

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Государственное образовательное учреждение
Высшего профессионального образования
Алтайский государственный технический университет
им. И.И.Ползунова



НАУКА И МОЛОДЕЖЬ – 2008

V Всероссийская научно-техническая конференция
студентов, аспирантов и молодых ученых

СЕКЦИЯ

ИНФОРМАЦИОННЫЕ И ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

подсекция

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И
АВТОМАТИЗИРОВАННЫХ СИСТЕМ**

Барнаул – 2008

ББК 784.584 (2 Рос 537) 638.1

V Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых "Наука и молодежь – 2008". Секция «Информационные и образовательные технологии». Подсекция «Программное обеспечение вычислительной техники и автоматизированных систем». / Алт. гос. техн. ун-т им. И.И.Ползунова. – Барнаул: изд-во АлтГТУ, 2008. – 64 с.

В сборнике представлены работы научно-технической конференции студентов, аспирантов и молодых ученых, проходившей в 24 апреля 2008 г.

Организационный комитет конференции:

Максименко А.А., проректор по НИР – председатель, Марков А.М., зам. проректора по НИР – зам. председателя, Стопорева Т.А. – ответственный секретарь Центра НИРС – секретарь оргкомитета, Кантор С.А., заведующий кафедрой «Прикладная математика» АлтГТУ – руководитель секции.

Научный руководитель подсекции: к.ф.-м.н., профессор, Кантор С.А.

Секретарь подсекции: к.т.н., доцент, Сорокин А.В.

Редактор сборника: Сорокин А.В.

© Алтайский государственный технический университет им. И.И.Ползунова

СОДЕРЖАНИЕ

Карпенко Д.А., Крючкова Е.Н. Система оценки наличия плагиата и качества программ.....	5
Данилюк Н.П., Бубнова Н.Д. Разработка графического редактора с расширенными возможностями.....	6
Кондраханов О.В., Кузьмин А.А., Кузьмин А.Г. Разработка программного комплекса для обработки тепловизионных снимков.....	7
Ушмодина Е.Г., Кайгородова М.А. Использование информационных систем в управлении интеллектуальной и промышленной собственностью.....	8
Ушмодин Н.А., Шальнев А.А. Использование полнотекстового поиска для нахождения документов.....	11
Сартакова Е.А., Кайгородова М.А. Использование web-технологий в управлении качеством образовательных услуг.....	12
Свиридова Н.С., Тушев А.Н. Компьютерная система тестирования знаний как элемент обратной связи системы управления учебным процессом вуза.....	15
Зрюмов П.А., Зрюмов Е.А. Аналитический обзор методов обработки данных цветовой температуры объекта.....	16
Незамаев А.А., Лукоянычев В.Г. Программная реализация комплекса лабораторных работ по электротехнике.....	17
Аксёненко Д.М., Корнева Г.А. Программное обеспечение для оптимизации конструктивных размеров деталей на основе критериев напряженно-деформированного состояния.....	18
Калашник Е.С., Тушев А.Н. Программное обеспечение для обработки кардиосигнала в холтеровских мониторах на базе мобильных устройств.....	19
Чанов В.В., Семкин Б.В. Разработка программного инструментария для статистических исследований качества обучения студентов в вузе.....	20
Жуков Д.В., Лукоянычев В.Г. Организация on-line доступа к электронным учебным материалам.....	21
Дунаева В.Е., Ананьев П.И. Разработка программного обеспечения для автоматизации деятельности инженера кафедры по подготовке документации, связанной с учебным процессом.....	21
Белоножка П.А., Ким Н.В., Мулявко П.А., Боровцов Е.Г. Система сбора, обработки и предоставления информации о потребляемых в вузе IT-услугах.....	25
Крайванова В.А., Крючкова Е.Н. Концепция адаптивного управления интеллектуальными объектами на основе логического вывода.....	26
Попов В.В., Кантор С.А. Моделирование возбуждений в неоне.....	28
Старолетов С.М., Крючкова Е.Н. Вопросы проектирования системы тестирования распределенных недетерминированных систем.....	31
Бурындин С.А., Тамплон А.В. Расчет метрик программных систем на основе анализа кода и истории изменений.....	35

Гозман Д.М., Крючкова Е.Н. Кластеризация текстов большой размерности на естественном языке.....	39
Силкин П.А., Лукоянычев В.Г. Программное обеспечение для работы с картографической информацией (на примере обработки маршрутов).....	40
Константинов А.Ю., Крючкова Е.Н. Конструктор нейросетей с возможностью импорта/экспорта нейросетей в базу данных.....	43
Слепова О.В., Ананьев П.И., Кайгородова М.А. Электронный журнал преподавателя как инструмент повышения качества образовательных услуг	45
Рубилина Е.С., Инютин С.М. Разработка системы прогнозирования и анализа свойств химических веществ и их смесей.....	47
Колесникова О.А., Трясак А.Ю., Сучкова Л.И. Разработка программного комплекса по курсу «Теория автоматов»	49
Барбарош А.Е., Кантор С.А. Расчет тепловых процессов	50
Колосовский М.А., Крючкова Е.Н. Применение суффиксных деревьев для эффективной обработки строк	52
Акиншин А.А., Крючкова Е.Н. Функция Гранди в применении к играм на триангуляции Делоне	55
Закурдаев А.В., Крючкова Е.Н. Применение кодирования и сбалансированных деревьев для представления данных сохраняющих отношение порядка	58
Василенко А.Н., Андреева А.Ю. Система управления образовательными ресурсами электронной библиотеки АлтГТУ	60
Мишина В.А., Астахова А.В. Использование Project Expert для финансового моделирования деятельности предприятия.....	61

СИСТЕМА ОЦЕНКИ НАЛИЧИЯ ПЛАГИАТА И КАЧЕСТВА ПРОГРАММ

Карпенко Д.А. – студент, Крючкова Е. Н. – к.ф.м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

В наши дни очень остро встала проблема обнаружения наличия плагиата в исходных текстах программ. Эта проблема относится как к области разработки прикладного ПО, так и к области обучения специалистов в области информатизации. Причём, плагиат присутствует как в небольших лабораторных работах, так и, увы, в серьёзных научных работах.

Мы предлагаем собственную методику оценки наличия плагиата в программных текстах. Также возможна оценка качества программного текста.

Существует масса различных метрик оценки программного текста. К ним относятся метрики размера программ, метрики сложности потока управления, объектно-ориентированные метрики и т. д. Предлагается выполнить оценку сравниваемых программных текстов по нескольким метрикам из различных групп метрик.

Далее, получаем показатель схожести по некоторой метрике:

$$q_{ln} = |q_1 - q_2| / \max(q_1, q_2) \quad (1)$$

где n – номер метрики, q_1 и q_2 – показатели метрики для 1-ой и 2-ой программы соответственно.

Затем получаем общий показатель схожести, как взвешенную сумму k показателей для различных метрик. Если данная сумма выше некоторого порога, то производится с равнение программ интеллектуальным алгоритмом либо делается запрос серверу с помощью механизма RMI на интеллектуальное сравнение с тем кодом из базы готовых решений, с которым выявлено наибольшее сходство.

Интеллектуальное сравнение осуществляется следующим образом.

Во-первых, в программе может быть три типа конструкций: линейный, ветвление и цикл. Характеристикой сложности структуры вычислительной программы является глубина иерархии структуры. Тогда возможно использование следующего подхода: составляется двумерная матрица, первый столбец которой отражает конструкции верхнего уровня каждой функции программы в порядке их следования. Каждая строка таблицы аналогичным образом отражает ветку семантического дерева, уходящую от данной конструкции на максимальную глубину (вложенность). Каждая клетка также имеет указанные характеристики порядка временной сложности и видимых переменных. В таком случае, алгоритм анализа схожести двух программ может быть разработан на базе имеющихся алгоритмов анализа схожести двух изображений.

Во-вторых, две похожие программы имеют похожие данные: переменные, флаги, массивы, строки и т. д. В типичных языках программирования высокого уровня можно приблизительно оценить, какими данными пользовался программист в некоторой конструкции. Один из способов это осуществить – определить все видимые из данного участка переменные некоторого типа путём подъёма вверх по семантическому дереву. При этом, библиотечные типы данных, например `AnsiString` или `ArrayList`, стоит приравнять к базовым. Соответственно, коэффициент схождения, полученный при расчётах для конструкции надо скорректировать при помощи коэффициента:

$$A = \sum_{t=1}^W c |q_1(t) - q_2(t)|, \quad (2)$$

где c – некоторая константы, t – тип данных, q_1, q_2 – количество переменных соответствующего типа в оцениваемой конструкции первой и второй программы соответственно, W – количество различных типов данных в фрагментах 1 и 2.

Качество программы будем оценивать по двум следующим критериям:

- 1) Максимальный порядок временной сложности;

2) Взвешенный показатель по нескольким метрикам.

Естественно, оценка качества будет производиться сравнением показателя, полученного оценкой программы по вышеуказанным критериям, с показателем эталонной программы.

Для того, чтобы реализовать максимальную гибкость и настраиваемость системы, предполагается построить её по следующей схеме: Вся система построена на базе синтаксического анализатора (СА), непосредственно обрабатывающего исходные тексты. Блок синтаксического анализа представляет собой нисходящий алгоритм разбора (LL) с использованием принципа прозрачного синтаксиса и синтаксически-управляемого перевода, с дельта-функциями, формирующими вышеуказанную матрицу. Для настройки анализируемых языков программирования, данный блок должен генерироваться специальной программой – генератором анализаторов.

РАЗРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА С РАСШИРЕННЫМИ ВОЗМОЖНОСТЯМИ

Данилюк Н.П. – студент, Бубнова Н.Д. – ст. преподаватель
Алтайский государственный технический университет (г. Барнаул)

Первые графические редакторы появились еще на рубеже 80 годов и со временем эти проекты переросли в огромные комплексы для решения конкретных задач. Таких как моделирование трехмерных объектов, построение чертежей, создание трехмерных моделей для игр, кино и других областей. Все эти программные продукты предлагает фирма Autodesk, которая является монополистом на рынке программ для работы с графикой. Программные продукты фирмы обладают высоким качеством и позволяют решать очень большой спектр задач. И если учесть, что фирма производитель является монополистом, то цена программных продуктов оказывается высокой.

Разработанная автором программа выполняет аналогичные функции, но при этом является доступной по цене. Она позволяет выполнять следующие задачи:

- визуальная работа с объектами и их составными частями, включая свойства самих объектов (цвет, материал, текстуры);
- полный набор инструментов по работе с трехмерными объектами и возможность стандартных действий как выравнивание, клонирование, построение массивов объектов и т.д.
- поддержка командной строки, с возможностью управления объектами и самим редактором.

Этого набора функций достаточно, чтобы создавать объекты и впоследствии использовать их в качестве трехмерных моделей для других моделирующих программ. Для обеспечения конкурентоспособности программа реализует дополнительные функции:

- динамическая загрузка и отключение библиотек (на ходу создаются и удаляются компоненты на рабочей форме, а также динамически меняются функциональные возможности программы);
- создание объектов из стандартных шаблонов или загрузка их из файлов (в том числе поддержка форматов файлов программ AutoCAD и 3D Max);
- сохранение в графические файлы результатов.

Эти дополнения позволяют расширить возможности программы, и позволяют более эффективно использовать трехмерный графический редактор.

Пользовательский интерфейс программы схож в общих чертах с программой 3Ds Max. В окно выводятся 3 панели с проекциями на каждую из координатных плоскостей и четвертая панель со свободной камерой. Кроме того каждую панель можно настроить по своему желанию и выбрать угол зрения, удобный для пользователя. По краям окна расположены панели с инструментами для управления положением в пространстве и

управления трехмерными моделями.

Внутренняя структура программы модульная. Центральным модулем является ядро системы. Связь между модулями реализована через сообщения. Это позволяет модернизировать программу без серьезного вмешательства в другие компоненты системы.

Перечислим модули и их назначения:

- ядро системы – модуль, который организует обмен данными между модулями посредством сообщений;
- модуль графики OpenGL – модуль, отвечающий за вывод графической информации на экран и за получение информации с экрана, вводимой средствами OpenGL;
- модуль для работы с интерфейсом – предназначен для создания объектов интерфейса библиотек пользователей в рабочем окне и для ввода информации пользователем;
- модуль по работе с объектами – модуль, в котором находятся все данные о трехмерных объектах и функции по их преобразованию;
- модуль по работе с подключаемыми библиотеками – реализует функции по работе с библиотеками и предоставляет пользователю интерфейс для работы с программой.

Модульность гарантирует легкую интеграцию внешних библиотек и связь с между модулями. Пользователю предоставляется набор компонент для разработки библиотек к программе. Следовательно, пользователь может создавать библиотеки на языке высокого уровня C# и подключать их во время выполнения программы.

Достоинством разработанного программного обеспечения является то, что оно предоставляет возможности для:

- построения трехмерных объектов и манипулирования ими. Задачи подобного типа решают программы, такие как 3Ds max, Maya 3D;
- реализации вычислительных алгоритмов и построения графиков, как программы MatLab, Matematica;
- подключения модулей, реализующих новые алгоритмы работы с объектами;
- автоматического изменения интерфейса.

Таким образом, разработанный комплекс предоставляет более широкие возможности и позволяет решать задачи быстрее и экономичнее по сравнению с другими аналогичными программами.

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ОБРАБОТКИ ТЕПЛОВИЗИОННЫХ СНИМКОВ

Кондраханов О.В., Кузьмин А.А. – студенты, Кузьмин А.Г. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

В последнее время наблюдается стремительное развитие новых технологий и средств измерений и контроля (СВЧ-технологии, ультразвуковые приборы, инфракрасные и т.д.). Эти технологии несут в себе совершенно новые возможности, такие как неразрушающий контроль, измерения под нагрузкой, то есть в реальных условиях работы, цифровую обработку результатов измерений и т.д. С 70-х г.г. тепловидение находит всё более широкое применение в медицинской и технической диагностике, навигации, геологической разведке, метеорологии, дефектоскопии, в научно-технических исследованиях тепловых процессов, в военном деле и т.д. В настоящее время в связи с развитием и широким внедрением инфракрасной техники в различные отрасли, задача исследования возможностей ее применения становится актуальной.

Анализ рынка программных продуктов, используемых в тепловидении, показывает, что имеющиеся разработки не реализуют весь набор функций, необходимый лаборатории энергоаудита АлтГТУ для эффективной работы. Цель настоящей работы - создание

программного обеспечения для обработки результатов термографирования поверхности различных объектов, адаптированного для нужд лаборатории энергоаудита АлтГТУ.

Исходя из поставленной задачи, была спроектирована и реализована структура программного комплекса:

1. Слой данных, который содержит описание базовых классов данных, необходимых для работы всей системы;
2. Слой сеанса, который включает в себя функции по обработке термограмм, используя базовые классы, а так же хранит информацию о состоянии и контексте работы;
3. Слой визуализации данных, который содержит методы для наглядного представления информации и текущего состояния работы программы;
4. Слой пользовательского интерфейса, который предоставляет пользователю все необходимые средства для удобной работы.

Программный комплекс включает в себя следующие модули:

1. Базовый модуль для решения общих задач термографии:
 - 1.1. Преобразование тепловизионного снимка в графический формат с возможностью последующего сохранения в графическом формате;
 - 1.2. Выделение различных областей обработки (эллипс, линия, многоугольник произвольной формы) для анализа температурного поля внутри областей обработки;
 - 1.3. Просмотр изображения в любом масштабе с возможностью выделения областей обработки в масштабируемом участке;
 - 1.4. Получение 2D профиля и гистограммы области;
 - 1.5. Формирование различных отчетов в формате MS Excel или MS Word;
2. Модуль для решения специфических задач в области строительства;
3. Модуль для решения специфических задач в области электротехники.

Таким образом, разрабатываемый программный продукт объединяет комплекс необходимых функций, адаптирован к потребностям лаборатории энергоаудита АлтГТУ и в полной мере позволяет решать стоящие перед ней задачи.

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ В УПРАВЛЕНИИ ИНТЕЛЛЕКТУАЛЬНОЙ И ПРОМЫШЛЕННОЙ СОБСТВЕННОСТЬЮ

Ушмодина Е.Г. – студентка, Кайгородова М.А. – к.э.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Прогресс не стоит на месте. Человек постоянно разрабатывает какие – то новые идеи, изобретения, программы, совершенствует уже ранее разработанные, создает художественные и научные произведения. Все это имеет отношение к интеллектуальной собственности (ИС).

Интеллектуальная собственность состоит из двух частей (ветвей) - промышленной собственности и права на произведения художественного творчества. Каждая из этих ветвей ИС регламентируется и защищается своими законами - промышленная собственность охраняется патентным законодательством, а художественная собственность - законом об авторских правах.

В настоящее время в России изобретениям предоставляется правовая охрана в соответствии со ст. 44 Конституции РФ, международными соглашениями, Гражданским кодексом РФ, Патентным законом РФ [1]. Наличие патента дает его обладателю право запрещать практически любое несанкционированное использование его интеллектуальной собственности другими лицами. Право на легальное использование изобретений, полезных моделей и промышленных образцов может быть предоставлено другим лицам лишь при условии выплаты вознаграждения патентообладателю (на основе лицензионного договора

или договора уступки патента). Исключительные права дают патентообладателю возможность не только окупить затраты на создание нового решения, но и получать прибыль на протяжении длительного времени за счет единоличного контроля определенной рыночной ниши.

Для нашего общества все еще непривычны многие понятия и термины в области ИС, начиная от самого словосочетания "интеллектуальная собственность". Понятие "промышленная собственность" довольно часто путается с недвижимостью в виде заводов, фабрик с их зданиями, станками и т.п. , в виде трубопроводов и других транспортных средств, портов и т.д., о чем свидетельствуют публичные выступления руководителей высоких рангов, которым, к сожалению, в молодости не был прочитан курс об интеллектуальной собственности [2].

В справочной литературе под интеллектуальной собственностью понимается совокупность исключительных прав как личного, так и имущественного характера на результаты интеллектуальной собственности и в первую очередь творческой деятельности, а также на некоторые иные приравненные к ним объекты, конкретный перечень которых устанавливается законодательством соответствующей страны с учетом принятых ею международных обязательств.

Понятие промышленной собственности означает исключительное право на владение и распоряжение творческими результатами научно-технического характера. Традиционными объектами промышленной собственности следует считать изобретения и полезные модели как новые технические решения, промышленные образцы как дизайнерские решения внешнего вида промышленных товаров и продуктов [3].

В соответствии с Патентным законом РФ «патент – это установленной формы документ, в соответствующем порядке выданный федеральным органом исполнительной власти по интеллектуальной собственности и удостоверяющий приоритет, авторство изобретения, полезной модели, промышленного образца и исключительное право в отношении этих объектов» [1].

В настоящее время в Алтайском Государственном Техническом Университете существует отдел «Интеллектуальной и промышленной собственности», который осуществляет защиту интеллектуальной собственности патентами на изобретения и полезные модели, а также возможна регистрация программ. Данный отдел создан в целях повышения эффективности информационного обеспечения руководителей, научно-технических работников, профессорско-преподавательского состава, докторов, аспирантов и студентов университета.

Основными задачами отдела являются:

- помощь в оформлении заявок на охранные документы на изобретения.
- правовая охрана объектов интеллектуальной собственности, создаваемых при осуществлении учебной и научной деятельности;
- создание и пополнение фонда патентной и патентно-правовой литературы;
- пропаганда новых идей, научных знаний, развитие изобретательской и инновационной деятельности.

Для учета поданных заявок на изобретения, полезные модели, программы для ЭВМ, а также выданных патентов в отделе интеллектуальной и промышленной собственности использовали программу «Patent». На рисунке 1 изображено главное окно данной программы.

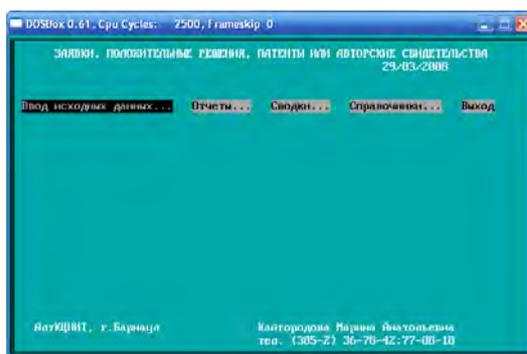


Рисунок 1 – Программа «Patent»

Программный продукт «Patent» был разработан еще в 1998 году на языке Clarion для MS DOS. В то время эта программа была незаменима. Она не только освобождала сотрудников отдела от бумажной работы, но и позволяла легко найти нужную информацию в фонде патентной и патентно-правовой литературы, а также посмотреть кто был автором интересующего изобретения, дату выдачи патента, размер вознаграждения. Не менее важной является информация о дате прекращения патента. В случае, если действие патента истекло и продление патента не оплачено, то эта запись высвечивалась другим цветом. Программный продукт использовался и для составления годовых отчетов, в которых отображалась доля кафедр или факультетов в разработке того или иного изобретения, рац. предложения или программных средств. На рисунке 2 показан пример одного из таких отчетов.

Факультет	Заявки		Пат. решения		Патенты		Прогр. ср-ва	
	Всего	Со студ.	Всего	Со студ.	Всего	Со студ.	Всего	Со студ.
ЛПХ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ЛТ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ДВС	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
СХП	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
СтудПИ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ИТОГО	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Рисунок 2 – Годовой отчет об изобретениях по факультету АТФ

Изучив программу «Patent» можно сделать выводы о её недостатках:

- главным неудобством является то, что она разработана под DOS. Раньше это было незаменимым, а сегодня эту программу не возможно запустить на компьютере с современной ОС;
- база данных разработана на Clarion, а этот язык в основном используется для написания информационно — поисковых систем;
- при редактировании журнала регистрации пользователю предстает не очень удобный интерфейс и чтобы выйти из журнала необходимо, чтобы курсор перешёл на первую запись, т.е. выйти иногда можно, только если несколько раз нажать Esc;
- очень большая вложенность данных, т.е., например, чтобы заполнить поле автор, нужно дождаться, когда откроется три окошка, как видно из приведенного ниже рисунка 3:

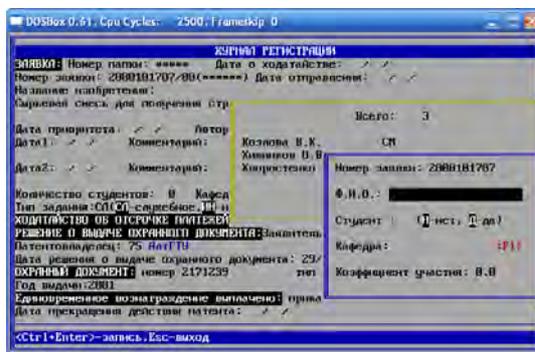


Рисунок 3 – Ввод автора в журнале регистрации

- возникает сложность при формировании отчетов и их распечатке;
- тратится очень много времени на загрузку данных;
- данная база не предназначена для работы в сети.

Для повышения качества работы с программой необходимо разработать программный продукт под Windows, исправить недостатки и реализовать новые возможности, разработать версию, обладающую способностью работать в сети и с возможностью дальнейшего интегрирования разрабатываемого программного продукта с единой автоматизированной информационной системой вуза.

Список литературы

1. Патентный закон Российской Федерации от 23 сентября 1992 г. № 3517-1 с изменениями и дополнениями, внесенными Федеральным законом от 07 февраля 2003 г. № 22-ФЗ, введенным в действие с 11.03.2003.
2. Сергеев А.П. Право интеллектуальной собственности в Российской Федерации: Учебник. – М., 2004
3. Корчагин А.Д. Интеллектуальная собственность: словарь-справочник. – М.: ИНФРА-М, 2005

ИСПОЛЬЗОВАНИЕ ПОЛНОТЕКСТОВОГО ПОИСКА ДЛЯ НАХОЖДЕНИЯ ДОКУМЕНТОВ

Ушмодин Н.А. – студент, Шальнев А.А. – ст. преподаватель
Алтайский государственный технический университет (г. Барнаул)

Компьютеризация расширяется с каждым днем, так как хранение информации в цифровом виде становится удобней и дешевле. Большинство людей, при работе с компьютером использует его для хранения и обработки документов. Количество типов и форматов документов растет с каждым годом. В результате персональные компьютеры заполняются файлами и их пользователи не в силах запомнить, что у них хранится и как его найти. Поэтому и было предложено создать систему, способную помочь пользователям с легкостью находить интересующую их информацию во множестве документов.

