

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение
Высшего профессионального образования
Алтайский государственный технический университет
им. И.И.Ползунова



НАУКА И МОЛОДЕЖЬ – 2014

**XI Всероссийская научно-техническая конференция
студентов, аспирантов и молодых ученых**

СЕКЦИЯ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

подсекция

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И
АВТОМАТИЗИРОВАННЫХ СИСТЕМ**

Барнаул – 2014

УДК 004

XI Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых "Наука и молодежь – 2014". Секция «Информационные технологии». Подсекция «Программное обеспечение вычислительной техники и автоматизированных систем». / Алт. гос. техн. ун-т им. И.И.Ползунова. – Барнаул: изд-во АлтГТУ, 2014. – 89 с.

В сборнике представлены работы научно-технической конференции студентов, аспирантов и молодых ученых, проходившей 24 апреля 2014 г.

Редакционная коллегия сборника:

Кантор С.А., заведующий кафедрой «Прикладная математика» АлтГТУ – руководитель секции, Крючкова Е.Н., профессор, зам. зав. кафедрой ПМ, Сорокин А.В., доцент каф. ПМ, ответственный за НИРС на кафедре ПМ

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

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

Компьютерная верстка: Сорокин А.В.

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

СОДЕРЖАНИЕ

Вольф В.Я., Астахова А.В. Разработка программного обеспечения для скрининга заболеваний органа зрения.....	5
Ворона О.И., Леонов С.Л., Коновалов В.К. Обработка данных компьютерных томограмм	8
Гузев Д.С., Боровцов Е.Г. Инструментарий управления доменной средой Samba4 для операционной системы GNU/Linux	10
Есипенко С.П., Крючкова Е.Н. Проблема скелетной реконструкции произвольных биологических и механических объектов в системах компьютерного зрения.....	11
Зимонина А.В., Астахова А.В. Разработка программного обеспечения формирования и исполнения договоров поставки (на материалах фирмы ООО «Алейрон»)	13
Казakov М.Г., Крючкова Е.Н. Адаптивность классификации изображений на основе автоматической генерации обучающей выборки	15
Колосовский М.А., Крючкова Е.Н. Алгоритм обнаружения автомобилей на основе вычитания фона для видеонаблюдения за пешеходными переходами.....	19
Коновалова К.В., Бубнова Н.Д., Ловцкая О.В. Разработка системы управления базой метаданных геоинформационного Интернет-портала ИВЭП СО РАН	21
Корякин Н.А., Третьяков И.В. Защита информации на мобильных устройствах под управление ОС Android в корпоративной среде.....	24
Костин К.В., Кантор С.А., Созутов А.Е. Разработка программного обеспечения для автоматизированного построения графика работы мастеров по обслуживанию заявок	26
Кронберг И.А., Старолетов С.М. Создание универсальной среды для проведения вебинаров.....	29
Лаптев М.А., Некрасов Д.В., Сорокин А.В. Использование генетического алгоритма для решения задачи теории расписаний.....	31
Овчинникова А.А., Бразовская Н.В. Разработка программного комплекса для обучения студентов экономических специальностей финансово-коммерческим операциям с помощью интерактивных методов	34
Еремин Я.Б., Путинцев Р.В., Боровцов Е.Г. Разработка сетевого универсального органайзера студента.....	37
Зыков Р.В., Поддубный В.И. Разработка программного обеспечения для системы определения точных координат движущегося объекта с применением спутниковых радионавигационных систем	39
Савченко В.В., Крючкова Е.Н. Алгоритм семантического поиска в больших текстовых коллекциях	42

Смирнова А.Ю., Крючкова Е.Н., Тамплон А.В. Разработка программного обеспечения автоматизированной системы судейства и проведения соревнований в авиамodelьном спорте	46
Бобб А.Я., Хатников А.С., Троицкий В.С. Средство профилирования для анализа проблем производительности в АБС RS-Bank	49
Уваров Д.А., Силин Д. И., Крючкова Е.Н. Инструментальная система эффективной разработки интерактивного обучающего курса	49
Шамин А.С., Кошелев К.Б. Численное моделирование движения подземных вод с использованием параллельных вычислений.....	53
Гуленко К.А., Старолетов С.М. Проектирование и реализация каркаса для создания электронных образовательных материалов	55
Чураков М., Старолетов С.М. Лицензии свободного программного обеспечения.....	56
Краюшкин М.С., Старолетов С.М. Разработка утилиты для учета и анализа http трафика под ОС Linux	58
Стручев Р.К., Старолетов С.М. Комплекс программ для создания, просмотра и оценки анимированных изображений в формате gif.....	59
Ненайденко А.С., Боровцов Е.Г. Методы Data Mining как современные подходы для решения статистических задач прогнозирования	63
Смолянинов А.Ю., Крайванова В.А. Консольный клиент к музыкальному сайту.....	66
Бобков К.А., Крючкова Е.Н. Распределенный комплекс управления устройствами в умном доме.....	68
Шитов Д.А., Старолетов С.М. Концепция построения искусственного интеллекта на основе абстрактного мышления и ассоциативной структуры мозга человека	70
Кааль С.В., Перепелкин Е.А. Программная реализация модели баланса графа ссылок.....	72
Турубарова Е.А., Крайванова В.А. Плагин визуализации данных для фреймворка symfony 2	76
Гриднева А.Е., Корягина С.И., Моисеева А.А., Шевелев И.В., Яковенко Р.А., Рогозин К.И., Сорокин А.В. Дистанционная поддержка курса.....	78
Никитин А.А., Андреева А.Ю. Проблема определения параметров частиц по монохромному изображению процесса сварки	82
Головин М.А., Крючкова Е.Н. Контент-зависимое изменение размеров изображения	86

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СКРИНИНГА ЗАБОЛЕВАНИЙ ОРГАНА ЗРЕНИЯ

Вольф В.Я. – студент, Астахова А.В. – к.э.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Зрительная система является весьма чувствительным и одним из важнейших анализаторов человека, который передает ему около 80% информации об окружающей действительности. Глаз обеспечивает получение представления о движении предмета, о его освещенности, цвете, форме, величине, а также расстоянии, на котором предмет находится от человека. Глаза функционируют весь период нашего бодрствования, не переходя в состояние зрительного покоя. В итоге в многообразной трудовой деятельности людей, в выполнении ими многочисленных, часто – весьма тонких работ, глазам принадлежит первостепенная роль.

К сожалению, зрительная система человека, как и любая другая, дает сбои. И эти сбои зачастую связаны с постоянными зрительными нагрузками, будь то чтение книги, просмотр телевизора, работа на компьютере или другая работа, требующая напряжения зрения. Современная статистика показывает, что патология зрительной системы – далеко не редкое явление. И эта статистика свидетельствует о том, что патология органа зрения неуклонно растет с каждым годом. Однако любое заболевание не наступает внезапно, оно проходит так называемый функциональный этап развития, который является одним из важнейших, поскольку считается переходным от нормы к патологии. Именно, на данном этапе своевременное выявление имеющихся функциональных изменений, их профилактика и устранение могут позволить снизить уровень заболеваемости. Однако существует ряд проблем, препятствующих этому. В Алтайском крае основными из них в настоящее время являются:

- низкая доступность специализированной офтальмологической помощи для населения;
- сложности организационного характера при записи на прием к врачу-офтальмологу;
- недостаточная оснащенность современным, высокоэффективным диагностическим и лечебным оборудованием;
- отсутствие информационной базы данных, позволяющей оперативно получить необходимую информацию о проведении обследований, о заболеваемости болезнями органа зрения, о профилактических лечебных мероприятиях, проводимых среди населения.

Система удаленной скрининг диагностики зрения способна решить названные проблемы. Такая диагностика позволяет на безвозмездной основе упростить предварительное выявление у пациентов отклонений в состоянии органа зрения от нормы, а также провести централизованный сбор, статистическую обработку и анализ результатов данных, полученных при проведении скрининговых обследований населения. Основная цель удаленного скрининга – раннее выявление заболеваний на основе оптометрических и офтальмологических обследований в интерактивном режиме.

Над компьютерными программами дистанционной диагностики состояния органа зрения работали многие организации, в том числе, Новосибирский филиал ФГУ МНТК «Микрохирургия глаза им. академика С.Н. Федорова» [1]. Следует отметить, что все эти проекты разработаны таким образом, что при проведении исследования используется лишь одна известная, редко две методики проверки зрительных функций, которых оказывается недостаточно для точной диагностики наличия или отсутствия патологических изменений органа зрения. Кроме того, упомянутые работы, помимо проекта Новосибирского филиала ФГУ МНТК Микрохирургия глаза им. академика С.Н. Федорова, не предусматривают

возможность формирования и последующего анализа результатов исследования с использованием базы данных (БД) пациентов.

Проблема проведения массовых офтальмологических профилактических осмотров, связанная с низкой доступностью специализированной помощи населению, вынуждает специалистов искать пути её решения посредством решения задач по разработке различных компьютерных программ автоматизации офтальмологического скрининга взрослого населения. Разработка и создание программы автоматизированного офтальмологического скрининга, является задачей составляющей содержание данного проекта, который выполняется по заявке ГБОУ ВПО АГМУ Минздрава России и ФГБУ «НИИ КППЗ» СО РАМН, чем и обусловлена актуальность работы.

Цель проекта: разработать и создать систему автоматизированного скрининга заболеваний органа зрения, для своевременного выявления и профилактики глазных болезней у взрослого населения Алтайского края. Данная цель может быть разбита на следующие подцели:

- упрощение методики проведения исследования основных зрительных функций у взрослых с целью выявления отклонений от нормы;
- создание автоматизированной системы, позволяющей проводить массовые диагностические исследования взрослого населения, без участия врача-офтальмолога;
- создание автоматизированной системы, способной накопить и систематизировать данные, полученные в результате проведенных исследований, с целью последующего статистического анализа для разработки профилактических мероприятий по предупреждению глазных заболеваний.

Задачи работы:

- разработка проекта программного обеспечения для достижения поставленной цели и ее подцелей в двух режимах тестирования пациентов:
 - на локальном компьютере;
 - в сетевом варианте с использованием интернет-технологий;
- отладка проекта на подмножестве реальных данных заказчика;
- разработка технологической документации по эксплуатации системы.

Результатом работы является система автоматизированного скрининга заболеваний органа зрения, которая имеет клиент-серверную сетевую архитектуру и позволяет провести ряд оптометрических и офтальмологических обследований в интерактивном режиме посредством сети Интернет (см. рисунок 1).

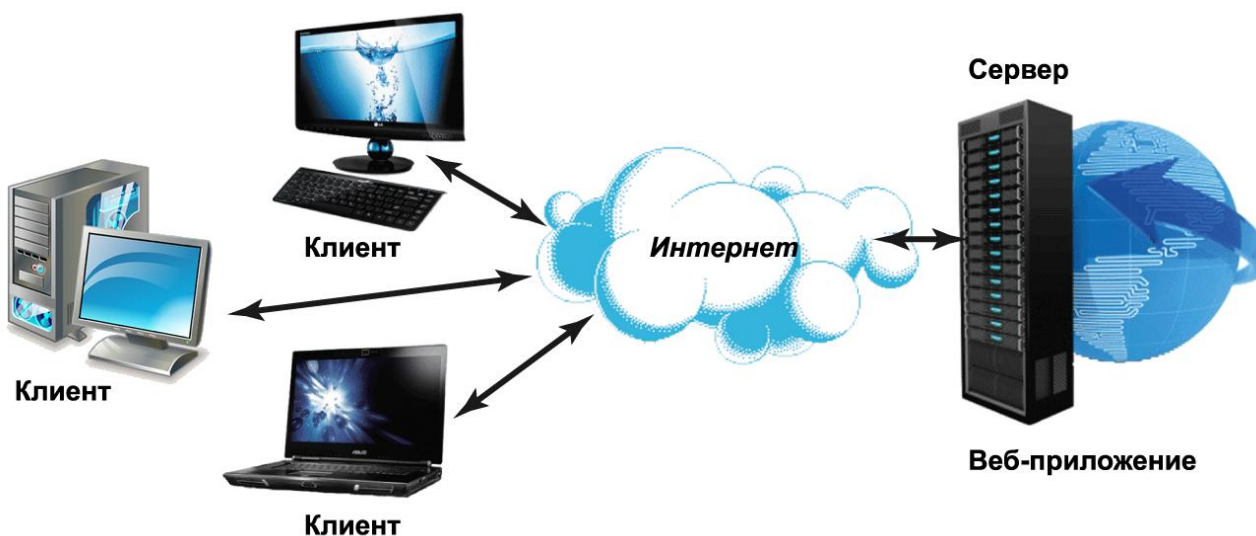


Рисунок 1 – Клиент-серверная сетевая архитектура

В системе реализуются следующие возможности.

Тест на исследование остроты зрения. В тесте в качестве опто типов применяются кольца Ландольта. Угловой размер и количество опто типов, показываемых пациенту, а также количество допустимых ошибок строго соответствует мировому стандарту при исследовании остроты зрения.

Дуохромный тест для определения состояния рефракции на момент проведения обследования, ее сдвиг в сторону гиперметропии (дальнозоркости) или миопии (близорукости) [2].

Тест Амслера, который позволяет косвенно исключить или выявить патологию макулярной зоны — участка сетчатки, ответственного за состояние центрального зрения [2].

Тест на исследование полей зрения может выявить значительные изменения в поле зрения, связанные с заболеваниями глаз, такими как хориоретинит, отслойка сетчатки; аномалиями зрительного нерва, например при далеко зашедшей глаукоме, или внутричерепными заболеваниями, связанными с повышением внутричерепного давления: опухоль мозга, ишемическое повреждение или кровоизлияние в него.[2].

Тест на глаукому можно рассматривать как продолжение теста на исследование полей зрения. Он позволяет получить результаты, предназначенные для построения надежного и быстрого диагностического алгоритма, позволяющего выявить глаукому на ранней стадии.

Стоит отметить, что результаты удаленного скрининга являются лишь предварительными: при выявлении патологии органа зрения пациенту следует незамедлительно обратиться к врачу офтальмологу для установления более точного диагноза.

Помимо проведения оптометрических и офтальмологических обследований в интерактивном режиме, система предполагает автоматическое занесение в БД и хранение в ней персональных данных пациента (ФИО, дата рождения, адрес регистрации, номер СНИЛС), а также сведений обо всех проведенных обследованиях (дата обследования, полученные результаты). Таким образом, в базу данных попадают результаты не только текущего обследования, но и предыдущих обследований, если таковые имелись. Кроме этого, пациенту предоставляется возможность получения на руки результатов обследования и записи на прием к врачу-офтальмологу. Кроме того, врачу офтальмологу – на основании данных, хранимых в БД, предоставляется возможность формирования отчетов и диаграмм, отражающих зависимость уровня заболеваемости различными глазными болезнями от возраста, района проживания и т.д.

В результате создания системы должны быть улучшены значения следующих основных показателей, характеризующих состояние офтальмологической помощи в Алтайском крае:

- улучшена доступность специализированной офтальмологической помощи населению Алтайского края;
- увеличен процент охвата населения профилактическими осмотрами на предмет раннего выявления зрительных патологий;
- уменьшено время, затрачиваемое на обследование значительной части населения для предварительного выявления отклонений зрительной системы от нормы;
- время сбора и обработки статистических данных, характеризующих патологическую пораженность населения Алтайского края болезнями органа зрения;
- профилактические мероприятия, разработанные на основании анализа полученных данных обследования, позволят снизить заболеваемость взрослого населения болезнями органа зрения.

Список литературы

1. Новосибирский филиал ФГУ МНТК «Микрохирургия глаза им. академика С.Н. Федорова» [Электронный ресурс]. – Режим доступа: <http://www.mntk.nsk.ru>
2. Копаева В.Г. (ред.) Глазные болезни. – М.: Издательство «Медицина», 2002. – 560 с.: ил.

ОБРАБОТКА ДАННЫХ КОМПЬЮТЕРНЫХ ТОМОГРАММ

Ворона О.И. – студент, Леонов С.Л. – д.т.н., профессор,
Коновалов В.К. – д.м.н., профессор*
Алтайский государственный технический университет (г. Барнаул)
Алтайский государственный медицинский университет*(г. Барнаул)

Большинство заболеваний легких, особенно рак, туберкулез и пневмония, относятся к группе социально значимых. За четыре года количество ежегодно регистрируемых новых случаев раковых заболеваний в мире выросло почти на полтора миллиона, с 12,7 миллиона в 2008 году до 14,1 миллиона в 2012 году. Первое место как среди вновь диагностируемых злокачественных новообразований, так и среди причин смерти от онкологии занимает рак легких [1]. Чаще всего в мире в 2012 году диагностировали рак легких (1,8 миллиона случаев, 13 процентов от общего числа).

Существует несколько способов исследования заболеваний легких и один из них это компьютерная томография. Компьютерная томография в настоящее время является методом выбора в диагностике объемных процессов средостения. Она позволяет выявить на ранних стадиях патологическое образование, правильно оценить его размеры и контуры, изучить его структуру, установить степень инвазии на окружающие органы и ткани, позволяет уточнить распространенность опухолевого процесса, тем самым определить его стадию. Денситометрическая характеристика плотности объемных образований при компьютерной томографии позволяет отличить новообразования от кист, с высокой точностью установить гистологическое строение жировых опухолей [2].

Сейчас последовательность действий, выглядит следующим образом [3]:

Врач-рентгенолог работает с серией срезов, полученных с помощью компьютерного томографа. На каждом срезе подходящем для анализа выбирается область интереса, которая представляет собой прямоугольную область, она должна быть не менее 6 пикселей в любом разрешении и не содержать в себе что-либо кроме тканей самого шаровидного образования.

После выделения области интереса, значения яркости пикселей в шкале Хаунсфилда [4] сохраняются в файл для дальнейшей обработки. Шкала единиц Хаунсфилда— шкала линейного ослабления излучения по отношению к дистиллированной воде, рентгеновская плотность которой была принята за 0 HU (при стандартных давлении и температуре). Для материала X с линейным коэффициентом ослабления μ_X , величина HU определяется по формуле

$$\frac{\mu_X - \mu_{water}}{\mu_{water} - \mu_{air}} \times 1000$$

где μ_{water} и μ_{air} — линейные коэффициенты ослабления для воды и воздуха при стандартных условиях. Таким образом, одна единица Хаунсфилда соответствует 0,1 % разницы в ослаблении излучения между водой и воздухом, или приблизительно 0,1 % коэффициента ослабления воды, так как коэффициент ослабления воздуха практически равен нулю.

Из полученных значений получают набор параметров, по которому возможно проводится дифференциальная диагностика в программе «Конструктор нейронных сетей», которая на основании существующей базы пациентов и сконструированной по ней сети, выдает вероятность того или иного заболевания.

На основании полученных данных врач-рентгенолог принимает решение о диагнозе.

Когда диагноз становится верифицированным, данные пациента добавляется в рабочую базу пациентов.

Файлы компьютерных томограмм имеют специальный формат dicom [5]. Dicom - отраслевой стандарт создания, хранения, передачи и визуализации медицинских изображений и документов обследованных пациентов. Файлы типа dicom имеют четкую структуру, они состоят из заголовка и тела файла. Файлы, с которыми оперирует томограф, имеют тип CT- image, изображение черно-белого спектра. Значение яркости пикселя, такого изображения, соответствует шкале Хаунсфилда и изменяется от -1024 до +3071, но обычный компьютерный монитор способен отображать до 256 оттенков серого цвета. В связи со значительной шириной шкалы Хаунсфилда и неспособностью существующих мониторов отразить весь её диапазон в черно-белом спектре, используется программный перерасчет серого градиента в зависимости от интересующего интервала шкалы.

Программное обеспечение написано на языке Java. Для того, чтобы программа работала с dicom CT- image файлом как с файлом обычного изображения используется плагин dcm4che2. Dcm4che2 - это открытое программное обеспечение на языке Java, для работы в сфере здравоохранения, оно является программной реализацией стандарта dicom, а потому содержит все необходимые инструменты для работы с ним.

Программа позволяет:

1. Загружать и просматривать последовательность dicom- изображений.
2. Масштабировать изображения с использованием метода ближайшего соседа или билинейного сглаживания.
3. Выделять область интереса и трансформировать ее на различных слоях.
4. Просматривать значение плотности указанного пикселя и информацию о выделенной области.
5. Сохранять интересующую последовательность в выбранную папку в виде набора txt файлов.
6. Получать значение параметров на основании плотностей
7. Отправлять параметры на сервер с программой «Конструктор нейронных сетей»
8. Получать ответ от сервера и выдавать результат врачу

На разработанное программное обеспечение получено свидетельство о государственной регистрации. В дальнейшем планируется доработка программного обеспечения с целью расчета фрактальных параметров томограмм.

Список литературы

3. Рак: основные цифры и факты [Электронный ресурс]: «Всемирная организация здравоохранения». – Электрон. текст. дан. – 2013. – Режим доступа: <http://www.who.int/cancer/about/facts/ru/>
4. Власов, П.В. Комплексная лучевая диагностика объемных образований средостения. Часть первая / П.В.Власов // Медицинская визуализация. – 2005. – №3. – С. 29-41.
5. Шкала Хаунсфилда [Электронный ресурс]: Википедия. – Электрон. текст. дан. – 2013. – Режим доступа: http://ru.wikipedia.org/wiki/Шкала_Хаунсфилда
6. Ворона, О. И. Получение и анализ данных компьютерных томограмм. / О.И.Ворона, С.Л.Леонов, В.К.Коновалов//X Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых "Наука и молодежь 2013". Секция «Информационные технологии». Подсекция «Программное обеспечение вычислительной техники и автоматизированных систем». / Алт. гос. техн. ун-т им. И.И. Ползунова. – Барнаул: изд-во АлтГТУ, 2013. – С.16-18.
7. Dicom Standard [Электронный ресурс]: Официальный сайт организации, разрабатывающей стандарт DICOM. – Электрон. текст. дан. – 2013. – Режим доступа: <http://medical.nema.org/>

ИНСТРУМЕНТАРИЙ УПРАВЛЕНИЯ ДОМЕННОЙ СРЕДОЙ SAMBA4 ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ GNU/LINUX

Гузеев Д.С. – студент, Боровцов Е.Г. – к.т.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Развитие информационных технологий повлекло за собой создание множества сервисов, обладающих необходимым для бизнеса функционалом. При этом на уровне предприятия все сервисы должны составлять единую информационную среду, тесно интегрируясь друг с другом. В роли основы такой инфраструктуры выступает доменная среда, связывающая разрозненные сервисы воедино.

В последние годы наметилась тенденция к использованию, в том числе и на государственном уровне, свободного программного обеспечения. Одним из вариантов построения доменной среды является создание ее на базе открытого программного обеспечения Samba версии 4. Открытые средства для управления доменной средой Samba4 в настоящее время слабо развиты, основным вариантом является использование Microsoft Windows RSAT. Данный инструментарий управления работает на ОС Microsoft Windows, что не позволяет полностью отказаться от использования проприетарного программного обеспечения.

Целью данной дипломной работы является разработка инструментария управления доменной средой Samba4 для операционной системы GNU/Linux с возможностью управления доменными пользователями и компьютерами, а так же управления групповыми политиками. Разработка ведется на языке C++ с использованием фреймворка Qt4.

Основными особенностями Samba4 являются:

- Свободное программное обеспечение с открытым исходным кодом;
- Встроенный DNS-сервер, интеграция с существующим DNS-сервером;
- Работа с протоколом Kerberos;
- Управление групповыми политиками;
- Возможности репликации с Microsoft Active Directory
- Возможности интеграции с приложениями, имеющими поддержку интеграции с Microsoft Active Directory

Доменная среда строится вокруг службы каталогов, являющейся центральным хранилищем информации о пользователях, компьютерах и политиках предприятия. Основным протоколом доступа к службе каталогов в настоящее время является LDAP (Lightweight Directory Access Protocol), получивший наибольшее распространение благодаря своей легкости при сохранении необходимого функционала. LDAP работает через стек протоколов TCP/IP.

Основные преимущества внедрения доменной среды:

- Единая точка аутентификации;
- Единая точка управления политиками;
- Повышенный уровень информационной безопасности;
- Интеграция с корпоративными приложениями и оборудованием;
- Единое хранилище конфигурации приложений.

Разрабатываемое программное обеспечение должно обладать следующими особенностями:

- Кроссплатформенность — ПО должно работать как на операционной системе GNU/Linux, так и на других операционных системах;
- Обеспечение необходимого функционала управления доменной средой;
- Интуитивно-понятный оконный графический интерфейс (GUI).

Функционал управления доменной средой включает в себя:

- Глобальное управление сервисом — запуск, остановка, перезапуск;
- Управление пользователями и компьютерами — добавление, редактирование, удаление, блокирование учетных записей пользователей;
- Предоставление информации о состоянии системы;
- Управление групповыми политиками.

В дальнейшем данный проект может быть расширен добавлением дополнительного функционала, к примеру, многопользовательского режима работы приложения с возможностью разграничения прав пользователей, отправкой e-mail при обнаружении неполадок в доменной среде.

Список литературы

1. Бланшет Ж. QT4 программирование GUI на C++. Второе издание / Ж. Бланшет, М. Саммерфилд — Кудиц-Пресс, 2008. — 738 с.
2. Введение в протокол облегченного доступа к каталогам [Электронный ресурс] – Режим доступа: <http://support.microsoft.com/kb/196455/ru>
3. Samba[Электронный ресурс] – Режим доступа: <https://wiki.samba.org/index.php/Samba>
4. Головач В.В. Дизайн пользовательского интерфейса / В.В. Головач – 2001. - 141с.

ПРОБЛЕМА СКЕЛЕТНОЙ РЕКОНСТРУКЦИИ ПРОИЗВОЛЬНЫХ БИОЛОГИЧЕСКИХ И МЕХАНИЧЕСКИХ ОБЪЕКТОВ В СИСТЕМАХ КОМПЬЮТЕРНОГО ЗРЕНИЯ

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

Одной из важнейших проблем современного компьютерного зрения является моделирование движения живых существ и, в частности, человека. С ней приходится сталкиваться при проектировании самых различных систем и устройств. На данный момент, в большинстве случаев обходятся грубыми методами и узкоспециализированными решениями. Но с развитием техники растет потребность в более точных и универсальных подходах к решению данной проблемы.

Очень часто, для эффективного изучения некоторого объекта или процесса достаточно выделить его ключевые характеристики и сфокусироваться на них. Такой характеристикой для большинства движущихся живых существ является скелет. Под скелетом здесь подразумевается упрощенная модель реального скелета, в которой кости представляются отрезками. Зная только расположение скелета объекта в различные моменты времени, можно достаточно точно определять, что делает объект, а также предсказывать его положение в пространстве.

Проблема скелетной реконструкции

Скелетная реконструкция изучается достаточно давно. Есть и коммерческие разработки[1,2]. Однако зачастую они слишком дороги и могут применяться лишь в специально оборудованных помещениях и с использованием специализированной техники. С другой стороны, с ростом вычислительных мощностей, появляется возможность, применяя сложные математические методы, использовать обычные камеры или относительно дешевые датчики глубины для анализа движения и реконструкции скелета. А это, в свою очередь, открывает широкие перспективы для развития таких направлений, как дополнительная

реальность, автоматическое управление автомобилем, видеонаблюдение и безопасность, компьютерная графика и анимация, игровые и развлекательные устройства и другие.

Несмотря на то, что уже разработаны достаточно перспективные подходы для решения проблемы скелетной реконструкции и анализа движения [3,4], их применение на практике остается под вопросом. Готовых реализаций и решений очень мало, причем многие из них анализируют только движение человека (например, [5]).

Прежде чем рассматривать возможные подходы скелетной реконструкции, необходимо сформулировать задачу. В общем виде, дана видеопоследовательность, на ней необходимо выделить объект, определить его структуру скелета, и для каждого кадра последовательности определить пространственную конфигурацию костей. Скелет представляет собой набор отрезков в пространстве. Отрезки могут соединяться только в конечных точках. Если рассматривать скелет в виде графа, в котором ребрами являются кости, а вершинами – точки их соединения, то он всегда связный. Кроме того, если запретить в данном графе циклы (это позволительно для живых существ), то в результате, скелет можно представить в виде дерева (обычно корню соответствует некоторый базовый сустав). Полученный скелет и будет решением задачи.

Конкретная постановка задачи зависит от типа входных данных (последовательность с одной камеры, с нескольких произвольных камер, с нескольких синхронизированных камер, с камеры и датчика глубины), а также способа оценки степени соответствия полученного скелета реальному скелету объекта.

Стоит отметить, что различные механические системы (роботы, строительные краны и другие) тоже могут моделироваться набором отрезков, и для них также можно решать задачу скелетной реконструкции.

Алгоритмы и методы решения

Если проанализировать современные подходы для скелетной реконструкции, то можно выделить несколько групп.

В первую входят решения, основанные на маркерах. Это наиболее простой подход, заключающийся в том, что к объекту прикрепляются специальные высококонтрастные пометки-маркеры, которые легко выделяются и идентифицируются на изображении. Обычно такие маркеры помещают на суставы или другие ключевые точки объекта. При использовании многокамерных конфигураций или датчиков глубины, очень просто отследить положение в пространстве отдельных суставов, и соответственно, восстановить конфигурацию костей объекта. Такие подходы применяются в современных коммерческих системах захвата движения (motion capture). Они обеспечивают хорошую точность, достаточно устойчивы и не требуют слишком сложных программных комплексов, однако для применения в обычных условиях не подходят. В некотором роде, к данным методам можно отнести решения, основанные на поиске лица [6] или других ключевых частей тела (кисти, ладони). Выделив область лица, можно предположить, где будет расположено туловище, затем руки и ноги. К сожалению, данные методы хоть и просты в реализации, слишком ограничены (человек должен смотреть в камеру) и для серьезного применения не подходят.

Вторая группа, развивает идеи первой, но использует более универсальные подсказки, такие как текстура, цвет, углы и ребра, шаблоны и другие. Эти подходы более универсальны, но все же, каждый из этих методов работает только при определенных условиях. Так, например, методы, основанные на выделении участков с кожей, перестают работать при сильном изменении освещения.

Третью группу составляют методы, основанные на вероятностных моделях. В них, конфигурация костей описывается вектором случайных величин (положения в пространстве корневого сустава, угловые смещения костей). Для них определяется распределение вероятности, которое пересчитывается каждый кадр с учетом данных выделенных из

текущего изображения. К сожалению, движение человека или других живых существ, является достаточно сложным процессом, поэтому пересчет распределения вероятности параметров модели выполняется с помощью специальных техник. В основном используется Particle Filtering [3] и ее модификации, Particle Swarm Optimization [4] и другие. На данный момент, вероятностные методы являются наиболее перспективными для скелетной реконструкции. Они могут применяться как для многокамерных конфигураций [4], так и для изображения с одной камеры [3].

Список литературы

1. Organic Motion [Электронный ресурс]: – Режим доступа: <http://www.organicmotion.com>.
2. iPi Soft [Электронный ресурс]: – Режим доступа: <http://ipisoft.com>.
3. Sidenbladh, H. Stochastic tracking of 3D human figures using 2D image motion [Текст] / H. Sidenbladh, M. Black, D. Fleet // ECCV 2000. – 2000. – С. 702-718.
4. John, V. Markerless human articulated tracking using hierarchical particle swarm optimization [Электронный ресурс] / V. John, E. Trucco, S. Ivecovic // Image and Vision Computing. – 2010. – Режим доступа: <http://dx.doi.org/10.1016/j.imavis.2010.03.008>.
5. Human Motion Imitation [Электронный ресурс]: – Режим доступа: <http://humim.org/tracker>.
6. Siddiqui, M. Robust real-time upper body limb detection and tracking [Электронный ресурс] / M. Siddiqui, G. Medioni // VSSN '06. – 2006. – Режим доступа: <http://iris.usc.edu/outlines/papers/2006/siddi-medioni-vssn06.pdf>.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ФОРМИРОВАНИЯ И ИСПОЛНЕНИЯ ДОГОВОРОВ ПОСТАВКИ (НА МАТЕРИАЛАХ ФИРМЫ ООО «АЛЕЙРОН»)

Зимонина А.В. – студент, Астахова А.В. – к.э.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Фирма ООО«Алейрон» занимается переработкой сырья и поставкой готового зерна заказчиком. Одним из базовых документов её деятельности является договор поставки. Основные работы по формированию договора и учётных документов принадлежат менеджеру по продажам. Менеджер по продажам принимает заказ, формирует договор согласовывает его с директором фирмы и представителями фирмы заказчика. После того как договор вступил в стадию исполнения, менеджер по продажам занимается формированием учётных документов. Формирование документов проводится на основе их шаблонов, ввод информации в которые осуществляется вручную, что приводит к временным затратам и замедляет процесс предоставления документов заказчику. Поиск информации по составленным договорам осуществляется либо просмотром отдельных документов, либо через обращение к бухгалтерии фирмы. Автоматизация данных задач позволит ускорить процессы формирования и исполнения договоров поставки.

