

УДК: 681.3.06

Сологуб Г.Б.

## РАЗРАБОТКА СИСТЕМЫ ИМИТАЦИОННОГО ТЕСТИРОВАНИЯ

Глеб Борисович Сологуб родился в 1986 г. в Москве. Студент МАИ. Основные научные интересы – в области теории принятия решений, компьютерной технологии обучения. Автор восьми научных работ.

E-mail: [glebsologub@ya.ru](mailto:glebsologub@ya.ru)

Ключевые слова: тест, экзамен, контроль знаний, модель знаний, обучаемый, преподаватель, студент, вопрос, задание, оценка, шкала.

## A DEVELOPMENT OF SIMULATION TEST SYSTEM

Gleb B. Sologub was born in 1986, in Moscow. He is a Student at the MAI. His research interests are in decision making theory, computer-aided education technology. He has published 8 technical papers.

Key words: test, examination, educational assessment, knowledge model, trainee, student, teacher, question, task, grade, scale.

## **Аннотация**

*В статье описана структура, функциональность и особенности разрабатываемого автором комплекса программных средств для создания тестов. Рассмотрен механизм последовательной генерации вопросов и методика оценивания результатов тестирования.*

## **Abstract**

*A structure, functionality and features are described for a software tool set meant for generation of tests. A scheme of sequential question generation is presented as well as a technique to evaluate testing results.*

## **Список сокращений**

*СИТ – система имитационного тестирования*

*СА – средство автора*

*СП – средство преподавателя*

*СТ – средство тестируемого*

*БД – база данных*

*ПК – персональный компьютер*

*МТЗ – модель требуемых знаний*

*МТекЗ – модель текущих знаний*

*ВЗ – вопрос-задание*

*МКТ – модель компьютерного тестирования*

*ВСОО – вектор семантики ответа обучаемого*

*ВСПО – вектор семантики правильного ответа*

*ССО – суммарная средняя оценка*

*РП – решающее правило*

*ССВ – список следующих вопросов*

*AJAX – Asynchronous JavaScript and XML*

*DHTML – Dynamic HyperText Markup Language*

*JSON – JavaScript Object Notation*

## **1. Принципы создания системы имитационного тестирования**

Разработка системы имитационного тестирования (СИТ) базируется на основных положениях кибернетической компьютерной технологии обучения, разработанной на кафедре «Математическая кибернетика» Московского авиационного института [1,2]:

Процесс обучения – информационный процесс, представляющий собой последовательность операций, каждая из которых определяет соответствующий этап изучения дисциплины: овладение информацией, выработка понимания, выработка умения, выработка навыков. Результатом выполнения этих операций является последовательное формирование в сознании обучаемого смысловых моделей изучаемой дисциплины.

Процесс обучения на каждом этапе является управляемым по принципу обратной связи процессом, что позволяет адаптировать его к текущему уровню знаний обучаемого.

Процесс обучения моделируется как интеллектуальный процесс на основе построения моделей требуемых и текущих знаний (МТЗ и МтекЗ), в

соответствии с которыми изменяются алгоритм выработки заданий обучаемому и форма контроля знаний.

## 2. Система имитационного тестирования

### 2.1. Структура системы

Структура СИТ представлена на рис. 1:

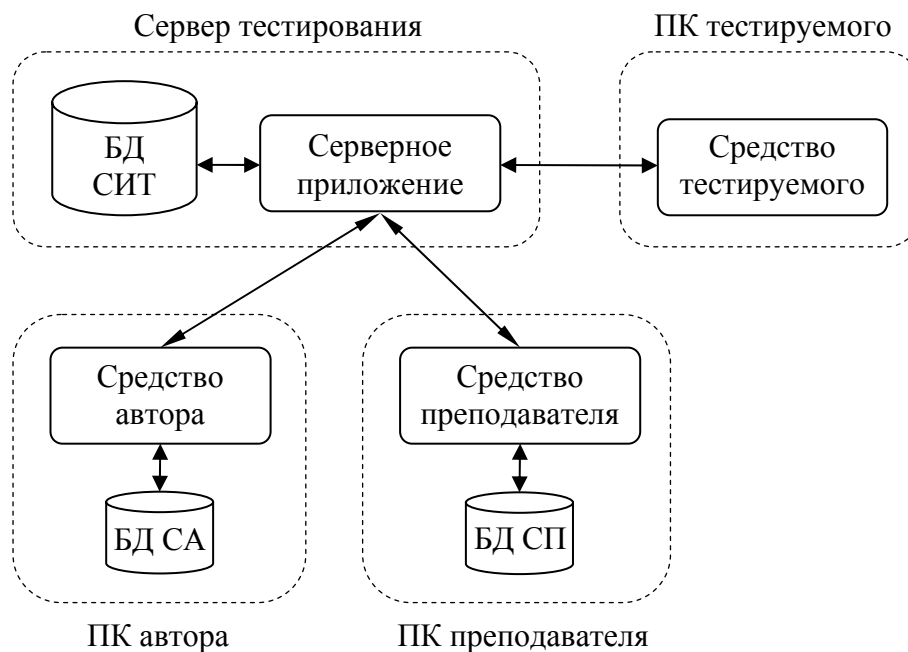


Рис. 1. Структура СИТ.

Она включает:

- базу данных (БД СИТ) и серверное приложение;
- средство автора (СА) с его базой данных;
- средство преподавателя (СП) с его базой данных;

- средство тестируемого (СТ).

Средство автора содержит набор инструментов, предназначенных для облегчения или частичной автоматизации следующих операций, осуществляемых автором учебного курса при построении модели требуемых знаний (МТЗ):

- создание вопросов-заданий (ВЗ) по шаблону на основе встроенных стандартных и подключаемых пользовательских типов ВЗ;
- указание для каждого ВЗ вектора семантики правильного ответа (ВСПО), принадлежности к определенной категории и сложности ВЗ;
- построение дерева учебного курса и привязка к его узлам соответствующих ВЗ;
- описание логики адаптивного задания вопросов путем составления для каждого ВЗ (или группы ВЗ) набора решающих правил (РП), по которым в зависимости от количественных и качественных характеристик ответа тестируемого формируется список следующих вопросов (ССВ).

Средство автора позволяет визуально представить вопросы-задания именно так, как они будут выглядеть на экране тестируемого (т.е. СА представляет собой визуальный редактор вопросов-заданий).

Средство преподавателя предназначено для создания, настройки и администрирования компьютерного экзамена (тестирования) на основе

построенных МТЗ. Оно позволяет построить модель компьютерного тестирования (МКТ), с помощью:

- выбора вопросов или тем, которые войдут в конкретное тестирование, фильтрации их по категории и сложности;
- выбора решающих правил, которые будут применяться при генерации вопросов;
- задания параметров процесса тестирования таких, как общее время выполнения и время ответа, количественная или качественная шкала оценивания, набор стартовых вопросов.

Кроме того, СП позволяет:

- управлять доступом обучаемых к созданным тестам;
- контролировать ход выполнения заданий тестируемыми;
- получать протоколы и сводные отчеты по проведенным тестированиям;
- формировать модели текущих знаний (МТекЗ) обучаемых по итогам прохождения контроля знаний.

Средства автора и преподавателя имеют свои собственные базы данных, в которых будут храниться построенные и редактируемые МТЗ и МКТ соответственно. Кроме того, они позволяют осуществлять экспорт построенных моделей в базу данных системы имитационного тестирования (БД СИТ), расположенную на сервере тестирования, а также импортировать имеющиеся в

БД СИТ модели с помощью серверного приложения, которое осуществляет следующие функции:

- работа с БД СИТ (получение/отправление ВЗ, МТЗ, МКТ, информации о тестируемом и тестированиях и т.п.);
- генерация вопросов-заданий и оценивание ответов тестируемого, реализация процесса тестирования;
- выдача информации о тестируемых и тестированиях преподавателю.

