

## НЕКОТОРЫЕ ВОПРОСЫ АВТОМАТИЗАЦИИ КОНТРОЛЯ ЗНАНИЙ\*

Миннегалиева Чулпан Бакиевна, Мухамедшин Дамир Рафкатович,  
Русецкий Кирилл Валерьевич, Паркалов Алексей Викторович

### Аннотация

В статье рассматриваются проблемы автоматизации контроля знаний. Предлагается метод генерации заданий с использованием переменных разного типа при помощи системы ExerciseBook. Приводятся примеры составления заданий. Описывается способ предварительной оценки деятельности пользователя через историю взаимодействия с системой.

**Ключевые слова:** *электронное обучение, генерация заданий, автоматический контроль знаний, электронный образовательный ресурс.*

В составе электронных образовательных ресурсов, обучающих систем, дистанционных курсов присутствует блок контроля знаний. Автоматизация контроля знаний имеет ряд проблем, пути решения которых в настоящее время изучаются разными специалистами. Повышение объективности проверки знаний является одной из главных задач. Для этой цели работа ведется в нескольких направлениях, среди них наполнение базы заданий и отслеживание деятельности пользователя.

В данной работе описаны возможности системы контроля знаний по математике ExerciseBook, пробная версия которой разработана нами при помощи языка PHP с использованием СУБД MySQL. Для отображения и обмена информацией использовались фреймворки Bootstrap и jQuery. Система обеспечивает генерацию заданий и автоматическую проверку ответов, также содержит инструменты для определения степени самостоятельности выполнения заданий. Архитектура системы построена на концепции MVC (Model-view-controller «модель-представление-поведение»). MVC — схема использования нескольких шаблонов проектирования, с помощью которых модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные.

Основной задачей являлся поиск новых приемов генерации заданий. Отметим, что при работе с математическими заданиями приходится уделять постоянное внимание на корректное отображение для пользователя формул, матриц, векторов и т. д. Также в ходе разработки системы решалась проблема повышения объективности контроля знаний через отслеживание истории взаимодействия пользователя с системой.

---

\* Работа выполнена при финансовой поддержке РФФИ (проект № 15-57-04085) «Модели и средства мультимедального синтеза текстов для интеллектуальных обучающих систем».

Одной из составляющих блока контроля знаний является база заданий, учебно-тренировочных задач. Тестирование становится органической частью современного образовательного процесса [11]. На сегодняшний день очевидно, что база заданий для тестирования должна содержать большое число упражнений, задач, вопросов. На практике продумывание, составление, проверка большого количества заданий любого типа — достаточно трудоемкий процесс. Таким образом, актуальной является проблема генерации задач различной сложности для автоматизации проверки знаний.

Генерации задач посвящен целый ряд работ, в которых рассматриваются общие вопросы, описываются различные подходы и предлагаются конкретные решения для использования на практике. Обзор средств генерации приведен в [9]. В результате анализа существующих и используемых в различных системах способов автор выделил несколько подходов к созданию генераторов: программируемые генераторы, параметрические генераторы, редакторы генераторов, выбирающие генераторы. В работе предложен метод создания генераторов на основе предметно-ориентированного языка, поддерживающего использование систем компьютерной алгебры и компьютерной вёрстки. Также описаны инструментальные средства для автоматизации процесса подготовки заданий и для создания генераторов пользователями высокой квалификации. Язык создания генераторов подробно представлен в [10]. В работе [2] автором представлен способ описания заданий генераторного типа с помощью деревьев И/ИЛИ и язык создания алгоритмов генерации тестовых заданий GILT, а также его транслятор. Статья [3] посвящена обзору систем разработки многовариантных тестовых заданий Platan и Possum, приведены примеры заданий по алгебре, физике, макроэкономике, математической логике и другим дисциплинам. С появлением систем компьютерной алгебры преподаватели изучают их возможности для генерации заданий. Например, рассматривается создание в среде математического пакета MathCAD системы документов для генерации, сохранения и проверки тестовых заданий на установление соответствия [8].