Существующие решения сторонних разработчиков такие как DocsVision 3.6 [3] и система «Парус» [4] являются интегрированными в огромные программные комплексы, имеют большие базы данных и очень много избыточной информации, все это лишь путает пользователей. А так же в данных разработках отсутствует система гибкого поиска информации, что осложняет процесс анализа информации и дальнейшего планирования поставок.

Цель дипломного проектирования - разработка и создание программного обеспечения для формирования и исполнения договоров поставки предприятия ООО «Алейрон». Программа должна обладать данными функциями:

- Создание, хранение и доступ к справочникам предприятия.
- Формирования договора и спецификаций к нему.
- Формирование исполнительных документов (ТТН, счёт на оплату, счёт-фактура, акт исполнения работ).

Одной из основных целей разработки является предоставление пользователю удобной работы с данными хранящимися в программе. Для этого необходимо разработать систему гибкого и жёсткого поиска. Жёсткий поиск предоставляет информацию из базы данных на основе заранее заданных (составленных) запросов, например: Спецификации договора с указанным пользователем номером. Гибкий поиск позволит пользователю самостоятельно выбирать параметры поиска, и задавать их значения для фильтрации данных.

Для достижения данной цели необходимо было решить следующие задачи:

- Изучение выбранную предметную область.
- Проектирование и реализация базы данных.
- Разработка интуитивно понятного интерфейса и его реализация.
- Разработка и реализация системы жёсткого и гибкого поиска.
- Отладка проекта на подмножестве реальных данных заказчика.
- Разработка технологической документации по эксплуатации системы.

В результате работы была разработана база данных для формирования, хранения и исполнения договоров поставки, включающая в себя информацию справочников:

- Сырья:

Сырьё - необработанная крупа, определённой культуры которая определяет процесс её переработки;

- Зерна:

Зерно - крупа полученная в результате обработки и очистки крупы от примесей;

- ГОСТ-ов:

Стандартов обработки сырья, а так же определяют итоговый объём крупы после переработки;

- Контрагентов;

- Банков;

- Городов;

- Единиц измерения.

Информацию для ведения журнала договоров и формирования учётных документов:

- ТТН;

- Счёт-фактура;

- Акт;

- Счёт на оплату ;

А так же был реализован удобный интерфейс для внесения, просмотра и поиска необходимых данных.

Для удобства анализа информации и формирования отчётов был создан поиск по жёстким запросам, таким как:

1. Список договоров поставки указанного сырья за определённую дату
2. Список спецификаций относящихся к определённому договору
3. Итоговая стоимость переработки сырья по определённому договору
4. Итоговая стоимость (или объёмы сырья) переработки сырья поставленного определённым заказчиком – может использоваться для установления является ли заказчик постоянным клиентом, и возможно ли ему предоставление определённой скидки на стоимость переработки

5. Список договоров, контрагенты которых пользуются определённым банком
6. Список договоров заключённых в определённом городе – в дальнейшем развитии фирмы может использоваться как определение наиболее развитого и актуального для заказчиков филиала
7. Список ТТН, счёт-фактур, счетов на оплату относящихся к определённому договору
8. Итоговая стоимость переработанного сырья за определённый период
9. Список договоров по определённому зерну за период
10. История расценок на переработку установленных для определённого заказчика за период.

Из результатов поиска пользователю по двойному клику предоставляется возможность перехода к соответствующей таблице данных отсортированной в соответствии с выбранным запросом.

Кроме жёстких запросов была разработана логика формирования и выполнения гибких запросов, которая состоит из данных задач:

- Описание логики поиска
- Описание структуры выходных данных
- Выбор варианта хранения сформированного запроса
- Выполнение запроса и выгрузка результата для последующего анализа (в MS Excel).

Логика поиска определяется одним или несколькими параметрами(критериями). Каждый параметр представляет собой отношение, определяющее допустимые значения одного из отобранных полей, которыми ограничивается результат поиска. Если критериев несколько, то они соединятся между собой логическим условием «И». Это означает, что в результаты поиска будут включена информация, одновременно удовлетворяющая всем введённым параметрам(критериям).

В результате создания данного ПО должен ускориться процесс формирования и исполнения договоров поставки, предоставления необходимых документов заказчику, поиска необходимой информации.

Список литературы

1. Гражданский кодекс РФ часть вторая
2. Ананьев П.И., Кайгородова М.А. Основы баз данных. Учебное пособие/ АлтГТУ им. И.И. Ползунова. - Барнаул: 2010. -189 с. - ил.
3. Описание коммерческого ПО DocVision: <http://www.docsvision.com/kupit/dv-catalog/dv5.html>
4. Описание коммерческого ПО «Парус»: <http://www.parus.com/solutions/corporate/system/>

АДАПТИВНОСТЬ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ОБУЧАЮЩЕЙ ВЫБОРКИ

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

Введение

Задачи компьютерного восприятия изображений представленных в растровом виде являются вероятно наиболее сложными среди задач компьютерного зрения. При этом часть задач из этой области можно считать в той или иной степени разрешенными, например задачи локализации конкретного объекта или же задачи поиска экземпляров объекта. Другая

область задач – задачи классификации изображений остаются крайне тяжелыми для машинного решения [1]. Это связано, со многими факторами, в первую очередь с непониманием принципов работы восприятия человека. Так же задача усложняется возможными нечеткими трактовками принадлежности изображений классам и пересечением классов между собой, вплоть до включения.

В данной статье рассматривается подход к классификации изображений, опирающийся на представление изображений в виде набора многомерных векторов – дескрипторов локальных визуальных особенностей, обычно инвариантных к определенному классу трансформаций изображений [2]. При этом подразумевается наличие обучающей выборки изображений, соответствующим определенным классам. Наиболее эффективным на сегодняшний день считается применение этапа квантования, которое представляет весь спектр дескрипторов из обучающей выборки в виде набора так называемых визуальных слов (Bag Of Words), для чего используется кластеризация [3]. На этом этапе многомерные вектора фактически трансформируются в одномерные индексы. Общая схема классификации при этом может быть представлена в виде следующих шагов:

Извлечение дескрипторов локальных визуальных особенностей из набора обучающих изображений.

Кластеризация дескрипторов по заданному количеству визуальных слов.

Для каждого дескриптора производится поиск ближайшего центра кластера, после чего многомерный вектор заменяется на индекс этого центра (визуального слова).

Изображения обучающей выборки представляются в виде нормализованных гистограмм частоты появления визуальных слов в них.

Основываясь на представлении обучающей выборки классов как множества многомерных векторов – гистограмм, производится обучение классификаторов, например используя метод опорных векторов.

Анализируемое изображение аналогичным образом представляется в виде многомерного вектора – гистограммы.

Исходя из результата работы классификаторов, принимается решение об отнесении анализируемого изображения к тому или иному классу.

Особую важность в этом методе представляет обучающая выборка. С одной стороны, количество изображений в каждом классе должно быть достаточным для составления количественных характеристик класса, с другой стороны на связность изображений между собой накладываются определенные ограничения. Если изображения представляющие класс визуально разрозненны, то для классификатора проблематично подобрать достаточно четко описывающую их модель. Если же изображения чрезмерно похожи, по уменьшается охват классификатора, и увеличивается процент ложноотрицательных результатов работы классификатора. При этих сложностях, обычно обучающая выборка составляется в ручном режиме, что осложняет перенастройку системы под новые требования – в первую очередь при появлении новых классов.

Предлагаемый подход

С учетом объема изображений, доступных в Интернете, возможны попытки автоматизации процесса формирования обучающей выборки путей использования поисковых систем. Рассматриваемой поисковой системой, по причине наибольшей распространенности, была выбрана Google Custom Search. Традиционное количество обучающих изображений варьируется в диапазоне до 50, таким образом текущий лимит Google Custom Search на 100 первых элементов поисковой выдачи вполне укладывается в требуемые характеристики. Задача сбора обучающей выборки несколько упрощается тем,

что имеется возможность предварительно фильтровать поисковую выдачу, используя поверхностный анализ изображений проводимый поисковой системой. Так, например, доступна фильтрация по типу формата изображения, диапазону размеров и содержанию – в данном случае интерес представляют только фотографии (синтетические изображения, например так называемый clipart, довольно плохо поддаются анализу с использованием дескрипторов локальных особенностей).

Для оценки качества автоматически полученной обучающей выборки можно использовать метод confusion test, заключающийся в перекрестном анализе изображений представляющих классы с различными классификаторами. При этом для каждого изображения вычисляется наиболее близкий класс исходя из результата работы классификаторов, и составляется confusion matrix, в которой по горизонталям идут фактические изображения классов, а по вертикалям изображения, по которым была предсказана принадлежность к классу. В случае идеально работающих классификаторов confusion matrix должна представлять собой диагональную матрицу. В таблице 1 представлен результат такого перекрестного анализа, при этом классификаторы обучались на 50 изображениях из каждого класса, а для анализа были использованы по 40 изображений классов таким образом, чтобы они не пересекались с уже использованными для обучения. Из полученных данных видно, что подобная обучающая выборка вполне может использоваться для задач классификации изображений. Влияние количества изображений используемых для обучения классификаторов на число верных срабатываний показано в таблице 2.

Таблица 1

	кошка	автомобиль	стол	самолет	цветок	колесо	мост	небо	облака	дверь	окно	дом
кошка	36	1	0	0	2	0	1	1	0	0	0	0
автомобиль	0	18	2	3	3	5	1	1	0	3	1	4
стол	0	4	28	1	0	3	0	0	0	1	1	3
самолет	4	3	2	13	3	1	4	3	3	2	3	0
цветок	3	1	0	2	24	1	3	0	5	1	0	1
колесо	2	6	2	5	5	9	5	0	1	4	0	1
мост	9	0	0	0	0	1	22	0	0	0	4	5
небо	1	0	0	1	1	2	3	6	25	1	0	0
облака	4	0	0	2	2	2	1	5	25	0	0	0
дверь	0	0	6	1	1	2	1	0	0	22	4	4
окно	2	1	3	0	1	4	1	0	1	7	16	5
дом	2	1	0	0	0	2	4	0	0	0	7	25

Таблица 2

обучающая выборка	кошка	автомобиль	стол	самолет	цветок	колесо	мост	небо	облака	дверь	окно	дом
5	0.70	0.22	0.28	0.28	0.20	0.17	0.38	0.22	0.35	0.43	0.25	0.65
10	0.75	0.25	0.50	0.35	0.30	0.15	0.68	0.12	0.62	0.38	0.38	0.55
15	0.77	0.20	0.62	0.35	0.35	0.22	0.65	0.10	0.62	0.40	0.38	0.57
20	0.82	0.38	0.57	0.32	0.47	0.28	0.57	0.15	0.62	0.43	0.32	0.55
25	0.82	0.40	0.57	0.32	0.38	0.30	0.55	0.12	0.52	0.50	0.32	0.60
30	0.88	0.45	0.62	0.28	0.50	0.32	0.68	0.15	0.52	0.45	0.25	0.70
35	0.88	0.45	0.68	0.32	0.45	0.30	0.65	0.12	0.57	0.45	0.35	0.55
40	0.88	0.43	0.65	0.35	0.57	0.43	0.50	0.10	0.55	0.40	0.28	0.60
45	0.88	0.40	0.68	0.35	0.57	0.38	0.62	0.12	0.62	0.47	0.30	0.60
50	0.90	0.45	0.70	0.32	0.60	0.22	0.55	0.15	0.62	0.55	0.40	0.62

Стоит отметить, что переход от абстрактного набора изображений, относящихся к определенному классу название которого не имеет прямой связи с содержимым, к генерации набора опираясь на название класса в качестве поискового запроса, позволяет использовать дополнительные механизмы при классификации. Например, появляется возможность построить семантические связи между классами, исходя из его названия и расположения соответствующего понятия в семантическом графе. На примере графа, представленного в [4], была построена модель верификации результатов классификации.

Заключение

Использование автоматически полученной обучающей выборки способно решить часть проблем связанных со сложностью перенастройки системы классификации под изменившиеся требования.

Список литературы

1. Szeliski R. Computer Vision: Algorithms and Applications. - Springer-Verlang New York, 2010.
2. Lowe D. G. Object Recognition from Local Scale-Invariant Features. - The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.
3. Csurka G., Dance, C., Fan, L., Willamowski, J., Bray, C. Visual categorization with bags of keypoints // Workshop on statistical learning in computer vision, 2004.
4. Крайванова В.А., Кротова А.О., Крючкова Е.Н. Построение взвешенного лексикона на основе лингвистических словарей. - Материалы Всероссийской конференции «Знания – Онтологии - Теории», 2011.

АЛГОРИТМ ОБНАРУЖЕНИЯ АВТОМОБИЛЕЙ НА ОСНОВЕ ВЫЧИТАНИЯ ФОНА ДЛЯ ВИДЕОНАБЛЮДЕНИЯ ЗА ПЕШЕХОДНЫМИ ПЕРЕХОДАМИ*

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

В последнее время благодаря широкому классу решаемых задач активно развиваются технологии видеонаблюдения. Особенно заметны системы интеллектуального видеонаблюдения, способные в автоматическом режиме отслеживать ситуацию и минимизировать человеческие ресурсы для обработки информации. Примером систем интеллектуального видеонаблюдения являются системы автоматической фиксации нарушений правил дорожного движения, способные фиксировать такие нарушения, как превышение скорости, пересечения сплошной или стоп-линии, проезд на красный свет. В данной статье рассматривается задача обнаружения нарушений на нерегулируемом пешеходном переходе, которая является принципиально более сложной, чем обнаружение упомянутых видов нарушений. Это связано со сложностью одновременного сопровождения на видео автомобилей и пешеходов, а также со сложностью обнаружения нарушения по траекториям участников движения.

Ключевыми частями интеллектуальной системы видеонаблюдения являются алгоритмы компьютерного зрения, способные находить и сопровождать на видео участников движения. Их работа осложняется рядом факторов, среди которых:

- Неравномерность освещения;
- Необходимость работы в режиме реального времени;
- Перекрытие объектов друг другом;
- Шумы камеры и погодные условия (дождь, туман, снег и другие).

В этой статье описывается алгоритм для обнаружения автомобилей. Обнаружение автомобилей в данной задаче видеонаблюдения имеет ряд особенностей, которые следует учесть для построения наиболее подходящего алгоритма:

- Автомобили всегда видны фронтально или под небольшим углом;
- Траектории движения автомобилей подчинены ряду простых шаблонов;
- В наблюдаемой ситуации присутствуют два вида участников движения (автомобили и пешеходы), принципиально отличающиеся по внешнему виду.

Проведя анализ используемых подходов для обнаружения автомобилей, было выявлено, что использование *метода вычитания фона (background subtraction)* является наиболее часто используемым подходом благодаря своей простоте реализации и быстрдействию [1-4]. Алгоритмы на основе вычитания фона строят *модель фона* (изображение статических объектов сцены), сравнивают ее с текущим кадром, получая области, принадлежащие движущимся объектам. Стоит заметить, что такие алгоритмы чувствительны к шумам камеры, перепадам освещения, их работу нарушает появление тени от объектов, а также похожесть объекта и фона. Поэтому для верификации найденных областей используют дополнительные алгоритмы [5-7].

На данный момент мы остановились на алгоритме только на основе вычитания фона, так как области, получаемые методом вычитания фона и соответствующие автомобилям, существенно больше областей, соответствующих пешеходам, поэтому размер области является хорошим признаком для классификации этой области. Предлагаемый алгоритм

* Работа выполнена при поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере (госконтракт №0068 ГУ1/2013)

поддерживает модель фона (рисунок 1а) как усреднение 200 последних кадров. Кадр видео (рисунок 1б) сравнивается с текущей моделью фона, пиксели с разницей выше заданного порога формируют маску активных пикселей (рисунок 1в). Далее сканирующее окно заданного размера находит такие позиции, где внутри окна оказывается не менее 90% активных пикселей. Среди таких позиций выделяются локальные максимумы по числу активных пикселей. Эти максимумы считаются найденными объектами (рисунок 4г). Кроме того, порядок сканирования строится, учитывая специфику наблюдаемой сцены. Данные для тестирования алгоритма взяты из [8].

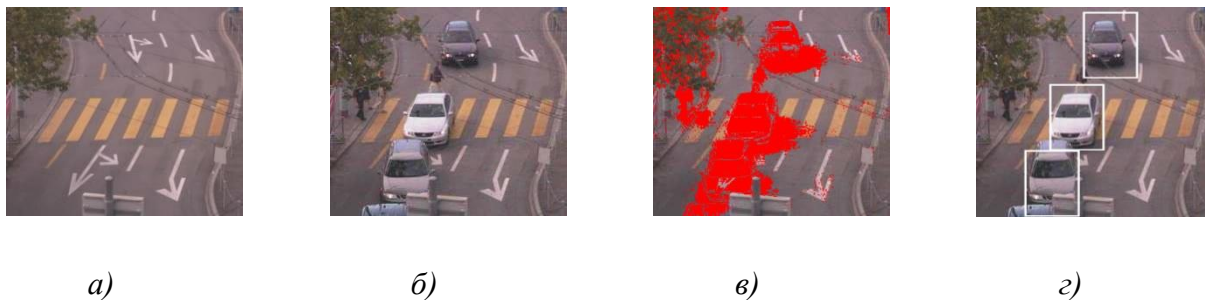


Рисунок 1 – а) Построенная модель фона; б) текущий кадр; в) маска активных пикселей; г) обнаруженные автомобили

В статье был предложен алгоритм обнаружения автомобилей, основанный на методе вычитания фона, для видеонаблюдения за пешеходными переходами с целью контроля соблюдения правил дорожного движения. Алгоритм справляется с задачей обнаружения автомобилей, однако имеет ряд существенных недостатков, связанных с невозможностью обрабатывать перекрытия автомобилей друг другом и связанных с чувствительностью к теням и неравномерности освещения. Поэтому дальнейшим направлением работы будет усиление предложенного детектора алгоритмом, использующим в качестве признаков объекта не только размер активной области, но и другие признаки.

Список литературы

1. Jyothi kiran M., Roy K. A Video Surveillance System for Speed Detection in Vehicles. International Journal of Engineering Trends and Technology. 2013. Volume 4, Issue 5, paper №25. Pp. 1437-1441.
2. Celil O., Fatih C. Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. Journal of Mathematical & Computational Applications. 2009, Vol. 14, Issue 3, pp. 187-196.
3. Norbert Erich B. Classification of vehicles for urban traffic scenes. PhD thesis, Kingston University. 2010.
4. Coifman B, Beymer D., McLauchlan P., Malik J. A real-time computer vision system for vehicle tracking and traffic surveillance. Journal of Transportation Research Part C: Emerging Technologies. Volume 6, Issue 4, August 1998, Pages 271–288.
5. Negri P., Clady X., Prevost L. Benchmarking haar and histograms of oriented gradients features applied to vehicle detection. Fourth International Conference on Informatics in Control, Automation and Robotics, Robotics and Automation. France, May 9-12, 2007. Pp. 359-364.

6. Sasidharan S., Kumar K. Vehicle Detection in Images using SVM. International Journal of Advanced Electrical and Electronics Engineering. 2013. Volume 2, Issue 6, pp. 62-68.
7. Chiu C.-C., Ku M.-Y., Wang C.-Y. Automatic Traffic Surveillance System for Vision-Based Vehicle Recognition and Tracking. Journal of Information Science and Engineering. 2010. Volume 26, pp. 611-629.
8. Leibe B., Schindler K., Van Gool L. Coupled Detection and Trajectory Estimation for Multi-Object Tracking // 2007 IEEE 11th International Conference on Computer Vision. Rio de Janeiro, Brazil. October 2007. P. 1–8.

РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ БАЗОЙ МЕТАДАННЫХ ГЕОИНФОРМАЦИОННОГО ИНТЕРНЕТ-ПОРТАЛА ИВЭП СО РАН

Коновалова К.В. – студент, Бубнова Н.Д. – ст. преп., Ловцкая О.В. – с.н.с.*
Алтайский государственный технический университет (г. Барнаул)
Институт водных и экологических проблем СО РАН* (г. Барнаул)

Метаданные – структурированные данные, которые описывают характеристики объектов-носителей данных, способствующие идентификации, обнаружению, оценке и управлению этими данными [1]. Метаданные используются в различных предметных областях и информационных технологиях: базы данных, технологии текстового поиска, веб-сервисы, библиотечное дело, медицина, геоинформатика и др. Широкое распространение метаданных говорит об их важности и ценности для общества. Метаданные помогают документировать данные, структурировать, а также создавать механизмы по поиску и идентификации данных.

Институт водных и экологических проблем (ИВЭП) Сибирского отделения Российской академии наук был основан 1 сентября 1987 года на базе лабораторий ряда институтов Сибирского отделения РАН, проводивших исследования природно-ресурсной и экологической направленности. Организация Института была вызвана необходимостью проведения фундаментальных исследований по изучению процессов и явлений в окружающей среде, а также оценки состояния водных ресурсов и решения проблем сбалансированного природопользования [2].

За время существования института в научных подразделениях было накоплено большое количество разнообразной информации о проводимых исследованиях. В связи с немалым объемом информации и трудностью проведения поиска среди разрозненных данных возникла потребность в структуризации и стандартизации с использованием международных стандартов метоописания, а также в создании единой базы данных института.

На начальном этапе был проведен отбор информации, включаемой в базу метаданных. На основе международных и отечественных стандартов (ISO 19115:2003 «Geographic information Metadata», ГОСТ Р 52573-2006 «Географическая информация. Метаданные») был разработан единый образец файла метазаписи, который включает в себя следующие поля:

1. Заголовок
2. Аннотация
3. Код классификатора.
4. Ключевые слова
5. Производители данных
6. Персоны, связанные с данными
7. Первая публикация
8. Носитель информации
9. Дата создания данных

10. Формы разрешений на данные
11. Авторы метаданных
12. Вложенная ссылка (характеристика типа набора данных текст, снимок, HTML и т.п., собственно ссылка имя прилагаемого файла, ссылка на источник в Интернет и т. п.)
13. Координаты
14. Географическая привязка

Файлы метаданных могут содержать не все поля, но в целом структура метазаписи едина, и не нарушена.

Следующий этап в разработке единой базы института – реализация интерфейса доступа к метаданным.

На данном этапе первым делом был выбран способ классификации метазаписей. Географические данные, накопленные за время существования ИВЭП, можно классифицировать используя линейную классификацию, когда отсутствует как таковая классификация и все данные располагаются независимо друг от друга. Чтобы структурировать данные можно применить классификации по подразделениям института, по водным бассейнам, по водным объектам, по административным районам, по субъектам федерации, по среде исследования, по теме исследования и т.д. Так же можно организовать иерархическую классификацию, когда заданное множество делится на подмножества, образуя древовидную структуру.

Каждая из классификаций имеет свои недостатки:

1. Линейная слишком проста и представляет собой длинный список без возможности быстрого поиска;
2. Классификации, основанные на каком-либо одном общем критерии для группы, слишком не однозначны и имеют множество пересечений;
3. Иерархическая классификация хороша в своей простоте и структурированности, но она не отражает полный перечень возможных подклассификаций геоданных.

Наиболее подходящей классификацией для структурирования географических данных является фасетная классификация. Фасетная классификация – классификация, дающая возможность классифицировать объекты одновременно по нескольким различным признакам [3]. Преимущества этого вида классификаций в том, что они облегчают многоаспектное классифицирование объектов, позволяя воедино собрать все аспекты рассмотрения. Фасетное классифицирование легче поддается изменениям при введении новых понятий, допускает большую глубину деления. Недостаток фасетной системы классификации заключается в сложности построения (т.к. необходимо учитывать все многообразие классификационных признаков).

Структурно фасетную классификацию можно представить в следующем виде [4]:

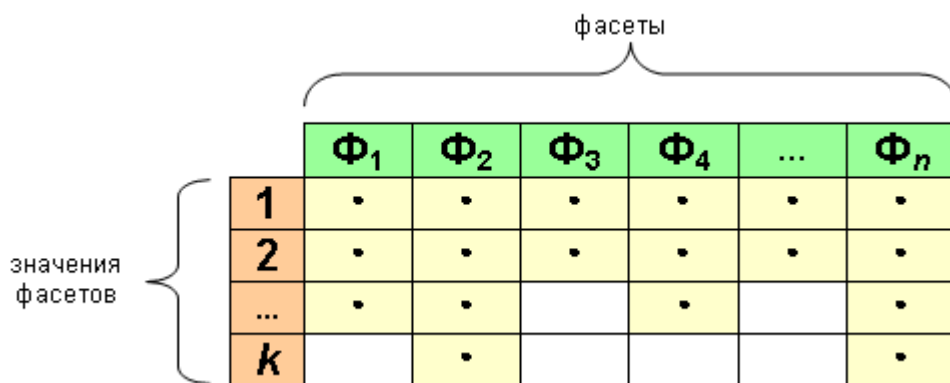


Рисунок 1 – Структура фасетной классификации

Здесь под фасетом подразумевается признак, по которому в многоаспектной классификации осуществляется деление.

В качестве базы данных для хранения классификаций выбрана графовая база данных Neo4j [5], представляющая собой разновидность баз данных с реализацией сетевой модели в виде графа и его обобщений. Основные элементы – узлы и отношения. Фасетная классификация географических объектов, а так же связанных с ними проектов, слоев и т. п., наполнена семантикой и графовая база данных Neo4j позволяет осуществить все необходимые функции. Каждый из узлов в базе – узел классификации, каждое отношение – связь одного узла классификации с другим, совокупность отношений – отдельная классификация. Каждое отношение может быть нагружено своим смыслом, каждый узел – своими свойствами. В системе могут существовать два узла с одинаковым названием, но несущие совершенно различный смысл, т.к. у них свои связи и свои свойства. Neo4j также позволяет осуществлять поиск по необходимым параметрам.

Интерфейс доступа к базе данных реализуется на языке Java с использованием веб-фреймворка Spring MVC [6]. Выбор данных инструментов позволяет получать доступ к системе, имея только браузер, нет необходимости в каких-либо дополнительно устанавливаемых компонентах. Отсутствует привязка к конкретной операционной системе. Архитектура системы на основе паттерна MVC (модель-представление-контроллер) [7] является очень гибкой: интерфейс всегда может быть заменен альтернативной реализацией; не затрагивая вид, можно изменить поведение системы, для этого достаточно заменить контроллер; модель данных не зависит от представления.

Система состоит из двух частей:

1. Интерфейс для администрирования базы метаданных. Он содержит инструменты для добавления и изменения классификаций.
2. Интерфейс пользовательской части. Содержит инструменты для различного поиска метаданных и их отображения.

Создание единой базы метаданных позволит собрать воедино, структурировать и сделать более доступной информацию, собранную в результате многолетних исследований подразделениями Института водных и экологических проблем.

Список литературы

1. Вязилов, Е.Д. Современные подходы по созданию метаданных [Текст] / Е.Д. Вязилов, А.Е. Кобелев // Современные проблемы дистанционного зондирования Земли из космоса: Физические основы, методы и технологии мониторинга окружающей среды, потенциально опасных явлений и объектов: сб. статей. – Москва, 2010. – С. 194-204.
2. Институт водных и экологических проблем СО РАН [Электронный ресурс] – Режим доступа: <http://www.iwep.ru/ru/>
3. Гендина Н.И. Неиерархические классификации [Презентация].
4. Автоматизированные информационные технологии. [Электронный ресурс]. – Режим доступа: http://fakit.narod.ru/uch2003/p4_2_1_2.html
5. Neo4j. [Электронный ресурс]. – Режим доступа: <http://www.neo4j.org/>
6. Spring Framework. – [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki/Spring_Framework
7. Model – View – Controller. – [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org/wiki/Model-View-Controller>

ЗАЩИТА ИНФОРМАЦИИ НА МОБИЛЬНЫХ УСТРОЙСТВАХ ПОД УПРАВЛЕНИЕ ОС ANDROID В КОРПОРАТИВНОЙ СРЕДЕ

Корякин Н.А. – студент, Третьяков И.В. – ведущий разработчик мобильных приложений*
Алтайский государственный технический университет (г. Барнаул)
ООО «Энтерра Софт» * (г. Барнаул)

Использование персональных устройств на рабочих местах обязывает корпорации разрабатывать и использовать технологии, которые обеспечивают информационную и экономическую безопасность, что является актуальной задачей. Необходимо контролировать персональные устройства и обеспечивать их безопасное использование, при этом нужно учитывать интересы владельцев. Для эффективного решения данной задачи необходимо уделять должное внимание современному технологическому развитию, возможностям мобильных устройств, а также безопасной работе с конфиденциальной информацией. При решении задач подобного класса наиболее используемым является подход «Bring-Your-Own-Device» (BYOD). Корпорации используют BYOD как способ привлечения и удержания талантов. Ограничения в использовании персональных устройств отрицательно отражаются на показателях качества работы сотрудников, т.к. для них это более удобно. Личное оборудование зачастую более высокопроизводительно и эффективно, чем то которое предоставляет организация, что подтверждает необходимость внедрения технических решений рассматриваемой задачи.



Рисунок 1 – График использования мобильных устройств на рабочих местах

С ростом числа приложений, контента и данных, доступ к которым может осуществляться с самых разнообразных устройств, становится жизненно необходимым внедрение системы управления мобильными устройствами MobileDeviceManagement. MDM включает в себя центрально-ориентированную систему функций по защите и управлению данными и приложениями, а также центрально-ориентированную систему информационных функций. Необходим менеджмент всего жизненного цикла устройств, и ИТ-персонал несет полную ответственность за мобильные устройства, находящиеся в собственности компании, в том числе за настройку их аппаратной и программной части, поддержку приложений и соблюдение обязательств предприятия в их отношении. Если же сотрудникам разрешено использование личных устройств на местах (BYOD), то необходимо разделять обязанности, а также определить совместную ответственность предприятия и

пользователя, что должно быть надлежащим образом организовано, доведено до сведения и проконтролировано.

Symantec Mobile Management и McAfee Enterprise Mobile Management – одни из самых распространенных MDM. Они обеспечивают защиту данных, а также управление мобильными устройствами. Несмотря на многочисленные позитивные характеристики современных MDM, многие компании предпочитают их не использовать. Основной из причин такого решения является высокая цена гарантированной безопасности. Внедрение такой системы предполагает установку на устройство специального набора программного обеспечения, с помощью которого можно подключаться к корпоративным сервисам. В этом случае на мобильном устройстве фактически создается защищенная среда, через которую пользователь получает доступ к корпоративным данным. При этом ему приходится осваивать новый набор приложений, который входит в состав самого решения. На устройстве создаются две рабочие среды: персональная, доступ к которой имеет пользователь, и корпоративная, контролируемая ИТ-администраторами компании, при этом администратор не имеет доступа к персональной части устройства. Программное обеспечение мобильного клиента, а также постоянная поддержка связи с корпоративной сетью сильно нагружают устройства, поэтому их производительность снижается, и они теряют свою ценность для сотрудников организации.

