УДК 004.31

Параметризация заданий контроля как средство повышения надежности тестов в системах компьютерного обучения

Пантелеев Е.Р., д-р техн. наук, Архипов А.Л., асп.

Предложен способ повышения надежности тестов в системах компьютерного обучения, основанный на представлении контрольного задания в виде шаблона с переменными параметрами. Определены основные проблемы разработки и использования параметризованных заданий. Предложена распределенная компонентная архитектура программных средств контроля знаний с помощью параметризованных заданий. Рассмотрен пример ее реализации.

Ключевые слова: компьютерное обучение, контроль знаний, надежность тестов.

Parameterization of Control Tasks as Tool for Increasing Tests Reliability in Systems of Computer Learning

E.R. Panteleev, Doctor of Engineering, A.L. Arkhipov, Post Graduate Student

The article contains the method of increasing the tests reliability for computer training, based on the representation of the control questions as a template with variable parameters is pro-posed. The basic problems of development and using of parameterized questions are considered. A distributed architecture of knowledge control components for supporting parameterized questions is described. An example of its implementation is included.

Keywords: computer learning, computer-assisted testing, reliability of computer-assisted testing.

Согласно ГОСТ27.002-89, под надежностью понимается свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения. Применительно к системам компьютерного обучения можно говорить, что объектом является тест, а его функцией – объективный контроль знаний. Под объективностью контроля понимается независимость его результатов от каких-либо иных факторов, кроме уровня знаний студента. Мерой надежности является коэффициент корреляции r_{xy} [1] результатов двух последовательных применений компьютерного теста для контроля знаний одной и той же группы студентов:

$$r_{xy} = \frac{M\left[\left(X - M(X)\right)\left(Y - M(Y)\right)\right]}{S_x S_y},$$

Здесь X и Y – выборки результатов первого и повторного тестирования; $M(*)$ – математические ожидания соответствующих величин; S_x, S_y – средние квадратические отклонения по выборкам. Если r_{xy} близок к 1, это может свидетельствовать о том, что, сколько бы раз тест ни был применен для оценки знаний контрольной группы, он покажет примерно одинаковые результаты, то есть даст объективную оценку знаний. Эта гипотеза справедлива, если исключено или ослаблено влияние результатов первого теста на второй, что может иметь место в результате механического запоминания вариантов ответа, данных студентом при прохождении первого теста. На практике это обычно достигается за счет использования параллельных вариантов тестов с эквивалентными по сложности заданиями. Например, в системе интернет-обучения ГИПЕРТЕСТ [2] параллельные варианты контрольных заданий объединяются в группу со случайной выборкой. Резервирование уменьшает вероятность запоминания, но одновременно увеличивает трудозатраты на разработку теста и порождает проблему доказательства эквивалентности заданий.

Предлагается повысить надежность компьютерных тестов за счет параметризации заданий контроля. В рамках этого подхода контрольное задание представляет собой шаблон с переменными параметрами, уникальное сочетание значений которых генерируется в момент обращения студента к заданию. Например, шаблон задания «Вычислите значение выражения X+Y» включает в себя переменные X и Y и в конкретном варианте заполнения примет вид задания открытой формы с числовым ответом: «Вычислите значение выражения 3+7». Данный подход лишает смысла запоминание ранее данного варианта ответа и при должной процедуре подбора параметров обеспечивает эквивалентность возможных вариантов заданий. Ниже рассматривается один из возможных подходов к разработке параметризованных контрольных заданий, реализованный в системе интернет-обучения ГИПЕРТЕСТ [2].

