

А. А. Незнанов, О. В. Максименкова

Концептуальные и логические модели для распределённых программных систем поддержки контрольно-измерительных мероприятий

Аннотация. В статье рассмотрены подходы, модели и методы проектирования и реализации программных средств поддержки измерений в образовании как части комплексных систем поддержки учебного процесса. Основным результатом является формализация понятия контрольно-измерительного материала, его жизненного цикла и смежных понятий с технической точки зрения на современном этапе развития программной инженерии. Формализация проведена на концептуальном уровне с обсуждением некоторых вопросов реализации и обеспечения интероперабельности систем учебного назначения.

Ключевые слова и фразы: измерения в образовании, программирование, программное средство, контрольно-измерительный материал, контрольно-измерительное мероприятие.

Введение

Повышение эффективности современных сложных и разнородных учебных процессов 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*) [1, 2], обучение посредством облачных решений (*cloud-based learning*) [3, 4] и др. Работая над пониманием включения подсистем контроля знаний в надсистемы поддержки учебного процесса на разных уровнях или наличие межпрограммного взаимодействия между ними, в данной

Статья подготовлена в результате работы в рамках Программы фундаментальных исследований Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ) и с использованием средств субсидии в рамках государственной поддержки ведущих университетов Российской Федерации «5-100».

© А. А. Незнанов, О. В. Максименкова, 2017

© Высшая школа экономики, 2017

© Программные системы: теория и приложения, 2017

DOI: 10.25209/2079-3316-2017-8-4-31-46

работе сосредоточимся на изучении вопросов, связанных с программными системами, отнесённым в классификации А.И. Башмакова и И.А. Башмакова в [5] к компьютерным системам контроля знаний (КСКЗ). Появление распределённых и облачных решений размывло границы между указанными классами программных средств и вызвало необходимость существенного пересмотра классификации. Поэтому мы будем говорить о программных средствах поддержки измерений в образовании (ПСПИВО).

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 [6–8], программные комплексы сопровождения олимпиад [9–11] и др. Практически любое ПСПИВО, 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 [12].

Передача контрольно-измерительного материала из одной компьютерной системы в другую актуализирует вопросы интероперабельности ПСПИВО. На уровне реализации учебных компьютерных систем интероперабельность непрерывно совершенствуется. Соответствующие форматы для компьютерных тестовых систем обобщены О. В. Максименковой совместно с коллегами [13]. На мировом уровне приняты стандарты передачи данных и метаданных в системах электронного обучения [14–16]. 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 [15, 16].

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

- представление контрольно-измерительных материалов в компьютерных системах, независимо 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.

В контексте распределённых систем в образовании вопросы интеграции внутренних и внешних модулей приобрели особую актуальность и требуют решения, как на технологическом уровне, так и на уровне математического моделирования. Данная статья посвящена построению математических моделей, лежащих в основе распределённых программных систем поддержки контрольно-измерительных мероприятий, в которой контрольно-измерительный материал понимается как объект данных и согласован с требованиями к интероперабельности, предъявляемым к современным программным системам в образовании [15, 16].

Нам придётся начать с концептуального уровня, поскольку в российской литературе не отражены с точки зрения информационных технологий даже самые общие терминологические вопросы измерений в образовании, хотя бы немного выходящие за рамки педагогических тестов. При этом мы будем использовать классическую терминологию высшей школы (студент, преподаватель и т.п.), хотя всё сказанное будет верно для всех уровней обучения.

1. Основные термины и определения

Контрольно-измерительный материал (КИМ) — методический материал, который:

- (1) связан с системой методических материалов, например, учебной дисциплиной;
- (2) подразумевает возможность получения ответа студента на задание из КИМ;
- (3) имеет схему и правила оценивания, позволяющие ставить в соответствие ответу качественную или количественную оценку;

(4) обладает характеристиками корректности, трудности, дискриминативности и другими, устанавливаемыми по результатам анализа ответов.

Жизненный цикл контрольно-измерительного материала (ЖЦ КИМ) —

- (1) набор активностей, отношений и зависимостей, в которых участвует КИМ;
- (2) период времени от постановки задачи на разработку КИМ (или возникновением идеи его создания) до удаления КИМ из банка.