В настоящее время ведется разработка программного обеспечения описанного класса. Данное техническое решение обладает низкой стоимостью и обеспечивает надежность работы, возможность обработки критических ситуаций и при этом не требует больших ресурсов вычислительной мощности. Возможности мобильного устройства зависят от его текущего местоположения, времени суток и должности сотрудника. Система управления мобильными устройствами централизована и гарантирует информационную безопасность для организации, которая ее использует. Конечный пользователь должен быть заинтересован в возможностях MDM, поэтому влияние корпоративной среды устройства на его персональную среду минимально. Для настройки, контроля и сопровождения устройств реализована консоль администратора, с помощью которой можно сконфигурировать оборудование сотрудников, а также создать правила использования персональных устройств и контролировать нарушения этих правил. Система обладает удобным для пользователя функционалом - потерянное или украденное устройство всегда можно заблокировать или же обнаружить с помощью запроса последних известных координат. Основным недостатком программного продукта является отсутствие кроссплатформенности клиентского приложения, которое работает только под операционной системой Android минимальной версии 2.2.

Список литературы

1. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. ДКМ Пресс – Администрирование и защита. ISBN 5-94074-172-X; 2002.
2. Шифрование и защита данных в GoogleAndroid. «ТехноМир».– Портал [Электронный ресурс]. [2005-2013]. – Режим доступа: <http://www.mobimag.ru/Articles/5046>.
3. Малюк А.А. Информационная безопасность: концептуальные и методологические основы защиты информации. Москва. – Горячая линия – Телеком. 2004.
4. ForeScout Technologies Inc. NetworkAccessControl [Электронный ресурс] - 2014. – Режим доступа: <https://www.forescout.com/solutions/network-access-control/>
5. RemotiumVirtualMobilePlatform (VMP) [Электронный ресурс] - 2014. – Режим доступа: <https://www.remotium.com/#about>

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗИРОВАННОГО ПОСТОЕНИЯ ГРАФИКА РАБОТЫ МАСТЕРОВ ПО ОБСЛУЖИВАНИЮ ЗАЯВОК

Костин К.В. – студент, Кантор С.А. – к.ф.-м.н., профессор,
Созутов А.Е. – технический директор компании «Ритейл Сервис»*
Алтайский государственный технический университет (г. Барнаул)
ООО «Ритейл Сервис»* (г. Барнаул)

Транспортная задача в своем классическом варианте формулируется следующим образом – необходимо найти оптимальный план перевозок продукта из пунктов наличия в пункты потребления используя заранее предопределенное количество транспортных средств.

На практике в данной формулировке эта задача редко применяется, так как в каждой конкретной предметной области существуют свои собственные ограничения. Это могут быть ограничения на время транспорта в пути или ограничения на посещения некоторых пунктов некоторыми транспортными средствами.

В связи с оптимизацией деятельности компании возникла следующая задача, которую можно отнести к классу транспортных задач.

Компания занимается установкой, ремонтом и обслуживанием оборудования на объектах по всему краю и имеет представительства в некоторых населенных пунктах. Работы, выполняемые компанией, осуществляются в соответствии с поступающими заявками, из которых следует определенный набор требований.

1. Период, то есть начальная и конечная даты выполнения заявки. Если она будет обслужена после конечной даты, то считается просроченной.

2. Перечень работ, которые необходимо выполнить, откуда можно определить примерное время, затрачиваемое на выполнение заявки.

3. Местоположение объекта, на котором необходимо осуществить работы.

4. Приоритет. Каждое установленное компанией оборудование должно проходить планово-предупредительные ремонты, поэтому существуют плановые заявки, имеющие наименьший приоритет. Наивысший приоритет имеют самые критические заявки, например поломка оборудования, влияющего на ход производства у заказчика. Могут быть не критические заявки, например перенастройка оборудования.

Обслуживанием заявок занимаются мастера. Существует базовое расписание работы мастеров, в котором определено какой мастер выезжает из на выполнение заявки из какого представительства, его продолжительность работы в каждый конкретный день.

Необходимо, имея список заявок и базовое расписание, составить график работы мастеров по обслуживанию заявок. При этом графики работы должны сравниваться в соответствии со следующими критериями:

1. количество просроченных заявок;
2. общее количество маршрутов в расписании
3. количество дней, потраченных на выполнение заявок;
4. максимальное количество мастеров, работающих в один день.
5. суммарное время работы всех мастеров.

Критерии перечислены в порядке уменьшения важности. Это означает, что из двух расписаний предпочтение отдается тому, которое лучше по i -му критерию, а по критериям $1, \dots, i-1$ они равны.

Данная задача является NP-полной [1], поэтому за приемлемое время найти оптимальное решение невозможно. Однако, в данной постановке, оптимальное решение задачи не требуется. Это связано с тем, что в реальности точно по плану работы не могут быть выполнены: какая-то заявка будет выполняться быстрее, какая-то дольше, мастер по какой-либо причине может не выйти на работу и т.д. Поэтому поставим задачу о нахождении рационального решения за приемлемое время. Существует целый класс алгоритмов,

позволяющих находить приближенное решение этой задачи, называемых эвристическими [2]. Эвристический алгоритм – это алгоритм, правильность которого в общем случае не доказана, но про который известно, что на практике он дает достаточно хорошее решение.

Для решения был выбран генетический алгоритм [3], одно из преимуществ которого – простота при его распараллеливании. Для реализации метода необходимо определить структуру организма. Это сделано следующим образом. Назовем базовыми пункты, где расположены представительства компании и из которых мастера выезжают для выполнения заявок. Пусть имеется n базовых городов, пронумеруем их от 1 до n . На практике все населенные пункты разбиваются на районы, и каждый район обслуживается только из одного базового города. Поэтому можно сделать хорошее начальное приближение к искомому решению – каждую заявку поместить в список ближайшего к ней базового города. Поэтому разобьем множество заявок на n частей, и будем считать, что имеется массив R размерности n , где n – количество базовых городов, $R[i]$ – список заявок, выполняемых мастерами из i -го базового города.

Пусть m – продолжительность периода, на который составляется график работ. Определим матрицу A размерности $n*m$, где $A[i][j]$ – список мастеров, которые выезжают из i -го базового города в j -ый день.

Кроме того, имеется матрица S , описывающая время перемещения мастера из одного пункта в другой, $S[i][j]$ — время в минутах, на проезд от пункта i до пункта, j .

Теперь на основе массивов R , A и S необходимо составить расписание. Состояние мастера в какой-либо момент времени можно описать его местоположением, текущим временем работы и базовым городом. Выполняя заявку, мастер переходит из одного состояния в другое, то есть меняется его местоположение и время работы. Мастер может выполнить заявку из какого-то состояния, если в сумме текущее время работы, время проезда до места выполнения заявки, время выполнения заявки и возврат в базовый город будут меньше времени работы мастера в текущий день. Считаем, что мастера из списка $A[i][j]$ выезжают по очереди и по очереди выполняют заявки из $R[i]$, то есть первый мастер выполняет максимальное количество заявок, которое может выполнить, потом второй мастер максимальное количество заявок, начиная с той, которую не смог выполнить первый и т.д. Если мастера в j -ом списке закончились, то переходим к следующему дню и аналогично составляем расписание. Таким образом, по A , R и S однозначно вычисляется расписание.

Матрица S не изменяется в время работы алгоритма. Каждый организм описывается индивидуальными массивом R и матрицей A . Поколение - набор организмов. Популяция — набор сменяющих друг друга поколений. В одной популяции на каждой итерации алгоритма строится новое поколение на основе предыдущего с использованием методов скрещивания и мутации. Скрещивание - получение организма потомка на основе двух родительских организмов, мутация — случайное изменение структуры организма.

Реализованы следующие виды скрещивания (с указанием вероятности их наступления):

▲ Скрещивание по списку заявок, вероятность 0,7. Организм потомка принимаем равным второму организму. Выбираем случайное число i , выбираем случайный подсписок из списка $R[i]$ первого организма и помещаем его в случайное место списка $R[i]$ потомка. При этом некоторые заявки будут встречаться в потомке два раза, поэтому удаляем второе вхождение каждой из них.

▲ Скрещивание по спискам мастеров, вероятность 0,3. Массив R организма потомка с равной вероятностью копируем у одного из родителей. Аналогично списки $A[i][j]$, для любых возможных i и j , с равной вероятностью копируем у одного из родителей

Были реализованы следующие виды мутации:

1. Мутация перемещения заявок между базовыми городами, вероятность 0,02. Из случайного списка заявок берем случайную заявку и перемещаем ее в случайное место случайного списка.
2. Мутация перемещения заявок внутри списка базового города, вероятность 0,4. Берем

- случайный список заявок, в нем случайно выбираем 2 заявки и меняем их местами.
3. Мутация реверсирования, вероятность 0,2. Берем случайный список заявок, в нем выбираем случайный подсписок и реверсируем его.
 4. Мутация уменьшения количества мастеров, вероятность 0,1. Из случайного списка $A[i][j]$ удаляем последнего мастера
 5. Мутация увеличения количества мастеров, вероятность 0,1. В случайный $A[i][j]$ добавляем мастера, который может выехать из i -го базового города в день j .
 6. Мутация изменения порядка мастеров, вероятность 0,2. В случайном списке $A[i][j]$ выбираем два случайных мастера и меняем их местами.
 7. Оптимизация путем изменения порядка соседних заявок, вероятность 0,01. Перебираем все рядом стоящие в списках заявки и меняем их местами, если расписание улучшилось, то оставляем их, иначе меняем обратно. Вероятность данной мутации очень мала потому, что она выполняется очень долго и существенно увеличивает время работы алгоритма. Данная мутация применяется к лучшим организмам популяции после каждой итерации алгоритма.

Были реализованы следующие методы параллельной работы алгоритма:

- Реализован многопопуляционный параллельный генетический алгоритм - это значит, что при работе используется несколько популяций, каждая из которых обрабатывается в собственном потоке. Через некоторое количество итераций происходит обмен лучшими организмами между популяциями. Реализована работа с двумя популяциями.

- На каждой итерации получения нового поколения происходит большое количество операций скрещивания. Можно заметить, что эти операции независимы, поэтому реализовано их параллельное выполнение. Так же реализовано параллельное выполнение мутаций. Эта работа осуществляется в двух потоках.

- Из алгоритма составления расписания на основе организма видно, что для каждого базового города разбиение списка заявок на маршруты независимо. Реализовано параллельное вычисление расписания с использованием двух потоков.

Все представленные выше значения были подобраны в результате численных экспериментов.

При дальнейшем развитии работы планируется добавление новых условий, таких как:

- учет необходимого оборудования для ремонта;
- учет квалификации сотрудника для выполнения заявки.

Результаты работы алгоритма при 250 заявках, 17 мастерах и периоде планирования 6 дней представлены в таблице:

Критерии	Время работы программы, в секундах				
	0	5	10	15	20
Количество просроченных заявок	29	16	6	0	0
Общее количество маршрутов	33	33	32	30	28
Количество дней на выполнение заявок	5	5	4	4	3
Максимальное количество мастеров	17	16	15	15	14
Время работы всех мастеров, в минутах	12815	13162	12773	12210	11145

Список литературы

1. Michael R. Garey. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* / Michael R. Garey, David S. Johnson.- NewYork: W.H. Freeman and Company 1979. - 339 p.
2. Michalewicz Z. How to solve it. Modern heuristics. / Michalewicz Z., Fogel D.B Berlin. - Springer, 2004. – 155 p.
3. Francisco B. Pereira. Bio-inspired algorithms for the vehicle routing problem / Francisco B. Pereira, Jorge Tavares. - Coimbra: Springer, 2008 – 221 p.

СОЗДАНИЕ УНИВЕРСАЛЬНОЙ СРЕДЫ ДЛЯ ПРОВЕДЕНИЯ ВЕБИНАРОВ

Кронберг И.А. – студент, Старолетов С.М. – к.ф.-м.н., ст. преп.
Алтайский государственный технический университет (г. Барнаул)

Постановка и анализ проблемы

Современные системы дистанционного (электронного) обучения можно разделить на две категории:

1. Системы, обеспечивающие асинхронный характер взаимодействия обучающихся и преподавателей. Для таких систем характерны такие функции как:
 - 1.1. Создание и управление электронными образовательными курсами.
 - 1.2. Проведение лабораторных работ, контрольных опросов и их оценка.
2. Системы, обеспечивающие синхронный характер взаимодействия обучающихся и преподавателей. Для таких систем характерны такие функции как:
 - 2.1. Организация online вещания и обмена мультимедийной информацией между преподавателем и обучающимися.
 - 2.2. Возможность коллективного выполнения заданий в режиме реального времени(collaborativenetwork).

На рынке существует сравнительно мало решений поддерживающих как асинхронные, так и синхронные способы организации занятий. Те же из них, что обеспечивают в равной степени и позволяют проводить оба типа занятий, часто имеют высокую цену приобретения и использования лицензионной копии программного комплекса.

Постановка задачи

На базе результатов анализа известных систем дистанционного (электронного) обучения:

1. Определить функциональность будущей системы.
2. Подобрать инструментарий необходимый для реализации системы, на основе современных тенденций в области разработки программного обеспечения.
3. Спроектировать базу данных и составить диаграмму классов
4. Реализовать базовые функции системы:
 - 4.1. Возможность создания и управления электронными учебными курсами
 - 4.2. Организация online занятий и семинаров с использованием большинства доступных средств взаимодействия (текстовый чат, видео-конференция, интерактивная доска)
5. Осуществить установку и тестирование системы на удаленном хосте.

Решение

Для решения поставленных задач были использованы следующие инструменты и технологии:

Symfony 2 - свободно распространяемый веб фреймворк (CMF) , написанный на PHP5, базирующийся на архитектуре Model-View-Controller. Имеет поддержку множества баз

данных (MySQL, PostgreSQL, SQLite или любая другая PDO-совместимая СУБД). Информация о реляционной базе данных в проекте должна быть связана с объектной моделью.

WebRTC — интернет-протокол, проект с открытым исходным кодом, предназначенный для организации передачи потоковых данных между браузерами или другими поддерживающими его приложениями по технологии точка-точка (Peer – to - Peer)

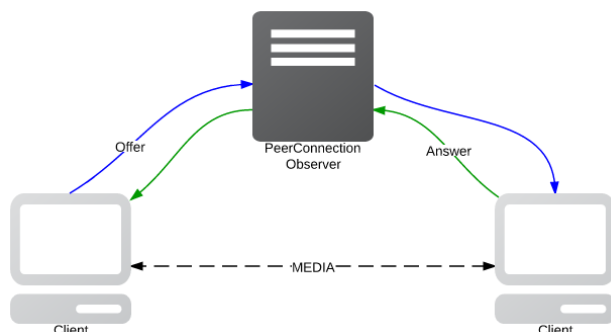


Рисунок 1 – Схема организации передачи данных Peer – to –Peer

WebRTC позволяет получать медиаданные (аудио и видео) через браузер и устанавливать соединение между двумя и более клиентами, через которое могут передаваться обычные данные и медиапотоки.

Непосредственно при создании данного приложения была использована javascript библиотека Peer.js, которое является надстройкой надвстроенной в браузеры функциональностью WebRTC API (последние версии Mozilla, GoogleChrome, Opera) и обеспечивает стабильное соединение и передачу медиа-данных. Созданная система состоит из двух модулей:

1. Модуль создания и управления электронными курсами;
2. Модуль для организации on-line занятий.

Для обеспечения создания и хранения электронных курсов была разработан и реализован при помощи Symfony2 проект базы данных.

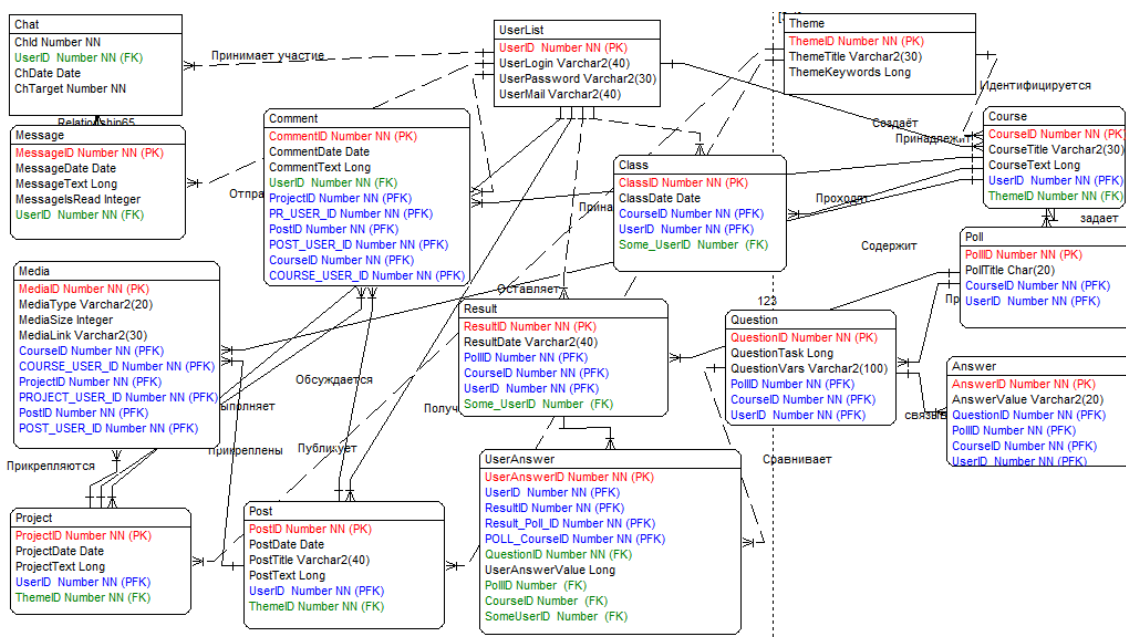


Рисунок 2 – Проект базы данных приложения

Модуль проведения занятий включает в себя такие возможности как:

1. Создание отдельных виртуальных комнат для проведения занятий
2. Групповой текстовый чат
3. Живое видео – вещание с камеры преподавателя (при необходимости и с камеры обучаемого)
4. Интерактивная доска с возможностью рисования.
5. Область для показа слайдов.

Результаты

Разработанная система объединяет в себе такие функции как:

- 1) Создание и управление электронными образовательными курсами.
- 2) Организация он-лайн вещания и обмена мультимедийной информацией между преподавателем и обучающимися.

Функционала системы достаточно, чтобы качественно организовать проведение индивидуальных занятий и занятий для малых групп (от 5 до 25 человек).

В перспективе, планируется создания специализированных приложений для мобильных устройств на базе операционных систем Windows 8/Phone, Android, AppleIOS.

Список литературы

1. Symfony 2 Frameworkofficialsite [Электронныйресурс]. – М., 2010. – Режимдоступа: <http://symfony.com/>
2. Lynda Education Javascript tutorials library – 2013 – Режимдоступа: <http://www.lynda.com/aboutus/website-use-policy>
3. Webrtcofficial website – 2013 – Режимдоступа: <http://www.webrtc.org/>

ИСПОЛЬЗОВАНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ТЕОРИИ РАСПИСАНИЙ

Лаптев М.А., Некрасов Д.В. – студенты, Сорокин А.В. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Проблема решения задачи упорядочения из области теории расписаний достаточно долгое время занимает умы многих исследователей, начиная с 60-х годов, когда была опубликована работа Джонсона [1,2]. Как было показано в работе, в общем случае нет конкретного общего решения задачи упорядочения, хотя в частных случаях получаются достаточно простые алгоритмы ее решения.

Как правило, задачу упорядочения связывают с производственной задачей, хотя на самом деле ее можно применять в различных областях. Остановимся на ее формулировке, привязанной к производству, имеющей смысл в серийном производстве.

Данная задача, представляет собой задачу выбора порядка обслуживания, оптимизирующего какой-либо существенный показатель качества функционирования системы. Одними из первых ученых, исследовавших задачу упорядочения, были американские ученые Беллман и Джонсон. Приведем постановку этой задачи в их формулировке.

Имеется m станков, выполняющих различные технологические операции, на которых необходимо обработать набор n различных деталей с номерами $i=1,2,\dots,n$. Предполагается, что дано t_{ij} - время обработки i -ой детали на j -ом станке, не являющееся нулевым. Каждая

деталь имеет одинаковый технологический маршрут обработки, в котором прописан порядок «движения» детали от первого станка к последнему. Такой маршрут называется «жёстким». Ограничения, фигурирующие в задаче, формулируются следующим образом:

1. Для каждой детали обработка начинается с первого станка;
2. Для каждой детали обработка на j -ом станке может начинаться не раньше, чем окончится её обработка на станке $j-1$;
3. На каждом станке одновременно может обрабатываться не более одной детали;
4. Начавшаяся операция не прерывается до полного её завершения.

Необходимо найти такую последовательность обработки набора n -деталей, которая минимизирует время окончания обработки последней детали этой последовательности.

Данная задача в общем случае при $m \geq 3$ является NP-полной и, следовательно, для её решения не существует алгоритма, завершающегося за полиномиальное число операций. В связи с этим для её решения применяют подходы, находящие решения, приближённые к оптимальному. Хотя степень приближенности здесь оценить достаточно сложно. Среди таких подходов можно выделить методы, основанные на случайном поиске: метод отжига [3,4], генетические алгоритмы [7-9] и др., и детерминированные методы (подходы), основанные на «правилах предпочтения» или на «правилах Петрова» [5].

В данной работе для решения задачи упорядочения будет рассмотрен метод, основанный на "Генетическом алгоритме". Будет представлено его сравнение с простой эвристикой и подходом, основанным на "правилах Петрова".

Генетический алгоритм — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомым параметров с использованием механизмов, аналогичных естественному отбору в природе. Он является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием методов естественной эволюции, таких как наследование, мутации, отбор и кроссинговер. Отличительной особенностью генетического алгоритма является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе.

В работе реализован генетический алгоритм, состоящий из следующих основных шагов:

1. Генерируется первое поколение, каждая особь, которой содержит случайную последовательность деталей.
2. Для каждой особи считается выживаемость (Fitness). Она показывает близость последовательности к «идеальной». На основании выживаемости рассчитывается вероятность оставить потомство.
3. Создаётся потомство, которое заменяет текущее поколение. Каждая особь потомства имеет двух родителей, признаки которых данная особь содержит в себе (плюс мутации - признаки, не содержащиеся ни в одном из родителей).
4. Процесс «скрещивания» двух особей называется кроссинговером. Кроссинговер предусматривает наследование случайных признаков родителей, а также появление новых признаков (мутаций).
5. Повторяем пункты 2-4 некоторое число раз. Изменяя данный параметр или размер популяции, можно регулировать точность ответа и время выполнения программы.
6. Среди всех существовавших особей запоминается наиболее удачная, она и будет ответом на задачу.

Может быть показано, что каждое следующее поколение будет сходиться к идеальному решению, алгоритм даёт всего лишь приближённое решение. Чтобы ответить на вопрос, насколько это решение точное было проведено исследование, которое показало, что алгоритм проявляет себя состоятельным при малых и средних размерностях. При таких начальных условиях он обгоняет по точности подход, основанный на «правилах Петрова» и

простую эвристику. При больших объемах задачи также возможно использование генетического алгоритма, но оно будет оправдано только при наличии больших вычислительных мощностей, так как время работы программы имеет асимптотику:

$$(\text{число итераций}) \cdot (\text{размер популяции}) \cdot n^2 \cdot m.$$

Результаты временных затрат программ, реализующих подход Петрова, простую эвристику и генетический алгоритм, в зависимости от размерности задачи упорядочения приведены в таблицах.

Статистика по задаче Джонсона pxm

	input1.txt 5*7		input2.txt 7*7		input3.txt 50*50		input4.txt 75*50	
	Время работы, сек	Результат, мин	Время работы, сек	Результат, мин	Время работы, сек	Результат, мин	Время работы, сек	Результат, мин
Правила Петрова	0.000336	94	0.000408	107	0.002017	6527	0.002917	8485
Простая эвристика с n последовательностями	0.000256	99	0.000193	103	0.005589	6676	0.007672	8544
Ген с размером популяции 10	0.001292	88	0.001186	101	0.014009	6589	0.020896	8553
Ген с размером популяции 25	0.002729	89	0.002826	103	0.032027	6586	0.036464	8537
Ген с размером популяции 50	0.005306	89	0.005723	101	0.057771	6571	0.084593	8506
Ген с размером популяции 100	0.011629	88	0.011966	101	0.109234	6495	0.175487	8451

	input5.txt 50*100		input6.txt 112*105		input7.txt 500*500		input8.txt 1000*800	
	Время работы, сек	Результат, мин	Время работы, сек	Результат, мин	Время работы, сек	Результат, мин	Время работы, сек	Результат, мин
Правила Петрова	0.003008	10282	0.007010	15207	0.113109	72317	0.337491	131528
Простая эвристика с n последовательностями	0.009865	10227	0.044368	15270	2.640627	73048	16.079928	132189
Ген с размером популяции 10	0.026561	10312	0.061694	15290	0.985156	72857	3.292825	132358
Ген с размером популяции 25	0.051172	10098	0.107757	15185	2.391346	72955	7.074391	132221
Ген с размером популяции 50	0.109293	10148	0.196233	15045	4.299564	72811	13.577553	132144
Ген с размером популяции 100	0.222914	10132	0.421647	15091	8.485434	72909	26.438415	132103

Список литературы

1. Конвей Р.В., Максвелл В.Л., Миллер Л.В. Теория расписаний. М., Наука, 1975. – 360 с.
2. Танаев В.С., Шкурба В.В. Введение в теорию расписаний. М., Наука, 1975. – 256 с.
3. http://ru.wikipedia.org/wiki/Алгоритм_имитации_отжига
4. Лопатин А.С. Метод отжига. – СПбГУ. - <http://www.math.spbu.ru/user/gran/sb1/lopatin.pdf>
5. Шукис А.А. Исследование операций: Задачи упорядочения. – Изд-во: АПИ, 1980. – 98 с.

6. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы: Учебное пособие. — 2-е изд. — М: Физматлит, 2006. — 320 с.
7. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы — 2-е изд. — М: Горячая линия-Телеком, 2008. — С. 452 с.
<http://habrahabr.ru/post/128704/>

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ОБУЧЕНИЯ СТУДЕНТОВ ЭКОНОМИЧЕСКИХ СПЕЦИАЛЬНОСТЕЙ ФИНАНСОВО-КОММЕРЧЕСКИМ ОПЕРАЦИЯМ С ПОМОЩЬЮ ИНТЕРАКТИВНЫХ МЕТОДОВ

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

В современном обществе образование играет очень важную роль. Постоянное расширение числа компьютерных классов, создание на их основе информационно-вычислительных сетей вузов и школ с последующим объединением в региональные сети, расширение информационной наполненности таких сетей приводят к качественным изменениям в методике преподавания дисциплин, создают предпосылки к созданию компьютерной образовательной среды. Создание и развитие такой среды предполагает решение следующих задач.

Первая задача связана с переосмыслением методики преподавания дисциплин в вузах. Традиционная методика обучения и методика обучения с использованием компьютера сильно разнятся между собой. Назрела существенная необходимость разработки структурированных учебных материалов для использования их в КОП, разработки методики их подачи и осуществления контроля знаний в КОП.

Вторая важная задача состоит в разработке общих подходов и принципов для объединения КОП в единую учебно-информационную среду. Прежде всего, большое внимание должно быть уделено вопросам навигации как от фрагмента к фрагменту в рамках отдельного компьютеризированного курса, так и от курса к курсу.

Унификация принципов навигации, разработка общих подходов и, возможно, инструментальных средств для создания различных типов КОП вне зависимости от разделов знаний должны строиться на основе того, что компьютерная технология обучения - это образовательный процесс, основанный на едином средстве обучения - компьютере, и взаимодействие пользователя с новой обучающей программой должно строиться на основе привычных ему навыков.

При разработке компьютерных обучающих программ следует учитывать эргономический аспект разработки. Пользователь КОП может проводить за компьютером достаточно длительное время (особенно если пользователь работает в режиме самообразования). При этом ему часто требуется повторять большое число однотипных манипуляций (выбор режима работы, ввод исходных данных и т.п.). Отсюда следует, что при разработке пользовательского и графического интерфейса КОП необходимо ориентироваться на требования инженерной психологии и эргономики.

Информационное наполнение КОП состоит из различных компонент, реализующие ее определенные функциональные свойства.

Выделим три основные компоненты информационного наполнения обучающих программ:

- текстовая;
- вычислительная;
- имитационная.

Текстовая компонента. Любая обучающая программа немыслима без использования в большей или меньшей мере текстового материала (от изложения теоретической части курса и до надписей на нестандартных кнопках). При разработке программ следует стремиться к тому, чтобы весь текстовый материал курса размещался вне вычислительной и имитационной компоненты. Выполнение этого требования позволяет осуществлять модификацию КОП, а также, при необходимости, создавать иноязычные варианты обучающей программы.

Вычислительная компонента. Во многих КОП используется сложный и уникальный математический аппарат для поддержки процедуры обучения, визуализации полученных результатов, построения оценочной части контролирующего или тестирующего раздела обучающей программы и решения других задач. Создание подобных программ трудоемкий процесс.

Имитационная или моделирующая компонента. Необходимость использования моделей в процессе обучения обусловлена следующими причинами: моделирование позволяет за время одного занятия рассмотреть и проанализировать процессы, которые в реальной жизни занимают дни, недели, месяцы и годы, а использование моделей позволяет сформировать адекватное представление об исследуемом процессе.

Технологический аспект разработки компьютерных обучающих программ связан с кодированием, сопровождением и развитием во времени КОП как программного продукта.

При разработке обучающего комплекса была выбрана тема «Финансово-коммерческие операции». В современном мире практически любая деятельность связана с денежными операциями. Мы можем говорить как о большой фирме, так и об отдельном человеке, который каждый день сталкивается с денежными отношениями. Цель нашей работы смоделировать реальные ситуации финансово – коммерческой деятельности, которые позволят понять и принять правильное решение в конкретном случае, потому что без определенных знаний и навыков о финансово – коммерческих операциях довольно трудно понять, как посчитать ту или иную процентную ставку, срок ссуды или сумму, которую необходимо будет выплатить через какой-то промежуток времени.

В рамках научно – исследовательской работы разработан обучающий программный комплекс для студентов экономических специальностей, в котором были реализованы следующие финансово – коммерческие операции:

- операции по схеме простых процентов;
- операции по схеме сложных процентов;
- операции дисконтирования;
- финансово-товарные потоки;
- инфляция;
- операции с облигациями;
- операции с акциями;
- модели сравнения операций.

Программный комплекс состоит из нескольких модулей, его структура представлена на Рис.1.

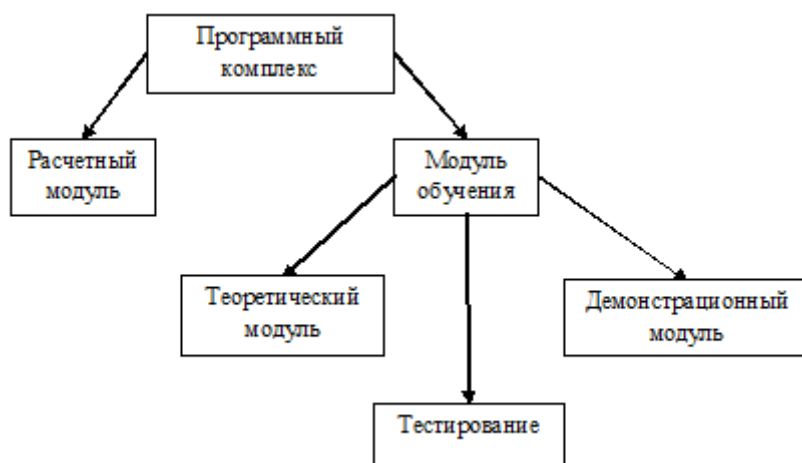


Рисунок 1 – Структура программного комплекса

Преимущества комплекса состоит в том, что с ним может работать человек, который может совсем не иметь никаких представлений о финансово – коммерческих операциях. Сначала он может изучить теорию по данной теме в теоретическом модуле, далее в демонстрационном модуле будут представлены ролики для наглядного представления пройденного материала. После этого можно будет проверить полученные знания, перейдя самотестирование. Тест предусматривает применение учебного материала для решения практических и теоретических задач, может использоваться как на любом уровне учебной деятельности. Кроме того, тест с выбором ответа связан с деятельностью в ситуации выбора приемлемого и отклонения неприемлемого, с чем каждый человек встречается на каждом шагу. Тесты позволяют в кратчайший срок проверить знания больших групп учащихся, выявить пробелы при изложении учебного материала, применить методы математической статистики для оценки степени его усвоения всеми испытуемыми и т.д. При прохождении тестов в данном программном комплексе результатом будет не только балл, но также будут показаны вопросы, в которых были выбраны неправильные ответы. Если человеку необходимо только посчитать какую-то интересующую его величину, он может сразу перейти в расчетный модуль и получить нужный результат, введя исходные данные. В настоящий момент реализован расчетный модуль, и начата работа над реализацией остальных модулей.