Встроенные в большинство операционных систем поисковое ПО позволяет выполнить поиск текстового файла по одному или нескольким критериям: по имени, размеру и дате последнего изменения файла, слову или фразе в документе и другим параметрам. Но чаще всего мы не помним ни того, как называется файл, ни где он находится, ни когда последний раз к нему обращались, - вспоминаем лишь несколько слов, содержащихся в искомом документе. Поиск по этим ключевым словам, конечно, выполняется, но если на компьютере десятки и даже сотни тысяч документов, то выполнение этого поиска может растянуться на несколько часов. Для решения подобных проблем существует индексированный поиск, который заключается в использовании фильтров для извлечения из документов текста и свойств. Этот процесс и называется индексированием. Эти сведения могут включать текст документа, характеристики и параметры (свойства) документа, такие как имя автора [1].

Индекс — это мгновенный слепок информации, полученной из первоисточника.

Важными характеристиками индексного поиска являются размер индекса и скорость индексации. От скорости индексации зависит скорость повторного прохода по уже проиндексированным ресурсам. Для локального поиска важность характеристик меняется местами: размер индекса находится на первом месте, а скорость индексирования менее важна, поскольку индексирование и переиндексирование осуществляются достаточно редко, и их завершения можно подождать. В то же время файл индекса может занимать достаточно большой объем, который пригодился бы и для других целей.

Когда локальная поисковая система составляет индексный файл, она должна уметь получать информацию о содержимом документов самых разных форматов, например текстовые документы, мультимедийные файлы, архивы почтовых сообщений и т.п. В случае

мультимедийной информации индексирования усложняется, поскольку извлечение данных становится трудоемкой задачей и иногда невозможно. Чем больше форматов файлов «понимает» локальная поисковая система, тем она ценнее для пользователя. Локальная поисковая система при составлении индекса может учитывать тип документа, который используется для составления индекса. Так, в формате MP3 есть специальные поля, в которых принято записывать автора композиции, стиль и другие характеристики. Локальный поисковик может применять эти дополнительные сведения для более точного поиска [2].

Еще одним требованием к локальной поисковой машине является полнота, то есть учета всех соответствующих запросу документов, хранящихся на локальных дисках компьютера или на проиндексированных внешних носителях. Это требование существенно, ведь для пользователя может быть принципиально важным наличие или отсутствие искомого документа. Технологии индексации позволяют производить поиск даже по отключенным дискам, но возникает проблема размера индекса, который должен всегда быть доступен для считывания, то есть храниться на локальном диске компьютера.

Доступные на сегодняшний день системы полнотекстового поиска либо являются коммерческими, либо не обладают необходимым функционалом. Многие из имеющихся бесплатных программ не способны, например, осуществлять индексирование внешних носителей.

Поставленную задачу можно решить целиком полагаясь на свои силы. То есть необходимо рассмотреть уже существующие технологии индексирования, выбрать наиболее подходящую и реализовать её на одном из языков программирования. Так же придется решать где будет храниться индексы, для этого можно использовать различные СУБД, можно как хранилище использовать файловую систему. Немаловажно разработать эффективную подсистему выборки информации из индексов, чтобы организовать быстрый и гибкий поиск. После этого нужно приступить к разработке пользовательского интерфейса. Интерфейс должен быть простой в использовании, так как он нацелен на рядового пользователя. Однако функциональность системы от этого страдать не должна.

После ознакомления с поставленными задачами и изучения предметной области, был составлен план решения их решения, который способствует успешной разработке программного продукта в приемлемые сроки.

Список литературы

1. Сергеева Н. Служба индексирования Microsoft // Онлайн-журнал, 2005. http://softkey.info/reviews/review1216.php&referer1=softkey_info&compid=1
2. Apache Lucene – Overview. // <http://lucene.apache.org/java/docs>

ИСПОЛЬЗОВАНИЕ WEB-ТЕХНОЛОГИЙ В УПРАВЛЕНИИ КАЧЕСТВОМ ОБРАЗОВАТЕЛЬНЫХ УСЛУГ

Сартакова Е.А. – студентка, Кайгородова М.А. – к.э.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Система образования относится к числу важнейших социальных институтов. В современном мире значение образования как важнейшего фактора формирования нового качества экономики и общества увеличивается вместе с ростом влияния человеческого капитала [1].

В качестве ключевых позиций при построении современной образовательной системы можно выделить открытость, качество, мобильность и интерактивность. Соблюдение данных принципов невозможно без применения различных инновационных подходов, особенно в области IT-технологий.

В связи с этим возрастает актуальность использования различных автоматизированных информационных систем (АИС) и сетевых технологий при проектировании систем управления и контроля учебной деятельности высших учебных заведений и других образовательных учреждений.

На сегодняшний день отсутствует методология создания таких систем в российском образовании как на федеральном уровне, так и на уровне большинства высших учебных заведений. В настоящее время вузы должны разрабатывать ее самостоятельно, что сопряжено со значительными финансовыми и временными затратами. Данные проблемы актуальны и для Алтайского Государственного Технического Университета им И.И. Ползунова. И одной из приоритетных задач в развитии АИС управления качеством образовательных услуг являются проектирование приложений, использующих web-технологии.

С появлением локальных сетей, подключением таких сетей к Интернет, созданием внутрикорпоративных, сетей, появляется возможность с любого рабочего места учреждения получить доступ к информационному ресурсу сети. Однако, при попытке использовать существующие базы данных возникают проблемы, связанные с требованием к однородности рабочих мест, сильнейшим трафиком в сети, загрузкой файлового сервера и невозможностью удаленной работы. Решением проблемы могло бы стать использование унифицированного интерфейса World Wide Web (WWW) для доступа к ресурсам университета.

Из этих предпосылок возникает задача преобразования накопленных данных в гипертекстовые документы WWW, задача поддержки актуальности преобразованной структуры. Другими словами, задача предоставления WWW - доступа к существующим базам данных [2].

При обеспечении удаленного доступа к существующим базам данных, возможен ряд путей - комплексов технологических и организационных решений. Практика использования WWW-технологии для доступа к существующим БД предоставляет широкий спектр технологических решений, по разному связанных между собой - перекрывающихся, взаимодействующих и т.д. Выбор конкретных решений при обеспечении доступа зависит от специфики конкретной СУБД и от ряда других факторов, как то: наличие специалистов, способных с минимальными издержками освоить определенную ветвь технологических решений, существование других БД и т.д.

Удаленный доступ к существующим базам данных может осуществляться по одному из трех путей.

- Однократное или периодическое преобразование содержимого БД в статические документы;
- Динамическое создание гипертекстовых документов на основе содержимого БД;
- Создание информационного хранилища на основе высокопроизводительной СУБД с языком запросов SQL. Периодическая загрузка данных в хранилище из основных СУБД.

Трудоемкость обеспечения удаленного доступа к базам данных зависит от выбранной технологии. Оценим трудоемкость каждого из предложенных путей:

В первом случае потребуется последовательное преобразование всех данных, находящихся в исходной БД. Разработка средств вывода содержимого таблицы в формате HTML с необходимым форматированием и текстовым сопровождением будет занимать порядка 1-3-х дней для одного разработчика. Разработка средств построения индексной структуры к выводимым данным является более творческой работой и может занять 1-3 недели для одного разработчика [2].

Трудоемкость построения интерфейсов для второго и третьего вариантов в общем случае, эквивалентна трудоемкости построения этих интерфейсов при создании исходной информационной системы с использованием традиционных средств разработки (не CASE-средств). В последней технологии дополнительные трудозатраты пойдут на перегрузку данных в Информационном Хранилище.

Но, несмотря на кажущуюся громоздкость третьего варианта, для задач обеспечения удаленного доступа к содержимому нескольких баз данных или БД сложной структуры накладные расходы существенно уменьшаются.

Из вышесказанного можно сделать вывод, что при выборе вариантов реализации необходимо учитывать цели и задачи проектируемой системы и отталкиваться от структуры хранимых данных и трудовых ресурсов, подключенных к разработке продукта. Рассмотрим проектируемую систему с этой точки зрения.

Целью разработки ПО «Просмотр семестровых рейтингов студентов и результатов промежуточных аттестаций» является воплощение некоторых принципов Модульно-рейтинговой системы квалиметрии учебной деятельности студентов (МРСК), внедряемой в АлтГТУ с 2005/06 учебного года.

МРСК, внедренная в АлтГТУ, является приоритетно важным элементом системы менеджмента качества образования в университете. Квалиметрия учебной деятельности студентов - это мониторинг и измерение их знаний, умений, навыков с целью определения соответствия требованиям ГОС ВПО и запросам потребителей [3].

Главными целями МРСК являются повышение качества подготовки выпускников университета, реализация системного подхода в проведении менеджмента образовательной деятельности на основе комплексных рейтинговых оценок качества учебной работы студентов при освоении ими программ высшего профессионального образования.

Рассмотрим особенности построения АИС для ВУЗа, учитывающей основные принципы МРСК:

В настоящее время используется два подхода при проектировании информационных систем: функциональный и объектно-ориентированный [3, 4].

Функциональный подход является наиболее приемлемым для выполнения регламентированных задач. Кроме того, функциональное моделирование следует применять в тех случаях, когда организационная структура находится в процессе изменения или вообще слабо оформлена.

Эти факторы обусловили использование функционального подхода при проектировании АИС управления качеством образовательных услуг в АлтГТУ.

АИС управления качеством образовательных услуг АлтГТУ является интегрированной системой, в основе которой лежит единая база данных, осуществляющая связь между подсистемами «Абитуриент», «МРСК», «АСТ-Тест» и «Контингент» [3]. Так как проектируемый программный продукт предполагает интегрирование с разработанной системой, использующей технологию баз данных, при выборе технологии размещения данных, необходимо учитывать уже существующую структуру. Поэтому при разработке продукта была использована ранее спроектированная структура базы данных.

Анализ структуры и области применения проектируемой системы позволил прийти к следующему выводу: для данной системы наиболее эффективным будет использование технологии Информационного Хранилища. Это связано с тем, что после установления синхронизации данных информационного хранилища с основными БД возможен перенос пользовательских интерфейсов на информационное хранилище. Это существенно повысит надежность и производительность, позволит организовать распределенные рабочие места.

Таким образом, программное обеспечение для отображения семестровых рейтингов студентов и результатов промежуточных аттестаций позволит удаленно (с использованием сети Internet) в любой момент времени получить следующие виды рейтинговых оценок: входной рейтинг; рейтинг по данной дисциплине, изучаемой в текущем семестре; обобщенный рейтинг, учитывающий входной рейтинг и успехи за всё время с начала обучения в вузе до момента аттестации; выходной рейтинг, определяемый с учетом результатов государственного экзамена и защиты дипломного проекта (дипломной работы). Подобная возможность является удачным воплощением идеи МРСК об открытости результатов обучения для студентов.

В заключение, рассмотрим преимущества, которые будет давать проектируемая система,

в случае ее внедрения. Данный автоматизированный комплекс позволит:

- организовать эффективную работу сотрудников Учебного отдела и профессорско-преподавательского состава АлтГТУ;
- обеспечить открытость информации об индивидуальных рейтингах и результатах промежуточной аттестации для студентов;
- выполнять расчет текущих рейтингов по данным рубежных испытаний в период проведения плановых аттестаций студентов
- анализировать успеваемость студентов;
- снизить временные затраты на обработку данных, необходимых Учебному отделу и профессорско-преподавательскому составу ВУЗа;
- отслеживать изменения данных во времени.

Все это позволит повысить качество образовательных услуг, предоставляемых данным ВУЗом и станет еще одним шагом на пути создания единой информационной среды для всех звеньев территориально разветвленной структуры университета.

Список литературы

1. Яковлев А.М. Обретая уверенность: роль знаний в постиндустриальном обществе [Текст]/ М.А. Яковлев // Вест. Московской высшей школы бизнеса. – 2005. – № 2. – С. 13–18.
2. Нужин С., Родионов М. Фаддеенков Е. Использование технологий WWW для доступа к базам данных [Электронный ресурс] / С. Нужин, М. Родионов, Е. Фаддеенков // интернет-издание ЦНИТ НГУ <http://www.otd.tstu.ru>, 2003. -50 с.
3. Вендров А.М. Проектирование программного обеспечения экономических информационных систем.-М.:Финансы и статистика,2002.-352 с.
4. Устинова Г.М. Информационные системы менеджмента: Основные аналитические технологии в поддержке принятия решений.-СПб:Издательство “ДиаСофтЮП”, 2000.-368 с.

КОМПЬЮТЕРНАЯ СИСТЕМА ТЕСТИРОВАНИЯ ЗНАНИЙ КАК ЭЛЕМЕНТ ОБРАТНОЙ СВЯЗИ СИСТЕМЫ УПРАВЛЕНИЯ УЧЕБНЫМ ПРОЦЕССОМ ВУЗА

Свиридова Н.С. – студентка, Тушев А.Н. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Высшее учебное заведение – это одно из важных социально-экономических систем, так как оно создает в интересах всего общества такую ценность, как интеллектуальный ресурс, который инвестируется во все отрасли народного хозяйства. Современная высшая школа диктует необходимость пересмотра основных принципов управления системой высшего образования.

Учебный процесс в вузе представляет собой сложный объект управления, включающий сложное взаимодействие таких элементов, как целей обучения, содержания дисциплины, методов, организации, форм и средств обучения, и самое главное, контроля результатов обучения и коррекции.

Знания студентов являются выходом системы управления учебным процессом вуза. Уровень достигнутых знаний характеризуется оценкой, полученной студентом. Оценка – это информация, которая по каналу обратной связи поступает в управляющую систему, характеризует текущее состояние объекта управления и дает возможность вносить коррективы в учебный процесс. Студенты получают оценки на экзаменах, зачетах, а также проходя **тестирование**, которое является более **объективным** способом определения уровня знаний.

Компьютерная система тестирования – это информационная система, предназначенная для проверки знаний в рамках учебного процесса. Ее состав может быть определен, исходя из структуры любой информационной системы, например, включать следующие модули: система проведения тестирования, модуль аутентификации, база участников, протокол тестирования, модуль проверки, статистический анализ, шкала оценок, база тестов, система редактирования, система администрирования.

При анализе результатов компьютерного тестирования возможно определение доли правильных ответов, которые студенты дали на один и тот же вопрос или группу вопросов, относящихся к одной теме. Если она будет составлять 50% и менее, то преподаватель может изменить сложившуюся ситуацию: пройти непонятый вопрос еще раз, изменить форму подачи материала, устранить ошибки в методике преподавания и т. д. Это является особенно существенным для промежуточного контроля по каждой пройденной теме, когда у преподавателя есть возможность и время внести коррективы в учебный процесс, а у студента – возможность оценить уровень своей подготовленности.

Таким образом, компьютерное тестирование выступает как **инструмент управления учебным процессом, как элемент обратной связи**, который дает возможность анализировать учебный процесс, вносить в него коррективы, т. е. осуществлять полноценное управление процессом обучения. Следовательно, постоянное использование компьютерных тестов в качестве промежуточного контроля успеваемости определяет учебный процесс как систему непрерывного контроля и самоконтроля студентов, которая вместе с модульным подходом построения учебного курса дает возможность повысить качество занятий в целом, преподаватель получает «обратную связь», а студенты — возможность в течение семестра отслеживать уровень своей подготовленности.

Разрабатываемая программа, реализуемая в данной работе, является тестовой оболочкой базы данных и имеет многоцелевое назначение. Базу данных, используемую в программе, можно видоизменять и настраивать путем введения новых вопросов и задач, что дает возможность организовать обратную связь в системе управления учебным процессом.

АНАЛИТИЧЕСКИЙ ОБЗОР МЕТОДОВ ОБРАБОТКИ ДАННЫХ ЦВЕТОВОЙ ТЕМПЕРАТУРЫ ОБЪЕКТА

Зрюмов П.А. – студент, Зрюмов Е.А. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Недавнее принятие Россией условий Киотского протокола требует в ближайшее время от собственников производств дополнительных мероприятий по снижению выбросов вредных веществ в атмосферу. Также неизбежно приближается момент вступления России во Всемирную Торговую Организацию, которая выдвигает жесткие требования к оборудованию и производству в целом. Горелки должны обеспечивать выполнение всех действующих международных требований по эмиссиям при сжигании жидкого топлива ЕL и природного газа. Таким образом, контроль и управление процессом горения принимает все большую актуальность.

В настоящее время определение цветовой температуры объекта напрямую не производится. Однако существуют и широко применяются сигнализаторы наличия пламени, по ИК-излучению или ионизационному току, но данные приборы не дают количественной оценки качеству расходования топлива (эффективность сжигания).

В данный момент назрела необходимость с помощью систем искусственного интеллекта в реальном режиме времени определять цветовую температуру объекта, что позволит оценить качество расходования топлива.

Целью работы является разработка программного обеспечения для сбора и обработки данных о цветовой температуре объекта. Задачами работы являются: аналитический обзор существующих алгоритмов и методов построения систем сбора и обработки данных о цветовой

температуре объекта, обзор существующих методов обработки цветных изображений, захват цветного видеоизображения с видеоустройства, реализация обработки цветных изображений.

Одним из методов обработки цветных изображений является сегментация. Под сегментацией вообще понимается процесс разбиения изображения на отдельные области. Предположим, что нашей задачей является сегментация объектов на RGB изображении, цвет которых лежит в некотором определенном диапазоне. Для некоторой выборки векторов в цветовом пространстве, репрезентативной по отношению к интересующим нас цветам, мы получаем оценку «среднего» цвета, подлежащего выделению. Пусть вектор a RGB пространства обозначает этот средний цвет. Задача сегментации заключается в том, чтобы классифицировать каждый пиксель данного изображения в соответствии с тем, попадает ли его цвет в заданный диапазон или нет.[1]

Методы распознавания образов делятся на две основные категории: методы, основанные на теории решений, и структурные методы. Первая категория имеет дело с образами, описанными с помощью количественных дескрипторов, таких как длина, площадь, текстура. Вторая категория методов ориентирована на образы, для описания которых лучше подходят качественные дескрипторы. В распознавании образов центральную роль играет принцип «обучения» на выборке известных образов. Для распознавания образов эффективнее всего использовать нейронные сети. [2]

Из приведенного аналитического обзора можно сделать вывод, что применение систем искусственного интеллекта для обработки цветных изображений оправдано и является актуальной задачей при исследовании цветовой температуры пламени сжигания углеводорода.

Список литературы

1. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс // Москва: Изд-во Техносфера, 2005
2. Круг П.Г. Нейронные сети и нейрокомпьютеры / П.Г. Круг // Москва: Изд-во МЭИ, 2002

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ КОМПЛЕКСА ЛАБОРАТОРНЫХ РАБОТ ПО ЭЛЕКТРОТЕХНИКЕ

Незамаев А.А. – студент, В.Г. Лукоянычев – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

В во всех технических ВУЗах, как и в АлтГТУ, имеется ряд специальностей, на которых изучается предмет «Электротехника». В частности, эта, дисциплина, изучается студентами специальностей ПОВТ, ПОИ, КЗОИ. При выполнении лабораторных работ требуется вручную выполнять чертежи электрических схем, производить расчеты токов в электрических цепях. Эта работа занимает довольно много времени. К тому же в случае обнаружения ошибки на каком либо этапе – и на этапе построения схемы, и на этапе составления уравнений для расчетов, и на этапе решения полученной системы уравнений – студенту приходится заново выполнять все, начиная с места допущения ошибки.

Специально для облегчения выполнения лабораторных работ в 1997 году студентом специальности ИВТ была создана программа. Программа работала под управлением операционной системы MS DOS, удовлетворяла требованиям того времени. Она позволяла рисовать несложные электрические схемы, проверять их корректность, проводить расчет электрических цепей по законам Кирхгофа и по методу контурных токов.

Но с тех пор ОС MS DOS морально устарела, и используется только на старых маломощных машинах. В связи с увеличением вычислительной мощности компьютеров и существенным понижением их стоимости, стали активно внедряться операционные системы типа Windows 9x/2000/XP. Хотя в Windows и существует поддержка приложений MS DOS, но существует ряд ограничений, из-за которых ряд приложений будет либо вообще не работать, либо работать с ошибками. Это происходит из-за того, что DOS-приложения могут

использовать прерывания DOS, прямой доступ к диску, что блокируется Windows. Также приложения DOS, работающие в графическом режиме не могут отображаться в окне Windows.

В связи с выше сказанным, потребовалось написание аналогичной программы, работающей под управлением Windows, имеющей также ряд существенных доработок и усовершенствований. Целью данной работы являлось написание таковой программы.

Созданная программа позволяет в достаточной степени автоматизировать выполнение лабораторной работы студентом.

Программа позволяет решать следующие задачи:

- построение электрических схем при помощи собственного редактора схем, в соответствии с выдаваемыми заданиями;
- проверка корректности построенных схем;
- проведение синтаксического анализа элементов схемы;
- проведение расчета проводимости (величина, обратная сопротивлению);
- ввод и решение систем линейных алгебраических уравнений;
- проведение расчета токов в ветвях получаемых схем следующими методами
 - по законам Кирхгофа,
 - по методу контурных токов,
 - по методу узловых потенциалов,
 - по методу эквивалентного генератора;
- сохранение все полученные результаты в файлы на диске,
- формирование отчеты по проделанной работе.

Кроме того, программа имеет удобный пользовательский интерфейс, систему помощи.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ОПТИМИЗАЦИИ КОНСТРУКТИВНЫХ РАЗМЕРОВ ДЕТАЛИ НА ОСНОВЕ КРИТЕРИЕВ НАПРЯЖЕННО-ДЕФОРМИРОВАННОГО СОСТОЯНИЯ

Аксёненко Д.М. – студент, Корнева Г.А. – ст. преподаватель
Алтайский государственный технический университет (г. Барнаул)

В современном мире прогресс не возможен без конструирования новых устройств и деталей. И одним из основных показателей, который отвечает за успешное решение поставленной задачи, является время. Введение новых стандартов и требований к конструкциям (эргономика, современность материалов и т. д.), обуславливают необходимость повышения оперативности при разработке. Практически невозможно использовать традиционный подход к разработке конструкций (последовательные этапы проектирования, изготовления опытных образцов и их длительных стендовых и эксплуатационных испытаний).

Интенсивность и оперативность процесса разработки новых конструкций возможна только при использовании современных технологий основанных на компьютерном моделировании поведения конструкции при различных нагрузках (для анализа сложных конструкций не существуют точных аналитических методов, необходимо использование численных методов расчета). По результатам которого можно выявить проблемные зоны или неудачные конструкторские решения, и оптимизировать конструкцию на основе полученной информации без необходимости изготовления дорогостоящих опытных образцов. Конечно, на финальной стадии их создание необходимо, но при компьютерном моделировании можно заведомо определить неудачные решения.

Все выше перечисленное объясняет необходимость программного обеспечения для оптимизации конструктивных размеров детали на основе критериев напряженно-деформированного состояния.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ОБРАБОТКИ КАРДИОСИГНАЛА В ХОЛТЕРОВСКИХ МОНИТОРАХ НА БАЗЕ МОБИЛЬНЫХ УСТРОЙСТВ

Калашник Е.С. – студентка, Тушев А.Н. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Анализируя мировой опыт в области компьютеризация медицинского оборудования, можно с уверенностью ответить, что это необходимость. Причем это именно та необходимость, от которой во многом будет зависеть эффективность медицинского обслуживания населения, в конечном результате — его качество и продолжительность жизни, а также и экономические аспекты деятельности медицинских учреждений, которыми занимается Всемирная организация здравоохранения.