Средство тестируемого осуществляет следующие функции:

- взаимодействие с серверным приложением: аутентификация и регистрация пользователя, выбор теста, получение вопросов-заданий и отправление векторов семантики ответа обучаемого (ВСОО);
- отображение ВЗ на экране, предоставление удобного интерфейса для выполнения заданий непосредственно на экране компьютера;
- отображение личной статистики, а также рекомендаций по итогам прохождения тестов для тестируемого.

## 2.2 Механизм последовательной генерации вопросов-заданий и методика оценивания результатов тестирования

В процессе тестирования обучаемый отвечает на последовательно задаваемые вопросы, включенные преподавателем в состав теста.

Первый вопрос выдается случайным образом из списка стартовых вопросов, заданного преподавателем. Последующие вопросы генерируются на основании количественных и качественных результатов выполнения заданий.

За ответ на каждый вопрос выставляется оценка по шкале от 0 до 100. Для получения такой оценки используются следующие формальные построения.

Каждый вопрос-задание можно представить в виде следующего фрейма:

(Вопрос:

Семантический элемент 1 (значение слота 1)

Семантический элемент 2 (значение слота 2)

.....

Семантический элемент k (значение слота k)).

При ответе на вопрос происходит заполнение семантических элементов вопроса конкретными значениями, таким образом получается вектор семантики ответа обучаемого (ВСОО). При этом для каждого вопроса известен правильный ответ на него (ВСПО – вектор семантики правильного ответа).

Множество всех возможных ответов на вопрос имеет вид  $V=S_1 \times S_2 \times \dots \times S_k$ , где  $S_i$  – множество значений  $i$ -го семантического элемента вопроса,  $i=1 \dots k$ .

На элементах  $V$  можно задать метрику. В простейшем случае она выглядит так:

$$d(v_1, v_2) = (eq(s_1^1, s_2^1) + eq(s_1^2, s_2^2) + \dots + eq(s_1^k, s_2^k)) / k,$$

где  $v_1 = \langle s_1^1, s_1^2, \dots, s_1^k \rangle$ ,  $v_2 = \langle s_2^1, s_2^2, \dots, s_2^k \rangle$ ;  $v_1, v_2 \in V$ , а функция  $eq$  имеет вид

$eq(x,y)=\{0, \text{ если } x=y; 1 \text{ в противном случае}\}.$

Содержательный смысл множеств  $S_i$  в зависимости от типа вопроса может быть различным.

Например, один из стандартных типов вопроса-задания – выбор единственного значения из набора альтернатив. В этом случае вектор ответа имеет единичную размерность, а  $S$  является множеством заданных альтернатив.

Другой стандартный тип ВЗ предусматривает ввод текстового значения (термина, формулировки теоремы и т.п.), т.е.  $S$  является множеством текстов.

Еще один вариант задания – конструирование формулы из заданных элементов. Здесь  $V=S \times \dots \times S$ ,  $S$  – множество этих элементов.

Возможны другие формулировки ВЗ, которые тем не менее в большинстве случаев могут быть заданы указанным способом. В частности, задание может являться комбинацией стандартных ВЗ (например, в каждом задании контрольной работы может быть необходимо найти значение нескольких величин и/или указать ответы на качественные вопросы).

С помощью введенной метрики можно получить оценку ответа обучаемого:

$$g(\text{BCOO})=(1-d(\text{BCOO},\text{BCПО})) \cdot 100.$$

Задаваемые вопросы могут иметь различную сложность. Учет сложности вопроса осуществляется в соответствии с рядом естественных требований:

- неполный или частично верный ответ на сложный вопрос должен оцениваться выше, чем аналогичный ответ на легкий вопрос;

- полностью правильный ответ на легкий вопрос не должен оцениваться ниже, чем аналогичный ответ на сложный вопрос (чтобы не ухудшать общую оценку);
- абсолютно неверный ответ на сложный вопрос не должен оцениваться выше, чем аналогичный ответ на легкий вопрос (чтобы не завышать общую оценку).