Список литературы

1. Мецкевич А.А. Финансовая математика. – М:ОЛМА – ПРЕСС Инвест: Институт экономических стратегий, 2003. – 128 с.
2. Фомин Г.П. Математические модели и модели в коммерческой деятельности. – М:Финансы и статистика, 2005. – 616 с.
3. Основы финансовой математики [Электронный ресурс]: - Режим доступа: <http://becmology.blogspot.com/2011/02/blog-post.html>.
4. Бочаров П.П., Касимов Ю.Ф. Финансовая математика. – М:Гардарики, 2002. – 624 с.
5. Применение информационных технологий в сфере образования и обучения [Электронный ресурс]: - Режим доступа: <http://www.snfpo.ru/help/articles/a1.htm>
6. Тестирование как один из эффективных способов проверки знаний учащихся [Электронный ресурс]: - Режим доступа: <http://nsportal.ru/shkola/biologiya/library/testirovanie-kak-odin-iz-effektivnyh-sposobov-proverki-znaniy-uchashchihhsya>.

РАЗРАБОТКА СЕТЕВОГО УНИВЕРСАЛЬНОГО ОРГАНАЙЗЕРА СТУДЕНТА

Еремин Я.Б., Путинцев Р.В. – студенты, Боровцов Е.Г. – к.т.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Программа-органайзер — компьютерная программа, относящаяся к прикладному программному обеспечению и предназначенная для накопления пользовательской информации, а затем оперативного поиска по ней, организации дел и контроля за их выполнением, отслеживания определённых пользователем событий. Является одной из форм персонального органайзера [1].

Функции типичной компьютерной программы-органайзера связаны с обеспечением работы следующих подразделов:

- календарь;
- менеджер контактов (пользовательская адресно-телефонная книга);
- записная книжка и листки-заметки (аналог бумажных листовок-листочков);
- события, привязанные к определенной дате и времени (например встречи);
- планировщик задач (заданий) для контроля за их выполнением;
- напоминатели-будильники об определённых пользователем событиях.

Некоторые программы-органайзеры могут как не иметь какого-либо из перечисленных подразделов, так и обеспечивать дополнительную функциональность, например, позволять работать с электронной почтой, таким образом выполняя функции почтового клиента.

Онлайн-органайзер отличается от программы-органайзера тем, что не устанавливается на локальном компьютере, но эта группа органайзеров не работает при отсутствии интернета.

Разрабатываемая программа является специализированным ПО, предназначена для просмотра, составления, и редактирования расписания занятий, а так же планирования учебной деятельности на срок до семестра. Вся информация за неделю представлена в наглядном виде, на одном экране — использовать разрозненные инструменты больше нет необходимости.

Исходя из проведенного обзора аналогов и анализа отзывов, можно сделать вывод, что среднестатистическому студенту требуется дешевое (а лучше бесплатное) приложение, работающее на широком спектре операционных систем. В то же время, такие инструменты как органайзер должны иметь высокую доступность и надежность, например для просмотра расписания при отсутствии доступа в сеть интернет. Не смотря на то, что Робот Пашка [2] работает на множестве операционных систем, это преимущество нивелируется фатальным недостатком, потому как веб сервис не удовлетворяет вышеуказанным требованиям. Студенту необходима возможность резервного копирования, страхующая его от отказа оборудования, и позволяющая поделиться копией расписания и задач с одноклассниками.

Органайзер является средством управления временем, поэтому скорость составления расписания является важным фактором, который влияет на заинтересованность пользователя и порог вхождения. Если порог слишком большой, продолжение использования программы окажется под вопросом. Недопустимо заставлять пользователя читать инструкции перед началом использования, как того требует DKTimeTable [3]. Для упрощения работы с несколькими устройствами необходимо обеспечить синхронизацию. Это позволит ускорить работу с приложением и поможет избежать ввода повторяющихся данных.

Интерфейс такого инструмента должен быть приятным, наглядным и интуитивно понятным, без отвлекающих деталей. Это обосновано тем, что подобные программы используются каждый день. Примером такого продукта может служить iStudiez [4], с учетом направленности на российских студентов. При этом широкий набор функций не должен усложнять или загромождать графический интерфейс.

Для упрощения планирования, органайзер должен иметь развитую систему задач, включающую возможность работы с сериями заданий. Это позволит планировать работу на семестр, легко и изящно отслеживать собственную успеваемость и правильно расставлять приоритеты. Такая функциональность повышает продуктивность работы, сокращает время на выбор задачи для выполнения, препятствует возникновению задолженностей из-за рассеянности студента.

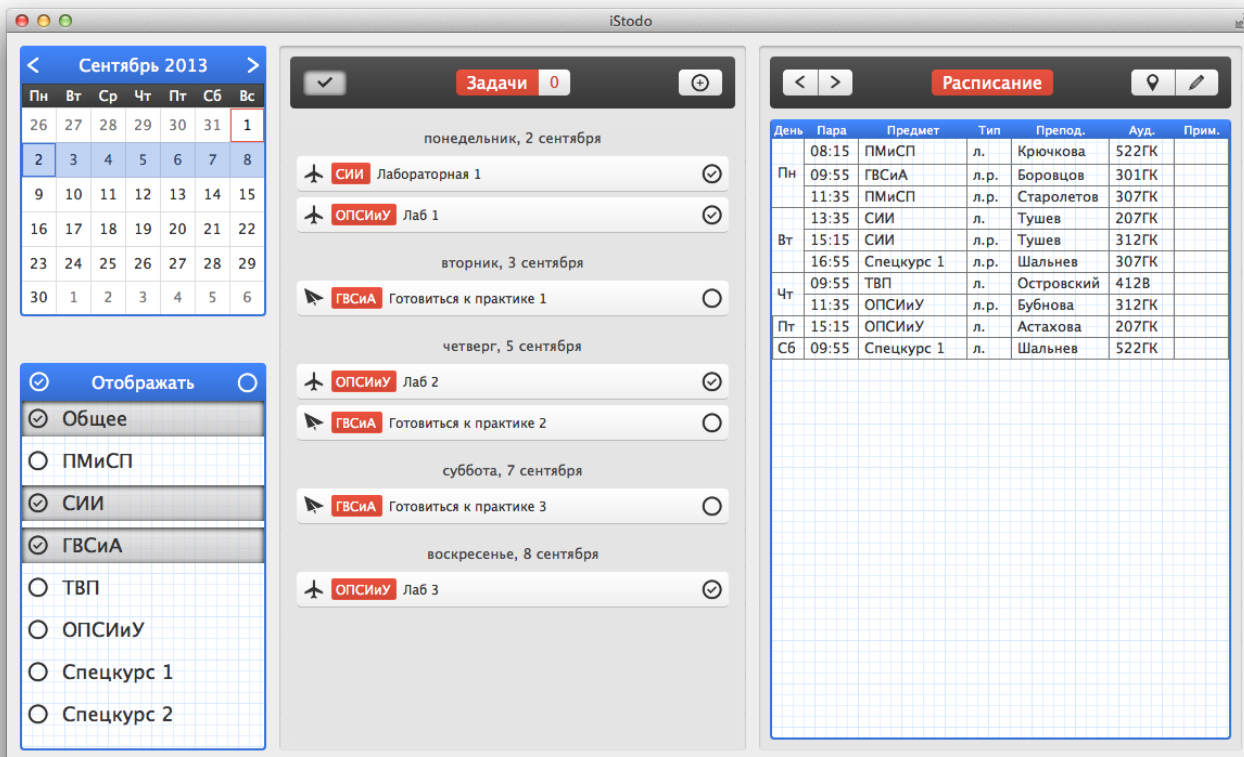


Рисунок 1 – основное окно программы

Ключевые особенности разрабатываемого ПО:

- Визуальный редактор расписания;
- Поддержка расписания из отличающихся (четные/нечетные) недель;
- Возможность экспорта расписания и задач в формате iCal:
 - Позволяет просматривать расписание в календаре мобильных устройств;
- Создан специально для российских студентов
- Мощная система задач:
 - работа с сериями заданий;
 - фильтрация;
 - приоритеты;
 - сроки выполнения.
- Поддержка основных ОС:
 - Windows;
 - OS X;
 - Linux.
- Возможность создания резервных копий:

Позволяет перенести всю информацию на другой компьютер, поделиться данными с одноклассниками;

- Адаптирован к экранам нетбуков(1024*600 px).

Запланированные возможности программного комплекса

Планируется версия для мобильных устройств, которая призвана помочь студентам пользоваться органайзером в любое время, вне зависимости от местоположения. Основной особенностью приложения может стать тесная интеграция с уже существующим личным кабинетом студента АлтГТУ. В таком случае возможна автоматическая загрузка и обновление расписания занятий, что позволит пользователю начать работу с программой незамедлительно. Другими положительными следствиями интеграции станут:

- Просмотр расписания экзаменов:
 - на календаре
 - в специальном окне, в виде списка
- Просмотр оценок за аттестации и по итогам сессии
- Возможность передавать важную информацию(например об отмене занятий) напрямую в пользовательское приложение

Кроме того, может быть реализована версия для преподавателей со схожим интерфейсом.

Список литературы

1. Персональный органайзер [Электронный ресурс]: материалы сайта – Режим доступа – http://ru.wikipedia.org/wiki/Персональный_органайзер – свободный.
2. Блог компании Робот Пашка [Электронный ресурс]: материалы сайта – Режим доступа – <http://habrahabr.ru/company/robotpashka/> – свободный.
3. Сайт программы DKTimeTable [Электронный ресурс]: материалы сайта – Режим доступа – <http://www.dktimeable.nm.ru/> - свободный.
4. iStudiez Pro Guide [Электронный ресурс]: материалы сайта – Режим доступа – <http://istudentpro.com/guide/index.html> – свободный.
5. Стивенс У.Р., Феннер Б., Рудофф Э.М. С80 UNIX: разработка сетевых приложений. 3е изд- СПб.: Питер, 2007. - 1039с.: ил.
6. Шлее М. Qt4.8. Профессиональное программирование на C++. - СПб. БХВ-Петербург. 2010, - 896 с.: ил. + DVD

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СИСТЕМЫ ОПРЕДЕЛЕНИЯ ТОЧНЫХ КООРДИНАТ ДВИЖУЩЕГОСЯ ОБЪЕКТА С ПРИМЕНЕНИЕМ СПУТНИКОВЫХ РАДИОНАВИГАЦИОННЫХ СИСТЕМ

Зыков Р.В. – студент, Поддубный В.И. – д.т.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Одним из наиболее эффективных способов повышения производительности труда в снижения затрат в сельском хозяйстве является применение систем точного земледелия, использующих спутниковые навигационные системы (GPS/ГЛОНАСС). Применение этих систем позволяет обеспечить рабочее движение сельскохозяйственных машин по задаваемой траектории без участия водителя с высокой точностью. Это позволяет уменьшить затраты на обработку участка земли, расход топлива, снизить затраты механизатора на управление.

Широкое распространение получила продукция компаний Тримбле, Клаас и других зарубежных компаний. Следует отметить, что данные системы эффективно функционируют только при наличии дифференциального сервиса, обеспечивающего получение текущих координат сельскохозяйственной машины с достаточной для данного вида работ точностью. В странах Европы и северной Америки используется бесплатный дифференциальный сервис WAAS и EGNOS, обеспечивающие достаточную точность для большинства видов выполняемых полевых работ. Для проведения работ с большой точностью применяются специальные платные дифференциальные сервисы. Широкое распространение получил спутниковый сервис Qmnistar, позволяющий получить текущие координаты с точностью до сантиметров.

На территории России, особенно в ее восточной части, европейские сервисы не работают. Поэтому приходится применять платные дифференциальные сервисы, которые являются дорогостоящими для российских сельхозпроизводителей. Также стоимость импортных систем точного земледелия очень высока и в большинстве случаев они не могут быть использованы на российской сельскохозяйственной технике вследствие несогласованности параметров (в частности тугое рулевое управление отечественных колесных тракторов).

Исходя из этого, представляется целесообразным разработка отечественной системы, сопоставимой по качественным показателям работы с импортными системами и более привлекательными по цене. Одним из наиболее важным для эффективной работы системы является обеспечение достаточной точности определения текущих координат колесной сельскохозяйственной машины.

Для решения данной проблемы был рассмотрен так называемый DGPS-сервис (Differential GPS), обеспечивающий получение дополнительных дифференциальных поправок, уточняющих местоположение GPS-приемника. Наиболее эффективным и простым в реализации является метод коррекции координат.

Данный метод предполагает, что корректируются координаты потребителя, определенные им по сигналам навигационного искусственного спутника Земли, при стандартном режиме использования сетевой спутниковой радионавигационной системы. Корректирующая информация формируется на корректирующей базовой станции (координаты центра антенны которой заранее известны с высокой точностью) путем сопоставления эталонных координат с координатами, вычисленными в результате стандартного навигационного сеанса, проводимого на корректирующей базовой станции с помощью оборудования наивысшего класса точности либо путем измерения координат на протяжении промежутка времени и установки средних значений, которые и принимаются за эталонные. Полученные дифференциальные поправки (как разности истинных и измеренных координат базовой станции) передаются в составе корректирующей информации потребителю, который и уточняет по ним свои координаты.

Алгоритм этого метода может быть записан в виде формулы:

$$\Delta X = X_{КСЭ} - \tilde{X}_{КС}; \quad X_{П} = \tilde{X}_{П} + \Delta X \quad (1)$$

где $\tilde{X}_{КС}$, $\tilde{X}_{П}$ – векторы оценок координат контрольно-корректирующей станции и потребителя по сигналам спутниковой радионавигационной системы;

$X_{КСЭ}$ – вектор эталонных координат контрольно-корректирующей станции;

ΔX – вектор поправок;

$X_{П}$ - вектор уточненных координат потребителя.

Спутник GPS/ГЛОНАСС

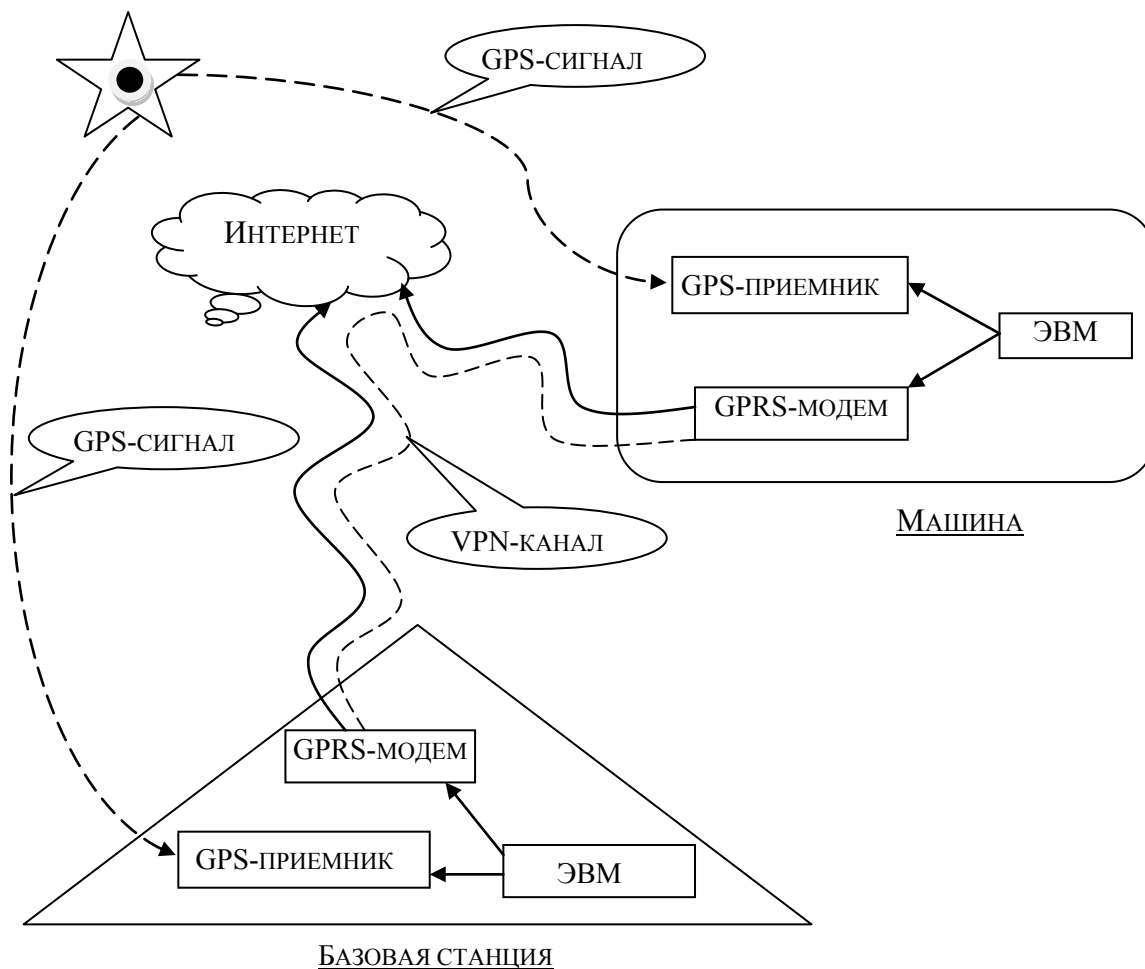


Рисунок 1 – Схема работы системы передачи дифференциальных поправок

В качестве оборудования для приема GPS/ГЛОНАСС сигнала используется отечественная разработка Ижевского радиозавода - отладочное средство на основе приемника МНП-М7. Приемник базовой станции настроен на частоту приема GPS-координат 2 Гц, приемник машины – 10 Гц. Подключение к компьютеру осуществляется через порт USB и с помощью преобразователя FTDI, являющегося одним из компонентов отладочного средства и специальных драйверов, в операционной системе эмулируются два виртуальных COM-порта, один из которых можно использовать для принятия данных по протоколу MNP-Binary.

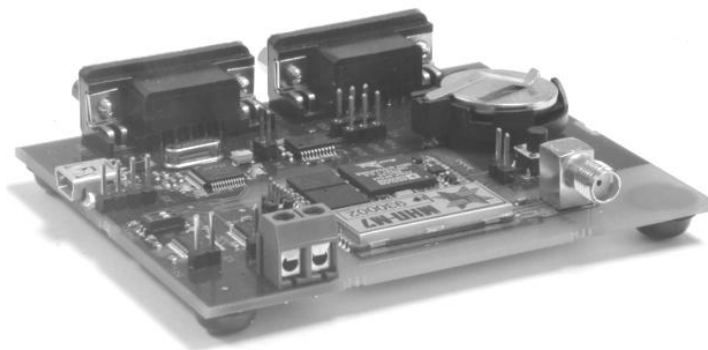


Рисунок 2 – Внешний вид отладочного средства с приемником МНП-М7

Чтобы организовать передачу поправок с базовой станции на движущиеся машины, используются 3G/GPRS USB-модемы. После подключения их к интернету, настраивается виртуальная частная сеть (VPN), в которой по протоколу UDP ведется широковещательная рассылка дифференциальных поправок с частотой 2 Гц.

Выбор языка программирования

Для реализации данной системы был выбран язык программирования C++, т.к. он обеспечивает лучшие способы взаимодействия с устройствами по низкоуровневым бинарным протоколам. Также были использованы компоненты библиотеки Qt версии 4.8.5, в частности QtSerialPort для передачи данных по виртуальному последовательному порту, QtUdpSocket для сетевого взаимодействия и передачи поправок с базовой станции. Система сигналов-слотов библиотеки широко использовалась при построении пользовательского интерфейса и взаимодействии между объектами в программе.

Перспективой развития данной системы является модуль автоматического управления электроприводом руля при помощи устройства аналогового ввода-вывода, блока управления, электродвигателя и датчика угла поворота колес.

Список литературы

1. Шебшаевич В.С. Сетевые спутниковые радионавигационные системы [Текст] / В.С. Шебшаевич – М.: «Радио и связь», 1993. – 408 с.
2. Qt Project. Официальный сайт [Электронный ресурс]. – Режим доступа: <http://www.qt-project.org>
3. ОАО «Ижевский радиозавод». Официальный сайт [Электронный ресурс]. – Режим доступа: <http://www.irz.ru>
4. Приемник навигационный МНП-М7. Руководство по эксплуатации ЦВИЯ.468157.113 [Электронный ресурс]. – Тестовые данные (1,34Кб). – Ижевск: ИПЗ-Литера, 2010. – 1 электрон. опт. диск (CD-ROM).

АЛГОРИТМ СЕМАНТИЧЕСКОГО ПОИСКА В БОЛЬШИХ ТЕКСТОВЫХ КОЛЛЕКЦИЯХ

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

1. Введение

В наше время, в условиях большого и стремительно растущего объема информации, актуальна задача поиска в больших текстовых коллекциях [1]. Одним из вариантов поиска является семантический поиск, т.е. поиск с точки зрения содержащейся в тексте информации [2,3,4]. Среди наиболее популярных систем семантического поиска можно выделить Google, SearchMonkey, Powerset, Freebase и AskNet. Однако они имеют определенные недостатки, такие как: применение семантики лишь для незначительного улучшения результатов поиска, ограничение на длину запроса, снижение качества поиска с увеличением поискового запроса. Кроме того большинство из таких поисковых систем работают только с английским языком.

Изменчивость синтаксических конструкций и вариативность лексики естественных языков, разнообразие стилей изложения материала существенно усложняют решение данной задачи.

2. Семантический поиск

В данной работе представлен результат разработки системы семантического поиска для больших текстовых коллекций на русском языке. Ключевая особенность предлагаемой системы - снятие ограничений на величину поискового запроса.

Исходными данными для поиска являются текстовые коллекции и запрос пользователя, который также представляет собой текстовую коллекцию. Исходя из предположения, что большая текстовая коллекция в общем случае неоднородна по своему содержанию и при поиске интересна лишь ее определенная часть, текст можно разделить на фрагменты. Как правило, такие фрагменты - это страницы, абзацы или наборы из нескольких предложений. Фрагменты будем называть «окнами», таким образом, задача сводится к поиску одного окна или группы окон.

Для каждого окна запроса и поисковых коллекций строится граф семантических связей - «семантический граф». Семантический граф представляет собой ориентированный граф, вершинами которого являются слова русского языка, представленные в нормальной форме, а ребра характеризуются весом и типом семантической связи. Направление ребра зависит от типа семантической связи, например отношение объект - действие, объект - свойство, действие - время. Для построения семантического графа каждое предложение из окна коллекции последовательно обрабатывается семантическим анализатором. Для каждого очередного предложения строится семантический граф G_{new} . Все ребра семантического графа G_{new} имеют вес равный единице. На каждой итерации семантический граф G_i предыдущей итерации объединяется с графом G_{new} обрабатываемого предложения, причем веса ребер нового графа G_{i+1} вычисляются как сумма весов ребер графов G_i и G_{new} , что приводит к увеличению весов однотипных ребер. После этого результирующий семантический граф G_{i+1} используется для следующей итерации. После обработки всех предложений окна, веса ребер полученного графа нормируются путем деления на общее количество семантических связей окна. Из графа удаляются ребра с весом меньше некоторого достаточно малого δ , для отсекаемых семантически мало значащих ребер.

Далее необходимо подсчитать величину коэффициента соответствия семантического графа запроса и семантического графа окна. Задача поиска наибольшего общего подграфа, даже с учетом совпадения не только вершин, но и типов ребер - это NP- полная задача. Кроме того один и тот же смысл содержится в текстах разного стилистического оформления, вершины графа запроса и графа окна коллекции могут быть не только синонимами, но и каким-либо образом связанными по смыслу словами, могут содержать обобщающие сведения или только частичную информацию. Например, при обработке запроса, связанного с поиском текста «хордовые животные, обитающие в тайге» к хордовым, обитающим в тайге, в том числе относятся и зайцы тайги. Очевидно, улавливается связь: «хордовые \rightarrow зайцы». В то же время между хордовыми и зайцами не должно быть полного отождествления, т. к. хордовые - это не только зайцы.

Для поиска связанных по смыслу слов был использован словарь, полученный методом автоматической обработки краткого толкового словаря, в котором представлен перечень слов в нормальной форме [5]. Каждому слову сопоставлен набор слов, связанных с ним ассоциативной, синонимичной и т.д. связью. Таким образом, словарь представляет собой направленный граф $G_{word} = (V_{word}, U_{word})$, где вершины V_{word} - это слова в нормальной форме, а ребра U_{word} имеют действительные весовые коэффициенты от 0 до 1. Назовем граф G_{word} графом справочника. За коэффициент связанности слов a_k и a_m - вершин семантического графа запроса $G_{request}$ и семантического графа окна коллекции G_{text} возьмем произведение весов от таких же слов до общего предка в графе справочника. При совпадении слов данный коэффициент будет равен 1, иначе будет принадлежать промежутку $[0;1]$. Фрагмент графа G_{word} представлен на рисунке 1.

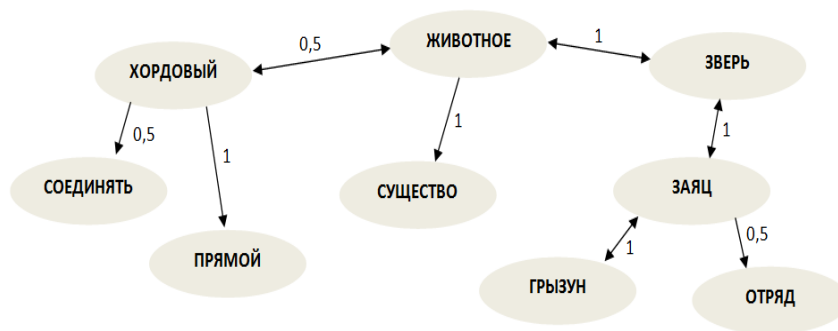


Рисунок 1 – Граф справочника

На следующем этапе ищем наилучшее совпадение семантического графа запроса с графом окна. Значение коэффициента релевантности (соответствия) при совпадении графов рассчитываем по формуле 1:

$$S = \sum_{k=1}^n d_k * r_k * l_k * m_k, \quad (1)$$

где n – количество совпавших ребер графа запроса и окна, k – индекс соответствующего ребра, d_k , r_k – произведение весов ребер в графе справочника от слова запроса и слова окна до общего предка соответственно, l_k – вес ребра k семантического графа запроса, m_k – вес ребра k семантического графа окна. Вариантов совпадения подграфов много, но нас интересует максимальное значение коэффициента соответствия. Определив максимальное значение по всем окнам текстовой коллекции, мы получим общее значение – коэффициент релевантности запроса и текстовой коллекции.

3. Тесты и результаты

Для оценки полученной системы были отобраны текстовые коллекции большого объема, удовлетворяющие одному и тому же запросу к поисковой системе google.ru. Поисковый запрос строился исходя из содержания определенных коллекций. В зависимости от настроек системы были получены различные значения коэффициента соответствия между поисковым запросом и коллекцией. Однако для коллекций, по содержанию которых строился запрос или коллекций аналогичного содержания, значение коэффициента минимум на порядок превосходило значение коэффициента других коллекций, не похожих по содержанию.

Временные затраты на обработку коллекции в зависимости от количества предложений в запросе и окне коллекции представлены на рисунке 2.

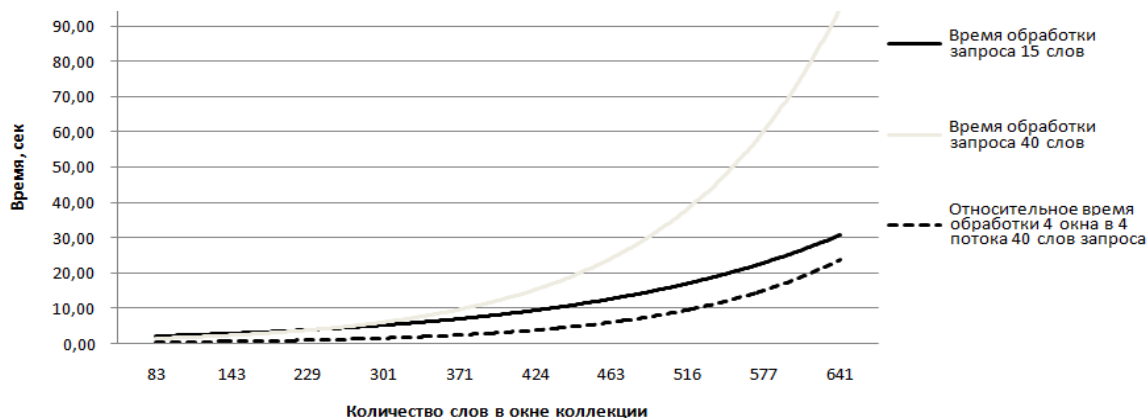


Рисунок 2 – Временные затраты

Так же было проведено тестирование с добавлением «шума», т.е. добавлением в текст окна предложений из других текстовых коллекций, не соответствующих запросу. Результаты отклонения коэффициента релевантности приведены на рисунке 3.

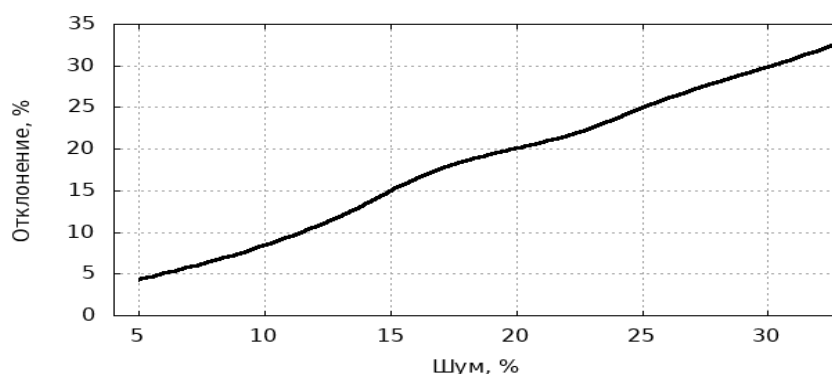


Рисунок 3 – Отклонение коэффициента релевантности из-за шума

Эксперименты показали, что отклонение коэффициента релевантности линейно возрастает с увеличением шума, т.е. алгоритм расчета коэффициента корректен и действительно отображает соответствие между окном коллекции и поисковым запросом.

К сожалению, предложенная реализация алгоритма имеет значительное увеличение времени обработки с ростом размеров окна и запроса. Для преодоления этого недостатка текстовую коллекцию и запрос следует разделять на окна оптимального размера с точки зрения времени обработки. Параллельная обработка окон текстовой коллекции может на многопоточных и многопроцессорных системах выполняться параллельно, что значительно повышает скорость обработки запроса.

Список литературы

1. Hannah Bast, Marjan Celikik Efficient Fuzzy Search in Large Text Collections // ACM Transactions on Information Systems, 2010.
2. Mathieu d'Aquin, Enrico Motta Watson, more than a Semantic Web search engine // IOS Press Amsterdam, 2011.
3. Elbedweihy K., Wrigley S.N., Ciravegna F., Reinhard D., Bernstein A. Evaluating Semantic Search Systems to Identify Future Directions of Research // Second International Workshop on Evaluation of Semantic Technologies, 843, page 25-36, 2012.
4. Tsoumakas G., Laliotis M., Markantonatos N., Vlahavas I. Large-Scale Semantic Indexing of Biomedical Publications at BioASQ // BioASQ Workshop, 2013.
5. Крайванова В.А., Кротова А.О., Крючкова Е.Н. Построение взвешенного лексикона на основе лингвистических словарей // Материалы Всероссийской конференции с международным участием ЗОНТ-2011, Т.2, Новосибирск, стр. 32-38, 2011.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ СУДЕЙСТВА И ПРОВЕДЕНИЯ СОРЕВНОВАНИЙ В АВИАМОДЕЛЬНОМ СПОРТЕ

Смирнова А.Ю. – студент, Крючкова Е.Н. – к.ф.-м.н., профессор,
Тамплон А.В. – технический директор компании «Enterra»*
Алтайский государственный технический университет (г. Барнаул)
ООО «Энтерра Софт» * (г. Барнаул)

В последние десятилетия компьютерная техника и технологии развиваются крайне интенсивно и, как следствие, они играют все более важную роль у нашей жизни. Они привнесли много нового практически во все сферы человеческой деятельности, в том числе и в организацию спортивных соревнований.

