docs.moodle.org/29/en/Aiken_format) (дата обращения: 12.02.2015).
115. GIFT format // Moodle. 2015. URL: [https://docs.moodle.org/29/en/GIFT\\_format](https://docs.moodle.org/29/en/GIFT_format) (дата обращения: 12.02.2015).
116. Moodle XML format // Moodle. 2015. URL: [https://docs.moodle.org/29/en/Moodle\\_XML\\_format](https://docs.moodle.org/29/en/Moodle_XML_format) (дата обращения: 25.05.2015).
117. Uploading questions // Blackboard Help. 2015. URL: [https://help.blackboard.com/en-us/Learn/9.1\\_SP\\_10\\_and\\_SP\\_11/Instructor/070\\_Tests\\_Surveys\\_Pools/106\\_Uploading\\_Questions#question\\_format](https://help.blackboard.com/en-us/Learn/9.1_SP_10_and_SP_11/Instructor/070_Tests_Surveys_Pools/106_Uploading_Questions#question_format) (дата обращения: 24.07.2015).
118. Documentation // SCORM Assessment Engine. 2015. URL: <https://docs.google.com/document/d/1zzLk8sp3MHsnLCNeAJbqRcMv-DIg8Y4J0oRAUro5gbk/edit> (дата обращения: 24.07.2015).
119. WebCT format // Moodle. 2015. URL: [https://docs.moodle.org/29/en/Import\\_questions#WebCT\\_format](https://docs.moodle.org/29/en/Import_questions#WebCT_format) (дата обращения: 21.05.2015).
120. IMS Question & Test Interoperability™ Specification // IMS Global learning consortium. 2012. URL: <http://www.imsglobal.org/question/> (дата обращения: 24.07.2015).
121. IMS Global learning consortium [Электронный ресурс] // IMS Global learning consortium: [сайт]. [2001-2015]. URL: <http://www.imsglobal.org/> (дата обращения: 24.07.2015).

122. Accessible Portable Item Protocol® (APIP®) // IMS Global learning consortium. 2014. URL: <http://www.imsglobal.org/apip/> (дата обращения: 24.07.2015).
123. McAlpine M. Design Requirements of a Databank, CAA Centre, Glasgow, Blueprint ISBN 1-904020-04-6, 2002.
124. Molina J.G., Sanmartin I.P.J. Modeling item banking: Analysis and design of a computerized system // Revista Electronica de Metodologia Aplicada, Vol. 13, No. 2, 2008. pp. 1-14.
125. Cella D., Gershon R., Lai J.S., Choi S. The Future of Outcomes Measurement: Item Banking, Tailored Short-Forms, and Computerized Adaptive Assessment // Quality of Life Research, Vol. 16, 2007. pp. 133-141.
126. Question bank // Moodle. 2015. URL: [https://docs.moodle.org/29/en/Question\\_bank](https://docs.moodle.org/29/en/Question_bank) (дата обращения: 24.07.2015).
127. Third-party question types // Moodle. 2015. URL: [https://docs.moodle.org/29/en/Third-party\\_question\\_types](https://docs.moodle.org/29/en/Third-party_question_types) (дата обращения: 25.07.2015).
128. Import questions // Moodle. 2015. URL: [https://docs.moodle.org/29/en/Import\\_questions](https://docs.moodle.org/29/en/Import_questions) (дата обращения: 25.07.2015).
129. About SEB // Safe Exam Browser. 2015. URL: [http://safeexambrowser.org/about\\_overview\\_en.html](http://safeexambrowser.org/about_overview_en.html) (дата обращения: 25.07.2015).
130. Quiz statistics report // Moodle. 2015. URL: [https://docs.moodle.org/29/en/Quiz\\_statistics\\_report](https://docs.moodle.org/29/en/Quiz_statistics_report) (дата обращения: 25.07.2015).
131. Respondus 4.0 // Respondus. Assessment tools for learning systems. 2015. URL: <https://www.respondus.com/products/respondus/> (дата обращения: 24.07.2015).
132. Introducing tao 3.0 // OAT. Home of tao. 2015. URL: <http://www.taotesting.com> (дата обращения: 26.07.2015).



133. RDF // W3C. Semantic Web. 2014. URL: <http://www.w3.org/RDF/> (дата обращения: 27.07.2015).
134. // Learnosity: [сайт]. [2015]. URL: <http://www.learnosity.com> (дата обращения: 28.07.2015).
135. СДО "Прометей" // Виртуальные технологии в образовании. 2015. URL: [http://www.prometeus.ru/actual/01\\_products/lms/opisanie.html](http://www.prometeus.ru/actual/01_products/lms/opisanie.html) (дата обращения: 28.07.2015).
136. Официальный сайт EdX // Официальный сайт EdX. URL: <https://www.edx.org/about-us> (дата обращения: 11.06.2014).
137. // Building and running an EdX course. URL: <http://edx.readthedocs.org/projects/edx-partner-course-staff/en/latest/index.html> (дата обращения: 06.10.2014).
138. Joint committee on standards for educational and psychological testing of the AERA, APA, and NCME. Standards for educational and psychological testing, American educational research association, Washington, DC, 1999.
139. Brennan R.L. Generalizability Theory and Classical Test Theory // Applied measurement in education, Vol. 24, 2011. pp. 1–21.
140. Wainer H, Braun HI, editors. Test validity. Routledge, 1988. 272 pp.
141. Cronbach L.J. Coefficient alpha and the internal structure of tests // Psychometrika, Vol. 16, No. 3, 1951. pp. 297-334.
142. Kuder G.F., Richardson M.W. The theory of the estimation of test reliability // Psychometrika, Vol. 2, No. 3, 1937. pp. 151-160.
143. Бодалев А.А., Столин В.В., Аванесов В.С. Общая психодиагностика. СПб.: Речь, 2000. 440 с.
144. Кендалл М., Стюарт А. Статистические выводы и связи. М: Наука, Физматлит, 1973. 899 с.

145. DiBattista D., Kurzawa L. Examination of the Quality of Multiple-choice Items on Classroom Tests // *The Canadian Journal for the Scholarship of Teaching and Learning*, Vol. 4, No. 2, 2011. P. Article 4.
146. Haladyna T.M., Downing S.M., Rodriguez M.C. A review of multiple-choice item-writing guidelines for classroom assessment. // *Applied Measurement in Education*, Vol. 15, No. 3, 2002. pp. 309-334.
147. Lord F.M. An Empirical Study of the Normality and Independence of Errors of Measurement in Test Scores // *Psychometrika*, Vol. 25, No. 1, March 1960. pp. 91-104.
148. Lord F.M. An analysis of the Verbal Scholastic Aptitude Test using Birnbaums' three-parameter logistic model, Educational Test Service, Princeton, NJ, ETS Research Bulletin RB-67-34, 1967.
149. Bock R.D. Fitting a response model for n dichotomous items // Annual meeting of the Psychometric Society. Madison, WI.
150. Andersen E.B. Asymptotic properties of conditional maximum likelihood estimators // *Journal of the Royal Statistical Society, Series B*, No. 32. pp. 283-301.
151. Mislevy R.J. Bayes modal estimation in item response models // *Psychometrika*, Vol. 51, 1986. pp. 177-195.
152. Swaminathan H., Gifford J.A. Bayesian estimation in the Rasch model // *Journal of Educational Statistics*, Vol. 7, 1982. pp. 175-191.
153. Swaminathan H., Gifford J.A. Bayesian estimation in the two-parameter logistic model // *Psychometrika*, No. 50, 1985. pp. 349-364.
154. Swaminathan H., Gifford J.A. Bayesian estimation in the three-parameter logistic model // *Psychometrika*, Vol. 51, 1986. pp. 589-601.
155. Urry V.W. Approximations to item parameters of mental test models and their uses. // *Educational and Psychological Measurement*, Vol. 34, 1974. pp. 253-269.