Перечисленным требованиям удовлетворяет метод учета сложности вопроса, в соответствии с которым оценка за ответ на вопрос вычисляется по формуле

$$g(\text{BCOO}) = C(1 - d(\text{BCOO}, \text{ВСПО})) \cdot 100,$$

где  $C(x) = x^{\exp(-c)}$ ;  $c$  – сложность вопроса, причем  $c = 0$  соответствует вопрос нормальной,  $c < 0$  – пониженной, а  $c > 0$  – повышенной (рис. 2).

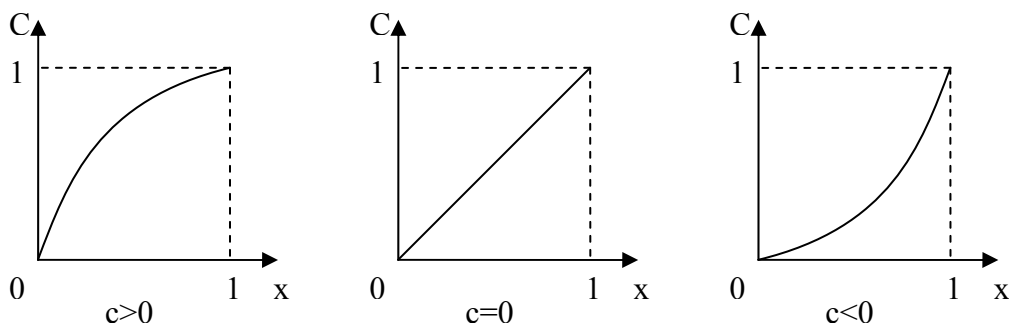


Рис. 2. Вид функции  $C(x)$  в зависимости от сложности вопроса.

При формировании каждого следующего вопроса последовательно применяется набор решающих правил (РП), которые в зависимости от качественных или количественных характеристик ответа обучаемого могут

выдавать тот или иной список следующих вопросов (ССВ). Решающие правила составляются автором или преподавателем.

Можно выделить 3 типа решающих правил:

- 1) семантическое правило: «Если  $eq(BCOO^i, BCPO^i) = 1$  (т.е. ответ на  $i$ -й семантический элемент вопроса неверен), выдать список  $C^c_i$ »;
- 2) оценочное правило: «Если  $I_{jmin} \leq g(BCOO) \leq I_{jmax}$  (т.е. оценка за ответ принадлежит отрезку  $[I_{jmin}; I_{jmax}]$ ), выдать список  $C^o_j$ »;
- 3) тривиальное правило: «Выдать список  $C$ ».

Из выданного правилом списка исключаются уже заданные вопросы. Если полученный список следующих вопросов непуст, перебор правил заканчивается. После чего случайным образом из ССВ выбирается один вопрос, который и будет предложен тестируемому.

Если ни одно из решающих правил не срабатывает или они попросту не заданы, то выдается случайный вопрос из списка незаданных стартовых вопросов, если таковой непуст, в противном случае выдается случайный незаданный вопрос, если такие имеются.

Если все возможные вопросы уже были заданы, то тест завершается.

В процессе тестирования, помимо оценки за ответ на каждый вопрос, вычисляется также суммарная средняя оценка (ССО):  $\hat{g}_n = (g_1 + \dots + g_n)/n$ , где  $n$  – число заданных вопросов,  $g_i$  – оценка за  $i$ -й вопрос,  $i=1, \dots, n$ .

