



Рисунок 3. Главное окно программы «Архитектура компьютера»

Элемент «ЛЕКЦИОННЫЙ МАТЕРИАЛ» означает модуль лекционного материала, где содержится информация по таким разделам как:

- Понятие об архитектуре компьютера;
- Архитектура микропроцессора;
- Программирование на ассемблере;
- Эволюция архитектур микропроцессоров и микро-ЭВМ;
- Внешние устройства компьютера.

Элемент «Лабораторные работы» содержит лабораторный практикум по данной дисциплине.

Элемент «СРС/СРСП» открывает окно с планом лабораторных занятий и СРСП.

В элементе «Об Авторе» можно просмотреть портфолио автора программы.

Данный курс связан с изучением одного из разделов современной информатики и предназначен для формирования представлений об основных понятиях архитектуры персонального компьютера (ПК). Так как тенденция развития вычислительной техники растет с каждым годом, то мы считаем, что будущим специалистам нужны базовые знания об архитектуре современного ПК.

Разработка тестирующего программного комплекса в виде web-приложения

Автор: Расчетов И. С.

Научный руководитель: Радченко П.Н. Магистр информатики, ст.преподаватель

Костанайский государственный педагогический институт

Контрольное тестирование и устный опрос являются одними из наиболее широко используемых и хорошо разработанных средств проверки знаний в высшем образовании. Классический тест представляет собой последовательность достаточно простых вопросов. На каждый вопрос имеется простой ответ, который может быть формально проверен и оценен как правильный, неправильный или частично правильный (например, неполный). Вопросы обычно классифицируются по типам соответственно типу ожидаемого ответа. Классические типы вопросов делятся на вопросы типа [да/нет], вопросы типа [много вариантов/один ответ] (МВ/ОО), вопросы типа [много вариантов/много ответов] (МВ/МО) и вопросы открытого типа с текстовым или числовым ответом. Более «продвинутые» типы вопросов включают вопросы на соответствие, вопросы на правильную последовательность, вопросы на указывание (ответ – одна или несколько областей на рисунке), а также графические вопросы (ответ – простой граф). Кроме этого, каждая предметная область может иметь некоторые специфические типы вопросов.

Тестирующие и опрашивающие компоненты были первыми интерактивными компонентами, примененными в Web-основанном обучении (WBE – web-based education), и в настоящее время они являются наиболее разработанными. Существующие WBE системы отличаются по многим аспектам, касающимся выполнения контрольных тестов и опросов. При выборе современной технологии для разработки и выдачи Web-основанных контрольных опросов в Технологическом Университете Д. Карнеги была создана комплексная методика сравнения доступных систем. Данная статья содержит исчерпывающий обзор характеристик, важных при оценке современных технологий Web-тестирования.

Чтобы сравнить существующие альтернативы, мы проанализировали жизненный цикл вопроса в Web-основанном обучении (Таблица 1). Мы разделили жизненный цикл вопроса в три стадии: подготовка (до активной жизни), выдача (активная жизнь), и оценка (после активной жизни). Каждая из этих стадий далее разделена на меньшие стадии. Для каждой из этих стадий мы исследовали набор возможных технологий поддержки.

Жизнь вопроса начинается с его создания. WBE-система на стадии создания должна помочь автору, предоставляя технологию и инструментарий для создания вопроса. Все создаваемые вопросы (их содержание и метаданные) хранятся в системе. Активная жизнь сохраненного вопроса начинается, когда он выбран для представления как часть тестирования или опроса. Этот выбор мог быть сделан как статически, преподавателем во время разработки курса, так и динамически, системой во время выполнения (случайно или согласно некоторой модели обучения).

Затем, система выдает вопрос: она выводит вопрос на экран, предоставляет обучаемому интерфейс для ответа и принимает ответ для оценки. На стадии оценки система должна произвести следующие действия: оценить ответ как правильный или неправильный, выдать обучаемому

информацию обратной связи о правильности ответа, проставить бал и сделать запись о действиях обучаемого.