Конфигурация (*configuration*) — совокупность версий значимых артефактов в системе.

Версия (*version*) — состояние артефакта, которое может быть восстановлено в любой момент времени независимо от истории изменения системы.

Версионирование (*versioning*) — процесс создания, отслеживания и восстановления версий.

Контрольно-измерительное мероприятие определено в контексте образования как система подпроцессов учебного процесса, основная цель которой — выявление достигнутого уровня заявленных результатов обучения. Эти подпроцессы могут иметь дополнительные цели, а также интегрироваться с другими активностями в рамках учебного процесса.

Компьютерная доставка КИМ (*delivering*) (исходя из [19, 20] и др.) — 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.

Банк КИМ — база данных управления конфигурацией множества КИМ, совместно используемых для поддержки учебного процесса в целом и для проведения контрольно-измерительных материалов в частности.

Определение ЖЦ КИМ дано по аналогии с определениями жизненного цикла системы [17] и программного обеспечения [18]; оно действительно двойственно и уясняется из контекста. Хорошую методологическую основу для управления ЖЦ КИМ предлагает управление конфигурацией (*configuration management*) в рамках системной инженерии. Мы также упростили слишком формальные определения из ISO/IEC/IEEE 24765-2010 "Systems and software engineering — Vocabulary".

Банк (хранилище) КИМ (далее — банк) является важной частью ПСПИВО и образует основу системы поддержки ЖЦ КИМ. Авторами в литературе не обнаружено однозначно трактуемого определения банка в общем виде, то есть одновременно конструктивного, технологического и для произвольных типов КИМ. Для однозначности понимания мы ввели его, базируясь на определении банка тестовых заданий [21].

Банк тестовых заданий (БТЗ) — база данных, содержащая уникально-идентифицируемые тестовые задания и необходимые данные, позволяющие осуществлять выбор для нужд сборки тестов.

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. Специалистами в области организации массовых тестирований выделены этапы разработки тестов [12], 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 ПСПИВО или согласовании взаимодействия нескольких систем необходима детализация указанных процессов и понимание характеристик теста в каждом из них. Для обеспечения поддержки произвольных КИМ в распределённых ПСПИВО 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.

Перечисленные особенности приводят нас к пониманию ЖЦ КИМ, как классической модели многостадийного жизненного цикла версионизируемого информационного артефакта, встроенного в систему с отношениями часть/целое и детализации/обобщения:

$$AM_{LC} \stackrel{\text{def}}{=} \langle EAM, LCD, AResults, LO \rangle,$$

где

EAM — собственно КИМ;

AResults = $\langle AR_1, \dots, AR_{ARCount} \rangle$ — 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, возможные способы доставки и др.).

Так как каждый следующий результат в векторе *AResults* имеет увеличенную отметку времени, 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 ПСПИВУ, когда КИМ используется в различных, территориально удалённых пунктах и требуется специальная система синхронизации версионизируемого артефакта, в нашем случае КИМ [22].

3. Модель контрольно-измерительного материала

Повсеместная автоматизация, ещё до появления облачных и онтологически-управляемых решений, привела к тому, что исследователи в области образования были вынуждены признать необходимость согласования определений базовых сущностей процессов измерений в образовании, включённых в процессы автоматизации, с особенностями их компьютерных реализаций. Речь идёт о реструктуризации определения тестлетов для компьютерного адаптивного тестирования (КАТ). В 1999 году тестлет уже использовался как единица КАТ, но всё ещё определялся, как группа тестовых заданий в стандартизованном образовательном тесте, относящаяся к одной базовой части,

например, тексту или аудиозаписи [23]. Такое определение не позволяло описывать тестлет как информационный артефакт компьютерной системы, алгоритма или математически моделировать его. Новый смысл тестлет приобрел в 2007 году в работе Вайнера, Бредлоу и Ванга, которые дополнили характеристики тестовых заданий правилом оценивания (*scoring rule*) и расширили тестлет до группы тестовых заданий, которые разрабатываются как единый блок, и управляются совместно [24].

