

# Автоматизированная система распознавания алгоритма решения математической задачи

*В статье рассмотрена технология организации тестового задания в автоматизированной среде для распознавания выбора алгоритма решения математической задачи. Обоснован выбор иерархической структуры вопросов запроса данных для специального типа тестовых заданий. Представлена реализация указанного типа задания в программной среде openSEE.*

**Ключевые слова:** алгоритм решения задачи, тренажер, генератор задач, конечный автомат, openSEE.

## AUTOMATION PROGRAM FOR RECOGNITION OF ALGORITHM SOLUTION OF MATHEMATIC TASK

*In the article are been describing technology for manage of testing task in computer program. It was found for recognition of algorithm solution of mathematic task. There are been justified the using hierarchical structure for a special set of testing questions. Also, there has been presented the release of the described tasks in the computer program openSEE.*

**Keywords:** algorithm solution of task, computer trainer, generator input data, finite state automaton, openSEE.

### Введение

Одной из нетривиальных задач процесса обучения является контроль алгоритма решения задач. Многие контролирующие системы ориентируются на сравнение финального результата с некоторым эталонным значением. При этом не менее важно осуществлять верификацию и оценку используемого пути решения задачи. С точки зрения автоматизированной системы задача должна быть хорошо формализуемой, чтобы была возможность распознать алгоритм решения. С этой стороны математические задачи поддаются хорошей формализации и процесс решения вычислительной задачи представляет собой последовательность применяемых математических моделей, теорем и положений.

При подготовке анализа состояния проблемы решено выявить ряд систем контроля знаний известных разработчиков, а также авторских приложений, поскольку именно реализованные и воплощенные в

программный продукт идеи позволяют лучше оценить продвижение принципов контроля знаний с помощью информационных систем. В ходе обзора рассмотрены следующие приложения и сервисы: Система Интерактивного Тестирования Знаний «СИНТеЗ» ([www.sintest.ru](http://www.sintest.ru)), Программа тестирования знаний «Айрен» ([irenproject.ru](http://irenproject.ru)), программа для создания тестов и тестирования Indigo ([indigotech.ru](http://indigotech.ru)), программа MyTestX ([mytest.klyaksa.net](http://mytest.klyaksa.net)), АСТ-Тест ([www.ast-centre.ru](http://www.ast-centre.ru)), система дистанционного обучения Moodle ([moodle.org](http://moodle.org)), сервис OpenTest или StartExem ([opentest.ru](http://opentest.ru), [startexem.ru](http://startexem.ru)), LetsTest ([letstest.ru](http://letstest.ru)). Большинство настольных приложений ориентированы на организацию тестов первых трех уровней, таких как тесты на выбор ответа (первый уровень), задание на определение порядка и/или соответствия (второй уровень) и задания, предполагающие ввод ответа непосредственно с клавиатуры (третий уровень) и его вариации. Среди этих приложений – система интерактивного тестирования

СИНТеЗ, сервис OpenTest, MyTestX. Сервисы тестирования знаний в Интернете также ориентированы на такие типовые задания, при этом характерно упрощение функций для реализации в веб-приложении.

Естественно, что тесты первых трех уровней используют эталонный ответ для определения правильности введенного результата. Сервис OpenTest, среда Moodle, а также некоторые другие программы, позволяют подготовить ряд эталонных ответов с разной степенью полноты и точности, что предоставляет возможность внести ответы с допущенными ошибками и идентифицировать их. Недостаток подобных подходов заключается в анализе только конечного одного результата, при этом система тестирования не взаимодействует с пользователем на других этапах и дает возможность пользователю отметить, на основании каких предположений был выведен тот или иной ответ.