В наше время трудно представить международное или национальное спортивное событие без соответствующего компьютерного сопровождения. В этой области компьютерные системы используются для составления турнирных таблиц, планирования состязаний, ввода, обработки и публикации результатов и для автоматизации других действий. Для крупных соревнований разрабатываются заказные системы с узкой специализацией под конкретный вид спорта.

Соревнования по авиамодельному спорту представляют собой последовательное выполнение участниками упражнений, порядок которых оговаривается заранее. Для выполнения упражнений участники разбиваются на группы. За порядком выполнения участником упражнения следит хронометрист (один для каждого участника). Он отмечает время выполнения участником упражнения и, при нарушении правил соревнований, назначает штрафы.

Согласно правилам проведения соревнований по авиамодельному спорту в классе моделей F-3-K, в качестве компьютерной системы проведения соревнований используется программа F3KScore. Она обладает широким набором функционала для организации и проведения соревнований.

К сожалению, функционал F3KScore не включает в себя возможности получения результата участника в текущем соревновании. На данный момент, данные передаются от хронометристов главному судье в бумажном виде и вручную заносятся в программу. Это вызывает ошибки при измерении и внесении результатов и значительно усложняет порядок проведения соревнования.

Постановка задачи

1. Определить функциональности будущей программы
2. Подобрать инструментарий необходимый для реализации системы, на основе современных тенденций в области разработки программного обеспечения.
3. Составить диаграмму использования ПО пользователем и составить диаграмму классов.
4. Реализовать базовые функции программы.
 - А) Таймер для получения времени выполнения упражнения.
 - Б) Возможность обмена данными между будущей программой и F3KScore.
 - В) Возможность воспроизведения звуковых файлов с возможностью выбора языка.
 - Г) Возможность назначения штрафов.
 - Д) Просмотр предварительных результатов выполняемого упражнения
5. Осуществить установку и тестирование системы.

Решение

Для решения поставленной задачи была выбрана клиент-серверная архитектура приложения. Она представлена на рисунке 1.

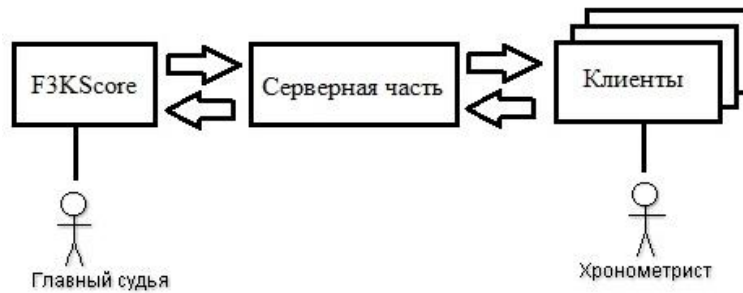


Рисунок 1 – Архитектура полученной системы

Принципы работы клиента и сервера представлены на рисунке 2.



Рисунок 2 – Принципы работы клиента и сервера

Был написан специализированный мобильный клиент, позволяющий вносить информацию о прохождении участником соответствующего упражнения. Информацию о текущем упражнении и участнике приложение получает с сервера, напрямую взаимодействующего с программой F3KScore. Данные о результатах также передаются на сервер, где проходят дальнейшую обработку. Клиентская часть реализована на базе платформы Android, что позволяет обеспечить требуемый уровень мобильности и гибкости использования.

Взаимодействие между клиентом и сервером происходит с использования протокола TCP, который позволяет обеспечить упорядоченные, надежные двусторонние потоки байтов.

Серверная часть приложения устанавливается на одном локальном хосте с F3KScore. Используя сгенерированные программой файлы Excel, она получает список упражнений и порядок выступления участников. По запросу клиента, она передает ему данные об упражнении и участнике, который будет проходить его с данным хронометристом. По окончании рабочего времени, серверная часть получает результаты выполнения упражнения и автоматически вносит их в программу F3KScore, где они могут быть проанализированы и обработаны главным судьей соревнований. Серверная часть приложения реализована на языке C# с использованием фреймворка .NET.

Взаимодействие между сервером и программой F3KScore организуется посредством обмена данными через системный буфер.

Результаты

В ходе данной работы было спроектировано и создано приложение, обладающее следующим функционалом:

- Получение и контроль текущего времени выполнения упражнения с помощью таймера
- Получение промежуточных результатов выполнения упражнения
- Обмен данными с F3KScore
- Воспроизведение звуковых файлов на разных языках

Список литературы

1. AndroidDevelopers [Электронный ресурс] – Режим доступа: <http://developer.android.com>
2. F3KScore - software program for managing F3K contests [Электронный ресурс] – Режим доступа: <http://olgo1.com/F3KScore/>
3. Synchronous Server Socket Example [Электронный ресурс] – Режим доступа: [http://msdn.microsoft.com/en-us/library/6y0e13d3\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/6y0e13d3(v=vs.110).aspx)
4. ProfessorWeb. Клиент-серверное приложение на потоковом сожете TCP [Электронный ресурс] – Режим доступа: http://professorweb.ru/my/csharp/web/level3/3_2.php
5. Meier R. Professional Android 4 Application Development – IN: John Wiley & Sons, Inc., 2012. – 817 с.

СРЕДСТВО ПРОФИЛИРОВАНИЯ ДЛЯ АНАЛИЗА ПРОБЛЕМ ПРОИЗВОДИТЕЛЬНОСТИ В АБС RS-BANK

Бобб А.Я., Хатников А.С. – студенты, Троицкий В.С. – к.ф.-м.н., доцент
Алтайский государственный технический университет (г. Барнаул)

RS-Bank v.5.5 — интегрированная версия банковской системы, ориентированная на потребности коммерческих банков категории SBM (Small and Medium Business). Характеризуется широкой базовой функциональностью, открытостью системы для самостоятельного развития, тиражностью системы и невысокой стоимостью. Функционирует в трехзвенной архитектуре (сервер БД, приложений и рабочие станции пользователей). В процессе эксплуатации системы, необходимо осуществлять мониторинг ее работоспособности и производительности. Штатных средств в АБС для этого нет. Процесс оценки ее работоспособности и производительности выполняется методом измерения времени выполнения различных операций с помощью обычного секундомера. Подобные технологии используются и разработчиками АБС, если судить по отчетам о тестировании программных продуктов размещенным на их сайте. С таким подходом невозможно ответить на вопрос «Быстро ли работает АБС в текущий момент?» и все жалобы пользователей о том, что система очень медленно работает, остаются без ответа. Т.е. даже если измерить время выполнения каждой процедуры, его не с чем сравнить. Нет данных сколько времени эта процедура работала вчера, месяц назад или год. Соответственно и искать (а тем более устранить) причины медленной работы системы не предоставляется возможным. Нами предлагается к реализации ПО, которое в автоматическом режиме будет проводить мониторинг работоспособности и производительности АБС RS-Bank. Целесообразно разделить ПО на две части, сбор данных и их анализ. ПО для сбора данных будет периодически запускать заданные операции в RS-Bank, рассчитывать время их выполнения и сохранять полученные результаты в БД. А ПО для анализа - извлекать накопленные данные и отображать их в удобочитаемом виде. Функциональность разрабатываемого ПО со временем будет расширена. Добавится автоматическое выявление причин влияющих на производительность и работоспособность системы. Это потребует доработки ПО сбора данных и разработки нового модуля для поиска причин на основе аппарата анализа временных рядов. В результате мы получим ПО, которое в автоматическом режиме проводит мониторинг работоспособности и производительности АБС и в случае выхода контролируемых параметров за допустимые границы находит причины, которые привели к этому.

ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА ЭФФЕКТИВНОЙ РАЗРАБОТКИ ИНТЕРАКТИВНОГО ОБУЧАЮЩЕГО КУРСА

Уваров Д.А., Силин Д.И. – студенты, Крючкова Е.Н. – к.ф.-м.н., профессор
Алтайский государственный технический университет (г. Барнаул)

В современном мире вычислительная техника используется практически во всех сферах деятельности человека, поэтому пользователь должен обладать соответствующими навыками обращения с программным обеспечением (ПО). Человек не рождается с заложенным знанием как работать с тем или иным ПО, поэтому каждому приходится проходить этап обучения. Этот путь можно пройти по-разному: экспериментальное изучение, изучение с помощью инструктора, использование материалов видео курсов, использование интерактивного курса.

Особый интерес вызывает процесс построения интерактивных курсов обучения, так как это наиболее эффективный метод. Рассмотрим набор требований к обучающей программе, содержащий критически важные элементы:

- окружение пользователя должно быть максимально приближено к тому, как выглядит изучаемая программа на его компьютере;
- система должна контролировать и ограничивать действия пользователя для того, чтобы не потерять прогресс и, чтобы пользователь не заблудился в сложной иерархии ещё не изученных элементов управления;
- система должна иметь возможности подсказать пользователю необходимые данные.

Наиболее очевидным способом построения обучающего контента является разработка специального программного продукта для каждой программы. При наличии исходных кодов программы обучающая программа может иметь неограниченные возможности. Однако такая разработка имеет слишком большую стоимость, сопоставимую со стоимостью разработки основного продукта. В данной работе предлагается способ построения инструментальной системы для генерации обучающих курсов, основанной на обработке сообщений операционной системы (ОС).

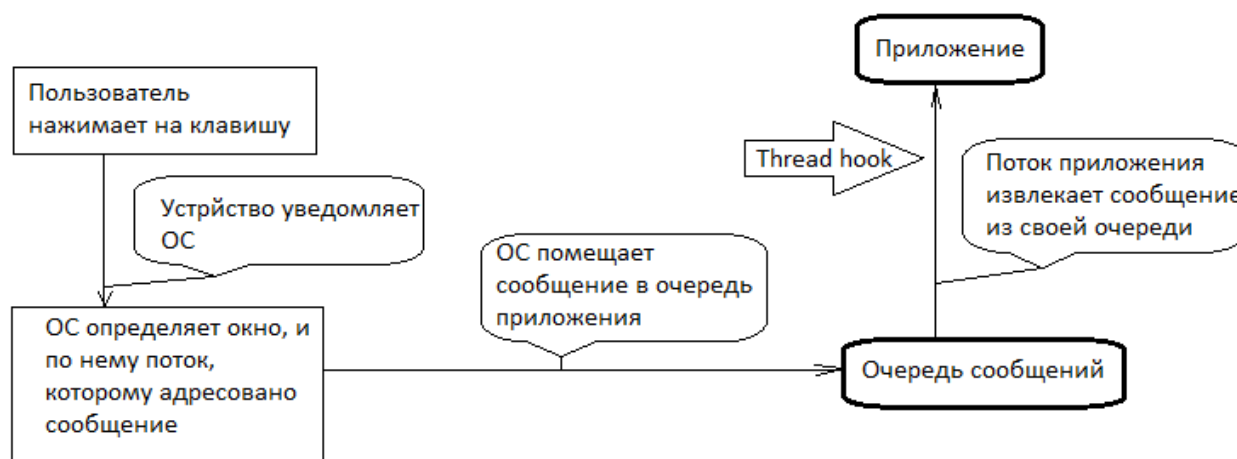


Рисунок 1 – Схема взаимодействия пользователя с приложением в ОС Windows

Любая программа выполняется в среде операционной системы. Именно ОС сообщает приложению о сигналах, поступающих с устройств ввода. Конечно это позволяет вмешаться в процесс воздействия пользователя на программу, однако события ОС на этом уровне слишком примитивны чтобы их можно было классифицировать с точки зрения функционального смысла. Следующие места «внедрения» и информативность сообщений в них зависит от ОС и от архитектуры приложения. Рассмотрим особенности передачи сигналов ввода в приложение, основанное на стандартных элементах ОС, в Windows.

Действия пользователя вызывают события в ОС, которые рассылаются к соответствующим приложениям. Каждый элемент управления - кнопка, поле для ввода или что-то иное, является окном с точки зрения операционной системы. Таким образом, когда сообщение о событии помещено в очередь, в нём находится информация об окне(элементе), которому оно адресовано. Windows предусматривает несколько возможностей установок ловушек (hook) на пересылаемые сообщения, различающиеся по положению в архитектуре, типу обрабатываемых сообщений и возможностью их модифицировать. Для целей построения обучающей системы, авторов этой работы заинтересовал хук `WH_GETMESSAGE`. Архитектура приложения указана на рисунке 1. Функция-обработчик срабатывает каждый раз, когда поток (на который установлена ловушка) обращается к ОС для получения следующего сообщения из очереди.

Эта функция может модифицировать сообщение любым образом. Разрабатываемая авторами инструментальная база имеет архитектуру, представленную на рисунке 2.

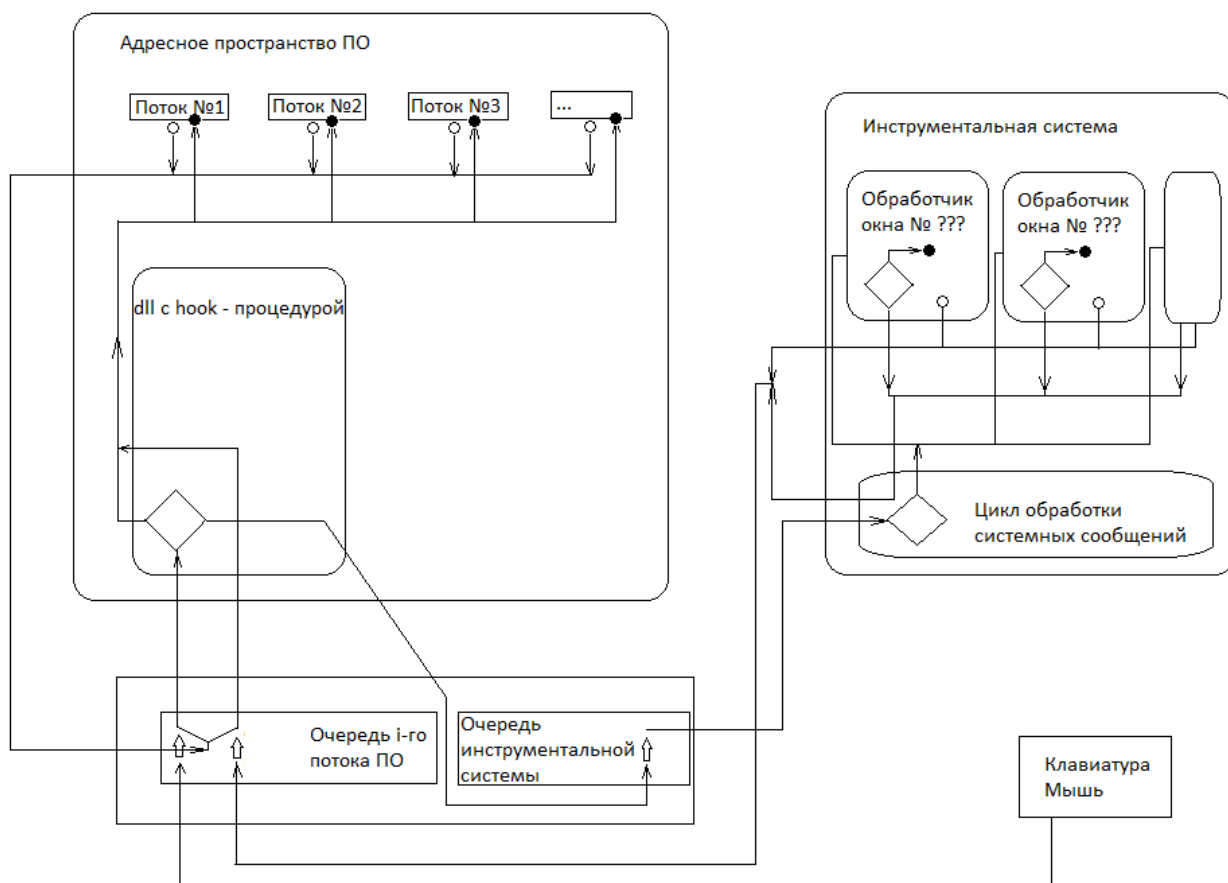


Рисунок 2 – Архитектура инструментальной системы

Система работает следующим образом: установленная функция-ловушка перенаправляет все события ввода приложению на внешний модуль, где некоторый обработчик, заданный для конкретного окна в конкретный момент времени, определяет, что следует сделать с этим сообщением (изменить, пропустить или заблокировать). Общение hook модуля с инструментальной системой так же происходит с помощью системных сообщений.

Рассмотрим этот процесс подробнее. В случае, если обработчик решает пропустить сообщение, он посылает программе полученное сообщение ещё раз, но уже от своего имени. Это сообщение уже не перенаправляется. Так как модуль и приложение общаются через системную очередь, порядок сообщений не нарушается. Сообщение содержит четыре параметра: идентификатор окна-получателя, тип сообщения и два параметра. Для пересылки полной информации другому окну необходимо два сообщения, т.к. первые два параметра не могут переносить информацию. Сообщения разных потоков отправляются с учётом синхронизации таким образом, что если удалить все сообщения от ОС, то обе половинки одного сообщения отправленного приложением лежат в очереди модуля последовательно. При необходимости обработчик может инициировать передачу команд приложению.

Предлагаемый в данной работе инструментарий предназначен решить две задачи:

- 1) Быстрое построение обучающего задания.
- 2) Контроль над выполнением обучающимся текущего задания.

Решение первой задачи

Назовём обучающий контент уроком, а разработчика - инструктором. Предоставим ему два окна – в одном (основном) будет приложение, к которому создаётся урок, а в другом (дополнительном) - инструментарий управления. На основании описанной архитектуры можно предложить следующую реализацию поставленной проблемы.

Инструктор осуществляет клик по любому элементу основного окна. При этом в программе не происходит соответствующего события, лишь необходимый элемент подсвечивается по контуру. В дополнительное окно выводится информация о типе объекта и о возможных действиях с ним. Например для кнопки «клик, двойной клик», для фалжка – «снять/установить», для поля ввода – набрать текст, соответствующий заданным правилам. После выбора действия создаётся «обработчик», который привязывается к соответствующему окну на этом шаге (при прохождении теста обучаемым именно этот «обработчик» будет определять допустимость действий пользователя). Очевидно, что задача – не всегда линейная последовательность действий. Разрешим инструктору создавать параллельные обработчики для нескольких окон. Назовём неупорядоченное множество действий – шагом урока. Когда создание шага окончено и инструктору необходимо перейти к следующему шагу, ему необходимо лишь уведомить инструментарий об этом (например, нажатием горячей клавиши или соответствующим кликом). После этого все обработчики текущего шага иницируют сообщения (которые ждут от пользователя) и посылают приложению от своего имени (события, посылаемые модулем приложению не перенаправляются). Таким образом ПО переходит в нужное состояние и инструктор может продолжить урок. Разумеется, инструктору нужна возможность добавлять комментарии к заданию и шагам, однако реализация этой функциональности не вызывает затруднений.

Таким образом становится простым составление урока по любой программе, использующей классические средства Windows для отображения интерфейса пользователя. В описанной схеме создание урока занимает время, не значительно превышающее сумму необходимого для прохождения урока и для составления задания к уроку и комментариев к ключевым шагам.

Решение второй задачи

Оно заложено в решении первой задачи. Созданные обработчики позволяют предоставить обучаемому ровно такую свободу действий, которая заявлена в требованиях к обучающему приложению. Так же обработчик, зная какое действие ожидается от пользователя, способен сообщить идентификатор нужного окна. Используя идентификатор можно послать системе сообщение об изменении фонового цвета элемента, тем самым указав обучаемому следующий шаг. Это целесообразно в случае нескольких неудачных попыток. Ещё одна интересная возможность – задавать некоторое множество предварительных шагов, которые выполняет обучающая программа до передачи действий пользователя, таким образом можно создавать предварительно настроенное окружение в целевой программе. Основная сложность на этом этапе – правильно восстановить идентификаторы окон при загрузке приложения повторно, т.к. каждый раз при регистрации элемента управления ОС выдаёт ему новый номер. Однако любой элемент можно идентифицировать достаточно точно, если построить граф (дерево) окон (у каждого окна есть окно-владелец, кроме корневых, которые принадлежат потоку), где каждый узел описывается типом элемента и некоторыми иными параметрами, набор которых зависит от типа (например от текста на кнопке).

Разработанная архитектура позволит изящно реализовать удобный и эффективный инструментарий для построения интерактивных обучающих курсов.

Список литературы

1. Джеффри Рихтер, Кристоф Назар. Windows via C/C++. /Русская Редакция - Питер, 2009. - 896 с.
2. Официальная документация MSDN. [Электронный ресурс]: Режим доступа: <http://msdn.microsoft.com/> - свободный.

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ПОДЗЕМНЫХ ВОД С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Шамин А.С. – студент, Кошелев К.Б. – к.ф.-м.н., с.н.с.*
Алтайский государственный технический университет (г. Барнаул)
Институт водных и экологических проблем СО РАН* (г. Барнаул)

В современных условиях развития вычислительной техники появляется возможность проведения сложных расчетов даже на персональных компьютерах. В следствие этого, задачи, которые несколько лет назад казались нерешаемыми, становятся доступны любому специалисту.

К таким задачам относится изучение и прогнозирование движения подземных вод. С этой задачей тесно связана также задача прогнозирования состояния земельных ресурсов, с целью предотвращения катастрофических явлений связанных с маловодиями или наводнениями. А в случаях, когда их невозможно избежать, принятия своевременных мер по снижению урона, причиненного данными явлениями.

Важность решения таких задач невозможно недооценивать, в особенности в виду событий произошедших летом 2013 года в Амурской области, когда в результате наводнения пострадал целый регион России.

В настоящее время актуальна задача изучения движения подземных вод. Главные требования к решению задачи заключаются в:

- ▲ полноте модели, выбранной для решения задачи;
- ▲ реализуемости решения на персональных компьютерах;
- ▲ возможности его применения на территории Алтайского края.

Целью данной работы является разработка программного продукта, предназначенного для численного моделирования движения подземных вод и использующего параллельные вычисления для повышения эффективности расчетов. Для того, чтобы приступить к реализации и разработке алгоритма программы, необходимо определить математическую модель, на основе которой, будет выбран метод решения.

В данной работе будет использована модель основанная на моделях, описанных в работах [1-3]. Для описания фильтрации грунтовых вод будет использоваться уравнение Буссинеска в следующем виде:

$$\mu \frac{\partial H}{\partial t} = \frac{\partial}{\partial x} \left(T_x(x, y, H) \frac{\partial H}{\partial x} \right) + \frac{\partial}{\partial y} \left(T_y(x, y, H) \frac{\partial H}{\partial y} \right) + V(x, y, t), \quad (1)$$

где x, y - декартовы координаты вдоль поверхности земли;

t - время;

$H(x, y, t)$ - абсолютная отметка уровня подземных вод;

T_x, T_y - гидравлические проводимости в направлениях x, y соответственно;

V - гидравлический поток на единицу объема, представляющий собой известные источники и стоки воды. Отрицательные значения представляют собой стоки, а положительные – вхождения воды.

$$T_x = \left\{ \begin{array}{l} k_x \cdot (z_{кр} + z_n), H \geq z_{кр} \\ k_x \cdot (H + z_n), z_n < H < z_{кр} \\ 0, H < z_n \end{array} \right\} \quad (2)$$

$$T_y = \left\{ \begin{array}{l} k_y \cdot (z_{кр} + z_n), H \geq z_{кр} \\ k_y \cdot (H + z_n), z_n < H < z_{кр} \\ 0, H < z_n \end{array} \right\}, \quad (3)$$

где k_x, k_y - коэффициенты фильтрации в направлениях x, y соответственно;

z_n - отметка подошвы слоя;

$z_{кр}$ - отметка кровли слоя.

$$\mu = \left\{ \begin{array}{l} \mu^*, H \geq z_{кр} \\ \mu^{**}, H < z_{кр} \end{array} \right\}, \quad (4)$$

где μ^* - коэффициент упругости горизонта;

μ^{**} - коэффициент гравитационной водоотдачи.

Граничные условия уравнения (2) в данной работе будем определять как:

$$\frac{\partial H}{\partial n} = 0, \quad (5)$$

где n - нормаль к границе области.

Вследствие того, что программный продукт должен быть основан на парадигме параллельного программирования, метод решения уравнения (1) необходимо выбирать таким образом, чтобы его можно было масштабировать на произвольное количество вычислительных единиц. Основываясь на требовании к масштабируемости метода, сформулируем еще одно требование: программный продукт должен реализовывать выбранный метод с использованием различных технологий параллельного программирования.

Таким образом, моделирование движения подземных вод может быть произведено на различных вычислительных устройствах, даже в рамках одной вычислительной системы (даже персонального компьютера). Это сделает возможным сравнение производительности устройств и выбор наиболее предпочтительного из них для решения задач численного моделирования.

Список литературы

1. Антонцев С.Н., Епихов Г.П., Кашеваров А.А. Системное математическое моделирование процессов водообмена. - Новосибирск: Наука, 1986. - 215 с.

2. Зеегофер Ю.О., Клюквин А.Н., Пашковский И.С., Рошаль А.А. Постоянно действующие модели гидrolитосферы территорий городских агломераций (на примере Московской агломерации). - М.: Наука.1990. - 198 с.
3. Автоматизированные сеточные модели бассейнов подземных вод/ Всесоюз. научн.-исслед. ин-т гидрогеологии и инженерной геологии; Сост.: И.И. Крашин, Е.А. Полшков, Е.К. Орфаниди и др. - М.: Недраб 1992. - 176 с.

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ КАРКАСА ДЛЯ СОЗДАНИЯ ЭЛЕКТРОННЫХ ОБРАЗОВАТЕЛЬНЫХ МАТЕРИАЛОВ

Гуленко К.А. – студент, Старолетов С.М. – к.ф.-м.н., ст. преп.
Алтайский государственный технический университет (г. Барнаул)

Актуальность проблемы

Повсеместное использование компьютерной техники и информационных технологий привело к тому, что на сегодняшний день, практически во всех сферах деятельности человека используются компьютерные технологии. Сфера образования не является исключением. За последние 10-15 лет компьютерные технологии вошли практически во все учебные заведения как средство обучения, либо как средство создания обучающих материалов (электронные учебники, тесты, презентации). Электронные образовательные материалы наглядны, легко усваиваемы, доступны и направлены на повышение эффективности образовательного процесса.

Постановка задачи

Задача данной работы – реализовать каркас для создания электронных образовательных материалов. Многим образовательным учреждениям требуется программное обеспечение, которое дает возможность создавать электронные образовательные материалы. ПО должно обладать достаточно большим набором функций и интуитивно-понятным интерфейсом. Желательно, чтобы программа была не требовательна к ресурсам системы и имела возможность создания электронных образовательных материалов с использованием файлов различных форматов (текст, изображения, музыка, видео) и возможность создания тестов самоконтроля.

Решение задачи

Существующее ПО для создания электронных образовательных материалов отличаются либо излишней сложностью при работе с ними, либо недостатком важных функций. В данной работе предпринята попытка реализации всех необходимых возможностей в совокупности с достаточной простотой пользовательского интерфейса.

Для обеспечения кроссплатформенности системы было принято решение о реализации ее в виде web-приложения. Это обеспечит его работу в любой операционной системе, где возможен доступ в Интернет через web-браузер.

Для реализации пользовательского интерфейса был выбран Wordpress – система управления содержимым сайта с открытым исходным кодом, распространяемая под GNU GPL. Включает множество плагинов, предназначенных для реализации различных функций. Wordpress написан на языке PHP и имеет возможность редактирования исходного кода, что позволяет не придерживаться стандартных средств системы, а изменять их под конкретные цели. Система использует СУБД MySQL.

Результаты

В результате проведенной работы были спроектированы пользовательский интерфейс на основе Wordpress, позволяющий создавать электронные образовательные материалы, в дальнейшем система будет внедрена – т.е. этот проект будет запущен в сети Интернет.

Список литературы

1. Официальный сайт Wordpress [Электронный ресурс]: материалы сайта – Режим доступа: <http://ru.wordpress.org/>.
2. Официальный сайт PHP [Электронный ресурс]: материалы сайта – Режим доступа: <http://www.php.net/>.
3. Официальный сайт Denver [Электронный ресурс]: материалы сайта – Режим доступа: <http://www.denwer.ru/>.

ЛИЦЕНЗИИ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Чураков М. – студент, Старолетов С.М. – к.ф.-м.н., ст. преп.
Алтайский государственный технический университет (г. Барнаул)

В настоящее время все больше и больше разработчиков хотят публиковать свои работы в виде открытых проектов и делиться своим кодом. Существуют великое множество открытых программ для самых различных видов задач. Примерами таких программ являются: Gimp – популярный растровый графический редактор, по своим возможностям практически не уступающий место известному редактору Adobe Photoshop. Из систем программирования в учебных заведениях, в том числе и в АлтГТУ им. Ползунова И.И. используют Geany и Lazarus. Многие web-разработчики используют свободное ПО как основу для своих работ, примером такого ПО являются любая CMS (WordPress, Joomla и т.п.).

Существует большое количество лицензий open-source и часто бывает непонятно: что же на самом деле они означают. Какие же права получают разработчики выбирая ту или иную версию открытой лицензии?

Сам смысл понятия лицензии на ПО - это правовой инструмент, определяющий использование и распространение программного обеспечения, защищённого авторским правом.

Лицензирование является самым простым способом передачи вашего продукта либо в общественное достояние, либо по предоставлению разрешения в каждом конкретном случае. Выпустив свой продукт на публику, вы лишитесь всех прав на него, а если же ваша программа нанесла какой-либо ущерб, то это еще грозит и судебными тяжбами. Во втором случае, вы потратите очень много времени на выдачу разрешений, чем на саму разработку программы.

Лицензии open-source позволяют как и обычному пользователю так и разработчику внести свои изменения в проект, при этом не спрашивая ни у кого какого-либо разрешения. Также они защитят вас как автора, и это уже даёт гарантию, что ваше имя будет указано в списке разработчиков. Существенным плюсом лицензии считается то, что она защитит вашу работу от кражи другими лицами.

Рассмотрим наиболее популярные лицензии такие как GPL, LGPL и BSD.

GNU General Public License (GNU GPL) - одна из наиболее применяемых лицензий для open-source проектов.

GPL предоставляет получателям компьютерных программ следующие права:

1. свободу запуска программы с любой целью;

2. свободу изучения того, как программа работает, и её модификации (предварительным условием для этого является доступ к исходному коду);
3. свободу распространения копий как исходного, так и исполняемого кода;
4. свободу улучшения программы, и выпуска улучшений в публичный доступ (предварительным условием для этого является доступ к исходному коду).

В общем случае распространитель программы, полученной на условиях GPL, либо программы, основанной на таковой, обязан предоставить получателю возможность получить соответствующий исходный код.

Используя лицензию GNU GPL вы получаете широкий диапазон прав: копирование, распространение и модификация ПО.

Это значит, что вы можете копировать ПО по своему усмотрению: на домашний ПК, на любой сервер, любому человеку, к тому же количество копий неограниченно. Вам позволено распространять и модифицировать ПО по вашему усмотрению. Также вы можете зарабатывать на данном продукте. Только для этого вам придётся приложить копию лицензии GPL, в которой поясняется, что ПО на самом деле бесплатное. Однако большое количество разработчиков зарабатывают на технической поддержке своего продукта.

Использовать GPL очень легко. Необходимо лишь внести необходимую информацию в исходный код своей программы и приложить к нему копию GPL.

GNU Lesser General Public License (LGPL) – производная от GNU GPL лицензия. Основные положения у них полностью одинаковые, однако LGPL даёт немного больше прав, чем GPL.