Грамотное включение сгенерированных заданий в учебный процесс является не менее важным вопросом и рассматривается разными авторами. Обращается внимание на то, что использование генератора позволяет не только создавать индивидуальные задания, но и сопровождать их обучающими функциями посредством соответствующих указаний, помогающих студенту наметить план решения и осуществить его [4]. Одной из наиболее эффективных форм работы со студентами является индивидуальная работа. Отмечается, что время на реализацию индивидуальной работы преподаватель может выделить за счет использования различных технологических средств и приемов, среди таких технологий авторы называют использование открытых параметрически генерируемых материалов [7].

Методы генерации учебно-тренировочных задач рассматривали Башмаков А.И. и Башмаков И.А. в [1]. Модель учебно-тренировочной задачи данными авторами изучается как кортеж, состоящий из цели, исходных данных, ограничений, модели ситуации, результатов (ответов), эталонной модели деятельности, функции оценивания результатов, функции оценивания деятельности [1, с 445]. Остановимся на модели ситуаций. Она включает структуру, которая определяет форму выражений (математических зависимостей, правил и т. д.), входящих в модель ситуации, значения параметров, интерпретацию. Подобная декомпозиция позволяет дифференцировать генерацию учебно-тренировочных задач путем варьирования значений параметров при неизменной структуре модели ситуации и генерацию, предусматривающую трансформацию структуры. Интерпретация характеризует смысл (то есть предметное содержание) элементов структуры и значений параметров. В рамках этой работы данная составляющая рассматриваться не будет.

Рассмотрим генерацию структуры и параметров. Генерация значений параметров в настоящее время возможна в разных системах поддержки обучения. Например, при работе в системе Moodle можно создать «вычисляемый вопрос» с использованием чисел, выбираемых случайным образом из определенного набора при прохождении теста [6]. При само-

стоятельной разработке обучающей программы или программы для тестирования с помощью современных языков программирования генерация параметров тоже легко реализуема. Генерация структуры задачи является более сложной проблемой, но обеспечивает разнообразие заданий, повышение уровня объективности контроля. Приемы генерации структуры и параметров математических задач с использованием системы компьютерной алгебры Mathematica рассмотрены нами в [5]. Отметим также, что разработчик системы Mathematica компания Wolfram Research на своем сайте предоставляет возможность пройти тестирование по сгенерированным заданиям [12].

Для решения проблемы автоматического составления необходимого количества заданий мы использовали переменные, генерируемые «на лету». Каждая отдельная переменная в системе является объектом. Для этого объекта реализована специальная модель, предоставляющая функционал для подсчета значения переменной и манипуляций с объектом. Модель переменной содержит в себе такие свойства, как идентификатор переменной, название, идентификатор задачи, в рамках которой переменная применяется, тип переменной, параметры, подсчитанное значение.

При составлении шаблонов заданий пользователь (преподаватель) имеет возможность создать переменные разных типов. Первый тип переменной — «случайное число», значение при этом может быть целым числом или десятичной дробью. Второй тип — «случайное из списка». В этом случае указывается список значений и количество возвращаемых значений. Еще один тип — это «очередь», указывается список значений, необходимость возвращения использованных значений в конец очереди или отсутствие такой необходимости. Тип переменной, существенно расширяющий возможность генерации заданий, — это «произвольная функция». Значением функции может быть целое или дробное число, матрица, вектор, комплексное число, произвольное значение (строка). Данный тип предоставляет возможность генерировать разные данные, используя язык РНР и дополнительные функции для работы с матрицами, комплексными числами и т. д. Все перечисленные переменные могут быть использованы в тексте задания, вариантах ответов и в помощи к заданию. Преподаватель имеет возможность включать созданные переменные в другие задания, то есть использовать их многократно.

Использование разных типов переменных позволяет генерировать задания. В начале работы создается достаточное количество переменных типа «случайное число» или «случайное из списка». Диапазон изменения параметров можно настроить отдельно для каждой переменной. Затем создаются переменные «произвольная функция». В системе проверки знаний к уже созданным переменным можно обращаться как к аргументам функций, этим достигается достаточная сложность структуры задания.