Настольные приложения СИНТеЗ, MyTestX и веб-система Moodle



**Денис Николаевич Буторин,**  
к.п.н., доцент кафедры педагогики,  
психологии и частных методик  
Тел.: (39151) 5-74-25  
Эл. почта: subritto@ngs.ru  
Филиал Красноярского  
государственного педагогического  
университета им. В.П. Астафьева в г.  
Ачинске  
afkgpu.ru

**Denis N. Butorin,**  
Ph.D (Pedagogical Science), Master of  
Science  
Tel.: (39151) 5-74-25  
E-mail: subritto@ngs.ru  
Krasnoyarsk state pedagogics university  
named after V.P. Astafyev, Achinsk branch  
afkgpu.ru

удобны тем, что позволяют форми-  
ровать и вести базу данных с ре-  
зультатами и вести полный цикл  
анализа и мониторинга знаний на-  
чиная с создания тестовых заданий  
и завершая анализом результатов и  
типами ответов. Вместе с тем сре-  
да электронного обучения Moodle  
предоставляет возможности рас-  
ширения функционала с помощью  
подключаемых модулей. Так, на-  
пример, дополнительное расшире-  
ние реализует задания типа соот-  
ветствия путем перемещения тек-  
ста или изображения на картинку  
манипулятором типа мышь мето-  
дом drag-n-drop.

Поэтому **проблема** исследова-  
ния заключается в выборе способа  
построения инструмента контро-  
ля и анализа алгоритма решения  
учебных математических задач  
в процессе обучения, используя  
традиционные виды тестов. **Цель**  
же состоит в разработке и реали-  
зации программного инструмента  
для контроля алгоритма решения  
учебных математических задач с  
использованием принципов техно-  
логий тестирования.

Для достижения цели поставле-  
ны следующие задачи:

1. Выявить и обосновать под-  
ход для контроля алгоритма реше-  
ния учебных математических за-  
дач.
2. Выявить и определить клю-  
чевые компоненты разрабатывае-  
мой программной среды.
3. Подготовить программную  
среду и продемонстрировать ее на  
типовых математических задачах.

### 1. Обоснование решения

Обоснование решение заключа-  
ется в следующем. Среди популяр-  
ных приложений и сервисов тести-  
рования выявлено преимуществен-  
ное использование тестов первого  
(выбор ответа), второго (установ-  
ление порядка или соответствия)  
и третьего (прямой ввод ответа)  
уровня. В результате чего, а также  
с учетом популярности и простоты  
реализованных типов возникает  
идея создания тестовых заданий,  
в которых контролируется реше-  
ние задачи на ряде этапов. Дей-  
ствительно, сам анализ алгоритма

решения учебных математических  
задач означает выбор определенно-  
го пути решения и фиксации выбо-  
ра маршрута решения в некоторых  
опорных точках. Поэтому идея ре-  
ализации контроля алгоритма сведе-  
на к анализу промежуточных вари-  
антов ответов, а также используе-  
мых математических моделей и со-  
ответствия между их элементами.  
Процесс решения задачи представ-  
ляет собой конструирование реше-  
ния из математических моделей  
(визуальных интерфейсных блоков  
в программе) и установление со-  
ответствия между исходными дан-  
ными и переменными моделей (ме-  
тодом drag-n-drop в интерфейсной  
части). Пути решения задачи обес-  
печиваются большим выбором мо-  
делей и их параметрами. Благодаря  
этому можно получить ясное пред-  
ставление о том, какими правила-  
ми руководствовался испытуемый.  
Ведь при очном анализе решения  
преподаватель также контролирует  
применение тех или иных формул,  
моделей и принципов на отдельных  
этапах, правильное сопоставление  
исходных данных параметрам и пе-  
ременных в формулах.

Контроль правильности и пол-  
ноты решения реализуется через  
эталонные пути решения и чис-  
ленные значения решения. На это  
есть ряд доводов. Во-первых, без  
эталонных путей необходимо ре-  
ализовать безошибочный алгоритм  
определения равносильности од-  
ной формулы другой. В рамках раз-  
работки программного комплекса  
тестирующего обучаемых это не-  
посильная задача. Так как предпо-  
лагает приведение одной форму-  
лы в другую через равносильные  
преобразования, а это необходимо  
делать в приложении на лету, с  
учетом, что тестироваться должны  
одновременно множество участни-  
ков. Кроме того, определить равно-  
сильность функций (формул в зада-  
че) можно по достаточному числу  
точек, если значения функций сов-  
падают, то можно говорить об их  
равносильности, однако данный  
метод не является точным. Метод  
предполагает совершение длитель-  
ных вычислительных процедур  
на каждую проверку. Во-вторых,  
контроль только по промежуточ-

ным значениям в формулах сводит на нет весь смысл идеи с выбором моделей и установлением соответствия между исходными данными и их переменными. Данный контроль в этом случае можно было бы реализовать и через механизмы диалогового решения задачи.