156. Urry V.W. ANCILLES: Item parameter estimation program with normal ogive and logistic three-parameter model options. Washington, DC: U.S. Civil Service Commission, Development Center, 1978.
157. McDonald R.P. Non-linear factor analysis (Psychometric Monograph No.15). Iowa City: Psychometric society.
158. McDonald R.P. Future directiona for item response theory // International Journal of Educational Research, Vol. 13, No. 2, 1989. pp. 205-220.
159. Кобзарь А.И. Прикладная математическая статистика. Для инженеров и научных работников. М.: Физматлит, 2006. 816 с.
160. Montgomery D.C., Runger G.C. Applied Statistics and Probability for Engineers. 6th ed. Wiley, 2013. 832 pp.
161. Wainer H., Kiely G. Item Clusters and Computerized Adaptive Testing: A Case for Testlets. // Journal of Educational Measurement, No. 24. pp. 185-202.
162. Samejima F. Estimation of Latent Ability Using a Response Pattern of Graded Scores (Psychometric monograph No.17). Richmond, VA: The William Byrd Press, 1969.
163. Reckase M.D. Designing item pools to optimize the functioning of a computerized adaptive test // Psychological Test and Assessment Modeling, Vol. 52, No. 2, 2010. pp. 127-141.
164. Van der Linden W.J., Veldkamp B.P. Constraining Item Exposure in Computerized Adaptive Testing With Shadow Tests // Journal of Educational and Behavioral Statistics, Vol. 29, No. 3, 2004. pp. 273-291.
165. Незнанов А.А., Максименкова О.В. Концептуальные и логические модели для распределённых программных систем поддержки контрольно-измерительных мероприятий // Программные системы: теория и приложения, Т. 8, № 4, 2017. С. 31-46.

166. Tanenbaum A.S., van Steen M. Distributed Systems: Principles and Paradigms. 2nd ed. Upper Saddle River, NJ: Pearson Education, 2016. 702 pp.
167. Hendrickson A., Huff K., Luecht R. Claims, Evidence, and Achievement-Level Descriptors as a Foundation for Item Design and Test Specifications // Applied measurement in education, Vol. 23, 2010. pp. 358–377.
168. Maksimenkova O., Podbelskiy V. Programming as a Part of the Software Engineering Education // SYRCOSE-2010. 2010. pp. 165-169.
169. Trant J. Studying Tagging and Folksonomy: A Review and Framework // Journal of Digital Infomation, Vol. 10, No. 1, 2009.
170. Lasić-Lazić J., Špiranec S., Ivanjko T. Tag-Resource-User: A Review of Approaches in Studying Folksonomies // Qualitative and Quantitative Methods in Libraries (QQML), Vol. 4, 2015. pp. 699-707.
171. Sharif A. Combining ontology and folksonomy: An Integrated Approach to Knowledge Representation // thThe Emerging Trends in Technology: Libraries between Web 2.0, Semantic Web and Search Technology. Milan. 2009.
172. Gasevic D., Zouaq A., Torniai C., Jovanovic J., Hatala M. An approach to folksonomy-based ontology maintenance for learning environments // IEEE transaction on learning technologies, Vol. 4, No. 4, 2011. pp. 301-314.
173. Подбельский В.В., Максименкова О.В., Казаков М.А. Визуализация структуры педагогического теста // Ярославский педагогический вестник, Т. III, № 1, 2012. С. 21-31.
174. Request for Comments (RFC) Pages // The Internet Engineering Task Force (IETF). URL: <http://ietf.org/rfc.html> (дата обращения: 08.11.2016).
175. Robles M., Braathen S. Online assessment techniques // The Delta Pi Epsilon Journal, Vol. 31, No. 4, 2002. pp. 465-478.

176. Benson A.D. Assessing participant learning in online environments // Facilitating Learning in Online Environments: New Directions for Adult and Continuing Education, Vol. 103, 2010. P. 69.
177. Gaytan J., McEwen B.C. Effective Online Instructional and Assessment Strategies // The American Journal of Distance Education, Vol. 21, No. 3, 2007. pp. 117-132.
178. When we share, everyone wins - Creative Commons [Электронный ресурс] // Creative Commons: [сайт]. URL: <https://creativecommons.org> (дата обращения: 21.10.2016).
179. Owen R.J. A Bayesian sequential procedure for quantal response in the context of adaptive mental testing // Journal of the American Statistical Association, No. 70, 1975. pp. 351-356.
180. Lord F.M., Novick M.R. Statistical theories of mental test scores. Reading, MA: Addison-Wesley, 1968.
181. Veerkamp W.J.J., Berger M.P.F. Item-selection criteria for adaptive testing // Journal of Educational and Behavioral Statistics, No. 22, 1997. pp. 203-226.
182. Klein E., Fox R.H., van der Linden W.J. A multivariate multilevel approach to simultaneous modeling of accuracy and speed on test items // Psychometrika, No. 74. pp. 21-48.
183. Rani M., Nayak R., Vyas O.P. An ontology-based adaptive personalized e-learning system, assisted by software agents on cloud storage // Journal Knowledge-Based Systems, Vol. 90, No. C, 2015. pp. 33-48.
184. Таха Х.А. Исследование операций. Вильямс, 2016. 912 с.
185. Abbott S, editor. Hidden curriculum // In: The glossary of education reform. 2014.
186. Липский В. Комбинаторика для программистов. Мир, 1988. 216 с.

187. Беллман Р. Динамическое программирование. М: Издательство иностранной литературы, 1960.
188. Sniedovich M. Dynamic programming: foundations and principles. 2nd ed. CRC, 2010. 624 pp.
189. Seidelman A.E. The Impact of Assessment Delivery Method on Student Achievement in Language Arts. 2014.
190. Delivering E-Learning: A Complete Strategy for Design, Application and Assessment. Kogan Page, 2009. 180 pp.
191. Assessment Systems. Test proctoring and security // Assessment Systems. URL: <http://www.assess.com/test-delivery> (дата обращения: 01.11.2016).
192. Baril J.L., Vajnovszki V. Gray Code for Derangements // Discrete Applied Mathematics, Vol. 1-3, No. 140, 2004. pp. 207-221.
193. Merlini D., Sprugnoli R., Verri M.C. An Analysis of a Simple Algorithm for Random Derangements // Proceedings of the 10th Italian Conference on Theoretical Computer Science (ICTCS'07). Rome. 2007. pp. 139-151.
194. Durstenfeld R. Algorithm 235: Random permutation // Communications of the ACM, Vol. 7, No. 7, 1964. pp. 420-420.
195. Кормен Т.Х., Лайзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ. М: "Вильямс", 2013. 1328 с.
196. Raman K., Joachims T. Methods for ordinal peer grading // Proceedings of the 20th ACM SIGKDD International conference on Knowledge discovery and data mining. New York. 2014. pp. 1037-1046.
197. Chuesathuchon C. EDU-COM 2008 International Conference. Sustainability in Higher Education: Directions for Change // Item Banking with Rasch Measurement: an Example for Primary Mathematics in Thailand. 2008. pp. 105-117.

198. Localization Platform for Translating Digital Content [Электронный ресурс] // Transifex: [сайт]. URL: <https://www.transifex.com> (дата обращения: 03.11.2016).
199. Максименкова О.В. Новые технологии в образовании // Вопросы локализации образовательного программного обеспечения с открытым кодом на примере EdX. Троицк-Москва. 2014. С. 497-499.
200. Salzberg B., Tsotras V.J. Comparison of access methods for time-evolving data // ACM Computing Surveys (CSUR), Vol. 31, No. 2, June 1999. pp. 158-221.
201. Allen G.N., March S.T. Modeling temporal dynamics for business systems // Journal of Database Management, Vol. 14, No. 3, 2003. pp. 21-36.
202. Announcing the Open Review Toolkit [Электронный ресурс] // Freedom to Tinker. Research and expert commentary on digital technologies in public life: [сайт]. [2016]. URL: <https://freedom-to-tinker.com/2016/11/27/announcing-the-open-review-toolkit/> (дата обращения: 5.12.2016).
203. Papushina I., Maksimenkova O., Kolomiets A. Digital Educational Mind Maps: a Computer Supported Collaborative Learning Practice on Marketing Master Program // Advances in Intelligent Systems and Computing. Interactive Collaborative Learning Proceedings of the 19th ICL Conference. 2016. Vol. 1. pp. 34-48.
204. Максименкова О.В., Папушина Ю.О. Формирующее оценивание при внедрении метода интеллект-карт в процесс обучения на магистерской программе «Маркетинг» // журнал Российского университета дружбы народов, № 3, 2015. С. 24-35.
205. Maksimenkova O., Podbelskiy V. On practice of using open data in construction of training and assessment tasks for programming courses // 10th International Conference on Computer Science & Education (ICCSE). Cambridge. 2015. pp. 233-236.