Хотя параметризованное задание контроля, предъявленное студенту, может внешне не отличаться от стандартных заданий (например, в ранее приведенном примере оно имеет вид задания открытой формы; возможно также его представление в виде задания закрытой формы: меню с одиночным и множест-

венным выбором, упорядочение предопределенных элементов ответа и т.д.), оно принципиально отличается от них алгоритмом формирования задания и оценки правильности ответа студента. Эти алгоритмы специфичны для каждого конкретного задания и поэтому не могут быть предусмотрены заранее ни одной системой компьютерного обучения. Проблема расширения типов контрольных заданий в системе интернет-обучения ГИПЕРТЕСТ решена путем реализации унифицированного программного интерфейса на базе стандарта SOAPweb-сервисов [6]. Интерфейс предусматривает две операции: «Генерировать задание» и «Оценить ответ». Это решение неоднократно апробировано (контрольные задания, предусматривающие ввод ответа в виде формулы [3] или алгоритма, записанного на одном из языков программирования [3]) и доказало свою практическую состоятельность. По сравнению с использованием для той же цели стандарта SCORM [5] принятое решение имеет существенное для компьютерного контроля знаний преимущество – и генерация задания, и контроль ответа выполняются на стороне сервера, а не в клиентской странице браузера. Это, во-первых, снимает ограничения на сложность алгоритма проверки ответа, а во-вторых, сводит к нулю возможность «взлома» задания. Таким образом, практическая реализация параметризованных заданий контроля связана с необходимостью разработки шаблона контрольного задания, а также программирования процедуры-генератора вариантов и процедуры оценки правильности ответа в контексте заданной параметризации.

Наиболее сложной из перечисленных задач является подключение разработанных программных компонентов к ГИПЕРТЕСТ. Для решения этой задачи предложена распределенная компонентная архитектура, объединяющая ключевые компоненты в формате SOAP web-сервисов (система контроля знаний ГИПЕРТЕСТ, модуль генерации вариантов заданий и проверки правильности ответа). Такая архитектура обеспечивает открытость, возможность подключения неограниченного количества генераторов заданий и модулей проверки (далее – модуль ГиП), но требует решения ряда технических проблем. Дело в том, что проверка правильности ответа в многопользовательской системе должна быть выполнена именно для того варианта задания, который был предъявлен конкретному пользователю. Возникает необходимость сохранения контекста задания на интервале времени между генерацией и проверкой. Эта операция не несет предметной специфики задания, поэтому для ее выполнения в архитектуру системы добавлен компонент-диспетчер (далее – Диспетчер) (рис. 1). В его функции входит переадресация запроса на генерацию контрольного задания указанному при вызове диспетчера компоненту и сохранение полученного варианта параметров для данной сессии (функция getTAsk), а также переадресация вызова компонента проверки с передачей ему сохраненных параметров (функция sendAnswer). Таким образом, в компетенции разработчика параметризованного задания остаются только процедуры генерации заданий и проверки ответа.