Поэтому задание, контролирующее алгоритм решения математической задачи, представляет собой комбинацию большого числа вопросов на установку соответствия и контроль идентичности промежуточных значений формул. Чтобы избежать слишком большой вариативности при конструировании решений, следует сужать круг используемых моделей (формул, функций) через ограничения в программном интерфейсе приложения.

## 2. Программная среда openSEE

Для распознавания алгоритма решения математических задач создана и развивается специальная автоматизированная система, которая представляет собой открытую среду openSEE. Открытая научная образовательная среда openSEE (open scientific educational environment, свидетельство ОФРНиО № 16723) [1] развивается в рамках проектов Красноярского государственного педагогического университета им.

В.П. Астафьева и филиала КГПУ в г. Ачинске (<http://opensee.ru>).

При разработке открытой научной образовательной среды ставятся следующие цели.

1. Облегчить рутинные процессы по обучению, структуризации научных и экспертных знаний. Повысить эффективность создания качественного образовательного продукта с широкой базой ресурсов (курсов, учебных модулей, контрольных материалов, моделей задач и т.д.). Перевести на новый качественный уровень процесс обучения, контроля знаний и компетенций, как по организации, так и по содержанию.

2. Предоставить комплексные педагогические инструменты для образовательных целей, реализовать интеграцию с другими сервисами, а также реализовать среду в качестве многофункционального открытого сервиса.

3. Подготовить платформу для реализации педагогических исследований в области изучения процесса обучения и выявления оптимальных траекторий обучения субъектов по их характеристикам [2–4]. Сюда входят конструктор диагностик и система сбора и анализа статистических данных. Это следует реализовать, чтобы не было необходимости создавать каждый раз некую систему для проведения

частных педагогических экспериментов.

В первую очередь реализуются функции контроля знаний и компетенций [5, 6]. Для этого важно выработать подход к анализу алгоритмов решения задач. Основными компонентами точки зрения архитектуры являются следующие объекты.

**Модули контроля уровней знаний.** В качестве таких модулей в первую очередь созданы компоненты оценки уровня знаний на основе узнавания, распознавания и воспроизведения, т.е. тесты первого, второго и третьего уровня. Несмотря на свою простоту и тривиальность организации, при правильной, адекватной постановке заданий тесты данных уровней являются достаточно эффективным инструментом оценки уровня знаний.

**Тренажеры компетенций.** Развитие, контроль и проверка различного рода компетенций, по мнению многих педагогов и ученых, пока возможны только при анализе решения конкретных задач. Наилучшим образом для этого подходят проблемно-ориентированные задачи. А.М. Матюшкин выделяет три основных класса проблемных задач: задачи класса поиска цели, поиска условия действия и поиска способа действия. В рамках работ автора [7–11] создана программная система ITis Learning System (зарегистрирована в ОФАП за № 10017), организующая решение проблемных задач класса поиска условия действия. Программная система успешно апробирована в КГПУ на таких дисциплинах, как «Архитектура ЭВМ» и «Компьютерные сети», поскольку именно в них чаще всего встречаются проблемные ситуации класса поиска условия действия и они поддаются четкой формализации. При этом потенциальные возможности созданной системы не ограничиваются только указанными дисциплинами.