Определим основные составные части модели КИМ, отвечающей современным подходам активного обучения, осуществляемого частично или полностью при помощи компьютерных систем. Зачастую проектирование обучения начинается с формулировки целей и предполагаемых результатов обучения, для которых определяются формы контроля и формулируются КИМ [25]. Это означает, что уже на этапе идеи о создании КИМ известны и должны быть сохранены связи с результатами обучения, на измерение которых нацелен данный КИМ. Результаты обучения могут быть представлены произвольным образом: знаниями, умениями и навыками, компетенциями и проч.

Второй существенной частью модели является содержательная часть КИМ, необходимая для его понимания, выполнения и структурирования ответа. Это предполагает привязку КИМ к содержанию учебной дисциплины или её части. Членение учебного контекста эффективно организуется таксономически или онтологически, значимость подобного структурирования обоснована, например, в [26]. Мы предполагаем подразделять контекст КИМ на два типа: локальный и глобальный. Локальный контекст никак не привязывает КИМ к учебной дисциплине, учебной программе и прочим составляющим учебного процесса. 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 самостоятельность описания задания с точки зрения элементов описания КИМ. Максимально возможный контекст — образовательная программа. Для удобства проектирования компьютерных систем контекст может наследоваться, то есть допускает расширение. Например, контекст курсовой работы есть дисциплина или группа дисциплин, а контекст выпускной квалификационной работы — образовательная программа.

Третья часть КИМ, соответственно, включает элементы описания данного КИМ, необходимые для его автоматического сопоставления с таксономией или онтологией результатов обучения, идентификации в системе, соотношения с автором, сопоставления со статусами

и соотношения с временными метками. Исходя из вышесказанного, представляется разумным формализовать КИМ в виде четвёрки:

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

где

TaskDesc — кортеж элементов описания КИМ;

Body — тело КИМ, то есть содержательная часть КИМ, непосредственно используемое в контрольно-измерительном мероприятии;

CCCS — непустое множество проверяемых результатов обучения;

Pr — множество пререквизитов КИМ.

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

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, представляющая собой контейнер с данными, необходимыми для проведения контрольно-измерительного мероприятия и представляемая в виде:

$$\text{Body} \stackrel{\text{def}}{=} \langle \text{CEAM}, \text{Assignment}, \text{Context}, \text{GR} \rangle.$$

Компоненты *Body* имеют такой смысл: $\text{CEAM} = \{EAM_1, \dots, EAM_z\}$ — множество ссылок на вложенные КИМ. Если $\text{CEAM} = \emptyset$, 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 $\text{Assignment}_1, \dots, \text{Assignment}_z$ 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 вложенных КИМ $Context_1, \dots, Context_z$. GR — схема оценивания КИМ, являющаяся отображением результата выполнения КИМ в оценку по произвольной шкале. Эта функция может включать правила и сценарии (алгоритмы) оценивания, включая простейшие, при которых проверяется совпадение ответа на КИМ с шаблоном.

Возможные типы контейнеров: тестовое задание, задание для взаимного оценивания, индивидуальное задание по программированию, веб-квест, задание построения интеллект-карты и др.

4. Модель результата контрольно-измерительного мероприятия, использующего КИМ

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.

$$AM_{ER} \stackrel{\text{def}}{=} \langle (EAM_1 \dots EAM_z), Cohort, Answer, Env \rangle.$$

В случае композитного КИМ модель результата ссылается не только на сам КИМ, обозначенный EAM_1 , но 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.

5. Обсуждение реализации

Основой поддержки перечисленных выше моделей, объединённых моделью ЖЦ, является банк, обязательно включающий базу знаний, средства интеграции и обеспечения интероперабельности. Поскольку речь идёт о распределённых системах, архитектура банка также может быть распределённой. Банк могут составлять базы данных уже используемых в работе программных систем. Предложенные модели позволяют определить непротиворечивый программный интерфейс (*application programming interface, API*) банка, абстрагироваться от форматов данных и способов реализации в отдельных базах и управлять банком как единым целым.