Сердечно-сосудистые болезни становятся причиной смерти все большего числа людей в мире, поражая в равной степени и бедных, и богатых.

ССЗ являются основной причиной смерти во всем мире: ни по какой другой причине ежегодно не умирает столько людей, сколько от ССЗ;

Помимо прочих способов профилактики сердечно-сосудистых заболеваний я предлагаю прикрепить 1 или более датчиков, считывающих кардиосигнал, который в свою очередь записывать на специальное НID-устройство, там оцифровывается, а затем по USB передается на КПК с установленным специальным программным обеспечением.

Сейчас практически все устройства подключаются к компьютеру через USB-интерфейс. Поэтому во многих новых ПК COM-порт отсутствует вообще.

USB-интерфейс — типовое решение по сопряжению нового внешнего устройства с компьютером, точнее, это НID-интерфейс, базирующийся на протоколе USB 1.1.

Хотя многие и считают, что НID-интерфейс (Human Interface Device) предназначен исключительно для клавиатуры, мыши и джойстика, он годится для множества решений, связанных с сопряжением внешних устройств и компьютера.

Главная задача программно-алгоритмического обеспечения анализа ЭКГ в кардиомониторах — текущий контроль сердечной деятельности человека, т. е. измерение ЧСС и распознавание наиболее выраженных нарушений ритма сердца.

Основная задача, решаемая алгоритмами анализа формы QRS-комплекса, — это отнесение обнаруженных комплексов к одному из двух классов — «норма» и «патология».

Предварительная цифровая фильтрация ЭКС предшествует алгоритмам, осуществляющим анализ сигнала, и служит для выполнения преобразований сигнала, улучшающих условия работы и повышающих эффективность этих алгоритмов.

Алгоритмы распознавания желудочкового комплекса ЭКГ (QRS-комплекса) решают следующие основные задачи: обнаружение комплекса (т. е. установление факта его наличия на анализируемом участке ЭКС) и определение характерных точек комплекса (опорной точки, служащей для измерения RR-интервала, точек начала и конца комплекса, а также крайних точек и вершин его зубцов).

Основная задача алгоритма, основанного на анализе структуры сигнала, состоит в том, чтобы, анализируя эту входную последовательность, распознать появление QRS-комплекса и определить его характерные точки, описывающие отдельные зубцы.

РАЗРАБОТКА ПРОГРАММНОГО ИНСТРУМЕНТАРИЯ ДЛЯ СТАТИСТИЧЕСКИХ ИССЛЕДОВАНИЙ КАЧЕСТВА ОБУЧЕНИЯ СТУДЕНТОВ В ВУЗЕ

Чанов В.В. – студент, Семкин Б.В. – д.ф.-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Современный этап развития общества характеризуется беспрецедентным спросом и широкой диверсификацией высшего образования наряду с все большим осознанием его решающего значения для социально-культурного и экономического развития государства. В этих условиях возрастают требования к уровню подготовки специалистов, инициирующие поиск эффективных моделей организации и управления учебно-познавательной деятельностью студентов в целях повышения их конкурентоспособности и профессиональной компетентности.

В настоящее время в практике работы отечественных вузов широко используются кумулятивные показатели оценки успешности учебно-познавательной деятельности студентов, которые составляют основу рейтинговых систем контроля.

В связи с этим, актуальна проблема мониторинга и анализа показателей успеваемости студентов, их связи с такими данными как: входной балл при поступлении в ВУЗ, вид конкурса, персоналии преподавателей и т. п. Важным представляется также сравнение показателей успеваемости “медалистов”, победителей олимпиад с обычными студентами и так далее. Но такой анализ не будет объективен на единичных студентах или на некоторых группах студентов, нужны огромные массивы данных, чтобы получить репрезентативные результаты.

В АлтГТУ несколько лет назад была создана и продолжает пополняться по сегодняшний день единая база данных, содержащая полную информацию о каждом студенте, его успеваемости в разрезе сессии, отдельных экзаменов, о результатах сдачи вступительных экзаменов. Объем базы настолько велик, что не представляется практически возможным качественно проанализировать эту информацию без использования специальных программных средств.

Поэтому актуально написание программного продукта с достаточно широкими аналитическими возможностями. С его помощью возможно автоматизировать процесс анализа вышеуказанных данных с целью выявления различных закономерностей и получения статистических выводов, важных для отслеживания факторов, влияющих на успеваемость студентов. Например, будет возможна проверка зависимости показателей успеваемости студентов от следующих факторов:

- форма обучения (дневная, вечерняя, заочная);
- является ли студент бюджетником;
- входные баллы, с которыми студент поступил в институт (результаты ЕГЭ);
- закончил ли студент школу с медалью;
- качество преподавания предмета и предъявляемые студентам требования и т. п.

Такой статистический анализ и сопоставление факторов будут способствовать чёткому пониманию ситуации, сложившейся в ВУЗе и его подразделениях, выявлению отклонений от нормального протекания процесса обучения, а также выработке рекомендаций по принятию управленческих решений.

Таким образом, актуальность данной работы обуславливается:

- переходом к новой образовательной парадигме, в основе которой непрерывный рост качества и как следствие совершенствование квалиметрии;
- модернизацией системы высшего образования;
- необходимостью научного осмысления и разработки технологий контроля качества обучения студентов, способствующих обеспечению продуктивности образовательного процесса.

ОРГАНИЗАЦИЯ ON-LINE ДОСТУПА К ЭЛЕКТРОННЫМ УЧЕБНЫМ МАТЕРИАЛАМ

Жуков Д.В. – студент, Лукоянычев В.Г. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

В настоящее время библиотека АлтГТУ располагает большим количеством учебных материалов. Количество экземпляров некоторых из этих материалов очень мало, но они есть в электронном виде.

На сегодняшний день домашние компьютеры, подключенные к интернету, получили широкое распространение.

Для повышения эффективности учебного процесса целесообразно предоставить доступ студентам к имеющимся электронным учебным материалам.

Целью выполненной работы является разработка электронного читального зала предназначенного для решения следующих задач:

- предоставление студентам доступа к учебным материалам, методическим пособиям, диссертациям и т. д.;
- разграничение прав работы с документами (печать, работа с содержимым документа и т.д.).

Наличие электронного читального зала позволит студенту в домашних условиях получить доступ к необходимым материалам, что должно положительно сказаться на уровне его обучения. Данный программный продукт будет очень полезен для дистанционного обучения студентов, т.е. для студентов которые не имеют возможности прийти в библиотеку.

На основе поставленных задач была спроектирована и реализована следующая структура программного продукта:

- модуль авторизации и разграничение доступа пользователей;
- модуль предоставления доступа к материалам;
- классы по работе с документами формата PDF.

Таким образом разработанный программный продукт предоставляет более широкие возможности работы с учебными материалами.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ДЕЯТЕЛЬНОСТИ ИНЖЕНЕРА КАФЕДРЫ ПО ПОДГОТОВКЕ ДОКУМЕНТАЦИИ, СВЯЗАННОЙ С УЧЕБНЫМ ПРОЦЕССОМ

Дунаева В.Е. – студентка, Ананьев П.И. – ст. преподаватель
Алтайский государственный технический университет (г. Барнаул)

В современном мире высшее образование играет важную роль в жизни людей. У человека с высшим образованием есть широкие возможности для самореализации. Но получение высшего образования не возможно без четкой и хорошо организованной структуры управления ВУЗом. Хорошо организованная структура подразумевает строгую иерархию и четкое распределение обязанностей. К тому же организация учебного процесса на отдельной кафедре, отвечающей за определенные специальности, была бы не возможной без такой должности как инженер кафедры.

У инженера кафедры множество различных обязанностей. В связи с принятием новых стандартов и соглашений, увеличивается как количество оформляемых бумаг и обязанностей,

так и объем нагрузки. Не увеличивается только количество отведенных для работы часов. Поэтому в данный момент весьма актуально хотя бы частично автоматизировать и облегчить работу инженера кафедры.

Но так как автоматизировать всю работу в принципе не возможно, рассмотрим только обязанности инженера кафедры по подготовке документов, связанных с учебным процессом.

К документам, связанным с учебным процессом отнесем:

1. различные виды расписания (расписание кафедры, преподавателей, потока, аудиторий);
2. сводные ведомости;
3. выписки к дипломам;
4. ведомости учета выполнения учебной, научной, методической и воспитательной работе на кафедре;
5. учебные поручения преподавателей.

В настоящее время в АлтГТУ используются два программных решения «Бюро расписаний АлтГТУ» и «Контингент».

Программный продукт «Бюро расписаний АлтГТУ» применяют для составления расписаний занятий во всем университете, формировании и печати расписаний потоков и преподавателей. Однако расписание кафедры и аудиторий формирует непосредственно инженер этой кафедры, а это весьма сложный и рутинный процесс.

Программный продукт «Контингент» - программа разработанная для деканата. Деканат каждый семестр вводит оценки студентов и отмечает дисциплины проходимые студентами. В конце обучения студента, для выдачи диплома инженер кафедры берет в деканате семестровые ведомости с оценками студентов и самостоятельно формирует сводную ведомость по группам. Затем сводная ведомость сверяется с зачетными книжками студентов. После чего для каждого студента оформляется выписка к диплому.

По подготовленным заведующим кафедры учебным поручениям преподавателей, а так же по фактам выполнения плана преподавателями, инженер кафедры формирует ведомости учета выполнения учебной, научной, методической и воспитательной работе на кафедре (форма №6) по каждому семестру и итоговый за год, затем передает ведомость о нагрузках преподавателей в бухгалтерию для начисления преподавателям заработной платы.

Все эти выше перечисленные документы формируются и оформляются самостоятельно в MS Word или в MS Excel, без какой-либо автоматизации.

Особенностью программ «Контингент» и «Бюро расписаний АлтГТУ» является то, что они используют одну базу данных на Oracle для хранения данных и использования. Обе программы используют общие данные о:

- факультетах (название, краткое название, статус, декан, должность руководителя)
- специальностях (наименование, краткое наименование, шифр)
- кафедрах (наименование, краткое наименование, кабинет, телефон)
- преподавателях (ФИО, должность, звание, степень)
- группах (название, год поступления)
- дисциплинах (наименование, краткое название, шифр)

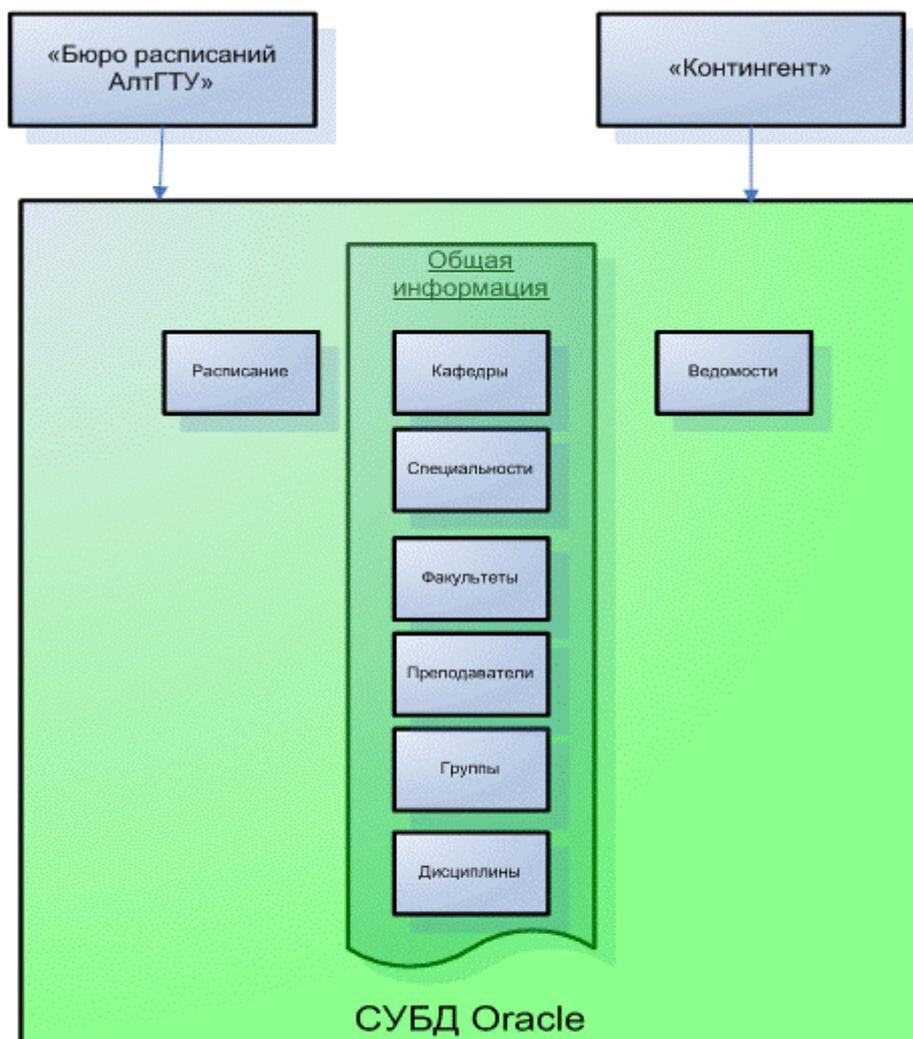


Рис. 1 Взаимодействие программ с БД до внедрения решения

В базе данных АлтГТУ для реализации поставленных задач не хватало только данных о нагрузках преподавателей. Соответствующие таблицы по нагрузкам были добавлены.

В результате была написана программа на С#, использующая БД Oracle, для автоматизации деятельности инженера кафедры, связанной с учебным процессом.

Полученный программный продукт предусматривает разграничение прав доступа для того, чтобы инженер не мог просматривать конфиденциальную информацию по чужим кафедрам (оклады, нагрузки).

Разработанный программный продукт позволяет распечатывать расписание кафедры, расписание отдельного преподавателя кафедры, расписание потока и расписание занятости аудиторий. Для формирования расписаний берутся данные из БД АлтГТУ, введенные с помощью «Бюро расписания». Помимо этого, разработанный программный продукт формирует сводную ведомость по группам выпускников кафедры и выписки к дипломам установленного образца для каждого выпускника, при этом используются данные,

накопленные за все периоды обучения студента из базы данных АлтГТУ, введенные с помощью программы «Контингент».

Для ввода учебных поручений будет использоваться разработанный программный продукт для частичной автоматизации инженера кафедры. Предусмотрен ввод 2 способами:

- 1- считывание информации из xls - файла с использованием COM - технологии;
- 2- ввод непосредственно в программе, с помощью специальной формы.

Из учебных поручений будет формироваться предварительная форма №6, которую можно будет изменить и вывести на печать. Учебные поручения так же можно выводить на печать.

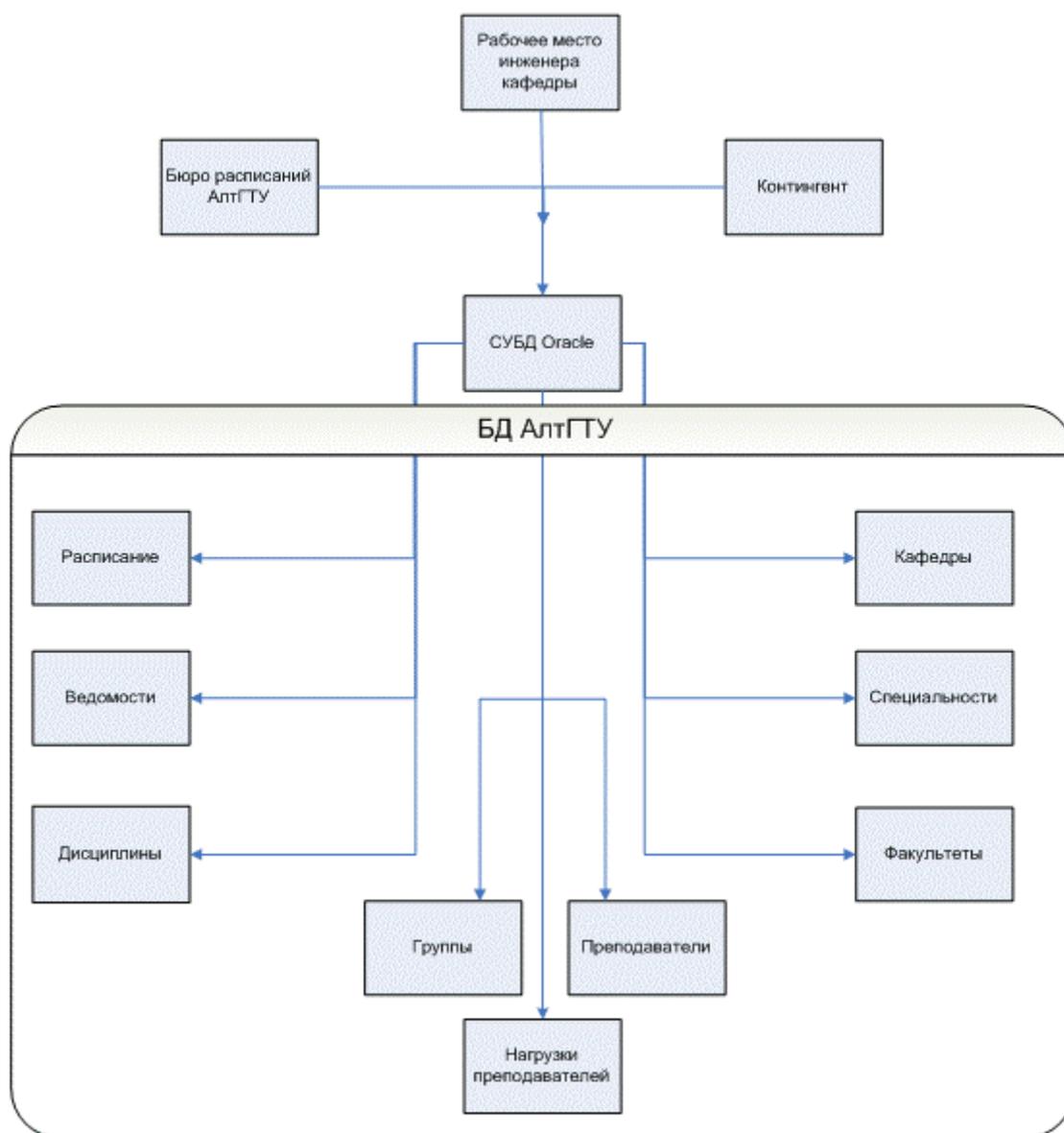


Рис. 2 Взаимодействие программ с БД после внедрения решения

После внедрения программного решения (Рис 2), документы связанные с учебным процессом будут формироваться из дополненной базы данных АлтГТУ, таким образом сократится время затрачиваемое на формирование необходимых документов, уменьшится объем дублируемой информации, увеличится производительность инженеров кафедры, уменьшится вероятность человеческой ошибки.

СИСТЕМА СБОРА, ОБРАБОТКИ И ПРЕДОСТАВЛЕНИЯ ИНФОРМАЦИИ О ПОТРЕБЛЯЕМЫХ В ВУЗЕ ИТ-УСЛУГАХ

Белоножка П.А., Ким Н.В., Мулявко П.А. – студенты,
Боровцов Е.Г. – к.т.н., профессор

Алтайский Государственный Технический Университет (г. Барнаул)

Алтайский Государственный Технический Университет на сегодняшний день является одним из крупнейших ВУЗов Сибири. АлтГТУ представляет собой достаточно сложную структурную организацию, состоящую из множества подразделений и отделов, где каждое подразделение выполняет свою функцию. В связи с бурным развитием инфраструктуры ВУЗа, сейчас уже практически все его отделы оснащены различной офисной техникой и персональными компьютерами.

На начальном этапе формирования технической базы нашего университета, персональные компьютеры использовались для небольшого круга задач, практически не влияя при этом на общую работу отдела. Однако неоспоримые удобства работы с данными в электронном формате привели к компьютеризации ВУЗа и на сегодняшний день в большинстве отделов имеется по несколько рабочих станций. Для достижения еще большего удобства эти компьютеры объединялись в локальные сети, а затем появилась и единая университетская сеть.

Необходимость поддержания корпоративной сети и ее пользователей привело к созданию нового подразделения. До недавнего времени таким подразделением был Алтайский краевой центр новых информационных технологий (АлтКЦНИТ), в настоящий момент этим занимается управление по информатизации.

Основными услугами, которые предоставляют специалисты управления по информатизации, являются обеспечение функционирования корпоративной сети, предоставление доступа в Интернет (построение СКС, подключение к ЛВС), различные разовые услуги (установка и настройка ПО, настройка техники) и так далее.

В рамках подключения к корпоративной сети отделам предоставляются почта в университетском домене, различные телематические услуги, доступ к сети Интернет, а также возможность использовать хостинг в домене `secna.ru` и `altstu.ru`.

Некоторые услуги являются не безвозмездными. Например, услуга предоставления доступа к сети Интернет. Подразделения рассчитываются за объем информации, которая была получена или передана в Интернет – за трафик. Одной из немаловажных задач специалистов технической поддержки является своевременное предоставление актуальных данных по статистике работы подразделений в сети Интернет, а также подробной информации о потребленных услугах.

Задача предоставления данных клиенту решается либо с помощью готового автоматизированного программного комплекса, либо при написании собственного продукта. Анализ рынка готовых решений показывает, что они обладают двумя весомыми недостатками: они не адаптированы для нашей конкретной задачи и не учитывают структуру ВУЗа, кроме того имеют достаточно высокую стоимость. Написание собственного программного продукта в рамках дипломного проекта позволяет решить поставленную задачу, а также избежать подобных недостатков, связанных с использованием коммерческих продуктов.

Программный продукт состоит из трех логических частей:

1. Первая часть обеспечивает сбор и начальную обработку полученной об услугах информации, передавая ее дальше;
2. Вторая часть обрабатывает эти данные, выдает статистическую информацию, создает различные отчеты;
3. Третья часть предоставляет интерфейс взаимодействия с пользователями, визуализацию обработанных данных.

Потребителей нашего ПО можно разделить на несколько категорий: системные администраторы, технический персонал, учетно-расчетной группы, конечные пользователи.

Данный программный продукт позволяет:

- администраторам строить графики загруженности каналов по типам трафика и времени.
- пользователям получать актуальную информацию о потребленных ими it-услугах (количество входящего и исходящего трафика, вызов специалиста).
- техническому персоналу в удобной форме получать служебную информацию о клиентах.
- учетно-расчетной группы в удобной форме получать информацию о наименовании и количестве услуг, потребленных клиентами.

В результате, программный продукт позволяет оперативно получать необходимую информацию, а также сэкономить силы, затраты и время пользователей всех категорий. Пользователи могут регулировать затраты на it-услуги в течение всего периода пользования, а не при выставлении счета в конце месяца; учетно-расчетная группа - своевременно узнавать всю необходимую информацию; системные администраторы – получать возможность производить более тщательную диагностику и настройку сети.

КОНЦЕПЦИЯ АДАПТИВНОГО УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫМИ ОБЪЕКТАМИ НА ОСНОВЕ ЛОГИЧЕСКОГО ВЫВОДА

Крайванова В. А. – аспирант, Крючкова Е. Н. – к.ф.-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Эффективность современных компьютерных систем определяется не только скоростью работы и точностью получаемых результатов, но и способностью сохранять эти характеристики в условиях динамически изменяющейся внешней среды. Кроме того, концепция адаптивных компьютеров предлагает путь сдерживания дальнейшего роста сложности IT-систем [1]. В работе рассмотрена универсальная система адаптивного управления интеллектуальными объектами на основе языкового подхода и логического вывода.