Преподаватель может работать с обычными заданиями для тестирования, где не предусмотрена генерация. Если рассматривать генерацию, то для разработки простого задания автор задания должен определить переменные, значения которых будут генерироваться, и задать соответствующие настройки. Например, задание по теме «Скалярное произведение векторов». Введена переменная  $g$ , тип переменной — «случайное из списка», тип значения — целое число (можно указать дробное), указан список значений (рис. 1). Аналогично введены переменные  $g1$  и  $f1$ . В задании переменная  $g1$  используется дважды в целях рационального использования переменных, можно было ввести еще одну переменную. Координаты векторов будут принимать значения данных переменных.

На вкладке «Задание» вводится текст задания: «Найти скалярное произведение векторов  $a$  и  $b$ , если даны координаты векторов:  $a$  ( $@g$ ,  $@g1$ ) и  $b$  ( $@g1$ ,  $@f1$ )», при оформлении задания предусмотрено предъявление списка переменных (подсказка) пользователю, возможно добавление рисунка, таблицы, изменение шрифта и т. д. (рис. 2).

На вкладке «Ответы» заполняются варианты ответов, в данном случае —  $@g*@g1+@g1*@f1$  (рис. 3).

7

Задание    Ответы    **Переменные**    Помощь    Дополнительно

@g

Имя переменной:     Тип переменной:     Тип значения:

Список значений:

<input type="text" value="3"/>	<input type="text" value="4"/>
<input type="text" value="5"/>	<input type="text" value="6"/>
<input type="text" value="7"/>	<input type="text" value="8"/>
<input type="text" value="Добавьте значение"/>	

Количество возвращаемых значений:

Рис. 1

7

Задание    Ответы    Переменные    Помощь    Дополнительно

Файл    Изменить    Вставить    Вид    Формат    Таблица

Найти скалярное произведение векторов **a** и **b**, если даны координаты векторов: **a** (@g, @g1) и **b** (@g1, @f1)

Р

Сохранить

Рис. 2

7

Задание    **Ответы**    Переменные    Помощь    Дополнительно

Тип ответа:     Проверка ответа:     Точность проверки:

Правильные ответы:    

Показывать ученику результаты проверки ответа     Подсчитывать значения формул

Рис. 3

Если на странице «Помощь» набрать необходимый текст, у обучаемого появится возможность посмотреть подсказку, перейдя по ссылке. Вкладка «Дополнительно» отвечает за настройки, в числе которых доступность, количество баллов за верно выполненное задание, ограничение по времени, количество попыток, уровень сложности. В результате каждому обучаемому будет предложено индивидуальное задание (рис. 4).

Таким образом, для составления несложных заданий автор не должен обладать специальной квалификацией. Если же пользователь ознакомится с функциями РНР, среди кото-

рых есть математические функции и функции из расширений, возможности использования системы увеличиваются. Например, следующее задание: «Округлить данное число до целых». Вводится переменная  $x2$ , тип переменной — «случайное число», тип значения — «число с плавающей точкой (дробное)», ограничения, например, от 1 до 9, количество знаков после запятой – 3. Добавляется переменная  $roundx2$ , тип переменной — «произвольная функция», в соответствующее поле вводится код функции: `return (round(@x2))` (рис. 5). Тогда в ответе надо будет указать: « $@roundx2$ ».

Приведем пример работы с матрицами. Пусть  $m1$  — переменная «произвольная функция» с типом значения «матрица», генерирующая матрицу второго порядка. Тогда в соответствующем поле необходимо набрать код: `return [randomSample (1,10,2,1), randomSample (1,10,2,1),]. m2` — переменная тоже «произвольная функция», вычисляющая определитель  $m1$ , код функции: `return determinant(@m1)`. Формулировка задания: «Вычислить определитель матрицы  $@m1$ », ответ —  $@m2$ . Обучаемые получают разные задания, в которых предлагается вычислить определитель матрицы (рис. 6).