Введённой модели КИМ, например, может быть поставлена в соответствие JSON-схема контейнера КИМ (приведена часть):

```
1  {
2    "$schema<<: >>http://json-schema.org/draft-04/schema#",
3    "title<<: >>EAM Container",
4    "description<<: >>Schema for a EAM Container",
5    "type<<: >>object",
6    "properties":
7    {
8      "allOf":
9      [
10     { "$ref<<: >>BEP.json" },
11     {
12       "properties":
13       {
14         "ContainerType":
15         {
16           "enum":
17           [
18             "TestItem",
19             "Test",
20             "TestResults", ...
21           ]
22         },
23         "Content": {"type<<: >>object"}
24       },
25       "required":
26       [
27         "ContainerType",
28         "Content"
29       ]
30     }
31   ]
32 }
33 }
34 {
35 "$schema<<: >>http://json-schema.org/draft-04/schema#",
36 "title<<: >>BEP",
37 "description<<: >>Schema for a base entity properties",
38 "type<<: >>object",
39 "properties": {
40   "Id": {"type<<: >>string"},
41   "TS_LastChange": {"type<<: >>string"},
42   "Author": {"type<<: >>string"}
43 }
44 ...
45 }
```

Отметим, что реально используемые современные стандарты интероперабельности LMS, такие как IMS LTI [15], не затрагивают этот важнейший вопрос, сосредотачиваясь на процессах поддержки ролей преподавателя и студента при оценивании и последующем распространении данных о проведённом оценивании.

Заключение

В современном учебном процессе повсеместно применяются разнообразные ПСПИВО. Особую ценность представляют накапливаемые в базах данных этих систем КИМ. Стремление участников учебного процесса к разнообразию форм обучения и контроля актуализировало вопросы интеграции уже существующих ПСПИВО, а стремительное развитие облачных технологий поставило задачи проектирования и разработки технологически новых ПСПИВО с учётом распределённости архитектуры.

В данной работе конструктивно введены существенные для автоматизации измерений в образовании понятия КИМ, ЖЦ КИМ и банка КИМ. Предложен вариант алгебраизации основных объектов данных распределённой ПСПИВО с целью единообразного математического описания операций над этими объектами, таких как наиболее распространённая операция *композиции* КИМ.

Описанная в работе модель КИМ снимает проблемы универсального хранения КИМ, версионирования КИМ в банке, а также облегчает постановку задачи автоматической генерации композитных КИМ. Параметры модели отражают этапы ЖЦ КИМ и обеспечивают целостность процессов, поддерживаемых ПСПИВО. Отметим, что открытой остаётся проблема совместимости КИМ разного типа, например, в рамках одного композитного КИМ. Разрешить эту задачу можно, например, введя ограничения на композицию, которые накладываются известными типами контейнеров, реализованными в конкретной ПСПИВО.

Наиболее актуальные и востребованные задачи образования полностью описываются предложенными моделями. Модель КИМ полностью согласована с требованиями к проектированию средств измерений в образовании, включая модели формирующего оценивания, в которых помимо связи с результатами обучения фигурирует обратная связь. Например, построение пулов заданий для проведения сеанса компьютерного адаптивного тестирования [23]. В пулах задания обобщаются по тематикам или алгоритмам оценивания (параметры CCCS и GR модели КИМ).

В настоящее время предложенные в статье модели положены в основу распределённого программного комплекса Контрольно-Измерительные Материалы Разработка и Анализ (КИМРА) отдельные компоненты которого реализованы, апробированы и описаны в работе [27].

Первостепенными направлениями дальнейшей работы авторы считают построение новых алгоритмов для предложенных моделей, оптимизацию существующих алгоритмов с учётом требований интероперабельности и их реализацию в рамках облачной инфраструктуры, например, на базе платформы Microsoft Office 365 Education [28].