Главное отличие LGPL от GPL, это то, что LGPL позволяет нам связывать библиотеки с любой программой, в частности с теми, которые распространяются не по GPL и являются закрытыми. Так как лицензия GPL требует, чтобы на программное обеспечение с участками GPL также распространялась под GPL. А это лишает разработчиков возможности использовать GPL код для разработки ПО под патент или коммерческого ПО. Лицензия LGPL даёт нам это право.

BSD (Berkeley Software Distribution license) - это лицензионное соглашение, впервые применённое для распространения UNIX-подобных операционных систем BSD.

На данный момент существует две основные версии лицензии BSD – оригинальная и так называемая – модифицированная (New BSD License).

Оригинальная BSD содержит в себе 4 пункта:

1. При повторном распространении исходного кода должно оставаться указанное выше уведомление об авторском праве, этот список условий и последующий отказ от гарантий.
2. При повторном распространении двоичного кода должна сохраняться указанная выше информация об авторском праве, этот список условий и последующий отказ от гарантий в документации и/или в других материалах, поставляемых при распространении.
3. Все рекламные материалы, ссылающиеся на возможности или использование этого ПО, должны демонстрировать следующее уведомление: Этот продукт включает ПО, разработанное Университетом Калифорнии, Беркли и помогавшим лицам.
4. Ни название организации, ни имена ее сотрудников не могут быть использованы в качестве поддержки или продвижения продуктов, основанных на этом ПО без предварительного письменного разрешения.

В модифицированной версии (New BSD License), третий пункт пересмотрен и исключен.

Лицензия New BSD разрешает разработчикам неограниченное распространение с любой целью, не даёт никаких гарантий и не несёт никакой ответственности.

Мы рассмотрели три основные лицензии, но их существует около 70. Многие из них практически повторяют друг друга с небольшими изменениями, из-за чего сложнее выбрать нужную нам лицензию. Open Source Initiative работает над тем, чтобы сократить их количество до приемлемого.

Если вы собираетесь использовать лицензии, обязательно проконсультируйтесь с юристом, к тому же Open Source Initiative гарантируют защиту ваших прав по каждой из указанных лицензий.

Список литературы

1. Лицензия свободного программного обеспечения [электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki/Лицензия_свободного_программного_обеспечения.
2. A Short Guide To Open-Source And Similar Licenses [электронный ресурс]. – Режим доступа: <http://www.smashingmagazine.com/2010/03/24/a-short-guide-to-open-source-and-similar-licenses/>.
3. Ответы на вопросы о лицензиях GNU [электронный ресурс]. – Режим доступа: <http://www.gnu.org/licenses/gpl-faq.html>.

РАЗРАБОТКА УТИЛИТЫ ДЛЯ УЧЕТА И АНАЛИЗА HTTP ТРАФФИКА ПОД ОС LINUX

Краюшкин М.С. – студент, Старолетов С.М. – к.ф.-м.н., ст. преп.
Алтайский государственный технический университет (г. Барнаул)

Актуальность

Иногда требуется учитывать HTTP трафик, потребляемый и генерируемый компьютером. Такая задача появляется даже сейчас, когда Интернет стал недорогим, а широкие скоростные каналы связи легкодоступными даже в небольших городах. Например, когда используется дорогостоящее GPRS соединение или WEB-разработчикам требуется узнать, какой объем трафика генерируется их сайтом. Также актуальна необходимость фильтрации трафика с определенных сайтов(например, как часть функции родительского контроля). Под семейство ОС Windows и MacOS X имеется множество инструментов,имеющих описанный функционал. А под Linux подобных утилит с дружественным пользовательским интерфейсом нет. Поэтому была поставлена задача по написанию такого инструмента для ОС Linux.

Постановка задачи

Требуется написать инструмент, позволяющий просматривать статистику HTTP трафика по доменам, ограничивать и блокировать трафик с определенных сайтов.

Архитектура системы

Предполагается, что система будет состоять из двух основных частей – модуля ядра Linux и пользовательской утилиты. Общение модуля и пользовательской программы будет осуществляться через создаваемый ядром файл символического устройства.

Входящие и исходящие пакеты будут перехватываться через встроенную в ядро библиотеку netfilter. Далее анализироваться на принадлежность HTTP трафику и отправляться утилите на обработку.

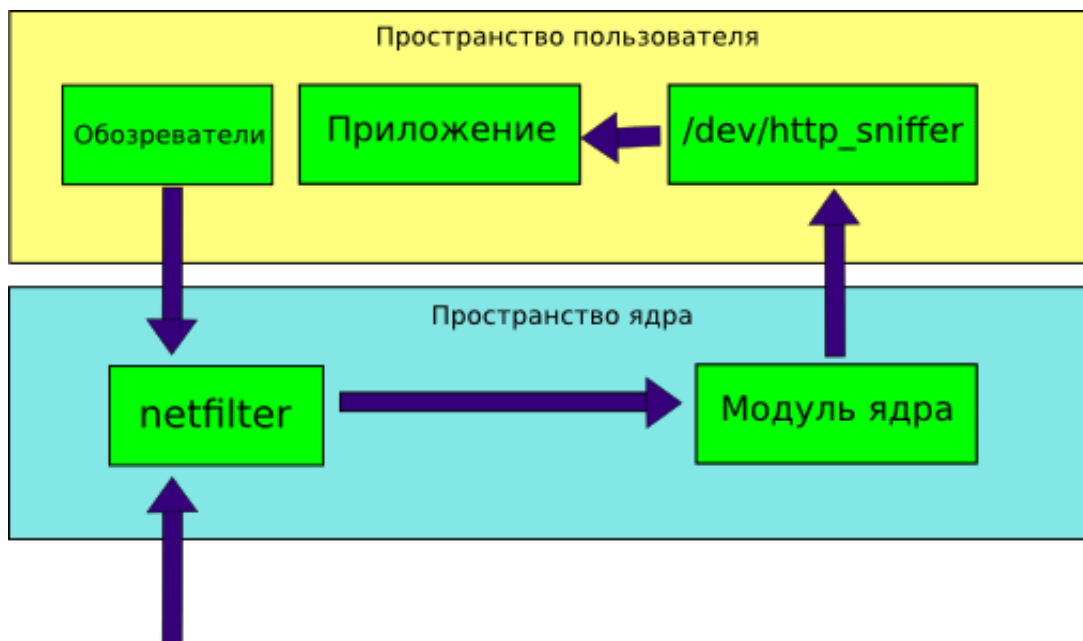


Рисунок 1 – Архитектура системы

Возможные улучшения системы

- 1) Учет также HTTPS трафика, в связи с его широким распространением.
- 2) Поддержка трафика IPv6.
- 3) Подсчет потребления трафика по конкретным приложениям.

Список литературы

1. Танненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е изд. – СПб.: Питер, 2012.– 960 с.
2. Корбет Д., Рубини А. Linux Device Drivers, 3-е изд. – O'Reilly. 2012. – 620 с.
3. How SKB's work [Электронный ресурс]. – Режим доступа: http://vger.kernel.org/~davem/skb_data.html.

КОМПЛЕКС ПРОГРАММ ДЛЯ СОЗДАНИЯ, ПРОСМОТРА И ОЦЕНКИ АНИМИРОВАННЫХ ИЗОБРАЖЕНИЙ В ФОРМАТЕ GIF

Стручев Р.К. – студент, Старолетов С.М. – к.ф.-м.н., ст. преп.
 Алтайский государственный технический университет (г. Барнаул)

Актуальность

В настоящее время в социальных сетях и на различных веб-порталах люди делятся друг с другом анимированными изображениями в формате gif. Анимацию в этом формате можно устанавливать как фотографию пользователя в карточку профиля, ставить заставкой на рабочий стол, прикреплять к сообщению в социальных сетях. Область использования таких файлов ограничивается только фантазией пользователя.

Существует множество средств для создания анимированных изображений в формате gif, которые можно разделить на следующие категории:

1. Онлайн средства на специализированных сайтах и порталах;
2. Приложения в социальных сетях;

3. Программы для компьютеров под различные операционные системы, такие как Windows, Mac OS, Linux и другие;
4. Приложения для мобильных устройств под различные операционные системы, такие как Android, iOS, Windows Mobile и другие;

На программы для создания анимированных изображений активно возрастает спрос среди пользователей интернета, преимущественно младшей и средней возрастной категории. Наиболее важным для пользователя являются простота, универсальность и доступность программного решения.

Краткий анализ существующих программных решений

Произведен детальный анализ четырех наиболее популярных приложений для создания анимаций с использованием мобильных устройств, расположенных в магазине приложений Google Play:

- Gif Boom - разработчик GifBoom Inc., установок более 1 000 000;
- Gif Camera - разработчик Listen5, установок более 500 000;
- Gif Камера - разработчик droidkitchir.com, установок более 100 000;
- Gif Maker - разработчик – Dexati, установок более 50 000.

Все рассмотренные приложения используют в основном для создания анимированных изображений с камеры мобильного устройства или из изображений, хранящихся на нём, имеют ряд дополнительных функций, например наложение текста, картинок и фильтров на анимацию, возможность задавать произвольное количество кадров в секунду или указывать задержку между ними, но также имеют ряд недостатков, например приложением Gif Boom нет возможности пользоваться без авторизации через учетную запись в социальных сетях, также нет возможности получить gif файл анимации, т.к. программа использует свой формат, в приложении Gif Камера требуется указывать множество параметров для создания анимации, что делает его сложным при использовании.

Сравнительный анализ существующих программных решений позволяют сделать вывод о том, что в настоящий момент не существует универсального и удобного для пользователя программного комплекса, который бы позволял создавать анимированные изображение напрямую с камеры мобильного устройства, из картинок или видео, хранящихся на нем или видео, которое находится на хостинге, например youtube.com, а также просматривать, комментировать и поднимать рейтинг анимациям, созданным другими пользователями.

Постановка задачи

Необходимо создать комплекс взаимодействующих программных средств, который позволял бы создавать анимированные изображение с камеры мобильного устройства или компьютера, из картинок или видео, хранящихся на устройстве или видео, которое находится на хостинге, например youtube.com, а также просматривать, комментировать и поднимать рейтинг анимациям, созданным другими пользователями. Программный комплекс должен состоять из следующих компонентов:

1. Приложение для мобильных устройств, функционирующих на операционной системе Android, с поддержкой актуальных версий ОС (поддержка, начиная с версии 2.2.x *Froyo*, включено более 99.7% от всех устройств на Android ОС). Приложение должно корректно работать как на мобильных телефонах разного форм-фактора, так и на планшетных компьютерах.

Основные функции приложения:

- ▲ Создание анимированных изображений в формате gif с камеры мобильного устройства или из изображений, уже имеющихся на нем;
- ▲ Возможность накладывать фильтры и текст на анимацию при её создании, сохранять анимацию в обратном порядке кадров;

- ▲ Ограничение анимации по длительности, возможность задавать произвольный fps (количество кадров в секунду) при создании анимации с камеры мобильного устройства, задержку между кадрами при создании анимации из изображений;
 - ▲ Публикация созданных анимаций на сервере, возможность удаление опубликованных файлов с сервера, возможность получить ссылку на страницу веб портала с выбранной анимацией и рассказать друзьям об анимации или о ссылке на неё средствами Android ОС;
 - ▲ Просмотр анимаций, опубликованных на сервере, разделенных по категориям «Лучшие дня», «Лучшие месяца», «Лучшие за все время», повышение рейтинга анимациям, возможность просматривать и писать комментарии, добавлять понравившиеся анимации в раздел «Избранное»;
 - ▲ Отмечать и кэшировать уже просмотренные анимации, кэш хранить на карте памяти, ограничить размер кэша.
2. База данных и веб сервис, обеспечивающие хранение, а также отправку и прием данных от приложения для мобильных устройств и веб портала. Посредством веб сервиса и базы данных обеспечивается обмен анимациями между пользователями.
 3. Веб портал, позволяющей пользователю создавать, просматривать, оставлять комментарии и поднимать рейтинг анимациям, используя браузер персонального компьютера. Веб портал имеет схожую с мобильным приложением структуру и возможности, позволяет создавать анимации с веб камеры, из картинок и видео, как находящегося у пользователя на компьютере, так и на видео хостинге.

Технологии и архитектура

Взаимодействие компонентов программного комплекса представлено на рисунке 1.

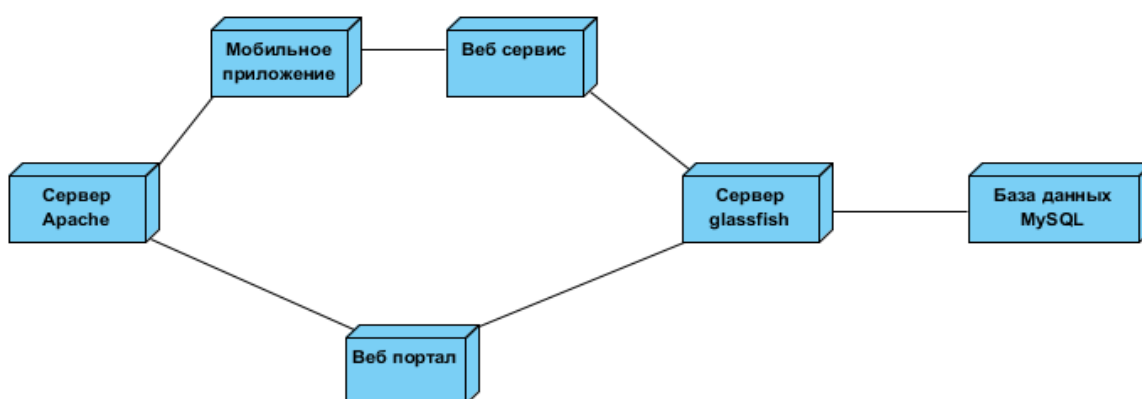


Рисунок 1 – Диаграмма развертывания

Мобильное приложение написано на языке Java, подключен native development kit т.к. используется библиотека для работы с фильтрами с открытым исходным кодом, написанная на C++. Приложение обменивается данными с Spring RESTful веб сервисом в формате json. Анимации создаются из массива изображений с использованием библиотеке с открытым исходным кодом AnimatedGifEncoder, самостоятельно адаптированной под Android ОС, для отображения анимаций используется библиотека с открытым исходным кодом GifDecoder, самостоятельно адаптированная под Android ОС.

Веб портал написан на языке Java, используется фреймворк для веб приложений JavaServer Faces и библиотека Primefaces. Функционирует портал на сервере приложений Glassfish, доступен на 8080 порту.

Сервер Apache используется для загрузки анимаций в мобильное приложение и веб портал, доступен на 80 порту.

Внедрение и сопровождение

Данный программный комплекс функционирует и доступен по адресу gif-creator.ru. В процессе функционирования было выявлено и решено множество проблем, некоторые из них:

1. Нежелательное содержание анимаций (эротическое, насилие, детская порнография и другое), которые пользователи публикуют;
2. Разработка системы подачи и обработки жалоб от пользователей;
3. Разработка алгоритма отслеживания «плохих» пользователей;
4. Изменение дизайна приложения;

Со статистикой установок мобильного приложения можно ознакомиться на рисунках 2 и

3.

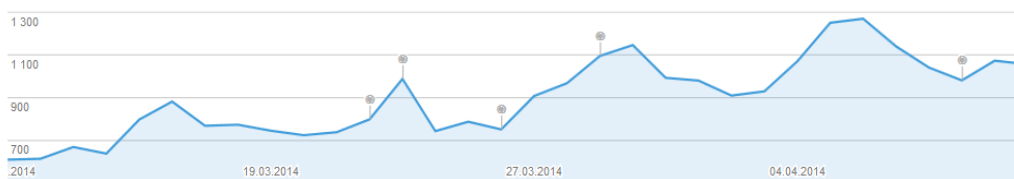


Рисунок 2 – Количество установок приложения за день



<input checked="" type="checkbox"/>	Соединенные Штаты	8 655	22,47 %
<input checked="" type="checkbox"/>	Индонезия	3 969	10,30 %
<input checked="" type="checkbox"/>	Россия	2 784	7,23 %
<input type="checkbox"/>	Италия	1 964	5,10 %
<input type="checkbox"/>	Бразилия	1 498	3,89 %
<input type="checkbox"/>	Республика Корея	1 496	3,88 %
<input type="checkbox"/>	Мексика	1 308	3,40 %
<input type="checkbox"/>	Польша	1 250	3,25 %
<input type="checkbox"/>	Великобритания	1 030	2,67 %
<input type="checkbox"/>	Индия	958	2,49 %
<input type="checkbox"/>	Другое	13 605	35,32 %

Рисунок 3 – Местоположения пользователей

Заключение

В рамках данной работы создан комплекс программ, отвечающий всем требованиям пользователя и позволяющий в несколько кликов создавать анимированное изображение практически из любого источника.

На разработанное программное обеспечение подана заявка о его государственной регистрации. Планируется сопровождение и доработка программного обеспечения для удовлетворения новых требований и расширения круга пользователей.

Список литературы

1. Android Developers [Электронный ресурс]. – Режим доступа: <http://developer.android.com>
2. RESTful Web Services [Электронный ресурс]. – Режим доступа: <https://spring.io/guides/gs/rest-service>.
4. Open Server - платформа для веб разработчиков [Электронный ресурс]. – Режим доступа: <http://open-server.ru/docs.html>.

5. Primefaces - ultimate JSF component suite [Электронный ресурс]. – Режим доступа: <http://www.primefaces.org/gettingStarted> .

МЕТОДЫ DATA MINING КАК СОВРЕМЕННЫЕ ПОДХОДЫ ДЛЯ РЕШЕНИЯ СТАТИСТИЧЕСКИХ ЗАДАЧ ПРОГНОЗИРОВАНИЯ

Ненайденко А.С. – аспирант, Боровцов Е.Г. – к.т.н., профессор
Алтайский государственный технический университет (г. Барнаул)

С каждым годом количество информации в мире увеличивается колоссальными темпами. Согласно оценкам аналитиков, информация удваивается каждые 2-3 года. Этот огромный поток данных приходит из различных областей науки, бизнеса, Интернета и других источников. Из-за такого количества информации, очень малая ее часть будет когда-либо увидена человеческим глазом, а главное обработана и проанализирована. Вместе с тем, во всем этом потоке информации содержится просто огромное количество полезных знаний. Именно поэтому существует потребность в умении быстро и качественно анализировать эти горы информации и выделять в ней существенные моменты, а также строить на ее основе прогнозы на будущее.

Алтайский Государственный Технический Университет, являясь одним из крупнейших вузов России, имеет огромные массивы информации, которая также быстро увеличивается с каждым годом. Эта информация используется в различных задачах функционирования и развития университета. Сбор данных об абитуриентах, анализ успеваемости студентов, расчет нагрузки на телефонную и сетевую инфраструктуру и многие другие задачи могут быть направлены на обработку и последующий анализ накопленной информации с целью извлечения пользы или даже какой-либо выгоды для вуза. Эти задачи анализа, в той или иной степени, могут быть решены с помощью задач статистического прогнозирования.

Конечно, статистика как наука обеспечивает математическим аппаратом методы сбора данных, их обработки и анализа для выявления закономерностей. Аппарат этот хорошо проработан и изучен. Но для работы с современными объемами данных тяжело опираться только лишь на классические подходы статистики. Традиционные методы анализа данных в основном ориентированы на проверку заранее сформулированных гипотез. Зачастую выдвижение даже самих этих гипотез, относительно зависимости данных, является далеко нетривиальной задачей для аналитика. Кроме того, большинство статистических методов для выявления взаимосвязей и закономерностей в данных используют концепцию усреднения по выборке, приводящую к операциям над несуществующими величинами. Альтернативной традиционной статистики является так называемый подход Data Mining.

Data Mining (иногда называемая Knowledge Discovery In Data - обнаружение знаний в данных) изучает процесс нахождения новых, действительных и потенциально полезных знаний преимущественно в базах данных. В основе Data Mining лежит комплекс наук, основными из них являются: статистика, искусственный интеллект и теория баз данных.

Область Data Mining выросла из одного семинара в 1989 году до десятков международных конференций в наше время, с тысячами исследователей во многих странах мира. Data Mining широко используется во многих областях с большим объемом данных. В науке - астрономии, биологии, биоинформатике, медицине, физике и других областях. В бизнесе - торговле, телекоммуникациях, банковском деле, промышленном производстве и т.д.

Основная особенность Data Mining - это сочетание широкого математического инструментария (от классического статистического анализа, до новых кибернетических методов) и последних достижений в сфере информационных технологий. В технологии Data

Mining объединились строго формализованные методы и методы неформального анализа, т.е. количественный и качественный анализ данных.

Все методы Data Mining, в зависимости от решаемых задач, можно разбить на несколько групп. Одной из таких групп будут являться прогнозирующие методы. Именно они являются предметом изучения в данной статье.

Прогнозирующие методы используют значения одних переменных, которые уже имеются в полученных данных, для предсказания/прогнозирования неизвестных (пропущенных) или будущих значений других (целевых) переменных. К таким методом, направленным на получение прогнозирующих результатов, относятся следующие методы: нейронные сети, деревья решений, линейная регрессия, метод ближайшего соседа и др. (сравнительная характеристика представлена в Таблице 1). Рассмотрим подробнее наиболее популярные и широкоиспользуемые из них.

Таблица 1 - Сравнительная характеристика методов Data Mining

Метод	Точность	Масштабируемость	Интерпретируемость	Пригодность к использованию
Деревья решений	Низкая	Высокая	Высокая	Высокая/нейтральная
к-ближайшего соседа	Низкая	Очень низкая	Высокая/нейтральная	Нейтральная
Нейронные сети	Высокая	Низкая	Низкая	Низкая
Линейная регрессия	Нейтральная	Высокая	Высокая/нейтральная	Высокая
Метод	Трудоемкость	Разносторонность	Быстрота	Популярность
Деревья решений	Высокая	Высокая	Высокая/нейтральная	Высокая/нейтральная
к-ближайшего соседа	Нейтральная/низкая	Низкая	Высокая	Низкая
Нейронные сети	Нейтральная	Низкая	Очень низкая	Низкая
Линейная регрессия	Нейтральная	Нейтральная	Высокая	Низкая

Деревья решений (Decision Trees, деревья классификации или регрессионные деревья) - используется в области статистики и анализа данных для прогнозных моделей, а также как визуальный и аналитический инструмент поддержки принятия решений, где рассчитываются ожидаемые значения (ожидаемая полезность) конкурирующих альтернатив.

Достоинства метода: надежность, простота в понимании и интерпритации; не требует подготовки данных; способен работать как с категориальными, так и с интервальными переменными, использует модель "белого ящика"; позволяет оценить модель при помощи статистических тестов; позволяет работать с большим объемом информации без специальных подготовительных процедур.

Недостатки метода: проблема получения оптимального дерева - NP-полная задача, даже для простых задач; зачастую создаются слишком сложные конструкции, которые не достаточно полно представляют данные.

Метод к-ближайшего соседа (k-Nearest Neighbour) относится к классу методов, работа которых основывается на хранении данных в памяти для сравнения с новыми элементами. При появлении новой записи для прогнозирования находятся отклонения между этой записью и подобными наборами данных, и наиболее подобные (или k ближайших соседей) идентифицируется.

Достоинства метода: простота использования полученных результатов; решения не уникальны для конкретной ситуации, возможно их использование для других случаев; целью поиска является не гарантированно верное решение, а лучшее из возможных.

Недостатки метода: не создает каких-либо моделей или правил, обобщающих предыдущий опыт - в выборе решения они основываются на всем массиве доступных

исторических данных; сложность выбора меры "близости" (метрики); большая вычислительная трудоемкость - как следствие полного перебора обучающей выборки; типичные задачи метода - это задачи небольшой размерности.

Нейронные сети (Neural Networks) - это модели биологических нейронных сетей мозга, в которых нейроны имитируются относительно простыми, часто однотипными, элементами (искусственными нейронами).

Способности нейронной сети к прогнозированию напрямую следуют из ее способности к обобщению и выделению скрытых зависимостей между входными и выходными данными. После обучения сеть способна предсказать будущее значение некой последовательности на основе предыдущих значений и (или) каких-то существующих в настоящий момент факторов. При этом, прогнозирование возможно только тогда, когда предыдущие изменения действительно в какой-то степени определяют будущее.

Применение нейросетевых методов позволяет решить некоторые проблемы экономико-статистического моделирования, повысить адекватность математических моделей, приблизить их к экономической реальности.

Достоинства метода: результативность при решении неформализованных или плохо формализованных задач - т.е. отсутствие необходимости в строгой математической спецификации модели; устойчивость к частым изменениям среды; результативность при работе с большим объемом противоречивой информации, со скрытыми зависимостями; результативность работы с неполной, "зашумленной" информацией.

Недостатки метода: для эффективного прогнозирования, необходим некоторый минимум наблюдений - но не всегда он есть; требуются значительные временные затраты для достижения удовлетворительного результата; надежно интерпретировать результаты, а также численно оценивать значимость получаемых прогнозов способны специалисты, имеющие навыки в моделировании нейронных сетей; отсутствие прозрачности вычислений.

Линейная регрессия (Linear Regression) - используемая в статистике регрессионная модель зависимости одной (объясняемой, зависимой) переменной Y от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) X с линейной функцией зависимости. Данный метод наиболее изучен и часто используется в эконометрике.

Достоинства метода: простота; гибкость; высокое быстродействие; прозрачность моделирования - т.е. доступность для анализа на всех промежуточных вычислениях.

Недостатки метода: неприспособленность к решению существенно не линейных задач; сложность определения функциональной зависимости; низкая адаптивность.

Если проанализировать вышеописанные плюсы и минусы методов Data Mining, применяемых в прогнозировании, можно сделать вывод о том, что невозможно однозначно и четко сказать какой из методов предпочтительнее, лучше и дает более качественный результат. Этот результат всегда будет зависеть от поставленной перед аналитиком задачи, а также от уже имеющегося в его распоряжении набора данных, а также от их качества. Кроме того, в таком основополагающем показателе как точность, вовсе нельзя дать корректную оценку, так как она основывается ни только на применяемом методе, но в значительной степени на опыте и квалификации исследователя, доступности данных и от располагаемой аппаратной мощности.

Исходя из этого, в использовании подходов интеллектуального анализа, нужно стремиться к использованию некоторой золотой середины и комбинировать различные методы с целью получения более надежных прогнозов.

Список литературы

1. Библиотека алгоритмов- Линейная регрессия [Электронный ресурс]. - Режим доступа: <http://alglib.sources.ru/dataanalysis/linearregression.php>.

2. Википедия Дерево принятия решений [Электронный ресурс]. - Режим доступа: http://ru.wikipedia.org/wiki/Дерево_принятия_решений.
3. Круг П.Г. Нейронные сети и нейрокомпьютеры .- М.: Изд-во МЭИ, 2002. - 176 с.
4. Методы и модели анализа данных: OLAP и Data Mining / А.А. Барсегян [и др.]. - СПб.: Изд-во БХВ-Петербург, 2004. - 336 с.
5. НОУ ИНТУИТ Учебный курс Data Mining [Электронный ресурс]. - Режим доступа: <http://www.intuit.ru/studies/courses/6/6/info>.
6. Паклин Н.Б. Бизнес-аналитика: от данных к знаниям / Н.Б. Паклин, В.И Орешков. - СПб.: Изд-во Питер, 2013. -704 с.
7. Чучуева И.А. Модель прогнозирования временных рядов по выборке максимального подобия: дис. ... канд. техн. наук : 05.13.18. - М., 2012. - 153 с.

КОНСОЛЬНЫЙ КЛИЕНТ К МУЗЫКАЛЬНОМУ САЙТУ

Смолянинов А.Ю. – студент, Крайванова В.А. – к.ф.-м.н., доцент
Алтайский государственный технический университет (г. Барнаул)

В наши дни все больше и больше людей пользуются онлайн сервисами для прослушивания музыки. Это очень удобно, так как музыку можно слушать с любого устройства с доступом в сеть. Если раньше для размещения большой коллекции было необходимо выделять значительные ресурсы на локальных носителях и принимать определенные меры для их сохранности, то теперь в этом нет необходимости.

Одним из самых популярных сервисов онлайн музыки является Soundcloud[4]. На этом сайте каждый исполнитель может выложить свои композиции для всеобщего прослушивания и скачивания. Он отличается от всех музыкальных сайтов тем, что позволяет привязывать комментарий не только к целой композиции, а к отдельным ее частям.

Сервис предоставляет удобное API и SDK для доступа к данным сайта, что позволяет создавать приложения на любой платформе. Однако, в классе консольных приложений представлен только один CLI-клиент, реализующий лишь базовые функции, без возможности комментирования. Однако, данный вид приложений востребован среди пользователей *nix систем. Поэтому создание консольного плеера для Soundcloud является актуальной задачей.

Многие люди любят пользоваться консольными (или CLI) утилитами. Такое ПО запускается и работает из эмулятора терминала. Из этой потребности и появился консольный плеер для Soundcloud.com.

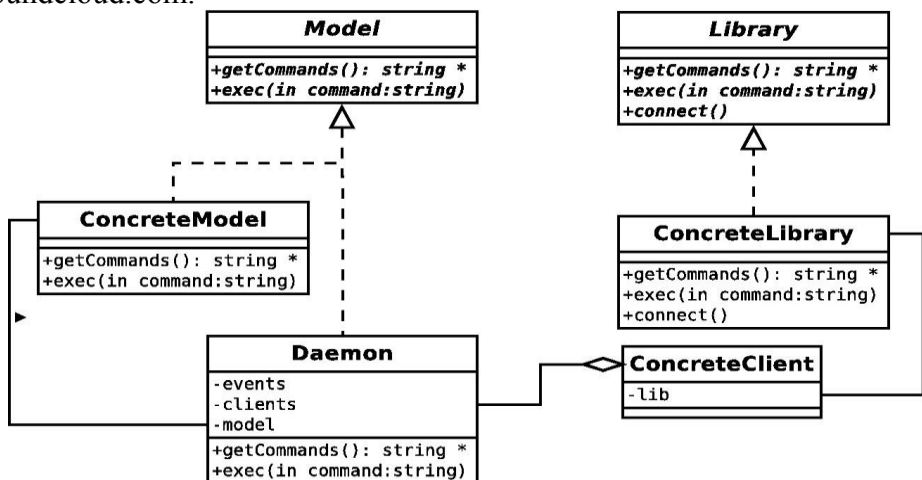


Рисунок 1 – Клиент-серверная архитектура консольного плеера

Разработанный плеер имеет клиент-серверную архитектуру, реализующую паттерн MVC. Такая архитектура выбрана из соображений расширения и масштабируемости, так как позволяет создавать новые клиенты на разных платформах абстрагируясь от реализации доступа к данным сервиса. Именно это является ключевым отличием от немногочисленных аналогов, имеющих монолитную архитектуру и привязанных к одной платформе.

Контроллер — это «ядро» плеера. Именно он проигрывает музыку и ждет команд от клиентов. Контроллер предоставляет доступ к сервисам модели и интерфейсы управления для клиентов.

К функциям контроллера относятся обработка запросов клиентов (проиграть трек, пауза) и подписка клиентов на события(сменился трек, трек приостановлен). Клиент может подключиться к контроллеру различными способами, с помощью различных протоколов. Это может быть одиночный REST-запрос, соединение на основе запроса/ответа или постоянное подключение. Реализована возможность добавить новый протокол (за счет абстракции протокола взаимодействия) и настроить, какие конкретно протоколы должны запускаться при старте.

Контроллер расширяет набор списка команд модели возможностью подключения *connect*, подписки/отписки от событий *subscribe/unsubscribe*, список доступных событий *getEventList*. Также могут появиться дополнительные команды управления какими-либо процессами не относящиеся к источнику данных.

Клиенты могут быть реализованы не только в консоли, выбор платформы реализации ничем не ограничен. Для абстрагирования от логики связи и получения данных клиенты пользуются библиотеками, реализующие функции доступа к контроллеру для своей платформы.

В роли модели выступает провайдер данных, а именно провайдер доступа к API сервиса Soundcloud. Сама модель играет роль поставщика данных: о исполнителе, о треке, о комментариях к треку, список треков по жанрам и пр.

