

Система тестирования, основанная на генерации вопросов и тестовых заданий

В.В.Кручинин, к.т.н., заместитель директора по научной работе

Тел.: (3822)41-35-72; E-mail: kru@tcde.ru, vvkru@mail.ru

Томский межвузовский центр дистанционного образования

<http://www.tcde.ru>

М.Ф.Молочко, асп.

Тел.: (3822)41-49-60, E-mail: mmf@tcde.ru

Томский университет систем управления и радиоэлектроники

<http://www.tcde.ru>

The shortcomings of today's knowledge testing control are considered. Usage of generator programs to produce banks of test questions and sums/problems is suggested. A testing system based on such generators, implemented in the educational process, is described.



В.В.Кручинин

Введение

Тестовый контроль знаний является важным элементом дистанционной технологии обучения. Как правило, такой контроль основан на некотором фиксированном банке вопросов и тестовое задание для студента со-

ставляется из этого банка вопросов. Опыт Томского межвузовского центра дистанционного образования (ТМЦДО) свидетельствует, что через сравнительно небольшой период времени при наличии достаточно большого числа студентов (1000 и более) в Интернете появляются сайты, на которых размещаются банки правильных ответов на все вопросы из банка дисциплины. Борьба с закрытием подобных сайтов не приносит удовлетворительных результатов, поскольку эти сайты достаточно легко перемещаются в сети Интернет.

Одним из выходов из создавшегося положения при проведении тестовых контрольных работ и экзаменов является выдача уникальных, индивидуальных заданий. Однако это накладывает на преподавателей вузов дополнительные нагрузки. Например, расширение базы вопросов

для организации компьютерного тестового экзамена от 100 до 1000 наталкивается на огромное сопротивление методистов кафедр, хотя в целом этот банк вопросов проблему не решает, т.к. при достаточно большой аудитории студентов быстро составляются базы правильных ответов к ним. Основная идея решения этой проблемы состоит в том, чтобы построить программную систему генерации тестовых заданий и вопросов [1, 2], которая сможет выдать уникальное тестовое задание каждому студенту.

Генераторы вопросов и тестовых заданий

Генераторы вопросов и тестовых заданий

Генератор – это программа, которая создает банк тестовых вопросов с правильными вариантами ответов и выдает этот банк в некотором заданном формате [2,3]. Полученное таким образом описание поступает на вход некоторой системы компьютерного тестирования, и производится опрос студента, затем ответы студента сравниваются с эталонными ответами, полученными в ходе генерации банка, затем на основе некоторого алгоритма ставится оценка.



М.Ф.Молочко

Генераторы можно разделить на два больших класса: задач и вопросов. Для генератора задач строят шаблон задачи, который содержит: текстовую формулировку задачи, границы изменения параметров задачи, алгоритм генерации параметров задачи, алгоритм решения задачи.

Генератор вопросов – это программа, которая создает описание вопроса, используя базу знаний. Здесь существует огромное количество разнообразных моделей: фреймовые, сетевые, табличные, продукционные, логико-лингвистические и другие [2,3]. Однако чем сложнее модель генератора, тем труднее построить алгоритм семантической проверки сгенерированного вопроса. В простейшем варианте генератора меню-вопросов можно использовать следующую схему:

1) дается описание некоторой ситуации, фрагмента текста и прочее;

2) записывается список правильных утверждений, применительно описанной выше ситуации;

3) записывается список неправильных утверждений;

4) случайно выбираются два верных и два неверных утверждения;

5) вопрос формируется следующим образом:
Дано <описание ситуации> и следующие утверждения:

<утверждение 1>;

<утверждение 2>;

<утверждение 3>;

<утверждение 4>.

Выберите верные (или неверные).

Рассмотрим основные шаги алгоритма генератора:

1. Выбор параметров и элементов задания – это, собственно говоря, шаг, на котором производится синтез задания. Выбор значений параметров и элементов задания осуществляется, как правило, на основе использования датчика случайных чисел. Это в некоторых случаях существенно снижает параметры генерирующего алгоритма. Если параметры генерировать от входных параметров студента,

то необходимо подбирать хеш-функции, которые вычисляли бы значения ключей. Хорошие результаты дает использование алгоритмов нумерации, которые перечисляют все варианты заданий данного алгоритма генерации[3].