206. Radchenko I., Maksimenkova O. Principles of Citizen Science in Open Educational Projects Based on Open Data // Proceedings of the 12th Central and Eastern European Software Engineering Conference in Russia. 2016.
207. Максименкова О.В., Подбельский В.В. Практика использования открытых данных в курсе "Программирование" образовательной программы бакалавриата "Программная инженерия" // Образование и наука, Т. 139, № 10, 2016. С. 107-121.
208. Подбельский В.В., Максименкова О.В., Бабич К.С. Об обеспечении интероперабельности программных средств поддержки тестирования в образовании // Информационные технологии, № 7, 2016. С. 535-541.
209. Maksimenkova O., Neznanov A. The PASCA: A Mail Based Randomized Blinded Peer Assessment System for Complex Artifacts // Procedia Computer Science, Vol. 96, 2016. pp. 826-837.
210. Maksimenkova O., Neznanov A., Kolomiets A. On business processes of computer-supported collaborative learning: A case of peer assessment system development // Business Informatics, Vol. 38, No. 4, 2016. pp. 35-46.
211. Госстандарт России. ГОСТ Р ИСО/МЭК 12119-2000. Информационная технология. Пакеты программ. Требования к качеству и тестирование // ВНИИМС. URL: [http://www.vniims.ru/009lab/docs/12119\\_2000.pdf](http://www.vniims.ru/009lab/docs/12119_2000.pdf) (дата обращения: 15.12.2016).
212. Wright B.D., Stone M.H. Best Test Design. Chicago: Mesa Press, 1979. 240 pp.
213. McDonald R.P. Test theory: a unified treatment. Lawrence Erlbaum Associates, Inc. Publishers, 1999. 504 pp.
214. Irvine H.S., Kyllonen P.C. Item generation for test development. Lawrence Erlbaum associatedes, 2002.
215. Popham W.J. Test better, teach better : the instructional role of assessment. ASCD, 2003.



216. Lee G. A Disattenuated Part-Whole Correlation Formula // Journal of Educational Measurement, Vol. 37, No. 3, 2000. pp. 279-280.
217. Scalise K., Gifford B. Computer-Based Assessment in E-Learning: A Framework for Constructing "Intermediate Constraint" Questions and Tasks for Technology Platforms // The Journal of Technology, Learning, and Assessment, Vol. 4, No. 6, Jun. 2006.
218. Kang M., Heo H., Jo I., Shin J., Seo J. Developing an Educational Performance Indicator for New Millennium Learners // JRTE, Vol. 43, No. 2, 2011. pp. 157-170.
219. Kubiszyn T., Borich G. Educational testing and measurement: Classroom application and practice. John Wiley & Sons.
220. Parshall C.G., Houghton P.D.B., Kromrey J.D. Equating Error and Statistical Bias in Small Sample Linear Equating // Journal of Educational Measurement, Vol. 32, No. 1, 1995. pp. 37-54.
221. Pellegrino J.W., Quellmalz E.S. Perspectives on the Integration of Technology and Assessment // JRTE, Vol. 43, No. 2, 11. pp. 119–134.
222. Visone J.D. Science or Reading: What is Being Measured by Standardized Tests? // American Secondary Education 39, Vol. 39, No. 1, 2010.
223. Klein L.W., Jarjoura D. The Importance of Content Representation for Common-Item Equating with Nonrandom Groups // Journal of Educational Measurement, Vol. 22, No. 3, 1985. pp. 197-206.
224. Reiner C.M., Bothell T.W., Sudweeks R.R., Wood B. Preparing effective essay questions: A Self-directed Workbook for Educators. New Forums Press, 2002.
225. Mustakerov I., Borissova D. A conceptual approach for development of educational Web-based e-testing system // Expert Systems with Applications, No. 38, 2011.

226. Huang Y.M., Lin Y.T., Cheng S.C. An adaptive testing system for supporting versatile educational assessment // *Computers & Education*, No. 52, 2009. pp. 53–67.
227. Taylor C.S., Lee Y. Stability of Rasch Scales Over Time // *Applied measurement in education*, No. 23, 2010. pp. 87–113.
228. Lee W.C., Ban J.C. A Comparison of IRT Linking Procedures // *Measurement in Education*, Vol. 23, No. 1, Nov. 2010. pp. 23-48.
229. Kim S., von Davier A.A., Haberman S. Practical Application of a Synthetic Linking Function on Small-Sample Equating // *Applied measurement in education*, Vol. 24, 2011. pp. 95–114.
230. Rincon-Gallardo S., Quesenberry C.P., O'Reilly F.J. Conditional Probability Integral Transformations and Goodness-of-Fit Tests for Multivariate Normal Distributions // *The Annals of Statistics*, Vol. 7, No. 5, Sep. 1979. pp. 1052-1057.
231. Kallenberg W.C.M., Ledwina T. Consistency and Monte Carlo Simulation of a Data Driven Version of Smooth Goodness-of-Fit Tests // *The Annals of Statistics*, Vol. 23, No. 5, Oct. 1995. pp. 1594-1608.
232. ГОСТ Р 50779.42-99 (ИСО 8258-91). Статистические методы. Контрольные карты Шухарта. М.: ИПК Издательство стандартов, 1999. 32 с.
233. Thayn S. An Evaluation of Multiple Choice Test Questions Deliberately Designed to Include Multiple Correct Answers, Brigham Young University, Ann Arbor, Dissertations & Theses 3450158, 2011. 107 pp.
234. Sinharay S., Holland P. Choice of Anchor Test in Equating, Educational Testing Service, Princeton, Research & Development RR-06-35, 2006.
235. Spearman C. Demonstration of Formulæ for True Measurement of Correlation // *The American Journal of Psychology*, Vol. 18, No. 2, Apr. 1907. pp. 161-169.

236. Spearman C. "General Intelligence," Objectively Determined and Measured // The American Journal of Psychology, Vol. 15, No. 2, Apr. 1904. pp. 201-292.
237. Spearman C. Proof and Disproof of Correlation // The American Journal of Psychology, Vol. 16, No. 2, Apr. 1905. pp. 228-231.
238. Magno C. A brief history of educational assessment in the Philippines // Educational Measurement and Evaluation Review, Vol. 1, 2010. pp. 140-149.
239. Niu W. Western influences on Chinese educational testing // Comparative Education, Vol. 43, No. 1, 2007. pp. 71-91.
240. Keller W. Mapping Objects to Tables. A Pattern Language. // European Conf. on Pattern Languages of Programming. 1997.
241. Fowler M., al. E. Patterns of Enterprise Application Architecture. Addison-Wesley, 2007.
242. Micceri T. The Unicorn, The Normal Curve, and Other Inprobable Creatures // Psychological Bulletin, Vol. 105, No. 1, 1989. pp. 156-166.
243. Neyman J., Pearson E.S. On the problem of the most efficient tests of statistical hypotheses // Philosophical transactions of the royal society of London, Series A, Containing papers of a mathematical of physical character, Vol. 231, 1933. pp. 289-337.
244. Lord F.M. A Survey of Observed Test-Score Distributions with Respect to Skewness and Kurtosis // Educational and Psychological Measurement, Vol. 15, No. 4, December 1955. pp. 383-389.
245. История педагогической теории измерений [Электронный ресурс] // <http://testolog.narod.ru>: [сайт]. URL: <http://testolog.narod.ru/Theory77.html> (дата обращения: 15.04.2015).
246. Hamer J., Purchase H.C., Denny P., Luxton-Reily A. Quality of Peer Assessment in CS1 // 5th International Workshop on Computing Education Research. Berkeley. 2009. pp. 10-11.