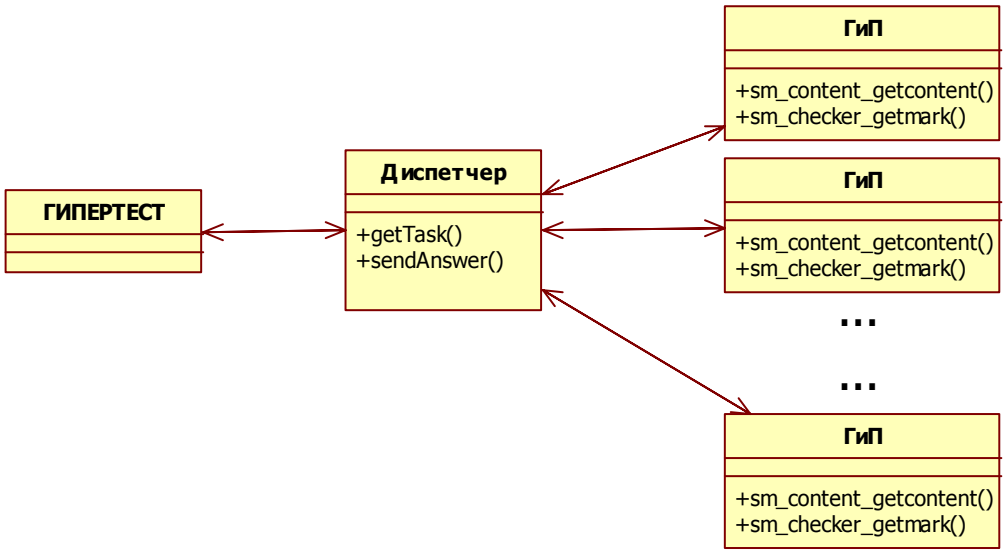

Рис. 1. Пример параметризованного задания в тесте контроля знаний

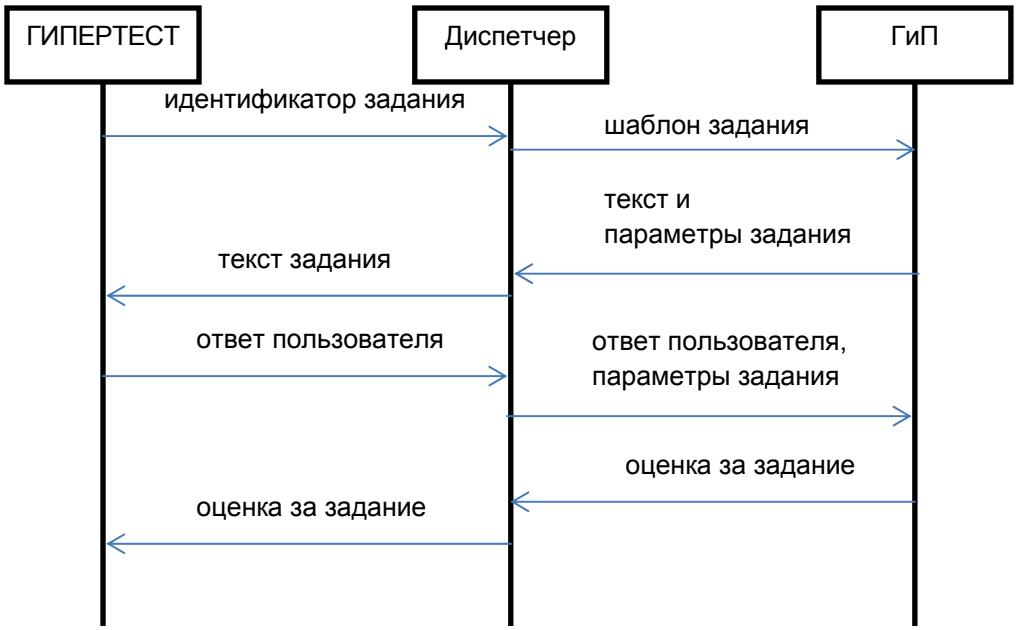

Рис. 2. Сценарии запроса контрольного задания и обработки ответа

Запрос параметризованного задания выполняется по следующему сценарию (рис. 2):

1. ГИПЕРТЕСТ запрашивает у Диспетчера текст задания. Параметром запроса служит идентификатор задания, определенный в текущем тесте.
2. Диспетчер вызывает метод модуля ГиП генерации задания, передает ему шаблон и получает от него текст задания и параметры.
3. Диспетчер сохраняет в базе данных значения параметров задания для текущего сеанса.
4. Текст задания передается подсистеме контроля знаний в ГИПЕРТЕСТ.

Сценарий проверки ответа пользователя следующий (рис. 2):

1. ГИПЕРТЕСТ отправляет Диспетчеру ответ пользователя.
2. Диспетчер извлекает параметры задания из базы данных, вызывает метод модуля ГиП проверки ответа, передает ему параметры задания и ответ пользователя, получает от него оценку в виде нормированного вещественного числа в диапазоне 0–1, соответствующего мере правильности ответа.
3. Диспетчер пересылает оценку подсистеме контроля знаний в ГИПЕРТЕСТ.

Модуль ГиП может быть реализован как на PHP, так и на ASP.NET, или на любой другой платформе, поддерживающей SOAPweb-сервисы. Единственное требование к web-сервису – соответствие спецификации интерфейса модулей ГиП (набор методов и их параметры).

Шаблон задания контроля с параметрами может быть формально представлен следующей пятеркой:

T = <B,P,V,F,A>.