Еще один простой пример, демонстрирующий возможность изменения структуры задания. Введены две переменные  $x$  и  $y$ , которые принимают случайные целые значения. Пере-

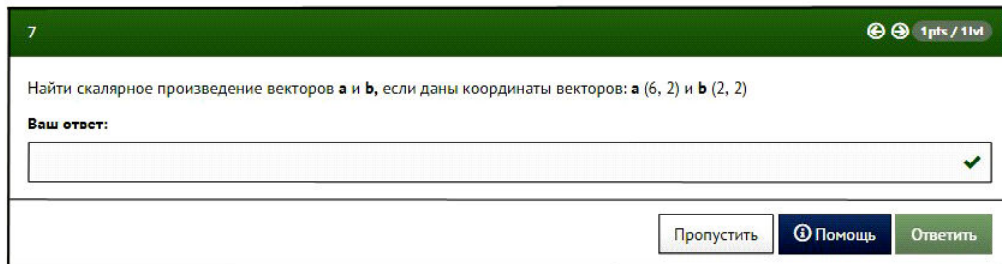


Рис. 4

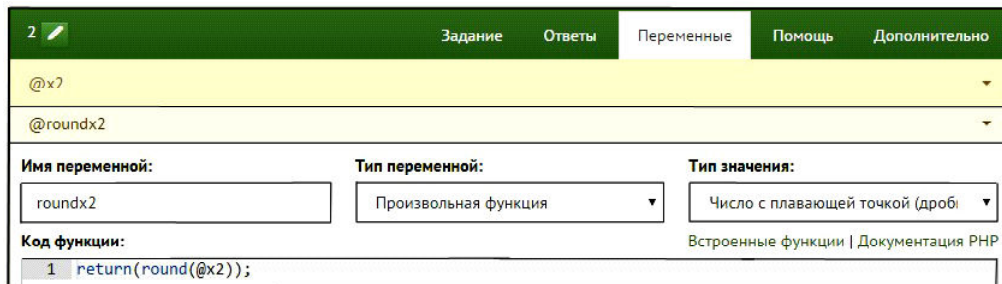


Рис. 5

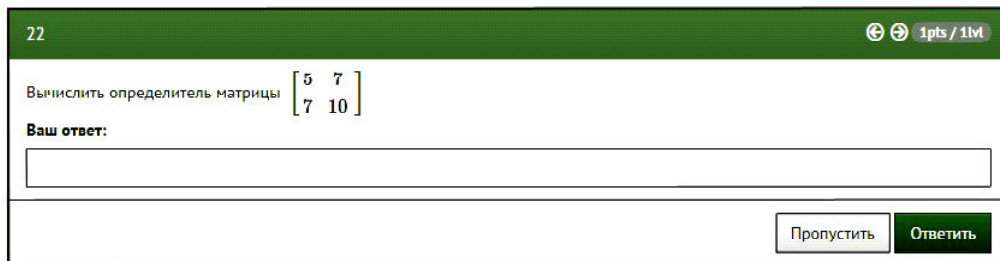


Рис. 6

менная  $r1$  необходима для составления задания, принимает случайное значение из списка  $\{«+», «*»\}$ . Переменная  $ans1$  — произвольная функция, код функции:  $if (@r1 == «+») return (@y + @x); else return (@y * @x)$ . То есть в зависимости от значения  $r1$   $ans1$  будет вычислять сумму или произведение чисел. Тогда задание имеет вид «Вычислить:  $@y @r1 @x$ », ответ —  $@ans1$ . Обучаемым будут предъявлены разные задания на нахождение суммы или произведения чисел, например «Вычислить:  $5+7$ », «Вычислить:  $3*8$ ».

Для каждого пользователя и для каждого задания генерируется один набор переменных, который используется при повторном запросе. Это предоставляет возможность организовать помощь обучаемому, например указать шаги решения задачи.

В системе предусмотрены разные варианты ответов: текст, один ответ из множества, несколько ответов из множества, матрица / вектор. Корректная проверка ответов может существенно упростить задачу для проверяющего знания. Ответственность за выполнение этой задачи возложена на модель ответов. Среди свойств модели ответов можно указать такие как список вариантов ответов, флаг правильности ответа пользователя, который может использоваться публично, идентификатор задания, для которого генерируются или проверяются ответы, список правильных ответов, флаг публичности правильности ответа пользователя.