Существующие WBE–средства и системы значительно различаются по типу и степени поддержки, которую они обеспечивают на каждой из упомянутых выше стадий. Простые системы обычно обеспечивают частичную поддержку некоторых стадий. Продвинутое системы обеспечивают всестороннюю поддержку всех перечисленных стадий. Мощность системы и степень обеспечиваемой поддержки серьезно зависят от уровня технологий, используемых на основных стадиях (подготовка, выдача и оценка). Ниже мы анализируем альтернативы, исследуемые в настоящее время:

До	Во время	После
Подготовка	Выдача	Оценка
Создание	Представление	Оценка
Хранение	Взаимодействие	Проставление баллов и запись
Выбор	Получение ответа	Выдача обратной связи

Таблица 1. Стадии жизненного цикла тестового вопроса.

Вопросы создаются авторами: педагогами и разработчиками. Современный вопрос имеет следующие компоненты: сам вопрос (основа), набор возможных ответов, указатель на правильные ответы, тип интерфейса представления, информация обратной связи, выдаваемая обучаемому вне зависимости от его ответа и специфическая обратная связь для каждого варианта ответа. Кроме того, автор может использовать метаданные, такие как темы разделов, ключевые слова, часть курса, соответствующая тесту, вес или сложность вопроса, допустимое время, число попыток и т.д. Эти метаданные могут использоваться для выбора выдаваемого вопроса, а также для проставления баллов.

Варианты поддержки на стадии создания обычно зависят от технологии, используемой для хранения отдельного вопроса в системе. В настоящее время, нам известно два различных способа хранения вопроса: в формате представления и во внутреннем формате. В контексте Web–основанного обучения, хранение вопроса в формате представления означает его хранение как части HTML–кода (обычно в виде HTML–формы). Такие вопросы могут также называться статическими вопросами. Они являются «черными ящиками» для WBE–системы. Система может представлять статические вопросы только «как есть» (в том виде, в котором они были созданы). Создание вопросов этого типа часто не поддерживается WBE–системой, так как это может быть сделано в любом HTML–редакторе.

Хранение вопроса во внутреннем формате обычно означает его хранение в записи базы данных, где различные части вопроса (основа, ответы, и обратная связь) сохранены в различных полях этой записи. Вопрос, как его видит обучаемый, генерируется из внутреннего формата на стадии

выдачи. Внутренний формат дает возможность более гибкого манипулирования вопросом: тот же самый вопрос может быть представлен в различных формах (например, открытый тип или множественный выбор) или с помощью различных интерфейсов (например, набором иконок или списком выбора). Порядок альтернатив в вопросах множественного выбора может меняться. Это обеспечивает более высокий уровень индивидуализации. Это полезно с педагогической точки зрения и снижает возможность обмана. Существует два основных подхода к созданию вопросов во внутреннем формате: графический пользовательский интерфейс (GUI) и специальный язык разметки вопросов. Каждый из этих подходов имеет свои достоинства и недостатки. В настоящее время, подход, основанный на использовании GUI, намного более популярен. Он используется всеми продвинутыми коммерческими WBE-системами, такими как [Blackboard 1998; Question Mark 1998; WBT Systems 1999; WebCT 1999]. Однако, некоторые WBE-системы используют GUI, но не хранят вопросы во внутреннем формате. Вместо этого, такие системы сразу генерируют HTML-вопросы и сохраняют их в статической форме.