247. Kollar I., Fischer F. Peer assessment as collaborative learning: A cognitive perspective // Learning and Instruction, Vol. 20, 2010. pp. 344-348.

## ПРИЛОЖЕНИЕ 1. ГЛОССАРИЙ

## П 1.1. Процессы и роли измерений в образовании

1. **Студент** [*student*] – участник учебного процесса высшего учебного заведения (среднего или общеобразовательного), проходящий обучение.
2. **Преподаватель** [*teacher*] – участник учебного процесса высшего учебного заведения (среднего или общеобразовательного), занимающийся подготовкой методического обеспечения, доставкой учебных материалов и организацией текущего и промежуточного контроля студентов.

## П 1.2. Образование и измерения в образовании

3. **Дистрактор** [*distractor, foil*] – неверный вариант ответа в списке ответов задания с выбором ответа.
4. **Измерение в образовании** [*educational measurement*] – процедура приписывания чисел некоторым характеристикам объектов в соответствии с определёнными правилами [25].
5. **Тестирование** [*testing*] – этап жизненного цикла теста, представляющий собой процедуру (доставку, прохождение и сбор результатов) проведения контрольно-измерительных мероприятий с использованием теста.
6. **Валидация теста** [*test validation*] – процесс получения показателей теста, которые разработчик планировал получить, подтверждающих, что результаты данного теста могут трактоваться именно так как планировал разработчик.
7. **Валидность теста** [*test validity*] – свойство измерительного инструмента, показывающее, что инструмент измеряет именно те характеристики, которые заявлены.
8. **Данные** [*data*] – 1) любая информация, представленная в форме, пригодной для хранения, передачи и обработки компьютерами; 2) представление информации в формализованном виде, пригодном для передачи, интерпретации или обработки [211].

9. **Дидактическая единица** – одна из предметных тем, подлежащих обязательному освещению в процессе подготовки специалистов, обучающихся по данной дисциплине (предмету).
10. **Надёжность теста** [*test reliability*] – устойчивость во времени результатов, полученных при помощи теста; внутренняя согласованность теста используется для определения того, что все задания теста направлены на измерение одного и того же результата.
11. **Дискриминативность** [*discrimination*] – различающая способность тестового задания. Способность задания отличать тех, кто знает от тех, кто не знает, тех, кто владеет навыком от тех, кто не владеет навыком и проч. в зависимости от того, на измерение чего нацелено тестовое задание.
- 11.1. **Коэффициент дискриминативности** [*discrimination index*] – количественная характеристика, описывающая дискриминативность тестового задания.
12. **Дифференцирующая способность** тестового задания [*differential item functioning (DIF)*] – см. дискриминативность.
13. **Жизненный цикл контрольно-измерительного материала (ЖЦ КИМ)** [*educational assessment material lifecycle, EAML*] – период времени между постановкой задачи на разработку или идеи создания КИМ до его удаления из хранилища КИМ.
14. **Контрольно-измерительный материал (КИМ)** [*educational assessment material, EAM*] - методический материал, который: 1) связан с какой-либо системой методических материалов, например, учебной дисциплиной [*course*]; 2) подразумевает возможность получения ответа [*answer*] студента на задание [*assignment*]; 3) имеет схему и правила оценивания [*grading rule*], позволяющие ставить в соответствие ответу качественную или количественную оценку [*grade*]; 4) обладает характеристиками корректности, трудности, дискриминативности и др.

15. **Классическая теория тестирования** [*classical test theory*] – психометрическая теория, представляющая факторы, воздействующие на эмпирически-наблюдаемый результат выполнения теста [138].
16. **Компьютерное адаптивное тестирование** [*computer adaptive testing, CAT*] – компьютерный тест, в котором задания и последовательность их представления зависят от содержания и свойств тестовых заданий, а также успешности ответов испытуемого.
17. **Компьютерный тест** [*computerized test, computer-based test*] – тест, управляемый компьютером.
18. **Компьютерное тестирование** [*computer-based testing*] – тестирование, при котором доставка, поддержка процесса прохождения и сбора результатов проводятся с использованием компьютера.
19. **Трудность тестового задания** [*item difficulty*] – свойство тестового задания, представляемое пропорцией испытуемых верно ответивших на задание к общему числу испытуемых.
- 19.1. **Коэффициент трудности ( $p$ -коэффициент)** [*difficulty index,  $p$ -value*] – количественная характеристика, описывающая трудность тестового задания.
20. **Тест** [*test*] – описание процедуры педагогических измерений (тестирования), состоящее из проверенных на валидность и надежность системы тестовых заданий, стандартизированной схемы проведения и заранее спроектированной технологии обработки и анализа полученных результатов.
21. **Тестлет** [*testlet*] – группа тестовых заданий, которая может быть разработана как единый блок и предполагает единую схему (сценарий) оценивания.
22. **Тестовое задание** [*test question, test item, item*] – элемент теста (утверждение, вопрос, упражнение, задание) и набор непротиворечивых правил его оценивания.
23. **Тестовое задание с альтернативным ответом** (син. Задание с альтернативным выбором, да/нет) [*true/false question*] – предполагает выбор, является ли утверждение истинным или ложным.

24. **Тестовое задание с множеством опций** (задание закрытого типа с одним правильным ответом) [*multiple choice question, MCQ*] – тестовое задание, в котором предлагается выбрать один верный ответ из списка предложенных.
25. **Тестовое задание с множеством альтернативных ответов** (син. Задание закрытого типа с множественным выбором) [*multiple response question, MRQ*]
- 25.1.1. (син. Англ. *Assertion-reason question*) Модифицированный вариант задания с альтернативным ответом, в котором предполагают выбор истинности или ложности для каждого варианта ответа в списке предложенных.
- 25.1.2. Модифицированный вариант задания с множеством опций, в котором предлагают указать несколько верных ответов из списка предложенных.
26. **Тестология** [*testology*] – междисциплинарная наука о создании качественных и научно обоснованных измерительных диагностических методик
27. **Педагогический тест** [*educational test*] – множество специально составленных заданий специфической формы, позволяющее по ответам испытуемых объективно измерить на определённой шкале их уровень подготовленности по конкретным разделам определённой области знаний.
28. **Первичный балл** [*raw score*] – не приведённая к шкале величина, показывающая достижение студента непосредственно сразу после измерения; результат непосредственного применения правила оценивания.
29. **Функция успеха** [*item response function*] – вероятность того, что определённый участник тестирования решит определённое задание.
30. **Банк тестовых заданий (БТЗ)** [*item bank*] –
- 30.1. пул тестовых заданий, обычно охватывающий полный набор заданий, разработанных для проведения тестирований по дисциплине (дисциплинам) или используемый для подбора заданий при помощи адаптивного алгоритма.
- 30.2. база данных, содержащая уникально-идентифицируемые тестовые задания и необходимые данные, позволяющие осуществлять выбор [*selection*] для нужд сборки тестов [123].



31. Бланковое тестирование [blank testing, paper and pencil testing] – традиционный вид тестирования, при котором испытуемый получает бланки или буклеты с тестовыми заданиями и заполняет их вручную ручкой или карандашом.