Список литературы

- [1] E. L. Baker. *Ontology-based educational design: seeing is believing*, Resource Paper no. 13, The Regents of the University of California, 2012. ↑³¹
- [2] D. Kabenov, R. Muratkhan, D. Satybaldina, B. Razahova. “Ontology-based testing system for evaluation of student’s knowledge”, *International Journal of Philosophy Study*, 1:1 (2013), 8 p. ↑³¹
- [3] M. Boushman, O. Labouidya, N. El Kamoun. “Design of a cloud learning system based on multi-agents approach”, *International Journal of Advanced Computer Science and Applications*, 6:3 (2015), pp. 20–26. ↑³¹
- [4] D. G. Velev. “Challenges and opportunities of cloud-based mobile learning”, *International Journal of Information and Education Technology*, 4:1 (2014), pp. 49–53. ↑³¹
- [5] А. И. Башмаков, И. А. Башмаков. *Разработка компьютерных учебников и обучающих систем*, Информационно-издательский дом «Филинь», М., 2003, 616 с. ↑³²
- [6] R. Anson, J. A. Goodman. “A peer assessment system to improve student team experiences”, *Journal of Education for Business*, 89:1 (2014), pp. 27–34. ↑³²
- [7] M. Freeman, J. McKenzie. “SPARK, a confidential web-based template for self and peer assessment of student teamwork: benefits of evaluating across different subjects”, *British Journal of Educational Technology*, 33:5 (2002), pp. 551–569. ↑³²
- [8] V. Hyrynen, H. Hamalainen, J. Ikonen, J. Porras. “MyPeerReview: an online peer-reviewing system for programming courses”, *Proceedings of the 10th Koli Calling International Conference on Computing Education Research* (Koli National Park, Finland, 2010), pp. 94–99. ↑³²
- [9] M. P. Conlon. “RockTest: a programming contest management system”, *Journal of Computing Science in Colleges*, 20:5 (2005), pp. 27–35. ↑³²
- [10] S. K. Andrianoff, D. B. Levine, S. D. Gewand, G. A. Heissenberger. “A testing-based framework for programming contests”, *Proceedings of the 2003 OOPSLA workshop on eclipse technology eXchange* (Anaheim, California, October 2003), pp. 94–98. ↑³²

- [11] S. D. Gewand, G. Q. Heissenberger. “The Corona Project: a testing framework for programming contests”, *Journal of Computing Science in Colleges*, **19**:5 (2004), pp. 348–349. [↑] [32](#)
- [12] D. J. Weiss. “Item banking, test development, and test delivery”, *The APA Handbook on Testing and Assessment*, ed. Geisinger K. F., American Psychological Association, Washington DC, 2011. [↑] [32,35](#)
- [13] В. В. Подбельский, О. В. Максименкова, К. С. Бабич. «Об обеспечении интероперабельности программных средств поддержки тестирования в образовании», *Информационные технологии*, 2016, №7, с. 535–541. [↑] [32](#)
- [14] E. Kurilovas. “Interoperability, standards and metadata for e-learning”, *Intelligent Distributed Computing III*, Springer, Berlin, 2009, pp. 121–130. [↑] [32](#)
- [15] *IMS global learning tools interoperability implementation guide*, Version 1.2 Final Specification, IMS Global Learning Consortium, 2015 (data accessed: 07.04.2017). [↑] [32,33,41](#)
- [16] *Interchange schema documentation*, Ed-Fi Data Standard v2.0, Ed-Fi Alliance, 2017 (data accessed: 28.04.2017), URL: <https://techdocs.edfi.org/display/EFDS20/Interchange+Schema+Documentation> [↑] [32,33](#)
- [17] *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*, v. 1.8, BKCASE Editorial Board, 2017 (data accessed: 24.04.2017), URL: <http://www.sebokwiki.org> [↑] [35](#)
- [18] P. Bourque, R. Fairley (eds.). *Guide to the Software Engineering Body of Knowledge, Version 3.0*, IEEE Computer Society, 2014. [↑] [35](#)
- [19] K. Fee. *Delivering E-Learning: a complete strategy for design, application and assessment*, Kogan Page, 2009, 180 p. [↑] [34](#)
- [20] *Assessment Systems. Test proctoring and security*, Assessment Systems (data accessed: 01.11.2016), URL: <http://www.assess.com/test-delivery> [↑] [34](#)
- [21] M. McAlpine. *Design requirements of a databank*, Blueprint, CAA Centre, Glasgow, 2002, ISBN: 1-904020-04-6. [↑] [35](#)
- [22] A. S. Tanenbaum, M. van Steen. *Distributed systems: principles and paradigms*, 2nd ed., Pearson Education, Upper Saddle River, NJ, 2016, 702 p. [↑] [36](#)
- [23] E. T. Bradlow, H. Wainer, X. Wang. “A Bayesian random effects model for testlets”, *Psychometrika*, **64**:2, pp. 153–168. [↑] [37,41](#)
- [24] H. Wainer, E. T. Bradlow, X. Wang. *Testlet response theory and its applications*, Cambridge University Press, New York, NY, 2007. [↑] [37](#)
- [25] M. K. Tallent-Runnels, J. A. Thomas, W. Y. Lan, S. C. C. A. M. Shaw, X. Liu. “Teaching courses online: a review of the research”, *Review of Educational Research*, **76**:1 (2016), pp. 93–135. [↑] [37](#)
- [26] R. D. Tennyson, O. C. Park. “The teaching of concepts: a review of instructional design research literature”, *Review of Educational Research*, **50**:1 (2016), pp. 55–70. [↑] [37](#)