В системе контроля знаний преподаватель имеет возможность объединять задания в блоки. Пользователи распределяются по группам. В блоках указывается доступность задания для конкретных групп пользователей. Для каждой задачи можно установить родительскую задачу, которая позволяет подразделять основную задачу на подзадачи. Каждая подзадача в системе является заданием, модель подзадачи необходима, для того чтобы создать для каждой подзадачи свой объект задания. Модель подзадачи использует модель заданий.

Большое значение в системе проверки знаний имеет контроль за степенью самостоятельности выполнения обучаемым задания, так как проверка знаний может проводиться удаленно. ExerciseBook для реализации такого контроля использует историю взаимодействия пользователя с системой, записывая основные действия с учетом времени, начиная со времени работы над каждым заданием и заканчивая копированием текста на странице. Эти данные доступны преподавателю, который после проведения небольшого анализа может предварительно оценить степень самостоятельности работы обучаемого. Преподаватель может посмотреть время работы пользователя в общем и над каждым заданием, введенные ответы, количество верных ответов, количество и время переходов на другие страницы, отвлечения. Все эти данные предоставляются в количественном виде и в виде графиков, удобных для анализа.

Система предоставляет преподавателю возможность просмотреть общую статистику групп обучаемых (рис. 7). В этом случае отображаются общие данные по группе по тем же позициям, которые были доступны по отдельному пользователю. Статистика по заданиям позволяет оценить время работы по каждому заданию и успешность пользователей при работе с данным заданием. И такая же статистика предоставляется по каждому блоку заданий. Таким образом, оценивается не только результат, но и деятельность обучаемого.

Как показало пробное тестирование, система ExerciseBook (<http://exercisebook.ru/>) позволяет решить задачу проверки знаний, обеспечивая интуитивно понятную работу, достаточные возможности для настройки, обработку и визуализацию нетривиальных данных, одинаковую или разную по плану преподавателя сложность заданий, контроль за «списываниями». При этом она предоставляет возможность для перепроверки ответов преподавателем или проверки ответов вручную, а при автоматической проверке делает это с установленной погрешностью или из списка возможных ответов. Типы переменных, описанные в статье, используются при составлении моделей задач, к ним могут быть добавлены новые. В настоящее время работа продолжается. Каждый зарегистрированный в системе преподаватель создает свою базу заданий, на данный момент пробная база заполняется.

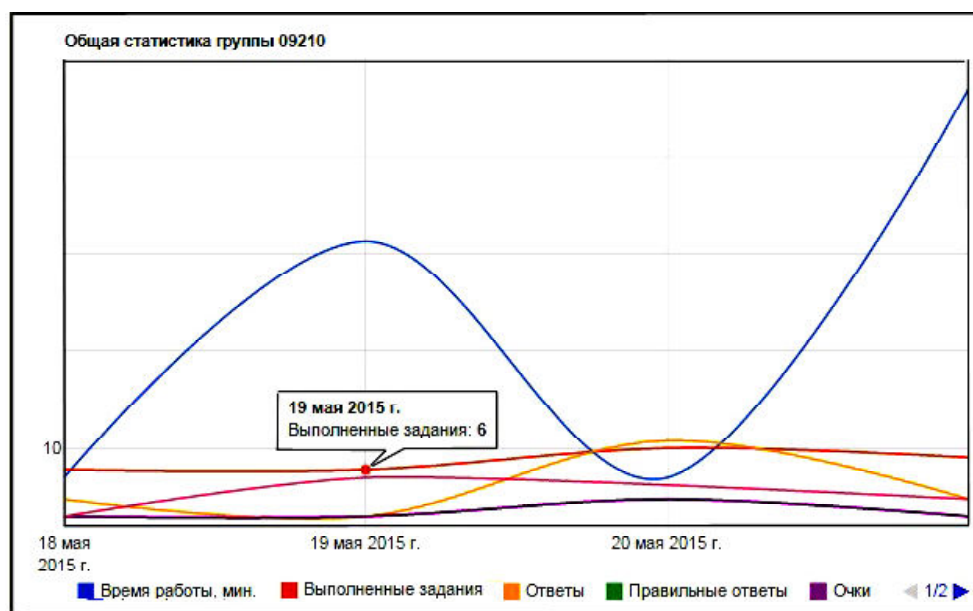


Рис. 7

Безопасность системы имеет несколько уровней (БД, архитектура, проверка и фильтрация входных и выходных данных), тем самым исключая возможность несанкционированного доступа к системе. Разработанная система может применяться при обучении и контроле. Стоит также отметить, что система ориентирована на проверку знаний по математическим дисциплинам, но может быть применима и в других областях.