### П 1.3. Измерения

1. **Биофизическая шкала** [*biophysical scale*] – шкала измерений свойства физического фактора (стимула), модифицированная таким образом, чтобы по результатам измерений этого свойства можно было прогнозировать уровень или характер реакции биологического объекта на воздействие этого фактора [88].
2. **Значение величины** [*value of a quantity*] – выражение размера величины по соответствующей шкале в виде некоторого числа принятых единиц, чисел, баллов или иных знаков (обозначений) [88].
3. **Измерение** [*measurement*] – сравнение конкретного проявления измеряемого свойства (измеряемой величины) со шкалой (частью шкалы) измерений этого свойства (величины) в целях получения результата измерения (оценки свойства или значения величины) [88].
4. **Измеряемое свойство** [*measurable property*] – общее свойство объектов измерений, которое выбрано для исследования путём измерения [88].
5. **Измеряемая величина** [*measurable quantity*] – измеряемое свойство, характеризующее количественными различиями [88].
6. **Метод измерения** [*method of measurement*] – приём или совокупность приёмов сравнения конкретного проявления измеряемого свойства (измеряемой величины) со шкалой измерений этого свойства (величины) [88].
7. **Оценка свойства** [*evaluation of a property*] – выражение местоположения качественного свойства конкретного объекта измерений на соответствующей шкале наименований [88].
8. **Результат измерения** [*result of measurement*] – значение величины или оценка свойства, полученное(ая) путём измерений [88].

9. **Шкала наименований** [*scale of denominations*] – шкала измерения качественного свойства, характеризующаяся только соотношениями эквивалентности или отличиями проявления этого свойства [88].
10. **Шкала порядка** [*scale of an order*] – шкала измерений количественного свойства (величины), характеризующаяся соотношениями эквивалентности и порядка по возрастанию (убыванию) различных проявлений свойства [88].
11. **Шкала разностей** (интервалов, интервальная шкала) [*scale of differences (intervals)*] – шкала измерений количественного свойства (величины), характеризующаяся соотношениями эквивалентности, порядка, пропорциональности (допускающими в ряде случаев операцию суммирования) различных проявлений свойства [88].
12. **Номинальная шкала** [*nominal measurement scale*] – см. **Шкала наименований**.
13. **Шкала отношений** [*scale of ratios*] – шкала измерений количественного свойства (величины), характеризующаяся соотношениями эквивалентности, порядка, пропорциональности (в ряде случаев допускающими суммирование) различных проявлений свойства [88].
14. **Шкала** [*scale*] – отображение множества различных проявлений количественного или качественного свойства на принятое по соглашению упорядоченное множество чисел или другую систему логически связанных знаков (обозначений) [88].
15. **Шкала величины** [*scale of a quantity*] – шкала измерений количественного свойства [88].
16. **Спецификация шкалы измерений** [*specification for scale of measurements*] – принятый по соглашению документ, содержащий определение шкалы и (или) описание правил и процедур воспроизведения данной шкалы (или единицы шкалы, если она существует) [88].
17. **Ноль шкалы** [*zero of a scale*] – начальная точка шкал порядка (некоторых), интервалов, отношений и абсолютных шкал [88].

18. **Точка шкалы** [*point of a scale*] – одно отдельное число или знак (обозначение) из спецификации шкалы измерений [88].
19. **Единица измерений (величины)** [*unit of measurement*] – величина фиксированного размера, которой условно (по определению) присвоено числовое значение, равное единице в соответствующей шкале измерений) [88].
20. **Объект измерений** [*object of measurement*] – объект деятельности (тело, вещество, явление, процесс), одно или несколько конкретных проявлений количественных или качественных свойств которого подлежат измерению [88].

## ПРИЛОЖЕНИЕ 2. ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ

1. **БКИМ** – Банк Контрольно-Измерительных Материалов.
2. **БТЗ** – Банк Тестовых Заданий.
3. **ДЕ** – Дидактическая Единица.
4. **ЖЦ КИМ** – Жизненный Цикл Контрольно-Измерительного Материала.
5. **ИвО** – Измерения в Образовании.
6. **ИТЗ** – Инновационное Тестовое Задание.
7. **ИОС** – Информативная Обратная Связь.
8. **КИМ** – Контрольно-Измерительный Материал.
9. **КИМРА** – Контрольно-Измерительные Материалы Разработка и Анализ, название программного комплекса.
10. **ОЭ** – Открытая Экспертиза.
11. **ПСУН** – Программное Средство Учебного Назначения.
12. **ПСПИвО** – Программной Средство Поддержки Измерений в Образовании.
13. **РИСУН** – Распределённая Информационная Система Учебного Назначения.
14. **СВО** – сессия взаимного оценивания.
15. **СТТ** – *Classical Test Theory*, Классическая Теория Тестирования.
16. **ТЗ** – Тестовое Задание.
17. **IRT** – *Item Response Theory*, Современная Теория Тестирования, Теория Параметризации Педагогических Тестов).
18. **CAT** – *Computerized Adaptive Test* (Компьютерное адаптивное тестирование).
19. **DIF** – *Differential Item Functioning*.
20. **DDIF** – *dichotomous DIF*.
21. **MOOC** – *Massive Open Online Courses*
22. **OER** – *Open Educational Resources*.
23. **TRT** – *Testlet Response Theory* (Теория тестлетов).

### ПРИЛОЖЕНИЕ 3. РЕАЛИЗАЦИЯ АЛГОРИТМА РАНДОМИЗАЦИИ

```

Public Sub AuthorsRanomization(_
 ASH As Excel.Worksheet, AAutArr() As Variant, APeerCount As Integer)

 Dim ArrCount As Integer, i As Integer, j As Integer, u As Integer
 Dim k As Integer, Ind As Integer, RandomNum As Integer
 Dim Counters() As Integer, ReMap() As Integer

 Dim q As Long
 'Dim PeerArrs() As Long

 ASH.UsedRange.ClearContents ' Clear sheet
 ArrCount = UBound(AAutArr) + 1
 If ArrCount < APeerCount Then
 MsgBox "Authors count (" & ArrCount & ") less then peer count (" &
 APeerCount & ")!"
 Exit Sub
 End If
 '
 ReDim Counters(APeerCount)
 ReDim ReMap(ArrCount)
 ''' Int ReMap
 For i = 0 To ArrCount - 1
 ReMap(i) = i
 Next i

 Randomize
 For i = ArrCount - 1 To 1 Step -1
 'Remap current PRI
 Ind = Int((i - 1) * Rnd)
 k = ReMap(i)
 ReMap(i) = ReMap(Ind)
 ReMap(Ind) = k
 Next i
 ''' Init Counters
 For j = 0 To APeerCount - 1
 Counters(j) = j + 1
 Next j
 'Generate shift
 k = 1
 For i = 0 To ArrCount - 1
 ASH.Cells(k, 1) = AAutArr(i)
 For j = 0 To APeerCount - 1
 Counters(j) = Counters(j) + 1
 If Counters(j) >= ArrCount Then Counters(j) = 0
 ASH.Cells(k, j + 2) = AAutArr(ReMap(Counters(j)))
 Next j
 k = k + 1
 Next i
End Sub

```

```

Private Sub RemoveFromArr(ByRef AArr() As Long, AInd As Integer)
 Dim i As Integer
 For i = AInd + 1 To UBound(AArr)
 AArr(i - 1) = AArr(i)
 Next
 ReDim Preserve AArr(UBound(AArr))
End Sub

```

```

// Глобальные структуры данных
// Двумерный массив беспорядков ReMaps
// Первый индекс - номер перестановки (0 - само исходное множество)
// Второй индекс - номер элемента перестановки

```

```

// Проверка элементов предыдущих беспорядков
function CheckEqualityWithOtherMaps(
 ReMaps: array of array of Integer, // Массив беспорядков
 ACurDerangement: Integer, // Текущий беспорядок
 ARow: Integer, // Элемент исходного множества
 Avalue: Integer

```

```

): Boolean; // Результат проверки

```

```

var

```

```

 i: Integer;

```

```

begin

```

```

 Result := False;

```

```

 for i := 0 to ACurDerangement - 1 begin

```

```

 if ReMaps[ARow, i] = Avalue then begin

```

```

 Result := True;