Объект управления получает внешнюю информацию в виде значений взаимосвязанных параметров различной природы: результатов измерений, запросов к базам данных и т. д. Частью входной информации также могут быть управляющие воздействия. В общем случае можно считать, что входная информация представлена в виде цепочек некоторого языка A . На лексиконе языка A целесообразно ввести нечеткие отношения обобщения и синонимичности. Отношение обобщения формирует иерархию понятий языка, а отношение синонимичности связывает лексемы-синонимы. Предполагается, что фразу языка A можно перевести в дерево синтаксической конструкции, в котором узлы помечены лексемами языка, а ребра — типами отношений между ними. Тогда логику интеллектуального объекта естественно задавать в виде набора правил контекстной замены, определенных на синтаксических конструкциях языка A . Правило контекстной замены задается упорядоченной парой $r = \langle h, c \rangle$, где h и c – гипотеза и следствие правила, представленные помеченными деревьями, узлами которых могут являться лексемы языка A и переменные, а ребра помечены типами отношений.

Задача логического вывода – построить список команд для объекта управления из исходных гипотез путем построения цепочек логического вывода. Специфика существующих алгоритмов, решающих эту задачу, в том, что они имеют высокую временную сложность и могут в процессе работы приходиться в состояние комбинаторного взрыва. Эта особенность проявляется на двух уровнях: в момент выбора правила, применяемого на данном шаге, и в

момент унификации.

В большинстве случаев фраза на языке A характеризуется уникальным упорядоченным набором слов – пометок вершин. Схемы построения фраз, задаваемые пометками ребер, определяются не обширным лексиконом, а структурной спецификой выбранного языка, и потому имеют тенденцию к повторению. Шаблон *Template*(ϕ) фразы ϕ - это дерево фразы ϕ без пометок вершин. Для эффективного выбора применяемого правила и минимизации количества операций унификации используется механизм кластеризации правил на основе шаблонов гипотез правил.

Унификация фразы и гипотезы правила представляет собой задачу изоморфизма подграфа для двух ориентированных помеченных деревьев с учетом переменных [2]. Для снижения временной сложности этого этапа нами разработана система отсечений на основе анализа структурных характеристик узлов деревьев: высоты поддерева, мощности ближайшего ветвления и мощности максимального ветвления в поддереве.

Формирование команды для объекта управления происходит через подстановку полученных в процессе вывода деревьев в целевые команды. Целевая команда представляет собой дерево, аналогичное гипотезе правила, то есть кроме узлов-констант дерево содержит узлы-переменные, реализующие параметры команды.

Универсальность и эффективность систем, основанных на логическом выводе, зависит от размера, полноты и избыточности базы знаний системы. В большинстве существующих реализаций такие базы составляются вручную. Мы предлагаем адаптивный алгоритм построения базы знаний на основе уже имеющейся базы на этом же языке. База знаний системы имеет два уровня: уровень лексикона и уровень правил контекстной замены.

Обучение уровня лексикона заключается в изучении системой новых слов и в построении отношений синонимичности и обобщения между ними. Механизм обучения лексикона основывается на анализе места в дереве фразы, на котором встретилось незнакомое системе слово. Если в процессе унификации дерева некоторой фразы с правилом незнакомое слово a подставляется в переменную v , то делается предположение, что слово a обобщается понятиями переменной v . Если незнакомое слово a унифицируется с узлом-константой b , то делается предположение, что слово a является синонимом слова b . Так как эти предположения могут быть ошибочными, отношениям между словами a и b назначается некоторая степень достоверности $\tau_r(a,b)$, где r – тип отношения (обобщение или синонимичность). Если два слова не связаны отношением r , то $\tau_r(a,b)=0$. Достоверное отношение между словами (например, отношение, заданное пользователем) имеет $\tau_r(a,b)=1$. Достоверность отношения учитывается при вычислении степени релевантности получаемого результата.

Обучение уровня базы правил системы заключается в формировании новых правил на основе уже имеющейся базы с учетом поступающей в объект внешней информации. Адаптивное обучение системы происходит на двух этапах: при инициализации системы и непосредственно в процессе логического вывода. Алгоритм ассоциативного обучения вычлняет из пары неформализованных фраз *<гипотеза, следствие>* на естественном языке логическое ядро правила. При определении структуры логического ядра добавляемого правила используются шаблоны уже имеющихся в базе правил. Адаптивный алгоритм дополняется набором эвристик: обобщение пропозициональных узлов в переменные и, как следствие, возможное слияние нескольких правил, обучение и структурирование лексикона системы, устранение вводных конструкций. В процессе интерактивного общения с пользователем система может обучаться с учителем, то есть запрашивать у пользователя уточнение базы знаний.

Предложенная математическая модель находится в стадии программной реализации в виде самообучающегося робота, который по неформальному описанию проблемы строит список возможных действий для её решения. В качестве языка для данной экспериментальной разработке выбран естественный человеческий язык, как наиболее сложный лингвистический феномен, обобщающий такие особенности других языковых

знаковых систем, созданных человечеством, как многозначность, синонимичность, избыточность и т.д. На данный момент базовые алгоритмы логического вывода и обучения системы реализованы на языке Java. Для базы знаний и обучающих выборок модуля инициализирующего обучения используется XML-формат. В качестве модуля синтаксического анализа используются словари и библиотеки автоматической обработки текста Dialing[3]. Модульная структура робота делает основные компоненты обучения и логического вывода независимыми от конкретного языка A и специфики объекта управления, что позволяет говорить об универсальности системы.

Список литературы

1. Вонт Р. Адаптивные и проактивные компьютерные системы / Вонт Р., Перинг Т., Тенненхаус Д. – Открытые системы #10/2003
2. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи - М.: Мир, 1982.
3. Сокирко А.В. Реализация первичного семантического анализа в системе Диалинг / Тр. междунар. семинара «Диалог'2000» по компьютерной лингвистике и её приложениям, - Протвино, 2000

МОДЕЛИРОВАНИЕ ВОЗБУЖДЕНИЙ В НЕОНЕ

Попов В.В. – аспирант, Кантор С.А. – к.ф.-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Целенаправленное создание наноструктурных материалов в режиме "самосборки" требует очень глубокого и достаточно полного представления о взаимодействии атомов и кластеров – строительных элементов создаваемых материалов. По своим физико-химическим свойствам кластеры занимают промежуточное положение между атомами и молекулами с одной стороны и конденсированным веществом – с другой. Эволюция кластеров ведет к образованию либо газовой фазы, либо конденсированной фазы, проходя ряд метастабильных состояний. Они обладают высокой химической активностью, а цепь процессов перехода от одного состояния к другому является сложной и неравновесной. Еще более сложные взаимодействия кластеров происходят в условиях внешних воздействий, с помощью которых можно и управлять этими процессами [1, 2].

Многие физико-химические свойства вещества определяются электронным строением. Знание электронного строения позволяет не только объяснять обнаруженное поведение вещества, но и предсказывать, создавать материалы с заранее заданными свойствами. В исследовании электронной структуры атомов, молекул, твердых тел достигнут значительный успех. Наиболее популярными являются методы расчета электронной структуры вещества в основном состоянии в приближении функционала электронной плотности. Однако больший интерес представляют возбуждения: тепловые, вакансионные, примесные, электромагнитные, в которых пребывают электроны реального вещества.

Для описания возбуждений сначала рассмотрим атом, находящийся в равновесном поле. При этом прочие частицы, окружающие интересующий нас атом, включим в понятие термостата. Такие возбуждения, способные спонтанно распадаться за конечное время τ , и, согласно соотношению неопределённости, обладают конечной шириной $\Gamma \sim \hbar/\tau$ энергетического уровня, характеризующего квазистационарную систему. Волновая функция, описывающая эту систему, должна содержать такой экспоненциальный множитель, чтобы все вероятности, определяющиеся квадратом модуля волновой функции, затухали по закону

$\exp(-\Gamma t/\hbar)$: $\Psi(\mathbf{r},t) = \psi(\mathbf{r}) \exp\{-i(E - i\Gamma/2) t/\hbar\}$. Будем считать при этом, что для описания возбуждений аппроксимация центрального поля оправдана в той же степени, в какой она оправдана для описания незаполненных оболочек основного состояния в атоме с оператором энергии \mathbf{H} . Тогда функцию $\psi(\mathbf{r})$ удобно искать в виде $\psi(\mathbf{r})=R(r)Y(\theta,\varphi)$. Более того, пусть угловая функция $Y(\theta,\varphi)$ удовлетворяет тем же уравнениям: $-\Delta Y=l(l+1)Y$, $-i \partial Y/\partial \varphi=mY$, но регулярна только по φ . Произвольное поведение этой функции на сфере по переменной θ снимает ограничение на целочисленные значения параметра l . Будем считать его, в общем случае, комплексным: $L=l+x+iy$, где l , как и в основном состоянии, пусть пробегает целочисленные значения, тогда область изменения комплексной добавки $x+iy$ может быть ограничена, $|x|<0.5$ и $|y|<0.5$. Таким образом, задача отыскания спектра орбитальных возбуждений атома может быть сведена к проблеме собственных значений стационарного уравнения Шредингера $(\mathbf{H}+\hbar^2/2m^* (u+iv)/r^2) \psi(\mathbf{r}) = \varepsilon\psi(\mathbf{r})$. Здесь $u=x(x+2l+1)-y^2$, $v=y(2x+2l+1)$. Заметим, что при $y \neq 0$ данная задача является существенно неэрмитовой с комплексными значениями энергий $\varepsilon=E-i\Gamma/2$. Перебирая все возможные значения $|x|<0.5$ и $|y|<0.5$ в процессе поиска самосогласованных решений уравнений можно проследить по минимуму полной энергии возбуждённого атома за его спектральными характеристиками.

Эту идею мы обобщили для описания возбуждений в кластерах. Возбуждения описаны в рамках метода функционала электронной плотности с учетом ширины энергетических уровней. В качестве объекта исследования выбраны атомы неона, поскольку интерпретация экспериментальных данных именно для них должна быть простой и однозначной, в отличие, например, от интерпретации экспериментальных данных для атомов щелочных металлов из-за наличия у них сверхтонкой структуры в основном состоянии. Неон считается химически не активным, поэтому часто используется в качестве буферного газа. В гелий-неоновых лазерах он является рабочим веществом, которому резонансным образом передаются возбуждения от гелия.

Задачу на отыскание собственных значений стационарного уравнения Шредингера мы решали численно в базисе функций гауссового типа [3], используя десять функций в разложении по $l=0$, и пятнадцать функций – в разложении по $l=1$. Оценки показали, что базис такой длины вполне пригоден при решении указанных выше уравнений методом Рутана для атома неона, если $|x|<0,1$ и $|y|<0,5$.

Результаты расчета спектральных характеристик показывают, что энергии одноэлектронных внешних состояний $2s$ -, $2p$ -симметрии слабо меняются с изменением параметра y . Для состояний $1s$ -симметрии наблюдается значительное поднятие вверх по шкале энергий с увеличением параметра y , трактуемое нами как коллапс атома – явление, интенсивно обсуждаемое в научной литературе на примере атома водорода. Отметим, что полученные нами зависимости при $x=0$, незначительно меняются с изменением параметра x в области $|x|<0,1$.

Сама же полная энергия претерпевает существенно большие изменения с ростом x . Результаты самосогласованных вычислений показывают, что $\text{Re } \varepsilon$ – действительная часть полной энергии возбужденного атома может быть как больше, так и меньше энергии его основного состояния при $x=0$, $y=0$. Оказывается, существует множество возбужденных состояний атома неона, таких, что $\text{Re } \varepsilon$ – действительная часть полной энергии меньше энергии его основного состояния. Однако большая часть из них – короткоживущие. Наибольший же интерес представляют неограниченно долго живущие состояния, когда мнимая часть полной энергии $\text{Im } \varepsilon$ обращается в нуль. Заметим, что модуль этой величины есть вероятность распада возбуждения в единицу времени. К неограниченно долго живущим состояниям с $\text{Im } \varepsilon=0$, кроме основного состояния при $x=0$ и $y=0$, относятся возбуждения со значениями параметра y из интервала $0,3 < y < 0,5$. Однако в изолированном атоме создать

возбуждения столь больших интенсивностей, соответствующие значениям $y > 0,3$ скорее всего нереально. Практически используемые поля предполагают изменение параметра y до тысячных, в крайнем случае, до сотых долей единицы.

Тем не менее, получить стабильное возбуждение можно, если, например, поместить такой атом в поле, созданное другим атомом. Для проверки этого предположения мы вычислили полную энергию двух атомов неона как функцию расстояния между ними для пяти небольших значений параметра $y=0, \pm 4 \cdot 10^{-6}, \pm 8 \cdot 10^{-6}$ и $x=0$. Результаты расчетов показывают, что энергия основного состояния системы двух атомов при любых расстояниях между ними больше энергии основного состояния двух атомов, изолированных друг от друга. Это означает, что система Ne_2 в основном состоянии является неустойчивой. Атомам неона энергетически более выгодно существовать порознь. Нами обнаружено, что $Re \epsilon$ – действительная часть полной энергии системы Ne_2 в возбужденном состоянии падает при сближении атомов. Она падает и с ростом y – при увеличении интенсивности внешнего воздействия. Скачкообразное поведение $Re \epsilon$ при $y \neq 0$ мы связываем с перестройкой электронной структуры электронов p -симметрии. Отметим, что в кластерах, состоящих из атомов, не содержащих электроны p -симметрии, таких как He, Li и Be , $Re \epsilon$ меняется плавно с изменением расстояний между атомами в том числе и при $y \neq 0$.

Нами обнаружено, что в небольшом интервале расстояний около 2 и 6 боровских радиусов $Im \epsilon = 0$, что указывает на неограниченно большие времена жизни Ne_2 . Интервал межатомных расстояний, включающий в себя значение 6 боровских радиусов, вполне соответствует экспериментально наблюдаемому. Существование долгоживущего возбуждения Ne_2 с расстояниями между атомами порядка 2 боровских радиусов, реализуемое при очень больших значениях $Re \epsilon$, в литературных источниках нами не обнаружено.

Полученные данные позволяют предполагать, что если интересующий нас атом поместить в поле двух, трех, или большего числа атомов, то может образоваться кластер, стабильный по энергии и с неограниченно большим временем жизни при примерно таких расстояниях между атомами, что и в Ne_2 . Для проверки этого предположения мы провели самосогласованные вычисления энергетических характеристик электронов в Ne_3 при тех же наборах параметров x и y . Результаты расчета $Re \epsilon$ – действительной части полной энергии Ne_3 в зависимости от расстояния между атомами качественно не отличаются от соответствующей зависимости в системе Ne_2 . Однако, мнимая часть полной энергии $Im \epsilon$ системы Ne_3 обращается в нуль при чуть меньших расстояниях между атомами, чем в системе Ne_2 . Да и сам интервал расстояний, при которых $Im \epsilon = 0$, включающий в себя значение 6 боровских радиусов, заметно сократился, однако интервал расстояний, включающий в себя значение 2 боровских радиусов, увеличился.

Для проверки обнаруженных закономерностей мы вычислили полную энергию четырех атомов неона как функцию расстояния между этими атомами при тех же значениях параметров $y=0, \pm 4 \cdot 10^{-6}, \pm 8 \cdot 10^{-6}$ и $x=0$. Результаты расчетов показывают, что стал более четко выраженным минимум $Re \epsilon$ при расстоянии 23 боровских радиуса, когда $y = \pm 4 \cdot 10^{-6}$. Более глубоким стал минимум $Re \epsilon$ при расстоянии порядка 8 боровских радиусов, когда $y = \pm 8 \cdot 10^{-6}$. Изменение положений минимумов действительной части полной энергии указывает на перестройку электронной структуры в Ne_4 при других расстояниях между атомами, нежели это происходит в Ne_3 или в Ne_2 . Мнимая часть полной энергии $Im \epsilon$ системы Ne_4 , обращается в нуль при еще меньших расстояниях между атомами, чем в системе Ne_3 . И сам интервал расстояний, при которых $Im \epsilon = 0$, включающий в себя значение 6 боровских радиусов, еще заметнее сократился, а интервал расстояний, включающий в себя значение 2 боровских радиуса, увеличился. Таким образом, закономерности изменений полной энергии в целом подтвердились и на системе четырех атомов неона. Использование различного вида

функционалов для обменно-корреляционной энергии [4-6] приводит лишь к незначительным количественным изменениям спектральных характеристик кластеров неона. Еще меньшие изменения обнаружены в результатах расчета как действительной, так и мнимой части полной энергии.

Предложенный метод описания возбуждений в кластерах с учётом ширины энергетических уровней атома оказался полезным для поиска долгоживущих состояний. Показано, что атом неона, помещенный в поле большой интенсивности, сравнимое с интенсивностью поля в самом атоме, может перейти в возбужденное состояние с полной энергией, меньшей энергии основного состояния, и пребывать в таком состоянии неограниченно долго. Обнаружены неограниченно долго живущие возбуждения в двух, трех и четырехатомных кластерах в полях реально достижимых небольших интенсивностей при расстояниях между атомами порядка 6 боровских радиусов и при расстояниях между атомами порядка 2 боровских радиусов, если поля чрезвычайно велики. Оказалось, что использование различного вида функционалов для обменно-корреляционной энергии приводит лишь к незначительным количественным изменениям спектральных характеристик кластеров неона и к еще меньшим изменениям полной энергии.

Список литературы

1. Слабко В. В., Хачатрян Г. Г., Александровский А. С. Управляемая внешним световым полем самоорганизованная агрегация малых металлических частиц // Письма в ЖЭТФ. 2006. Т.84. №6. С.360-364.
2. Балыкин В. И. Атомно-проекционная параллельная фабрикация наноструктур // УФН. 2007. Т.177. №7. С.780-786.
3. Huzinaga S. Gaussian-type functions for polyatomic systems. I // J. Chem. Phys. 1965. V. 42. N4. P.1293-1302.
4. Barth von U., Hedin L. A local exchange-correlation potential for the spin-polarized case // J. Phys. C. 1972. V.5. N13. P.1629-1642.
5. Vosko S. H., Wilk L., Nusair M. Accurate spin-dependent electron liquid correlation energies for local spin density calculation // Canad. J. Phys. 1980. V.58. N8. P.1200-1215.
6. Perdew J. P., Wang L. Accurate and simple analytic representation of the electron-gas correlation energy // Phys. Rev. B. 1992. V.45. N23. P.13244-13249.

ВОПРОСЫ ПРОЕКТИРОВАНИЯ СИСТЕМЫ ТЕСТИРОВАНИЯ РАСПРЕДЕЛЕННЫХ НЕДЕТЕРМИНИРОВАННЫХ СИСТЕМ

Старолетов С.М. – аспирант, Крючкова Е.Н. – к.ф.-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

В настоящее время среди программного обеспечения мы все чаще и чаще сталкиваемся с понятием не отдельной программы, а распределенного комплекса программ. С каждым днём такое ПО становится все сложнее и сложнее, а с ростом сложности идет и возрастание количества ошибок в нем, причем находить и исправлять ошибки в распределенном программном продукте намного труднее. Поэтому производители промышленного ПО заинтересованы в системе, позволяющей выявлять ошибки времени выполнения в процессе разработки распределенных программ.

Рассмотрим основные требования при разработке системы тестирования современных

распределенных приложений.

1. Разработка распределенных приложений ведется обычно на языках Java и C# при помощи популярных средств разработки. Поскольку разработчики привыкли к своим средствам, целесообразно обеспечить максимальную интеграцию системы тестирования с такими средами как Microsoft Visual Studio и Eclipse SDK.

2. Тестирование нераспределенных, “stand-alone” систем заключается в том, что происходит воздействие на тестируемое ПО, анализируется реакция системы на воздействие, сравнивается с тем, что должно получиться в эталоне, и так проделывается, чтобы покрыть максимальное число ситуаций с потенциально возможными ошибками. Тестирование распределенных систем отличается тем, что сначала необходимо создать инфраструктуру, обеспечивающую воздействие на распределенную систему, анализ реакций системы на такие воздействия и проверку правильности работы всей распределенной системы.

3. Для того, чтобы проверять правильность работы системы, необходимо формализовать требования к “неделимым” частям системы. Если бы система была нераспределенной, то такими неделимыми частями системы были бы объекты и их методы. Но поскольку мы рассматриваем распределенные системы, под неделимыми частями системы необходимо считать события (Events), поскольку они возникают асинхронно, и только через возникновение событий можно отследить взаимодействие между частями системы. Формализация предполагает написание предусловий и постусловий для каждого события, возникающего в системе. Согласно пунктам 1 и 2, лучше всего будет добавить в языки C# и Java специальные языковые конструкции, позволяющие описывать пред- и постусловия для событий и реализовать препроцессор для обработки “своих” конструкций.

4. Сам процесс “наблюдения” за системой предполагает создание математической модели тестируемой программы.

В процессе описания модели нас не будут интересовать переменные, поля классов, нас будут интересовать состояния системы и взаимодействие (посылка сообщений от объекта к объекту, от приложения к приложению, от сервера к клиенту и обратно, а также вызов удаленных методов). Состояния системы - это наиболее важные с точки зрения разработчиков точки исходного кода, в которых происходит важное взаимодействие в системе. Состояния системы и множество переходов определяются программистом и описываются на разработанном XML языке описания моделей. Для описания модели мы используем принцип – “код и модель-одно целое”, т.е. описание модели реализуется в комментариях к исходному коду программы(программ), причем для разных языков программирования язык описания модели является единым, и далее описание модели обрабатывается собственным синтаксическим анализатором и игнорируется компилятором среды разработки.

Конечный автомат модели программы тестируемой системы, как обычно, имеет вид пятерки: $M = (Q, \Sigma, \delta, q_0, F)$ где

Q — конечное множество состояний автомата – строится на основе файлов проектного решения путем синтаксического анализа комментариев на языке описания моделей;

q_0 — начальное состояние автомата ($q_0 \in Q$) – точка входа, запускающая систему на выполнение;

F — множество заключительных состояний системы, $F \subset Q$;

Σ — алфавит (в данном случае - произвольные текстовые сообщения, идентифицирующие переход);

δ — функция переходов. В нашей математической модели программы определим δ следующим образом: $\delta : K \times (D \cup (E \cup \{\varepsilon\})) \times P \times (N \cup \{1\}) \rightarrow 2^{K \times (E \cup \{1\}) \times W}$, где

$K \in Q$ - состояние программной системы; D - словесное действие, однозначно

идентифицирующие переход из данного состояния; E – событие (Event) или исключительная ситуация (Exception), $E \in Q$, которые могут быть сгенерированы во время перехода в следующее состояние; P – вероятность перехода из заданного состояния по действию D .

Наша модель программной системы является вероятностной. Вероятности различаются на априорную вероятность, примерно определяемую программистом, а также апостериорную – вероятность, вычисляемую динамически на какой-то момент времени работы программной системы тестирующей подсистемой на основе модели программы.

N – кратность. Система может иметь несколько экземпляров одного объекта, потоков, клиентов, работающих одновременно. Если логика работы программы зависит от числа подключенных клиентов, то в процессе перехода из состояния в состояние необходимо учитывать кратность. W – флаг состояния системы, определяемого разработчиком как ошибочный. Если система попала в такое состояние, значит имеются какие-то проблемы. Ошибочные состояния и исключительные ситуации используются нейросетевым модулем тестирующей подсистемы для анализа, работала ли система устойчиво в данный прогон.

После описания модели системы в виде конечного автомата программистом в исходном коде программы на специальном XML языке описания моделей (назовем его AXML) при помощи разработанных средств предлагается создать проект модели всей системы, куда включить файлы исходных кодов всех модулей распределенной системы с указанием типа языка и IDE. Планируется поддержка исходных кодов на языках и средах разработки Java Eclipse SDK, MS Visual Studio 2005, Ruby Rails, PHP, а также возможности их будущего расширения путем реализации шаблонов поддержки средств разработки. В настоящее время разрабатывается модуль расширения (Add-in) для среды MS Visual Studio.