## Литература

1. Башмаков А.И., Башмаков И.А. Разработка компьютерных учебников и обучающих систем. М.: «Филин», 2003.
2. Зорин Ю.А. Автоматизация построения многовариантных тестовых заданий на основе деревьев И/ИЛИ / Дисс. на соискание ученой степени канд. техн. наук. Томск, 2015.
3. Зорин Ю.А., Посов И.А. Инструментальные системы построения и получения многовариантных тестовых заданий // Компьютерные инструменты в образовании, 2014. № 1. С. 14–25.
4. Кручинин В.В., Магазинников Л.И., Морозова Ю.В. Модели и алгоритмы компьютерных самостоятельных работ на основе генерации тестовых заданий // Известия Томского политехнического университета, 2006. Т. 309. № 8. С. 258–263.
5. Миннегалиева Ч.Б. Приемы повышения эффективности использования электронных образовательных ресурсов // Образование и саморазвитие. Казань, 2014. № 2 (40). С. 105–108
6. Осокин А. Е. Использование вычисляемого типа тестовых вопросов в системе Moodle // Информатика и образование: границы коммуникаций. 2013. № 5 (13). С. 385–386.
7. Поздняков С.Н., Рыбин С.В., Чухнов А.С. Информационно-коммуникационные технологии в курсе дискретной математики СПбГЭТУ «ЛЭТИ» // Образовательные технологии и общество, 2014. Т. 17. № 4. С. 576–592.
8. Попова Г.И. Системы генерации индивидуальных заданий в среде MathCAD // Образовательные технологии, 2013. № 4. С. 96–104.
9. Посов И.А. Автоматизация процесса разработки и использования многовариантных учебных заданий / Дисс. на соискание ученой степени канд. техн. наук. СПб., 2012.
10. Посов И.А. Программирование генераторов задач // Компьютерные инструменты в образовании, 2010. № 3. С. 19–31

11. Хабибуллина Г.З., Хайруллина Л.Э. Обучение будущих учителей естественнонаучного цикла составлению компьютерных тестов на занятиях по современным средствам оценивания результатов обучения // Казанский педагогический журнал, 2014. № 3 (104). С. 81–85.
12. Генератор задач [Электронный ресурс]. Режим доступа: <http://www.wolframalpha.com/pro/problem-generator/> (дата обращения 22.12.2014).

## SOME ASPECTS OF KNOWLEDGE CONTROL AUTOMATION

Minnegalieva Ch. B., Mukhamedshin D.R., Rusetski K.V., Parkalov A.V.

### Abstract

This article deals with problems of automatic control of knowledge. The method for generation of learning tasks using variables of different types with a help of ExerciseBook system is proposed in the article. Examples of tasks formation are shown. A method for pre-evaluation of the user through the history of interaction with the system is described.

**Keywords:** *eLearning, learning tasks generation, automatic control of knowledge, electronic educational resource.*

Миннегалиева Чулпан Бакиевна,  
кандидат педагогических наук, доцент  
кафедры информационных систем КФУ,  
[mchulpan@gmail.com](mailto:mchulpan@gmail.com)

Мухамедшин Дамир Рафкатович,  
младший научный сотрудник НИИ  
«Прикладная семиотика» АН РТ,  
[i@damirmuh.ru](mailto:i@damirmuh.ru)

Русецкий Кирилл Валерьевич,  
аспирант кафедры интеллектуальных  
информационных технологий БГУИР,  
[rusetski.k@gmail.com](mailto:rusetki.k@gmail.com)

Паркалов Алексей Викторович,  
аспирант кафедры интеллектуальных  
информационных технологий БГУИР,  
[a.parkalov@gmail.com](mailto:a.parkalov@gmail.com)



Наши авторы, 2014.

Our authors, 2014.