```

```

 Break;

```

```

 end;

```

```

 end; //i

```

```

end;

```

```

// Генератор очередного беспорядка с номером ACurDerangement

```

```

procedure GenReMapOtherDR(

```

```

 ReMaps: array of array of Integer, // Массив беспорядков

```

```

 ACurDerangement: Integer);

```

```

var

```

```

 i, j, k: Integer;

```

```

 p, tmp: Integer;

```

```

 IsFound, IsFixed, IsOver: Boolean;

```

```

begin

```

```

 IsFound := False;

```

```

 while not IsFound do begin

```

```

 for j := 0 To High(ReMaps) do begin

```

```

 ReMaps[j, ACurDerangement] := j

```

```

 end; //j

```

```

 j := High(ReMaps);

```

```

 IsFixed := False;

```

```

 IsOver := False;

```

```

 while not IsOver do begin

```

```

 p := Int((j) * Rnd)

```

```

 Tmp := (p, ACurDerangement)

```

```

 if (Tmp = j) or

```

```

 CheckEqualityWithOtherMaps(ReMaps, ACurDerangement, j, Tmp) then begin

```

```

 IsFixed := True

```

```

 IsOver := True
 end else begin
 k := ReMaps[j, ACurDerangement];
 ReMaps[j, ACurDerangement] := ReMaps[p, ACurDerangement];
 ReMaps[p, ACurDerangement] := k
 end;
 j := j - 1;
 if j = 0 then begin
 IsOver := True;
 end;
end; // IsOver
Tmp := ReMaps[0, ACurDerangement];
if not IsFixed and (Tmp <> 0) and
 not CheckEqualityWithOtherMaps(ReMaps, ACurDerangement, 0, Tmp) then begin
 IsFound := True;
end;
end; // IsFound
end;

procedure AuthorsRandomization(
 // Массив идентификаторов объектов (множество Authors)
 AAutArr: array of Integer,
 APeerCount: Integer // Число рецензентов одного артефакта
);
var
 ArrCount: Integer;
 i, j, u: Integer;
 k: Integer;
 ReMaps: array of array of Integer;
begin
 ArrCount := High(AAutArr) + 1;
 if ArrCount < 6 then begin
 Message('Authors count (' + IntToStr(ArrCount) + ') too small (<6)!');
 Exit;
 end;
 if ArrCount < APeerCount then begin
 Message('Authors count (' + IntToStr(ArrCount) + ') less then peer count (' +
 IntToStr(APeerCount) + ')!')
 Exit;
 end;
 ReMaps[0] := AAutArr; // Load first array (Identifiers of Authors)
 // Remap Peers:
 for j = 0 to APeerCount-1
 GenReMapOtherDR(ReMaps, j);
 end; //j ' Count of Peers per artifact

 for i := 0 to High(ReMaps) do begin
 for j := 0 to High(ReMaps[i]) do begin
 ReMaps[i, j] = AAutArr[ReMaps[i, j]]
 end;
 end;
 // Copy ReMaps to destination
 ...
end;

```

## ПРИЛОЖЕНИЕ 4. РУБРИКИ ЗАДАНИЯ WEBQUEST

| Assingment | Fail (0) | Acceptable (0,25)                                                                                                                           | Satisfactory (0,5)                                                                                                                                      | Good (0,75)                                                                                                                                                     | Excellent (1)                                                                                                                              | Score |
|------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1          | Not done | The list contains less than 5 games. Examples and explanations are missed. List is bad formatted.                                           | The list contains more than 5, but less than 9 games. Some explanations are missed.                                                                     | All the 9 examples are listed. Mistakes in division into groups by 3, some links are missed.                                                                    | All the 9 examples are listed, and correctly divided into groups by 3. Links to sources are present.                                       |       |
| 2          | Not done | Niche and audience study is poor. The goals of the game are unclear and confusing. The required achievements are not study or poor studied. | Game positioning is wage (audience and niche are unclear). The goals are unclear. Required achievements are not studied or poor studied.                | Game is well positioned by niche and audience. The goals are classified and almost clear. Required achievements are listed without attempt of study or analyze. | Game is clearly positioned by niche and audience. The goals are well classified and clear. Required achievements are studied and analyzed. |       |
| 3          | Not done | Differences and similarities are not or poor described and analyzed. Matrix of features missed or bad-organized. List of features is wage.  | Differences and similarities are poor described and analyzed. Suggested a matrix of features. List of features does not reflect the games of this type. | Differences and similarities are described and analyzed. Suggested a matrix of features. The matrix is not easy to read and                                     | Differences and similarities are described and analyzed. Suggested well-formatted and clear matrix of features. List of features is full   |       |



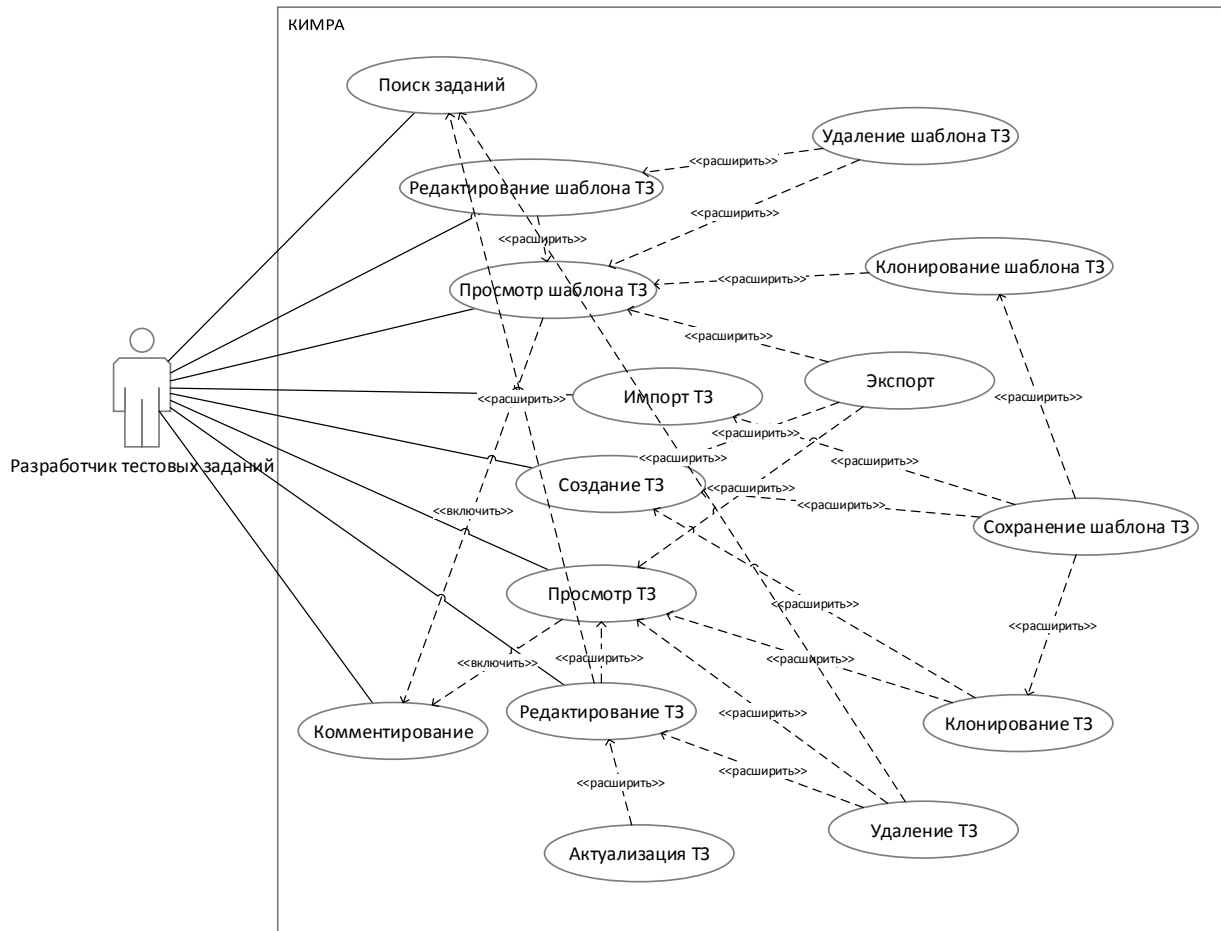
|   |                                                            |                                                                                                                                             |                                                                                                                                        |                                                                                                                                             |                                                                                                                  |  |
|---|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|--|
|   |                                                            |                                                                                                                                             |                                                                                                                                        | understand. List of features is not full, but partly reflects the games of this type.                                                       | and reflects the games of this type.                                                                             |  |
| 4 | Not done or original features are absurd and not explained | Less than 3 original features are suggested, explained and well-documented. The features are more or less relevant the genre and the niche. | 3 original features are suggested without explanations or explanation is absurd. The features seem to be relevant the genre and niche. | 3 original features are suggested, but poor explained and documented. The features are relevant the genre and niche.                        | 3 original features are suggested, explained and well-documented. The features are relevant the genre and niche. |  |
| 5 | Not done                                                   | UI patterns are listed. Examples and explanations are missed                                                                                | UI patterns are listed. Some of examples and explanations are missed.                                                                  | UI patterns are listed and well-classified. The examples are given.                                                                         | UI patterns are listed and perfectly classified. The examples are given.                                         |  |
| 6 | Not done                                                   | Input/output types are listed. Examples and explanation are missed or poor.                                                                 | Input/output types are listed. Some examples or explanation are given.                                                                 | Input/output types are well listed and described (for example, formatting is missing, explanations are wage, etc.). The examples are given. | Input/output types are perfectly listed and described. The examples are given.                                   |  |