В результате работы синтаксического анализатора в местах определения состояний системы и переходов в теле программы создается специальный код, который во время работы системы обеспечивает отсылку данных о том, в каком состоянии находится система в данный момент, на многопоточный сервер системы тестирования. Это позволяет строить динамически обновляемый конечный автомат работы системы ($M_{дин}$). Используя данные, получаемые в процессе работы тестируемой системы, можно определить корректность построения модели системы (все переходы из состояния в состояние описаны), построить граф переходов, определить время перехода из состояния в состояние, наиболее часто вызываемые состояния (а следовательно и узкие места системы), апостериорную вероятность исхода для каждой пары состояний (можно сравнить ее с априорной вероятностью). Используя эти данные, обучающим нейросетевым модулем с некоторой долей вероятности можно сделать вывод о правильности или неправильности работы программы в данном прогоне.

Кроме динамической модели ($M_{дин}$), синтаксическим анализатором строится статическая модель системы ($M_{стат}$), при правильной модели ($M_{дин} \subset M_{стат}$), описывающая состояния для всех файлов с исходным кодом проекта. Статическая модель позволяет проводить offline тестирование модели, определять достижимость состояний, узкие места системы без ее запуска.

Предполагается также проводить нагрузочное тестирование распределенных систем для языков, поддерживающих сериализацию объектов и Reflection. При этом вводится понятие модели виртуального пользователя ($M_{польз} \subset M_{дин}$), действия которого записываются, а потом воссоздаются при помощи средств сохранения и восстановления объектов в местах перехода между состояниями. Такой механизм будет работать правильно в том случае, если автомат системы таков, что последовательность переходов из состояния в состояние не зависит от уже свершившихся прошлых переходов (цепь Маркова).

Блок — схема работы системы представлена на рисунке 1.

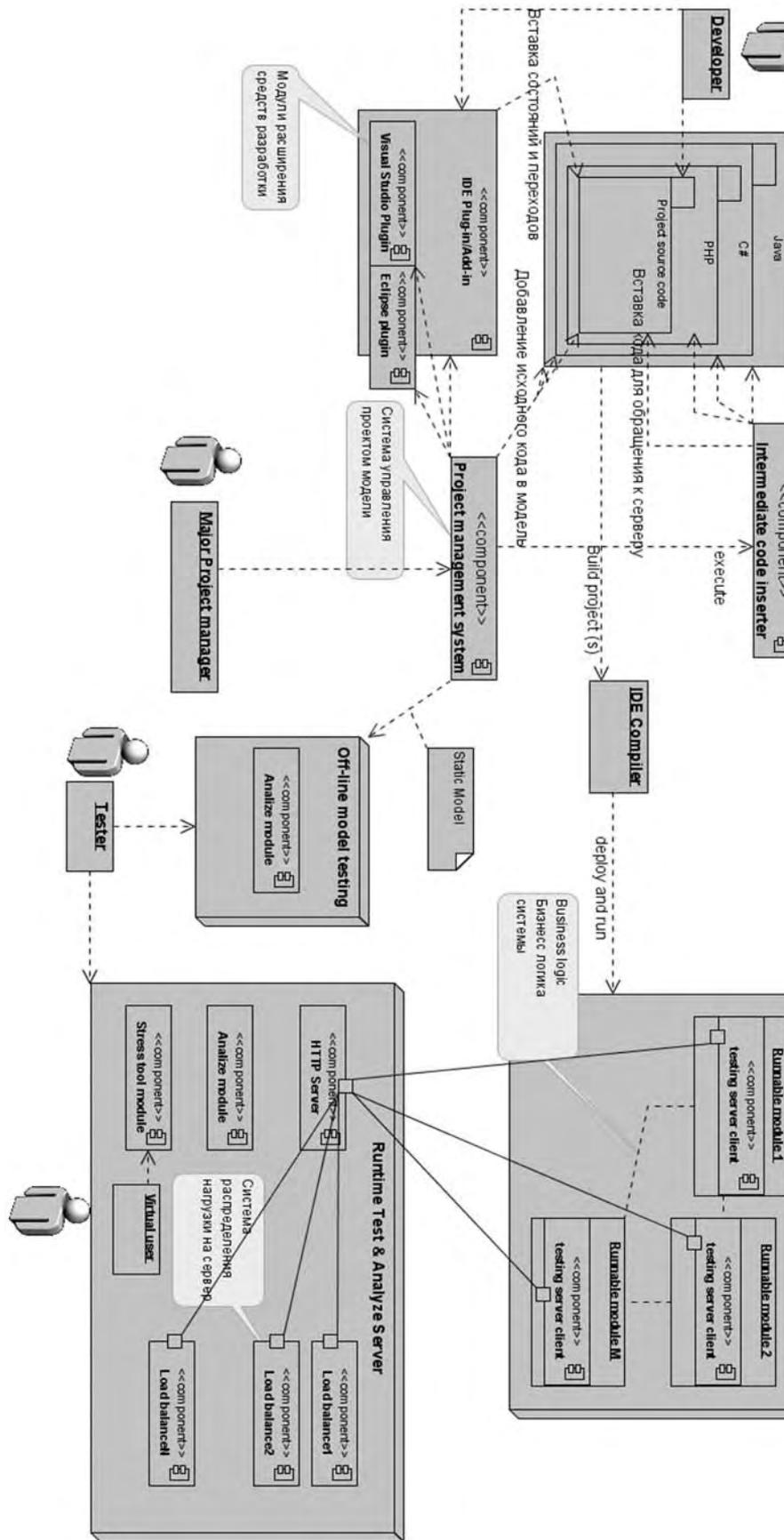


Рисунок 1: Архитектура системы

Таким образом, в работе приведены новые методы для построения и использования

моделей при тестировании программ. После реализации всех модулей системы (срок - 2 года) будет проведено моделирование и проверка работы распределенной системы дистанционного образования АлтГТУ, а также средств проведения командной олимпиады ACM по программированию.

Список литературы

1. Материалы конференции Microsoft Academic Days 2005 Russia. Режим доступа - <http://www.microsoft.com/Rus/AcademicDays2005/Default.aspx>
2. Andreas Blass, Yuri Gurevich, Lev Nachmanson, Margus Veanes. Play to Test. Режим доступа - <http://research.microsoft.com>
3. Avery J. Visual Studio Hacks. - O'Reilly Media, Inc., 2005. – 500 p.
4. Campbell C., Grieskamp W, Nachmanson L. Model-Based Testing of Object-Oriented Reactive Systems with Spec Explorer. - Microsoft Technical Report MSR-TR-2005-59, 2005. – 35 p.

РАСЧЁТ МЕТРИК ПРОГРАММНЫХ СИСТЕМ НА ОСНОВЕ АНАЛИЗА КОДА И ИСТОРИИ ИЗМЕНЕНИЙ

Бурындин С.А. – студент,
Тамплон А.В. – исполнительный директор фирмы «Энтерра»
Алтайский государственный технический университет (г. Барнаул)

Введение

В современном мире разработка ПО превратилась в одну из самых дорогостоящих индустрий и любые узкие места в технологическом процессе его создания могут привести к нежелательным результатам. Удлинение сроков разработки ПО чревато удорожанием конечного продукта, а не выявленные в ходе тестирования ошибки приводят как минимум к снижению его производительности. Поэтому, разработка сложных систем требует постоянного принятия решений, которые существенно влияют на успешность завершения проекта. Возможность оценить последствия принятия таких решений на возможно более ранних этапах разработки системы может упростить и ускорить работу над проектом.

Особенно важно наличие критериев оценки принимаемых решений, а также возможность выразить эти оценки количественно. В этом случае имеется возможность оценить, какие характеристики системы улучшаются в результате принятия какого либо решения, а какие ухудшаются. Таким образом разрабатываемую систему можно "измерить".

Метрики программных проектов – это количественные показатели, отражающие их отдельные характеристики. Метрика – это мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций.

Введение строгих количественных метрик в программирование способствует решению ряда практических задач: - предсказывать вероятное число ошибок в системе с самого начала проектирования; - на основе анализа фазы проектирования системы предсказывать уровень сложности последующего сопровождения; - на основе анализа исходного кода программ прогнозировать уровень сложности процессов тестирования и процент остающихся ошибок; - по оценкам сложности фазы проектирования системы определять конечный размер кода; - определять корреляцию отдельных характеристик программного кода с качеством готовой системы; - контролировать стадии развития проекта; - анализировать явные и скрытые дефекты; - на основе экспериментального сравнения выявлять лучшие методы и технологии.

1. Постановка задачи.

Большое значение имеют метрики при сопровождении программных систем. Поскольку зачастую сопровождать систему должна группа людей, которая не занималась разработкой этой системы. Проанализировать и оценить полученную для сопровождения систему, найдя в ней достоинства и недостатки довольно сложно. Еще сложнее начать вносить существенные изменения в такую систему, дополняя ее новыми возможностями.

Анализ программных систем с помощью специализированного инструмента требует адекватного представления этим инструментом понятий используемого языка программирования. По сложности этот инструмент сопоставим с компилятором для используемого при разработке системы языка программирования.

По мере роста актуальности программных метрик на рынке стали появляться различные "измерительные" программы. Одни из них исследовали характеристики проектов и ПО комплексно, другие ориентировались на вполне конкретные цели: анализ исходного кода, размеров и структуры отдельных модулей и т. д.

Единственным недостатком всех существующих систем анализа кода является их статичность. То есть мы получаем оценку некоторого снимка проекта. Например, пусть метрика M равна 50. А что это значение нам даёт? Даже если мы будем знать верхнюю и нижнюю допустимые границы метрики, мы сможем оценить лишь текущее состояние проекта. К примеру, значение вышеупомянутой метрики выходит за допустимые границы, соответственно напрашивается вывод о плохой работе разработчиков. Получается, программистам придётся с самого начала создавать свой продукт? Как раз здесь проявляется недостаток статичности: мы не можем определить с какого момента ухудшились показатели по этой метрике. И как быть?

Существуют различные инструменты, позволяющие программистам облегчить разработку и сопровождение проектов. Одним из таких инструментов являются системы контроля версий.

Под системой контроля версий понимается механизм сохранения промежуточных состояний кода разрабатываемого программного обеспечения. То есть с помощью этой системы программист может управлять своими файлами во времени: смотреть историю изменений файлов и каталогов, возвращаться к более ранним версиям кода, объединять несколько версий файла.

Вот если метрики применить не к конечному коду, а к версиям, которые хранятся в репозитории системы контроля версий, и по полученным количественным показателям построить график, получим наглядную картину, отражающую результат работы программиста за некоторый промежуток времени. Посмотрев на результат анализа версий, разработчик сразу определит, с какого момента его код перестал отвечать требованиям и исправит проблему.

2. Обзор существующих метрик.

Рассмотрим метрики, наиболее актуальные для программных проектов.

2.1. Метрики размера

Количественные метрики

Самой распространенной метрикой исходного кода ПО, отражающей размер программного проекта, является показатель количества строк кода (Source Lines Of Code, SLOC). В большинстве случаев главное предназначение этой группы метрик, независимо от способа их вычисления, состоит в том, чтобы оценить объем работ по проекту, и, соответственно, быть основой для таких показателей, как стоимость и длительность его реализации.

2.2. Метрики сложности

Другая распространенная группа метрик программных проектов – показатели, характеризующие их сложность. Эти метрики используются главным образом для апостериорного анализа, однако могут применяться и на ранних стадиях работы при осуществлении проектирования.

Основная цель метрик сложности – выявить наиболее критичные участки программного проекта, которые являются потенциальными источниками ошибок и повышенных рисков на всех стадиях его жизненного цикла.

Одна из самых распространенных таких метрик – цикломатическая сложность, впервые предложенная Томасом МакКейбом (Thomas McCabe) в 1976 г. Данная метрика предназначена для оценивания сложности потока управления программы (control flow graph) и вычисляется на основе ориентированного графа, где вычислительные операторы или выражения представляются в виде узлов, а передача управления между узлами – в виде дуг.

Формула вычисления цикломатической сложности выглядит следующим образом:

$$C = e - n + p,$$

где e – число ребер, n – число узлов, p – число соединенных компонент графа управляющей логики. Упрощенно формулу можно рассматривать как количество ветвлений, которые может проходить программа, увеличенное на единицу.

Как правило, при вычислении цикломатической сложности логические операторы не принимаются во внимание, допускается также упрощенный подход, в соответствии с которым собственно построение графа не производится, а показатель определяется на основании подсчета числа операторов управляющей логики (if, switch и т. д.) и возможного количества путей исполнения программы. Метрика цикломатической сложности может быть рассчитана для модуля, метода и других структурных единиц программы.

В процессе анализа значений показателя для отдельных структурных элементов можно выявить элементы с высоким значением показателя (к примеру, нормальное значение показателя для метода – не выше 5–7), что свидетельствует о сложности их управляющей логики и, соответственно, высоких трудозатратах на разработку, тестирование и сопровождение.

Вычисление метрики в ходе реализации проекта (а при детальном проектировании оно возможно еще на этом этапе, не дожидаясь стадии кодирования) позволяет своевременно определить наиболее сложные, сопровождающиеся высокими рисками, структурные единицы и принять меры по устранению рисков за счет внесения коррективов.

В целом, метрики цикломатической сложности являются весьма хорошим способом своевременно зажечь перед разработчиками «красный свет», прекратить дальнейшее усложнение отдельных составляющих проекта и упростить их, предупредив вероятные проблемы с запутанным и нестабильным кодом в будущем.

2.3. Объектно-ориентированные метрики

С переходом от структурной к объектно-ориентированной (ОО) парадигме программирования возникла потребность и в создании соответствующих метрик. Поскольку основные элементы конструирования ОО программ – это классы, реализующие функциональность посредством методов, то именно классы и методы являются категориями, которыми преимущественно оперируют ОО метрики. Помимо таких очевидных метрик, как общее число классов, методов, атрибутов, средних показателей числа методов и атрибутов на класс и пр., применяются гораздо более комплексные метрики, с помощью которых можно судить не только об объеме исходного кода ОО проекта, но и о его сложности, качестве, соответствии основным принципам ОО парадигмы и т. д. Рассмотрим некоторые метрики.

1. Связанность между классами объектов (CBO).

Связанность между классами объектов (Coupling Between Object classes-**CBO**) позволяет определить количество классов, с которыми связан данный класс. Это означает, что один класс использует методы или экземпляры другого класса.

Вероятное влияние метрики на характеристики системы:

- Слишком большая связанность классов отрицательно влияет на модульность проекта и не позволяет повторно использовать классы.
- Большое количество взаимосвязей увеличивает зависимость других частей системы от данного класса и усложняет сопровождение системы в целом.
- Сильно взаимосвязанная система требует большего количества тестов и времени на тестирование.

Заметим, что причины связанности классов в данной метрике значения не имеют.

2. Количество потомков (NOC).

Количество потомков (Number Of Child - **NOC**) позволяет определить количество непосредственных потомков данного класса.

Вероятное влияние метрики на характеристики системы:

- Чем больше потомков у класса, тем больше повторное использование его методов. Это положительное свойство.
- Чем больше потомков у класса, тем больше вероятность неправильного использования базового класса-родителя. Это отрицательное свойство.
- Чем больше потомков у класса, тем большее влияние он оказывает на систему в целом. Такой класс требует более тщательного тестирования.

3. Количество переопределенных методов в подклассе (NMO).

Количество переопределенных методов в подклассе (Number of Methods Overwritten-**NMO**). Эта метрика определяет количество методов класса-предка, переопределенных в классе-потомке. Правильное проектирование системы предполагает расширение классов-предков, а не их переопределение. Большая величина этой метрики сигнализирует об ошибках проектирования.

4. Отсутствие сцепления в методах (LCOM).

Отсутствие сцепления в методах (Lack Cohesion Of Methods-**LCOM**) позволяет оценить зависимость методов класса друг от друга. Для вычисления этой метрики подсчитывается количество пар методов, которые не используют общие атрибуты класса. Затем подсчитывается количество пар методов, которые используют общие переменные. Метрика **LCOM** равна разности между первым числом и вторым. Если при этом получается отрицательное число, то значение метрики считается равным 0.

Вероятное влияние метрики на характеристики системы:

- Отсутствие сцепления методов в классе желательно, поскольку увеличивает инкапсуляцию атрибутов в классе.
- Отсутствие сцепления методов увеличивает вероятность его расщепления на два подкласса.
- Большое сцепление методов класса увеличивает вероятность их ошибочного поведения.

Завершая наш далеко не полный обзор метрик программных проектов, можно утверждать, что в настоящий момент существует весьма обширное число показателей, с помощью которых можно измерять множество различных аспектов создания программного обеспечения. Зачастую речь идет не о том, что одна метрика лучше другой. Все они позволяют посмотреть на один и тот же процесс под разными углами зрения, поэтому используются в комплексе и только так могут служить отправной точкой для принятия объективных решений.

КЛАСТЕРИЗАЦИЯ ТЕКСТОВ БОЛЬШОЙ РАЗМЕРНОСТИ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Гозман Д.М. – студент, Крючкова Е.Н. – к.ф.-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

В настоящее время все большее количество информации переносится на цифровые носители, и встает проблема ее структуризации. Многие поисковые серверы уже занимаются сходной проблемой и даже частично ее решают, выбирая из множества текстов наиболее подходящий к смыслу запроса. Но на данный момент очень мало автоматических средств, позволяющих хорошо распределить научные, литературные и прочие тексты по их содержанию, по проблемам, в них затронутым. Решение же этой проблемы должно позволить структурировать всю информацию на естественном языке в электронном виде. Кроме того, поиск информации станет смысловым, а не привязанным к ключевым словам.

В данной работе предлагается следующая математическая модель представления текстов на естественном языке. Будем исходить из следующего предположения. Все существительные объединим в понятия, образующие иерархическую структуру. Так, синонимы будут стоять на одном уровне, а понятие «животное» будет предком понятия «собака». При этом у каждого понятия естественным образом появляется уровень абстракции – его глубина в лесе понятий.

Представим текст как последовательность предложений, каждое из которых превратим в семантическое дерево. Узлами дерева будут слова, у каждого слова будут потомки – это слова, которые являются в некотором смысле аргументами первого. Таким образом можно ввести в рассмотрение функции и их суперпозиции. Например, «Черная собака пробежала мимо блестящего автомобиля» превратится в дерево

«пробежала(черная(собака),мимо(блестящего(автомобиля)))»

Введем операцию приведения понятия к определенному уровню абстракции $L \geq 0$. Если слово находится в лесе ниже уровня L , заменим его соответствующим предком уровня L , если выше – оставим без изменений.

Для формализации смысловой близости предложений введем понятие степени различия двух предложений, заданных деревьями T_1 и T_2 , на определенном уровне абстракции $L \geq 0$. Приведем все понятия в деревьях T_1 и T_2 к уровню абстракции L . Теперь найдем максимальное общее поддерево C деревьев T_1 и T_2 . При этом будем считать, что дерево A является поддеревом дерева B , если из B можно удалить некоторые вершины, каждый раз присоединяя потомков удаляемой вершины к ее родителю так, что получится дерево A . Теперь степень различия между предложениями назовем $dif(T_1, T_2) = (|T_1| + |T_2| - 2|C|) / (|C| + 1)$.

Теперь введем понятие степени различия двух текстов. Пусть тексты представляют собой последовательности предложений A_i и B_i . Зафиксируем некий параметр $1 \leq t \leq \min(|A|, |B|)$. Найдем такие подпоследовательности \underline{A} и \underline{B} последовательностей A и B соответственно, что $|\underline{A}| = |\underline{B}| = t$ и сумма значений $dif(\underline{A}_i, \underline{B}_i)$ минимальна (обозначим эту сумму D_t). Определим t -различие, как $((1 + |A| + |B| - 2t) * (D_t + 1) - 1) / (t + 1)$. Минимум из t -различий по всем t и будет степенью различия исходных текстов. Понятие степени различия текстов позволяет естественным образом перейти к их кластеризации.

Рассмотрим два предложения:

«Собака слышала, как прошел человек»

«Кошка увидела черную собаку»

Если рассмотреть эти предложения на самом нижнем уровне абстракции, то общее поддерево будет пустым, но при более высоком уровне абстракции (нижняя строчка) общее поддерево будет *«чувствовать(существо, существо)»*. Соответственно, близость этих предложений составит $(4 + 4 - 2 * 3) / 4 = 0.5$.

Экспериментальная проверка алгоритма трех семантически близких текстов при разных уровнях абстракции показала следующие результаты.

Уровень 0 - высший.

0	0,5	0,75
0,5	0	0,5
0,75	0,5	0

Уровень 1 - средний.

0	2,5	2,083333
2,5	0	2,083333
2,083333	2,083333	0

Уровень 2 - низший.

0	3,583333	2,5
3,583333	0	2,083333
2,5	2,083333	0

Как видим, изменение уровня абстракции достаточно сильно влияет на результаты, и даже может несколько сдвинуть относительную характеристику близости текстов. Однако, при любом уровне абстракции семантически различные тексты остаются непохожими.

Список литературы

1. Люгер, Джордж, Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание.: Пер. с англ. – М. : Издательский дом «Вильямс», 2003. — 864 с.
2. Yang, Y., Guan, H., You. J. CLOPE: A fast and Effective Clustering Algorithm for Transactional Data In Proc. of SIGKDD'02, July 23-26, 2002, Edmonton, Alberta, Canada.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РАБОТЫ С КАРТОГРАФИЧЕСКОЙ ИНФОРМАЦИЕЙ (НА ПРИМЕРЕ ОБРАБОТКИ МАРШРУТОВ)

Силкин П.А. – студент, Лукоянычев В.Г. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

В современном мире точное географическое позиционирование, работа с картографической информацией приобретают все большее значение. Сегодня возможность построить и отследить маршрут движения на карте не просто дань моде – это насущная необходимость для пассажиров, туристов, путешественников, транспортных компаний, перевозчиков. Благодаря этой информации можно сэкономить массу времени, наметив маршрут заранее, выбрав самый оптимальный из возможных вариантов, изменив порядок пунктов назначения.

Помимо работы с различными маршрутами часто необходимо просто отметить что-либо на карте, добавить краткое описание объекта. Например, при проведении научной конференции участникам из других городов, а может быть и стран, обязательно необходимо точно объяснить, как добраться до места проведения конференции (университета, конференц-зала) и до места размещения, проживания. Обычно поступают следующим образом: выделяют на каждого иногороднего участника или делегацию по одному встречающему человеку, которой сопровождает их от вокзала (аэропорта) до места размещения. На следующий день этот человек сопровождает делегацию к месту проведения

конференции, часто он находится рядом и при проведении развлекательных мероприятий. Конечно, такой метод имеет свои плюсы, но все же не продуктивно выделять большое число встречающихся, ведь это отрыв их от работы, учебы. Делегации приезжают в город в одно и то же место (вокзал, аэропорт) и в одно место размещения отправляются, поэтому функциональная роль встречающих одинакова и не столь значительна. Просто иногороднего человека всегда проще встретить, чем на словах (по телефону или e-mail) объяснить, как добраться от вокзала до нужной гостиницы, общежития, адреса. Но если бы, с помощью того же e-mail опривить подробную схему проезда, с указанием маршрутов, названием остановок, ориентирами, по которым можно было бы определить местоположение, основными улицами города, возможно, с указанием достопримечательностей для самостоятельной экскурсии, то для большинства приезжих отпала бы необходимость в личном сопровождении.

Одной из самых больших трудностей, встающих перед человеком, собравшимся переместиться из населенного пункта А в В, является получение полной информации обо всех возможных вариантах перемещения, их стоимости, других важных характеристиках. Без обладания подобной информацией, а также без возможности наглядной работы с ней сложно сделать оптимальный выбор маршрута.