В настоящее время ведется работа по реализации проблемных задач класса поиска способа действий. Эти задачи характерны, прежде всего, для таких естественно-научных дисциплин, как

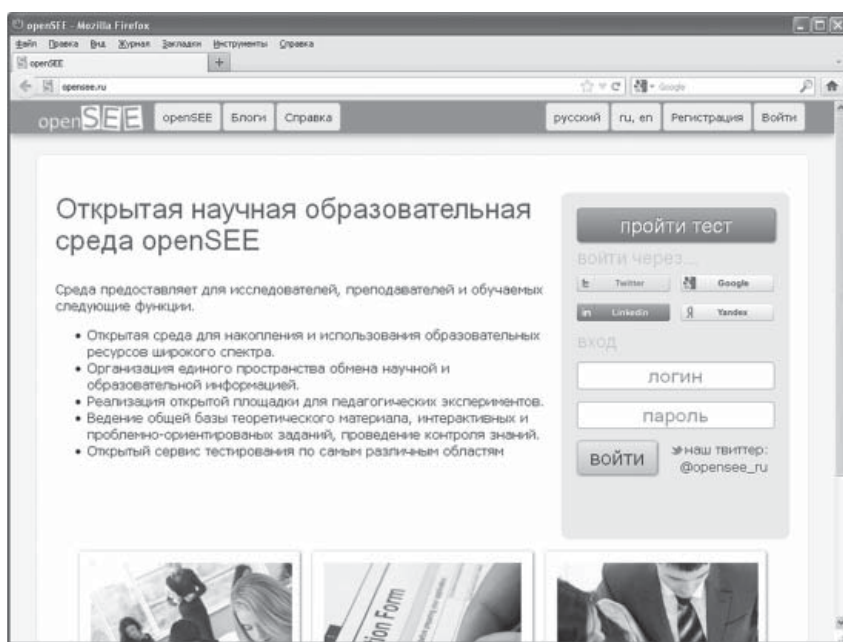


Рис. 1. Снимок экрана главной страницы автоматизированной системы

математика, физика и т.д. В основе противоречия задач такого типа лежат исходные данные и искомые величины. В качестве решения, или в терминах проблемного обучения – гипотезами, выступает математическая модель или их совокупность с указанным соответствием между входными и выходными данными, а также установленными необходимыми параметрами моделей. Здесь проявляется диалектическое единство между задачами данного класса, знания успешно формализуются с помощью семантических правил или ориентированного графа.

**Генераторы задач.** Поскольку среда представляет собой открытую систему, контроль знаний по статическим задачам будет эффективным либо в случае большого количества самих заданий, либо при реализации динамически генерируемых заданий. В связи с этим разрабатываются специальные модули-генераторы определенных задач, и возможно весьма узкоспециализированных. Например, модуль-генератор нелинейных уравнений с заданными характеристиками для решения их различными численными методами. Кроме того, важно реализовать подобные генераторы в виде конструктора для конечного преподавателя-методиста, чтобы позволить ему создавать задания сообразно психолого-педагогическим целям.

Хорошей формализации поддаются многие учебные задачи математических дисциплин класса поиска условия и способа действия. Рассмотрим задачи класса поиска условия действия. Формализованные знания в системе хранятся в виде дерева, эта структура является графическим представлением продукционных правил. Каждый узел верхнего уровня есть цель, которая требует для своего достижения условия, обозначаемые узлами нижних уровней. Любая проблема определяется как тройка: цель, условие действие, описание проблемы [12, 13]. В качестве инструментов решения задачи обучаемому представляют интерфейсные элементы проверки гипотез. Первоначально данный интерфейс реализован в виде консоли взаимодействия обу-

чаемого и среды, в которой обучаемый ищет и выбирает гипотезы для проверки, а система отвечает в формате «да/нет», выдавая дополнительную информацию о проблеме. В настоящее время ведется работа по созданию графического интерфейса по определенным предметным областям. В рамках исследовательской работы автора [11] создана программная система ITis Learning System, организующая решение проблемных задач класса поиска условия действия.

### 3. Программная реализация

Рассмотрим задачи класса поиска способа действий. Они характерны для таких естественно-научных дисциплин, как математика, физика и т.д. В большинстве математических задач присутствуют следующие элементы: набор исходных данных, выражающихся в виде чисел или переменных; описание условий или параметров, в рамках которых решается задача; а также набор неизвестных параметров (чисел, переменных и т.д.), тип которых может быть достаточно широким. Суть решения задачи заключается в следующем:

1. Необходимо правильно выбрать адекватные математические модели. В данной системе модель интерпретирована в классическом

стиле, как черный ящик, имеющий входные и выходные параметры.