- [27] A. Kolomiets, O. Maksimenkova, A. Neznanov. “On business processes of computer-supported collaborative learning: a case of peer assessment system development”, *Business Informatics*, **38**:4 (2016), pp. 35–46. ↑⁴²
- [28] Microsoft Office 365 for education, Microsoft (data accessed: 28.04.2017), URL: <https://www.microsoft.com/enus/education/products/office/default.aspx> ↑⁴²

Рекомендовал к публикации

д.ф.-м.н. С. В. Знаменский

Пример ссылки на эту публикацию:

А. А. Незнанов, О. В. Максименкова. «Концептуальные и логические модели для распределённых программных систем поддержки контрольно-измерительных мероприятий», *Программные системы: теория и приложения*, 2017, **8**:4(35), с. 31–46.

URL: http://psta.psir.ru/read/psta2017_4_31-46.pdf

Об авторах:



Алексей Андреевич Незнанов

к.т.н., доцент, старший научный сотрудник международной научно-учебной лаборатории интеллектуальных систем и структурного анализа ФКН НИУ ВШЭ

e-mail: aneznanov@hse.ru



Ольга Вениаминовна Максименкова

младший научный сотрудник международной научно-учебной лаборатории интеллектуальных систем и структурного анализа ФКН НИУ ВШЭ

e-mail: omaksimenkova@hse.ru

Aleksey Neznanov, Olga Maksimenkova. *Conceptual and Logical Models of Assessment Materials for Distributed Educational Software Systems.*

ABSTRACT. The paper concerns on the approaches, models and methods for educational software design and development. It focuses on complex support of computerized educational measurements. The main result is a formalization of educational assessment material concept, its lifecycle and relevant concepts from the technological point of view. The formalization is given at conceptual level. Several questions about educational software interoperability and implementation are discussed. (*In Russian*).

Key words and phrases: educational measurement, programming, software, assessment, assessment material.

References

- [1] E. L. Baker. *Ontology-based educational design: seeing is believing*, Resource Paper no. 13, The Regents of the University of California, 2012.
- [2] D. Kabenov, R. Muratkhan, D. Satybaldina, B. Razahova. "Ontology-based testing system for evaluation of student's knowledge", *International Journal of Philosophy Study*, 1:1 (2013), 8 p.
- [3] M. Boushman, O. Laboudiya, N. El Kamoun. "Design of a cloud learning system based on multi-agents approach", *International Journal of Advanced Computer Science and Applications*, 6:3 (2015), pp. 20–26.
- [4] D. G. Velev. "Challenges and opportunities of cloud-based mobile learning", *International Journal of Information and Education Technology*, 4:1 (2014), pp. 49–53.
- [5] A. I. Bashmakov, I. A. Bashmakov. *The development of electronic textbooks and learning systems*, Informatsionno-izdatel'skiy dom "Filin", M., 2003 (in Russian), 616 p.
- [6] R. Anson, J. A. Goodman. "A peer assessment system to improve student team experiences", *Journal of Education for Business*, 89:1 (2014), pp. 27–34.
- [7] M. Freeman, J. McKenzie. "SPARK, a confidential web-based template for self and peer assessment of student teamwork: benefits of evaluating across different subjects", *British Journal of Educational Technology*, 33:5 (2002), pp. 551–569.
- [8] V. Hyrynen, H. Hamalainen, J. Ikonen, J. Porras. "MyPeerReview: an online peer-reviewing system for programming courses", *Proceedings of the 10th Koli Calling International Conference on Computing Education Research* (Koli National Park, Finland, 2010), pp. 94–99.
- [9] M. P. Conlon. "RockTest: a programming contest management system", *Journal of Computing Science in Colleges*, 20:5 (2005), pp. 27–35.
- [10] S. K. Andrianoff, D. B. Levine, S. D. Gewand, G. A. Heissenberger. "A testing-based framework for programming contests", *Proceedings of the 2003 OOPSLA workshop on eclipse technology eXchange* (Anaheim, California, October 2003), pp. 94–98.
- [11] S. D. Gewand, G. Q. Heissenberger. "The Corona Project: a testing framework for programming contests", *Journal of Computing Science in Colleges*, 19:5 (2004), pp. 348–349.