Для туристов важным является проработка маршрута передвижения заранее, включение в него как можно большего числа достопримечательностей, интересных природных объектов. При этом важно контролировать удаленность пути от городов, сел, дорог, чтобы всегда знать, куда идти в непредвиденных ситуациях.

В настоящее время существуют программы и сайты, предоставляющие возможности по работе с картографической информацией. В большинстве своем они обеспечивают следующий сервис: карта города, иногда с детализацией до домов, с указанием номера; маршруты общественного транспорта; информация об организациях города, с возможностью получения краткой информации о ней и контактных данных; возможность устанавливать свои заметки и пояснения на карте; другие возможности.

Из программ наиболее известен «ДубльГИС», а среди российских сайтов - «Яндекс.Карты» (<http://maps.yandex.ru/russia>). Данные разработки при своей функциональности накладывают и ограничения: возможность бесплатной публичной публикации подготовленного фрагмента карты ограничена, либо малым размером участка, либо еще как-то; нельзя использовать свое изображение местности, т.е. добавлять информацию можно только поверх карт, предоставляемых разработчиком; нельзя строить свои маршруты транспорта, задавать их параметры; другие ограничения.

Основным ограничением, конечно, является отсутствие возможности создать карту с картографической информацией «с нуля», используя свой эскиз, фотоснимок, либо план местности. Дополнительную информацию пользователь может добавлять только поверх информации от разработчика программы, сайта. А ведь, часто информация на картах программ избыточна, из-за единого масштаба невозможно компактно расположить наиболее важные объекты, опустив длинные переезды транспортом, и т.д. Главное, чего не позволяют сделать выше названные программы, так это создать свою собственную карту, схему.

Для снятия данных ограничений, расширения функционала, предоставление университету инструментального средства по созданию карт и схем для иногородних гостей разрабатывается данное программное обеспечение.

С помощью программного обеспечения возможно решать следующие задачи:

1. создание баз данных маршрутов различных типов (городских, междугородних, региональных, национальных, межгосударственных и межконтинентальных), с возможностью произвольного деления на виды по желанию пользователя (например, ж/д, авиа или автобус, троллейбус); делением на классы (комфортности, скорости и др.); с делением на перевозчиков (по маршрутам, либо транспортным средствам); с возможностью задания остановочных пунктов, станций, вокзалов, их объединения в

- группы (населенные пункты, города); с возможностью уточнения информации о маршрутах посредством введения расписания, стоимости, продолжительности пути и другой информации; с возможностью наложения базы данных на карту, изменение карты, возможность использования нескольких карт для одной базы маршрутов;
2. просмотр базы данных маршрутов в программе конечного пользователя (desktop-приложение), наглядное представление пути с помощью карты, с возможностью поиска пути между двумя пунктами по широкому спектру условий, как-то: стоимость, продолжительность, указание конкретного временного периода, класса (комфортности, например), перевозчика и других условий;
 3. встраивание приложения по визуальному представлению маршрутов в различные программные комплексы, например, в браузер просмотра CD-диска.
 4. другие задачи.

Результатом работы является комплекс программ, работающий с базами данных маршрутов транспорта, отвечающий на часто возникающие вопросы при использовании транспорта городского, междугороднего, и т.д.

Состав комплекса:

- Программа создания, ведения (редактирования, обслуживания) базы данных городов. Для каждого города в базе данных должно храниться: остановки (пункты), маршруты, карты. Для каждой карты свои координаты остановок и маршрутов на ней. Программа должна иметь удобный, дружелюбный интерфейс пользователя. Работа с картой (задание координат) должна осуществляться с помощью мыши. Возможно, предусмотреть защиту базы данных (разделение доступа) и средства резервного копирования.
- Программа конечного пользователя, использующая базу данных в качестве источника данных и не изменяющая ее. Интерфейс должен быть дружелюбным, простым и наглядным. Функции, выполняемые этой программой:
 1. выбор (изменение) города;
 2. выбор (изменение) карты;
 3. вывод маршрута, выбранного пользователем: сведения и список остановок в обоих направлениях, отображение, для наглядности, маршрута на карте;
 4. вывод остановки, введенной пользователем: сведения и список маршрутов, проходящих через нее, отображение, для наглядности, остановки на карте;
 5. поиск пути (последовательности маршрутов) от начальной до конечной остановки (пункта), задаваемых пользователем, отображение этого пути на карте;
 6. предусмотреть встроенную помощь пользователю, обработку ошибок, контекстные подсказки, наличие информации о программном продукте.

Предусмотрена возможность изменения, дополнения, полной замены базы данных без изменения исходного кода программного продукта. Дополнение базы данных через интернет.

Помимо комплекса создана база данных, демонстрирующая возможности программы. Разработанный комплекс работоспособен в двух вариантах: для конечного пользователя (только использование базы данных, без программы ведения и создания) и в полноценном варианте.

КОНСТРУКТОР НЕЙРОСЕТЕЙ С ВОЗМОЖНОСТЬЮ ИМПОРТА/ЭКСПОРТА НЕЙРОСЕТЕЙ В БАЗУ ДАННЫХ

Константинов А.Ю. - студент, Крючкова Е.Н. - к.ф.-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

В связи с увеличением количества сайтов в глобальной сети Интернет возникает потребность автоматической оценки их качества. В данной работе предлагается построить систему оценки качества сайтов с помощью нейросетей. Поскольку это новый и практически неисследованный вопрос, неясна конструкция нейросети, следовательно, нейросеть должна быть легко модифицируемой. Эта проблема может быть решена с помощью автоматического конструирования нейросетей с сохранением сети в виде файла для последующих исследований. Более того, была поставлена задача генерации кода нейросети на каком-либо языке программирования высокого уровня для последующего использования нейросетей в прикладных задачах (не обязательно в задаче оценки качества сайтов).

Для автоматического конструирования нейросетей был спроектирован и реализован «Конструктор нейросетей» (далее просто «Конструктор»). «Конструктор» был реализован с помощью языка программирования С# с использованием объектных принципов программирования. Принцип работы «Конструктора» и конструирования нейросетей представлен на рис. 1.

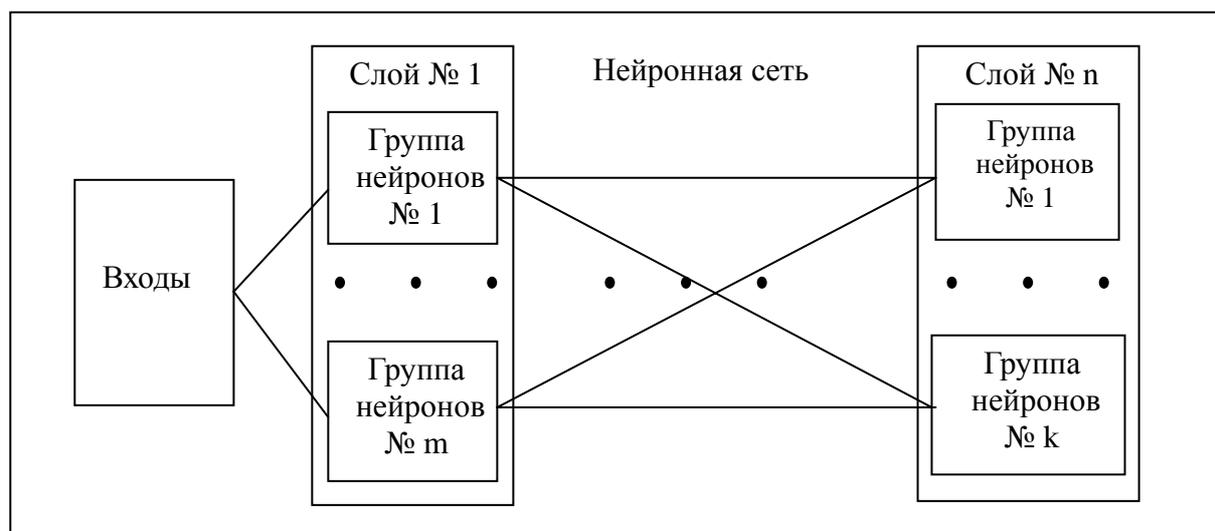


Рис. 1. Схема нейросети в общем случае

Как видно из рис. 1, сеть состоит из некоторого числа слоёв, слои состоят из некоторого числа нейронов. Нейроны могут отличаться между собой по количеству входов и по весам. Каждый нейрон можно просчитать, то есть вычислить пороговую функцию, аргументом которой является уровень активации нейрона (взвешенная сумма его входных сигналов $\sum w_i \cdot x_i$).

Нейроны с одинаковым количеством входов составляют группу нейронов. Группы нейронов могут образовывать связи друг с другом. Просчитать группу нейронов – просчитать каждый нейрон из этой группы.

Слой состоит из групп нейронов. Просчитать слой – значит просчитать все группы нейронов в слое.

Сеть может состоять из одного слоя (однослойная сеть) или нескольких слоёв (многослойная сеть). Просчитать сеть – просчитать все слои слева направо (как они расположены на экране). Т.е. каждый последующий слой берёт выходные значения нейронов

из предыдущего слоя. Если сеть с обратным распространением сигнала, то слой, к которому подключён слой, расположенный правее, берёт выходные значения нейронов данного слоя, просчитанные во время предыдущей итерации. Конечная цель просчёта сети – получить какие-либо выходные значения нейронов последнего (чаще всего) слоя.

На рис. 2 представлена нейросеть, реализующая функцию исключающего логического ИЛИ, созданная с помощью «Конструктора».

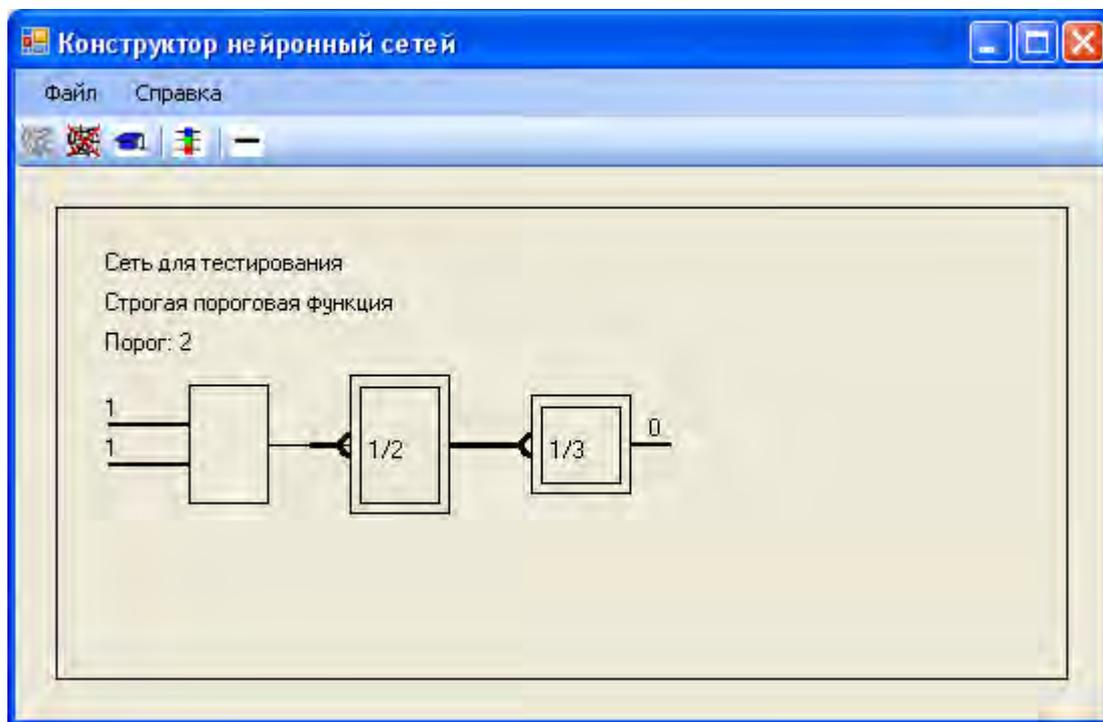


Рис. 2. Пример двухслойной нейронной сети, реализующей исключающее логическое ИЛИ

Выше было описано, что сделано. Далее пойдёт речь о том, что ещё предстоит сделать.

После того, как пользователь данного «Конструктора» найдет удачный вариант нейросети, решающей его задачу, ему будет необходимо сохранить результат работы. Было решено сохранять нейросеть в файле в XML-формате.

Рассмотрим преимущества данного решения:

- На основе XML-файла можно создавать базы данных. То есть для каждой конкретной задачи можно конструировать несколько нейросетей и сохранять их в базу данных. Затем выбирать из БД нейросети, экспериментировать с ними и выявлять нейросеть, которая лучше всего подходит для решения данной задачи;
- Объектно-ориентированным языкам программирования легче «понимать» файл в XML-формате. Следовательно, будет легче реализовать функции сохранения и загрузки нейросети из файла, а также функцию генерации исходного кода нейросети с помощью XML-файла;
- XML – мультиплатформенный язык.

После того, как найден удачный вариант нейросети для решения какой-либо задачи, эта сеть будет использоваться неоднократно. Результат работы нейросети можно использовать как результат работы отдельной программы, а можно использовать как результат работы отдельного модуля в более сложной программе. Поэтому встаёт вопрос о генерации исходного кода нейросети на языке программирования высокого уровня, который можно откомпилировать в исполняемый файл (отдельная программа), а можно вставить в исходный код другой программы как отдельную функцию.

Загрузка нейросети из файла подразумевает выполнение в программе каких-то команд. Эти команды фактически являются исходным кодом нейросети. Поэтому, если найти способ устанавливать однозначное соответствие между нейросетью, хранимой в оперативной памяти, и сетью, хранимой в файле, то можно найти способ генерации исходного кода, причём код будет генерироваться однозначно относительно сконструированной сети.

Автоматическая оценка качества сайтов – новый, неизученный вопрос. Поэтому, выдвинутое предположение о том, что использование нейросетей приведёт к успеху в решении данного вопроса, требует тщательной проверки. Применение «Конструктора» поможет в этом разобраться.

ЭЛЕКТРОННЫЙ ЖУРНАЛ ПРЕПОДАВАТЕЛЯ КАК ИНСТРУМЕНТ ПОВЫШЕНИЯ КАЧЕСТВА ОБРАЗОВАТЕЛЬНЫХ УСЛУГ

Слепова О.В. – студентка,

Ананьев П.И. – ст. преподаватель, Кайгородова М.А. – к.э.н., доцент
Алтайский государственный технический университет (г. Барнаул)

В любое время образовательный процесс играл очень важную роль, так как квалифицированные специалисты были нужны всегда. Учебный процесс представляет собой очень сложную структуру, каждый элемент которой взаимодействует друг с другом и зачастую зависит от другого элемента. Основной целью учебного процесса является передача знаний от преподавателя к студентам.

К профессорско-преподавательским должностям относятся должности декана факультета, заведующего кафедрой, профессора, доцента, старшего преподавателя, преподавателя и ассистента.

Студентом является лицо, в установленном порядке зачисленное приказом ректора в Университет для обучения по образовательной программе высшего профессионального образования. Студент получает высшее профессиональное образование по избранному направлению подготовки в пределах соответствующего государственного образовательного стандарта путем освоения соответствующей образовательной программы высшего профессионального образования [1].

Основой учебного процесса является контроль знаний полученных студентом. Контроль может осуществляться различными методами, но оценка знаний проводится в соответствии с внедренной в АлтГТУ с 2005/2006 учебного года модульно-рейтинговой системой квалиметрии учебной деятельности студентов (МРСК), которая является приоритетно важным элементом системы менеджмента качества образования в университете [2]. Квалиметрия учебной деятельности студентов - это мониторинг и измерение их знаний, умений, навыков с целью определения соответствия требованиям ГОС ВПО и запросам потребителей.

Главными целями МРСК являются повышение качества подготовки выпускников университета, реализация системного подхода в проведении менеджмента образовательной деятельности на основе комплексных рейтинговых оценок качества учебной работы студентов при освоении ими программ высшего профессионального образования [2].

МРСК представляет собой процесс, имеющий большой объем, трудоемкость и зачастую отнимает много личного времени преподавателя. Так же следует учитывать тот факт, что не всем преподавателям понятны особенности МРСК и, следовательно, часто возникают проблемы при расчетах промежуточных и итоговых результатов, от чего в свою очередь могут пострадать и интересы студентов.

После внедрения модульно-рейтинговой системы квалиметрии преподавателю стало намного сложнее оценивать знания студентов и рассчитывать промежуточные результаты успеваемости. Сложность выставления оценок заключается в большом диапазоне баллов,

что влечет за собой проблему, связанную с оцениванием результатов работы студента.

Кроме того, большую сложность стала представлять работа, связанная с вычислением промежуточных результатов и итоговых рейтингов студентов. При использовании 100-балльной системы оценки довольно сложно произвести необходимые расчеты, то есть возрастает вероятность возникновения ошибок в подсчетах.

Так как на сегодняшний день нет специализированного программного обеспечения, то все преподаватели производят расчет рейтингов различными способами. Самыми распространенными являются:

- баллы, заработанные студентом, записываются в журнал преподавателя, и все расчеты производятся вручную либо с применением калькулятора;
- баллы, заработанные студентом, записываются в журнал преподавателя, а расчеты производятся в MS Excel;
- у преподавателя есть КПК с MS Excel, все данные заносятся в него и расчеты производятся там же;
- баллы выставляются без каких-либо расчетов (приблизительно).

Высокую сложность также представляет сама система расчета. Она заключается в многочисленных формулах, которые не всегда понятны. Этот факт также часто вводит преподавателей в заблуждение, и они, вместо того чтобы следовать четкой схеме расчетов начинают выставлять приблизительные результаты, нарушая объективность оценивания и как следствие, снижая у студентов мотивацию к труду.

Из выше сказанного можно сформулировать следующие основные проблемы в использовании МРСК:

- процесс работы с МРСК сложен и трудоемок;
- большое число формул и коэффициентов;
- высокая вероятность возникновения ошибок во время расчета рейтингов;
- большие затраты времени на расчеты.

Для решения данной проблемы следует рассмотреть вариант наличия у преподавателя КПК с установленным соответствующим программным обеспечением. В данном случае преподаватель сможет всегда просмотреть необходимые данные, независимо от того, где он находится. Еще одна проблема, которую решает вариант с КПК это то, что у каждого преподавателя он свой, и ненужно ожидать пока освободится компьютер и тем самым экономится порой драгоценное время.

Разрабатываемое прикладное программное обеспечение для КПК должно выполнять следующие функции:

- отражение оценок студентов и расчет промежуточных и итоговых результатов работы студента;
- автоматизацию расчета результатов работы студентов;
- обеспечение возможности изменения всех введенных ранее результатов;
- корректировка весов для частей составляющих формы проверки знаний (например, сдача лабораторной работы состоит из сдачи и защиты, в соответствии с этим возможно изменить веса для этих частей);
- получение необходимых статистических данных.

Серверное программное обеспечение должно выполнять следующие функции:

- добавление, изменение и удаление форм проверки знаний;
- изменение количества форм проверки знаний;
- изменение соответствующих весов;
- определение форм проверки знаний и их количество для соответствующих групп и дисциплин;
- определение весов для каждой составляющей формы проверки знаний, которые потом будут использоваться при расчетах рейтинга студентов;
- отслеживание правильности определения весов;
- конвертирование БД для КПК и обратно;

- определение форм проверки знаний, входящих в расчет первой и второй аттестации. Внедрение электронного журнала преподавателя позволит обеспечить объективную оценку знаний студентов и тем самым повысит качество преподавания.

Список литературы

1. Сёмкин Б.В. Положение об учебном управлении, СМК ПП 03-2004.
2. Учебное управление АлтГТУ. Положение о модульно-рейтинговой системе квалитметрии учебной деятельности студентов от 2005.

РАЗРАБОТКА СИСТЕМЫ ПРОГНОЗИРОВАНИЯ И АНАЛИЗА СВОЙСТВ ХИМИЧЕСКИХ ВЕЩЕСТВ И ИХ СМЕСЕЙ

Рубилина Е.С. – студентка, Инютин С.М. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

В современной химии существует множество методов исследования химико-физических свойств химических веществ и их смесей. Большинство из них опираются на результаты экспериментов, в результате которых получают опытные значения искомых величин. А как быть, когда не удастся найти ни одного экспериментального значения нужного вещества или нет возможности в силу каких-либо причин проводить опыты? В этом случае основные свойства приходится рассчитывать. Расчеты могут основываться на теории, на корреляции экспериментальных значений или комбинации того и другого способа. В данном случае для исследований необходима структурная формула вещества для разбиения молекулы на составляющие группы и атомы.

С развитием и повсеместным распространением компьютерных технологий появилась возможность автоматизировать трудоемкие расчеты и организовать хранение больших объемов информации с обеспечением быстрого доступа к данным не только в больших научных центрах, но и на отдельных кафедрах, и даже в домашних условиях. В современном мире каждый интересующийся человек имеет возможность получить информацию, не прибегая к долгому поиску в книжных источниках. Все красиво, наглядно, понятно, а главное – быстро появляется на экране монитора с возможностью дальнейшего вывода на печать.

Деятельность химиков и технологов связана главным образом с превращением молекул. Основным назначением данной области знаний является получение сведений о поведении молекул в виде достоверных количественных данных о химических, физических и термодинамических свойствах чистых веществ и смесей. Многие необходимые данные уже имеются, а современная молекулярная теория обеспечивает надежную основу для предсказания других нужных величин.

Данный класс задач не сильно сложен, но требует поиска и поиска обработки большого количества определенной информации, проведения расчетов, которые можно автоматизировать при наличии базы данных коэффициентов и свойств для каждого расчетного метода.

В настоящее время на мировом рынке существует несколько программных решений задачи построения структурных формул веществ и проведению анализа некоторых химико-физических свойств этих веществ. У данных программных продуктов есть достоинства, такие, как, например, красота визуализации построенной структурной формулы, возможность минимизации энергии построенной формулы различными методами, что делает молекулу «правильной» в пространстве, возможность подключения дополнительных, предусмотренных производителями модулей для проведения анализа и расчетов. При этом реализованы достаточно сложные и трудоемкие расчетные методы.

Однако, у данных систем есть и неоспоримые минусы. К ним относятся, в первую очередь, коммерческая основа распространения данных продуктов, предполагающая

довольно большую стоимость, особенно если это касается крупных учреждений и организаций. Во-вторых, данные продукты являются в основном зарубежными разработками и не предполагают русификацию в принципе, что осложняет работу с программой человеку, плохо знакомому с иностранными языками. В-третьих, программный код и форматы получаемой информации являются закрытыми, что делает невозможным использование оной, кроме как в виде картинки. Кроме того, в большинстве методов требуется наличие определенных экспериментальных данных, а мы предполагаем в основном работу химика-теоретика, что делает нежелательным запрос большого количества опытных результатов. Однако, имея определенную информацию о результатах проведенных опытов, становится интересным решение определенного класса задач по определению свойств по имеющимся данным.

На данный момент были рассмотрены программные продукты, разработанные в основном в европейских вузах. Например, разработка Кембриджского университета COMSOL FEMLAB 2.3 – это мощный пакет для моделирования физических процессов и математических уравнений, проектирования (CAD) и визуализации данных. Включает свой язык программирования, совместимый с MatLab. Сложен для неподготовленного в программировании пользователя, т.к. требует в некоторых случаях написания кода для реализации визуальных проектов. Для решения уравнений и расчетов данных используется язык MatLab, что требует подготовки либо пользователя, либо настройки кем-то данного проекта. Требуется обязательной установка MatLab, который является коммерческим продуктом, что делает COMSOL еще более дорогим в использовании в государственных и коммерческих организациях.

ChemSW Chemsite version 3.01 Standard - программа для 3D моделирования молекулярных структур. Возможность импорта-экспорта в другие популярные пакеты в виде картинок. Имеет интуитивно понятный интерфейс графического редактора и большую базу 3d шаблонов. Система имеет собственную базу данных химических элементов и групп с привязками к структурному строению вещества. Производится преобразование формулы по энергетическим составляющим. Продукт имеет дополнительные модули для проведения теоретических исследований, которые поставляются за отдельную плату. Продукт защищен и является полностью коммерческим.