По умолчанию итоговая отметка выдается приближением средней суммарной оценки (по результатам оценивания ответов обучаемого на все

заданные в процессе тестирования вопросы) до отметки по равномерной целочисленной 100-балльной шкале (характеризует процент правильности выполнения).

Преподаватель может задать разбиение этой шкалы на интервалы и соответствующие им качественные/количественные отметки, например:

[0,50] – "неудовлетворительно" или "2",

[51,66] – "удовлетворительно" или "3",

[67,83] – "хорошо" или "4",

[84,100] – "отлично" или "5".

Преподаватель может задать ограничения: на время выполнения каждого задания, на общее время тестирования, на количество задаваемых вопросов.

Возможна следующая имитация традиционного экзамена.

Преподаватель указывает минимальное количество  $n$  вопросов, которые необходимо задать тестируемому для выставления отметки.

В процессе тестирования при ответе на  $n$  вопросов тестируемому выставляется предварительная отметка  $M_j$  и предлагается выбор: получить возможность повысить её до отметки  $M_{j+1}$ , ответив на  $m$  дополнительных вопросов и рискуя понижением отметки до  $M_i$ , либо завершить тестирование с полученной отметкой.

Минимальное число дополнительных вопросов для повышения отметки можно вычислить по формуле, которая выводится из условия, что

$I_j \min \leq \hat{g}_n \leq I_j \max$  и на все дополнительные вопросы тестируемый ответит правильно ( $g_{n+1} = \dots = g_{n+m} = 100$ ):

$$m = [n \cdot (I_{j+1} \min - \hat{g}_n) / (100 - I_{j+1} \min)] + 1,$$

где  $[\cdot]$  – целая часть числа, а  $I_{j+1} \min$  – нижняя граница интервала, которому соответствует более высокая отметка  $M_{j+1}$  по сравнению с  $M_j$  – текущей оценкой.

Наихудшую ССО при продолжении тестирования можно вычислить в предположении, что все ответы на дополнительные вопросы оказались неверными ( $g_{n+1} = \dots = g_{n+m} = 0$ ), по формуле  $\hat{g}_{n+m} = \hat{g}_n \cdot n / (n+m)$ .

### 2.3. Программная реализация системы имитационного тестирования

Система имитационного тестирования реализуется в виде комплекса взаимосвязанных между собой программных средств:

- средств автора, преподавателя и тестируемого – клиентских интернет-приложений, разработанных по технологии DHTML+JavaScript, работающих на любом компьютере при наличии Adobe Air – бесплатной, существующей в нескольких версиях для разных операционных систем платформы исполнения (полный актуальный список требований и ссылка для загрузки бесплатного дистрибутива Adobe Air размещены на сайте компании Adobe [3]);

- БД СИТ, реализованной с помощью аппаратно-независимой СУБД MySQL и серверного приложения, написанного на языке PHP, которые могут быть расположены как на локальном компьютере пользователя, так и на удаленном сервере в компьютерной сети, в т.ч. в сети интернет, при условии наличия установленной СУБД MySQL и интерпретатора PHP версии 4.1 и выше.

Связь между серверным и клиентскими приложениями осуществляется посредством AJAX-запросов по протоколу XMLHttpRequest [4]. Текстовые данные используются и передаются в формате JSON.