2. Правильно организовать передачу данных в математические модели и пересылку промежуточных данных между ними.

3. Верно интерпретировать результаты и соответствие их искомым неизвестным.

Все три этапа соответствуют основным действиям студента и наиболее важным рубежам контроля преподавателя. Модель реализации взаимодействия обучаемого и автоматизированной системы успешным образом скрывает вычислительные процессы и другие низкоуровневые операции. Это позволяет контролировать следующие навыки: использование математических моделей в расчетах; определение дальнейших действий по решению задачи при получении определенных значений величин и их типов; оценка адекватности интерпретации полученных и требуемых результатов.

В основе противоречия задач класса поиска способа действия лежат исходные данные и искомые величины. В качестве решения задачи (гипотезой) выступает математическая модель или их совокупность с указанным соответствием между входными и выходными данными, а также установленными необходимыми параметрами моделей.

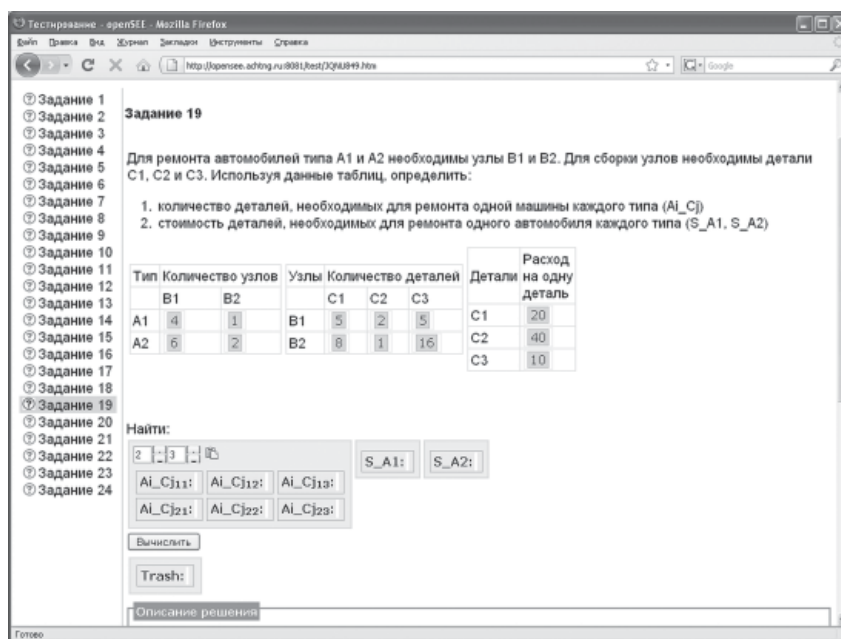


Рис. 2. Снимок экрана во время решения проблемной задачи, связанной с действиями над матрицами

Со стороны обучаемого процесс решения проблемной задачи этого класса состоит в «сборке» решения из коллекции вычислительных моделей. Обучаемый по ключевым словам или по справочнику коллекции находит необходимые модели, устанавливает соответствие между исходными данными задачи и входными параметрами модели. Таким же образом с помощью соответствия определяет интерпретацию выходных параметров моделей с искомыми данными моделей. В решении задачи может быть задействовано неограниченное число моделей и параметров.

Продемонстрируем идею на **примере**. Задача: «Для ремонта автомобилей типа А1 и А2 необходимы узлы В1 и В2. Для сборки узлов требуются детали С1, С2 и С3. Определить а) количество детали для ремонта одной машины и б) стоимость деталей для ремонта машин каждого типа». Конкретные данные в задаче указаны. Нетрудно увидеть, что для определения решения по первому пункту необходимо умножить матрицу машины-узлы на матрицу узлы-детали, а для второго пункта результат первого пункта в матричном виде умножить на вектор-столбец из расходов на каждую деталь. В обучающей системе действия студента сводятся к следующим операциям. В коллекции по ключевым словам подбираются вычислительные модели, в данном случае модель «Производство двух матриц». Устанавливаются необходимые настройки модели, для описываемой задачи это размерность входных и выходных матриц. Затем указываются соответствия между исходными данными задачи и входными параметрами моделей, а также между искомыми величинами и выходными данными моделей. В решении задачи может быть задействовано неограниченное число моделей и параметров.