Разумным подходом к решению рассмотренных выше проблем является создание программного продукта, автоматизирующего работу химика по расчетам основных свойств без проведения пользователем большой работы по поиску и сбору необходимой информации в печатных изданиях. Необходимость создания редактора структурных формул в рамках данного проекта очевидна.

Особенностью данного продукта предлагаю сделать общедоступность, бесплатность распространения. Не использовать в составе дополнительного коммерческого программного обеспечения в качестве обязательного. Дружественный интерфейс должен только помогать в работе начинающему и делать работу приятной у профессионала.

Для этого разработана структура справочного материала (экспериментально установленные значения определённых свойств, наиболее приближенные к реальности теоретические значения и т.д.). Поскольку одни и те же данные используются при расчетах по нескольким методам и зависят только от структуры молекулы, то предпочтительнее использовать базу данных свойств.

Была проведена большая исследовательская работа по изучению типов методов расчетов и законов, которые будут использоваться при проведении анализа физико-химических свойств в процессе работы программы.

Методы анализа являются по возможности универсальными, т.е. применимыми для большинства элементов органической и неорганической химии, либо имеют аналоги для конкретных случаев исключений.

Прежде всего, хочется отметить, что разработка системы прогнозирования и анализа свойств химических веществ - это прежде всего проектирование и создание:

1. графического редактора структурных формул химических веществ с дружественным пользователю интерфейсом (модуль «Графический редактор»);
2. хранения информации об имеющихся экспериментальных данных, наборе коэффициентов и других параметров по отдельным структурным единицам (модуль «База данных»);
3. модуля проведения анализа построенной в редакторе формулы при помощи имеющихся методов (модуль «Анализатор»).

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ПО КУРСУ «ТЕОРИЯ АВТОМАТОВ»

Колесникова О.А., Трясак А.Ю. – студенты, Сучкова Л.И. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

На сегодняшний день компьютерные программы активно внедряются практически во все сферы человеческой деятельности. Современные компьютерные обучающие программы позволяют улучшить процесс обучения, а также проконтролировать знания и навыки.

Курс «Теория автоматов» состоит из следующих разделов:

- формальные языки и грамматики;
- конечные автоматы-распознаватели, их синтез, детерминизация и минимизация;
- автоматы с магазинной памятью;
- машина Тьюринга;
- конечные автоматы-преобразователи, автоматы Мили и Мура, их синтез и минимизация;
- структурный синтез автоматов;
- асинхронные автоматы.

В настоящее время не существует автоматизированных обучающих систем, охватывающих всю программу курса «Теория автоматов». Однако, содержание курса таково, что создание подобного программного комплекса было бы очень полезным для обучения, так как изучение синтеза абстрактных и структурных автоматов представляет значительные трудности. Кроме того, данный курс содержит большой объем теоретической информации, включающей ряд достаточно сложных алгоритмов.

Целью данной работы является разработка программного комплекса по курсу «Теория автоматов». Основной задачей является автоматизация выполнения лабораторных работ по данному курсу и наглядное представление соответствующего теоретического материала.

На основе анализа поставленной задачи была спроектирована и реализована структура программного комплекса:

- классы типов данных и алгоритмов – реализуют типы автоматов и вспомогательных данных, алгоритмы абстрактного и структурного синтеза автоматов, минимизации и детерминизации;
- классы элементов управления – реализуют элементы управления интерфейса, необходимые для ввода и вывода данных, а также осуществляют синтаксический разбор введенных данных;
- классы для организации интерфейса – осуществляют обмен данными между элементами управления и соответствующими объектами классов типов данных и алгоритмов.

Программный комплекс включает в себя следующие модули:

- модуль по работе с машиной Тьюринга. Включает в себя функции интерпретации машины Тьюринга, в том числе в пошаговом режиме;
- модуль по работе с конечными автоматами-распознавателями. Включает в себя

функции интерпретации конечных автоматов-распознавателей, их минимизации и детерминизации;

- модуль по работе с конечными автоматами-преобразователями. Включает в себя функции интерпретации автоматов Мили и Мура, их минимизации, а также функции структурного синтеза и преобразования автомата Мили в автомат Мура и наоборот;
- модуль по работе с автоматами с магазинной памятью. Включает в себя функции нисходящего и восходящего разбора.

Кроме того, каждый из модулей включает в себя примеры выполнения соответствующих операций, а также небольшие тестовые задания для самоконтроля.

Разработанный комплекс охватывает основное содержание курса «Теория автоматов» для специальности «Вычислительные машины, комплексы, системы и сети» и будет использован студентами для обучения.

РАСЧЕТ ТЕПЛОВЫХ ПРОЦЕССОВ

Барбарош А.Е. – студент, Кантор С.А. – к.ф.-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

На сегодняшний день практически все существующие виды сварочных процессов основаны на применении технологии концентрированного нагрева участков изделий до температуры плавления или до пластического состояния. Нарушение технологии (например, неправильный нагрев или охлаждение) может явиться причиной появления различных дефектов (трещины, недовары и др.). Однако, если не учитывать результаты анализ теплового состояния металла, нельзя достаточно достоверно объяснить большую часть наблюдаемых явлений.

Разрабатываемая в приложении программа предназначена в помощь студентам, обучающимся сварочному производству, в изучении тепловых процессов и связанных с ними расчетов.

Программа позволяет производить расчеты для трех математических моделей.

Первая модель предполагает, что сварка производится на полубесконечном теле (то есть ограниченном только по оси OZ в одном направлении), все физические размеры источника пренебрежимо малы (точечный источник), его действие на тело непрерывно, а скорость движения не очень высока (как правило, не превышает 1 см/с).

Для следующих моделей считаем, что линейный (два параметра пренебрежимо малы по сравнению с третьим) источник непрерывно действует на пластину (тело, ограниченное только по оси OZ). При этом для второй модели источник сварки считаем подвижным, а для третьей быстро движущимся.

Формула мгновенной скорости охлаждения W имеет вид:

$$W = \frac{2 \cdot \pi \cdot \lambda \cdot v \cdot (500 - T_0)^2}{\eta_E \cdot I \cdot U} \quad (1)$$

для первой, и

$$W = \frac{2\pi \cdot \lambda \cdot c_\gamma \cdot (500 - T_0)^3}{\left(\frac{\eta_E \cdot I \cdot U}{V \cdot \delta}\right)^2} \quad (2)$$

для второй и третьей моделей
где

λ – коэффициент теплопроводности, Вт/см·К ;
 c_γ – коэффициент теплоемкости, Дж/см³·К;
 V – скорость сварки, см/с;
 T_0 – начальная температура, С
 $\eta_{\dot{E}}$ – эффективный коэффициент полезного действия, %
 I – сила тока, А
 U – напряжение дуги, В,
 δ – толщина пластины, мм.

При анализе холодных трещин необходимо, используя формулу (1) или (2), рассчитать мгновенную скорость охлаждения, сравнить ее с критическим значением $W_{кр}$, в случае ($W > W_{кр}$) сделать вывод о возникновении холодных трещин сварки и пересчитать начальную температуру как

$$T_0 = 500 - \sqrt{\frac{\eta_{\dot{E}} \cdot I \cdot U \cdot W}{2\pi \cdot \lambda \cdot V}}$$

для формулы первой и

$$T_0 = 500 - \sqrt[3]{\frac{W \cdot \left(\frac{\eta_{\dot{E}} \cdot I \cdot U}{V \cdot \delta}\right)^2}{2\pi \cdot \lambda \cdot c_\gamma}}$$

для второй и третьей моделей.

Формула изменения температуры имеет вид

$$\Delta T(x, y) = \frac{\eta_{\dot{E}} \cdot I \cdot U}{2\pi \cdot \lambda \cdot R} \cdot e^{\frac{-V}{2a}(R+x)} \quad (3)$$

для первой модели,

$$\Delta T(x, y) = \frac{\eta_{\dot{E}} \cdot I \cdot U}{2\pi \cdot \lambda \cdot \delta} \cdot e^{\frac{-V \cdot x}{2a}} \cdot K_0 \left(R \cdot \sqrt{\frac{V^2}{4a^2} + \frac{2\alpha}{c_\gamma \cdot \delta \cdot a}} \right) \quad (4)$$

для второй и

$$\Delta T(x, y) = \frac{\eta_{\dot{E}} \cdot I \cdot U}{V \cdot \delta \cdot \sqrt{4\pi \cdot \lambda \cdot c_\gamma \cdot \frac{x}{V}}} \cdot e^{\frac{V \cdot y - b \cdot x}{4a \cdot x - b \cdot \frac{x}{V}}} \quad (5)$$

для третьей модели соответственно.

Здесь:

α – коэффициент теплоотдачи, см²/с;
 a – коэффициент температуропроводности, см²/с;
 x и y – координаты требуемой точки относительно центра;
 $R = \sqrt{x^2 + y^2}$ – расстояние от центра до интересующей нас точки;
 K_0 – бесселева функция 2-го рода нулевого порядка от аргумента;

При расчете температур необходимо, используя формулы (3), (4) или (5), построить графики зависимости температуры $T = T_0 + \Delta T$ от координат x, y , а так же времени t (в этом случае одна координата становится функцией времени) в различных разрезах (например, температуры точек, лежащих на оси позади движущейся дуги).

При работе с термическими циклами точек необходимо, приняв в формулах (3), (4) или (5) $x = V \cdot t$, построить график зависимости $T(t)$ и наложить его на графики, полученные экспериментально, после чего выполнить их сравнение.

Уравнение зависимости температуры от глубины проплавления имеет вид

$$T_{i\bar{e}} = T_0 + \frac{\eta_{\bar{e}} \cdot I \cdot U}{2\pi \cdot \lambda \cdot z} \cdot e^{-\frac{V \cdot z}{2a}} \quad (6)$$

для первой и

$$T = T_0 + \frac{0,484 \cdot \eta_{\bar{e}} \cdot I \cdot U}{V \cdot c_{\gamma} \cdot \delta \cdot 2z} \cdot \left(1 - \frac{2\alpha \cdot z^2}{c_{\gamma} \cdot \delta \cdot 2a} \right) \quad (7)$$

для второй и третьей моделей.

При расчете глубины и ширины проплавления необходимо, используя формулу (6) или (7), определить глубину проплавления z как решение соответствующего уравнения. Решение получаем методом деления отрезка пополам для (6) и аналитически для (7).

Таким образом, программа предусматривает возможность расчета для различных режимов сварки и может быть разделена на несколько блоков:

1. В блоке «Анализ возникновения холодных трещин» на основании проводимых расчетов делается вывод о возникновении трещин при сварке, а так же в случае возникновения производится расчет необходимых изменений начальных условий;
2. В блоке «Расчет температур» производится построение графиков зависимости температур от расстояния до точки плавления, скорости сварки или времени;
3. В блоке «Расчет ширины проплавления» производится расчет ширины и глубины проплавления;
4. В блок «Термический цикл точки» производятся построения графиков термического цикла для различных точек изделий, а так же их сравнение с результатами, полученными экспериментально.

В дальнейшем в программу будут внесены дополнительные расчеты, в результате чего ее функциональность возрастет.

ПРИМЕНЕНИЕ СУФФИКСНЫХ ДЕРЕВЬЕВ ДЛЯ ЭФФЕКТИВНОЙ ОБРАБОТКИ СТРОК

Колосовский М.А. – студент, Крючкова Е. Н. – к.ф.-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Суффиксные деревья дают большие возможности для решения огромного количества задач на строки. Среди этих задач и крайне сложные задачи, решение которых немислимо без аппарата суффиксных деревьев, однако, человеку, умеющему использовать такую структуру данных, как суффиксные деревья, они могут показаться почти тривиальными. Особый интерес представляют алгоритмы, строящие эту структуру данных за время, линейно зависящее от длины входных данных. Это позволяет добиться линейной сложности решения большинства упомянутых задач.

Первый алгоритм для построения суффиксного дерева за линейное время был предложен Вайнером в 1973 году. Несколько лет спустя (1976 г.) Мак-Крейгом был предложен другой алгоритм более экономный по памяти. И, наконец, в 1995 году финн Эско Укконен предложил алгоритм, обладающий преимуществами алгоритма Мак-Крейга и имеющий более интуитивно понятное истолкование, а также позволяющий строить суффиксное дерево как бы в on-line режиме, т.е. обрабатывая поочередно символы строки слева направо.

Вот несколько наиболее известных задач, решаемых при помощи суффиксных деревьев:

1. Поиск образца (длины n) в строке (длины m). Сложность: препроцессинг $O(m)$, обработка образца $O(n)$. Стоит обратить внимание, что в отличие от хорошо знакомых методов КМП-поиска и Бойера-Мура, в которых препроцессингу подвергается искомый образец, в алгоритме, использующем суффиксные деревья, препроцессингу подвергается строка, в которой будут искаться образцы. Таким образом, в ситуации, когда заранее известна строка, а образцы меняются лучше использовать суффиксные деревья, получая временную сложность алгоритма $O(m+n')$, где n' – суммарная длина k искомым образцов, вместо $O(m \cdot k + n')$ для методов КМП и БМ.
2. Поиск набора образцов (суммарной длины n) в строке (длины m). Сложность: препроцессинг $O(m)$, обработка образцов $O(n)$. Этот алгоритм обладает той же особенностью, что и описанный в пункте 1 (т.е. препроцессингу подвергается строка, а не набор образцов, как в алгоритме Ахо-Корасика).
3. Линеаризация циклической строчки. Эта задача формулируется следующим образом: дана строка S , найти такой циклический сдвиг S , что получившаяся строка будет лексикографически наименьшей. Сложность: $O(m)$.
4. В данной строке найти наибольшую по длине подстроку, повторяющуюся ровно k (или не менее k) раз. Сложность: $O(m)$.
5. Наибольшая (по длине) общая подстрока нескольких строк. Сложность: $O(m)$, где m – суммарная длина всего набора строк. Интересно заметить, что в 1970 году Д. Кнут предположил, что не существует алгоритма для этой задачи линейной сложности.
6. Определение длины общего префикса двух данных суффиксов заданной строки за константное время. Сложность: препроцессинг $O(m)$.
7. Из суффиксного дерева возможно построение таких структур данных, как суффиксный массив и суффиксный автомат. Сложность: построение суффиксного дерева и перевод в другую структуру данных за $O(m)$.

Итак, теперь для того, чтобы объяснить, что представляет собой суффиксное дерево, рассмотрим несколько определений. Первой из них – бор.

Бор (словарь) – это дерево, каждое ребро которого помечено некоторым символом из алфавита. Причем из одной вершины не может выходить двух ребер (к потомкам), на которых записан один и тот же символ. Таким образом, каждой вершине соответствует некоторая строка, получаемая прочтением всех символов на пути от корня в эту вершину (например, корню дерева соответствует пуста строка). Говорят, что вершина *произносит* эту строку. Некоторые вершины бора помечены как *терминальные*. Любой бор (словарь) хранит некоторый набор строк (слов), а именно те слова, которые произносят терминальные вершины.

Суффиксное дерево для строки S – это бор, хранящий все суффиксы строки S .

Пусть длина базовой строки для суффиксного дерева равна m . Заметим, что в суффиксном дереве тогда будет храниться $O(m \cdot m)$ ребер. Очевидно, что, храня описанным образом суффиксы, не возможно достичь линейной оценки $O(m)$ построения суффиксного дерева. Для этого за основу суффиксного дерева берется не описанный выше бор, а, так называемый, *сжатый бор*. Сжатый бор отличается от обычного тем, что все его вершины либо не имеют детей (являются листьями), либо имеют не менее 2 детей, т.е. в сжатом боре цепочки из вершин с единственным ребенком «сжаты» в одно ребро. Таким образом, получаем, что в сжатом боре, хранящем все суффиксы данной строки, будет $O(m)$ вершин, т.к. имеется не более m листьев, являющихся терминальными вершинами, и в каждой внутренней вершине есть ветвление дерева. Есть еще одно отличие сжатого бора – пометка ребер. Заметим, что строка на каждом ребре соответствует какой-то подстроке исходной строки S , а значит, мы можем пометить ребро двумя индексами: началом и

концом подстроки в исходной строке S .

Добавление новой строки в бор (как в обычный, так и в сжатый) происходит за время $O(n)$, где n – длина добавляемой строки. Таким образом, примитивный алгоритм построения суффиксного дерева имеет сложность $O(m \cdot m)$, где m – длина исходной строки S . Реализация алгоритмов построения суффиксных деревьев лежит за рамками этой статьи.

Кратко опишем решение упомянутых выше задач:

1-2. Строим суффиксное дерево по строке. Далее, получив образец, движемся вдоль пути по дереву, произносящему это образец. Если такой путь существует, то значит, найдено вхождение образца в строку, причем число терминальных вершин в поддереве вершины, в которой мы остановились, соответствует числу вхождений. (Примечание: наше прохождение по сжатому (!) бору может остановиться и в середине ребра, тогда под вершиной будет подразумеваться, так называемая, *неявная вершина*, т.е. та, которая существовала бы в несжатом боре).

3. Пусть строка для линеаризации S длины m . Построим суффиксное дерево для строки SS (конкатенация двух строк S). Найдем путь в дереве от корня длины ровно m , причем, двигаться будем по тем ребрам, которые начинаются с лексикографически меньший символов. Такой путь обязательно найдется. Произнесенная строка и есть искомая строка.

4. Построим суффиксное дерево. Найдем вершину с максимальной глубиной в дереве, которая в своем поддереве содержит ровно k (не менее k) терминальных вершин. Строка, произносимая этой вершиной, отвечает необходимым требованиям.

5. Задача похожа на предыдущий пункт, но строится *обобщенное суффиксное дерево* для данного набора строк, где терминальные вершины хранят информацию о том, терминальными для какой строки они являются. Строка, которая произносится вершиной с наибольшей глубиной, содержащей в своем поддереве терминальные вершины от всех строк набора, является ответом.

6. Строим суффиксное дерево. Пусть нужно найти общий префикс для двух данных суффиксов, которым соответствуют терминальные вершины v и u . Найдем вершину – наименьшего общего предка этих вершин (для бинарных деревьев это можно сделать за константное время). Строка, произносимая этой вершиной, является ответом.

7. Решение этих задач требует объяснения строения указанных структур данных, поэтому оставим этот материал для заинтересованных читателей.

Несмотря, на то что, первые эффективные алгоритмы построения суффиксных деревьев были изобретены более 30 лет назад, суффиксные деревья не вошли в набор основных алгоритмов, которые знает любой программист, вероятно, потому что работы Вайнера и Мак-Крейга слишком сложны для понимания. Однако алгоритм Укконена появился относительно недавно и, возможно, в скором времени суффиксные деревья займут более достойное место в мире программирования, как очень мощное средство поиска и обработки информации.

Список литературы

1. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология. – СПб.: БХВ-Петербург, 2003. - 654 с

ФУНКЦИЯ ГРАНДИ В ПРИМЕНЕНИИ К ИГРАМ НА ТРИАНГУЛЯЦИИ ДЕЛОНЕ

Акиншин А.А. – студент, Крючкова Е. Н. – к.ф-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

1. Введение

В настоящее время активно процветает сфера исследований алгоритмов и решения оптимизационных задач в области информационных технологий. Одним из наиболее важных и актуальных направлений в данной области является рассмотрение прикладных задач, использующих разнообразные структуры данных и работающие с ними высокоэффективные алгоритмы. В число ключевых тематик данной области, помимо прочего, входит теория игр и геометрические задачи на построение и анализ. Достаточно важным является изучение проблемы конденсации разрозненных алгоритмов для выполнения поставленных целей. В настоящей статье рассматривается вышеуказанная проблема и методы её решения, а именно: анализируется ациклический граф некоторой игры, построенный на типовой системе геометрических объектов.

2. Постановка задачи

Рассмотрим следующую задачу: пусть на декартовой плоскости дана некоторая система точек, причём никакие четыре не лежат на одной окружности (во всей дальнейшей статье данное условие рассматривается как само собой разумеющееся). Пусть на этих точках сторонний наблюдатель строит некоторую триангуляцию, обладающую следующим свойством: описанная окружность каждого треугольника не содержит внутри себя точек из этой же системы. Треугольники разобьем на два класса: помеченные и непомеченные. Изначально все треугольники будут являться непомеченными. Зададим на полученной структуре следующую игру для двух человек: за ход разрешается рассмотреть некоторую связную область непомеченных треугольников и некоторую произвольную прямую. Результатом хода будем считать то обстоятельство, что все треугольники из данной области, пересекаемые прямой станут помеченными. Проигрывает тот, кто не может сделать ход. Для данной системы точек необходимо определить, кто имеет оптимальную стратегию: первый, второй или невозможно определить (зависит от того, как на данной системе точек построена триангуляция).

3. Теоретические положения

3.1. Диаграммы Вороного.

3.1.1. Определение. Пусть дана некоторая система точек. Для каждой точки P из системы определим ГМТ точек, для которых точка P ближе, чем любая другая точка исследуемой системы, как многоугольник Вороного. Совокупность многоугольников Вороного для каждой точки называется диаграммой Вороного для данной системы точек. Вершины многоугольников называются вершинами диаграммы Вороного, а соединяющие их отрезки (входящие в диаграмму) – рёбрами диаграммы Вороного.

3.1.2. Свойства диаграмм Вороного.

3.1.2.1. Каждая вершина диаграммы Вороного является точкой пересечения в точности трёх рёбер диаграммы.

3.1.2.2. Вершины диаграммы Вороного являются центрами окружностей, каждая из которых определяется тремя точками исходной системы, а сама диаграмма является регулярной со степенью вершин, равной трём.

3.1.2.3. Для каждой вершины диаграммы Вороного рассматриваемой системы соответствующая ей окружность не содержит никаких других точек данной системы.

3.1.2.4. Каждый ближайший сосед точки исходной системы определяет ребро в

- многоугольнике Вороного.
- 3.1.2.5. Многоугольник Вороного является неограниченным в том и только в том случае, когда соответствующая ему точка лежит на границе выпуклой оболочки точек исходной системы.
 - 3.1.2.6. Граф, двойственный диаграмме Вороного, является триангуляцией исходной системы точек.
 - 3.1.2.7. Оптимальный алгоритм построения диаграммы Вороного обладает сложностью $\theta(N \cdot \log N)$, где N – количество точек в исходной системе.
- 3.2. Триангуляция Делоне
- 3.2.1. Определение. Триангуляция системы точек – это разбиение выпуклой оболочки системы точек на треугольники с вершинами в этих точках так, что каждая точка является вершиной хотя бы одного треугольника. Триангуляция называется триангуляцией Делоне, если внутренняя область описанной окружности любого треугольника не содержит точек исходной системы.
 - 3.2.2. Свойства.
 - 3.2.2.1. Если в исходной системе точек никакие четыре точки не лежат на одной окружности, то существует ровно одна триангуляция Делоне.
 - 3.2.2.2. Триангуляция, представляющая собой граф, двойственный диаграмме Вороного, является триангуляцией Делоне.
 - 3.2.2.3. Оптимальный алгоритм построения триангуляции Делоне обладает сложностью $\theta(N \cdot \log N)$, где N – количество точек в исходной системе.
- 3.3. Теория игр.
- 3.3.1. Основные положения теории игр.
 - 3.3.1.1. Игра на ациклическом графе – это процесс, в котором участвуют двое так называемых игроков. Они по очереди двигают фишку по рёбрам графа из вершины в вершину. Тот, кто не может сделать ход, называется проигравшим.
 - 3.3.1.2. В любой игре на ациклическом графе один из игроков всегда имеет выигрышную стратегию.
 - 3.3.1.3. Состоянием игры называется текущее положение фишки. Состояние называется выигрышным, если имеется выигрышная стратегия из этого состояния, и проигрышным, в противном случае.
 - 3.3.1.4. Сумма игр – это игра на ациклическом графе, множество вершин которого является прямым произведением множеств вершин графов, входящих в сумму, а рёбра определяются по правилу (в настоящей статье приведён пример для суммы двух игр, для суммы n игр определение даётся аналогично): ребро $(u,v) \rightarrow (p,q)$ существует, если $u=p$ и во втором графе есть ребро $v \rightarrow q$ или $v=q$ и в первом графе есть ребро $u \rightarrow p$.
 - 3.3.2. Функция Шпрага-Гранди.
 - 3.3.2.1. Определение. Функция Шпрага-Гранди – это функция, представляющая собой отображение множества состояний игр на множество целых неотрицательных чисел, обладающее свойствами:
 - если состояние игры является проигрышным, то его функция Гранди равна нулю;
 - если состояние игры является выигрышным, то его функция Гранди равна минимальному целому неотрицательному числу, которое не попадает в множество значений функций Гранди состояний этой же игры, в которые мы можем попасть из текущей.
 - 3.3.2.1. Функция Гранди от суммы игр равна поразрядной сумме по модулю 2 функций Гранди составляющих её игр.