Здесь B – тело (неизменяемая часть) шаблона в формате html-кода; P – множество типизированных параметров в теле шаблона, выделенных синтаксистом [[имя параметра]] (с учетом специфики контроля инженерных знаний поддерживаются следующие типы: численный скаляр, вектор, матрица); V – множество допустимых значений параметров: одноэлементное (определяет константное значение параметра); аксиоматическое (определяет правило порождения значения, например, как процедуру выбора случайного значения из заданного интервала или элемента из списка); F: P → V – функция отображения множества параметров на множество допустимых значений; A – множество интерактивных элементов ввода ответа, например, кнопок или полей ввода.

Процедура генерации варианта выполняет лексический разбор html-кода шаблона, интерпретирует функцию F и заменяет параметры шаблона из P константами из V. Текст задания может содержать более одного интерактивного элемента ввода ответа, если это необходимо.

Синтаксис определения параметров может быть представлен следующими выражениями в форме Бэкуса:

```
F = имя "=" значение {";" имя "=" значение}
значение = ОДЗ|матрица
матрица = "(" столбцы ";" строки ";" ОДЗ{";"
ОДЗ} ")"
столбцы = число
строки = число
ОДЗ = число {"," число}|минимум ".." максимум
[шаг]
минимум = число
максимум = число
шаг = число
```

Здесь имя – имя параметра (последовательность латинских символов); число – целое или действительное число (корректная форма записи числового значения); ОДЗ – область допустимых значений скаляра, заданная перечислением или арифметической прогрессией в диапазоне минимум–максимум; столбцы и строки – константы, определяющие размер матрицы (вектора).

Например, шаблон задания «сложить два числа» может выглядеть так:

```
Вычислить значение выражения [[X]] +[[Y]]
<br/><inputname="answer">
X=10..50
Y=60..90
```

Здесь B = «Вычислить значение выражения [[X]]+[[Y]]», P = {X, Y}, V = {10..50, 60..90}, F определена операторами «=».

Пользователь увидит текст задания с подставленными значениями, например: «Вычислить значение выражения 31 + 72», и поле для ввода ответа.

Лексический разбор html-кода шаблона, интерпретацию функции F и замену параметров шаблона из P константами из V выполняет функция генерации варианта модуля ГиП.

Параллельно с разработкой шаблона задания с параметрами (для его редактирования в составе комплекса ГИПЕРТЕСТ имеются соответствующие инструменты) необходимо запрограммировать модуль ГиП. Например, для задания «сложить два числа» требуется реализовать следующий псевдокод:

```
X = ПОЛУЧИТЬ_ПАРАМЕТР('X');
Y = ПОЛУЧИТЬ_ПАРАМЕТР('Y');
R =
ПОЛУЧИТЬ_ОТВЕТ_ПОЛЬЗОВАТЕЛЯ('answer'
);
ЕСЛИ X + Y = R ТОГДА
  ВЕРНУТЬ 1;
ИНАЧЕ
  ВЕРНУТЬ 0;
```

Средствами системы ГИПЕРТЕСТ информация о шаблоне и обработчике регистрируется в базе данных Диспетчера.

Заключительный этап разработки включает в себя добавление задания с параметрами в тест компьютерного контроля знаний. С этой целью пользовательский интерфейс Диспетчера поставляет структурированное в формате xml описание зарегистрированных модулей ГиП, на основании которого редактор учебников ГИПЕРТЕСТ при создании процедуры контроля формирует ссылку на соответствующий модуль ГиП. Приведенный ниже пример для описания теста GraphAlgorithms с заданием Blocks содержит эту ссылку в поле параметра externaltestwsdl:

```
<?xml version="1.0" encoding="UTF-8"?>
<hypertestexternaltest>
<external-
testwsdl="http://exthtu.ivc:81/VarAlg/Handler.wsdl
">
<test id="1" name="GraphAlgorithms">
<problem id="1" weight="1" name="Blocks"/>
…
</test>
</externaltest>
</hypertestexternaltest>
```

Технология разработки модулей ГиП была апробирована в процессе модернизации учебника «Алгоритмы оптимизации на графах» (http://hypertest.ispu.ru:8080/). Одно из заданий контроля предусматривает проверку навыка применения алгоритма декомпозиции графа на блоки (рис. 3) [6]. Задача декомпозиции имеет большое прикладное значение, например, при анализе графа на планарность, который выполняется при проектировании плат печатного монтажа. Модуль ГиП выполняет автоматическую генерацию графа, границы размерности которого определены в шаблоне, и номер вершины, с которой начинается поиск блоков. Вид варианта контрольного задания показан на рис. 4.