Наилучшим выходом для хранилища вопросов является статический тест или опрос, то есть статическая последовательность вопросов. Опрос обычно представляется в форме HTML и создается с помощью авторских средств HTML-уровня. Статические тесты и опросы обычно жестко привязаны к некоторому определенному месту курса. При использовании этой, самой простой, технологии возникает одна проблема; все обучаемые получают одни и те же вопросы в одном и том же месте курса. Другая проблема состоит в том, что каждый вопрос, жестко встроенный в тест не может повторно использоваться. Лучшим вариантом для хранилища вопросов является поддерживаемый вручную фонд вопросов. Фонд может разрабатываться и поддерживаться группой преподавателей. Каждый вопрос в фонде – обычно статический, однако опросы – более гибки. Несложные средства управления фондом позволят преподавателям многократно использовать вопросы; все опросы могут быть собраны и добавлены к содержанию курса, когда это необходимо. Такой подход принято называть гибкостью времени создания. На следующий год тот же самый курс, следующая версия курса, или даже различные разделы в пределах того же самого курса могут включать в себя различные опросы без необходимости разрабатывать эти контрольные опросы с нуля.

Еще лучшим вариантом будет формирование на основе поддерживаемого вручную фонда вопросов базы данных вопросов. База данных добавляет то, что принято называть гибкостью времени выдачи. В отличие от поддерживаемого вручную списка, база данных формально структурирована и доступна системе выдачи вопросов. При наличии базы данных вопросов не только преподаватель может формировать опрос, когда это необходимо, но и сама система может генерировать опрос из набора вопросов. Естественно, вопросы могут быть выбраны случайно и помещены в опрос в случайном порядке. В результате, всем обучаемым могут быть

предложены индивидуальные опросы (то, чего преподаватель не может обеспечить вручную), что значительно снижает возможность обмана. Обратите внимание, что ведение базы данных вопросов не требует использования коммерческой СУБД. Продвинутое университетские системы, такие как QuestWriter [Bogley et al. 1996] или Carnegie Mellon Online [Rehak 1997] и многие коммерческие системы, такие как TopClass [WBT Systems 1999] или LearningSpace [Lotus 1999] используют готовые базы данных типа ORACLE или Lotus Notes для хранения фондов вопросов во внутреннем формате. Впрочем, имеются также системы, которые успешно имитируют базы данных с файловой системой UNIX, используя специально структурированную систему каталогов и файлов.

Проблема всех систем с автоматической генерацией опросов заключается в том, как обеспечить надлежащий набор вопросов, составляющих опрос. Самый простой способ достижения этого состоит в организации специализированных баз данных вопросов для каждого урока. Этот подход, примененный, например, в WebAssessor [ComputerPREP 1998], уменьшает возможность повторного использования одного и того же вопроса на различных уроках. Многие продвинутое системы подобно TopClass [WBT Systems 1999] могут поддерживать многочисленные фонды вопросов и использовать несколько фондов для генерации каждого опроса. Имея такой уровень поддержки, преподаватель может организовать отдельный фонд для каждой темы или каждой степени сложности вопросов и определять желаемое количество вопросов в сгенерированном опросе, принадлежащих тому или иному фонду.

Тип интерактивной технологии, используемой для получения ответов обучаемого, является одной из наиболее важных характеристик WBE-систем. Он определяет всю функциональность на стадии выдачи вопросов, а также влияет на стадии создания и оценки вопросов. В настоящее время, различают пять технологий: HTML-ссылки, HTML/CGI-формы, скриптовые языки, внедрение (plug-in) и Java.

HTML-ссылки – самая простая технология взаимодействия, реализующая набор возможных ответов как список HTML-ссылок. Каждая ссылка связана с определенной страницей обратной связи. При использовании этого подхода возникает две проблемы: сложность создания вопросов (логика вопроса должна быть жестко встроена в гипертекст курса) и поддержка всего двух типов вопросов: [да/нет] и [МВ/ОО]. Эта технология использовалась в основном на заре Web-основанного обучения, когда более продвинутое технологии взаимодействия, такие как CGI, JavaScript или Java еще не были разработаны [Holtz 1995].