|    |                                                           |                                                                                                                                                                      |                                                                                                                                                                  |                                                                                                                                                         |                                                                                                                                             |  |
|----|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|--|
| 7  | Not done or model or justification is absurd              | Given model is unclear, justification is missed. The explanation of project solution is given.                                                                       | Given model is clear, justification is partly missed or wage. The explanation of project solution is given.                                                      | Given model is not clear or and a justification is wage. The explanation of project solution is given.                                                  | Given a full clear and justified model of input/output for a designed application. The explanation of project solution is given.            |  |
| 8  | Not done or a set of UI pattern is not relevant or absurd | A set of UI patterns is poor and unclear. Examples and explanations are missed                                                                                       | A set of UI patterns is clear, but some patterns are not relevant the game. Examples and explanations are partly missed.                                         | A set of UI patterns is given and explained. Examples and explanations are partly missed.                                                               | A set of UI patterns is given and explained. Examples and prototypes are welcome.                                                           |  |
| 9  | Not done                                                  | Not all the game's engines used in the short-lists are studied. A pool of possible engines is not suitable for a designed game, the explanation missed or confusing. | All the game's engines used in the short-lists are studied. A pool of possible engines is not suitable for a designed game, the explanation missed or confusing. | All the game's engines used in the short-lists are studied. A pool of possible engines suitable for a designed game is not given and or poor explained. | All the game's engines used in the short-lists are studied. A pool of possible engines suitable for a designed game is given and explained. |  |
| 10 | Not done                                                  | A monetization model is suggested. Characteristics of the model are not                                                                                              | A monetization model is suggested. Characteristics of the model are exposed but                                                                                  | A monetization model is suggested. The justification is absent or                                                                                       | A monetization model is suggested and justified. The characteristics of the                                                                 |  |

|  |  |                    |                           |                                                                |                               |           |
|--|--|--------------------|---------------------------|----------------------------------------------------------------|-------------------------------|-----------|
|  |  | exposed or absurd. | leave gaps and confusing. | wage. The characteristics of the model are exposed in details. | model are exposed in details. |           |
|  |  |                    |                           |                                                                |                               |           |
|  |  |                    |                           |                                                                | <b>Total Score:</b>           | <b>10</b> |

## ПРИЛОЖЕНИЕ 5. АРТЕФАКТЫ ПРОЕКТИРОВАНИЯ ПО

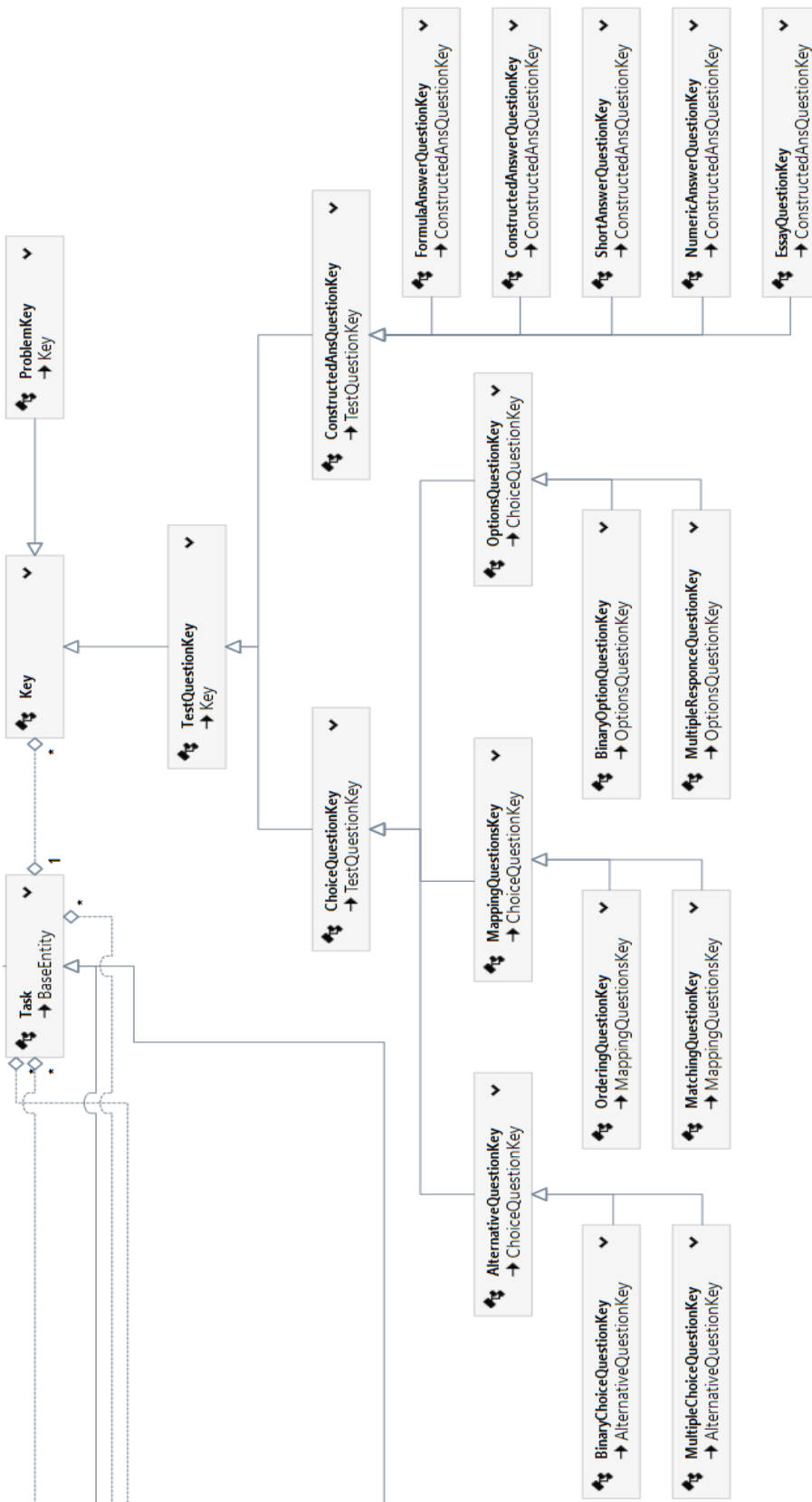
### П 1.4. Диаграммы прецедентов использования