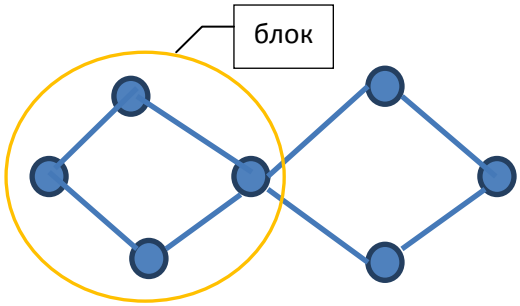

Рис. 3. Декомпозиция графа на блоки

В соответствии с идеологией параметризованных заданий, при повторном запросе этого задания числовые данные будут другими.

Заключение

Разработанная технология создания параметризованных заданий позволяет повысить надежность тестирования в системе интернет-обучения ГИПЕРТЕСТ за счет автоматизации процессов генерации текста задания и проверки ответа пользователя.

Полученные решения апробированы при разработке контрольных заданий для учебника «Алгоритмы оптимизации на графах».

	1	2	3	4	5	6	7	8	9	10	11	12
1				+			+			+		
2			+			+					+	
3		+		+								
4	+		+		+							
5				+								+
6		+						+				
7	+							+				
8						+	+		+	+		
9								+		+		
10	+							+	+			
11		+										+
12					+						+	

Рис. 4. Пример параметризованного задания в тесте контроля знаний

Параметризованные задания за счет применения индивидуальных процедур генерации текста задания и проверки ответа пользователя позволяют создавать задания практически неограниченной сложности. Они могут использоваться для контроля знаний в самых различных технических областях. Кроме того, введение новых типов данных (например, текстовых) и способов их обработки позволит применять данную технологию создания заданий и в гуманитарных областях знаний, например таких, как история или изучение иностранных языков.

Список литературы

1. **Романов А.Н., Торопцов В.С., Григорович Д.Б.** Технология дистанционного обучения в системе заочного экономического образования. – М.: ЮНИТИ-ДАНА, 2000. – 303 с.

2. **Среда** разработки программ дистанционного обучения и профильного тестирования ГИПЕРТЕСТ: инструментальные средства // Е.Р. Пантелеев, И.А. Ковшова, И.В. Малков и др. // Информационные технологии. – 2001. – № 8. – С. 34–40.

3. **Пантелеев Е.Р., Карпов Я.Э.** Разработка и интерпретация решения расчетных задач в среде Web-обучения // Информационные технологии. – 2010. – № 4. – С. 59–62.

4. **Интеграция** инструментов контроля навыков программирования в среду интернет-обучения / Е.Р. Пантелеев, А.Л. Архипов, А.В. Второв, Е.В. Ильина // Вестник ИГЭУ. – 2010. – Вып. 3. – С. 104–108.

5. **Advanced** Distributed Learning. Sharable Content Object Reference Model (SCORM) 2004 / пер. с англ. Е.В. Кузьминой. – М.: ФГУГНИИИТТ «Информика», 2005. – 29 с.

6. **SOAP** Version 1.2. W3C Recommendation 27 April 2007.

7. **Липский В.** Комбинаторика для программистов: пер. с польск. – М.: Мир, 1988. – 218 с.

Пантелеев Евгений Рафаилович,
ГОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина»,
доктор технических наук, профессор кафедры программного обеспечения компьютерных систем,
телефон/факс (4932) 26-98-26,
e-mail: erp@poks.ispu.ru

Архипов Ален Леонидович,
ГОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина»,
программист,
телефон/факс (4932) 26-98-26,
e-mail: alain@cmko.ispu.ru