Следует заметить, что в задачах обоих классов проявляется диалектическое единство, знания успешно формализуются с помощью семантических правил или ориентированного графа. Поиск решения осуществляется и проверяется с помощью прямых и обратных выводов.

Другой реализацией задачи класса поиска способа действия является задача на основе конечных автоматов. Основную идею данного вида задания составляет теория конечных автоматов. Конечный вид данного задания следует представлять в виде диалога между тестируемым и системой (преподавателем).

Подбирая соответствующим образом вопросы в данном задании (состояния конечного автомата), можно реализовать:

- поэтапный ввод данных при решении задачи;
- технологию подсказок при решении задач;
- ответ с возможными попытками и гибкой оценкой их;
- диалог из наводящих вопросов;
- диалог с системой, в ходе которого пользователь сам решает, какое действие нужно выполнить следующим.

Еще одной важной особенностью данного вида заданий является то, что для задания возможно описывать программный код для генерации исходных данных. Это позволяет:

- создавать задания исходя из педагогических потребностей преподавателя. Например, требуется дать возможность обучаемым получить опыт решения квадратных уравнений и ознакомиться с ситуацией, когда дискриминант меньше

нуля и уравнение не имеет реальных корней;

- создать исходные данные таким образом, чтобы задача решалась с ними. Например, исходные данные для систем линейных уравнений;

- создать генератор «красивых» исходных данных и/или «красивых» ответов в задании.

Общая идея заключается в том, что для успешной работы задания на основе теории конечных автоматов необходимо определиться, какую информацию следует запрашивать у пользователя. В реализуемой автоматизированной системе она может быть получена:

- через выбор вариантов ответов;
- через ввод данных в текстовые поля;
- через поиск действий по ключевым словам.

Перед тем как создавать задание, необходимо продумать этапы опроса, варианты ответов на каждый из них, а также что будет означать в педагогическом смысле выбор правильного или неправильного ответа на каждом этапе. Это необходимо, чтобы определить в задании набор следующих действий или выбрать следующий этап.

Структуру состояний конечного автомата удобно описать в виде графа. Каждый узел графа будет иметь две логические части. В одной из них описывается вариант