2. Проверка семантики задания – это важный шаг, поскольку не всякая комбинация параметров может привести к решению задачи. В простейшем случае необходимо проверить некоторую формулу, например, подкоренное выражение не должно быть меньше нуля. В более сложных случаях необходимо разрабатывать специальный алгоритм семантического контроля.

3. Формулировка задания – это заключительный шаг генерации, на котором производится преобразование внутреннего представления тестового задания на некоторый декларативный язык. Поскольку текст задания может содержать формулы, рисунки, таблицы и прочее, то нужно использовать язык описания документа для какого-либо конкретного пакета.

Необходимо также отметить, что если задание можно разбить на части, то приведенный обобщенный алгоритм можно применить к каждой отдельной части, поэтому весь генератор может представлять собой многократную итерационную схему.

Перечислим основные свойства генератора:

1) мощность генератора;

2) управление последовательностью генерации;

3) управление сложностью тестового задания;

Мощность генератора – это количество возможных вариантов заданий, которые может получить генератор. При разработке генератора нужно оценить его мощность, в некоторых случаях это довольно трудно сделать, поскольку может оказаться довольно сложная связь между отдельными генерируемыми элементами задания.

При проведении тестирования в компьютерных классах встает задача получения тестовых заданий на двух соседних компьюте-

рах. Здесь необходимо получить задания, которые, по крайней мере, не были бы похожи друг на друга. Вследствие этого встает задача установки семантического расстояния между двумя сгенерированными заданиями и управления генерацией с учетом этого расстояния.

Еще одним важным свойством генератора является возможность управления сложностью задания. В простейших случаях генератор выдает задания одной и той же сложности. В некоторых технологиях требуется генерировать задания с разной сложностью, начиная, как правило, с простых и заканчивая самыми сложными заданиями.

Система тестирования

Технология контроля знаний, принятая в ТМЦДО, предполагает использование компьютерных тестов в проведении промежуточного и итогового контроля [4]. Для промежуточного контроля используются компьютерные контрольные работы, а для итогового – компьютерный экзамен. Вследствие этого система тестирования состоит из двух составных частей:

- подсистемы проведения компьютерных контрольных работ;
- подсистемы приема компьютерных экзаменов.

Обе подсистемы поддерживают оригинальный язык описания теста [3]. Все экзамены и контрольные работы реализованы в виде описания на этом языке и хранятся в виде отдельных закодированных файлов. Генераторы в системе реализованы как динамически загружаемые библиотеки (DLL).

Первая система отправляется студенту и должна работать на его компьютере. Вторая работает в представительстве ТМЦДО. Причем в начале сессии она устанавливается, а в конце сессии удаляется. Подсистема проведения экзаменов ставит отметку непосредственно после окончания экзамена. Протокол проведения экзамена кодируется и записывается в базу протоколов. База протоколов создается на каждом локальном компьютере или одна в локальной сети. Эта база может быть исполь-

зована для контроля за ходом проведения экзаменационной сессии.

На рис. 1 показана структура системы проведения контрольных работ.

Запишем основные элементы этой структуры.

При запуске системы проведения контрольных работ модуль регистрации ищет регистрационный файл, в котором хранится следующая информация:

- учебное заведение;
- группа;
- фамилия, имя, отчество;
- идентификационный код студента.

Если файл не будет найден, то подсистема выдает окно для ввода регистрационной информации. После ввода регистрационной информации вызывается модуль загрузки теста или модуль загрузки генератора.

Модуль загрузки теста производит поиск, загрузку и декодирование файла, содержащего необходимый тест. Имя файла теста должно быть записано в командной строке при запуске системы. Если файл не будет найден, то модуль выдаст соответствующее сообщение об ошибке.

Модуль загрузки генератора производит поиск, загрузку DLL, генерацию теста в формате языка системы. Имя файла DLL должно быть записано в командной строке при запуске системы. Если файл не будет найден, то модуль выдаст соответствующее сообщение об ошибке. Если загрузка прошла успешно, то производится генерация теста и запись его в специальный буфер системы. После генерации модуль сохраняет параметры генерации теста в специальную структуру.

Транслятор производит синтаксический анализ описания теста и преобразование его во внутренний формат системы. Алгоритм разбора основан на методе рекурсивного спуска. При обнаружении ошибки транслятор выдает соответствующее сообщение.