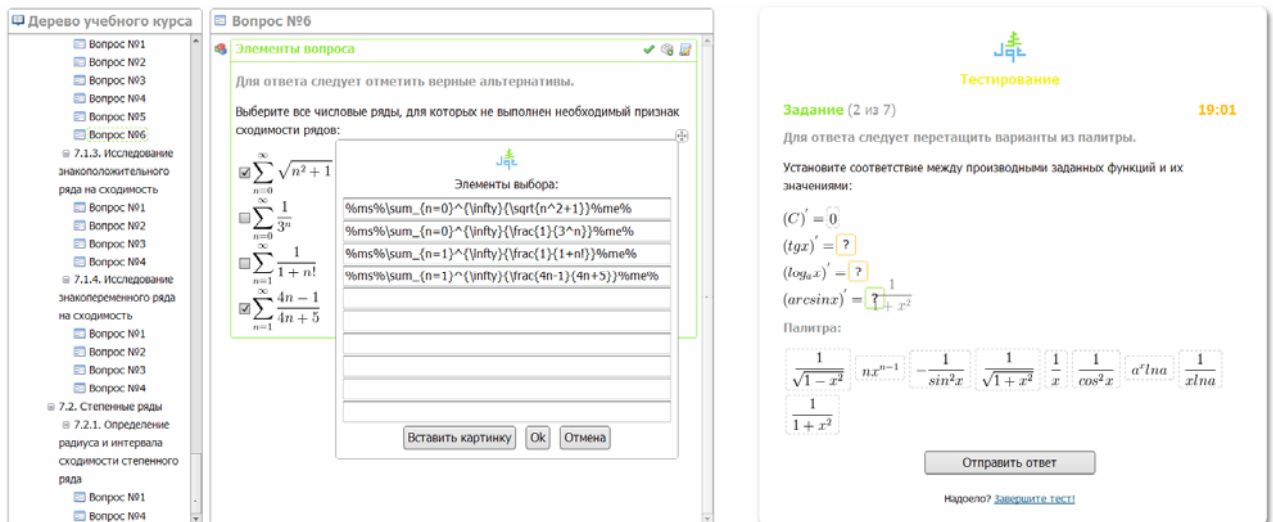


Рис. 3. Пользовательский интерфейс средства автора и средства тестируемого.

Простота и удобство передачи, хранения и обработки данных в клиентских приложениях, а также визуальная привлекательность их интерфейса (см. рис. 3) были достигнуты за счет использования JavaScript-библиотеки jQuery.

Разработка СИТ ведется в среде программирования Aptana Studio.

Все используемое при разработке программное обеспечение является бесплатным и свободно распространяемым.

### **3. Выводы**

В статье описана структура системы имитационного тестирования, состав и функциональное назначение входящих в неё программных средств, предложена общая схема организации взаимодействия между ними.

Рассмотрены особенности программной реализации системы.

Предложен механизм последовательной генерации вопросов в процессе тестирования, описана методика, применяемая для оценивания результатов тестирования.

Рассмотренные методы могут применяться при создании любой системы компьютерного тестирования.

Разрабатываемая автором система имитационного тестирования содержит комплекс программных средств, предназначенных для создания, организации и проведения компьютерного контроля знаний обучаемых в форме тестирования, контрольной работы или экзамена.

Заложенная в основу системы общность применяемых методов вкупе с возможностью их настройки под выбранные учебные курсы, реализация формального метода оценивания результатов контроля знаний и всесторонний подход к решению задачи контроля знаний приближают нас к созданию умной

многофункциональной системы тестирования, требования к которой сформулированы в [5].

#### Список литературы

1. Семенов, В.В. Компьютерные технологии в дистанционном обучении. / В.В. Семенов и др. // Новые информационные технологии в образовании: Обзор. информ. / НИИВО; Вып.12. – М., 1997. –64с.
2. Семенов, В.В. Развитие компьютерных технологий в дистанционном обучении. / В.В. Семенов и др. // Новые информационные технологии в образовании: Обзор. информ. / НИИВО; Вып.3. – М., 1999. –68с.
3. <http://www.adobe.com/ru/products/air/systemreqs/>
4. Н. Закас, Дж. Мак-Пик, Дж. Фосетт. Ајах для профессионалов. М.: Символ-Плюс, 2008.
5. Сологуб Г.Б. Умная многофункциональная система тестирования. 7-я международная конференция «Авиация и космонавтика – 2008»: Тезисы докладов. М.: Изд-во МАИ-ПРИНТ, 2008. – с. 210-211.