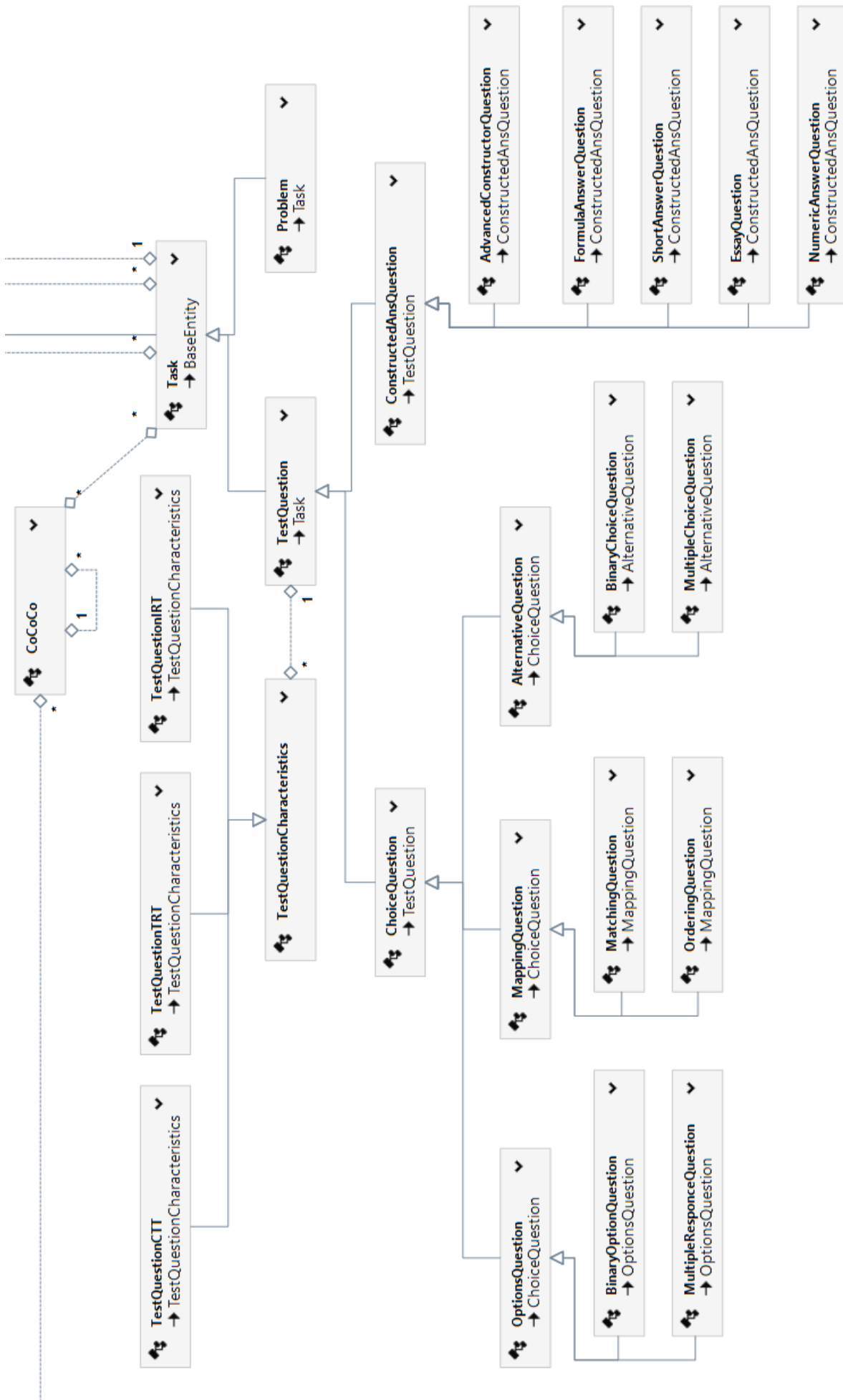


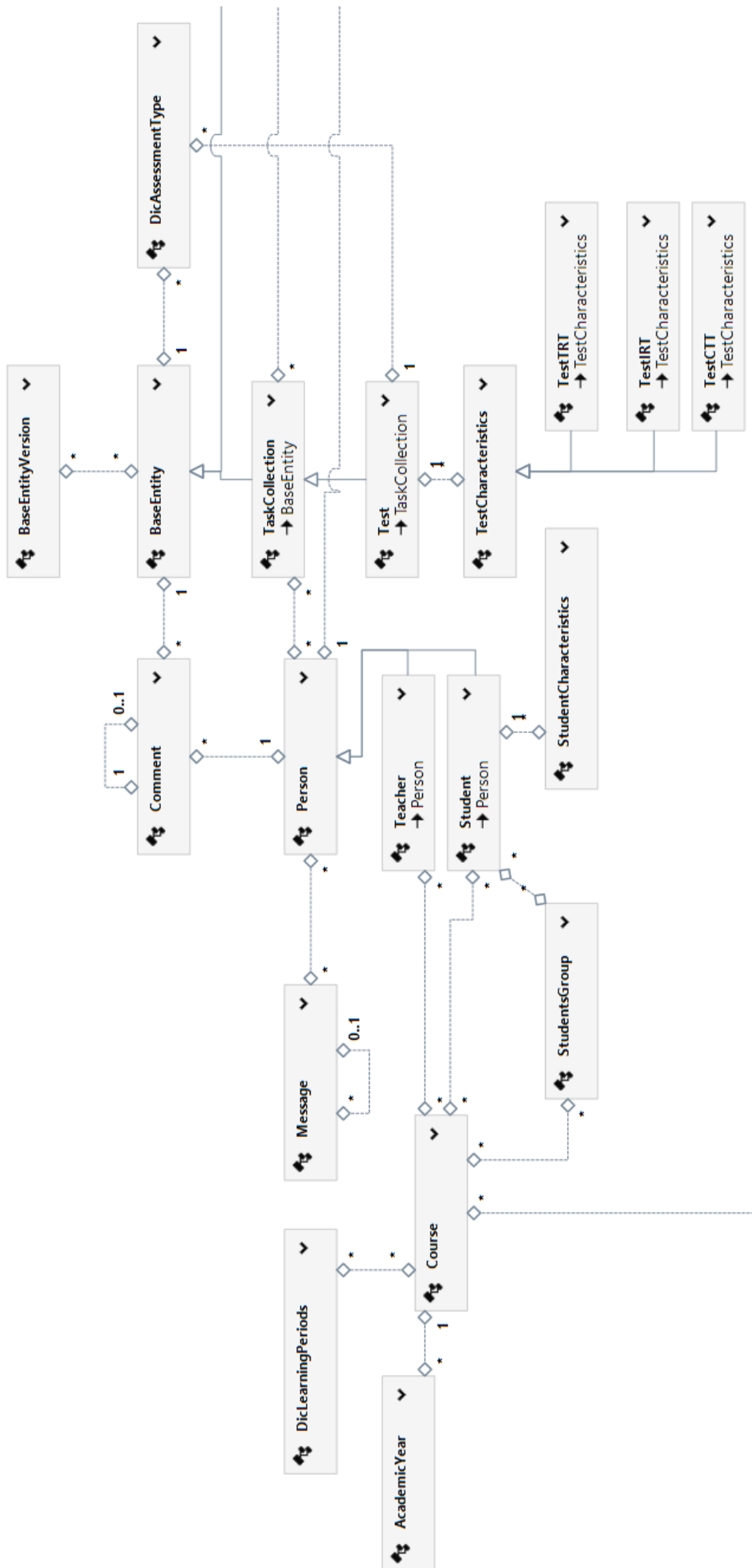


# ПРИЛОЖЕНИЕ 6. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ДАННЫХ









# ПРИЛОЖЕНИЕ 7. АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ КИМРА