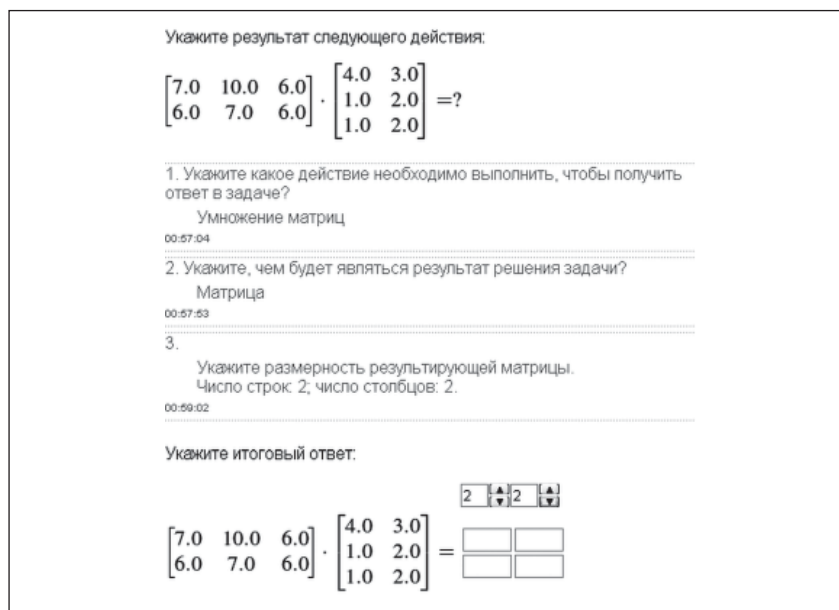


Рис. 3. Снимок экрана задачи на основе конечных автоматов

ответа пользователя, который соответствует переходу в это состояние. Другая логическая часть узла описывает тип действия, которое будет выполнено следующим, если пользователь перешел в это состояние. Граф в данном случае играет роль эталонной структуры, которая может включать различные пути решения задачи, в том числе и ложные. С учетом того, что структура поддерживает ветвления и переходы, с помощью данного подхода реализуются самые разнообразные алгоритмы решения.

Пусть есть задача умножения матриц. Алгоритм решения задачи состоит из следующих этапов:

- оценка размерности матриц;
- проверка возможности проведения операции;
- определение размерности результирующей матрицы;
- вычисление элементов матрицы.

При этом возможны ложные этапы и пути. При продвижении по правильному эталонному алгоритму решения тестируемому начисляется определенное заранее указанное количество баллов. Все действия обучаемого протоколируются.

Представленная реализация распознавания алгоритмов решения математических задач позволяет применять различный уровень сложности задач, автоматический план генерирования исходных данных. Кроме того, диалоговый режим предоставляет возможность внедрять новые виды проверок выполнения этапов. Использование данного подхода позволяет распространить применение автоматизированной системы для распознавания решения задач по физике, химии и другим естественно-научным дисциплинам.

#### 4. Результаты и выводы

В ходе апробации подхода к анализу алгоритма решения математических задач создан специальный программный модуль для системы openSEE. Разработан банк заданий, включающий 25 задач по высшей математике, а именно по темам: матрицы, векторы, систе-

мы уравнений, анализ функции и другие. Кроме того, подготовлены задания по информационным дисциплинам, таким как «Архитектура ПК» – 15 задач, «Компьютерные сети» – 20 задач и «Разработка баз данных» – 20 задач. В разработке заданий на основе конечных автоматов участвовало 5 преподавателей различных дисциплин, при этом затрачено около 6 недель на подготовку заданий и их верификацию. Разработанные тестовые задания использовались при проведении промежуточного контроля знаний по дисциплинам (12 раз), а также при проведении зачетов (4 раза) и экзаменов (3 раза).

Следует отметить, что оценка за выполнение тестового задания с применением инструментов анализа решения задач стала более гибкой. В программной среде она вычисляется автоматически путем суммирования баллов при вводе промежуточных результатов. Важно, что ввод промежуточных результатов организован в виде диалога, т.е. испытуемый выбирает путь решения задачи самостоятельно. Естественно, что разработчик теста должен тщательно продумать возможные варианты, ложные пути, а также оценку за каждый промежуточный результат. Это усложняет разработку, верификацию и скорость создания банка заданий по дисциплинам. Данное время частично компенсируется многовариантностью заданий с генерируемыми исходными данными, так как отсутствует необходимость подготавливать варианты. В среднем на одно задание по математике с 5–6 этапами решения задачи и 10–15 ветвями затрачивалось около 50 минут. За это же время преподаватели успевали подготовить упрощенный тест из 12 заданий в среде Moodle из тестов первого и второго уровня. Однако следует учитывать большую дидактическую ценность задач контроля алгоритма решения математических задач, а также задач на основе конечного автомата. Задания, включающие генераторы исходных данных, открывают возможности для их повторного многократного использования. Проблема списывания и угадывания

устранилась в достаточном объеме. Во время экспериментов на промежуточном контроле знаний в 3 группах при тестировании с контролем дисциплины преподавателем и при самостоятельной работе учащихся, анализируя результаты в программе, средняя оценка не отличалась больше чем на 10%. При тестировании без учителя результаты оказались в диапазоне 53–75% правильных ответов и с учителем – 46–70%. Учащиеся отметили удобство со своей стороны в том, что они теперь самостоятельно во время решения задачи отслеживают ее ход. Исключаются вычислительные ошибки и погрешности переписывания промежуточных результатов. Из недостатков следует отметить желание преподавателей видеть аналитическую информацию по результатам выполнения одного или группы заданий на основе конечных автоматов, наподобие тех, что предоставляют психолого-педагогические диагностики. Кроме того, при работе с задачами на конструирование решения из математических моделей необходимо начальное обучение тестируемых с принципами такого вида тестов. Обычно это требовалось провести однажды и затратить 10–20 минут на группу в зависимости от уровня навыков работы на компьютере.