В приложении используется пассивная модель: она не подписывает каких-либо клиентов на изменение данных, все взаимодействия происходят по методу запрос-ответ между моделью и контроллером. Каждая модель должна предоставлять две функции для вызова: *getCommandList()* и *exec(command)*.

Контроллер пользуется этими командами для предоставления клиентам возможностей и источников данных. От возможностей (*getCommandList()*) зависит то, какие данные будут предоставлены. Каждое клиентское приложение в зависимости от полученного списка решает, какие возможности он может предоставить пользователю и может ли он вообще работать с таким источником. *exec(command)*, соответственно, выполняет команду и возвращает результат.

Все данные как внутри, так и между компонентами системы обрабатываются по принципу конвейера.

Конвейер [3] в терминологии UNIX — некоторое множество процессов, для которых выполнено следующее перенаправление ввода-вывода: то, что выводит на поток стандартного вывода предыдущий процесс, попадает в поток стандартного ввода следующего процесса. Механизм конвейеров добавляет гибкость процессу взаимодействия программ. Все программы оперируют текстовыми данными, то есть получают на вход текст и выводят его же, причем, обработка данных внутри программы без лишнего копирования и хранения, что существенно повышает производительность.

Представление из паттерна MVC реализуется в клиентском приложении.

Клиентское приложение в общем случае предоставляет функции поиска и проигрывания музыкальных композиций, просмотра и добавления комментариев, сохранение на локальный диск и другие возможности, доступные клиентам сервиса Soundcloud. Управление в консольном клиенте реализовано с помощью привычных CLI и *curses утилит сочетаний клавиш. Переход по разделам интерфейса осуществляется с помощью цифровых символов.

Для удобства есть быстрый доступ к справке по горячим клавишам.

Таким образом, основные достоинства данного консольного клиента в предоставлении достаточно широкого функционала, в возможности его гибкой настройки и использовании в качестве независимого сервера.

Список литературы

1. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст]. / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес – СПб: Питер, 2001. – 381 с.
2. Конвейер (UNIX) [Электронный ресурс]. // Wikipedia. – 2014. – Режим доступа: [http://en.wikipedia.org/wiki/Pipeline_\(Unix\)](http://en.wikipedia.org/wiki/Pipeline_(Unix)), свободный.
3. Soundcloud.com [Электронный ресурс]. // Soundcloud. – 2014. – Режим доступа: <http://soundcloud.com/>, свободный.

РАСПРЕДЕЛЕННЫЙ КОМПЛЕКС УПРАВЛЕНИЯ УСТРОЙСТВАМИ В УМНОМ ДОМЕ

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

Умный дом (англ. smart home) [1] — жилой дом, организованный для проживания людей при помощи автоматизации и высокотехнологичных устройств. Под "умным" домом следует понимать систему, которая обеспечивает безопасность, комфорт и ресурсосбережение для всех пользователей. В простейшем случае она должна уметь распознавать конкретные ситуации, происходящие в доме, и соответствующим образом на них реагировать: одна из систем может управлять поведением других по заранее выработанным алгоритмам. Кроме того, от автоматизации нескольких подсистем обеспечивается синергетический эффект для всего комплекса.

В настоящее время компьютерные технологии повсеместно завоевывают все сферы жизнедеятельности современного человека. Сегодня уже не мыслимо полноценное существование без уже привычных нам вещей. Мобильные телефоны, портативные компьютеры, технологии связи уже давно стали частью нашей повседневной жизни. Таким образом, с развитием компьютерной индустрии, а также с ростом аппаратных возможностей бытовой электроники все больше становится актуальным вопрос об эффективном управлении всей техникой в доме. Интеграция всех устройств в единую программно-аппаратную систему, построенную на высоконадежном и гибком программном обеспечении, позволяет избавиться от множества элементов управления разной бытовой техникой в доме, что способствует росту комфортности использования всего имеющегося оборудования. Также использование единого управляющего центра решает проблемы с комплексным управлением множеством различных приборов в зависимости от случившегося внешнего события. Ярким примером данного взаимодействия может служить задание множества независимых действий для разных приборов в доме к обычному событию будильника, а именно — включение освещения с проигрыванием заранее установленной мелодии, прочтение актуальных на сегодня задач и погоды за окном, а также заранее приготовленное кофе. В результате интеграции всего оборудования мы получаем единый централизованный контроль и управление сложным комплексом инженерных систем жизнеобеспечения современного жилища. Это позволяет нам настроить и запрограммировать правильную реакцию на то или иное событие. Весь дом начинает работать как единый организм.

Таким образом, целью нашей работы является создание распределенных серверных и клиентских частей программного комплекса для автоматизирования процессов в «умном доме».

Исходя из поставленной цели, можно выделить основные требования, предъявляемые к разрабатываемому программному комплексу:

- обеспечение стабильности и надежности работы различных частей умного дома;
- возможность безболезненного расширения программного комплекса;
- возможность гибкой и одновременно простой настройки взаимодействия устройств и сервисов;
- наличие графического интерфейса управления умным домом, который должен быть интуитивно понятными и наглядным для обычного человека;
- наличие документированного API для легкой интеграции различных приложений в инфраструктуру «умного дома»;
- поддержка различных операционных систем:
 - Windows [2];
 - Linux [3];
 - Mac OS X/iOS [4].

После рассмотрения всех требований и анализа различных способов построения программного обеспечения, наш выбор был сделан в пользу архитектуры системы, основанной на сигналах и слотах. Данное решение обусловлено тем, что такой механизм позволяет наиболее полно и комплексно описать все процессы и взаимодействия, протекающие в умном доме. Каждое устройство или сервис обладает определенными связями с различными частями комплекса. Каждая связь полностью описана и может быть наглядно визуализирована посредством графического интерфейса управления. Множество всех связей образует единую настраиваемую пользователем систему с заданным поведением на внешние события и действия человека. Также стоит отметить, что данная система легко расширяется посредством добавления новых устройств и построения связей между ними и другими частями комплекса.

Для упрощения настройки и конфигурации комплекса был спроектирован графический интерфейс, позволяющий легко и изящно задавать поведение и реакцию комплекса на внешние события. Данной части программного обеспечения было уделено особое внимание, так как графический интерфейс должен быть наглядным и интуитивно понятным для простого человека, который использует данное ПО в своем доме.

Также для интеграции в комплекс различных внешних программ, нами было спроектировано специализированное API и набор библиотек, который может быть использован сторонними разработчиками в своих приложениях.

Запланированные возможности программного комплекса

На текущий момент нами планируется добавление возможности написания простейших запросов к слотам сервисов, что обеспечит еще большую гибкость поверх уже имеющейся системы сигналов и слотов, таким образом, добавив некоторые возможности динамического связывания. Данная возможность будет широко использоваться при обработке голосовых запросов, при которых статичное связывание не всегда возможно.

Список литературы

1. Умный дом [Электронный ресурс]: материалы сайта – Режим доступа – http://en.wikipedia.org/wiki/Home_automation – свободный.
2. Microsoft Windows [Электронный ресурс]: материалы сайта – Режим доступа – http://en.wikipedia.org/wiki/Microsoft_Windows – свободный.

3. Microsoft Windows [Электронный ресурс]: материалы сайта – Режим доступа – <https://en.wikipedia.org/wiki/Linux> – свободный.
4. Apple MacOS X [Электронный ресурс]: материалы сайта – Режим доступа – http://en.wikipedia.org/wiki/OS_X – свободный.
5. Стивенс У.Р., Феннер Б., Рудофф Э.М. C80 UNIX: разработка сетевых приложений. 3-е изд- СПб.: Питер, 2007. – 1039 с.: ил.

КОНЦЕПЦИЯ ПОСТРОЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА НА ОСНОВЕ АБСТРАКТНОГО МЫШЛЕНИЯ И АССОЦИАТИВНОЙ СТРУКТУРЫ МОЗГА ЧЕЛОВЕКА

Шитов Д.А. – студент, Старолетов С.М. - к.ф.-м.н., ст. преп.
Алтайский государственный технический университет (г. Барнаул)

1. Введение

Проблема создания искусственного интеллекта на протяжении многих лет исследуется многими математиками по всему миру. Актуальность данной проблемы заключается в том, что если удастся создать искусственный интеллект, то будет решён целый спектр задач, связанных с когнитивными функциями и алгоритмами, уподобляющими деятельностью человеческого мозга. В данной работе представлена концепция построения искусственного интеллекта на основе темпоральной иерархической модели памяти (НТМ — hierarchical temporal memory). В качестве основной идеи данной концепции лежит модель «память-предсказание», являющая собой не что иное, как ключевой механизм построения ассоциативной памяти человека и абстракции имеющейся информации. Идея состоит в том, что у любого разумного живого существа существует отдел мозга, отвечающий за хранение информации и ассоциативные связи между ними. У многих животных, в том числе у человека, есть кора головного мозга. Если взять её поверхностный срез высотой примерно 2 мм, то мы получим особый слой нервных клеток — неокортекс. Именно этот слой отвечает за разумное поведение и когнитивные функции живого существа.

Прежде чем говорить о том, почему была выбрана данная модель, необходимо привести примеры, позволяющие по аналогии объяснить правильность выбора. В современной медицине развивается направление замены синтетических веществ и материала на органические. Это даёт огромное преимущество, так как используются естественные для живого организма химические соединения. Создание органического материала в лабораторных условиях происходит полными или частичным копированием естественной структуры органических соединений, позволяя получать материал, практически неотличимый от присущего живым организмам. Как следствие, например, можно использовать восстановление сердечной мышцы «выращенными» тканями вместо использования искусственных клапанов. Плюсы этого метода заключаются в том, что ткани приживаются и используются организмом на протяжении всей жизни. Они не требуют замены и зарядки. Это лишь один из примеров использования воспроизведения органических соединений, дающий лучший результат по сравнению с синтетическими. Точно таким же способом мы можем воссоздать разумный интеллект (конечно же не пытаюсь в лабораторных условиях вырастить нервные клетки, а лишь позаимствовать идею) — попытаться воспроизвести процессы, проходящие в неокортексе человека, разработать схожую логическую структуру и представить её с математической точки зрения. В этом заключается основная особенность алгоритмов НТМ — проведение аналогов протекающих процессов в неокортексе человека с передачей данных между логическими элементами алгоритма.

2. Постановка задачи

Целью данной работы является создание модели и реализация алгоритмов темпоральной иерархической модели памяти, позволяющей частично воспроизвести когнитивные функции человеческого мозга за счёт построения ассоциативных связей во времени. Исследование наилучших параметров построенной модели и составление результатов исследования.

3. Основы алгоритмов темпоральной иерархической памяти.

Если рассмотреть неокортекс живого существа, то можно заметить, что он состоит из нескольких слоёв нервных клеток — нейронов, упорядоченных определённым образом. У человеческого мозга выделяют шесть слоёв нейронов, пять из которых полностью заполнены нейронами и один из которых нет. Внутри слоя каждый нейрон соединён с несколькими ближайшими нейронами с помощью синаптических связей. Слои соединены между собой с помощью продольных синапсов, проходящих через все слои, к которым присоединена часть синапсов каждого слоя. К каждому из продольных синапсов присоединена группа нервных клеток, отходящих от соответствующих естественных сенсоров или их мозговых центров (зрение, слух, обоняние, осязание, вкус). Благодаря процессам, протекающим в этом тоненьком слое мозга, живое существо может запоминать поступающую извне информацию, устанавливать ассоциации между разными группами информации и вести когнитивную разумную деятельность.

Алгоритмы НТМ предполагают наличие структур и алгоритмов, которые наиболее достоверно отвечали естественной структуре неокортекса живого существа и процессам, протекающим в нём. Для того, чтобы представить модель и описать логические структуры введём несколько определений и определим их назначение. Каждый нейрон головного мозга представим в виде клетки или ячейки, хранящей синаптические связи с другими клетками и своё состояние, определённое на конечном множестве состояний. Синапсы, проходящие вертикально через все слои неокортекса назовём проксимальными, то есть близкими, так как данные поступающие от сенсоров и идущие по проксимальным синапсам оказывают наибольшее влияние на нейроны, подключенные к ним по сравнению с нейронами, соединёнными между собой. Синапсы, соединяющие между собой клетки внутри слоя назовём дистальными, то есть удалёнными, так как влияние клеток друг на друга оказывает эффект лишь в совокупности — изменение состояния одной клетки не может оказать существенного влияния на всю структуру. Клетки, подключенные с проксимальным синапсам в совокупности с самим синапсом назовём колонкой. Таким образом, мы получаем то, что неокортекс можно представить в виде огромного набора колонок, состоящих в среднем из 10000 клеток. Данную совокупность колонок назовём регионом. Регион — набор колонок, колонка — набор клеток.

Модель темпоральной иерархической памяти состоит из пространственного и темпорального группировщиков. Пространственный группировщик позволяет привести получаемые с сенсоров данные к разреженному представлению. Это репрезентация необходима для возможности запоминания большего количества информации без потери её актуальности. Темпоральный группировщик строит внутренние представления на основе данных полученных от пространственного группировщика.

Благодаря тому, что алгоритмы пространственного и темпорального группировщика работают стохастично, мы не можем в произвольный момент времени сказать какие клетки будут активными, а какие не активными.

Рассмотрим работу пространственного и темпорального группировщиков. С сенсоров поступает информация представленная битовым набором. Каждый из проксимальных синапсов получает информацию только от отдельного участка исходного битового образа. Количество активных битов на входе проксимального синапса является параметром, отвечающим за возможность колонки стать активной — у каждой колонки есть пороговое

значение подключенных активных битов. Превосходя это значение колонка переводится в активное состояние и может участвовать в алгоритме темпорального группировщика. После установления активных колонок мы прибегнем к подавлению других колонок от лучшей колонки в радиусе её подавления. После этого в работу вступает темпоральный группировщик, который непосредственно работает с активными колонками и в зависимости от поступившего сигнала и прошлого состояния клетки изменяет или не изменяет их состояния. На выходе — в верхних слоях региона мы получаем абстрактные представления информации, где каждый из слоёв указывает наиболее вероятную и актуальную последовательность появления следующей информации.

Фактически, в представленной структуре клетки, изменяя своё состояние активируют или деактивируют синапсы, к которым они подключены и по которым сигналы проходят по всему региону.

Список литературы

1. Хогинс Д. Об интеллекте / Хокинс Д., Блейкли С. – М.:Издательский дом «Вильямс» - 2007.
2. Hierarchical Temporal Memory including HTM Cortical Learning Algorithms [Электронный ресурс]. - 2011. - Режим доступа: <http://numenta.org/cla-white-paper.html>.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ БАЛАНСА ГРАФА ССЫЛОК

Кааль С.В. – студент, Перепелкин Е.А. – д.т.н., профессор
Алтайский государственный технический университет (г. Барнаул)

Введение

Развитие сети Интернет привело к появлению новых математических задач, связанных с обработкой больших коллекций гипертекстовых документов. Коллекцию гипертекстовых документов принято описывать в виде ориентированного графа – графа ссылок или веб-графа. Обычно на веб-графе строятся алгоритмы упорядочения результатов поиска в поисковых системах Интернета. Наиболее известные алгоритмы ранжирования гипертекстовых документов на основе веб-графа – *PageRank* и *HITS* [1]. Некоторые модификации этих алгоритмов используются в различных поисковых системах, например, *PageRank* используется в поисковой системе *Google*.

Данная работа посвящена созданию программной системы, реализующей алгоритм ранжирования вершин ориентированного графа на основе модели баланса вершин графа [2]. Проводится сравнение этого алгоритма с алгоритмами *PageRank* и *HITS*. Представлены описание программной системы и результаты численного эксперимента.

Модель баланса

Рассмотрим простой ориентированный граф G (без кратных дуг и петель), состоящий из n вершин. Вершины будем обозначать их номерами $i = 1, 2, \dots, n$; дуги – парами номеров (i, j) . Необходимо определить значимость вершин графа в виде пары неотрицательных чисел (p_i, q_i) , где p_i – «ранг лидера», q_i – «ранг посредника». Потребуем, чтобы выполнялось условие нормирования для рангов

$$\sum_{i=1}^n p_i = 1, \quad \sum_{i=1}^n q_i = 1. \quad (1)$$

Обозначим через m_i^- полустепень захода, через m_i^+ полустепень исхода i -й вершины. Модель баланса вершин графа строится на следующих предположениях:

1. Чем больше m_i^- , тем больше значимость вершины i как лидера и чем больше m_i^+ , тем больше значимость вершины i как посредника.
2. Значение p_i определяется значениями q_j вершин j , для которых существуют дуги (j, i) . Значение q_i определяется значениями p_j вершин j , для которых существуют дуги (i, j) .
3. Степень влияния q_j на p_i обратно пропорциональна полустепени исхода вершины j , степень влияния p_j на q_i обратно пропорциональна полустепени захода вершины j .

Учитывая сказанное, модель баланса рангов вершин графа можно записать в виде системы линейных алгебраических уравнений:

$$p_i = \alpha \sum_{j=1}^n a_{ij} q_j + (1 - \alpha) / n; \quad (2)$$

$$q_i = \beta \sum_{j=1}^n b_{ij} p_j + (1 - \beta) / n, \quad (3)$$

где $0 < \alpha < 1, 0 < \beta < 1$ – параметры модели; коэффициенты $a_{ij} \geq 0, b_{ij} \geq 0$ определяют вклад ранга вершины j в ранг вершины i .

Обозначим через L матрицу смежности графа G . Построим матрицу $\bar{L} = L + E_n$, где E_n – единичная матрица порядка n . Коэффициенты a_{ij}, b_{ij} свяжем с полустепенями исхода и захода вершины j следующим образом:

$$a_{ij} = \frac{\bar{L}_{ji}}{m_j^+ + 1}, \quad b_{ij} = \frac{\bar{L}_{ij}}{m_j^- + 1},$$

где \bar{L}_{ij} – элементы матрицы \bar{L} .

Перейдём к векторно-матричной форме записи уравнений баланса. Составим матрицы:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix}$$

размеров $n \times n$ и векторы

$$p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}, \quad q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix}, \quad e_n = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

размера n . Тогда система уравнений (2), (3) запишется в виде

$$p = \alpha Aq + (1 - \alpha)e_n / n; \quad (4)$$

$$q = \beta Ap + (1 - \beta)e_n / n. \quad (5)$$

В работе [2] показано, что система уравнений (4), (5) имеет единственное решение. Это решение удовлетворяет условию нормирования (1).

Расчёт рангов вершин графа можно проводить последовательно по формулам

$$p(k+1) = \alpha Aq(k) + (1 - \alpha)e_n / n; \quad (6)$$

$$q(k+1) = \beta Bp(k) + (1 - \beta)e_n / n; \quad (7)$$

где k – номер итерации.

При любых начальных значениях $p(0)$, $q(0)$ последовательность (6), (7) сходится к единственному решению системы уравнений (4), (5).

Параметры $0 < \alpha < 1$ и $0 < \beta < 1$ гарантируют существование единственного решения уравнений баланса и сходимость алгоритма.

Программная реализация

Программная система была разработана на языке программирования C++. При этом использовались библиотека линейной алгеброй *uBLAS*, которая является подмножеством библиотеки *Boost* [3]. В этой библиотеке есть классы разреженных матриц.

Программное обеспечение включает в себя:

1. основной модуль;
2. модуль ввода и редактирования данных;
3. модуль расчёта;
4. модуль визуализации данных.

Основной модуль является точкой входа в программу, связывает другие модули, а также содержит основные команды для работы с программным обеспечением.

Модуль ввода и редактирования данных используется для ввода данных непосредственно или с подготовленного файла, а также для генерации тестовых данных.

Модуль расчёта предназначен для расчёта рангов вершин с помощью предоставленных алгоритмов.

Модуль визуализации данных используется для демонстрации в удобной для пользователя форме результатов расчёта.

Взаимодействие модулей приведено на рисунке 1.

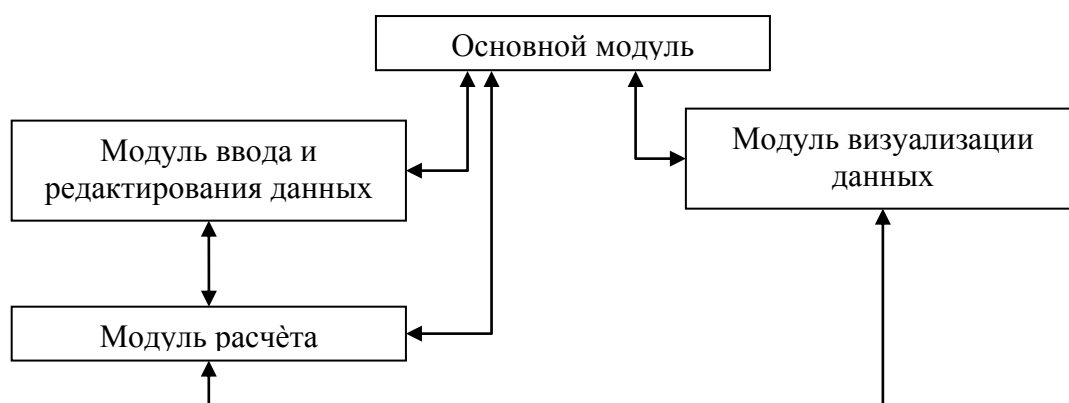


Рисунок 1 – Взаимодействие модулей программной системы.

Численный эксперимент

Рассмотрим граф из 1000 вершин в виде матрицы смежности, полученный с помощью датчика случайных чисел. Двадцать процентов элементов матрицы составляют единицы, остальные – нули. Проведём расчёт рангов вершин данного графа с помощью алгоритма баланса ориентированного графа (6), (7), алгоритмов *PageRank* и *HITS* [1,2]. Для удобства ссылки алгоритм модели баланса (6), (7) будем называть алгоритмом *BG*. Параметры α , β , d в алгоритмах *BG* и *PageRank* зададим равными 0,8. В таблице 1 представлены результаты расчётов для первых десяти наиболее значимых вершин и их ранги лидера.

Таблица 1

<i>BG</i>		<i>PageRank</i>		<i>HITS</i>	
Ранг	Вершина	Ранг	Вершина	Ранг	Вершина
0,0011548	301	0,00115133	946	0,00120265	946
0,00115396	946	0,00114878	301	0,00119733	301
0,00113982	767	0,00114139	767	0,00118393	673
0,00113909	195	0,00113998	256	0,00117938	576
0,00113795	673	0,00113632	295	0,00117723	195
0,00113624	256	0,00113509	673	0,0011727	767
0,00113376	576	0,00113278	195	0,00116546	256
0,00113207	480	0,00113217	480	0,00116335	295
0,00113176	295	0,00112602	513	0,00116275	513
0,00112692	513	0,00112552	576	0,0011622	480

Наиболее значимой вершиной алгоритм *BG* показал вершину 301. Алгоритмы *PageRank* и *HITS* показали наиболее значимой вершиной вершину 946.

Список литературы

1. Ланде Д.В., Снарский А.А., Безсуднов И.В. Интернетика: Навигация в сложных сетях: модели и алгоритмы. – М.: ЛИБРОКОМ, 2009.
2. Перепёлкин Е. А. Модель баланса ориентированного графа // Информационные технологии. 2004. № 10. С. 19-24.
3. Описание работы с библиотекой Boost::uBLAS [электронный ресурс] // Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики. [сайт]. - Режим доступа: http://aco.ifmo.ru/el_books/numerical_methods/library/uBlas_docs/index.htm.

ПЛАГИН ВИЗУАЛИЗАЦИИ ДАННЫХ ДЛЯ ФРЕЙМВОРКА SYMFONY 2

Турубарова Е.А. – студент, Крайванова В.А. – к.ф.-м.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Популярность использования информационных ресурсов из всемирной сети Интернет растёт, так же как и объемы информации различной тематики. Совершенствование технологий упростило сбор и хранение данных, а сеть подразумевает мгновенный доступ к ним. Это изобилие знаний и сведений может быть кладезем информации, будучи правильно представленным. Замечено, что сами по себе данные не представляют интереса для большинства людей, наиболее интересна информация, которую можно извлечь из этих данных, история, которую они способны рассказать. При рассмотрении больших массивов данных наиболее вероятно восприятие отдельных величин, а при визуализации возможен более широкий взгляд на информацию. Неудивительно, что на сегодняшний день в новостных компаниях существуют специальные отделы, которые занимаются только визуализацией данных [1].

Целью данной работы является разработка программного обеспечения в форме плагина для фреймворка Symfony 2, осуществляющего визуализацию данных. Плагин предоставляет следующий функционал:

- генерация визуализации в форматах: pdf, картинка и интерактивный;
- генерация нескольких видов диаграмм: столбчатой, линейной и круговой;
- диаграммы имеют настраиваемые параметры, такие как цвета и толщина составляющих компонентов, ширина и высота диаграммы;
- при формировании pdf существует следующие варианты: только диаграмма, диаграмма и таблица с исходными данными, только таблица с данными;
- реализован алгоритм масштабирования диаграмм с сохранением корректного отображения подписей.

Среди существующих плагинов для данного фреймворка не найдено предоставляющего единый интерфейс для одних и тех же данных с возможностью генерации нескольких форматов визуализации.

Плагин Symfony 2 представляет собой организованный согласно определенным соглашениям набор файлов (PHP файлы, таблицы стилей, JavaScript файлы, сопутствующие медиа файлы), которые вместе реализовывают завершённый функционал. Обычно для создания плагина используются так называемые сервисы. **Сервис** - это универсальное, независимое от конкретных платформ или языков программирования понятие для обозначения объекта, выполняющего определенную работу, "глобальную" для всего приложения [2].

Разработанный плагин визуализации данных использует идею сервисно-ориентированных приложений, он по сути является сервисом. На рисунке 1 представлена схема взаимодействия: тестовое приложение использует сервис визуализации, отправляя ему конфигурацию (параметры) и данные (объект-запрос из базы данных), и получает визуализацию в одном из трех форматов. Разработанный сервис содержит классы для генерации каждого формата визуализации. Класс генерации картинок в png использует распространенную PHP библиотеку GD для работы с графикой [3]. В этом же классе реализован алгоритм масштабирования с корректной отрисовкой подписей по осям. Класс генерации pdf использует PHP библиотеку TCPDF. Эта библиотека предоставляет достаточно много вариантов использования, но данный класс использует формирование pdf из готовой картинки (сформированной в классе генерации png) для работы с диаграммой и из

html-таблицы для выгрузки исходных данных. Класс генерации интерактивного js-компонента использует JavaScript-библиотеку под названием d3js. Отличительная черта этой библиотеки в том, что манипуляция данными в ней основывается на объектной модели документа (Document Object Model - DOM), предоставляя свой упрощенный интерфейс для DOM API.

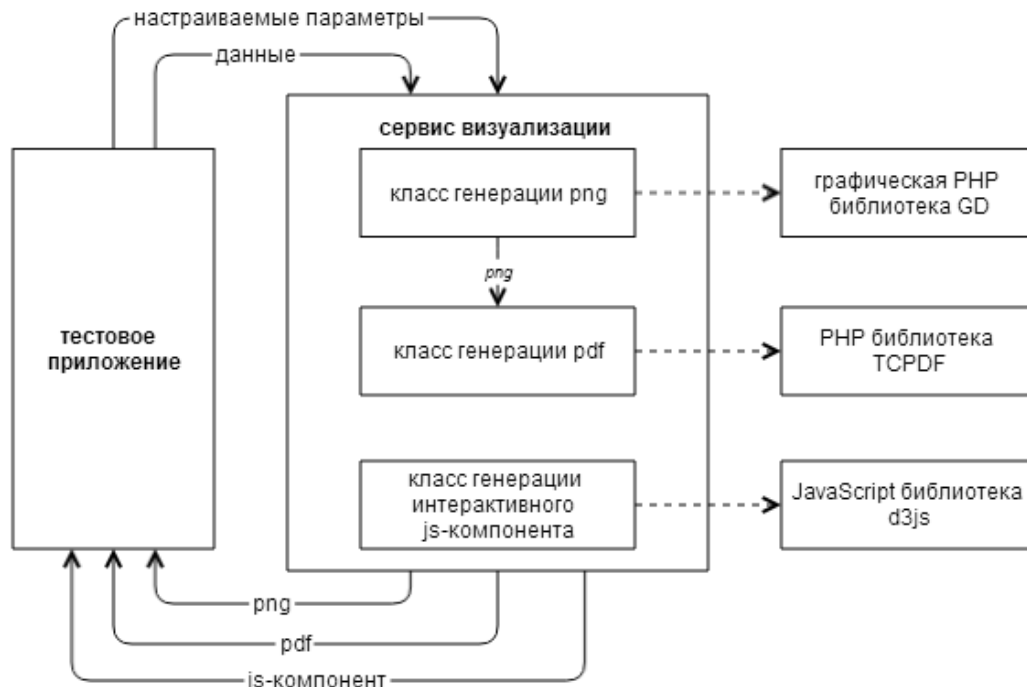


Рисунок 1 – Схема взаимодействия сервиса визуализации и тестового приложения

Для демонстрации работы плагина было разработано тестовое приложение, которое представляет собой опрос: страница со списком вопросов, откуда осуществляется переход на страницы для каждого вопроса с вариантами ответов, а также страница с визуализированными с помощью разработанного плагина результатами голосования.

Таким образом, основные достоинства данного плагина визуализации данных в предоставлении достаточно широкого функционала, в возможности его гибкой настройки и отделяемости от приложения в качестве независимого сервиса.

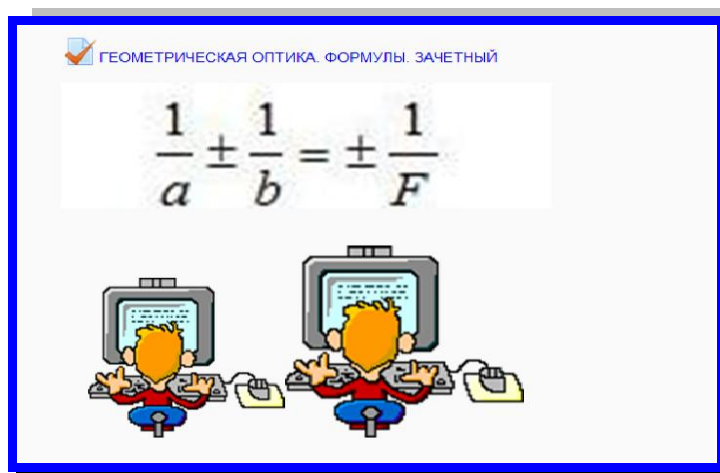
Список литературы

1. Нейтан Яу "Искусство визуализации в бизнесе. Как представить сложную информацию простыми образами." Москва: Манн, Иванов и Фербер, 2013 - 352 с.
2. Service Container [Электронный ресурс]. - Режим доступа: http://symfony.com/doc/2.2/book/service_container.html.
3. Обработка изображений и GD [Электронный ресурс]. - Режим доступа: <http://www.php.net/manual/ru/book.image.php>.
4. TCPDF. PHP class for generating PDF documents [Электронный ресурс]. - Режим доступа: tcpdf.org.
5. D3 Data-Driven Documents [Электронный ресурс]. - Режим доступа: <http://d3js.org/>.

ДИСТАНЦИОННАЯ ПОДДЕРЖКА КУРСА

Гриднева А.Е., Корягина С.И., Моисеева А.А., Шевелев И.В., Яковенко Р.А. – студенты
Рогозин К.И. – к.х.н., доцент, Сорокин А.В. – к.т.н., доцент
Алтайский государственный технический университет (г. Барнаул)

Развитие сетевых технологий, особенно базирующихся на использовании Интернета, стимулировало интенсивные поиски новых способов обучения физики. Это связано, прежде всего, с тем, что указанные технологии позволяют доставлять и репрезентировать значительно большие объемы учебного контента (включая контрольно-измерительные материалы), чем в случае традиционных методов обучения. Кроме того, работа в Интернете является интерактивной, предполагающей от обучаемых выполнения действий в реальном времени (принятия решений) в конкретных ситуациях, визуализирующихся на экране монитора. Это делает процесс обучения активным, формирующим требуемые организаторами учебного процесса компетенции, как «способности применять результаты обучения в определенном контексте».