Наиболее популярной технологией Web-тестирования, используемой в настоящий момент многочисленными коммерческими и университетскими системами является комбинация HTML-форм и CGI-скриптов. HTML-формы чрезвычайно удобны для представления основных типов вопросов. Вопросы типа [да/нет] и [МВ/ОО] представляются наборами иконок, списками выбора, всплывающими меню. Вопросы [МВ/МО] представляются

списками множественного выбора или наборами переключателей. Вопросы открытого типа реализуются в виде полей редактирования. Более продвинутые вопросы, такие как вопросы на соответствие или на правильную последовательность, также могут быть реализованы, при помощи форм. Кроме того, скрытые поля могут использоваться для хранения дополнительной информации о тесте, в которой может нуждаться CGI-скрипт. Значительные преимущества от использования технологии «стороны сервера» (к которой относится и технология «форма/CGI») и схожей с ней технологии «карты стороны серверной» возникают при реализации графических вопросов на указывание. Разработка теста относительно проста и может быть осуществлена с помощью HTML-редактора. Секретная информация, необходимая для вынесения оценки (такая, как параметры вопроса, ответы, обратная связь) может для безопасности храниться на стороне клиента, обеспечивая защиту обучаемых от кражи вопросов (единственной внешней информацией, необходимой хорошо разработанной системе, для оценки теста является идентификатор теста и идентификатор студента). Организация оценки на стороне сервера облегчает реализацию всех функций, необходимых на стадии оценки (таких, как запись результатов, проставление бала, выдача обратной связи). Все эти функции могут выполняться одним и тем же скриптом на стороне сервера. Основная проблема технологии «стороны сервера» – малая выразительная мощность. Эта технология хорошо реализует только основные типы тестов. Более продвинутые и более интерактивные типы тестов (например, тесты, использующие технологию drag-and-drop – перетаскивание объектов) не могут быть реализованы с помощью только технологии «стороны сервера». Создать вопрос с оценкой на стороне сервера довольно непросто потому, что функциональные возможности вопроса распространены между его HTML-представлениям (создаваемым вручную или сгенерированным автоматически) и CGI-скриптом, реализующем оценку. Другая серьезная проблема заключается в том, что вопросы, основанные на CGI, не работают, когда связь пользователя с сервером нарушается или очень замедляется.

Как мы уже говорили, выбор технологии взаимодействия заметно влияет на функциональные возможности стадии оценки. Оценка – это стадия, на которой ответы обучаемых определяются как правильные, неправильные или частично правильные (например, неполные). Обычно, правильные и неправильные ответы готовятся во время создания, так что оценка является или жестко встроенной в вопрос, как это делается в вопросах типа [МВ/ОО], или осуществляется путем простого сравнения (в вопросах открытого типа). Существует несколько случаев, требующих более продвинутой технологии оценки. В некоторых предметных областях может возникнуть ситуация, когда правильные ответы могут не иметь побуквенного соответствия с хранимым правильным ответом. Примеры могут служить: набор неупорядоченных слов, вещественное число, простое алгебраическое выражение. В этой ситуации требуется программа сравнения. Некоторые системы могут применять для этих целей специальные интеллектуальные

технологии. Наконец, в некоторых случаях для оценки ответов необходим «эксперт предметной области», такой как интерпретатор Lisp для программирования на Lisp, как в системе ELM-ART, или компьютерная алгебраическая система для алгебраических предметных областей. Первые два варианта оценки очень просты и могут быть реализованы с использованием любой интерфейсной технологией – даже JavaScript, может использоваться, чтобы написать простую программу сравнения. Если же требуются более сложные вычисления (как в случае интеллектуального соответствия ответа), приходится обращаться к полноценному программированию на PHP, Java или программе на стороне сервера, использующей CGI-интерфейс. Если для оценки требуется «эксперт предметной области», то единственной альтернативой в настоящее время является запуск «эксперта» на стороне сервера с CGI-шлюзом. Фактически, многие системы типа «эксперт предметной области» (например, компьютерная алгебраическая система Mathematica) имеют CGI-шлюз.