Таким образом, следует заключить, что прием анализа выбора алгоритма решения математических задач разработан и успешно применен в программной среде, а значит, цель исследования достигнута.

В настоящее время открытый сервис openSEE функционирует по адресу <http://opensee.ru>. Реализованные функции позволяют организовать и проводить тестирование знаний с помощью 5 видов тестовых заданий, включая задачи на основе конечного автомата. Приглашаем всех заинтересованных лиц к использованию сервиса openSEE, а также к сотрудничеству. Дальнейшая работа может быть связана с разработкой механизма параметризации решения задачи и выявления тенденций к выбору того или иного пути решения математических задач разного вида.

## Литература

1. *Буторин Д.Н.* Электронный информационный образовательный ресурс: «Открытая научная образовательная среда openSEE» // Хроники объединенного фонда электронных ресурсов «Наука и образование». – 2011. – № 2. – Режим доступа: <http://ofernio.ru/portal/newspaper/ofernio/2011/2.doc> (дата обращения: 04.03.2011).
2. *Пак Н.И.* Проективный подход в обучении как информационный процесс: монография / Краснояр. гос. пед. ун-т им. В.П. Астафьева. – Красноярск, 2008. – 112 с.
3. *Пак Н.И.* Сущность обучения с позиций информационного подхода // Открытое образование: опыт, проблемы, перспективы. – Красноярск: Краснояр. гос. пед. ун-т им. В.П. Астафьева, 2011. – С. 3–9.
4. *Пак Н.И.* Информационный подход и электронные средства обучения: монография. – Красноярск: Изд-во РИО КГПУ, 2013. – 196 с.
5. *Андреев А.Л.* Компетентностная парадигма в образовании: опыт философско-методологического анализа // Педагогика. – 2005. – № 4. – С. 19–27.
6. *Зимняя И.А.* Ключевые компетенции – новая парадигма результата образования // Высшее образование сегодня. – 2003. – № 5. – С. 34–42.
7. *Буторин Д.Н.* Применение методов оптимизации стохастических процессов с дискретным контролем для решения задачи автоматизированного обучения в образовательном процессе // Информатизация педагогического образования: материалы международной научно-практической конференции. – Екатеринбург, 2007. – Ч. 1. – С. 124–128.
8. *Буторин Д.Н.* Интеллектуальная адаптивная обучающая система для дистанционного образования // Информатика и образование. – 2007. – № 10. – С. 126–128.
9. *Буторин Д.Н.* Различные аспекты интеллектуальной обучающей системы основанной на проблемных ситуациях // Вестник СибГАУ. – 2008. – № 2(19). – С. 100–104.
10. *Буторин Д.Н.* Практическое внедрение проективной обучающей системы в педагогическую практику // Информатика и образование. – 2008. – № 6. – С. 100–103.
11. *Буторин Д.Н.* Машинная реализация методики проблемного обучения студентов информатике в программной среде: монография / Краснояр. гос. пед. ун-т им. В.П. Астафьева. – Красноярск, 2010. – 124 с.
12. *Матюшкин А.М.* Проблемные ситуации в мышлении и обучении. – М., 1972.
13. *Махмутов М.И.* Организация проблемного обучения в школе. – М.: Педагогика, 1977.
14. *Кузнецов А.А., Семенов А.Л., Бешенков С.А., Кушниренко А.Г.* Примерная программа по информатике и ИКТ (VII–IX классы) // Информатика и образование. – 2010. – № 11. – С. 3–28.
15. *Лернер И.Я.* Проблемное обучение. – М., 1974.
16. *Селевко Г.К.* Современные образовательные технологии: учебное пособие. – М.: Народное образование, 1998. – 256 с.