Данная статья посвящена дистанционной поддержке студенческого курса раздела «оптика», который был проведён в осеннем семестре 2013/2014 учебного года в Алтайском государственном техническом университете им. И.И. Ползунова. Эта поддержка осуществлялась через сайт Центра развития электронных образовательных ресурсов МГУ им. М.В. Ломоносова distant.msu.ru.



Она являлась обязательной частью курса, включающего и традиционные технологии обучения (лекции, лабораторные работы и практические занятия). Вес заданий, выполняемых дистанционно, в общем рейтинге студентов составлял 50 процентов.

Сам курс был раздел на три части: Геометрическая, Волновая и Квантовая оптика. Предлагаемые тесты были четырех типов в соответствии с 4 видами представления физического контента в следующей последовательности:

1. «Понятия», предполагающие оценку владения основными понятиями, законами и правилами, выраженными, прежде всего, вербально;
2. «Формулы». Эти тесты основаны на проверке усвоения символьных обозначений физических величин и взаимной связи между ними.
3. «Задачи». При прохождении данного типа тестов необходимо было владеть как понятийным, так и символьным аппаратом представления физических моделей при решении конкретных учебных заданий.
4. «Компьютерное моделирование». Задания строились на основе выполнения динамических моделей физических феноменов.

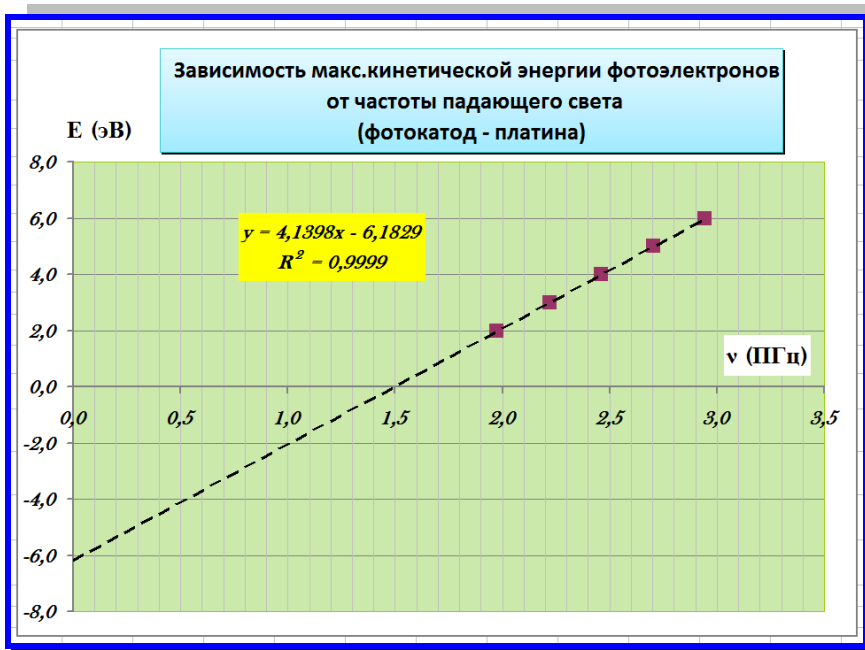
Таким образом, для успешного прохождения дистанционной части курса требовалось выполнение 12 специализированных тестов и одного исследовательского, который был необязательным и предлагался для студентов, которые имели желание получить дополнительные баллы к итоговому рейтингу.

Мы понимаем условность деления тестов первых трех типов, однако, такой подход позволяет пройти учебный материал на несколько раз, каждый раз фокусируясь на определенном, важном для понимания физических феноменов аспекте. Такой подход показал себя эффективным и востребованным студентами.

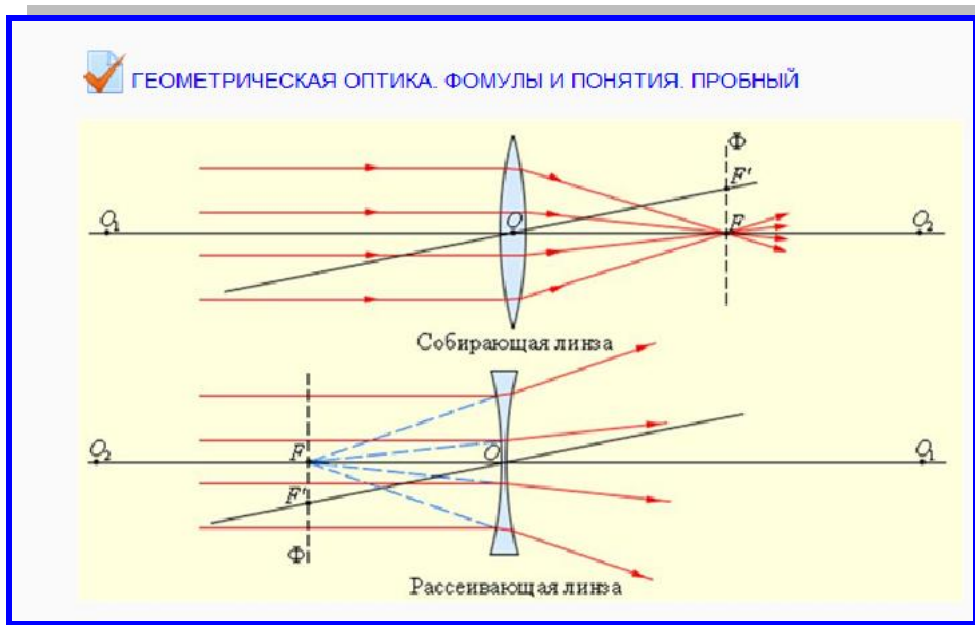
Тесты первых трех типов обычно включали в себя до 50 заданий, на прохождение которых давалось до 90 минут. Особенность подготовки к тестам заключалась в том, что перед включением «зачетного» давался «пробный», в котором выкладывался весь банк задач на «понятия», примеры представления заданий на «формулы» и образцы заданий на «задачи». Необходимость введения данных «пробных» тестов связана с тем, что студенты должны ознакомиться с идеологией и способами представления заданий.

ЭКСПЕРИМЕНТАЛЬНЫЕ ДАННЫЕ			
№ n/n	λ (нм)	ν (ПГц)	E (эВ)
1	152	1,974	2,0
2	135	2,222	3,0
3	122	2,459	4,0
4	111	2,703	5,0
5	102	2,941	6,0
РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТА			
№ n/n	h (фэВ*с)	A (эВ)	ν_0 (ПГц)
1	4,14	6,18	1,49

В «пробных» тестах студенты имели в интерактивном режиме по несколько попыток. Это предложение было востребовано студентами, особенно той их частью, которые ставили своей целью получение 100 баллов. Пробное тестирование открывалось за сутки до зачетного (обычно в субботу). Реальные тесты выставлялись по воскресеньям.

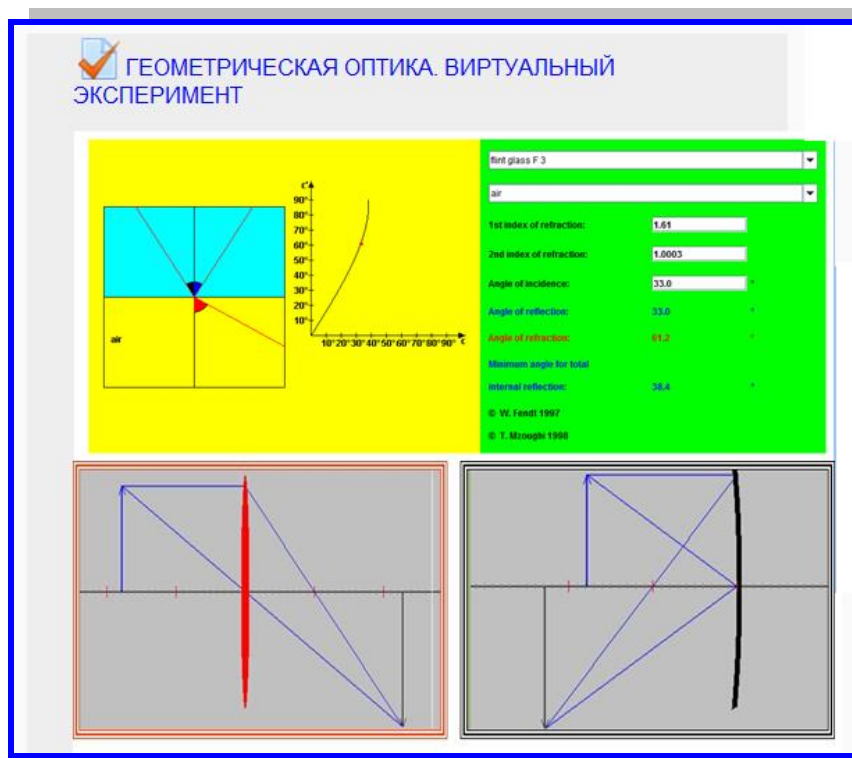


Особый вид тестов связан с использованием симуляций физических процессов. Нами с этой целью выбраны 2D «апплеты», написанные на Java и созданные в разных странах мира. Обычно данный вид тестов включал 8 заданий, в ходе выполнения которых студентам необходимо было создать на экране монитора требуемую учебную ситуацию, в рамках которой требовалось снять до 9 измерений со встроенных виртуальных средств измерения. В отличие от предыдущих типов тестов выполнение этого типа тестов не ограничивалось несколькими часами. На них давалось несколько дней (от 3 до 5). Это было обусловлено индивидуальными психологическими различиями студентов, которые обладают разной скоростью освоения новых знаний и приобретения требуемых навыков.



Для студентов, изъявивших получить дополнительные баллы в итоговом рейтинге, было предложено исследовательское задание, основой которого являлся виртуальный физический эксперимент – 2D апплет «Фотоэффект», созданный в Колорадском государственном университете. Необходимо было найти зависимость максимальной кинетической энергии

фотоэлектронов от частоты падающего света. С этой целью проводилась серия измерений, в ходе которой определялись значения длины волны падающего света по заданному набору значений кинетической энергии.



Полученные результаты использовались для вычисления частоты света. В программе Excel строился требуемый график с одновременным выводом уравнения линии тренда, а также его надежности аппроксимации. Перед студентами было поставлено условие: указанная надежность должна быть не ниже 0,999. Коэффициенты регрессии использовались для получения значений постоянной Планка (в эВ*с и Дж*с), работы выхода (в эВ и Дж), а также красной границы фотоэффекта (в ПГц и нм).

Указанные 6 значений и являлись ответами задания. Кроме того, все участники должны были выслать скрин-шот проведенных измерений. Задание выполняла треть студентов потока (59 из 181), из них 52 набрали 87 и более баллов.

Не менее важной особенностью проведенной дистанционной поддержки курса являлось то, что при разработке заданий участвовала группа волонтеров, которые сами являлись студентами этого же потока. По нашему мнению, невозможно создать эффективные инструменты обучения для студентов без участия самих студентов. Преподаватели и обучаемые – представители разных поколений, лишь совмещенных во времени. Первые имеют знания и опыт. Вторые воспринимают мир по-другому и имеют свой жизненный опыт.

Поэтому принципиально важно совместить возможности преподавателя с восприятиями студентов. В создании и функционировании предложенного ресурса участвовали 7 студентов-волонтеров. Их роль в реализации проекта была решающей: они не только предлагали дизайн и текст заданий, но и предлагали стратегию их использования. Они определяли количество заданий и попыток, последовательность, время прохождения заданий и режим (интерактивный или отложенный), добавляли свои эмоции в виде видео или анимаций и т.д. Не менее важным является то, что они вели работу со своими товарищами

как лично, так и через социальные сети, ведя блоги. В своих обращениях к ним волонтеры говорили: «95 баллов - это не результат. Вы можете сделать свою работу блестящей и совершенной. Мы верим в вас. Вы достойны результата в 100 баллов!».

Результат говорит сам за себя: средний балл по предложенным тестам обычно составлял не менее 80, а число студентов, набравших высший балл - не менее трети.

После завершения курса проводился социологический опрос. Был задан вопрос: является ли нужным подобное сопровождение традиционного обучения, и было ли оно полезным для подготовки к лабораторному практикуму, практическим занятиям и экзаменам. 97 процентов ответили «Да». Это говорит о том, что предлагаемый способ работы является востребованным самими обучаемыми.

Список литературы

1. Trego M. What is Active Learning? URL: <http://www.ecu.edu.au/>
2. Rogozin K. Physics learning instruments of the 21st century. WCPE 2012/ Book of Abstracts. P. 236.
3. Кузнецов С.И., Рогозин К.И. Тестирование как инструмент обучения. Гарантии качества профессионального образования.- Барнаул: Изд-во АлтГТУ, 2012. с. 106-110.
4. Taşar M.F., Bilici S.C., Pinar F. An Interview with David Hestenes: His life and achievements. *Eurasia Journal of Mathematics, Science & Technology Education*, 2012. 8 (2), 139-153.
5. Wells M., Hestenes D. & Swackhamer G. A modeling method for high school physics instruction. *American Journal of Physics*, 1995, 63, 606–619.
6. Rogozin K., Kuznetsov S., Kondrashova D., Lisina I. Cognitive test as a tool for physics learning WCPE 2012/Book of Abstracts. P. 88.

ПРОБЛЕМА ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ ЧАСТИЦ ПО МОНОХРОМНОМУ ИЗОБРАЖЕНИЮ ПРОЦЕССА СВАРКИ

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

Аннотация. В данной работе рассмотрен метод определения частиц на монохромной фотографии процесса сварки. Так как пролетающая частица оставляет практически линейный след на снимке, то проблема сводится к поиску отрезков на монохромном изображении.

Много задач, связанных с обработкой изображений, появляются в областях, где необходимо обрабатывать большие объемы информации сразу же по мере ее поступления. Например, когда идет сварка, необходимо определить скорость и угол направления движения частиц по фотографиям процесса. Пример фотографии представлен на рисунке 1. На изображении четко видны отрезки, которые при определенной выдержке фотографии представляют собой пути движения летящих частиц.

Необходимо выделить наибольшее количество таких отрезков, определить их скорость и угол относительно горизонта (рисунок 2). При решении этой задачи следует отметить, что на всех монохромных изображениях есть только следы частиц и больше ничего.

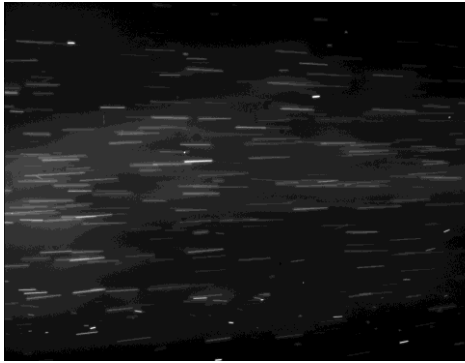


Рисунок 1 – Следы частиц

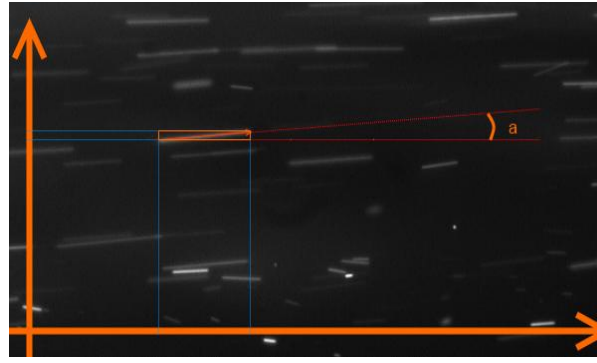


Рисунок 2 – α - угол относительно горизонта

Преобразуем монохромное изображение в бинарное, затем каждую связную область белого цвета (такая область, по предположению, должна быть одним отрезком) ограничим прямоугольником наименьшей площади (рисунок 2), стороны которого будут параллельны координатным осям, и одна из диагоналей этого прямоугольника будет содержать необходимую нам информацию (длина и угол относительно горизонта). Такое преобразование называется бинаризацией. Бинаризация бывает двух типов: с глобальным порогом и с адаптивным порогом. Рассмотрим их.

Бинаризация с глобальным порогом

Порог — это уровень яркости, который позволяет отнести некоторый пиксель изображения либо к пикселям фона, либо к пикселям объектов на фоне:

$$P_{i,j} = \begin{cases} 1, & P_{i,j} \geq T \\ 0, & P_{i,j} < T \end{cases}$$

где $P_{i,j}$ - уровень яркости пикселя с координатами (i,j) , $0 \leq P_{i,j} \leq 1$, T – порог, $0 \leq T \leq 1$, 1 и 0 — уровни яркости объектов и фона после преобразования соответственно.

Рассмотрим задачу выбора порога T . Тогда, зная некоторый порог T , можно легко провести бинаризацию изображения. Проблема осложняется тем, что освещенность на всем изображении неравномерная. Примеры плохо и достаточно хорошо выбранных порогов представлены на рисунках 3 и 4 соответственно.

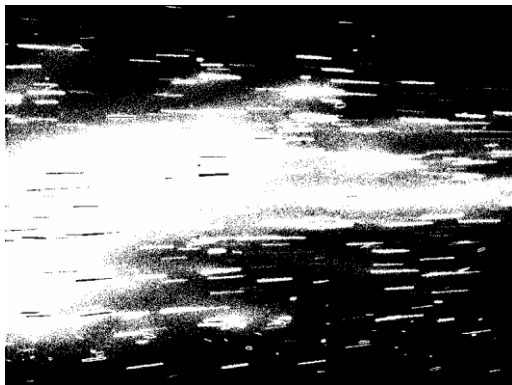


Рисунок 3 – Неправильно выбранный порог для бинаризации

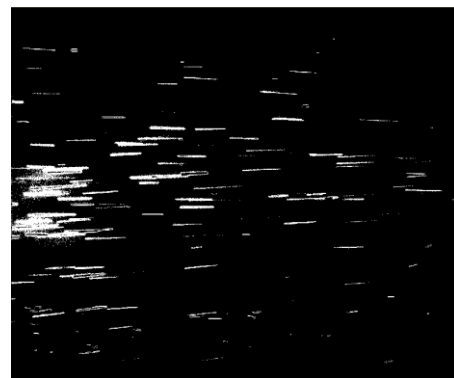


Рисунок 4 – Бинаризация с глобальным порогом (лучший результат)

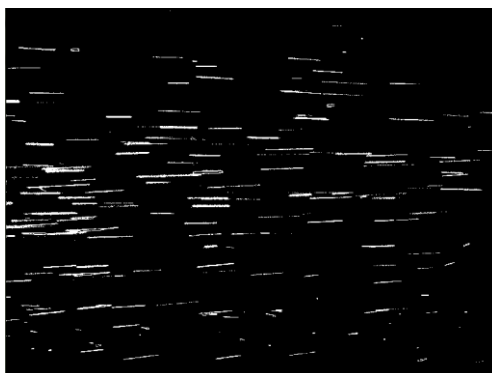


Рисунок 5 – Адаптивная бинаризация

Тем не менее, даже при выборе оптимального порога [1] для бинаризации мы получаем не самые хорошие результаты. На рисунке 4 представлен лучший результат, который мы получили с помощью бинаризации с глобальным порогом. Если взять порог чуть больше, то увеличится шум, если взять порог чуть меньше, то можно потерять значимые части отрезков (попросту говоря, уменьшится их длина).

Бинаризация с адаптивным порогом

Бинаризация с адаптивным порогом основана на разбиении всего изображения на участки небольшого размера, и далее на каждом участке выполнении бинаризации с автоматическим выбором порога. Такое преобразование позволяет точнее отделять объекты от фона, потому что на меньших участках освещенность равномернее, чем на всем изображении. На рисунке 5 представлен результат бинаризации с адаптивным порогом. Это действительно очень хороший результат. Но, тем не менее, здесь есть недостатки: некоторые видимые отрезки не являются связными областями. Примем этот результат за приемлемый и продолжим дальше решать задачу о распознавании отрезков.

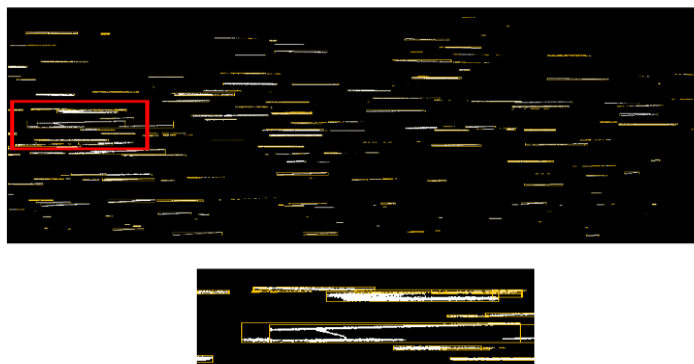


Рисунок 6 – Выделение связных областей

После бинаризации выделим связные области (рисунок 6). Чтобы выделить связные области прямоугольником наименьшей площади, рассмотрим наше изображение как массив из нулей и единиц, где ноль — это фон. Тогда все связные области с единицами, по предположению, являются отрезками. Каждую область из единиц легко рассмотреть в отдельности и определить в ней четыре граничных точки: (x_{\max}, y) , (x_{\min}, y) , (x, y_{\max}) , (x, y_{\min}) , где координаты с индексами \max и \min — максимальные и минимальные значения координат соответственно. По этим точкам и построим прямоугольник, параллельный осям координат, который будет точно содержать все точки нашей области и не более того.

На рисунке 6 видна очень грубая ошибка: сразу три отрезка попали в один прямоугольник. Выясним причину произошедшего. Все отрезки на оригинальном изображении имеют различную яркость, это вызвано различной освещенностью и различными z-индексами¹ отрезков. Ошибка произошла потому, что два непересекающихся отрезка были пересечены третьим с другим z-индексом. Необходимо избавиться от таких случаев.

Рассмотрим оригинальное изображение как функцию $z = f(x,y)$, $0 \leq x < W$, $0 \leq y < H$, $0 \leq f(x,y) \leq 1$, где z - уровень яркости пикселя с координатами (x, y) , W , H - ширина и высота изображения соответственно [2]. Участок графика этой функции представлен на рисунке 7.

Теперь поставим источник света над горизонтом изображения (рисунок 8). Хорошо освещенными станут самые яркие отрезки. При этом проблема пересечения двух ярких отрезков третьим с более слабой яркостью решится, потому что отрезок с более слабой яркостью будет разделен тенью от ближайшего к источнику света яркого отрезка.

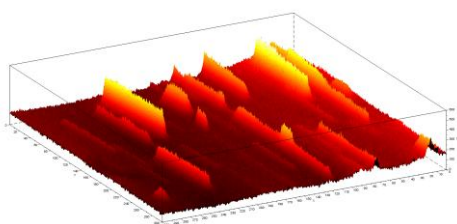


Рисунок 7 – График функции $f(x,y)$ для части изображения. Яркие отрезки находятся выше всех

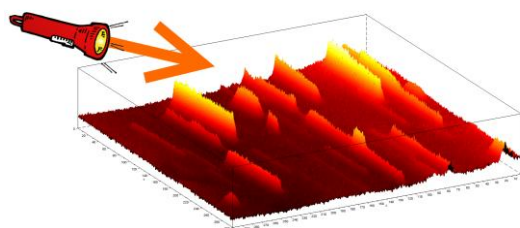


Рисунок 8 –Подсветим отрезки

Результат применения этого алгоритма к оригинальному изображению представлен на рисунке 9.

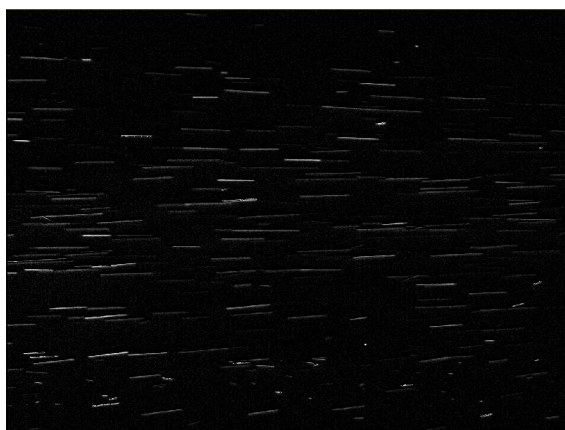


Рисунок 9 – Подсвеченные края

¹ Z-индексом отрезка в данном контексте называют число, которое определяет расстояние от объектива фотокамеры до пролетающей частицы.

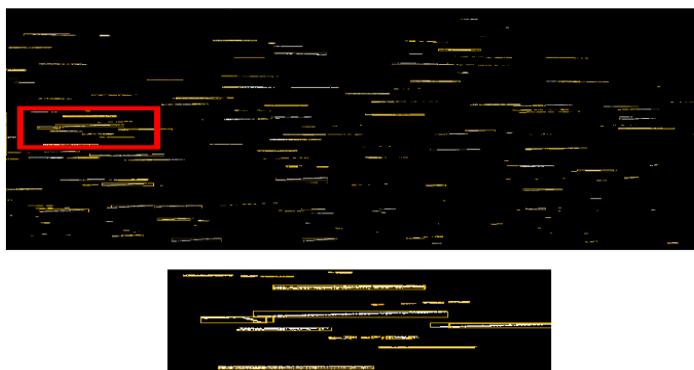


Рисунок 10 – Адаптивная бинаризация и выделение связных областей

Далее применяем адаптивную бинаризацию и выделяем связные области (рисунок 10). Видно, что этот результат лучше предыдущего.

Предложенный алгоритм позволяет избавиться от неравномерной освещенности, но отделение объектов от фона с помощью адаптивной бинаризации на данном этапе происходит с ошибками: некоторые отрезки разбиваются на кусочки, а некоторые вообще не отделяются от фона.

В общем и целом, получен метод выделения отрезков на изображении, но он не выдает точные результаты без участия человека, т. е. корректировать выделение отрезков придется самостоятельно.

Литература

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений.-М.: Техносфера, 2005.- 1072 с.
2. Кашкин В.Б., Сухонин А.И. Дистанционное зондирование Земли из космоса. Цифровая обработка изображений: Учебное пособие. - М.: Логос, 2001. - 264 с.: ил.

КОНТЕНТ-ЗАВИСИМОЕ ИЗМЕНЕНИЕ РАЗМЕРОВ ИЗОБРАЖЕНИЯ

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

Введение

В последние годы в нашей жизни появилось много различных устройств вывода изображений – стали доступнее широкоформатные телевизоры, проекторы, получили распространение различные виды портативных устройств: планшеты, смартфоны, игровые консоли и многое другое. Но чем разнообразнее ассортимент подобной техники, тем больше встречается различных разрешений и соотношений сторон дисплея. Обычно картинка с отличающимися пропорциями «вписывается» в размер экрана и при этом может выглядеть мелко и занимать далеко не всю площадь. Встает вопрос: как выводить одно и то же изображение на разных дисплеях так, чтобы можно было видеть всю картинку с максимальным использованием пространства вывода.

Методы решения

К решению этой проблемы можно подходить с разных сторон. Самый простой метод – «растянуть» изображение до размера экрана без сохранения пропорций (рис. 4). Если пропорции будут значительно отличаться от оригинала, картинка будет сильно искажена, просмотр портретов теряет весь смысл при растянутых лицах (рис. 8). Другой метод – кадрирование изображения или обрезка краев – имеет большой недостаток: невозможно сохранить важные детали по краям изображения.

Один из способов адаптации изображения под размеры окна вывода основан на принципе изменения размеров на основе содержимого изображения. Метод, предложенный Shai Avidan и Ariel Shamir [1], имеет название Seam Carving. В данной работе предлагается анализ эффективности алгоритма Seam Carving при обработке пейзажных изображений. Идея заключается в сопоставлении каждому пикселю картинки определенного значения – энергии, меры значимости в изображении. Группы пикселей с высокой энергией интерпретируются как важные области, которые нужно сохранить.

Описание алгоритма

Введем простую метрику для пространства цветов RGB: пусть есть два цвета, каждый из которых задан тремя целыми числами – $c_1 (r_1, g_1, b_1)$ и $c_2 (r_2, g_2, b_2)$, вычислим так называемое «манхэттенское» расстояние $d(c_1, c_2)$ по формуле (1). Пусть, для определенности, нас интересует изменение цвета в строке, тогда можно задать формулу энергии для пикселя: $E = d(l, m) + d(r, m)$, где m – цвет заданного пикселя, l и r – цвета пикселей слева и справа от него соответственно (для крайних будем считать что цвет пикселя за границей изображения равен цвету соседнего).

$$d(c_1, c_2) = |r_1 - r_2| + |g_1 - g_2| + |b_1 - b_2| \quad (1)$$

Рассмотрим работу алгоритма, если необходимо уменьшить изображение по ширине. Будем искать такой путь по пикселям от верхней стороны изображения до нижней, чтобы сумма энергий этих пикселей была минимальна. При этом в каждой строке будет выбран один пиксель, а пиксели в соседних строках будут располагаться в одном или соседних столбцах. После нахождения минимального по сумме энергий пути такой путь удаляется, таблица энергий пересчитывается, и процесс повторяется до достижения необходимой ширины.

Рассмотрим методы поиска таких путей. Очевидно, что из-за больших размерностей изображения неприменимы переборные алгоритмы, имеющие экспоненциальную сложность. Методы динамического программирования имеют полиномиальную сложность, поэтому рассмотрим один из приемов эффективного вычисления пути.

Будем последовательно обрабатывать строки сверху-вниз, пусть $dp[i][j]$ – сумма на минимальном пути до j -го пикселя в i -й строке, $p[i][j]$ – номер столбца, из которого мы пришли в этот пиксель, т.е. предыдущий пиксель на пути в (i, j) – это $(i - 1, p[i][j])$. Поскольку подразумевается, что все предыдущие строки мы уже обработали и знаем минимальные пути для каждого из их пикселей, то для вычисления минимального пути для пикселя (i, j) нужно перебрать не более трех вариантов: $(i - 1, j - 1)$, $(i - 1, j)$ и $(i - 1, j + 1)$. Найдем минимальный из них и запишем в $dp[i][j]$ сумму, а в $p[i][j]$ – номер столбца для восстановления пути.

Пересчет для следующего пикселя имеет в себе всего три варианта, удаление имеет такую же сложность, как и поиск, таким образом, одна итерация алгоритма имеет сложность $O(NM)$, где N и M – размеры изображения.

Примеры обработанных изображений



Рисунок 1 – Оригинал 1



Рисунок 2 – Карта энергии 1

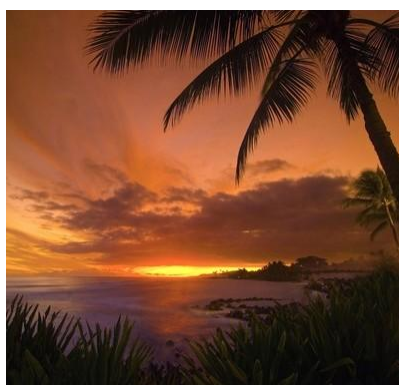


Рисунок 3 – Seam carving 1

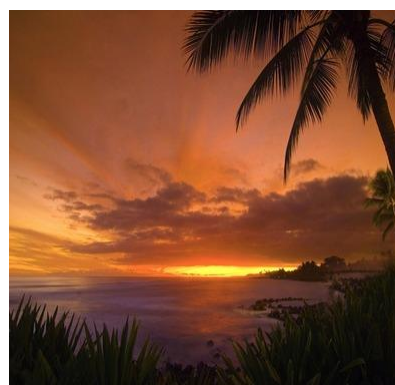


Рисунок 4 – Изменение пропорций 1



Рисунок 5 – Оригинал 2



Рисунок 6 – Карта энергии 2



Рисунок 7 – Seam carving 2



Рисунок 8 – Изменение пропорций 2

Заключение

Данный метод имеет очень широкий простор для модификации: можно задавать различные функции энергии, использовать предварительный анализ изображения, например распознавание лиц, для изменения значений энергии на интересующих областях, или делать это вручную. Также можно увеличить допустимую кривизну путей, увеличив максимальное расстояние между соседними пикселями в пути.

Список литературы

1. Avidan S., Shamir A. ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2007 – NY, ACM, 2007.
2. Hua H. Real time content-aware image resizing - Science in China Series F / Hua H, TianNan F., ROSIN Paul L., Chun Q. - Information Sciences, SCIENCE IN CHINA PRESS, 2009.