Модуль случайной выборки производит анализ теста, для каждой темы случайным образом выбирает заданное число вопросов и формирует единый список вопросов для про-

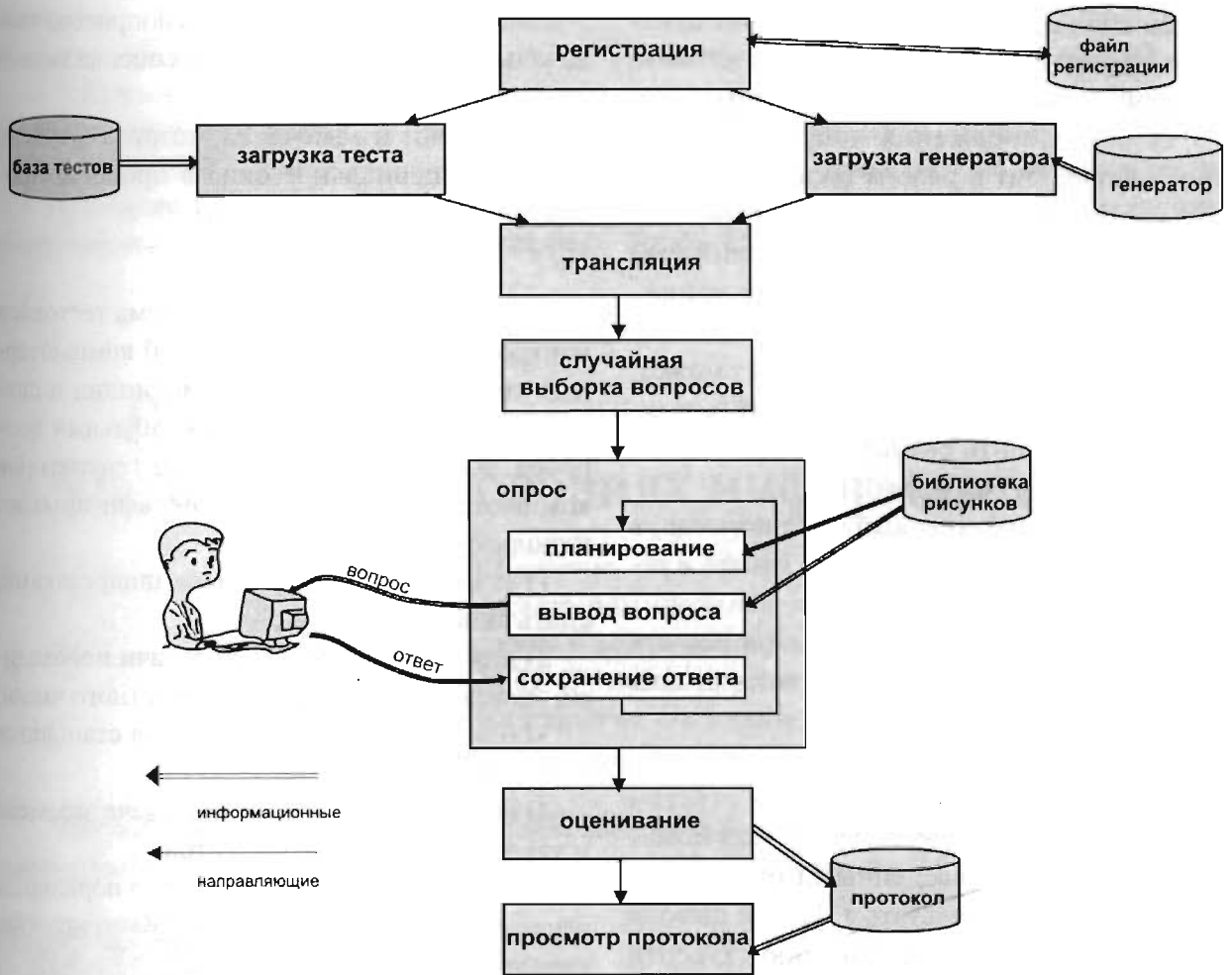


Рис. 1. Обобщенная структура системы проведения компьютерных контрольных работ

ведения тестирования. Выборка производится на основе перемешивания номеров вопросов, входящих в данную тему.

Модуль опроса выполняет следующие функции:

- 1) навигацию по списку вопросов (назад, вперед, на первый, на последний, по номеру вопроса);
- 2) вывод вопроса на экран с заданными параметрами отображения;
- 3) ввод и сохранение ответа студента.