Обычно в функции обратной связи входит: сообщение о том, что ответ правильный, не правильный или частично правильный; выдача правильного ответа, обеспечение индивидуальной обратной связи. Индивидуальная обратная связь может сообщать, что является верным в правильном ответе, что неверно в неправильном и частично неправильном ответе, обеспечивается некоторая мотивационная обратная связь, обеспечивать мотивирующую обратную связь, а также выдавать информацию или ссылки для продолжения обучения. Вся индивидуальная обратная связь обычно создается и хранится вместе с вопросом. Система, включающая определенные понятия или разделы с проставленными весовыми коэффициентами как часть метаданных вопроса, может обеспечивать качественную корректирующую обратную связь без непосредственного создания, так как она «знает», какие знания пропущены, и где их можно найти. Это означает, что мощность обратной связи определяется технологией создания и хранения вопросов. Количество информации, предоставляемое обратной связью, зависит от контекста. При самооценке обучаемый обычно получает всю возможную обратную связь – чем больше, тем лучше. Эта обратная связь – очень важный дидактический ресурс. В случае официального тестирования обучаемому обычно не предоставляется ни правильного варианта ответа, ни, даже, является ли его ответ правильным. Единственной обратной связью по окончании всего теста может быть общее число правильных ответов в тесте. Это значительно снижает возможность обмана, но также и возможность обучаться. Для поддержки процесса обучения, многие существующие WBE-системы делают оценку менее строгой и обеспечивают больше обратной связи, пытаются бороться с обманами другими средствами. Единственный способ объединить обучение и строгую оценку состоит в использовании более продвинутых технологий, позволяющих получить неограниченное число вопросов: использование параметризованных вопросов и генерация тестов на основе

базы знаний. В этой ситуации WBE–система может обеспечить полную обратную связь, препятствуя при этом обману.

Если тест предназначен исключительно для самооценки, то генерация обратной связи должна быть главной обязанностью WBE–системы на послетестовой стадии. Обучаемый – единственный, кому необходимо видеть результаты тестирования. В контексте оценки, главной обязанностью WBE–системы в процессе тестирования является проставление баллов по итогам выполнения обучаемым теста и запись этих данных для будущего использования. Окончательный балл и другие результаты теста важны для преподавателей, администраторов курса и непосредственно обучаемых (многие авторы отмечают, что возможность видеть свои результаты в режиме онлайн, отмечается большинством обучаемых как положительная особенность WBE–систем). Ранние WBE системы обеспечивали довольно ограниченную поддержку преподавателя на стадии оценки теста. Результаты или посылались преподавателю по электронной почте или регистрировались в специальном файле. В обоих случаях преподавателю приходилось ставить окончательный балл и делать записи лично: обработать результаты теста и поставить окончательный балл, сделать запись о результатах и обеспечить доступ к ним всем заинтересованным лицам согласно политике университета. Такой вариант просто реализовать, и он не требует от преподавателя изучения новых технологий. Вследствие последней причины возможность использования этой технологии все еще предусматривается в нескольких продвинутых системах [Carbone и Schendzielorz 1997]. Однако, если система которая не обеспечивает никаких других возможностей для ведения записей и проставления оценок, она явно находится ниже современного уровня. На сегодняшний день современная WBE–система должна быть способна оценить результаты теста автоматически и занести их в базу данных. Кроме того, она должна поддерживать несколько уровней ограничения доступа к записям для студентов, преподавателей и администраторов. Ограничения обычно определяются политикой университета.

Угроза из космоса – актуальная проблема человечества

Автор: Герасимов Е.Ф., студент 3 курса

Научный руководитель: Шевченко И.М., магистр физики

Костанайский государственный педагогический институт

Над проблемами мирового масштаба во все времена думали и ломали голову самые лучшие умы истории, и, конечно же, со временем, многие проблемы были решены, но многие другие так и остались нерешенными.

Эти проблемы характеризуются динамизмом, возникают как объективный фактор развития общества и для своего решения требуют объединённых усилий всего человечества. Глобальные проблемы взаимосвязаны, охватывают все стороны жизни людей и касаются всех стран мира.