- [12] D. J. Weiss. “Item banking, test development, and test delivery”, *The APA Handbook on Testing and Assessment*, ed. Geisinger K. F., American Psychological Association, Washington DC, 2011.
- [13] V. V. Podbel'skiy, O. V. Maksimenkova, K. S. Babich. “On educational testing software interoperability support”, *Informatsionnyye tekhnologii*, 2016, no.7, pp. 535–541 (in Russian).
- [14] E. Kurilovas. “Interoperability, standards and metadata for e-learning”, *Intelligent Distributed Computing III*, Springer, Berlin, 2009, pp. 121–130.
- [15] *IMS global learning tools interoperability implementation guide*, Version 1.2 Final Specification, IMS Global Learning Consortium, 2015 (data accessed: 07.04.2017).
- [16] *Interchange schema documentation*, Ed-Fi Data Standard v2.0, Ed-Fi Alliance, 2017 (data accessed: 28.04.2017), URL: <https://techdocs.edfi.org/display/EFDS20/Interchange+Schema+Documentation>
- [17] *The Guide to the Systems Engineering Body of Knowledge (SEBoK)*, v. 1.8, BKCASE Editorial Board, 2017 (data accessed: 24.04.2017), URL: <http://www.sebokwiki.org>
- [18] P. Bourque, R. Fairley (eds.). *Guide to the Software Engineering Body of Knowledge, Version 3.0*, IEEE Computer Society, 2014.
- [19] K. Fee. *Delivering E-Learning: a complete strategy for design, application and assessment*, Kogan Page, 2009, 180 p.
- [20] *Assessment Systems. Test proctoring and security*, Assessment Systems (data accessed: 01.11.2016), URL: <http://www.assess.com/test-delivery>
- [21] M. McAlpine. *Design requirements of a databank*, Blueprint, CAA Centre, Glasgow, 2002, ISBN: 1-904020-04-6.
- [22] A. S. Tanenbaum, M. van Steen. *Distributed systems: principles and paradigms*, 2nd ed., Pearson Education, Upper Saddle River, NJ, 2016, 702 p.
- [23] E. T. Bradlow, H. Wainer, X. Wang. “A Bayesian random effects model for testlets”, *Psychometrika*, **64**:2, pp. 153–168.
- [24] H. Wainer, E. T. Bradlow, X. Wang. *Testlet response theory and its applications*, Cambridge University Press, New York, NY, 2007.
- [25] M. K. Tallent-Runnels, J. A. Thomas, W. Y. Lan, S. C. C. A. M. Shaw, X. Liu. “Teaching courses online: a review of the research”, *Review of Educational Research*, **76**:1 (2016), pp. 93–135.
- [26] R. D. Tennyson, O. C. Park. “The teaching of concepts: a review of instructional design research literature”, *Review of Educational Research*, **50**:1 (2016), pp. 55–70.
- [27] A. Kolomiets, O. Maksimenkova, A. Neznanov. “On business processes of computer-supported collaborative learning: a case of peer assessment system development”, *Business Informatics*, **38**:4 (2016), pp. 35–46.
- [28] Microsoft Office 365 for education, Microsoft (data accessed: 28.04.2017), URL: <https://www.microsoft.com/enus/education/products/office/default.aspx>

Sample citation of this publication:

Aleksey Neznanov, Olga Maksimenkova. “Conceptual and Logical Models of Assessment Materials for Distributed Educational Software Systems”, *Program systems: Theory and applications*, 2017, **8**:4(35), pp. 31–46. (In Russian).

URL: http://psta.pspiras.ru/read/psta2017_4_31-46.pdf