4. Решение задачи

Внимательно проанализировав приведённые выше теоретические положения можно понять, что требуемая триангуляция является триангуляцией Делоне. Т.к. известно, что никакие четыре точки исходной системы не лежат на одной окружности, то можно заключить, что существует единственный способ получить требуемую триангуляцию. Это означает, что игрок, обладающий выигрышной стратегией, определяется однозначно. Заметим, что данную задачу несложно решить, используя функцию Гранди. Действительно, рассмотрим состояние игры как связную подмножество треугольников исходной триангуляции. Заметим, что ход игрока заключается в переходе от исходной игры к сумме игр более мелкой размерности. Свойство 3.3.2.2 позволит нам легко пересчитывать функцию Гранди для исходной игры. В конце решения сравним функцию Гранди от исходной триангуляции с нулём и получим ответ. С реализацией первой части задачи (построение триангуляции Делоне) проблем не возникает: существует стандартный алгоритм, который работает весьма эффективно (см. 3.2.2.3). В части подсчёта функций Гранди возникает вопрос об оптимальном хранении ответов для различных состояний игры. Данная проблема имеет множество решений, но ограничимся приведением двух наиболее действенных. Первый подход применим при малых ограничениях. В этом случае мы можем закодировать каждое состояние игры битовой маской по номерам содержащих его треугольников. Далее методом динамического программирования можно определить функции Гранди для всех подмножеств. Второй подход рационально применить при более серьёзных ограничениях. Заметим тот факт, что в первом подходе считалось очень много лишней информации: нас интересуют не все подмножества треугольников исходной триангуляции, а лишь связные. Заметив это, можно утверждать, что мощность множества допустимых состояний будет не так уж и велика. Из этого можно сделать вывод, что достаточно логично применить хэш-таблицу, в которой и будут храниться наши ответы.

5. Выводы

В настоящей работе было рассмотрено решение некоторой прикладной задачи, использующее ряд стандартных понятий и алгоритмов. В процессе этого анализа достаточно убедительно показано, какой мощный и эффективный инструмент рассмотрен в данной работе. В современном обществе рассмотренные темы имеют достаточно обширное применение. К примеру, триангуляция Делоне широко используется в области моделирования трехмерных объектов, построения аналогов реальных рельефов и ландшафтов, анализа данных воздушного лазерного сканирования [5], восстановления трехмерных горизонтально неоднородных тестовых скоростных моделей [6] и многих других. Теория игр имеет колоссальное количество практических приложений: её элементы активно используются в экономике, а так же нередко применяются в биологии, социологии, политике и других науках. Очень большое значение она имеет в сфере работы с системами искусственного интеллекта и кибернетики.

Список литературы

1. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение. - М.:Мир,1989
2. Шляйхер Игры и функционал Шпрага-Гранди, 2005
3. Материалы сайта <http://ru.wikipedia.org>
4. Материалы сайта http://www.citforum.ru/programming/theory/alg_triangel/
5. Материалы сайта <http://www.injgeogis.ru/publications/5/>
6. Материалы сайта <http://idg3.chph.ras.ru/sanina/rffi.htm>

АНАЛИЗ БИЕКТИВНОГО ОТОБРАЖЕНИЯ АВЛ-ДЕРЕВЬЕВ В КОДИРОВАННУЮ СТРУКТУРУ ДАННЫХ СОХРАНЯЮЩЕЕ ОТНОШЕНИЕ ПОРЯДКА

Закурдаев А. В. – студент, Крючкова Е. Н. – к.ф.-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

1. Введение

В современном мире, когда объёмы информации бьют все мыслимые и не мыслимые рекорды, как не когда ранее для практического применения практически во всех сферах человеческой деятельности важны эффективные алгоритмы и эффективные же методы представления данных. Именно поэтому процветает сфера исследования все возможных алгоритмов и задач, а так же методов оптимизации. Одним из наиболее важных направлений в оптимизации чего бы то ни было является правильный выбор структур представления данных, обеспечивающих одновременно высокую скорость доступа к данным, а так же высокую скорость выполнения ряда других операций, потребление сравнительно небольшого объёма оперативной памяти и, что очень желательно, простоту реализации. Хочется также отметить, что, как правило, многие задачи можно решить эффективной комбинацией различных хорошо известных алгоритмов, что значительно упрощает процесс поиска и оптимизации решения. В настоящей статье рассматривается вышеизложенная проблема и методы её решения, а именно: анализируется применение АВЛ-деревьев и кодирования для представления данных, удовлетворяющего всем вышеперечисленным параметрам.

2. Постановка задачи

Рассмотрим пример задачи: пусть имеется база данных клиентов некоторой крупной корпорации, причём число записей в ней достаточно велико. На каждого клиента, в базе данных, помимо очевидных полей таких как ФИО, возраст и т. д., содержится сложный набор информационных полей, по значениям которых клиентов требуется группировать, сортировать, искать и т. д., в общем, производить все основные операции с наборами данных. Например для отдела маркетинга этими полями могут быть предпочтения клиентов, отзывы о продуктах нашей компании, лояльность, частота использования услуг, оказываемых компанией, и т. д. Всё вышеперечисленное может помочь для анализа целевой аудитории, планирования промо-акций, рекламы, группировки клиентов по различным критериям и т. д. Требуется найти оптимальную структуру данных для эффективного представления базы данных, требующего как можно меньших затрат оперативной памяти и обеспечивающей быстрое выполнение всех вышеперечисленных операций. Понятно, что данную конкретную задачу можно обобщить. Ведь и в самом деле потребность в быстрой и эффективной обработке больших массивов данных есть практически во всех сферах человеческой деятельности, а время, особенно сегодня, как известно деньги и деньги зачастую большие.

3. Теоретические положения

3.1. Деревья.

3.1.1. Определение: дерево представляет собой конечное множество T одного или более узлов со следующими свойствами:

- существует один выделенный узел, а именно корень (root) данного дерева T .
- остальные узлы (за исключением корня) распределены среди $m \geq 0$ непересекающихся множеств T_1, \dots, T_m , и каждое из этих множеств, в свою очередь является деревом. Деревья T_1, \dots, T_m являются поддеревьями данного корня.

3.1.2. Бинарные деревья.

3.1.2.1. Определение: бинарным деревом называется дерево, каждая вершина которого имеет не более двух поддеревьев.

3.1.2.2. Причины популярности.

3.1.2.3. Основные операции: внесение, удаление, поиск, сортировка с помощью ДДП.

3.1.3. Сбалансированные деревья.

3.1.3.1. Абсолютно сбалансированные деревья.

3.1.3.1.1. Определение: дерево называется абсолютно сбалансированным, тогда и только тогда, когда невозможно построить дерево меньшей высоты, содержащие такое же число вершин.

3.1.3.1.2. Причина, по которой не используем: сложность реализации.

3.1.3.2. AVL деревья.

3.1.3.2.1. Определение: дерево является AVL сбалансированным тогда и только тогда, когда модуль разности высот его левого и правого поддеревьев равен нулю или единице.

3.1.3.2.2. Высота AVL сбалансированного дерева не более чем на 45% больше высоты абсолютно сбалансированного дерева.

3.1.3.2.3. Основные операции $O(\log(n))$.

3.2. Кодирование.

4. Решение задачи

Проанализировав возможные структуры данных, можно сделать вывод, что для реализации задачи более всего подходят бинарные деревья. Действительно, бинарные деревья, с одной стороны, легко реализуются, а с другой обладают достаточной скоростью выполнения требуемых операций (внесение, удаление, поиск, сортировка), большинство из которых работает за время, пропорциональное высоте дерева. Однако, использование простых бинарных деревьев всё же недопустимо, т. к. мы заранее ни чего не можем сказать о высоте дерева, которое получим после внесения всей интересующей информации. В худшем случае всё наше дерево вырождается в список, основные операции с которым работают уже за время, пропорциональное его длине, т. е. за n . Для того чтобы избежать этого логично было бы использовать так называемые сбалансированные деревья – деревья высота которых всегда пропорционально $\log(n)$. В лучшем случае она и будет равна $\log(n)$ если мы решим использовать абсолютно сбалансированные деревья. Однако, этот подход содержит один большой недостаток. Нам необходимо, чтобы операции внесения элемента в дерево и удаления элемента из него работали за время, пропорциональное высоте дерева. Операции же перестройки абсолютно сбалансированного дерева или очень сложны или не удовлетворяют критерию скорости, что делает вариант использования этого вида деревьев не приемлемым. Самым лучшим вариантом, на взгляд автора, являются AVL-деревья. Они лишь немного хуже абсолютно сбалансированных, но при этом намного легче в реализации.

Таким образом, мы разобрались со структурой данных для реализации задачи. Осталось определить то, что мы будем хранить в узлах нашего дерева. Для скорости выполнения многих операций логично было бы хранить там не сами данные а указатели на них и код, который бы точно описывал все важные с точки зрения реализации параметры каждой записи. Так многие операции с числами выполняются намного быстрее чем с другими типами данных, а тем более с их наборами, то мы получаем ещё один прирост производительности.

5. Выводы

В настоящей работе было рассмотрено решение некоторой прикладной задачи, использующее ряд стандартных понятий и алгоритмов. В процессе этого анализа достаточно убедительно показано, какой мощный и эффективный инструмент рассмотрен в данной работе. В современном обществе рассмотренные темы имеют достаточно обширное применение. Примеров доказывающих это можно привести великое множество: это практически весь огромный спектр задач, так или иначе связанных с базами данных.

Список литературы

1. Кнут Д.Э. Искусство программирования. – М.:Вильямс, 2000
2. Вирт Н. Алгоритмы и Структуры данных. – М.:Мир, 1989
3. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. – М.: Вильямс, 2005
4. Материалы сайта <http://ru.wikipedia>

СИСТЕМА УПРАВЛЕНИЯ ОБРАЗОВАТЕЛЬНЫМИ РЕСУРСАМИ ЭЛЕКТРОННОЙ БИБЛИОТЕКИ АЛТГТУ

Василенко А.Н. – студент, Андреева А.Ю. – к.ф.-м.н., доцент
Алтайский государственный технический университет (г. Барнаул)

В Алтайском государственном техническом университете имеется огромный банк электронной литературы, как вузовских, так и не вузовских авторов, который используется в процессе обучения, причем для каждого факультета, для каждой специальности имеется свой набор литературы специфический для каждого семестра. Получение этой литературы занимает достаточно много времени, а для студентов дистанционного и заочного обучения, зачастую серьезно затрудняет образовательный процесс. Также это усложняет процесс контроля использования литературы студентами и предоставление учащимся учебных материалов преподавателями.

В связи с этим, актуальной становится проблема создания электронной библиотеки учебных материалов АлтГТУ, которая объединит в себе весь имеющийся банк материалов и позволит получить удаленный доступ к ним, а также, позволит осуществлять сбор статистической информации и контроль над оборотом материалов.

Целью создания электронной библиотеки является повышение качества образовательных услуг за счет доступности обучающих материалов и дополнительной литературы, используемой в процессе обучения.

В рамках данной работы были поставлены следующие задачи:

- разработать структуру базы данных для хранения используемой информации;
- разработать способ доступа пользователя электронной библиотеки;
- разработать способ администрирования для учета, модификации, поиска, а также наполнения базы данных информацией;
- разработать модуль статистики, отражающей оборот и использование учебных материалов.

В результате проведенной работы было разработано следующее:

- база данных для хранения информации;
- web-интерфейс пользователя электронной библиотеки, структура которого схожа со структурой учебных планов вуза. Так как основное назначение web-интерфейса пользователя – предоставление качественного и удобного доступа студентам к материалам, то структура web-интерфейса пользователя должна быть ориентирована, в первую очередь, на студентов. Именно поэтому поиск материалов осуществляется по известным студенту параметрам, например, факультет, специальность, дисциплина, преподаватель и прочее. Такой подход является естественным для студента и не вызывает затруднений при поиске материалов. Также, в пользовательском web-интерфейсе отображается различная информация о материале, такая как формат документа, размер, количество страниц и прочее. Помимо этого имеются: фильтры различного типа, для фильтрации списка материалов по

различным атрибутам; постраничная навигация, позволяющая просматривать список материалов отдельными блоками по определенному количеству записей; сортировка списка по различным полям, по возрастанию и убыванию значений.

- Web-интерфейс администратора, предоставляющий удобный доступ к системе управления материалами. Содержит страницы редактирования и просмотра информации. Для удобства пользователя на страницах просмотра имеется сортировка, фильтр данных, постраничная навигация. Количество материалов достаточно велико и при добавлении приходится указывать множество дополнительной информации. Поэтому разработанная панель управления, должна максимально облегчить работу администратора. Для этого на странице редактирования материалов, возле каждого поля выбора имеется ссылка на соответствующий справочник, позволяющая без потери ранее введенных данных о добавляемом материале, дополнить базу данных справочной информацией, после чего продолжить добавление материала. Помимо этого, имеется поля для множественного выбора авторов и специальностей, позволяющие добавить сразу несколько элементов. Для удобства администратора, осуществляющего наполнение базы данных, таблица, содержащая информацию о специальностях, дисциплинах, семестрах и количестве студентов оснащена соответствующими полями, позволяющими изменять необходимые данные за два клика мыши.
- Панель статистики, обеспечивающая полный учет над использованием материалов электронной библиотеки преподавателями и студентами позволяет, во-первых, оценить деятельность преподавателей и активность студентов в данном направлении, во-вторых, дает возможность быстро выявлять проблемные разделы сайта, в-третьих, позволяет оценить качество электронной библиотеки в целом, исходя из количества посещаемости и востребованности материалов. Для большей наглядности, статистические данные представлены как в табличном виде, так и в графическом, в виде различного рода графиков и диаграмм.

Выполнение этих задач позволило:

- повысить доступность информации о дисциплинах и материалах, за счет создания эргономичного интерфейса пользователя, соответствующего структуре учебных планов и специальностей;
- повысить контроль над использованием литературы студентами и преподавателями;
- предоставить широкие возможности по поиску информации для студентов и преподавателей.

Таким образом, являясь одним из основных компонентов информационной среды вуза, электронная библиотека должна обеспечивать весь контингент учащихся полным и качественным учебным материалом, по каждой образовательной услуге.

ИСПОЛЬЗОВАНИЕ ПРОЕКТ ЭКСПЕРТ ДЛЯ ФИНАНСОВОГО МОДЕЛИРОВАНИЯ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ

Мишина В.А – аспирант, Астахова А.В. – к.э.н., профессор кафедры ПИЮ
Алтайская академия экономики и права (г. Барнаул)

Ввиду сложности, динамичности и неполной детерминированности производственно-экономических и финансовых процессов на предприятиях, как объектах управления, задача финансового планирования не имеет однозначного решения. Другими словами, невозможно аналитическими методами, то есть с использованием традиционных вычислений, строго доказать и получить единственный оптимальный вариант финансового плана на рассматриваемый период.

Ответ на этот вопрос может быть получен лишь в результате эксперимента. Для этого необходим инструмент, с помощью которого исследователь может не только задавать различные вопросы, но и получать ответы.

Одним из таких инструментов, хорошо дополняющим возможности аналитических методов и в то же время являющимся значительно менее трудоемким и не сопряженным с большим риском по сравнению с натуральными экспериментами, является имитационное моделирование.

Имитационная модель, используемая в качестве инструмента в процессе принятия решений по оценке и выбору вариантов инвестиций, отражая основные зависимости между существенными показателями деятельности предприятия, позволяет:

- осуществлять предварительную поверку новых стратегий планирования и правил принятия решений перед проведением производственного эксперимента, соответствующего действующей системе финансового планирования;
- изучать воздействие на функционирование производственной системы некоторых информационных и организационных изменений, а также изменений во внешней среде;
- получать информацию о функционировании объекта управления и о прогнозе его состояния в последующих плановых периодах – для принятия управленческих решений;
- детально наблюдать имитируемую систему с целью выявления наиболее существенных переменных, их взаимодействия с другими переменными, а также с целью обучения и разработки предложений по совершенствованию производственно-экономической и финансовой системы предприятия.

При моделировании потоков денежных средств в Project Expert используется косвенный метод расчета величины денежного потока, основанный на анализе статей баланса и отчета о финансовых результатах и позволяющий установить взаимосвязь, с одной стороны, между разными видами деятельности предприятия, с другой, – между реализацией и денежной выручкой на рассматриваемом отрезке времени. Моделирование движения денежных средств по направлениям деятельности позволяет пользователю модели увидеть, где конкретно овеществлена прибыль компании или куда инвестированы «живые» деньги.

Важной особенностью Project Expert является ориентация на принятие решений в условиях неполной определенности информации в будущих плановых периодах. Это означает, что в процессе расчетов при моделировании учитываются трудно формализуемые и трудно прогнозируемые факторы, например, уровень инфляции; коэффициенты обменного курса валют; ставки налогов; цены; планируемые объемы сбыта и пр. В этой связи для разработки стратегического плана деятельности предприятия, в том числе, планирования инвестиций и последующего анализа эффективности проекта используется сценарный подход, предусматривающий построение альтернативных моделей и проведение соответствующих расчетов на основе информации, отображающей различные варианты развития проекта. В результате эксперт может «проигрывать» различные варианты стратегий развития моделируемой системы и принимать обоснованные управленческие решения, направленные на достижение целей развития предприятия.

Для имитации производственно-экономических процессов и потоков денежных средств, в структуру программного комплекса Project Expert включен блок моделирования, состоящий из нескольких модулей. К ним относятся модули информационного описания моделируемой предметной области (модуль описания макроэкономического окружения; модуль описания компании; модуль формирования инвестиционного плана проекта; модуль моделирования операционного плана компании) и собственного модуль имитационного моделирования.

Так, информационное описание макроэкономического окружения позволяет осуществлять:

- выбор валют для расчетов на внутреннем и внешнем рынках; прогноз обменного курса;
- моделирование налогового режима;

- моделирование сценариев инфляции по различным статьям поступлений и выплат проекта.

Модуль информационного описания компании, реализующей проект инвестиций, предусматривает:

- моделирование текущего состояния компании; формирование активов и пассивов;
- формирование номенклатуры продукции и/или услуг;
- задание метода списания затрат для бухгалтерского учета (по среднему, Fifo или Lifo).

Модуль непосредственного формирования инвестиционного плана проекта предполагает формирование.

- сетевого графика реализации проекта;
- соответствующего графику календарного плана выполнения работ и схемы взаимосвязей стадий проекта;
- номенклатуры и объема ресурсов для реализации проекта;
- затрат (по видам затрат);
- условий оплаты за ресурсы;
- вновь создаваемых активов.

После построения финансовой модели компании с учетом вышеизложенных требований работа с Project Expert может быть представлена в виде следующих основных этапов:

- определение потребности в финансировании;
- разработка стратегии финансирования;
- анализ финансовых результатов;
- формирование и печать отчета;
- ввод и анализ данных о текущем состоянии проекта в процессе его реализации.

Для определения потребности в финансировании (рис.1) следует произвести предварительные расчеты по проекту: определить эффективность проекта без учета стоимости капитала, а также объем денежных средств, необходимый и достаточный для покрытия дефицита капитала по периодам планирования. Расчеты показателей производятся при этом с выбранным для каждого периода шагом моделирования.

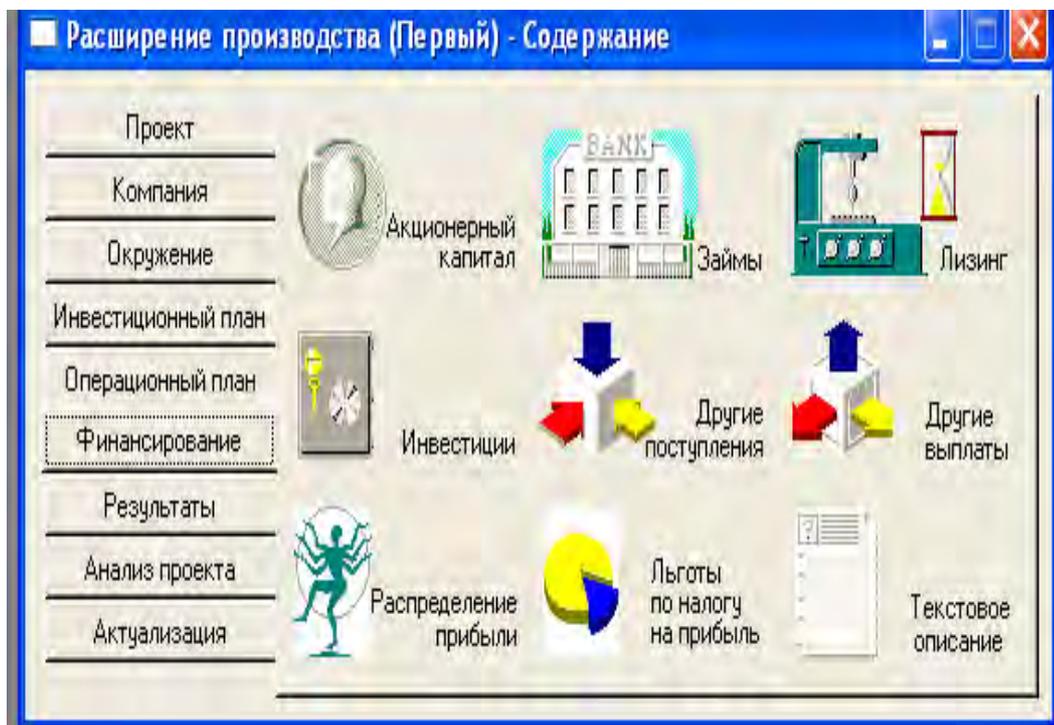


Рис.1. Меню для работы с модулем «Финансирование» в системе Project Expert

После определения потребности в финансировании при разработке плана инвестиций пользователь модели может выбрать один из нескольких способов финансирования (посредством привлечения акционерного капитала; заемных денежных средств; финансовой аренды и др.). В процессе разработки (выбора) проекта пользователь имеет возможность промоделировать объем и периодичность выплачиваемых дивидендов, а также стратегию использования свободных денежных средств, например, размещение денежных средств на депозит в коммерческом банке или приобретение акций сторонних предприятий.

Результаты вариантов моделирования инвестиций оцениваются в системе Project Expert специальным модулем «Результаты». В итоге моделирования система генерирует финансовые документы, соответствующие международным стандартам бухгалтерского учета (International Accounting Standards – IAS): прогноз движения денежных средств (Cash Flow); отчет о прибылях и убытках; балансовая ведомость; отчет об использовании прибыли. Если полученные оценочные показатели, содержащиеся в названных документах неприемлемы, пользователь системы изменяет принятые ранее плановые решения, и процесс моделирования возобновляется.

Описанный подход к моделированию проходит в настоящее время апробацию на нескольких предприятиях общественного питания г. Барнаула.