Процесс вывода вопроса является самым сложным, т.к. он может содержать текст, таблицы, рисунки, формулы, окна ввода и области выбора. Размер шрифта и разрешающая способность экрана могут быть разными. Важным элементом модуля опроса является подсистема

планирования размещения объектов в клиентском окне системы. Составной частью подсистемы планирования является программа планирования размещения формул. На вход этой программы подается строка символов с описанием формулы, записанной на языке представления формул [3] с заданным размером шрифта. На выходе выдаются размеры и параметры прямоугольной области отображения формулы. Результатом работы подсистемы планирования является список объектов отображения с указанием координат размещения в клиентском окне системы проведения контрольных работ.

Далее последовательно производится вывод списка объектов размещения в клиентское окно системы. Процесс планирования и

отображения производится при любом изменении параметров отображения (листания, изменении размеров окна или шрифта).

После отображения вопроса модуль опроса переходит в режим ожидания. Студент должен ввести ответ или нажать одну из управляющих клавиш: «предыдущий вопрос», «следующий вопрос», «оценивание» и «выход».

При работе этого модуля студент может:

- изменить параметры отображения (увеличить или уменьшить размеры шрифта, изменить цвет фона и символов);
- вывести гипертекстовый файл помощи, в котором записаны способы ввода ответа и режимы работы систем;
- ввести записи в специальное окно, содержание которого по окончании тестирования записывается в протокол теста.

Модуль оценки предназначен для определения оценки уровня знаний по ответам студента. Происходит сравнение эталонных значений ответов с введенными ответами и выдается окно с результатом. После вывода оценки программа по требованию студента производит запись протокола проведения контрольной работы в файл. При этом протокол кодируется специальным алгоритмом кодирования.

Студенту дается возможность просмотра протоколов после их создания. Структура протокола следующая:

1. Параметры контрольной работы: вуз, кафедра, разработчик контрольной работы, имя файла контрольной, название курса, номер контрольной работы, идентификационный номер контрольной в базе контрольных работ.
2. Параметры студента: фамилия, имя, отчество, номер группы, идентификационный номер в базе данных студента.
3. Дата и время начала и завершения контрольной работы.
4. Список номеров вопросов, ответы студента на каждый вопрос, указание об успешности сравнения.

5. Параметры оценки: всего вопросов, число успешно отвеченных, общая оценка за контрольную работу.

6. Заметки и замечания, которые студент вносил в специальное окно в процессе проведения контрольной работы.

Заключение

1. В настоящее время система тестового контроля ТМЦДО содержит 400 компьютерных тестов для различных дисциплин и специальностей, общим объемом 50 тысяч вопросов. Кроме этого имеется 20 генераторов компьютерных тестов. Уже первый опыт их использования показал, что:

- 1) студенты, пользующиеся шпаргалками, сдать экзамен не могут;
- 2) для решения тестовой задачи необходимо знание алгоритма, а не конкретного числа;
- 3) базы заготовленных ответов становятся уже не нужными;
- 4) не поможет студенту при сдаче экзамена и украденный генератор тестов.

Все это приводит к тому, что нерадивый студент уже не может сдать экзамен, поэтому качество обучения возрастает.

2. Переход на использование генераторов предъявляет новые требования к программно-методическому обеспечению:

- 1) вместо банка тестовых задач или вопросов необходимо создавать банки шаблонов задач или базу знаний;
- 2) поскольку генератор может получать огромное количество вопросов, то отладка таких программ становится достаточно сложным делом;
- 3) отдельной проблемой для использования генераторов является необходимость хранения параметров вопросов в протоколе экзамена.

Литература

1. Кручинин В.В. Разработка компьютерных учебных программ. – Томск: изд-во Томск. ун-та. 1998. – 211с.
2. Башмаков А.И. Разработка компьютерных и обучающих систем / Башмаков А.И., Баш-

маков И.А. – М.: Информационно-издательский дом «Филинь», 2003. – 616с.

3. Кручинин В.В. Генераторы в компьютерных учебных программах. – Томск: Изд-во Томск. ун-та, 2003. – 200с.

4. Исакова О.Ю. Технология проведения компьютерных экзаменов и контрольных ра-

бот в Томском межвузовском центре дистанционного образования / Исакова О.Ю., Кручинин В.В. // Дистанционные образовательные технологии. Пути реализации: Сб. научн. тр. – Вып.1. – Томск: Изд-во Томского гос. ун-та систем управления и радиоэлектроники, 2004 – С. 142-148.